Zebra RFID – EtherNet/IP

Sample Application



User Guide

ZEBRA and the stylized Zebra head are trademarks of Zebra Technologies Corporation, registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners. © 2021 Zebra Technologies Corporation and/or its affiliates. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements.

For further information regarding legal and proprietary statements, please go to:

SOFTWARE: <u>zebra.com/linkoslegal</u> COPYRIGHTS: <u>zebra.com/copyright</u> WARRANTY: <u>zebra.com/warranty</u> END USER LICENSE AGREEMENT: <u>zebra.com/eula</u>

Terms of Use

Proprietary Statement

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

Product Improvements

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

Liability Disclaimer

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

Limitation of Liability

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

Contents

List of Figures				
List of Tables	9			
About This Guide	10			
Introduction	10			
Chapter Descriptions	10			
Related Documents and Software	10			
Notational Conventions	11			
Getting Started	12			
Introduction	12			
Requirements	12			
Hardware Requirements	12			
Software Requirements	12			
Setting Up the FX9600 Reader with EtherNet/IP Application	12			
Update the Reader Software	13			
Install EtherNet/IP License	13			
Install and Run the EtherNet/IP Application	14			
Setting Up Sample App with Studio 5000 Logix Designer and PLC Controller				
Installing Studio 5000 Logix Designer				
Installing Zebra RFID Add-On Profiles Extension				
Installing Zebra RFID Module using EDS File				
Using RFID-EIP Sample Application				
Using Non-compatible Versions of Studio 5000 Logix Designer and PLC				
Importing Rungs				
Configuring the Application				
Setting the Reader IP With the Module	25			

Configure For the Appropriate	Controller 2	25
Updating the Configuration for	MSG Instructions	26
Creating Application using Ger	neric EtherNet/IP (without AOP)	27
Creating New Project with Corr	rect PLC Controller	28
Adding and Configuring EtherN	let/IP Module	29
Configuring EtherNet/IP Modul	e	31
Configure I/O Assembly in	Module in Module Creation Window by Providing As	S-
	ctures	
	Stop Inventory Using I/O Assemblies	
- · ·	ntroller	
Configure Driver for EtherNet/I	P and Find Controller using RSLinx	39
Set the Controller Path in Proje	ect 4	42
Load Project to PLC Controller	·	43
PEID Configuration and Operations Using the	e EtherNet/IP (EIP) Sample App	16
c 1	Operation	
-		
	e in the Reader	
00		
-		
3 0		
C C		
5		
o j		
5		
•		
-		
Access Response Data		I I
Error Codes	-	72

Error Codes	72
EtherNet/IP Stack Error	72
RFID Operation Specific Error	74
roubleshooting	76
Unable to Load Sample Application	76
	EtherNet/IP Stack Error RFID Operation Specific Error

Setting Proper Requested Packet Interval (RPI)	76
I/O Not Responding Status	77
Application EtherNet/IP Adapter Application Not Running	77
Reader IP Not Configured/Reachable	77
Communication Path Not Set Properly In MSG Instructions	77
Unable to Perform Inventory or Access Operation	78
Response Data format	78

List of Figures

Figure 1: Reader Administration Console	13
Figure 2: License Manager	14
Figure 3: User Application Page	14
Figure 4: EDS Hardware Installation Tool Option	17
Figure 5: New Empty Project	19
Figure 6: Controller Properties	19
Figure 7: Change Ethernet/IP Mode	20
Figure 8: Linear IP Mode	20
Figure 9: Module Type	21
Figure 10: Module Definition	22
Figure 11: Newly Added RFID Module	22
Figure 12: MainProgram - MainRoutine	23
Figure 13: Import Rungs	24
Figure 14: Import Rungs Window	24
Figure 15: Module Properties	25
Figure 16: Change Controller	26
Figure 17: Configuration Dialog Box	27
Figure 18: Communication tab	27
Figure 19: Studio 5000 Application	28
Figure 20: New Project Window	28
Figure 21: New Project Defaults	29
Figure 22: Create New Module	30
Figure 23: Select Module Type	30
Figure 24: New Module Configuration	32
Figure 25: RPI Selection	32

List of Figures

Figure 26: New Data Type	33
Figure 27: Inventory Command Data Structure	33
Figure 28: STATUS_MASK Data Structure	34
Figure 29: TIME_STAMP_UTC Data Structure	34
Figure 30: TAG_REPORT_12B Data Structure	35
Figure 31: INVENTORY_RESPONSE_12B Data Structure	35
Figure 32: Controller Tag	37
Figure 33: Ladder Logic	39
Figure 34: RSLinx Classic Window	40
Figure 35: Configure Drivers	40
Figure 36: Driver Configuration	41
Figure 37: Who Active Dialogue	42
Figure 38: Project Path	43
Figure 39: Communications > Dowload	43
Figure 40: Download Confirmation	44
Figure 41: Download Status	44
Figure 42: Set Remote Run	45
Figure 43: Green Status Modes	45
Figure 44: Rung Instructions for Reading Capabilities	46
Figure 45: ENABLE.GetReaderCaps flag = TRUE	47
Figure 46: ReaderCaps Data	47
Figure 47: Rungs Performing an Operation	50
Figure 48: Specifying the Profile Index Number	50
Figure 49: Controller Tags	51
Figure 50: Obtaining the Number of Profiles	51
Figure 51: Number of Available Profiles	52
Figure 52: Active Profile Instance from Profile List Class	52
Figure 53: Active Profile Instance	53
Figure 54: Get and Set Antenna Configuration	54
Figure 55: Specify Antenna ID	54
Figure 56: Antenna Configuration Values	55
Figure 57: Antenna ID to Modify	55
Figure 58: Pre-filter Explicit Message	56

Figure 59: Specify the Filter Index	57
Figure 60: PreFilterConfig Tag	57
Figure 61: Modify the MSG Instruction	58
Figure 62: Delete a Pre-filter	58
Figure 63: AccessPostFilter Operation	59
Figure 64: AccessPostFilter Tag	60
Figure 65: Configure Trigger Values	61
Figure 66: Trigger Values	61
Figure 67: GPIO Configuration	62
Figure 68: GPIOConfig Tags	63
Figure 69: Generate Events	64
Figure 70: Set Event Type	65
Figure 71: Inventory Operation	66
Figure 72: Green Enabled Status	66
Figure 73: Modifying OutputInventoryCommand	66
Figure 74: Reading Inventory Data	68
Figure 75: Stop Inventory Operation	68
Figure 76: Access Operation/Read Access Data	69
Figure 77: Access Operation	69
Figure 78: Access Read Operation	70
Figure 79: Access Write Operation	70
Figure 80: Access Response Data	71
Figure 81: Module Properties Window	77
Figure 82: Module Properties Window	78
Figure 83: Style Window	79

List of Tables

Table 1: Rung Instructions	46
Table 2: Profile List Explicit Message Data Model	49
Table 3: OutputInventoryCommand Field Descriptions	67
Table 4: EtherNet/IP Stack Errors	72
Table 5: RFID Operation Specific Errors	74

About This Guide

Introduction

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



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

Chapter Descriptions

Topics covered in this guide are as follows:

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

Related Documents and Software

The following documents provide more information.

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

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

About This Guide

Notational Conventions

The following conventions are used in this document:

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

Getting Started

Introduction

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

Requirements

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

Hardware Requirements

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



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

Software Requirements

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

Setting Up the FX9600 Reader with EtherNet/IP Application

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

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

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

Update the Reader Software

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

Figure 1 Reader Administration Console

stics ler	Reader Administra	ation Console	Help Check Status Check Status Check Status Check Status
Welcome to the	FX Series (8 port) Reader Administration Consol	e.	NXP Custom Operation Statistics Events Statistics Other Custom Operation Statistics
	Reader Software Version :	3.6.22	General Reader Configuration Read Point Configuration Advanced Antenna Configuration
d d	Reader Host Name :	FX9600EFB2F6	Region Configuration Certificates
	Reader Network IP Address :	10.17.131.178	Inventory and Read Tags Communication Settings LLRP SNMP
	Reader Serial Number :	17300010505713	Windess Services Services
	USB Port Status :	No Device Found	Date and Time Settions IPSec Settions Chance Password
	Power Source Type :	DC Supply	GPIO Settings User Application Deployment
	Power Negotiation Status :	Disabled	Manuse Profession the reader Ermane Version Information Ermane Version Information System Load Conference System Conference Statution Error Messages Tarto Park Software
			Into Party Someate

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

Download the latest software for the FX9600 Reader at:

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

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

Install EtherNet/IP License

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

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

Figure 2 License Manager

					H License Manager
on Statistics		License	Manager		License manager page enables user to acquire, release and list the available ficense provided they are valid. Il license present are levelid or no loces is available. License Unavailable the with be shown with the missione to proceen the license.
s instann assword ons re Log b		Manage License Operation: Device: Install and Run application: Server Type: Server URL: Activation ID:	Acquire Acquire ON-Line ON-Line Production Server Intps://zebra-licensing.flexinetop Activate	eration	Instance Licenses - User can equite the face on other or other Organization large - Advancements Organization large - Advancements Organization large - Advancements Organization large - Advancements Advancements - Advancements Advancements Advancements Advancements Advancements Advancements Advancements Advancements Advancem
Uom T	e kolez Lizerod Name no-festure-sthemetip	License Version	License(s) Erginy Dute License Gwart I	Hot 0 Praco	
					_

Install and Run the EtherNet/IP Application

The EIP application can be installed one of two ways:

- along with the license by enabling the check-box Install and Run application or
- by installing the application with the web-console as a Debian package.

To install the application as a Debian package:

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

us and an and a second s		Applications This page provides the details of installed application and also to it
eration Statistics	User Application Page	applications in the reader.
d Tags d Tags mmunication t Time ec	Existing Packages:	List of installed Apps - This drop down menu shall list the cur packages installed in the roader. Start/Stop - The mage deplays the running status as indicate below. Cick the image to logge the status. Indicates App is running
ise Manager Meta Dat	<u>a</u>	Indicates App is NOT running
		AutoStart - Selecting this check box shall enable the application
tions	- Install New Package:	run at startup. Uninstall - Shall remove the package from reader.
are I I I I I I I I I I I I I I I I I I I	Current Package file is selected	 How to create packages: Packages can be created using any standard debain package creation tools or manually. The guideline package creation for FX Series reader are listed below, The package shall contain timary executable compatible of ELF
m	package: zebraethernetip_1.0.9.deb Browse	LSB executable, ARM, version 1, GNU Linux. 2. The name of the binary executable must match the name of the
	Install	package, excluding version name. For example, if the package na package, <u>12,1,a</u> l (package-1 version 2.1), then the name of bina executable must be package-1. There can be more binaries apart
		be above said over the package. 3. The package share there as share package. 3. The package share the said the package share the terms of the package share the said the package share the said the package share t

Figure 3 User Application Page

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

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

刹 ZEBR	A	الله، ZEB	RA
Hone Status • Operation Statistics • Configure Status • Configure Status • Communication Date Time IP Sec License Manager Change Password CHO Podles • Firmware • System Log Diagnostics Shutdown Logout	H Deraton Successful Startis Startis Aufostart Uninstall Jerzenterheneitig Startis Startis Aufostart Uninstall Jerzenterheneitig Package Mama zebraterheneitig Startis Versenterheneitig Startis Versenterheneitig	Home Status > Operation Statistics > Configure Reader Read Tags > Communication Date Time II Sec Liconse Manager Change Password GPIO Applications Profiles > Firmware > System Log Diagnostics Shutdown Logout	H List of Installed apps List of Install Vew Package: List is use install List is use install List is use install List is use installed List is use



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

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

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

Installing Studio 5000 Logix Designer

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

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

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

Installing Zebra RFID Add-On Profiles Extension

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

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

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

Download the Zebra RFID AOP extension at:

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

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

Installing Zebra RFID Module using EDS File

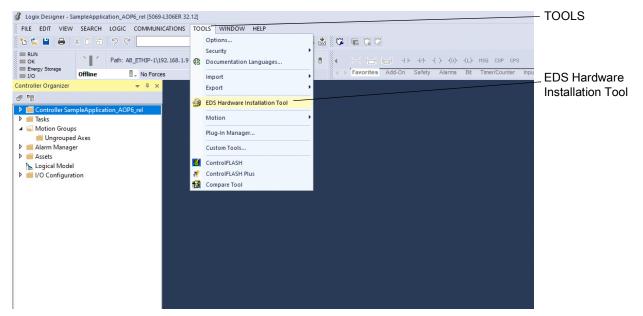
For the PLC that does not support AOP, use the EDS file to configure the Zebra RFID EtherNet/IP on the PLC. The EDS file comes with the FX9600 Industrial Ethernet Software package, which is available at:

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

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

To install the EDS file using Studio 5000 Logix Designer:

- 1. Select EDS Hardware Installation Tool from the TOOLS drop-down menu.
- Figure 4 EDS Hardware Installation Tool Option



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

Using RFID-EIP Sample Application

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



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

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

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

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

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

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

ổ New Project						?	×
Project Types				Search			×
💰 Logix	⊳ Co	mpact GuardLogix®	5370 S	afety Contr	oller		Î
🕥 View	⊳ Co	mpact GuardLogix® mpactLogix™ 5370 mpactLogix™ 5380 (Controlle	er	oller		
		5069-L306ER			380 Controller		
		5069-L306ERM	Compa	ctLogix™ 5	380 Controller		
		5069-L3100ERM	Compa	ctLogix™ 5	380 Controller		
		5069-L310ER	Compa	ctLogix™ 5	380 Controller		
		5069-L310ERM	Compa	ctLogix™ 5	380 Controller		
		5069-L310ER-NSE	Compa	ctLogix™ 5	380 Controller		
		5069-L320ER	Compa	ctLogix™ 5	380 Controller		~
	Name:	Zebra_EIP_Sam	pleAppli	catin			
	Locatio	on: C:\Users\vk253	4\Docur	nents\Stud	io 5000\Proj+ ~	Brow	se
		Cancel		Back	Next	Fini	ish

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

ổ New Project					?	\times
5069-L306ER Cor Zebra_EIP_SampleApp	mpactLogix™ 5380 (olicatin	Controller				
Revision:	32 v					
Security Authority:	No Protection			~]	
	Use only the select authorization	ed Security Aut	thority for authe	ntication and		
Secure With:	Logical Name <cor< p=""></cor<>	ntroller Name>				
	Permission Set			U		
Description:						
		Cancel	Back	Next	Fin	ish

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

Figure 5 New Empty Project

💰 Logix Designer - Z	ebra_EIP_Sar	mpleAppli	icatin [506	9-L306ER 32	.11]																-	□ ×
FILE EDIT VIEW	SEARCH	LOGIC	COMMU	NICATIONS	TOOLS	WINDOW	HELP															
🏠 🖆 🔛 🖨	х О б	26				v 🐤 🕈	5 🎜 🛛	D= 15a	6	• 🛎			17	16 (G)	2							
	N I	Path: <r< th=""><th></th><th></th><th></th><th></th><th></th><th></th><th>€ #</th><th></th><th>4 F</th><th></th><th></th><th></th><th></th><th></th><th>MSG COP CP</th><th></th><th></th><th></th><th></th><th></th></r<>							€ #		4 F						MSG COP CP					
Energy Storage			Vo For		▶. No I		æ		* 00										Compare	Compute/Math	Move/Looic	al File/Migr
Controller Organizer	Offline	1	1↓ No For	ces		u x	1 ,					avontes	Roditoni	Safety	Additio	DIL	TillenCounter	Inpuboutput	Compare	Compatermatin	noverLogic	al Thermise
					Ť	4 X																
A Controller Zeb	- FID Com	al a Anna Ka																				
Controller Zeb		рієдрріїс	atin																			
Controller	Fault Handl	ler																				
Power-Up	Handler																					
▲																						
▶ 5 MainPr	rogram																					
📕 Unschedul																						
 Motion Group Ungrouped 																						
Alarm Manage																						
👂 💼 Assets																						
he Logical Model																						
 I/O Configurat Empty 5069 Backp 																						
[0] 5069	9-L306ER Ze	bra_EIP_S	ampleApp	plicatin																		
🔺 💑 A1, Etherne																						
4 💑 A2, Etherne	306ER Zebra	i_EIP_Sam	pleApple	New Mo	dule																	
	306ER Zebra	_EIP_Sam	pleApp	Discover	Modules.																	
_			×	Cut		Ctrl+X																
				Сору		Ctrl+C																
			6.	Paste		Ctrl+V																
				Delete		Delete																
				Cross Re	ference	Ctrl+E																
				Export M	lodule																	
						Alt+Enter																τ ₽ ×
Description				Propertie	0		ors															
Major Fault Minor Fault				Print					0 Errors	<u> </u>		0 Warnings	0	01	Messages				Search			ρ
Willion Fault																						^
<						>																~
Te	- B																					Ŷ

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

Figure 6 Controller Properties

Nonvolatile	Memory	Capa	acity	Interne	t Protocol	Port	Configuration	S	ecurity	Alarm Log
General	Major Fa	ults	Minor	Faults	Date/Tim	e	Advanced	SFC	Execution	Project
Vendor:	Rockw	ell Autor	mation/Al	en-Bradle	у					
Туре:	5069-L	306ER (Compactl	.ogix''' 53	80 Controller				Change C	ontroller
Revision:	32.011									
Name:	Zebra	EIP_Sa	mpleAppl	icatin						
Description:								\wedge		
Chassis Type	<none< td=""><td>\$</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></none<>	\$								
Slot:		_						Ť		
		-								
EtherNet/IP Mode:	A1/A2:	Dual-IP							Change I	P Mode

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

Figure 7 Change Ethernet/IP Mode

Change EtherNet/IP Mode	×
Current mode:	A1/A2: Dual-IP
New mode:	A1/A2: Linear/DLR \checkmark
Move I/O and MSG Paths to port:	\sim
OK Car	Help

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

Figure 8 Linear IP Mode

🗳 Logix Designer - Zebra_EIP_SampleApplicatin [5069-L306ER 32.11]*								
FILE EDIT VIEW SEARCH LOGIC COM	MUNICATIONS TOOLS WIND	OW HELP							
ର 🖕 🖬 🖶 🗶 ମିନ୍ଦ୍ର 🗖 🖉	~ •	🏂 🖉 📴 ha	D 16 📩		🕫 🛍 G				
III RUN	, ,		100 00 000		5 m 40 C				
■ OK Path: <none></none>			🐐 🍰 🗉	< H H H H H	al +1+ +/+	-()(U)(L)) HSG COP CPS		
Energy Storage	Forces	æ						Dutput Compare Compute/	/ath Move/Logical File/N
	+ 4 ×	HV.				·			
Controller Organizer	₹ # X								
d) 12									
🔺 📹 Controller Zebra_EIP_SampleApplicatin									
Controller Tags									
Controller Fault Handler									
Power-Up Handler									
A A MainTask									
A MainProgram									
Unscheduled									
▲									
Ungrouped Axes									
Alarm Manager									
Assets									
be Logical Model									
 I/O Configuration Employed and Employed and									
Soos Backplane Soos Backplane Soos Backplane	Applicatio								
▲ So A1/A2, Ethernet	Applicadin								
5069-L306ER Zebra_EIP_SampleAp	olicatin								
Description		Errors							▼ 9
Major Fault		0	0 Errors	0 Warnings	0	0 Messages		Search	\$
Minor Fault									
<	,								
🗈 Controller Organizer 🖹 Logical Organizer	· · · · ·								
		<			_				>
Search Results 🚜 Watch									
roject saved to Recovery file.								ication Software: RSLinx Class	

- **12.** Add the new RFID module to the project for RFID assembly operation by right clicking **Ethernet Controller** under **I/O Configuration** in the left plane.
- 13. Select New Module to open a module selection dialogue with all available modules.

14. In the filter text box, enter FX9600 to display the Zebra specific RFID Reader Module (FX9600).

Figure 9 Module Type

ect Module Type			
atalog Module Discovery Fa	vorites		
FX9600		Clear Filters	Show Filters ≯
Catalog Number	Description	Vendor	Category
FX9600	RFID Reader	Zebra Technologies	Communication
1 of 546 Module Types Foun	d		Add to Favorites
Close on Create			Create Close Help

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

New Module		×
General*	General	
- Module Info - Internet Protocol - Vendor	Type: FX9600 RFID Reader Vendor: Zebra Technologies Parent: Local Name: FX9600 Description: Zebra RFID-EtherNet IP module Module Definition Revision: 2.001 Electronic Keying: Compatible Module Connection: Inventory Response, Acces	
Status: Creating	OK Cancel	Help

16. Add the module related information.

- a. Name: Name of the module as per your choice
- b. Description: Module Description (optional)
- c. IP Address: IP Address of RFID Reader
- d. The user can also choose the module definition (two available configurations) by clicking **Change...** below **Module Definition**. The **Module Definition** dialogue window displays and the user can choose the connection from the drop-down menu.

Figure 10 Module Definition

New Module					\times
	neral				
Connection Module Info	Module Definition		×		
- Internet Protocol Vendor					
P	Revision:	2 ~ 001 ≑		192, 168, 1,	
D	Electronic Keying:	Compatible Module \checkmark	-		
	Connection:	Inventory Response Standard Size EPC ID, A $ \smallsetminus $		· ·	
		Inventory Response Extended Size EPC ID, Access Inventory Response Standard Size EPC ID, Access	s Response s Response		
-					
			_		
	ОК	Cancel Help			
I					
Status: Creating		[ОК	Cancel	Help

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

- 17. Click **OK**. (In this example, Inventory Response Standard Size EPC ID, Access Response was selected.)
- 18. Click **OK** again in the New Module dialogue window.
- 19. Click on **Close** to close the module selection dialogue. The new Zebra RFID module is added in the project.
- Figure 11
 Newly Added RFID Module

Adde Defined Taps C Obrigance C Obrigance	Logix Designer - Zebra_EIP_SampleApplicatin (5069-L306E	D 32 111x						- 0	1
Solution Solut									
Addet Defined Taps Addet Defined Taps <th></th> <th></th> <th></th> <th>i ma</th> <th>194 IN1 IN1</th> <th></th> <th></th> <th></th> <th></th>				i ma	194 IN1 IN1				
Coto The organization Image: I with a coto Image: I with a cot		- 4 4 -	12* 108 C0 E9 100		Mar Car Sir				
Into Office Implementation Imp	III OK Path: <none></none>		% 👬 B	4 H Hell Hell H	+ +/+ -()(0)(L) MSG COP CPS			
Concluster Image: Concluster Zuhru ED/Sumplikuppication	I Energy Storage Offline I Vo Forces	▶, No Edits 🔒		 Favorites Add-On 	Safety Alarms I	Bit Timer/Counter Input/Outpu	t Compare Compute/Math	Move/Logical	Få
	ontroller Organizer	+ 4 ×							
Concoller Tags Concoll	9 -								
Model & Define It gs Ø FoRS00.01 Ø FoRS00.02 Ø FoRS00.02 Ø FoRS00.02 Ø FoRS00.02 Ø Forson A O Warnops Ø O Messopes Search Ø O Messopes Search Ø Catoria FRD-Ehen Net IP module attus Ø Offine Ø Catoria FRD-Ehen Net IP module Ø C	Controller Tags Controller Tags Controller Tags Controller Tauth Hundler Power-Up blandler Controller Tauth Hundler Power-Up blandler Controller Tauth Hundler Controller Tauth Program Unchrechtelded Controller Tauth Program Controller TauthProgram Controller Tauth Program Contr								
FORMOUT F	Module Defined Tags	^					_		
of Postbolo: of Postbolo: device SPD-EtherNet P module tata: dedule Fault Controller Organizer Secon Facult @ Wath	FX9600:11	8	0 Errors 🔥	0 Warnings 🚺	0 Messages		Search		
g Search Recurits 👼 Watch	FX960012 FX960002 Sevont RFID-EtherNet IP module tatus Offline Aodule Fault	>							
		¢						_	
	Search Results 🔊 Watch								

Importing Rungs

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

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

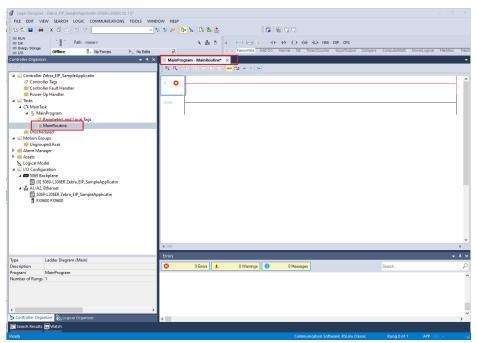


Figure 12 MainProgram - MainRoutine

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

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

Figure 13 Import Rungs

Logix Designer - Zebra_EIP_SampleApplicatin [5069-L306ER 32.11]*										- 0	×
FILE EDIT VIEW SEARCH LOGIC COMMUNICATIONS TOOLS WIND	OW HELP										
1 🖕 🔛 🖶 🔺 🗇 🎧 🤊 🤇 🔍 🗸 •	þ 🐤 🔎 🚺	🗽 🕞 🗞 📩		172	🛍 🛱 🕅	2					
RUN Path: <none></none>		8 4 6		h	- 44 - 4)(U)(L)- M	SS COP CPS				
Energy Storage	2							Compare Compute/Math	Move/Logical	File/Misc.	File/S
Controller Organizer		aram - MainRoutine*									-
		nam - Mainkoutiner 11 To Pl, DS ES									Ť
Controller Zebra_EIP_SampleApplicatin			nu 191 n. v	(46)							
Controller Zebra_cir_SampleApplicatin O Controller Tags	(End)										^
Controller Fault Handler					1						
Power-Up Handler		Cut Rung		Ctrl+X						1	
✓ G Tasks ✓ A MainTask		Copy Rung		Ctrl+C							
A L MainProgram	63	Paste		Ctrl+V							
Parameters and Local Tags		Delete Rung		Delete							
MainRoutine	S	Add Rung		Ctrl+R							
Unscheduled		Edit Rung									
Generation Groups Generation Groups Generation Groups		Edit Rung Comment									
Alarm Manager		Import Rungs									
🕨 💼 Assets		Export Rungs									
To: Logical Model	e	Start Pending Rung Ed	dits Ctr	1+Shift+S							
		Accept Pending Rung									
[0] 5069-L306ER Zebra_EIP_SampleApplicatin	白	Cancel Pending Rung	Edits								
4 🔓 A1/A2, Ethernet	14	Assemble Rung Edit									
5069-L306ER Zebra_EIP_SampleApplicatin	西	Cancel Rung Edit									
1 FX9600 FX9600		Verify Rung									
		Go To		Ctrl+G							
		Add Ladder Element		Alt+Insert							
	1=										~
	_		_	_	_		_		_	_	
Type Ladder Diagram (Main)	Errors									•	÷×
Description	•	0 Errors 🔒	0 Warning	p 🚺	0 M	lessages		Search			ρ
Program MainProgram Number of Runos 0											^
Number of Kungs 0											
<>											
T= Controller Organizer											~
Le controller organizer	<		_							_	>

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

Figure 14 Import Rungs Window

Logix Designer - Zebra_EIP_SampleApplicatin (5069-L306ER 32.11)*					
FILE EDIT VIEW SEARCH LOGIC COMMUNICATIONS TOOLS	WINDOW HELP				
5 🖕 🗎 🖶 🗶 🖻 🍈 🤊 🤆	× * * #	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			
KUN OK Derev Sense		848 ← H ⊟ ⊟ + + + ()		*	
In Drange Storage Offline . No Forces . No Forces	Edits 🔒	Favorites Add-On Alarms Bit	Timer/Counter Input/Output Compare Compute/Math Move/Logical	al File/Misc. File/Shift Sequencer F	
	a × 🗄 MainPr	rogram - MainRoutine* 🛛 🗙			
	લ, લ,	田 医气管炎 🔤 🏧 🖦 🕬			
a 🛁 Controller Zebra_EIP_SampleApplicatin		ENABLE GetReaderCase			ENABLE GetReaderCaps
Controller Tags Controller Fault Handler		ENABLE.GetReaderCaps	MSG Nessage Control Ge	Get_ReaderCaps.DN	ENABLE GetReaderCaps
Power-Up Handler				Get_ReaderCaps.ER COP	
a 🛁 Tasks				Source Get_F	eaderCaps.ERR ErrorCade
🖌 🖓 MainTask				Length	2
A b MainProgram Parameters and Local Tags					
Parameters and Local lags R0 MainRoutine					
Unscheduled		ENABLE.GetAntennaConf	MSG Meaning Control Out A	Get_AntennaConf_DN	ENABLE GetAntennaConf
Motion Groups	- I -			CONCHE Get_AnternaConf.ER COP	
iii Ungrouped Axes ▶ iiii Alarm Manager				Source Get_An	InnaConf.ERR ErrorCode
Assets				Length	2
bs Logical Model					
✓					
Store Backprane [10] 5069-L306ER Zebra_EIP_SampleApplicatin		ENABLE SetArternaConf	MSG	Set_AnternaConf.DN	ENABLE SetAnternaConf
4 💑 A1/A2, Ethernet			interior cardo salo	HON)-	
5069-L306ER Zebra_EIP_SampleApplicatin				FLERO- Source Set Ar	tennaContERR
g Exaeou Exaeou				Dest	ErrorCode 2
		ENABLE GetTriggerConf	MSG	Get_TriggerConf_DN	ENABLE GetTriggerConf
			Inessage Con	(ON)- Ord Trimpford ID	
				Source	Get_TriggerConf
	_			Dest	ErrorCode 2
Type Ladder Diagram (Main)					
Description Program MainProgram					
Number of Rungs 0		ENABLE.SetTriggerConf	MSG	Set_TriggerConf_DN	ENABLE.SetTriggerConf
			wessage control	-(DN) 0.1 Thundard CD	
				Source Set	TriggerConf.ERR
				Dest Langth	ErrorCode
	4=				•
	Errors				≁ å ×
	0	0 Errors 🗼 0 Warnings 🚺 65 Messe	265		Search 🔎
	Creating	tag: 'Set_GPIOConfig'			^
	Creating	tag: 'Set Frefilter' tag: 'Set_FrefilterClass'			
	Creating	tag: 'Set_TripgerConf' tag: 'StartInventory'			
	Creating	tag: 'TriggerConfig'			
<	Import of	mplete - 0 error(s), 0 warning(s)			
The Controller Organizer					, v
Search Results 🙀 Watch					

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

Configuring the Application

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

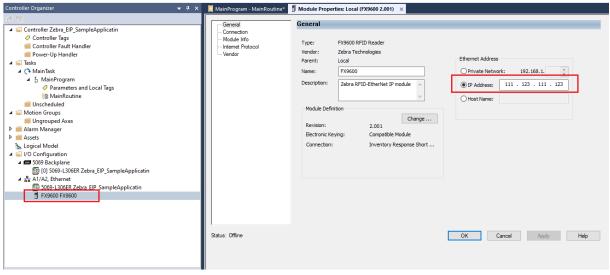
Users setting up the project from scratch do not need to do this as their project is already configured with the reader IP address. In this case, go to Updating the Configuration for MSG Instructions on page 26.

Setting the Reader IP With the Module

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

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

Figure 15 Module Properties



3. Click OK.

Configure For the Appropriate Controller

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

To change the controller in sample application:

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

Figure 16 Change Controller

Controller Organizer 👻 🖣 🗙	MainProgram - MainRoutine*) Module Properties: Local (FX9600 2.001) ×
đ 🖫		
Controller Zehrs, EP, SampleApplicatin Controller Zehr, EP, SampleApplicatin Controller Zehr, Handler Description Description Controller Zehr, Br, SampleApplicatin Discription Description Description Description	General Conection Module Ho Heren Protocol Vendor	Secretal Type: Proposed RFD Rander Worder: 2012 For Enderson Peretts: Und Peretts:
Mgior Fault Minor Fault		DK Cancel Apply Heb

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

Updating the Configuration for MSG Instructions

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



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

To update the MSG instructions with current module configuration:

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

Figure 17 Configuration Dialog Box

	MSG Mess	age Control Get_Re	aderCaps	EN) DN) (ER)
Message Configurat	ion - Get_ReaderCaps			
Configuration Comm	nunication Tag			
Message Type:	CIP Generic	~		
Service Get Attri	bute Single 🗸 🗸	Source Element:		
		Source Length:	0	(Bytes)
Code:	(Hex) Class: 64 (Hex)	Desthation	ReaderCaps	
Instance: 1	Attribute: 1 (Hex)	element.	New Tag	
	ble Walting ⊖ Start	O Done [Done Length: 0	
⊖ Enable ○ Ena ○ Error Code: Error Path: FX3600 Error Text:	ble Wating ◯ Start Extended Error Code:	O Done [Done Length: 0] Timed Out ♥	

2. Select the Communication tab.

Figure 18 Communication tab

Message Conf	iguration - Get_ReaderCaps ×
Configuration	Communication Tag
Path:	FX9600 Browse
O Broadca	Message Path Browser X
Communica	Path: FX9600 FX9600
CIP With Source	□ I/O Configuration (Octal) □-
Connec	
) Enable (OK Cancel Help
⊖ Error Code: Error Path: FX9 Error Text:	Extended Error Code: Timed Out 🕈 600
	OK Cancel Apply Help

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

Creating Application using Generic EtherNet/IP (without AOP)

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

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

Creating New Project with Correct PLC Controller

Create New Project by choosing correct PLC controller.

- 1. Open Studio 5000 Logix Designer application.
- Figure 19 Studio 5000 Application



- 2. Select New Project. The New Project dialog box displays.
- Figure 20 New Project Window

New Project					?	×
Project Types			Search			×
3 Logix	 Compa Compa 					Î
		9-L306ER	CompactLogix™ 5 CompactLogix™ 5			
	506	59-L3100ERM	CompactLogix [™] 5 CompactLogix [™] 5	380 Controller		
		59-L310ERM 59-L310ER-NSE	CompactLogix [™] 5 CompactLogix [™] 5	380 Controller		
	506	59-L320ER	CompactLogix [™] 5	380 Controller		Ŧ
	N <u>a</u> me:	3.00				
	Location:	C:\Users\vk253	4\Documents\Stud	lio 5000\Proji ~	Brow	se

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



💰 New Project				? ×
5069-L306ER Col FX9600_Sample_Ether	mpactLogix™ 5380 Controller ^{:Net_IP}			
Revision:	32 *			
Security <u>A</u> uthority:	No Protection		*	
	Use only the selected Security Auth authorization	ority for authentication and		
Secure With:	Logical Name <controller name=""></controller>			
	<u>Permission Set</u>		<u></u>	
Description:				
			_	
	Cancel	Back Next		Finish

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

🚰 Logix Designer - F/95600_Sample_EtherNet_IP [5056-1306ER 32.11]	- 8 ×
FILE EDIT VIEW SEARCH LOGIC COMMUNICATIONS TOOLS WINDOW HELP	
= 0K 1 Path: <none> 1 & 🚠 0 4 + + + + (1) <00 <00 ></none>	
United States and Stat	
Controller Organizer	
a 🖷	
Controller FX9600_Sample_EtherNet_IP	
Controller Fault Handler	
■ Power-Up Handler → S Taka	
A Q MainTask	
A MainProgram	
iii Unscheduled	
A C Motion Groups	
■ Ungrouped axes	
Assets	
🗽 Logical Model	
▲ 🛁 VO Configuration	
▲ ■ 500 Backplane	
4 品 AL Ethernet	
🔯 5069-L306ER FX9600_Sample_EtherNet, IP	
4 & A2 Ethernet	
10 5069-L306ER FX0600_Sample_EtherNet_IP	
Ta Controller Organizer Bological Organizer	
Eros	▼ # ×
O Errori 🛦 0 Warnings 🚺 0 Messages	Search P

Adding and Configuring EtherNet/IP Module

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

Figure 22 Create New Module

💰 Logix Designer - Fi														
										_				
				COMMUNICAT	rions	TOOLS		WOOW						
) 🐿 🛳 🔛 🖶			26				~	2 2	م .	12: 3	s D	6	幽	02
= RUN		12	Data at	ETHIP-1\192.								*		ä.,
OK Energy Storage											*	80	8	81
ii = 1/0	Off	ine	1	 No Forces 		▶ _↓ No E	dits		£.					8.€ ▶
Controller Organizer				+ 7 ×										
e 🖷														
🔺 📹 Controller FX9	500_S	ample	EtherNet	UP										
Controller														
Controller			er											
Power-Up	Hand	ler												
 Tasks MainTask 														
▶ 5 MainPr														
Unschedul														
🔺 🚍 Motion Group														
Ungrouped		5												
🕴 💼 Alarm Manage	r													
Assets														
Logical Model I/O Configurat														
A B 5069 Backg														
		SER FX	9600 Sam	ple_EtherNet										
🔺 👗 A1, Etherne			Module											
10 5069-L			nt Module											
4 💑 A2, Etherne			ver Module											
5069-L3														
	61	Paste		Ctrl+V										
		Prope	erties	Alt+Enter										
		Print		,										
	_													
<				>										
Bus Size														
<				>	enor	<u> </u>								
🗩 Search Results 🐺	Wate	h												
Create a module														

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

Figure 23 Select Module Type

	Shgw Filters ♥
Description	Vendor
Generic Ethernet Module	Rockwell Automation/A

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

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

Туре:	ETHERNET-MODULE Generic Ethern	et Module			
Vendor:	Rockwell Automation/Allen-Bradley				
Parent:	Local	Connection Para	meters		
Na <u>m</u> e: Description:			Assembly Instance:	Size:	
		Input:		125	(32-bit)
	~	Output:		124	(32-bit)
Comm <u>F</u> ormat:	Data - DINT 🗸 🗸	Configuration:		0	(8-bit)
Address / H	ost Name	<u>configuration</u> .		L.	
IP <u>A</u> ddre	SS:	<u>S</u> tatus Input:			
⊖ <u>H</u> ost Nar	ne:	Status Output:			

5. Please follow next section to configure the module.

Configuring EtherNet/IP Module



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

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

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

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

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

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

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

Example:

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



New Module	2				×
Type: Vendor: Parent: Na <u>m</u> e: Descri <u>p</u> tion:	ETHERNET-MODULE Generic Etherne Rockwell Automation/Allen-Bradley Local FX9600	Connection Par	ameters Assembly Instance: 101		8-bit) 8-bit)
Comm <u>F</u> orma Address / H IP <u>A</u> ddr <u>H</u> ost Na	ess: 192 . 168 . 1 . 9	O <u>u</u> tput: <u>Configuration</u> : <u>S</u> tatus Input: S <u>t</u> atus Output	1		8-bit)
Open Mod	u <u>l</u> e Properties	ОК	Can	cel H	lelp

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



Figure 25 RPI Selection

💰 Logix Designer - FX9600_Sample_EtherNet_IP[5089-1306EF 32.11]*	- 8 ×
BIE EDIT VIEW SEARCH LOGIC COMMUNICATIONS TOOLS WINDOW HELP	
1 1 1 日 日 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
표 NA 1 / Path rome>	
III 10 Offline 🗄 No Forces 🕨 No Edits 🔃 () Favorite's Ado-Un Salety Adams bit Imencounter approvulput compare computeriatin wovelugical results: resolution salety	
Controller Organizer 🗾 👻 🖡 🚺 Module Properties Report: Local (ETHERNET-MODULE 1.001) 🗙	-
General Connection Module Ho	
A G Controller FX9600_Sample_EtherNet_IP	
Controller Tags	
Controler Fault Fander	
A total by tanding of the second seco	
A (> MainTesk	
b 5 MainProgram Image: Second s	
Induces unough Module Fault Module Fault	
> Arm Manager	
D 🖬 Assets	
b Experi Model	
A good Sackplane	
To (0) 5069-L306ER FX9600 Sample EtherNet IP	
A 🖧 A1, Ethernet Status: Offine OK Cancel Apply Help	
5060-L306ER FX9600_Sample, EtherNet, IP	
3 THERNT-MODULE R9800 - & AL, themet	
A ga AJ, truement ∰ 500-1000 EFV5000, Sample, EtherNet, JP	
Modul Orlinea Tags A	
Ø FX9600C	
Description	
Satus Offline Voodel Fault v	
Te Controller Organizer	
Errors	▼ # ×
C DErm A OVersity O Offensage	Q
	P

Create User Defined Data Structures

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

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

To create a new data type:

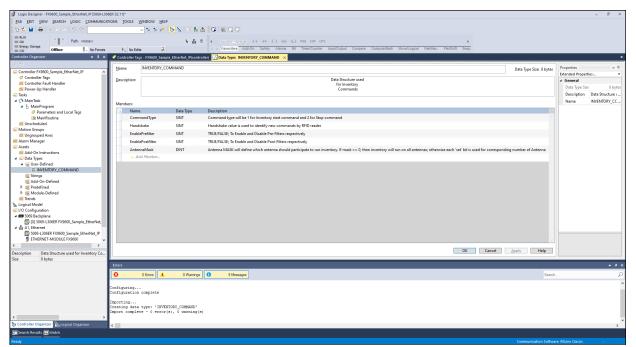
1. In the left pane, navigate to **Assets > Data types > User-Defined**.

Figure 26 New Data Type

💰 Logix Designer - I	FX9600_Samp	le_EtherNet	IP [5069-L306	6ER 32.11]*							
FILE EDIT VIEW	V SEARCH	LOGIC C	OMMUNICAT	TIONS TO	OLS	WINDOW	HEL	P			
1 🛀 🗎 🗦	¥ 🗇 🙃	9 %				~ 5 3	, 5	D= ba	6	🛎 🖁 🕻	2 🛍 C
RUN OK Energy Storage		Path: <no< th=""><td></td><td></td><td></td><th></th><th></th><td></td><th>% #</th><th>•</th><th>H</th></no<>							% #	•	H
≡ I/O	Offline	8.	No Forces	<u> </u>	No Edit	ts	₽,				Favori
Controller Organizer			▼ # ×	Contr	oller Tag	gs - FX9600	_Sam	ple_Ether1	Net_IP(co	ntroller)	×
				Scope:	FX9	9600_Sampl	e. ~	Show: A	II Tags		
Controller FX960		herNet_IP	^	Nam	-						
Controller Tag					(9600:C						
Controller Fa											
Tasks	andier				(9600:1						
A 🔁 Main Task				▶ F)	<9600:C)					
🔺 🔓 MainProg	gram										
	neters and Lo	cal Tags									
🗎 Mainf											
Motion Groups	1										
Ungrouped A	xes		- 11								
📕 Alarm Manager			- 11								
Assets			- 11								
Add-On Instr	ructions		- 11								
Data Types			_		7						
Stri	New Data Ty	oe									
📠 Adc	Import Data	Гуре									
	Cut			Ctrl+X							
Þ 🔝 Mo	Сору			Ctrl+C							
Trends	Paste			Ctrl+V							
I/O Config	Paste With C	onfiguration	n Ctr	l+Shift+V							
🔺 📾 5069 Backpla	ne										
👰 [0] 5069-L	.306ER FX960	0_Sample_E	therNet_								
▲ 꿂 A1, Ethernet											
5069-L30	T-MODULE F		rivet_IP								
A 💑 A2, Ethernet		A3000									
<			> `								
						_					
					lonitor	Tags (Ed	dit Tag	is /			

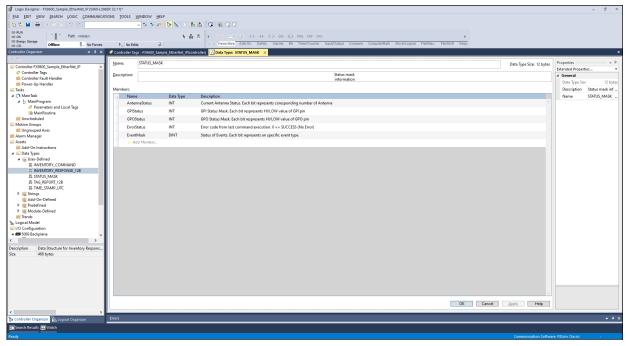
- 2. Right click on User-Defined and select New Data Type.
- 3. Create an Inventory Command Data Structure with the details in the below image.

Figure 27 Inventory Command Data Structure



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

Figure 28 STATUS_MASK Data Structure





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

Figure 29 TIME_STAMP_UTC Data Structure

Logix Designer - FX9600_Sample_EtherNet_IP (5069-L306)	6ER 32.11]	1					- 8 ×
EILE EDIT VIEW SEARCH LOGIC COMMUNICAT	TIONS 1	OOLS WIND	OW HELP				
15 🖆 🔛 🖶 🕹 🏠 🖄 🕫 🔛		~ *	5 🌮 📴 🗽	5 B. 20 10 10 10 10 10 10 10 10 10 10 10 10 10			
ERUN Path: <none></none>			1				
Energy Storage		No Edits	e.	+ > Favorites Add-On Safety Alarms Bit TimeriCounter Input/Output Compare Compute/Math Move/Logical File/Mitsc. File/Shift Sequ			
Controller Organizer - 7 ×				IP(controller) 👪 Data Type: STATUS_MASK 🔣 Data Type: TIME_STAMP_UTC 🔀			+
<i>a</i> 11							
Controller FX9600_Sample_EtherNet_IP	Nam	ie: TIM	E_STAMP_UTC		Data Type Size: 16 bytes	Properties Extended Proper	
Controller Tags	Desc	ription:		Time Stamp in UTC		General	ues •
Controller Fault Handler Power-Up Handler	- I					Data Type Siz	a 16 bytes
Tasks	Merr	nbers:				Description	Time Stamp in L
🔺 🔿 MainTask		Name	Data Type	Description		Name	TIME_STAMP_U
A h MainProgram		wYear	INT	Year			
Parameters and Local Tags MainRoutine		wMonth	INT	Month			
Unscheduled		wDayOfWe	ek INT	Day of the Week			
C Motion Groups		wDay	INT	Date			
iii Ungrouped Axes Alarm Manager		wHour	INT	Hour			
Alarm Manager		wMinute	INT	Minute			
Add-On Instructions		wSecond	INT	Seconds			
✓ G Data Types ✓ G User-Defined		wMilliseco		Mill Seconds			
A Vest Oser-Defined 읢 INVENTORY_COMMAND		Add Me		Milli Seconas			
器 INVENTORY_RESPONSE_12B		÷ Auu Me	mper				
器 STATUS_MASK							
器 TAG_REPORT_12B 器 TIME STAMP UTC							
Strings							
R Add-On-Defined							
▷ I Predefined ▷ I Module-Defined							
Trends							
he Logical Model							
I/O Configuration IIII 5069 Backplane V							
<							
Description Time Stamp in UTC							
Size 16 bytes							
	L			OK Cancel	Appry Help		
Controller Organizer	Errors						- ₽ ×
Search Results Watch							
Ready					Communication Software	e: RSLinx Classic	



Logix Designer - FX9600_Sample_EtherNet_IP [5069-L30		- 6 ×
ELE EDIT VIEW SEARCH LOGIC COMMUNICA		
B ok Path: <none></none>	🐐 🏭 🚺 📢 🖂 금 + + + + < > <0> - <0> - ces	
Energy Storage Offline . No Forces	▶ No Edits R. (+) Favorites Add-On Safety Alarma Bit Timer/Counter Input/Output Compare Compute/Hath Move/Logical File/Mits. File/Shift Sequ	
Controller Organizer 👻 👎 🗙	🛷 Controller Tags - FX9600_Sample_EtherNet_IP(controller) 👪 Data Type: STATUS_MASK 👪 Data Type: TIME_STAMP_UTC	•
0 T	Name: TAG_REPORT_128 Data Tee Sin	A hoter Properties + 0
Controller FX9600_Sample_EtherNet_IP	Name: TAG_REPORT_128 Data Type Size	Extended Properties
Controller Tags Controller Fault Handler	Description: TAG Report data with	▲ General
Power-Up Handler	12 Bytes EPC Length	Data Type Size 64 bytes
Tasks	Members:	Description TAG Report dat:
🔺 🕐 MainTask	Anne Data Type Description	Name TAG_REPORT_1/
 MainProgram Parameters and Local Tags 	TegEPC TAG_EPC_128	
© MainRoutine	TagPC INT	
Unscheduled	TsgCRC INT	
Motion Groups	AntennalD SINT	
Alarm Manager	RSSI SINT	
C Assets	Channelindex INT	
Add-On Instructions	SeenCount INT	
▲ ⊆ Data Types ▲ ⊆ User-Defined	PhaseInfo INT	
器 INVENTORY_COMMAND	FirstSeenTime TIME_STAMP_UTC	
間 INVENTORY_RESPONSE_12B	> Lasteenime TMESTADPUTC	
器 STATUS_MASK 器 TAG_REPORT_12B		
A TIME_STAMP_UTC		
▶ m Strings	Pad Member	
K Add-On-Defined ▶	* Ado Member	
Ig Module-Defined		
iii Trends		
h Logical Model		
I/O Configuration		
< >		
Description TAG Report data with 12 Bytes EPC L		
Size 64 bytes		
	OK Cancel Apply	Help
	OK Cancel Apply	
Controller Organizer	From the second s	+ # ×
Search Results A Watch		
E Search Results K Watch		
Pearly	Communica	tion Software: RSLinx Classic -

Figure 31 INVENTORY_RESPONSE_12B Data Structure

Logix Designer - FX9600_Sample_EtherNet_IP (5069-L306	ER 32.12]*	- 8 ×
FILE EDIT VIEW SEARCH LOGIC COMMUNICAT	IONS TOOLS WINDOW HELP	
ິ 🖕 💾 🖨 🗶 ປັດໄກ 🔍	◆ 5 年 月 日本 5 長 法(は 単 10 円)	
= RUN		
CK Path: A8_ETHIP-1\192.1		
UO Offline	🖡 No Edits 🔒 🤄 Favorities Saflety Alarma Bit Timer/Counter Input/Output Compute/Math Move/Logical File/Mac. File/Shift Sequencer Pr	
Controller Organizer 🚽 🕈 🗙	Data Type: INVENTORY_RESPONSE_128 ×	
0 T	Name (INVENTORY, RESPONSE, 128 Data Ter	ma Sina 464 buter Properties + 0
Controller Tags	Internoring consectors Data lys	/pe Size: 464 bytes Properties + 0 Extended Properties
Controller Fault Handler	Description: Data Structure for	▲ General
Power-Up Handler	Packet for Standard	Data Type Size 464 bytes
4 C MainTask	pokoki to stantaru EPC Length	Description Data Structure f
MainProgram	Members	Name INVENTORY RE
Parameters and Local Tags		
MainRoutine	Name Data Type Description StatusMask STATUS MASK Device specific status information	
Inscheduled		
Ungrouped Axes	PacketSequenceNo SINT Seq number used to identify new response packet	
🕨 💼 Alarm Manager	NumOfReports SINT Total Number of TAG reports in current response packet	
🔺 🖳 Assets	Pad INT Reserved	
Add-On Instructions	TagReports TAG_REPORT_128[7] Detailed TAG Report	
✓ G Data Types ✓ G User-Defined	+ Add Member	
(% INVENTORY_COMMAND		
# INVENTORY_RESPONSE_12B		
(%) INVENTORY_RESPONSE_64B		
器 STATUS_MASK 器 TAG_REPORT_12B		
ai TAG_REPORT_648		
(1) TIME STAMP_UTC		
Strings		
Add-On-Defined		
Improvement Improveme		
Trends		
bs Logical Model		
🔺 🛁 I/O Configuration 🗸 🗸		
< >		
Description Data Structure for Inventory Re		
Size 464 bytes		
	OK Cancel Apply	Help
	£	
< >	Enos	+ ¤ ×
Search Results 🗛 Watch		
Dandy		nmunication Software: DSLiny Classic

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

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

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

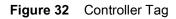
 InventoryResponse: To read the response from input assembly instance, it is required to copy data into a structured data type "INVENTORY_RESPONSE" which is created in previous section. This tag is used to copy input data into that structured data format.

To create a controller tag:

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

RUN OK Path: <none> Energy Storage</none>	8 8 8 4 14				÷				
UO Offline . No Forces	▶ No Edits	rites Add-On Safety Alarms Bit	Timer/Counter Input/Outp	out Compare Compute/Math Move/Logical File/Misc	: File/Shift Sequ				
troller Organizer 👻 👻	# × Controller Tags - FX9600_Sample_EtherNet_IP(controller)	×							
9 <u>11</u>	Scope: Scope: Shgw: All Taga			🗸 🔨 Enter Name	Filer				
Controller FX9600_Sample_EtherNet_IP	Name	Alias For		1				e. 1	1 18
Controller Tags	Name > FX9600:C	Alias For	Base Tag	Data Type AB:ETHERNET_MODULE:C:0	Description	External Access	Constant	Style	-
Controller Fault Handler						Read/Write			
Tasks	FX96001			AB:ETHERNET_MODULE_SINT_464Bytes:10		Read/Write			
A C Main Task	FX9600:0			AB:ETHERNET_MODULE_SINT_8Bytes:0:0		Read/Write			
MainProgram	< >								
iii Unscheduled									
Motion Groups									
🛑 Ungrouped Axes Alarm Manager									
Alarm Manager Assets									
Logical Model									
I/O Configuration									
5069 Backplane									
[0] 5069-L306ER FX9600_Sample_EtherNet_IP									
[0] 5069-L306ER FX9600_Sample_EtherNet_IP									
[0] 5069-L306ER FX9600_Sample_EtherNet_IP & A1, Ethernet									
[0] 5069-L306ER FX9600_Sample_EtherNet_IP A1, Ethernet 5069-L306ER FX9600_Sample_EtherNet_IP									
[0] 5069-L306ER FX9600_Sample_EtherNet_IP A1, Ethernet 5069-L306ER FX9600_Sample_EtherNet_IP THERNET-MODULE FX9600									
[0] 5069-L306ER FX9600_Sample_EtherNet_JP A1, Ethernet S069-L306ER FX9600_Sample_EtherNet_JP ETHERNET-MODULE FX9600 A2, Ethernet									
[0] 5069-L306ER FX9600_Sample_EtherNet_JP A1, Ethernet S069-L306ER FX9600_Sample_EtherNet_JP ETHERNET-MODULE FX9600 A2, Ethernet									
[0] 5069-L306ER FX9600_Sample_EtherNet_JP A1, Ethernet 5069-L306ER FX9600_Sample_EtherNet_JP ETHERNET-MODULE FX9600 A2, Ethernet									
[0] 5069-L306ER FX9600_Sample_EtherNet_JP A1, Ethernet S069-L306ER FX9600_Sample_EtherNet_JP ETHERNET-MODULE FX9600 A2, Ethernet									
[0] 5069-L306ER FX9600_Sample_EtherNet_JP A1, Ethernet S069-L306ER FX9600_Sample_EtherNet_JP ETHERNET-MODULE FX9600 A2, Ethernet									
[0] 5069-L306ER FX9600_Sample_EtherNet_JP A1, Ethernet 5069-L306ER FX9600_Sample_EtherNet_JP ETHERNET-MODULE FX9600 A2, Ethernet									
[0] 5069-L306ER FX9600_Sample_EtherNet_JP A1, Ethernet S069-L306ER FX9600_Sample_EtherNet_JP ETHERNET-MODULE FX9600 A2, Ethernet									
[0] 5069-L306ER FX9600_Sample_EtherNet_JP A1, Ethernet 5069-L306ER FX9600_Sample_EtherNet_JP ETHERNET-MODULE FX9600 A2, Ethernet									
[0] 5069-L306ER FX9600_Sample_EtherNet_JP A1, Ethernet 5069-L306ER FX9600_Sample_EtherNet_JP ETHERNET-MODULE FX9600 A2, Ethernet									
[0] 5069-L306ER FX9600_Sample_EtherNet_JP A1, Ethernet 5069-L306ER FX9600_Sample_EtherNet_JP ETHERNET-MODULE FX9600 A2, Ethernet									
[1] 009-L308EF F79800_Sample_EtherNet_IP [2] 4.1, Ethernet [3] 509-L308EF F79800_Sample_EtherNet_IP [3] 11-ENET-EtherNet_IP [3] 11-ENET-EtherNet_IP [3] 209-L308EF F79800_Sample_EtherNet_IP									
[1] 0 094-1308EF F78900_Sample_EtherNet_IP [2] 4.1 Ethernet [2] 094-1308EF F78900_Sample_EtherNet_IP [2] 11:E181ET-MOULE F78900 [2] 11:E181ET-MOULE F78900 [2] 21:E181ET-MOULE F78900 [2] 21:E181ET-MOULE F78900_Sample_EtherNet_IP	C () \ Wonder Tage \ Edit Tage \] <					_
[0] 5069-L306ER FX9600_Sample_EtherNet_JP A1. Ethernet 5069-L306ER FX9600_Sample_EtherNet_JP ETHERNET-MODULE FX9600 A2. Ethernet)			14					2
[1] 0094-1308EF F79800_Sample_EtherNet_IP [2] 4.1, Ethernet [2] 5094-1308EF F79800_Sample_EtherNet_IP [2] THERNET-NOULE F79800 [3] 4.2, Ethernet [2] 5094-1308EF F79800_Sample_EtherNet_IP) KantorTaga KEdi Taga /						Searc	ħ	_

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



🏫 🔛 🖶 🗶 🏛 🏠 🦻 약 🤇		1 🖌 🕞 🕹	🗱 🕼 🕼						
OK Path: AB_ETHIP-1\1	2.168.1.14				(U)(L)- MSG COP CPS		÷		
Energy Storage Offline . No Force	s 🕨 No Edits 🔒		 Favorites Add-Or 	Safety Alar	ms Bit Timer/Counter Input/Output Compare Co	mpute/Math Move/Logical File	Misc. File/Shift Sequ		
oller Organizer 🚽 👎	Controller Tags - FX9600_Sample	EtherNet_IP(controller) ×						
1	Scope: Transformation Scope: S	now: Al Tags				🗸 🏹 Enter Name i	iter		
Controller FX9600_Sample_EtherNet_IP	Name		 Force Mask 				Constant		
Controller Tags	EnableOperation	III - Value	Force Mask	 Style Decimal 	Data Type BOOL	Description	Constant		
Controller Fault Handler Power-Up Handler			()		INVENTORY COMMAND	Data Structure used f			
Tasks	InventoryCommand			{}		Data Structure used t Data Structure for Inv			
MainTask	InventoryResponse		()	()	INVENTORY_RESPONSE_12B	Data Structure for Inv.			
MainProgram	FX9600:1		{}	{}	AB:ETHERNET_MODULE_SINT_464Bytes:1:0				
Munscheduled	FX9600:O		{}	{}	AB:ETHERNET_MODULE_SINT_8Bytes:O:0				
Motion Groups Ungrouped Axes	FX9600:C		()	{}	AB:ETHERNET_MODULE:C:0				
VO Configuration ■ 5069 Backplane ∰ [0] 5069-I306ER FX9600_Sample_EtherN ♣ A1, Ethernet ∰ 5069-I306ER FX9600_Sample_EtherNet_ ∬ ETHERNET-MODULE FX9600	,								
/O Configuration ■ 5069 Backplane ■ 001 5069-L306ER FX9600_Sample_EtherN ↓ A1, Ethernet ■ 5069-L306ER FX9600_Sample_EtherNet_ ■ ETHERNET-MODULE FX9600 ▲ A2, Ethernet	,								
A1, Ethernet 5069-L306ER FX9600_Sample_EtherNet_ ETHERNET-MODULE FX9600 A2, Ethernet	,								

To Create a Ladder Logic:

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

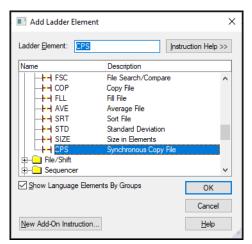
			5 M
Logix Designer - FX9600_Sample_EtherNet_IP (5069-L306ER 32.1			- 6 ×
EILE EDIT VIEW SEARCH LOGIC COMMUNICATIONS			
15 🖆 🗎 🖶 🗴 0 0 🗇 🥂	🗸 🏷 🎾 📴 🗽 🖒 Bo t		
RUN Path: <none></none>			
OK Path: <none> Path: <none></none></none>		(⊢ ⊢ ⊨ ++ ++ +) +() +()	· · · · · · · · · · · · · · · · · · ·
I III I/O Umine I No Forces	▶ _v No Edits 🔒	← Favorites Add-On Alarms Bit Timer/Counter Input/Output Compare Compute/Math Move/Logical	FileMisc. File/Shift Sequencer F
Controller Organizer 🚽 📮 🗸	× 🥏 Controller Tags - FX9600_Sample_EtherNe	P(controller) 🗏 MainProgram - MainRoutine* 🗙	
0 1	<u>ि</u> ् ् स ५ ९ ८ ८ ल छ		
A 🛁 Controller FX9600. Sample. EtherNet. IP			
Controller Tags	0 🛛		^
Controller Fault Handler			
Power-Up Handler			
🔺 🖳 Tasks	(End)		
🔺 🛟 MainTask			
 MainProgram Parameters and Local Tags 			
Parameters and Local lags B MainRoutine	1		
Unscheduled			
🖌 🚍 Motion Groups			
Ungrouped Axes			
🕨 💼 Alarm Manager			
Assets			
bs Logical Model			
✓			
Stop Backplane [0] 5069-L306ER FX9600_Sample_EtherNet_IP			
A S A1, Ethernet			
EB 5069-L306ER FX9600 Sample EtherNet IP			
ETHERNET-MODULE FX9600			
🔺 器 A2, Ethernet			
5069-L306ER FX9600_Sample_EtherNet_IP			
	-		
Type Ladder Diagram (Main)			
Description Program MainProgram			
Number of Rungs 1			
Number of Range 1			
<	3		v
Te Controller Organizer	<=		E State Sta
Errors			
O Errors A 0 Warnings	0 Messages		Search
U voinus a u wanings	0 messages		searon P
Ready			Communication Software: RSLinx Classic Rung 0 of 1 APP VER -
Reauy			Communication Software: RSLink Classic Rung 0 of 1 APP VER -

2. Double-click on Main Routine.

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

adder Element: XIC		Instruction Help	>>
Name	Description		_
H Rung H Branch H Branch Level ⊕ Aams	(SOR - EOR) (BST - BND) (NXB)		^
	Examine On		i
	Examine Off		
-+- OTE	Output Energize		
-+ - OTL	Output Latch		
	Output Unlatch		~
<u> S</u> how Language Eleme	nts By Groups	OK	-
		Cance	l.
New Add-On Instruction		Help	

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



- 9. Navigate to File/Shift and select (CPS) Synchronous Copy File.
- 10. Click OK to add the element.
- 11. Choose the Synchronous Copy File block and set:
 - Source: InventoryCommand
 - Dest: ModuleName:0
 - Length: 8.
- 12. Right-click the rung and select Add Rung to add a new rung.
- 13. Right-click on the rung and select Add Ladder Element.
- 14. Navigate to Bit operators and select (XIC) Examine On.
- 15. Click OK to add the element.
- 16. Double-click on the Examine On element.
- 17. Enter EnableOperation.

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

Length: 464.

Figure 33 Ladder Logic

💰 Logix Designer - FX9600_Sample_EtherNet_IP (5069-L30	30687.32.11)*	- ē ×
EILE EDIT VIEW SEARCH LOGIC COMMUNICA	CATIONS TOOLS MINDOW HELP	
🍾 🖕 🔛 🖶 🗶 🗇 🏠 🎐 💎	> かか 単 10 色 出 低 電 低 (2)	
RUN OK Path: <none></none>		
III Energy Storage		
I I/O Offline . No Forces		
Controller Organizer 👻 👎 🛪		<u> </u>
# 11		
Controller FX9600_Sample_EtherNet_IP	EnableOperation	CPS A
Controller Tags		Source FX96001
Controller Fault Handler		Dest InventoryResponse
Power-Up Handler		Length 464
Tesks		
A S Main lask	EnableOperation	CPS
Parameters and Local Tags		Source InventoryCommand Dest FX9600.0
MainRoutine		Length 8
Unscheduled		
Contract Motion Groups		
Ungrouped Axes	(£55)	
📁 Alarm Manager		
C Assets		
Add-On Instructions		
▲ G Data Types ▲ G User-Defined		
III INVENTORY COMMAND		
INVENTORY_RESPONSE_12B		
器 INVENTORY_RESPONSE_64B		
間 STATUS_MASK		
器 TAG_REPORT_12B		
間 TAG_REPORT_64B		
器 TIME_STAMP_UTC		
A 🖼 Strings		
器 TAG_EPC_12B 器 TAG EPC 648		
Add-On-Defined		
P In Predefined		
Module-Defined		
c >		
Type Ladder Diagram (Main)		
Description		
Program MainProgram	1	
Number of Rungs 0	1	
		• •
Controller Organizer Controller Organizer	Errors	+ # ×
Search Results 🗛 Watch		
		100 (End) of 2 100 1/00

Loading the Project to the PLC Controller

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

Configure Driver for EtherNet/IP and Find Controller using RSLinx

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

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

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

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





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

Figure 35 Configure Drivers

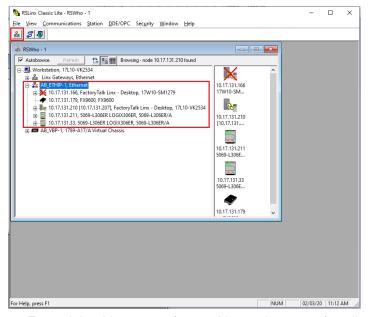
File View Communications Station DDE/OPC Security Window Help <td< th=""><th>? ×</th></td<>	? ×
Available Driver Types: EthenNet/IP Driver Configured Drivers Name and Description Add New RSLinx Classic Driver Add Ne	Dose Help Corfigure Statup Statu Stop Delete
For Help, press F1	NUM 02/03/20 10:51 AM

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

Figure 36 Driver Configuration

File View Communications Station DDE/OPC Security Window Help
Configure driver: AB_ETHIP-1 ? X
Avai Eff Browse Local Subnet Conf N Description Windows Default Broadcom NetXtreme Gigabit Eltitemet Bluetoth Device (Personal Area Network) Hyper-V Virtual Eltitemet Adapter #3 10.17.131.207 t t t t t t t t t t t t t t t t t t
OK Cancel Apply Help

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



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

- **13.** Make sure your PLC Controller appears in the list. If it does not appear debug the driver configuration and Ethernet network.
- 14. The PLC Controller communicat i on pathis nowset up and can be configured in the Studio 5000 Logix Designer project.

Set the Controller Path in Project

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

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

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

💰 Logix Designer - SampleApp_AOP (5069-L306ER 32.1)							
FILE EDIT VIEW SEARCH LOGIC COMMUNIC	CATIONS TOOLS WINDOW	V HELP					
12 🖕 🖬 🖶 🗶 🗊 🙃 🤊 🤆	~ *p *	5 / D h G B	•	00 8 00			
RIN							
Energy Storage	J.17.131.211*	% *			-(U)(L)- MSB COP CPS		
Unit of the Unit o	es 🕨 🔍 No Edits	a.	Favorites	Add-On Alarms B	t Timer/Counter Input/Out	tput Compare Compute	e/Math Move/Logical File/M
Controller Organizer	+ 7 × .	🗄 MainProgram - MainRo	utine 🗙 🛷 Controller 1	ags - SampleApp_AOP(controller)		
	Who Active (RSLinx C	(Jacolo)				- 0	×
🔺 🛁 Controller SampleApp_AOP	- gr who active (riseline c	idosic)					^
Controller Tags	Autobrowse Ref	fresh					
Controller Fault Handler	Workstation, 17L					Go Online	18(EN)
Power-Up Handler	🗄 🍰 Linx Gatewa						- Reny-
A G Motion Groups	😑 💑 AB_ETHIP-1,					Upload	
Ungrouped Axes		.166, FactoryTalk Linx - De	sktop, 17W10-SM1279			Download	
🕨 💼 Alarm Manager		.179, FX9600, FX9600	ryTalk Linx - Deskton 17L	10 10/2524		Update Eirmwar	JetReaderCaps
Assets		.211, 5069-L306ER LOGIX3		10-782334		opuate Drinwar	Enci 1000
b Logical Model ▲ ⊆ I/O Configuration		218, FA9000, FA9000				Glose	
A B 5069 Backplane		.33, 5069-L306ER LOGIX30	6ER, 5069-L306ER/A			Help	
0] 5069-L306ER SampleApp_AOP	⊕- AB_VBP-1, 17	789-A17/A Virtual Chassis					_
🔺 💑 A1/A2, Ethernet							
5069-L306ER SampleApp_AOP							1 State 1
FX9600 FX9600EFB2F6							
							eList.ERR
	Path: AB FTHIP-1						th 2
	Path: AB_ETHEP-1 Path in Project: <none></none>	1\10.17.131.211				Set Project Pa	th
	Paul in Projecti shorie >					Clear Project P	ath
							LE.GetProfileList
:							U)
i	b	4 =					`
							,
		Errors					→ # >
		CO Errors	A 0 Warnings	0 Mes	sages	Search	<u>م</u>
		omplete - 0 error(s)	, 0 warning(s)				
7							
:							
<	>						
Controller Organizer	<	:					,
Search Results AWatch							
				Communities	in fathers fitting flassin		100 1000

Figure 37 Who Active Dialogue

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

Figure 38 Project Path

🗳 Who Active (RSLinx Classic)	-		×
▲ Autobrowse Refresh	Upd	<u>Go Online</u> Upload Download ate <u>Eirmwa</u> <u>Close</u> Help	
Path: AB_ETHIP-1\10.17.131.211 Path in Project: AB_ETHIP-1\10.17.131.211	_	et Project F ar Project	

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

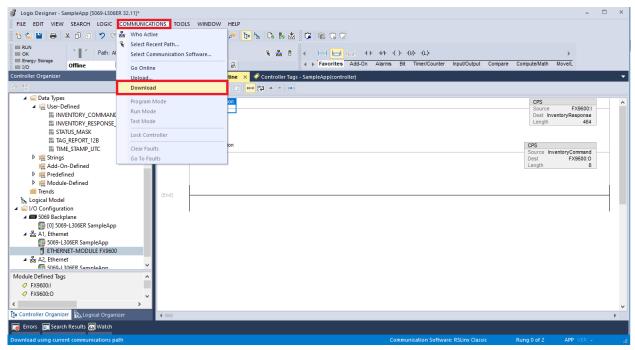
Load Project to PLC Controller

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

To load the project to the PLC Controller:

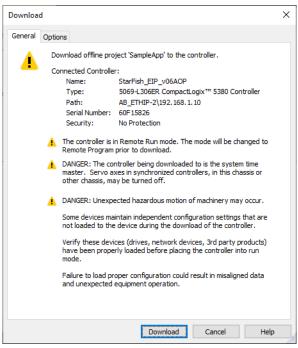
1. From the Studio 5000 Logix Designer project, select **Download** from the **COMMUNICATIONS** menu.

Figure 39 Communications > Dowload



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

Figure 40 Download Confirmation



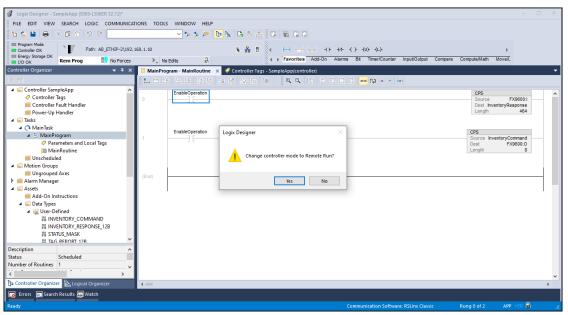
3. The project compiles and starts downloading to the PLC Controller.

Figure 41 Download Status

💰 Logix Designer - Sa											
FILE EDIT VIEW	SEARCH	LOGIC COMMUNIC	ATIONS TOOL								
1 🖕 🔛 🖶 🖓	¥ 🗇 🙃	2 6		- 🍫 🏂 🗾	h G 5 📩						
Program Mode Controller OK Energy Storage OK	`F	Path: AB_ETHIP-2\19				< H H H H + +			_) E	
III Energy Storage OK	Rem Prog	📃 No Forces		o Edits 🔒		♦ Favorites Add-On Ala	rms Bit Timer/Counter Input	t/Output Compare (Compute/Math	Move/L	
Controller Organizer		~ ₽ >	A B MainPro	gram - MainRoutine 🛛 🗙	Controller Tags	- StarFish_EIP_v06AOP(controller)					-
0 m					(db) +(e) 📮 🗱 bao						
191 INVE 191 STAT	NTORY_CO	SPONSE_12B	0	EnableOperation					CPS Source Dest Inve Length	FX9600:1 IntoryResponse 464	
	E_STAMP_U		1	EnableOperation	Downloading	olling for power up			CPS Source Inv Dest Length	entoryCommand FX9600:0 8	
Trends Tends	on		(End)			Cancel	<u> </u>				\dashv
€ [0] 5069 ▲	t 06ER Sampl	leApp									
Errors											▼ + ×
😒 0 Em	ors 🔺	0 Warnings	0 7	Messages				Sea	rch		Q
Verifying routine Verifying program Changing EtherNet, Polling for power	connecti IP Mode	ons		m'							<
Errors Search	Results 🚑	Watch									>
Ready							Communication Software: RSLi		Rung 0 of 2		

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

Figure 42 Set Remote Run



5. Studio 5000 Logix Designer displays green controller statuses.

Figure 43 Green Status Modes

Logix Designer - SampleApp [5069-L306ER 32.12]*		- 0	1 ×
EILE EDIT VIEW SEARCH LOGIC COMMUNICATION			<u> </u>
	- <u>1</u> 000 <u>m</u> mon <u>m</u> m		
Run Mode Controller OK Energy Storage OK		F	
I/O OK Rem Run I Vo Forces	▶ No Edits ↓ Favorites Add-On Alarms Bit Timer/Counter Input/Output Compare Counter Input/Output Counter Input/Output Compare Counter Input/Output Counter Input/Outp	Compute/Math Move/L	
Controller Organizer 🚽 🗸 🗸	MainProgram - MainRoutine 🗙 🛷 Controller Tags - SampleApp(controller)		•
a 🖷 🛛	L 当当 ビニ ビジ F 5 ビ 5 10 10 10 10 10 10 10 10 10 10 10 10 10		
🔺 🛁 Controller SampleApp	EnableOperation		
Controller Tags		CPS Source EX96001	^
Controller Fault Handler		Dest InventoryResponse	
Power-Up Handler		Length 464	
🔺 🛁 Tasks			
🔺 🛟 MainTask	EnableOperation	CPS	
🔺 🔓 MainProgram		Source InventoryCommand	
Parameters and Local Tags		Dest FX9600:O	
💼 MainRoutine		Length 8	
Unscheduled			
🔺 🚄 Motion Groups			
Ungrouped Axes	End)		_
👂 💼 Alarm Manager			
🔺 🛁 Assets			
Add-On Instructions			
🔺 🖳 Data Types			
A 🔛 User-Defined			
101 INVENTORY_COMMAND			
101 INVENTORY_RESPONSE_12B			
191 STATUS_MASK			
Description			
Status Scheduled			
Number of Routines 1			
····· · · · · · · · · · · · · · · · ·			
< >			~
	-		•
📷 Errors 🗊 Search Results 🐼 Watch			
Ready	Communication Software: RSLinx Classic R	Rung 0 of 2 APP VER 🔒	

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

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

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

Reading Reader Capabilities

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

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

Figure 44 Rung Instructions for Reading Capabilities

0 ENABLE.GetReaderCaps MSG Message Control Get_ReaderCaps (DN) (ER)		5 COP Source Get_ReaderCaps.ERR Dest ErrorCode Length 2
---	--	---

Table 1	Rung Instructio	ns
	Trang monuouo	110

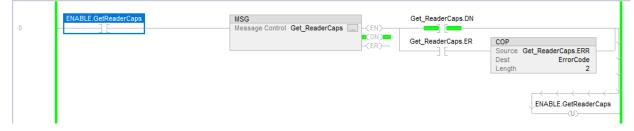
Number	Description
1	This is the Examine instruction to enable the Rung. It examines on the ENABLE.GetReaderCaps flag defined in the common ENABLE structure for all the flags. Once this flag is TRUE, the Rung becomes enabled and it executes the next instruction.
2	MSG instruction is to perform explicit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

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

Table 1 Rung Instructions (Continued)

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

Figure 45	ENABLE.GetReaderCaps flag = TRUE
-----------	----------------------------------



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

rigule 40 Reader Caps Data	Figure	46	ReaderCaps Data
----------------------------	--------	----	-----------------

Controller SampleApp AOP	Scope: SampleApp_AOF V Show: All Tags	√ T _v Reader	Caps		
Controller SampleApp_AOP Controller Tags	Name 28	🔺 Value 🔹 🔹 Force Mask	 Style 	Data 🛆 Properties	-
Controller Fault Handler	▶ Get_ReaderCaps	{}	{}	MESS 🔡 💱 🔎 🎼 🎼	Extend
Power-Up Handler	ReaderCaps	{}	{}	READ 🖌 General	
Tasks	ReaderCaps.ReaderID	'84:24:8D:EF:B2:F6'	{}		lerCaps
Ungrouped Axes	ReaderCaps.FirmwareVersion	'3.6.11.0'	{}		ture fo
Alarm Manager	ReaderCaps.ModelName	'96008'	{}	STRIN Usage <con Type Base</con 	ntroller>
Assets	ReaderCaps.CountryCode	356	Decimal	INT Alias For	
Be Logical Model	ReaderCaps.CommunicationStd	0	Decimal	INT Base Tag	
I/O Configuration	ReaderCaps.NumOfRFModes	33	Decimal	INT Data Type READ	ER_CA
[0] 5069-L306ER SampleApp_AOP	ReaderCaps.MinPower	1000	Decimal		pleApp
🔺 💑 A1/A2, Ethernet	ReaderCaps.MaxPower	3000	Decimal	INT External Read,	/Write
5069-L306ER SampleApp_AOP \$	▶ ReaderCaps.StepPower	10	Decimal	INT Constant No	
D FX9600 FX9600EFB2F6	ReaderCaps.NumOfAntennas	8	Decimal	SINT Required	
	ReaderCaps.NumOfGPI	4	Decimal	SINT Visible	
	ReaderCaps.NumOfGPO	4	Decimal	SINT Alarms 0	
	ReaderCaps.NationOFO	32	Decimal	⊿ Data	
	 ReaderCaps.waxivdmiPrenter 	52	Decimal	value	{.
				Force M Produced Conner	{.
				 Consumed Conne 	
				▲ Parameter Conne	
					Conne

Reader Capabilities Parameters

• Vendor Class: 64

• Data type size: 124 bytes.

Controller Organizer	▼ Ŧ × 🔀	Data Type: READER_CAPA	BILITIES ×				
Image: Second secon		Name: READER_C/	APABILITIES		Data Type Size: 124 bytes		
A Will User-Defined 器 ACCESS_AND_POST_FILTER 器 ACCESS_COMMAND 器 ACCESS_DATA 器 ACCESS RESPONSE	OST_FILTER [Description:		Structure for Reader Capabilities			
111 ANTENNA_CON	IFIGURATION	Name	Data Type	Description			
器 CONTROL_FLAG 器 EVENT REPORT		 ReaderID 	STRING32	Reader ID (ASCII String)			
181 GPIO_CONFIG	LONG	 FirmwareVersion 	STRING32	Firmware Version (ASCII String)			
間 GPI_CONFIG		 ModelName 	STRING32	Model Name (ASCII String)			
器 GPO_CONFIG 器 INVENTORY CO	-	-	GPO_CONFIG	CountryCode	INT	Code of the country	
111 INVENTORY_RES		CommunicationSt	d INT	Communication Standard			
111 PRE_FILTER_CLA				NumOfRFModes	INT	No of Supported RF Modes	
部 PRE_FILTER_CON 部 PROFILE_LIST	NFIG	MinPower	INT	Minimum Power supported			
111 PROFILE_LIST_C		MaxPower	INT	Maximum Power supported			
間 READER_CAPAB 調 STATUS MASK	ILITIES	StepPower	INT	Step size of power			
部 TAG_PATTERN		NumOfAntennas	SINT	No of Antenna Supported			
191 TAG REPORT 12		NumOfGPI	SINT	No of GPI Supported			
Description Structure for R Size 124 bytes	eader Capabilities	NumOfGPO	SINT	No of GPO Supported			
Jize 124 bytes		MaxNumPrefilter	SINT	No of Pre-Filters supported by the Reader			
		* Add Member					
۲	>			OK	ncel Apply Help		

Customize Configuration for RFID Operation

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

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



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

Reader Profile

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

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

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

Table 2 Profile List Explicit Message Data Model

Instance 0 has the following members:

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

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

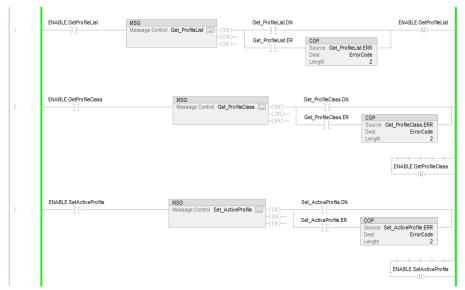


Figure 47 Rungs Performing an Operation

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

To specify the index of the profile name to read:

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

Figure 48 Specifying the Profile Index Number

Message Configuration	- Get_ProfileList		×
Configuration Communi	cation Tag		
Message Type:	CIP Generic	~	
Service Type: Get Attribut	e Single 🗸 🗸	Source Element	
Service e (He:	x) Class: 65 (Hex) Destination	0 (Bytes)
	Attribute: 1 (Hex) Element:	New Tag
⊖Enable ⊖Enable	Waiting 🔾 Start	Done	Done Length: 68
⊖ Error Code: Error Path: FX9600EFB2f Error Text:	Extended Error Code:		Timed Out 🕈
	ОК	Cancel	Apply Help

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

Figure 49 Controller Tags

ope: SampleApp_AOF V Show: All Tags		✓ T Profile		
Name	<u>=</u> ≣ ▲ Value	Force Mask	Style	Data 1
Get_ProfileClass	{	.) {	}	MESS
Get_ProfileList	{	.) {	}	MESS
ProfileClass	{	.} {	}	PROF
▲ ProfileList	{	.) {	}	PROF
ProfileList.ProfileName	'Maximum Data Rate	e' {]	}	STRIN
Set_ActiveProfile	{	.} {	}	MESS

- 5. To get the information about how many profiles are available in the reader and to know which is the currently active profile, the user should perform explicit message on the Profile List Class. Rung 2 was created to demonstrate this.
 - a. Click on the configuration for the MSG instruction in Rung 2 and change the attribute value to 3 (Num Instances).
 - **b.** Select the NumOfInstances field from the ProfileClass structure in the Destination Element field.

Figure 50 Obtaining the Number of Profiles

Message Configuration - Get_Prof	ïleClass			>
Configuration Communication Ta	9			
Message Type: CIP Gener	c	~		
Service Get Attribute Single	~	Source Element		\sim
		Source Length:	0 🌲	(Bytes)
Service e (Hex) Class:	65 (Hex)	Destination Element:	ProfileClass.	NumOfIn 🗸
Instance: 0 Attribute:	3 (Hex)	clement:	New Tag	
⊖ Enable ⊖ Enable Waiting	⊖ Start	Done	Done Length: 2	
	d Error Code:	S D010	Timed Out +	
	ОК	Cancel	Apply	Help

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

Figure 51 Number of Available Profiles

ppe: SampleApp_AOF V Show: All Tags			✓ T _→ Profile	
Name	== -	Value 🗧	Force Mask 🔷 🗧	Style
▶ Get_ProfileClass		{}	{}	
Get_ProfileList		{}	{}	
▲ ProfileClass		{}	{}	
ProfileClass.Revesion		0		Decimal
ProfileClass.MaxInstance		0		Decimal
ProfileClass.NumOfInstance		6		Decimal
ProfileClass.ActiveProfileInstance		0		Decimal
ProfileList		{}	{}	
Set ActiveProfile		{}	{}	

or

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

Figure 52	Active Profile Instance from Profile List Class
-----------	---

Message Co	onfiguratio	n - Get_Prof	fileClass				×
Configuratio	on Commun	nication Ta	g				
Message	<u>T</u> ype:	CIP Gener	ic) ~	•	
Service Type:	Get Attribu	te Single		\sim	Source Element	:	\sim
					Source Length:	0 🌲	(Bytes)
Ser <u>v</u> ice Code:	e (He	ex) <u>C</u> lass:	65	(Hex)	Destination Element:	ProfileClass.A	ctivePri 🗸
Instance:	0	Attri <u>b</u> ute:	64	(Hex)	Liement.	Ne <u>w</u> Tag	
		Wating	00-		* Data	Dans Legeth: 2	
 Error Coo Error Path: I Error Text: 	⊖ Enable le: FX9600EFB2	Extende	⊖ Sta ed Error (Oone	Done Length: 2	
			(ОК	Cancel	<u>A</u> pply	Help

- i. Click OK.
- **ii.** Perform the Get operation on the Profile List class in Rung 2 by toggling the ENABLE.GetProfileClass Examine On Instruction.
- iii. When complete, the Active Profile instance is available in the ActiveProfileInstance field of ProfileClass structure.

Figure 53 Active Profile Instance

ico <u>p</u> e: 😰 SampleApp_AOF 🗸 Sh <u>o</u> w: All Tags		~	Profile		
Name	== *	Value 🗧	Force Mask 🔷 🗧	Style	Data 1
▶ Get_ProfileClass		{}	{}		MESS
Get_ProfileList		{}	{}		MESS
▲ ProfileClass		{}	{}		PROF
ProfileClass.Revesion		0		Decimal	UINT
ProfileClass.MaxInstance		0		Decimal	UINT
ProfileClass.NumOfInstance		6		Decimal	UINT
ProfileClass.ActiveProfileInstance		2		Decimal	UINT
ProfileList		{}	{}		PROF
Set_ActiveProfile		{}	{}		MESS

Profile List Parameters:

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

Controller Organizer 👻 👎 🗙	BData Type: PROFILE_LIST ×	
0 1	Name: PROFILE_LIST	
🔺 🐖 User-Defined 📃 🔨		Data Type Size: 68 bytes
웲 ACCESS_AND_POST_FILTER	Description:	
111 ACCESS_COMMAND	Description:	
181 ACCESS_DATA		
볢 ACCESS_RESPONSE	Members:	
81 ANTENNA_CONFIGURATION	Name Data Type Description	
111 EVENT_REPORT		~
101 GPIO_CONFIG		
111 GPI_CONFIG	Î	
181 GPO_CONFIG	LEN - DINT	
111 INVENTORY_COMMAND	DATA - SINT[64]	
111 INVENTORY_RESPONSE_12B	* Add Member	
1911 INVENTORY_RESPONSE_64B		
器 PRE_FILTER_CONFIG		
191 PROFILE_LIST		
181 READER_CAPABILITIES		
181 SET_GET_CONFIG		
111 STATUS_MASK		
101 TAG_PATTERN		
間 TAG_REPORT_12B		
101 TAG REPORT 64R		
Description		
Size 68 bytes		
		×
< >	OK Cancel	Apply Help
T= Controller Organizer		

Changing the Active Profile in the Reader

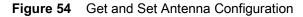
To change the Active Profile in the reader the user can perform Set Operation on the Active Profile Instance variable in the profile list explicit message. Rung 3 was created to demonstrate the functionality in sample application.

To change the active profile in reader:

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

Antenna Configuration

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



4	ENABLE.GetAntennaConf	COP MSG Source gAntennalD Dest Get_AntennaConf.Instance Length 4	itrol Get_AntennaConf(EN) ON (ER)
		Get_AntennaConf.ER Get_AntennaConf.ER Get_AntennaConf.ER Dest ErrorCode Length 2	ENABLE.GetAntennaConf
5	ENABLE.SetAntennaConf	COP MSG Source sAntennaD Dest AntennaConfig.AntennaD Length 1	ntrol Set_AntennaConf(EN) (DN) (ER)
		Set_AntennaConf.DN Set_AntennaConf.ER Source Set_AntennaConf.ER Dest ErrorCode Length 2	ENABLE.SetAntennaConf

To get the antenna configuration:

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

Figure 55 Specify Antenna ID

So	ope: Fig SampleApp_AOF > Show: All Tags			~ T_ Antenr	ia	
	Name	-=	Value	Force Mas	c +	Style
	▶ AntennaConfig		{	3	{}	
	▶ gAntennalD		~	1		Decimal
	Get_AntennaConf		{	.}	{}	
	sAntennalD			0		Decimal
	Set_AntennaConf		{	.}	{}	

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

Figure 56	Antenna Configuratio	n Values
-----------	----------------------	----------

sope: SampleApp_AOF ~ Show: All Tags		~	Antenna	
Name	== *	Value 🗧	Force Mask 🔷 🗧	Style
▲ AntennaConfig		{}	{}	
AntennaConfig.AntennalD		1		Decima
AntennaConfig.Sel		0		Decima
AntennaConfig.Session		0		Decima
AntennaConfig.Target		0		Decima
AntennaConfig.RFModeIndex		9		Decima
AntennaConfig.Pad		0		Decima
AntennaConfig.Tari		0		Decima
AntennaConfig.TagPopulation		300		Decima
AntennaConfig.PowerLevel		3000		Decima
▶ gAntennalD		1		Decima
▶ Get_AntennaConf		{}	{}	
▶ sAntennalD		0		Decima
Set_AntennaConf		{}	{}	

To modify the antenna configuration in the reader:

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

Figure 57 Antenna ID to Modify

MainProgram - MainRo X 🗸 Controller Tags - SampleApp_AOP(controller) ×		Antenna	
oge: SampleApp_AOF Show: All Tags Name				Style
AntennaConfig		{}	{}	Style
▶ gAntennalD		1		Decimal
▶ Get_AntennaConf		{}	{}	
▶ sAntennalD		0		Decimal
Set_AntennaConf		{}	{}	

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



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

Antenna configuration parameters:

Vendor class: 66

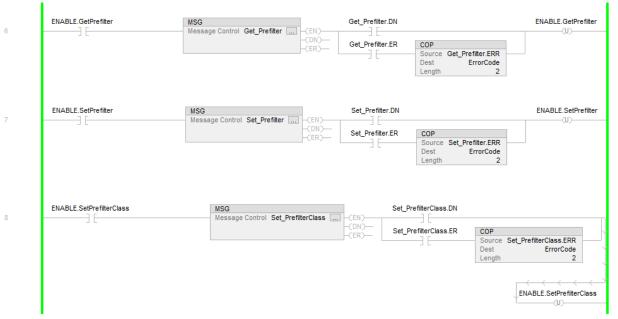
• Data type size: 12 bytes

Controller Organizer 🗾 👻 🕂 🗙	Data Type: ANTEN	INA_CONFIGURATION	×	
a =	Name: AN	ENNA_CONFIGURAT	ON	Data Type Size: 12 bytes
 ☐ Ungrouped Axes ▲ Aiarm Manager ▲ Assets ☐ Add-On Instructions ▲ Data Types 	Description:			
🔺 🐖 User-Defined	Name	Data Type	Description	
IN ACCESS_AND_POST_FILTER	AntennalD	SINT	0-8, 0=All Antenna Default : 0	^
81 ACCESS_COMMAND 81 ACCESS DATA	Sel	SINT	Asserted=0, Deasserted=1, SL ALL=2 Default : 2	
111 ACCESS_RESPONSE	Session	SINT	Session0=0, Session1=1, Session2=2, Session3=3 Default : 1	
해 ANTENNA_CONFIGURATION 해 CONTROL FLAGS	Target	SINT	State A=0, State B=1, AB FLIP=2 Default : 0	
81 EVENT_REPORT_LONG	RFModeln	dex SINT	0 - 39 (Depends on Region) Default: 0	
器 GPIO_CONFIG	Pad	SINT	Ignore pad	
部 GPI_CONFIG 部 GPO_CONFIG	Tari	INT	Depends on RF Mode Default: 0	
81 INVENTORY_COMMAND	TagPopula	tion INT	0 - 32767 Default: 100	
制 INVENTORY_RESPONSE_12B	PowerLeve	I INT	1000 - 3000 (dBm) Default: 3000	
部 PRE_FILTER_CLASS 部 PRE_FILTER_CONFIG	₩ Add Me	ember		
101 PROFILE LIST				
Description Size 12 bytes				
12 Dytes				
				~
× >			OK Cancel	Apply Help

Configuring Pre-Filter

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

Figure 58 Pre-filter Explicit Message



To get the pre-filter from a specific index:

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

Figure 59 Specify the Filter Index

Message Co	onfiguration	n - Get_Pref	ilter		×
Configuratio	on Commun	ication Ta	J		
Message	Type:	CIP Gener	c	~	·
Service Type:	Get Attribut	te Single	~	Source Element	· · · ·
				Source Length:	0 🔶 (Bytes)
Service Code:	e (He	x) Class:	67 (Hex)	Destination	PreFilterConfig 🗸
Instance:	1	Attribute:	1 (Hex)	Element:	New Tag
⊖ Enable	⊖ Enable	Wating	⊖Start	O Done	Done Length: 0
-		_	-	UDone	-
O Error Coo Error Path: I Error Text:	le: FX9600EFB2		d Error Code:		Timed Out 🕈
			ОК	Cancel	Apply Help

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

Figure 60 PreFilterConfig Tag

ope: SampleApp_AOF v Show: All Tags	 <u> </u>	∕ T _→ Prefil	
Name	Value 🗧	Force Mask 🔷 🗧	Style
Get_Prefilter	{}	{}	
PrefilterClassConfig	{}	{}	
PreFilterConfig	{}	{}	
PreFilterConfig.FilterID	0		Decima
PreFilterConfig.AntennalD	0		Decima
PreFilterConfig.MemoryBank	0		Decima
PreFilterConfig.Target	0		Decima
PreFilterConfig.Action	0		Decima
PreFilterConfig.Pad	0		Decima
PreFilterConfig.BitOffset	0		Decima
PreFilterConfig.BitCount	0		Decima
PreFilterConfig.Pad2	0		Decima
 PreFilterConfig.TagPattern 		{}	
Set_Prefilter	{}	{}	
Set_PrefilterClass	{}	{}	

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

- 1. Modify the pre-filter values in the PreFilterConfig tag.
- 2. Select the MainProgram MainRoutine tab.

L

3. Modify the MSG instruction in Rung 7 to specify the pre-filter index as an Instance on which the specified pre-filter needs to be added/modified.

Figure 61 Modify the MSG Instruction

Aessage Co	nfiguration	- Set Prefi	lter			
Configuration		ication Tag				
Message T	ype:	CIP Generi	c	~	1	
Service [Type: Service [Code: Instance:		e Single ×) Class: Attribute:	67 (Hex) 1 (Hex)	Source Element: Source Length: Destination Element:	S2 Vew Tag	g ~ (Bytes) ~
) Enable	O Enable	Waiting	⊖ Start	O Done	Done Length: 0	
) Error Code Error Path: F. Error Text:	e: X9600EFB2I		d Error Code:		🗌 Timed Out 🕈	
			OK	Cancel	Apply	Help

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

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

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

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

Figure 62 Delete a Pre-filter

ope: SampleApp_AOF ~ Show: All Tags	~	Prefil	
Name	Value 🗧	Force Mask 💦 🗧 🗧	Style
Get_Prefilter	{}	{}	
PrefilterClassConfig	{}	{}	
PrefilterClassConfig.Revision	0		Decimal
PrefilterClassConfig.MaxInstance	0		Decima
PrefilterClassConfig.NumOfInstance	0		Decimal
PrefilterClassConfig.DeleteInstance	<u>∽</u> 1		Decimal
PreFilterConfig	{}	{}	
Set_Prefilter	{}	{}	
Set_PrefilterClass	{}	{}	

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



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

Pre-Filter Configuration Parameters:

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

0 1			ame:	PRF FILTE	R_CONFIG		Tura Cirar 52 histor
	間 ACCESS_DATA	^				Dat	ta Type Size: 52 bytes
	器 ACCESS_RESPONSE 器 ANTENNA CONFIGURATION	De	scriptio	n:			
	11 CONTROL_FLAGS						
	锦 EVENT_REPORT_LONG	M	embers:				
	101 GPIO_CONFIG		a Na	ame	Data Type	Description	
	181 GPI_CONFIG		Fi	lterID	SINT	1 to 32	
	器 GPO_CONFIG 器 INVENTORY COMMAND		A	ntennalD	SINT	0 - 8, 0=All Antenna	
	111 INVENTORY_RESPONSE_12B		м	emoryBank	SINT	EPC=1, TID=2, USER=3	
	81 PRE_FILTER_CLASS		Ta	rget	SINT	SL=0, S0=1, S1=2, S2=3, S3=4	
	191 PRE_FILTER_CONFIG			ction	SINT	0 - 7	
	器 PROFILE_LIST 器 PROFILE LIST CLASS						
	111 READER_CAPABILITIES		Pa	d	SINT	Ignore Pad	
	111 STATUS_MASK		Bi	tOffset	INT	Bit offset from where Tag Pattern begins	
	111 TAG_PATTERN		Bi	tCount	INT	Tag Pattern Length in number of bits	
	111 TAG_REPORT_12B		Pa	ad2	INT	Ignore Pad	
	間 TAG_REPORT_64B 間 TIME_STAMP_UTC		▶ Ta	gPattern	STRING36	Tag Pattern	
	111 TRIGGER CONFIGURATION	~		Add Member			
Description							
Size	52 bytes						
<						OK Cancel App	ly Help

Configuring Access/Post-Filter

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

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

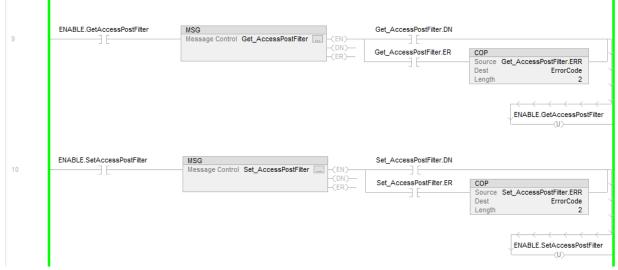


Figure 63 AccessPostFilter Operation

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

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

Figure 64	AccessPostFilter Tag
-----------	----------------------

MainProgram - MainRoutine 🤣 Controller Tags - SampleApp_AOP(cont	roller) ×			
cope: SampleApp_AOF ~ Show: All Tags		~	Post	
Name	== *	Value 🗧	Force Mask 🔹 🗧	Style
▲ AccessPostFilter		{}	{}	
AccessPostFilter.TagPatternA		{}	{}	
AccessPostFilter.TagPatternB		{}	{}	
AccessPostFilter.MatchPattern		0		Decimal
AccessPostFilter.PeakRSSILowerLimit		0		Decimal
AccessPostFilter.PeakRSSIUpperLimit		0		Decimal
AccessPostFilter.PeakRSSIMatchRange		0		Decimal
▶ Get_AccessPostFilter		{}	{}	
Set_AccessPostFilter		{}	{}	

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

Access/Post-Filter Configuration Parameters:

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

Controller Organizer 👻 👎 🗙	Data Type: ACCESS_AND_POST_FILTER	TER ×	
	Name: ACCESS_AND_POST_FIL Description:	FILTER Data Type Si	ize: 180 bytes
R ACCESS_AND_POST_FILTER R ACCESS_COMMAND 路 ACCESS DATA	Members:		
배 ACCESS_BRSPONSE 띎 ANTENNA_CONFIGURATION 빎 CONTROL_FLAGS 삚 EVENT_REPORT_LONG 빎 GPI_CONFIG 빎 GPI_CONFIG 빎 GPO_CONFIG 빎 GPO_CONFIG	✓ TagPatternA TAG_PA	PATTERN Pattern A Params	
INVENTORY, COMMAND INVENTORY, RESPONSE_12B INPERFILTER, CLASS IN PREFILTER, CONFIG IN PROFILE_LIST IN PROFILE_LIST_CLASS IN READER, CAPABILITES IN STATIF, MAKY	TagPattern - STRING36 Tag Pattern TagMask - STRING36 Tag Mask TagMaskBitCount - INT Tag Mask Length in noum MemoryBank - SNIT Reserved=0, EPC=1, TID=	imber of bits	
Description	Pad - SINT Ignore Pad		
Size 180 bytes	TagPatternB TAG_PA	PATTERN Pattern B Params	
	MatchPattern SINT	A_AND_B=0, NOTA_AND_B=1, NOTA_AND_NOTB=2, A_AND_NOTB=3	
	PeakRSSILowerLimit SINT	RSSI Filter Lower Limit	
	PeakRSSIUpperLimit SINT	RSSI Filter Higher Limit	
	PeakRSSIMatchRange SINT	WITHIN_RANGE = 0, OUTSIDE_RANGE = 1, GREATER_THAN_LOWER_LIMIT = 2, LOWER_THAN_UPPER_	LIMIT = 3
	* Add Member		
			v.
		OK Cancel Apply	Help

Trigger Configuration

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

Figure 65 Configure Trigger Values

11	ENABLE.GetTriggerConf	MSG Message Control Get_TriggerConf(EN)(DN)(ER)(ER)	Get_TriggerConf.DN Get_TriggerConf.ER	COP Source Get_TriggerConf Dest ErrorCode Length 2	ENABLE.GetTriggerConf
12	ENABLE.SetTriggerConf	MSG Message Control Set_TriggerConf(EN)(EN)(ER)-	Set_TriggerConf.DN	COP Source Set_TriggerConf.ERR Dest ErrorCode Length 2	ENABLE.SetTriggerConf

To get the value for configured triggers:

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



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

Figure 66 Trigger Values

e: 🖉 SampleApp_AOF 🗸 Show: All Tags			✓ Trigger
ame	== 🔺 Alias For	Base Tag	Data Type
Get_TriggerConf			MESSAGE
Set_TriggerConf			MESSAGE
TriggerConfig			TRIGGER_CONFIGURATION
TriggerConfig.StartTriggerType			SINT
TriggerConfig.StartGPIPort			SINT
TriggerConfig.StartEventType			SINT
TriggerConfig.Pad			SINT
TriggerConfig.PeriodicTime			DINT
 TriggerConfig.StartDelay 			DINT
TriggerConfig.StopTriggerType			SINT
 TriggerConfig.StopGPIPort 			SINT
TriggerConfig.StopEventType			SINT
TriggerConfig.EnableTagEvent			SINT
TriggerConfig.TagEventTimeout			INT
TriggerConfig.Count			INT
TriggerConfig.Timer			DINT

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

Trigger Configuration Parameters:

• Vendor Class: 69

• Data type size: 28 bytes

Controller Organizer 🚽 🕈 🗙	Data Type: TRIGGER_CON	FIGURATION ×	
이 미 해 ANTENNA_CONFIGURATION	Name: TRIGGER_C	ONFIGURATION	Data Type Size: 28 bytes
器 CONTROL_FLAGS 器 EVENT_REPORT_LONG 器 GPIO CONFIG	Description:		
器 GPI_CONFIG	Members:		
111 GPO_CONFIG	Name	Data Type	Description
器 INVENTORY_COMMAND 器 INVENTORY RESPONSE 12B	StartTriggerType	SINT	0=Immediate, 1=Periodic, 2=GPI
81 PRE_FILTER_CLASS	StartGPIPort	SINT	GPI pin number
88 PRE_FILTER_CONFIG	StartEventType	SINT	0=High to low, 1:low to high (GPI pin state transition)
왦 PROFILE_LIST 뫪 PROFILE LIST CLASS	Pad	SINT	Ignore Pad
181 READER_CAPABILITIES	PeriodicTime	DINT	Used when start trigger is "periodic" (millisecond)
IN STATUS_MASK	StartDelay	DINT	Used when start trigger is "periodic" (millisecond)
器 TAG_PATTERN 器 TAG_REPORT_12B	StopTriggerType	SINT	0=Immediate, 1=Duration, 2=GPI with timeout, 3=Tag observation, 4=N attempts
811 TAG_REPORT_64B	StopGPIPort	SINT	GPI pin number
81 TIME_STAMP_UTC 81 TRIGGER_CONFIGURATION	StopEventType	SINT	0=High to low, 1:low to high (GPI pin state transition)
Strings	EnableTagEvent	SINT	Enables Tag Events
📾 Add-On-Defined 🛛 👻	TagEventTimeout	INT	Tag Event Timeout (millisecond)
Description Size 28 bytes	Count	INT	Used when stop trigger is "Tag Observations" or "N attempts"
	Timer	DINT	When stop trigger type is "Duration" or as "Timeout" (millisecond)
	PeriodicReportDur	ation DINT	-1=disable Periodic reporting, 0=report at the end of inventory, n=report once every n second Default : -1
	* Add Member		· · · · · · · · · · · · · · · · · · ·
< >			OK Cancel Apply Help
· ·			

GPIO Configuration

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

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

Figure 67 GPIO Configuration

13	ENABLE.GetGPIOConfig	MSG Message Control Get_GPIOConfig(EN) -(CN) -(ER)-	COP Source Get_GPIOConfig.ERR Dest ErrorCode Length 2	ENABLE:GetGPIOConfig
14	ENABLE.SetGPI0Config	MSG - Message Control Set_GPIOConfig	COP Source Set_GPIOConfig.ERR Dest ErrorCode Length 2	ENABLE.SetGPIOConfig

To get the current status of GPIOs:

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

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

Figure 68 GPIOConfig Tags

rope: SampleApp_AOF Show: All Tags V GPIO						
Name	-8 +	Value	+	Force Mask 🛛 🗧	Style	Data Type
Get_GPIOConfig			{}	{}		MESSAGE
▲ GPIOConfig			{}	{}		GPIO_CONFIG
GPIOConfig.NumOfGPI			4		Decimal	SINT
 GPIOConfig.NumOfGPO 			4		Decimal	SINT
 GPIOConfig.Pad 			0		Decimal	INT
GPIOConfig.GPIConfig			{}	{}		GPI_CONFIG[8]
GPIOConfig.GPIConfig[0]			{}	{}		GPI_CONFIG
GPIOConfig.GPIConfig[0].GPIP	ort		1		Decimal	SINT
 GPIOConfig.GPIConfig[0].Enabl 	e		1		Decimal	SINT
GPIOConfig.GPIConfig[0].State			0		Decimal	SINT
GPIOConfig.GPIConfig[1]			{}	{}		GPI_CONFIG
GPIOConfig.GPIConfig[1].GPIP	ort		2		Decimal	SINT
GPIOConfig.GPIConfig[1].Enabl	e		1		Decimal	SINT
GPIOConfig.GPIConfig[1].State			0		Decimal	SINT
GPIOConfig.GPIConfig[2]			{}	{}		GPI_CONFIG
GPIOConfig.GPIConfig[3]			{}	{}		GPI_CONFIG
GPIOConfig.GPIConfig[4]			{}	{}		GPI_CONFIG
GPIOConfig.GPIConfig[5]			{}	{}		GPI_CONFIG
GPIOConfig.GPIConfig[6]			{}	{}		GPI_CONFIG
GPIOConfig.GPIConfig[7]			{}	{}		GPI_CONFIG
GPIOConfig.GPOConfig			{}	{}		GPO_CONFIG[8]
GPIOConfig.GPOConfig[0]			{}	{}		GPO_CONFIG
GPIOConfig.GPOConfig[0].GPC	Port		1		Decimal	SINT
GPIOConfig.GPOConfig[0].State	e		0		Decimal	SINT
GPIOConfig.GPOConfig[1]]	{}	{}		GPO_CONFIG
GPIOConfig.GPOConfig[1].GPC	Port		2		Decimal	SINT
GPIOConfig.GPOConfig[1].State	e		0		Decimal	SINT
GPIOConfig.GPOConfig[2]			{}	{}		GPO_CONFIG
GPIOConfig.GPOConfig[3]			{}	{}		GPO_CONFIG
GPIOConfig.GPOConfig[4]			{}	{}		GPO_CONFIG

2. To enable the GPI or set the GPO status:

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



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

GPIO Configuration Parameters:

• Vendor class: 6A

• Data type size: 68 bytes

Controller Organizer 🚽 🕂 🗸	B Dat	ta Type: G	PIO_CONF	IG ×			
	Nan	ne:	GPIO_CONFIG			Data Type Size: 68 bytes	
📕 Unscheduled 🥂 🖌						<u></u>	
Ungrouped Axes	Dese	cription:					
Alarm Manager							
A Ssets	Mer	mbers:					
Add-On Instructions		Name	2	Data Type	Description		
🔺 <u> Data Types</u>		Num	OfGPI	SINT	Number of GPI can be	read/write	1
🔺 🐖 User-Defined							
間 ACCESS_AND_POST_FILTER		Num	OfGPO	SINT	Number of GPO can b	e read/write	
器 ACCESS_COMMAND		Pad		INT	Ignore Pad		
181 ACCESS_DATA		+ GPIC	onfia	GPI_CONFIG[8]	GPI Configuration		
웲 ACCESS_RESPONSE		÷					
111 ANTENNA_CONFIGURATION			PiPort - Sil				-
器 CONTROL_FLAGS 器 EVENT_REPORT_LONG			GPI Pin Nu				
111 EVENT_REPORT_LONG			able - S/N				
111 GPI_CONFIG			R: Shows st				
111 GPO_CONFIG				GPI events			
111 INVENTORY_COMMAND			ate - SINT				
部 INVENTORY_RESPONSE_12B			R: Shows st W: Ignore	tate			
101 PRE FILTER CLASS			-				_
scription		➡ GPOC	Config	GPO_CONFIG[8]	GPO Configuration		
e 68 bytes		<u>^</u>					_
			POPort - S				
			GPO Pin N				
			ate - SINT	urrent state			
			W: New sta				
		× 4	dd Membe	r			-
							-1
							_
					OK Cancel	Apply Hel	р

Reading Event Report

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

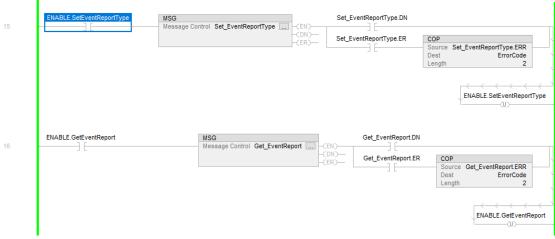


Figure 69 Generate Events

To get the event report:

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

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

Figure 70 Set Event Type

ope: 🔯 SampleApp_AOF 🗸	Show: All Tags				✓ Event
Name 😑	Value	+	Force Mask 🔷 🗧 🕈	Style	Data Type
▲ EventReport		{}	{}		EVENT_REPORT_LONG
EventReport.EventTyp	e	3		Decimal	SINT
EventReport.GPIPortN	lo	0		Decimal	SINT
EventReport.GPOPort	No	0		Decimal	SINT
EventReport.Antenna	D	0		Decimal	SINT
EventReport.Antenna	St	0		Decimal	SINT
EventReport.Tempera	tu	0		Decimal	SINT
EventReport.Tempera	tu	0		Decimal	SINT
EventReport.Tempera	tu	0		Decimal	SINT
EventReport.Exceptio	nl		{}		STRING260
Get_EventReport		{}	{}		MESSAGE
Set_EventReportType		{}	{}		MESSAGE

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

Event Report Configuration Parameters:

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

Controller Organizer 🗸 🗸 🤸	<	🔀 Data	Type: E	VENT_REPORT	LONG ×		
		Nam	e:	EVENT_REPO	RT_LONG		Data Type Size: 272 bytes
🔺 <u> </u> Data Types	۱I (but a type bizer zitz bytes
🔺 🐖 User-Defined		Desc	iption:				
器 ACCESS_AND_POST_FILTER							
191 ACCESS_COMMAND							
間 ACCESS_DATA		Mem	bers:				
間 ACCESS_RESPONSE			Name	2	Data Type	Description	
191 ANTENNA_CONFIGURATION	11		Event	Туре	SINT	To select app	propriate event
器 CONTROL_FLAGS 器 EVENT REPORT LONG			GPIPo	ortNo	SINT	Event report	ed on GPI PIN Number
181 GPIO CONFIG			GPOP	ortNo	SINT	GPI state	
調 GPI CONFIG						5	
191 GPO CONFIG			Anter	nnalD	SINT	Event Report	ted on Antenna ID
101 INVENTORY_COMMAND			Anter	nnaStatus	SINT	Antenna Sta	tus
間 INVENTORY_RESPONSE_12B	н.		Temp	eratureSource	SINT	Source of Te	mperature Event
器 PRE_FILTER_CLASS 器 PRE_FILTER_CONFIG			Temp	eratureAlarm	SINT	Temperature	e Alarm Level
111 PROFILE_LIST			Temp	eratureValue	SINT	Temperature	Value
部 PROFILE_LIST_CLASS			Except	tionInfo	STRING260	Exception In	formation
111 READER_CAPABILITIES				dd Member			
器 STATUS_MASK			* *	a renoem			
101 TAG PATTERN							¥.
< >>					OK	Cancel	Apply Help
Recentrality Operations Review 10							

Performing Inventory

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

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

Figure 71 Inventory Operation

17	Performinventory	CPS Source OutputInventoryCommand Dest FX9600EF82F6:O1 Length 8
18	PerformInventory	CPS Source FX9600EFB2F6:I1 Dest InputInventoryResponse Length 468

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

Figure 72 Green Enabled Status

17	Performinventory	CPS Source OutputInventoryCommand Dest FX9600EFB2F6:O1 Length 8
18	Performinventory	CPS Source FX9600EFB2F6:11 Dest InputhventoryResponse Length 468

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

Start Inventory Operation

To start the Inventory operation with the reader:

1. Modify the OutputInventoryCommand tag structure.

Figure 73 Modifying OutputInventoryCommand

ope: 🖉 SampleApp_AOF 🗸 Show: All Tags			~		ory	
Name	== •	Value	+	Force Mas	c 🔶	Style
InputInventoryResponse			{}		{}	
 OutputInventoryCommand 			{}		{}	
 OutputInventoryCommand.CommandType 			1			Decima
 OutputInventoryCommand.Handshake 		\sim	0			Decima
OutputInventoryCommand.EnablePrefilter			0			Decima
OutputInventoryCommand.EnablePostfilter			0			Decima
OutputInventoryCommand.AntennaMask			0			Decima

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

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

Table 3	OutputInventoryCommand Field Descriptions
---------	---

Read Inventory Data

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

Figure 74 Reading Inventory Data

pe: SampleApp_AOF V Show: All Tags	~	T_Inventory	
Name === A	Value 🔶	Force Mask 🔹 🔶	Style
▲ InputInventoryResponse	{}	{}	
InputInventoryResponse.ConnectionFault	0		Decima
InputInventoryResponse.StatusMask	{}	{}	
InputInventoryResponse.PacketSequenceNo	91	_	Decima
InputInventoryResponse.NumOfReports	7		Decima
InputInventoryResponse.Pad	0		Decima
▲ InputInventoryResponse.TagReports	{}	{}	
✓ InputInventoryResponse.TagReports[0]	{}	{}	
InputInventoryResponse.TagReports[0].TagEPC	'\$ADr\$12\$05\$14\$AE)\$	{}	
InputInventoryResponse.TagReports[0].TagPC	12288		Decima
InputInventoryResponse.TagReports[0].TagCRC	31736		Decima
InputInventoryResponse.TagReports[0].AntennalD	4		Decima
InputInventoryResponse.TagReports[0].RSSI	-30		Decima
InputInventoryResponse.TagReports[0].ChannelIndex	1		Decima
InputInventoryResponse.TagReports[0].SeenCount	1		Decima
InputInventoryResponse.TagReports[0].PhaseInfo	28107		Decima
InputInventoryResponse.TagReports[0].FirstSeenTime	{}	{}	
InputInventoryResponse.TagReports[0].LastSeenTime	{}	{}	
InputInventoryResponse.TagReports[0].AccessStatus	255		Decima
InputInventoryResponse.TagReports[0].Pad	0		Decima
InputInventoryResponse.TagReports[1]	{}	{}	
InputInventoryResponse.TagReports[2]	{}	{}	
InputInventoryResponse.TagReports[3]	{}	{}	
InputInventoryResponse.TagReports[4]	{}	{}	
InputInventoryResponse.TagReports[5]	{}	{}	
InputInventoryResponse.TagReports[6]	{}	{}	
 OutputInventoryCommand 	{}	{}	
OutputInventoryCommand.CommandType	1		Decima
OutputInventoryCommand.Handshake	1		Decima
OutputInventoryCommand.EnablePrefilter	0		Decimal

Stop Inventory Operation

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

ope: SampleApp_AOF V Show: All Tags		~	T_Inventory	
Name	□ ≣ ▲ Value	+	Force Mask 🔹 🗧	Style
InputInventoryResponse		{}	{}	
InputInventoryResponse.ConnectionFault		0		Decim
InputInventoryResponse.StatusMask		{}	{}	
InputInventoryResponse.PacketSequenceNo		42		Decima
InputInventoryResponse.NumOfReports		0		Decim
InputInventoryResponse.Pad		0		Decim
InputInventoryResponse.TagReports		{}	{}	
 OutputInventoryCommand 		{}	{}	
OutputInventoryCommand.CommandType		2		Decim
OutputInventoryCommand.Handshake		2		Decima
OutputInventoryCommand.EnablePrefilter	\sim	0		Decima
OutputInventoryCommand.EnablePostfilter		0		Decim
OutputInventoryCommand.AntennaMask		0		Decima
PerformInventory		1		Decim

Figure 75 Stop Inventory Operation

Performing Access Operation

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

Figure 76 Access Operation/Read Access Data

19		DPS Source OutputAccessCommand Dest FX9600EFB2F6:02 ength 332
20	PerformAccess	CPS Source FX9600EFB2F6:12 Dest InputAccessResponse Length 384

To perform the Access operation with the reader:

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

Figure 77 Access Operation

19	s p	PS ource OutputAccessCommand est FX9600EFB2F6:02 ength 332
20	PerformAccess	CPS Source FX9600EFB2F6:2 Dest InputAccessResponse Length 384
(End)		

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



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

Access Read Operation

To perform the Access Read operation:

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



NOTE: AccessData is required when performing AccessWrite.

Figure 78 Access Read Operation

Image: Comparison of the second se	
▶ InputAccessResponse {} {} {} ▲ OutputAccessCommand {} {} {} ▶ OutputAccessCommand.CommandType 1 Deci ▶ OutputAccessCommand.Handshake 2 Deci ▶ OutputAccessCommand.MemoryBank 1 Deci	
▲ OutputAccessCommand {} {} {} ▶ OutputAccessCommand.CommandType 1 Deci ▶ OutputAccessCommand.Handshake ✓ 2 Deci ▶ OutputAccessCommand.MemoryBank 1 Deci	le
OutputAccessCommand.CommandType 1 Deci OutputAccessCommand.Handshake 2 Deci OutputAccessCommand.MemoryBank 1 Deci	
▶ OutputAccessCommand.Handshake ✓ 2 Deci ▶ OutputAccessCommand.MemoryBank 1 Deci	
OutputAccessCommand.MemoryBank Deci	imal
	imal
OutputAccessCommand.EnableAccessFilter 0 Deci	imal
	imal
▶ OutputAccessCommand.AntennaMask 0 Deci	imal
▶ OutputAccessCommand.TagEPC '00@5\$A8\$80\$C8\$00\$ {}	
▶ OutputAccessCommand.Password 0 Deci	imal
▶ OutputAccessCommand.ByteOffset 4 Deci	imal
▶ OutputAccessCommand.ByteCount 48 Deci	imal
▶ OutputAccessCommand.AccessData {}	

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

Access Write Operation

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

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

Figure 79 Access Write Operation

ope: SampleApp_AOF Show: All Tags		~	<u>_</u>	T_ tAccess	
Name	== *	Value 🔶	Fo	rce Mask 👘 🗧 🗧	Style
InputAccessResponse		{}		{}	
 OutputAccessCommand 		{}		{}	
OutputAccessCommand.CommandType		2			Decimal
OutputAccessCommand.Handshake		 ✓ 2 			Decimal
OutputAccessCommand.MemoryBank		1			Decimal
OutputAccessCommand.EnableAccessFilter		0			Decimal
OutputAccessCommand.AntennaMask		0			Decimal
OutputAccessCommand.TagEPC		'00@5\$A8\$80\$C8\$00\$		{}	
OutputAccessCommand.Password		0			Decimal
OutputAccessCommand.ByteOffset		4			Decimal
OutputAccessCommand.ByteCount		48			Decimal
 OutputAccessCommand.AccessData 		{}		{}	
OutputAccessCommand.AccessData.PageIndex		0			Decimal
OutputAccessCommand.AccessData.PageData		'\$A8\$80\$C8\$00\$00\$12		{}	o cent

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

Access Response Data

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

Figure 80 Access Response Data

ope: 😰 SampleApp_AOF 🗸 Show: All Tags		V T_tAccess			
Name	== 🔺 Value	+	Force Mask 🛛 🗧	Style	
 InputAccessResponse 		{}	{}		
InputAccessResponse.ConnectionFaulted		0		Decima	
InputAccessResponse.StatusMask		{}	{}		
InputAccessResponse.PacketSequenceNumber		121		Decima	
InputAccessResponse.ResponseHeader		0		Decima	
InputAccessResponse.Pad		0		Decima	
 InputAccessResponse.TagReport 		{}	{}		
InputAccessResponse.TagReport.TagEPC			{}		
InputAccessResponse.TagReport.TagPC		0		Decima	
InputAccessResponse.TagReport.TagCRC		0		Decima	
InputAccessResponse.TagReport.AntennalD		0		Decima	
InputAccessResponse.TagReport.RSSI		0		Decima	
InputAccessResponse.TagReport.ChannelIndex		0		Decima	
InputAccessResponse.TagReport.SeenCount		0		Decima	
InputAccessResponse.TagReport.PhaseInfo		0		Decima	
InputAccessResponse.TagReport.FirstSeenTime		{}	{}		
InputAccessResponse.TagReport.LastSeenTime		{}	{}		
InputAccessResponse.TagReport.AccessStatus		0]	Decima	
InputAccessResponse.TagReport.Pad		0		Decima	
 InputAccessResponse.AccessData 		{}	{}		
InputAccessResponse.AccessData.PageIndex		0		Decima	
InputAccessResponse.AccessData.PageData			{}		

Error Codes and Troubleshooting

Error Codes

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

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

EtherNet/IP Stack Error

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

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

Table 4 EtherNet/IP Stack Errors

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

Table 4 EtherNet/IP Stack Errors (Continued)

RFID Operation Specific Error

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

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

Table 3 INTID Operation Specific Lifes	Table 5	RFID Operation Specific Errors
--	---------	---------------------------------------

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

Table 5 RFID Operation Specific Errors (Continued)

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

Table 5	RFID Operation	Specific Errors	(Continued)
---------	----------------	-----------------	-------------

Troubleshooting

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

Unable to Load Sample Application

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

Setting Proper Requested Packet Interval (RPI)

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

Figure 81 shows the RPI configuration window.



Module Properties: Local (FX960	0 2.001) ×
Module Properties: Local (FX9600 General Gonnection Hodule Info Internet Protocol Vendor	D2.2001) × Connection Requested Packet Interval (RPI): 20.0 Inhibit Module Major Fault On Controller If Connection Fails While in Run Mode Vuse Unicast Connection over EtherNet/IP
Status: Offline	OK Cancel Apply Help

I/O Not Responding Status

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

Application EtherNet/IP Adapter Application Not Running

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

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

Reader IP Not Configured/Reachable

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

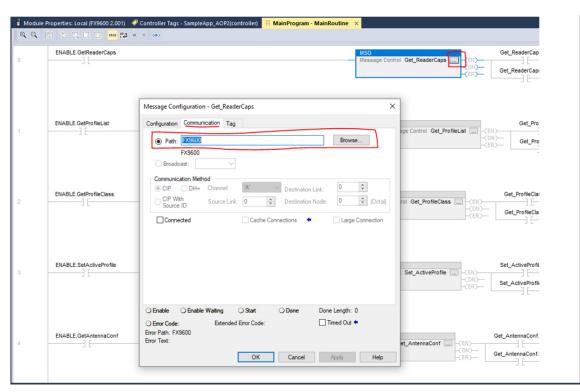
It is suggested that the reader and PLC remain in same subnet or assign the static IP address to both in the same subnet.

Communication Path Not Set Properly In MSG Instructions

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

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





Unable to Perform Inventory or Access Operation

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

Response Data format

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

Figure 83 Style Window

InputAccessResponse	{}	{}		ACCESS_RESPONSE
 InputInventoryResponse 	{}	{}		INVENTORY_RESPONSE_12B
InputInventoryResponse.ConnectionFault	0		Decimal	BOOL
InputInventoryResponse.StatusMask	{}	{}		STATUS_MASK
InputInventoryResponse.PacketSequenceNo	0		Decimal	SINT
InputInventoryResponse.NumOfReports	0		Decimal	SINT
InputInventoryResponse.Pad	0		Decimal	INT
 InputInventoryResponse.TagReports 	{}	{}		TAG_REPORT_12B[7]
 InputInventoryResponse.TagReports[0] 	{}	{}		TAG_REPORT_12B
InputInventoryResponse.TagReports[0].TagEPC		{}		TAG_EPC_12B
InputInventoryResponse.TagReports[0].TagEPC.LEN	0	_	Decimal	DINT
InputInventoryResponse.TagReports[0].TagEPC.DATA	{}	{}	ASCII 🗸	SI NT[12]
InputInventoryResponse.TagReports[0].TagPC	0		Binary	T
InputInventoryResponse.TagReports[0].TagCRC	0		Octal Decimal	T AI
InputInventoryResponse.TagReports[0].AntennalD	0		Hex	SLNT
InputInventoryResponse.TagReports[0].RSSI	0		ASCII	SI NT
InputInventoryResponse.TagReports[0].ChannelIndex	0		Decimal	INT
InputInventoryResponse.TagReports[0].SeenCount	0		Decimal	INT
InputInventoryResponse.TagReports[0].PhaseInfo	0		Decimal	INT
InputInventoryResponse.TagReports[0].FirstSeenTime	{}	{}		TIME_STAMP_UTC
InputInventoryResponse.TagReports[0].LastSeenTime	{}	{}		TIME_STAMP_UTC



www.zebra.com