

Industrial PC Platform

NY-series

IPC Machine Controller

Industrial Panel PC / Industrial Box PC

**Built-in EtherNet/IP™ Port
User's Manual**

NY532-1500

NY532-1400

NY532-1300

NY532-5400

NY512-1500

NY512-1400

NY512-1300

Industrial Panel PC
Industrial Box PC



NOTE

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.

No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

Trademarks

- Sysmac and SYSMAC are trademarks or registered trademarks of OMRON Corporation in Japan and other countries for OMRON factory automation products.
- Microsoft, Windows, Excel, and Visual Basic are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.
- EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- ODVA, CIP, CompoNet, DeviceNet, and EtherNet/IP are trademarks of ODVA.
- The SD and SDHC logos are trademarks of SD-3C, LLC.  
- Intel and Intel Core are trademarks of Intel Corporation in the U.S. and / or other countries.

Other company names and product names in this document are the trademarks or registered trademarks of their respective companies.

Copyrights

Microsoft product screen shots reprinted with permission from Microsoft Corporation.

Introduction

Thank you for purchasing an NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC.

This manual provides a collective term of Industrial Panel PC and Industrial Box PC which are applicable products as the NY-series Industrial PC. This manual also provides the range of devices that are directly controlled by the Controller functions embedded the Real-Time OS in the NY-series Industrial PC as the Controller.

This manual contains information that is necessary to use the NY-series Controller. Please read this manual and make sure you understand the functionality and performance of the NY-series Controller before you attempt to use it in a control system.

Keep this manual in a safe place where it will be available for reference during operation.

Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems (an electrical engineer or the equivalent).

- Personnel in charge of introducing FA systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of installing and maintaining FA systems.
- Personnel in charge of managing FA systems and facilities.

For programming, this manual is intended for personnel who understand the programming language specifications in international standard IEC 61131-3 or Japanese standard JIS B 3503.

Applicable Products

This manual covers the following products.

- NY-series IPC Machine Controller Industrial Panel PC
 - NY532-15□□
 - NY532-14□□
 - NY532-13□□
 - NY532-5400
- NY-series IPC Machine Controller Industrial Box PC
 - NY512-15□□
 - NY512-14□□
 - NY512-13□□

Part of the specifications and restrictions for the Industrial PC are given in other manuals. Refer to *Relevant Manuals* on page 2 and *Related Manuals* on page 21.

Relevant Manuals

The following table provides the relevant manuals for the NY-series Controller.

Read all of the manuals that are relevant to your system configuration and application before you use the NY-series Controller.

Most operations are performed from the Sysmac Studio Automation Software. Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for information on the Sysmac Studio.

Purpose of use	Manual									
	Basic information					NY-series Industrial Panel PC / Industrial Box PC Motion Control User's Manual	NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherCAT Port User's Manual	NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP Port User's Manual	NY-series NC Integrated Controller User's Manual	NY-series Troubleshooting Manual
	NY-series IPC Machine Controller Industrial Panel PC Hardware User's Manual	NY-series IPC Machine Controller Industrial Box PC Hardware User's Manual	NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Setup User's Manual	NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Software User's Manual	NY-series Instructions Reference Manual					
Introduction to NY-series Panel PCs	○									
Introduction to NY-series Box PCs		○								
Setting devices and hardware										
Using motion control	○	○				○				
Using EtherCAT							○			
Using EtherNet/IP								○		
Making setup ^{*1}										
Making initial settings			○							
Preparing to use Controllers										
Software settings										
Using motion control						○				
Using EtherCAT					○		○			
Using EtherNet/IP								○		
Using numerical control									○	
Writing the user program										
Using motion control							○	○		
Using EtherCAT					○	○		○		
Using EtherNet/IP									○	
Using numerical control										○
Programming error processing										○
Testing operation and debugging										
Using motion control						○				
Using EtherCAT					○		○			
Using EtherNet/IP								○		
Using numerical control									○	
Learning about error management and corrections ^{*2}										○
Maintenance										
Using motion control	○	○				○				
Using EtherCAT							○			
Using EtherNet/IP								○		

*1 Refer to the *NY-series Industrial Panel PC / Industrial Box PC Setup User's Manual* (Cat. No. W568) for how to set up and how to use the utilities on Windows.

*2 Refer to the *NY-series Troubleshooting Manual* (Cat. No. W564) for the error management concepts and the error items.

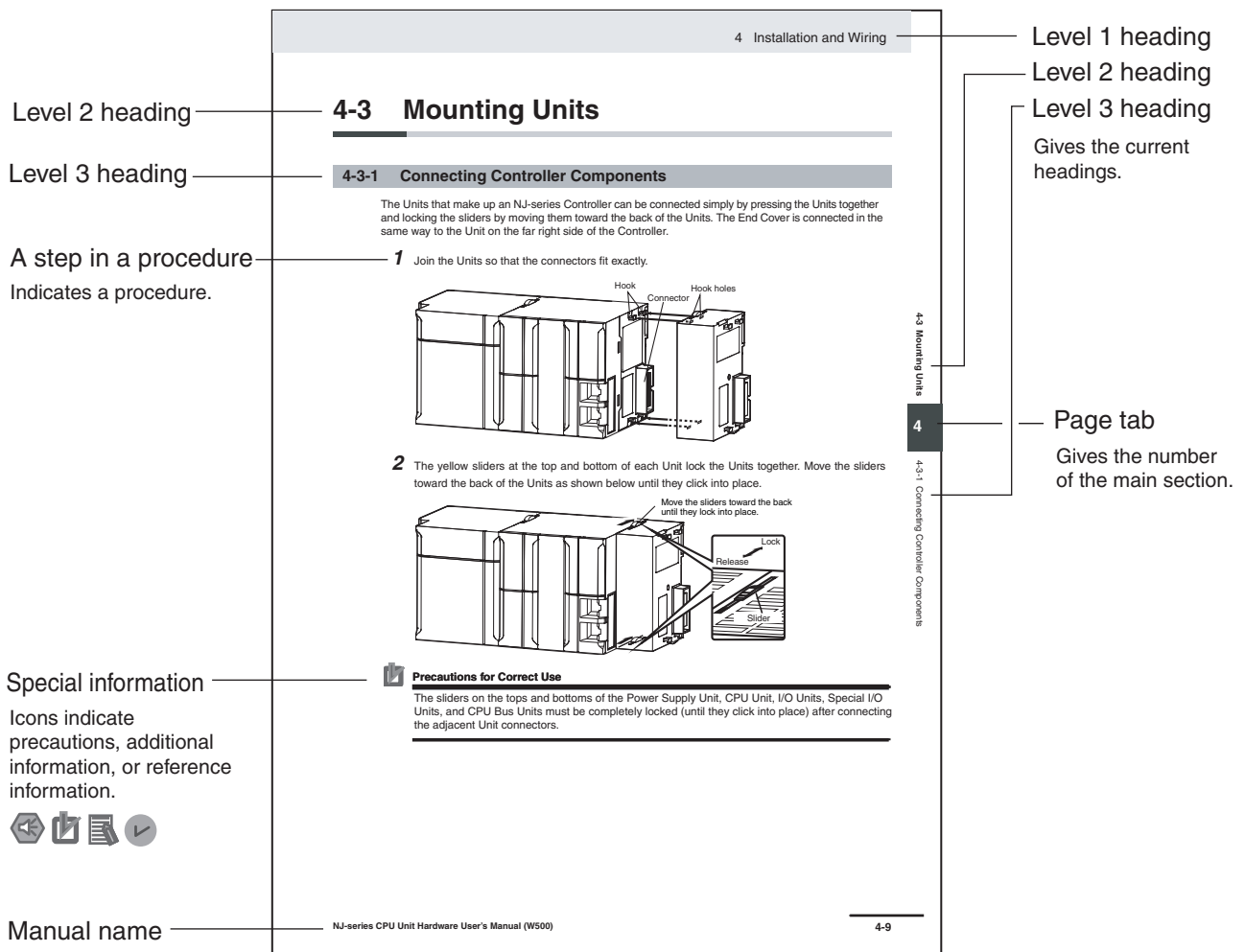
Manual Structure

Some of the descriptions of functions in this manual are common to NJ/NX-series. Therefore, note the following conditions.

- The same function names are used for the common functions of the NJ/NX/NY-series. If the term “CPU Unit” is included in the function names, such as the CPU Unit names, CPU Unit write protection and other functions, it indicates the “Controller” in the NY-series.
- The “CPU Unit” that is described in a list of function specifications in this manual also indicates the “Controller” in the NY-series.

Page Structure

The following page structure is used in this manual.



This illustration is provided only as a sample. It may not literally appear in this manual.

Special Information

Special information in this manual is classified as follows:



Precautions for Safe Use

Precautions on what to do and what not to do to ensure safe usage of the product.



Precautions for Correct Use

Precautions on what to do and what not to do to ensure proper operation and performance.



Additional Information

Additional information to read as required.

This information is provided to increase understanding or make operation easier.



Version Information

Information on differences in specifications and functionality for Controller with different unit versions and for different versions of the Sysmac Studio is given.

Note References are provided to more detailed or related information.

Precaution on Terminology

In this manual, “download” refers to transferring data from the Sysmac Studio to the physical Controller and “upload” refers to transferring data from the physical Controller to the Sysmac Studio.

For the Sysmac Studio, synchronization is used to both upload and download data. Here, “synchronize” means to automatically compare the data for the Sysmac Studio on the computer with the data in the physical Controller and transfer the data in the direction that is specified by the user.

Sections in this Manual

1	Introduction	10	FTP Server	1	10
2	Installing Ethernet Networks	11	FTP Client	2	11
3	System-defined Variables Related to the Built-in EtherNet/IP Port	12	SNMP Agent	3	12
4	Determining IP Addresses	13	Communications Performance and Communications Load	4	13
5	Sysmac Studio Settings for the Built-in EtherNet/IP Port	14	Checking Communications Status of Network and Troubleshooting	5	14
6	Testing Communications	A	Appendices	6	A
7	Tag Data Link Functions	I	Index	7	I
8	CIP Message Communications			8	
9	Socket Service			9	

CONTENTS

Introduction	1
Relevant Manuals	2
Manual Structure	3
Sections in this Manual	5
Terms and Conditions Agreement	12
Safety Precautions	14
Precautions for Safe Use	15
Precautions for Correct Use	16
Regulations and Standards	17
Versions	18
Related Manuals	21
Revision History	23

Section 1 Introduction

1-1 Introduction	1-2
1-1-1 EtherNet/IP Features	1-2
1-1-2 Features of Built-in EtherNet/IP Port on NY-series Industrial PCs	1-3
1-2 System Configuration and Configuration Devices	1-5
1-2-1 Devices Required to Construct a Network	1-5
1-2-2 Support Software Required to Construct a Network	1-6
1-3 Built-in EtherNet/IP Port	1-7
1-3-1 Specifications	1-7
1-3-2 Part Names and Functions	1-10
1-4 Introduction to Communications Services	1-12
1-4-1 IP Communications with Windows	1-12
1-4-2 CIP (Common Industrial Protocol) Communications Services	1-12
1-4-3 IP Routing	1-14
1-4-4 BOOTP Client	1-17
1-4-5 FTP Server	1-17
1-4-6 FTP Client	1-18
1-4-7 Socket Service	1-19
1-4-8 Specifying Host Names	1-19
1-4-9 SNMP Agent	1-20
1-5 EtherNet/IP Communications Procedures	1-21

Section 2 Installing Ethernet Networks

2-1 Selecting the Network Devices	2-2
2-1-1 Recommended Network Devices	2-2
2-1-2 Ethernet Switch Types	2-3

2-1-3	Ethernet Switch Functions	2-3
2-1-4	Precautions for Ethernet Switch Selection.....	2-4
2-2	Network Installation	2-6
2-2-1	Basic Installation Precautions.....	2-6
2-2-2	Recommended Network Devices	2-6
2-2-3	Precautions When Laying Twisted-pair Cable	2-6
2-2-4	Precautions When Installing and Connecting Ethernet Switches	2-9
2-3	Connecting to the Network.....	2-10
2-3-1	Ethernet Connectors.....	2-10
2-3-2	Connecting the Cable	2-11

Section 3 System-defined Variables Related to the Built-in EtherNet/IP Port

3-1	System-defined Variables Related to the Built-in EtherNet/IP Port	3-2
3-2	System-defined Variables	3-3
3-3	Specifications for Individual System-defined Variables	3-13

Section 4 Determining IP Addresses

4-1	IP Addresses	4-2
4-1-1	IP Address Configuration	4-2
4-1-2	Allocating IP Addresses	4-3
4-1-3	Subnet Masks	4-3
4-1-4	CIDR.....	4-4
4-2	Controller IP Address Settings.....	4-5
4-2-1	Determining IP Addresses	4-5
4-2-2	Setting IP Addresses	4-6
4-2-3	Online Connection	4-8
4-2-4	Checking the Current IP Address	4-10
4-3	Private and Global Addresses.....	4-11
4-3-1	Private and Global Addresses	4-11
4-3-2	Using a Private Address for the Built-in EtherNet/IP Port.....	4-12
4-3-3	Using a Global Address for the Built-in EtherNet/IP Port.....	4-13

Section 5 Sysmac Studio Settings for the Built-in EtherNet/IP Port

5-1	TCP/IP Settings Display	5-2
5-2	Link Settings Display	5-6
5-3	FTP Display	5-7
5-4	SNMP Settings Display	5-8
5-5	SNMP Trap Settings Display.....	5-10
5-6	CIP Settings Display.....	5-12

Section 6 Testing Communications

6-1	Testing Communications	6-2
6-1-1	PING Command	6-2
6-1-2	Using the PING Command	6-2
6-1-3	Host Computer Operation.....	6-3

Section 7 Tag Data Link Functions

7-1	Introduction to Tag Data Links	7-2
7-1-1	Tag Data Links	7-2
7-1-2	Data Link Data Areas	7-3
7-1-3	Tag Data Link Functions and Specifications	7-6
7-1-4	Overview of Operation	7-7
7-1-5	Starting and Stopping Tag Data Links	7-9
7-1-6	Controller Status	7-9
7-1-7	Concurrency of Tag Data Link Data	7-12
7-2	Setting Tag Data Links	7-16
7-2-1	Starting the Network Configurator	7-16
7-2-2	Tag Data Link Setting Procedure	7-18
7-2-3	Registering Devices	7-19
7-2-4	Creating Tags and Tag Sets	7-21
7-2-5	Connection Settings	7-32
7-2-6	Creating Connections Using the Wizard	7-42
7-2-7	Creating Connections by Device Dragging and Dropping	7-45
7-2-8	Connecting the Network Configurator to the Network	7-48
7-2-9	Downloading Tag Data Link Parameters	7-53
7-2-10	Uploading Tag Data Link Parameters	7-56
7-2-11	Verifying the Tag Data Links	7-59
7-2-12	Starting and Stopping Tag Data Links	7-63
7-2-13	Clearing the Device Parameters	7-64
7-2-14	Saving the Network Configuration File	7-66
7-2-15	Reading a Network Configuration File	7-68
7-2-16	Checking Connections	7-69
7-2-17	Changing Devices	7-71
7-2-18	Displaying Device Status	7-72
7-3	Ladder Programming for Tag Data Links	7-73
7-3-1	Ladder Programming for Tag Data Links	7-73
7-3-2	Status Flags Related to Tag Data Links	7-77
7-4	Tag Data Links with Other Models	7-78

Section 8 CIP Message Communications

8-1	Overview of the CIP Message Communications Service	8-3
8-1-1	Overview of the CIP Message Communications Service	8-3
8-1-2	Message Communications Service Specifications	8-3
8-2	CIP Messaging Communications Client Function	8-4
8-2-1	Overview	8-4
8-2-2	CIP Communications Instructions	8-5
8-2-3	Using CIP Communications Instructions	8-6
8-2-4	Route Path	8-7
8-2-5	Request Path (IOI)	8-10
8-2-6	Service Data and Response Data	8-14
8-2-7	Sample Programming for CIP Connectionless (UCMM) Message Communications	8-15
8-2-8	Sample Programming for CIP Connection (Class 3) Message Communications	8-20
8-2-9	Operation Timing	8-27
8-2-10	Response Codes	8-28
8-3	CIP Communication Server Functions	8-32
8-3-1	CIP Message Structure for Accessing CIP Objects	8-33
8-3-2	CIP Message Structure for Accessing Variables	8-33
8-4	Specifying Request Path	8-34
8-4-1	Examples of CIP Object Specifications	8-34
8-4-2	Examples of Variable Specifications	8-34
8-4-3	Logical Segment	8-35
8-4-4	Data Segment	8-36
8-4-5	Specifying Variable Names in Request Paths	8-37

8-5	CIP Object Services	8-40
8-5-1	CIP Objects Sent to the Built-in EtherNet/IP Port.....	8-40
8-5-2	Identity Object (Class ID: 01 Hex).....	8-40
8-5-3	TCP/IP Interface Object (Class ID: F5 hex).....	8-44
8-5-4	Ethernet Link Object (Class ID: F6 Hex).....	8-47
8-5-5	Controller Object (Class ID: C4 Hex).....	8-52
8-6	Read and Write Services for Variables	8-53
8-6-1	Read Services for Variables.....	8-53
8-6-2	Write Service for Variables.....	8-55
8-7	Variable Data Types	8-57
8-7-1	Data Type Codes.....	8-57
8-7-2	Common Format.....	8-58
8-7-3	Elementary Data Types.....	8-58
8-7-4	Derived Data Types.....	8-60

Section 9 Socket Service

9-1	Basic Knowledge on Socket Communications	9-2
9-1-1	Sockets.....	9-2
9-1-2	Port Numbers for Socket Services.....	9-2
9-2	Basic Knowledge on Protocols	9-3
9-2-1	Differences between TCP and UDP.....	9-3
9-2-2	Fragmenting of Send Data.....	9-5
9-2-3	Data Reception Processing.....	9-7
9-2-4	Broadcasting.....	9-10
9-3	Overview of Built-in EtherNet/IP Port Socket Services	9-11
9-3-1	Overview.....	9-11
9-3-2	Procedure.....	9-11
9-4	Settings Required for the Socket Services	9-12
9-5	Socket Service Instructions	9-13
9-6	Details on Using the Socket Services	9-14
9-6-1	Using the Socket Services.....	9-14
9-6-2	Procedure to Use Socket Services.....	9-15
9-6-3	Timing Chart for Output Variables Used in Communications.....	9-17
9-6-4	UDP Sample Programming.....	9-19
9-6-5	TCP Sample Programming.....	9-24
9-7	Precautions in Using Socket Services	9-30
9-7-1	Precautions for UDP and TCP Socket Services.....	9-30
9-7-2	Precautions for UDP Socket Services.....	9-30
9-7-3	Precautions for TCP Socket Services.....	9-30

Section 10 FTP Server

10-1	Overview and Specifications	10-2
10-1-1	Overview.....	10-2
10-1-2	Specifications.....	10-3
10-2	FTP Server Function Details	10-4
10-2-1	Supported Files.....	10-4
10-2-2	Connecting to the FTP Server.....	10-4
10-3	Using the FTP Server Function	10-7
10-3-1	Procedure.....	10-7
10-3-2	List of Settings Required for the FTP Server Function.....	10-7
10-4	FTP Server Application Example	10-8
10-5	Using FTP Commands	10-10
10-5-1	Table of Commands.....	10-10

10-5-2	Using the Commands.....	10-11
10-6	Using SD Memory Card Operations.....	10-16
10-6-1	File Types.....	10-16
10-6-2	Format of Variable Data.....	10-17
10-7	Application Example from a Host Computer.....	10-18

Section 11 FTP Client

11-1	Using the FTP Client to Transfer Files.....	11-2
11-1-1	Transferring Files.....	11-2
11-1-2	Connectable FTP Servers.....	11-2
11-1-3	File Transfer Options.....	11-3
11-1-4	Other Functions.....	11-4
11-2	FTP Client Communications Instructions.....	11-5
11-2-1	Functions of the FTP Client Communications Instructions.....	11-5
11-2-2	Restrictions on the FTP Client Communications Instructions.....	11-7
11-3	FTP Client Application Example.....	11-9

Section 12 SNMP Agent

12-1	SNMP Agent.....	12-2
12-1-1	Overview.....	12-2
12-1-2	Specifications.....	12-3
12-1-3	SNMP Messages.....	12-3
12-1-4	MIB Specifications.....	12-4
12-2	Procedure to Use the SNMP Agent.....	12-21
12-2-1	Procedures.....	12-21
12-2-2	Settings Required for the SNMP Agent.....	12-21

Section 13 Communications Performance and Communications Load

13-1	Communications System.....	13-2
13-1-1	Tag Data Link Communications Method.....	13-2
13-1-2	Calculating the Number of Connections.....	13-4
13-1-3	Packet Interval (RPI) Accuracy.....	13-5
13-2	Adjusting the Communications Load.....	13-6
13-2-1	Checking Bandwidth Usage for Tag Data Links.....	13-7
13-2-2	Tag Data Link Bandwidth Usage and RPI.....	13-8
13-2-3	Adjusting Device Bandwidth Usage.....	13-9
13-2-4	Changing the RPI.....	13-10
13-2-5	RPI Setting Examples.....	13-16
13-3	I/O Response Time in Tag Data Links.....	13-21
13-3-1	Timing of Data Transmissions.....	13-21
13-3-2	Built-in EtherNet/IP Port Data Processing Time.....	13-22
13-3-3	Relationship between Task Periods and Packet Intervals (RPIs).....	13-24
13-3-4	Maximum Tag Data Link I/O Response Time.....	13-25
13-4	Message Service Transmission Delay.....	13-27

Section 14 Checking Communications Status of Network and Troubleshooting

14-1	Device Monitoring on the Network Configurator.....	14-2
-------------	---	-------------

14-1-1	Starting the Device Monitoring.....	14-2
14-1-2	Status Displays of the Device Monitoring	14-2
14-2	Connection Status Codes and Troubleshooting	14-9

Appendices

A-1	Functional Comparison of the EtherNet/IP Port with Other Series.....	A-2
A-2	Use the Sysmac Studio to Set the Tag Data Links (EtherNet/IP Connections).....	A-3
A-2-1	Overview of the Tag Data Links (EtherNet/IP Connections) Settings with the Sysmac Studio... A-3	
A-2-2	Procedure to Make the EtherNet/IP Connection Settings with the Sysmac Studio	A-4
A-2-3	EtherNet/IP Connection Settings	A-5
A-2-4	Making the EtherNet/IP Connection Settings with the Sysmac Studio	A-10
A-2-5	Checking Communications Status with the Sysmac Studio and Troubleshooting.....	A-33
A-2-6	Troubleshooting	A-37
A-3	EDS File Management	A-43
A-3-1	Installing EDS Files.....	A-44
A-3-2	Creating EDS Files	A-44
A-3-3	Deleting EDS Files.....	A-45
A-3-4	Saving EDS Files.....	A-45
A-3-5	Searching EDS Files.....	A-46
A-3-6	Displaying EDS File Properties.....	A-47
A-3-7	Creating EDS Index Files	A-47
A-4	Precautions for Using the Network Configurator on Windows XP, Windows Vista, or Windows 7 or Higher.....	A-48
A-4-1	Changing Windows Firewall Settings	A-48
A-5	Variable Memory Allocation Methods	A-51
A-5-1	Variable Memory Allocation Rules	A-51
A-5-2	Important Case Examples	A-59
A-6	Precautions When Accessing External Outputs in Controllers	A-63
A-7	TCP State Transitions	A-64

Index

Terms and Conditions Agreement

Warranty, Limitations of Liability

Warranties

● Exclusive Warranty

Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

● Limitations

OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE.

Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.

● Buyer Remedy

Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron's analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

See <http://www.omron.com/global/> or contact your Omron representative for published information.

Limitation on Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Application Considerations

Suitability of Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY OR IN LARGE QUANTITIES WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.

Disclaimers

Performance Data

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron's test conditions, and the user must correlate it to actual application requirements. Actual performance is subject to the Omron's Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron's representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Safety Precautions

Refer to the following manuals for safety precautions.

- NY-series Industrial Panel PC Hardware User's Manual (Cat. No. W557)
- NY-series Industrial Box PC Hardware User's Manual (Cat. No. W556)
- NY-series Industrial Panel PC / Industrial Box PC Software User's Manual (Cat. No. W558)

Precautions for Safe Use

Refer to the following manuals for precautions for safe use.

- NY-series Industrial Panel PC Hardware User's Manual (Cat. No. W557)
- NY-series Industrial Box PC Hardware User's Manual (Cat. No. W556)
- NY-series Industrial Panel PC / Industrial Box PC Software User's Manual (Cat. No. W558)

Precautions for Correct Use

Refer to the following manuals for precautions for correct use.

- NY-series Industrial Panel PC Hardware User's Manual (Cat. No. W557)
- NY-series Industrial Box PC Hardware User's Manual (Cat. No. W556)
- NY-series Industrial Panel PC / Industrial Box PC Software User's Manual (Cat. No. W558)

Regulations and Standards

Refer to the following manuals for regulations and standards.

- NY-series Industrial Panel PC Hardware User's Manual (Cat. No. W557)
- NY-series Industrial Box PC Hardware User's Manual (Cat. No. W556)

Versions

Hardware revisions and unit versions are used to manage the hardware and software in NY-series Controllers and EtherCAT slaves. The hardware revision or unit version is updated each time there is a change in hardware or software specifications. Even when two Units or EtherCAT slaves have the same model number, they will have functional or performance differences if they have different hardware revisions or unit versions.

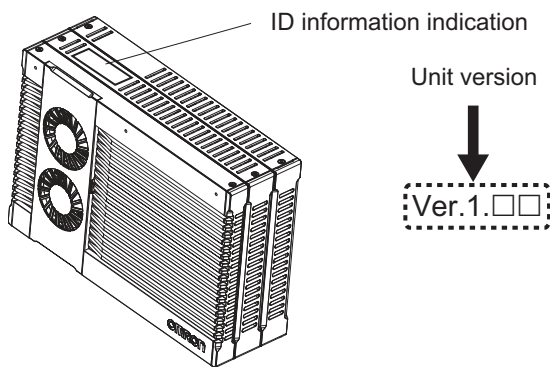
Checking Versions

You can check versions on the ID information indications or with the Sysmac Studio.

Checking Unit Versions on ID Information Indications

The unit version is given on the ID information indication on the back side of the product.

The ID information on an NY-series NY5□2-□□□□ Controller is shown below.



Checking Unit Versions with the Sysmac Studio

You can use the Sysmac Studio to check unit versions. The procedure is different for Units and for EtherCAT slaves.

● Checking the Unit Version of an NY-series Controller

You can use the Production Information while the Sysmac Studio is online to check the unit version of a Unit. You can only do this for the Controller.

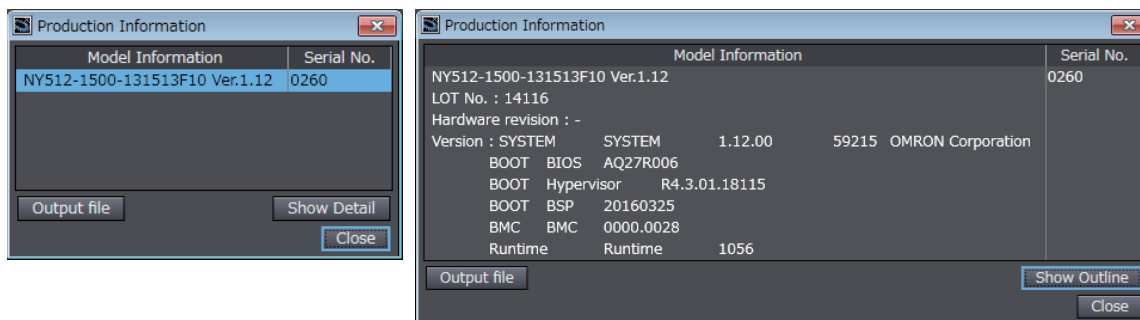
- 1 Right-click **CPU Rack** under **Configurations and Setup - CPU/Expansion Racks** in the Multiview Explorer and select **Production Information**.

The Production Information Dialog Box is displayed.

● Changing Information Displayed in Production Information Dialog Box

- 1 Click the **Show Detail** or **Show Outline** Button at the lower right of the Production Information Dialog Box.

The view will change between the production information details and outline.



Outline View

Detail View

The information that is displayed is different for the Outline View and Detail View. The Detail View displays the unit version, hardware revision, and other versions. The Outline View displays only the unit version.

● Checking the Unit Version of an EtherCAT Slave

You can use the Production Information while the Sysmac Studio is online to check the unit version of an EtherCAT slave. Use the following procedure to check the unit version.

- 1 Double-click **EtherCAT** under **Configurations and Setup** in the Multiview Explorer. Or, right-click **EtherCAT** under **Configurations and Setup** and select **Edit** from the menu.

The EtherCAT Tab Page is displayed.

- 2 Right-click the master on the EtherCAT Tab Page and select **Display Production Information**.

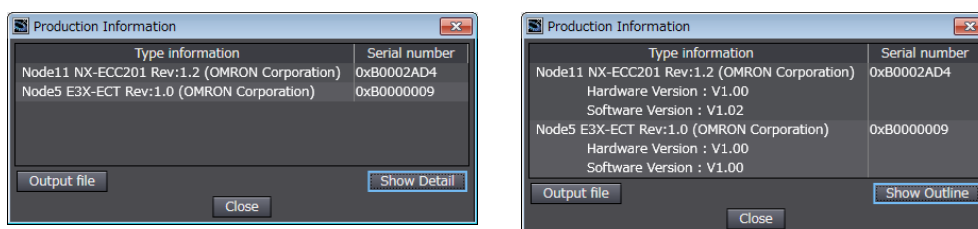
The Production Information Dialog Box is displayed.

The unit version is displayed after “Rev.”

● Changing Information Displayed in Production Information Dialog Box

- 1 Click the **Show Detail** or **Show Outline** Button at the lower right of the Production Information Dialog Box.

The view will change between the production information details and outline.



Outline View

Detail View

Unit Versions of Controllers and Peripheral Tool Versions

When you set tag data links for the built-in EtherNet/IP port on NY-series Controller, use the versions of the Network Configurator and the Sysmac Studio that are given in the following table.

CPU Unit		Network Configurator for EtherNet/IP		Sysmac Studio	
Model	Version	Version 3.59 or lower	Version 3.60	Version 1.16 or lower	Version 1.17
NY5□2-□□□□	Version 1.12 or later	NA	OK	NA	OK

Related Manuals

The followings are the manuals related to this manual. Use these manuals for reference.

Manual name	Cat. No.	Model numbers	Application	Description
NY-series IPC Machine Controller Industrial Panel PC Hardware User's Manual	W557	NY532-□□□□	Learning the basic specifications of the NY-series Industrial Panel PCs, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NY-series system is provided along with the following information on the Industrial Panel PC. <ul style="list-style-type: none"> • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NY-series IPC Machine Controller Industrial Panel PC Hardware User's Manual	W556	NY512-□□□□	Learning the basic specifications of the NY-series Industrial Box PCs, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NY-series system is provided along with the following information on the Industrial Box PC. <ul style="list-style-type: none"> • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Setup User's Manual	W568	NY532-□□□□ NY512-□□□□	Learning the initial settings of the NY-series Industrial PCs and preparations to use Controllers.	The following information is provided on an introduction to the entire NY-series system. <ul style="list-style-type: none"> • Two OS systems • Initial settings • Industrial PC Support Utility • NYCompolet • Industrial PC API • Backup and recovery
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Software User's Manual	W558	NY532-□□□□ NY512-□□□□	Learning how to program and set up the Controller functions of an NY-series Industrial PC.	The following information is provided on the NY-series Controller functions. <ul style="list-style-type: none"> • Controller operation • Controller features • Controller settings • Programming based on IEC 61131-3 language specifications
NY-series Instructions Reference Manual	W560	NY532-□□□□ NY512-□□□□	Learning detailed specifications on the basic instructions of an NY-series Industrial PC.	The instructions in the instruction set (IEC 61131-3 specifications) are described.
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Motion Control User's Manual	W559	NY532-□□□□ NY512-□□□□	Learning about motion control settings and programming concepts of an NY-series Industrial PC.	The settings and operation of the Controller and programming concepts for motion control are described.
NY-series Motion Control Instructions Reference Manual	W561	NY532-□□□□ NY512-□□□□	Learning about the specifications of the motion control instructions of an NY-series Industrial PC.	The motion control instructions are described.
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherCAT® Port User's Manual	W562	NY532-□□□□ NY512-□□□□	Using the built-in EtherCAT port in an NY-series Industrial PC.	Information on the built-in EtherCAT port is provided. This manual provides an introduction and provides information on the configuration, features, and setup.

Manual name	Cat. No.	Model numbers	Application	Description
NY-series IPC Machine Controller Industrial Panel PC / Industrial Box PC Built-in EtherNet/IP™ Port User's Manual	W563	NY532-□□□□ NY512-□□□□	Using the built-in EtherNet/IP port in an NY-series Industrial PC.	Information on the built-in EtherNet/IP port is provided. Information is provided on the basic setup, tag data links, and other features.
NJ/NY-series NC Integrated Controller User's Manual	O030	NJ501-5300 NY532-5400	Performing numerical control with NJ/NY-series Controllers.	Describes the functionality to perform the numerical control. Use this manual together with the <i>NJ/NY-series G code Instructions Reference Manual</i> (Cat. No. O031) when programming.
NJ/NY-series G code Instructions Reference Manual	O031	NJ501-5300 NY532-5400	Learning about the specifications of the G code/M code instructions.	The G code/M code instructions are described. Use this manual together with the <i>NJ/NY-series NC Integrated Controller User's Manual</i> (Cat. No. O030) when programming.
NY-series Troubleshooting Manual	W564	NY532-□□□□ NY512-□□□□	Learning about the errors that may be detected in an NY-series Industrial PC.	Concepts on managing errors that may be detected in an NY-series Controller and information on individual errors are described.
Sysmac Studio Version 1 Operation Manual	W504	SYSMAC- SE2□□□	Learning about the operating procedures and functions of the Sysmac Studio.	Describes the operating procedures of the Sysmac Studio.
CNC Operator Operation Manual	O032	SYSMAC -RTNC0□□□□D	Learning an introduction of the CNC Operator and how to use it.	An introduction of the CNC Operator, installation procedures, basic operations, connection operations, and operating procedures for main functions are described.

Revision History

A manual revision code appears as a suffix to the catalog number on the front and back covers of the manual.

Cat. No. W563-E1-05

↑
Revision code

Revision code	Date	Revised content
01	September 2016	Original production
02	April 2017	<ul style="list-style-type: none"> • Added information on functional support for unit version 1.14 of the Controllers. • Corrected mistakes.
03	October 2017	Corrected mistakes.
04	January 2019	Corrected mistakes.
05	July 2019	<ul style="list-style-type: none"> • Added information on functional support for unit version 1.21 of the Controllers. • Corrected mistakes.

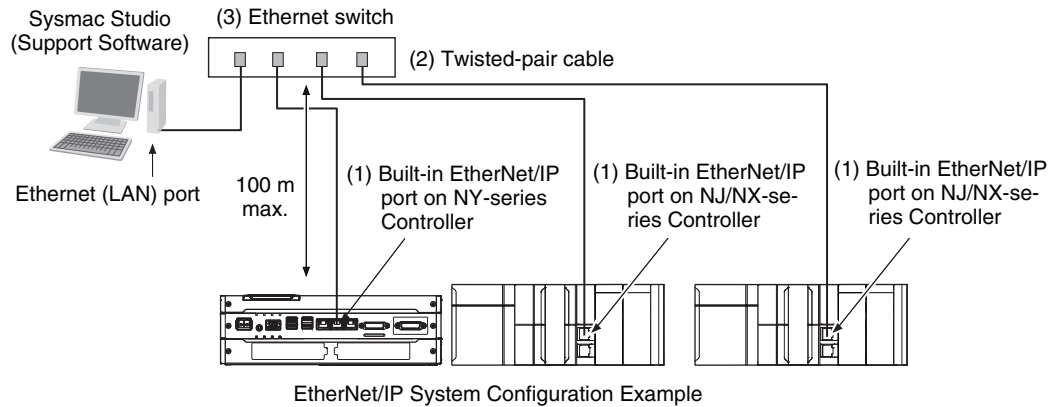
Introduction

1-1	Introduction	1-2
1-1-1	EtherNet/IP Features	1-2
1-1-2	Features of Built-in EtherNet/IP Port on NY-series Industrial PCs	1-3
1-2	System Configuration and Configuration Devices	1-5
1-2-1	Devices Required to Construct a Network	1-5
1-2-2	Support Software Required to Construct a Network	1-6
1-3	Built-in EtherNet/IP Port	1-7
1-3-1	Specifications	1-7
1-3-2	Part Names and Functions	1-10
1-4	Introduction to Communications Services	1-12
1-4-1	IP Communications with Windows	1-12
1-4-2	CIP (Common Industrial Protocol) Communications Services	1-12
1-4-3	IP Routing	1-14
1-4-4	BOOTP Client	1-17
1-4-5	FTP Server	1-17
1-4-6	FTP Client	1-18
1-4-7	Socket Service	1-19
1-4-8	Specifying Host Names	1-19
1-4-9	SNMP Agent	1-20
1-5	EtherNet/IP Communications Procedures	1-21

1-1 Introduction

1-1-1 EtherNet/IP Features

EtherNet/IP is an industrial multi-vendor network that uses Ethernet. The EtherNet/IP specifications are open standards managed by the ODVA (Open DeviceNet Vendor Association), just like DeviceNet. EtherNet/IP is not just a network between Controllers. It is also used as a field network. Because EtherNet/IP uses standard Ethernet technology, various general-purpose Ethernet devices can be used in the network.



● High-speed, High-capacity Data Exchange through Tag Data Links

The EtherNet/IP protocol supports implicit communications, which allows cyclic communications (called tag data links in this manual) with EtherNet/IP devices.

● Tag Data Link (Cyclic Communications) Cycle Time

Tag data links (cyclic communications) operate at the cyclic period specified for each application, regardless of the number of nodes. Data is exchanged over the network at the refresh cycle set for each connection, so the communications refresh cycle will not increase even if the number of nodes is increased, i.e., the concurrency of the connection's data is maintained. Because the refresh cycle can be set for each connection, each application can communicate at its ideal refresh cycle. For example, interprocess interlocks can be transferred at high speed, while the production commands and the status monitor information are transferred at low speed.

1-1-2 Features of Built-in EtherNet/IP Port on NY-series Industrial PCs

● Tag Data Links

Cyclic communications between Controllers or between Controllers and other devices are possible on an EtherNet/IP network. Tag data links can quickly perform data exchanges.

● Message Communications

You can send CIP commands to devices on the EtherNet/IP network when required by execution of CIP communications instructions in a program. As a result, it is possible to send and receive data with devices on the EtherNet/IP network.

● BOOTP Client

If the NY-series built-in EtherNet/IP port is set in the BOOTP settings, the BOOTP client operates when the Controller power is turned ON, and the IP address is obtained from the BOOTP server. It is possible to set all of the IP addresses of multiple built-in EtherNet/IP ports at the same time.

● Built-in FTP Server for File Transfers to and from Host Computers

An FTP server is built into the Controller. You can use it to read and write data within the Controller as files from workstations and computers with FTP clients. The FTP server enables the transfer of large amounts of data from a client without any additional ladder programming.

● FTP Client for File Transfers to and from Host Computers

An FTP client is built into the Controller so that you can read and write files on workstations and computers that have an FTP server from the Controller.

You can use the FTP client communications instructions to transfer one or more files between the Controller and an FTP server.

● Socket Services

Socket services can be used to send/receive data between general-purpose applications and Controllers. You can use these communications services to send and receive any data to and from remote nodes, i.e., between host computers and Controllers or between Controllers. You can execute socket communications instructions in order in a program to execute communications processes with the socket services. There are two socket services, the UDP socket service and TCP socket service.

● Host Names

You can directly specify IP addresses, but you can also use the host names instead of the IP addresses for SNMP managers, or the destinations of socket instructions and CIP communications instructions (DNS client or hosts settings). This is useful, for example, when server IP addresses change for system revisions because the IP addresses are automatically found when host names are used.

* A separate DNS server is necessary to use host names with the DNS client.

* The DNS server is specified directly using its IP address.

● Network Management with an SNMP Manager

The SNMP agent passes internal status information from the built-in EtherNet/IP port to network management software that uses an SNMP manager.

* A separate SNMP manager is necessary for network management.

● Complete Troubleshooting Functions

A variety of functions are provided to quickly identify and handle errors.

- Self-diagnosis at startup
- Event log that records the time of occurrence and other error details

● Transfer of IP Packets to Devices on Other Network Segments

The built-in EtherNet/IP ports have an IP routing function that sends IP packets to devices on other IP network segments.

- * To use the function, you must set the IP router table and default gateway settings for each device on the network appropriately for your network configuration. For details on the settings, refer to *5-1 TCP/IP Settings Display*.



Additional Information

CIP (Common Industrial Protocol)

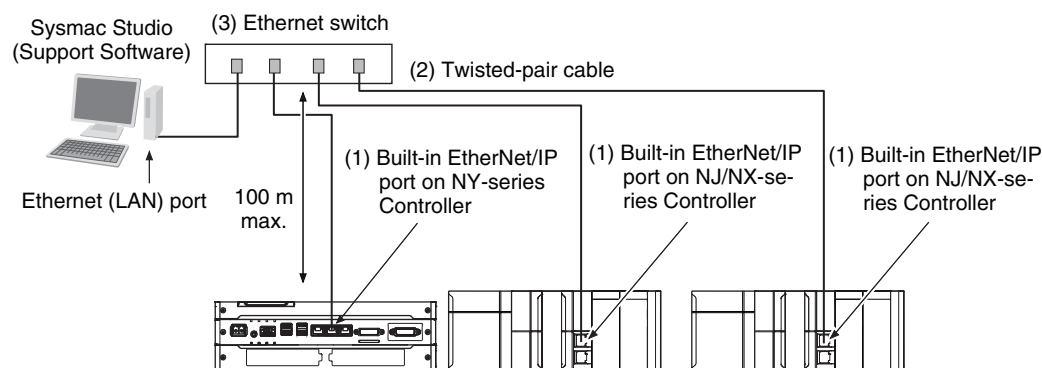
CIP is a shared industrial protocol for the OSI application layer. The CIP is used in networks such as EtherNet/IP, CompoNet, and DeviceNet. Data can be routed easily between networks that are based on the CIP. You can therefore easily configure a transparent network from the field device level to the host level. The CIP has the following advantages.

- Destination nodes are specified by a relative path, without fixed routing tables.
 - The CIP uses the producer/consumer model. Nodes in the network are arranged on the same level and it is possible to communicate with required devices whenever it is necessary. The consumer node will receive data sent from a producer node when the connection ID in the packet indicates that the node requires the data. Because the producer can send the same data with the same characteristics in a multicast format, the time required for the transfer is fixed and not dependent on the number of consumer nodes. (Either multicast or unicast can be selected.)
-

1-2 System Configuration and Configuration Devices

1-2-1 Devices Required to Construct a Network

The basic configuration for an EtherNet/IP system includes one Ethernet switch to which nodes are attached in star configuration using twisted-pair cable.



The following products are also required to build a network. Obtain them in advance.

Network device	Function
(1) Per Node NY-series (built-in EtherNet/IP port) (NY5□2-□□□□) NJ-series (built-in EtherNet/IP port) (NJ501-□□□□, 301-□□□□ or 101-□□□□) NX-series (built-in EtherNet/IP port) (NX701-□□□□) OMRON PLCs CJ2 (built-in EtherNet/IP port) (CJ2H-CPU□□-EIP or CJ2M-CPU3□) CJ-series EtherNet/IP Unit* (CJ1W-EIP21) CS-series EtherNet/IP Unit (CS1W-EIP21)	These Units are used to connect to an EtherNet/IP network.
(2) Twisted-pair cable	The twisted-pair cable has a RJ45 Modular Connector at each end. This cable is used to connect the built-in EtherNet/IP port or EtherNet/IP Unit to an Ethernet switch. Use an STP (shielded twisted-pair) cable of category 5, 5e, or higher.
(3) Ethernet switch	This is a relay device that connects multiple nodes in a star LAN. For details on recommended devices to configure a network, refer to <i>2-1-1 Recommended Network Devices</i> .

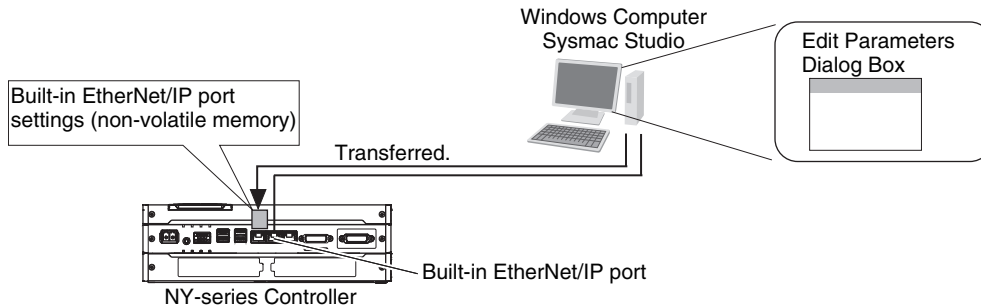
* The CJ1W-EIP21 can be mounted only to an NJ-series CPU Unit with unit version 1.01 or later and Sysmac Studio version 1.02 or higher.

1-2-2 Support Software Required to Construct a Network

This section describes the Support Software that is required to construct an EtherNet/IP network. The built-in EtherNet/IP port has Ethernet Settings and Tag Data Link Settings, which are stored in the non-volatile memory of the Controller. Support Software is provided for each, as described below.

● Built-in Ethernet/IP Settings: Sysmac Studio

Use the Sysmac Studio to set the basic settings, such as the local IP address and subnet mask of the built-in EtherNet/IP port. The Sysmac Studio can also be used to check if data I/O is being performed correctly for tag data links.



Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for details on the Sysmac Studio.

● Tag Data Link Settings: Network Configurator

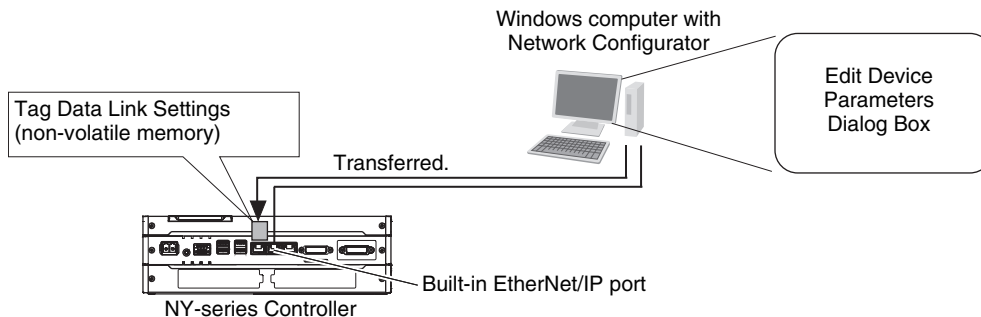
Use the Network Configurator to set the tag data links for the built-in EtherNet/IP port. (The Network Configurator is included in the Sysmac Studio Standard Edition.) The main functions of the Network Configurator are given below.

1) Setting and Monitoring Tag Data Links (Connections)

The network device configuration and tag data links (connections) can be created and edited. After connecting to the network, the device configuration and tag data link settings can be uploaded and monitored.

2) Multi-vendor Device Connections

EDS files can be installed and deleted so that you can construct, set, and manage networks that contain EtherNet/IP devices from other companies. The IP addresses of EtherNet/IP devices can also be changed.



For details on the Network Configurator, refer to *Section 7 Tag Data Link Functions*.



Additional Information

You can also use the Sysmac Studio to set the tag data links. Refer to *A-2 Use the Sysmac Studio to Set the Tag Data Links (EtherNet/IP Connections)* for details on setting the tag data links in the Sysmac Studio.

1-3 Built-in EtherNet/IP Port

1-3-1 Specifications

Item		Specifications
		NY5□2-□□□□
		Unit version 1.12 or later
Communications protocol		TCP/IP or UDP/IP
Supported services		Sysmac Studio connection, tag data link, CIP message communications, socket services, FTP server, FTP client, SNMP agent, DNS client, and BOOTP client
Number of ports		1 (With IP routing function)
Physical layer		100Base-TX, 10Base-T or 1000Base-T (1000Base-T or 100Base-TX is recommended.) *1
Transmission specifications	Media access method	CSMA/CD
	Modulation	Baseband
	Transmission paths	Star form
	Baud rate	1,000 Mbps (1000Base-T)
	Transmission media	Shielded twisted-pair (STP) cable, Category 5, 5e, or higher
	Transmission distance	100 m max. (distance between hub and node)
	Number of cascade connections	There is no limitation when an Ethernet switch is used.
CIP service: Tag data links (cyclic communications)	Number of connections	128
	Packet interval (refresh cycle)	1 to 10,000 ms in 1-ms increments Packet intervals can be set independently for each connection. (Data is refreshed over the network at the preset interval and does not depend on the number of nodes.)
	Allowed communications bandwidth per Unit	20,000 pps *2 Note The heartbeat is included.
	Number of registrable tags	256
	Tag types	Network variables CIO, Work, Holding, DM, and EM Areas cannot be used.

Item		Specifications
		NY5□2-□□□□
		Unit version 1.12 or later
CIP service: Tag data links (cyclic communications)	Number of tags per connection (= 1 tag set)	8 (7 tags when the tag set contains the Controller status)
	Maximum link data size per node	184,832 bytes
	Maximum data size per connection	1,444 bytes *3 Data concurrency is maintained within each connection. Refer to 7-1-7 <i>Concurrency of Tag Data Link Data</i> for methods to maintain concurrency.
	Number of registrable tag sets	128 (1 connection = 1 tag set)
	Maximum size of 1 tag set	722 words (The Controller status uses 1 word when the tag set contains the Controller status.)
	Changing tag data link parameters when Controller is in RUN mode	Supported *4
	Multi-cast packet filter *5	Supported
CIP message service: Explicit messages *6	Class 3 (connected)	Number of connections: 64 (clients + servers)
	UCMM (unconnected)	Number of clients that can communicate at one time: 32 max. Number of servers that can communicate at one time: 32 max.
	CIP routing	Supported. CIP routing is supported for the following remote Units: NY5□2-□□□□, NX701-□□□□, NJ501-□□□□, NJ301-□□□□, NJ101-□□□□, CS1W-EIP21, CJ1W-EIP21, CJ2H-CPU□□-EIP, and CJ2M-CPU3□. Using a combination of any Units above, communication can be extended up to a maximum of 8 levels.
SNMP	Agents	SNMPv1 or SNMPv2c
	MIB	MIB-II
EtherNet/IP conformance test		Conforms to CT13
Ethernet interface		10Base-T, 100Base-TX, or 1000Base-T Auto negotiation or fixed settings

*1 If tag data links are being used, use 100Base-TX or 1000Base-T.

*2 Here, pps means “packets per second” and indicates the number of packets that can be processed in one second.

*3 To use a data size of 505 bytes or higher, the system must support a large forward open (an optional CIP specification). The CS, CJ, NJ, NX, and NY-series Units support a large forward open, but before connecting to nodes of other companies, confirm that those devices also support it.

*4 If the parameters of the built-in EtherNet/IP port are changed, the port is restarted. When other nodes are in communications with the affected node, the communications will temporarily time out and automatically recover after the restart.

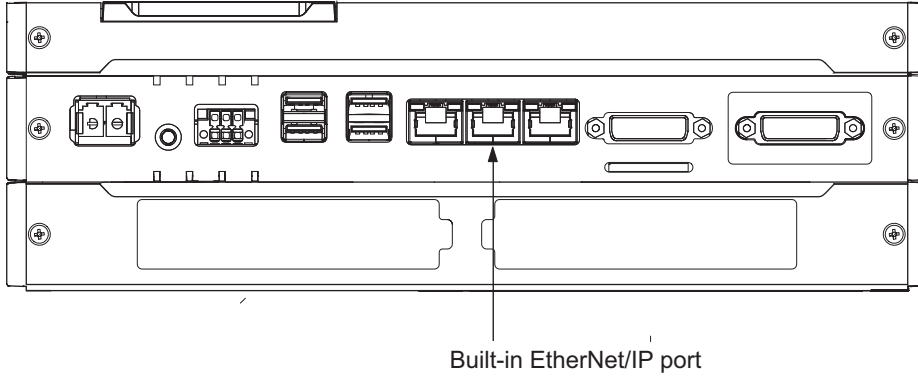
*5 Because the built-in EtherNet/IP port is equipped with an IGMP client (version 2), unnecessary multicast packets can be filtered by an Ethernet switch that supports IGMP snooping.

*6 The built-in EtherNet/IP port uses the TCP/UDP port numbers shown in the following table.
Do not set the same port number for more than one TCP/UDP service.

Service	Protocol	Port number	Remarks
EIP data links	UDP	2222	Fixed values
Used by system	UDP	2223, 2224	
	TCP	9610	
	TCP	9900	
CIP messages	TCP	44818	
FTP client data transfer port	TCP	20	
DNS client	TCP/UDP	53	
BOOTP client	UDP	68	
HTTP server	TCP	80	
Used by system, other	TCP/UDP	9600	
FTP client control port	TCP	21	
SNMP agent	UDP	161	
SNMP trap	UDP	162	

1-3-2 Part Names and Functions

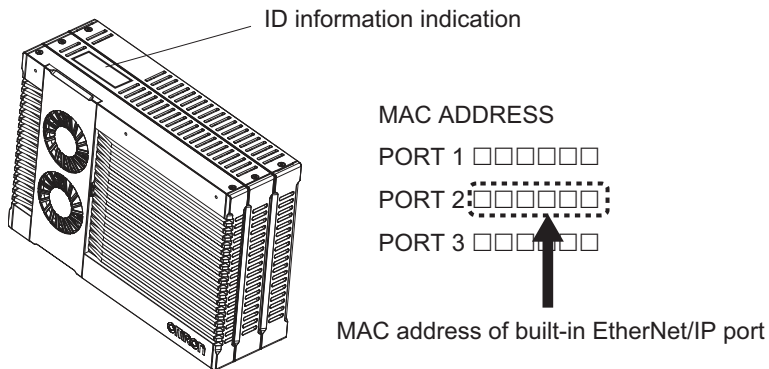
Parts and Names



MAC Address Notation

A specific MAC address is allocated to each device connected to the Ethernet network.

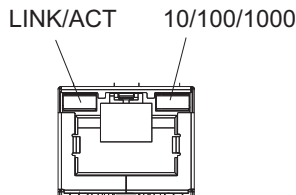
A MAC address of the built-in EtherNet/IP port is given in 6-digit hexadecimal on the ID information indication on the back side of the product as shown below.



Indicators (LEDs)

The following provides information on indicators associated with the built-in EtherNet/IP port on NY-series Controllers.

● LINK/ACT/10/100/1000



- LINK/ACT indicator: This shows the Ethernet communications status.
- 10/100/1000 indicator: This shows the baud rate status.

Indicator	Color	Status	Operating status
LINK/ACT	---	Not lit	Link not established. <ul style="list-style-type: none"> • The cable is not connected. • The power supply is OFF or was reset.
	Yellow	Flashing	Data communications in progress after establishing link.
		Lit	Links established.
10/100/1000	--	Not lit	Link established with a baud rate of 10 Mbps, or link not established.
	Green	Lit	Link established with a baud rate of 100 Mbps.
	Orange	Lit	Link established with a baud rate of 1000 Mbps.



Additional Information

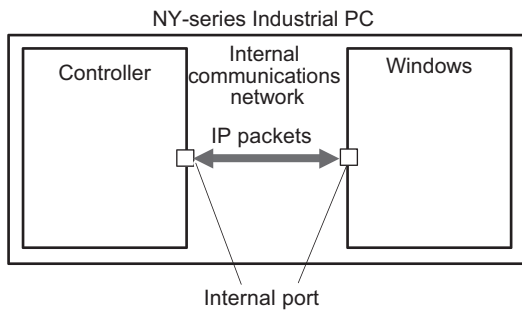
When the built-in EtherNet/IP port is set to disable, all the indicators will not light. Refer to *5-1 TCP/IP Settings Display* for the information on setting the built-in EtherNet/IP port.

1-4 Introduction to Communications Services

1-4-1 IP Communications with Windows

In an Industrial PC, both the Controller and Windows have an internal port, by which they can perform IP communications to exchange data in IP packets over the internal communications network.

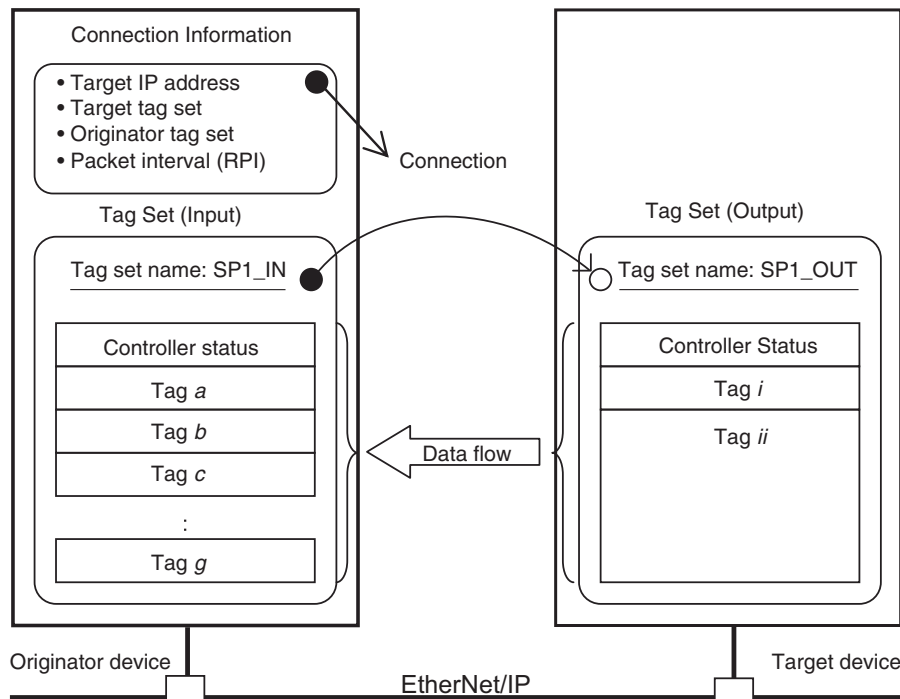
Refer to the *NY-series Industrial Panel PC / Industrial Box PC Setup User's Manual* (Cat. No. W568) for details on data exchange between Windows and the Controller.



1-4-2 CIP (Common Industrial Protocol) Communications Services

Tag Data Links (Cyclic Communications)

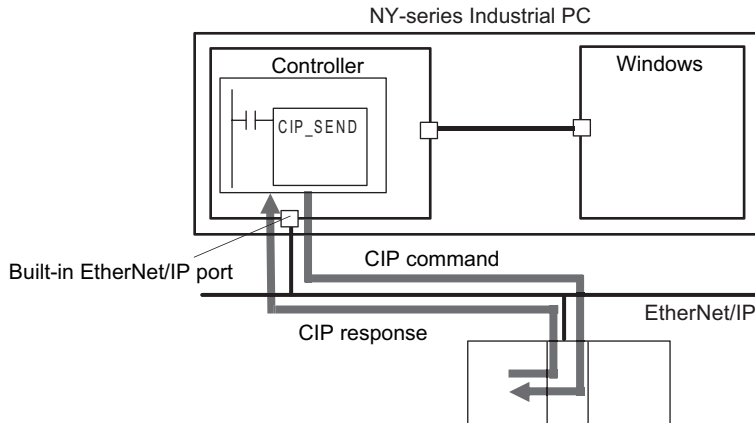
A program is not required to perform cyclic data exchanges with other devices on the EtherNet/IP network. Normally, a connection is started with the target device for each tag set that was created with the Network Configurator to start communications for tag data links for a built-in EtherNet/IP port. One connection is used per tag set.



* In this example, a connection is established with the originator's tag list with tags a to g (inputs), which are in a tag set called SP1_IN, and the target's tag list with tags i and ii (outputs), which are in a tag set called SP1_OUT.

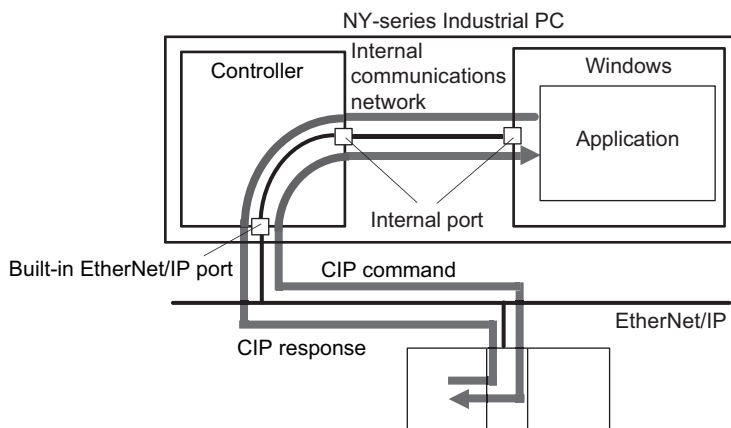
CIP Message Communications

User-specified CIP commands can be sent to devices on the EtherNet/IP network. CIP commands, such as those for reading and writing data, can be sent and their responses received by executing the CIP communications instructions from the user program in the NY-series Controller.



By specifying a route path, you can send CIP messages (CIP commands and responses) to a device on another CIP-based network segment via a built-in EtherNet/IP port or the EtherNet/IP Unit (CIP routing function for message communications). The maximum number of levels of CIP routing via the ports is eight for any combination of CS, CJ, NJ, NX, and NY-series Controller. Note that the number of levels of IP routing using an L3 Ethernet switch is not counted in the number of levels of CIP routing via the ports.

In addition, CIP messages can be routed between Windows installed in an NY-series Industrial PC and devices on an EtherNet/IP network via the internal ports and the built-in EtherNet/IP port on the Controller.



Additional Information

In CIP routing, a node (Unit) that routes information subtracts the equivalent of one hop from the timeout, deletes its own address from the route information, and relays the information to the next node (Unit).

When a timeout is specified, the timeout for the actual request service processing is set in the last hop. In the case of relay hops, the timeout for the relay route must be added to the timeout for the request.

OMRON products that support CIP subtract 5 seconds per hop.

1-4-3 IP Routing

The two built-in EtherNet/IP ports on the NY-series Controller both have the IP routing function. The IP routing function sends IP packets to other network segments based on the routing information set in the IP router table.

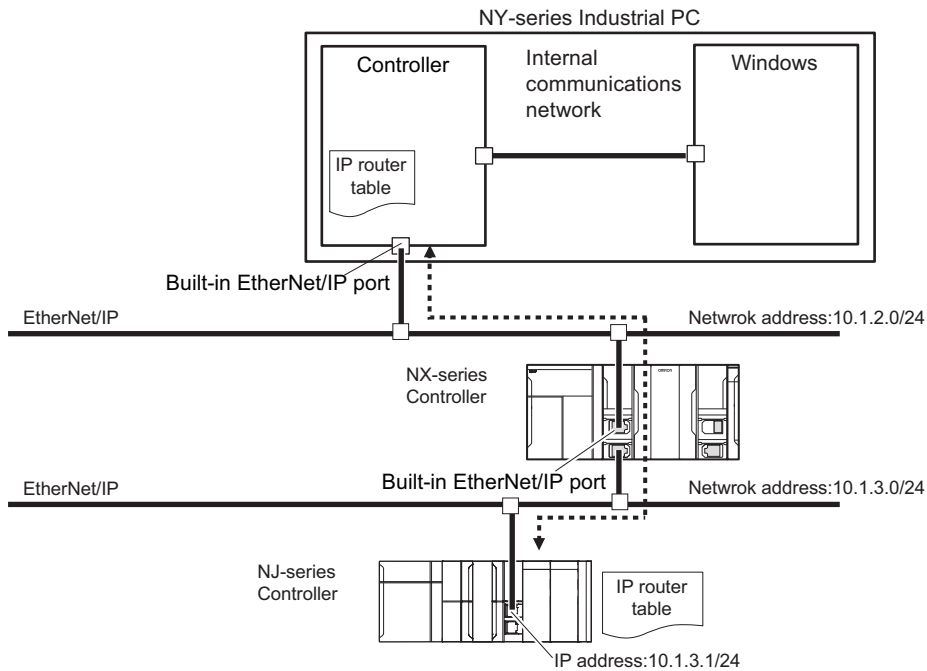
To communicate with devices on other network segments, you must set the IP router table and default gateway settings for the NY-series Controller and each device on the network appropriately for your network configuration.



Precautions for Correct Use

- You cannot create tag data links between multiple Controllers using IP routing on the NY-series Controller.

The following figure is an example of communications between the Controller in an NY-series Industrial PC and an NJ-series Controller on a different network segment. Here, you need to set the route information for communicating via the built-in EtherNet/IP port on the NX-series Controller in the IP router table and default gateway settings for the Controller in an NY-series Industrial PC.

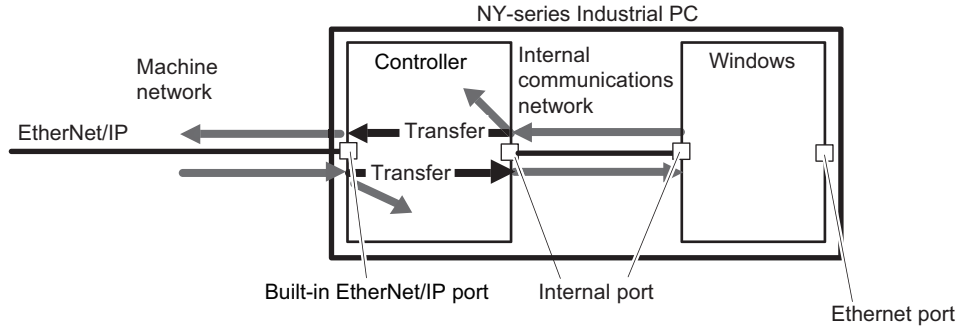


The NY-series Controller provides the following functions related to the IP routing function. Refer to *5-1 TCP/IP Settings Display* for the procedure to set each function.

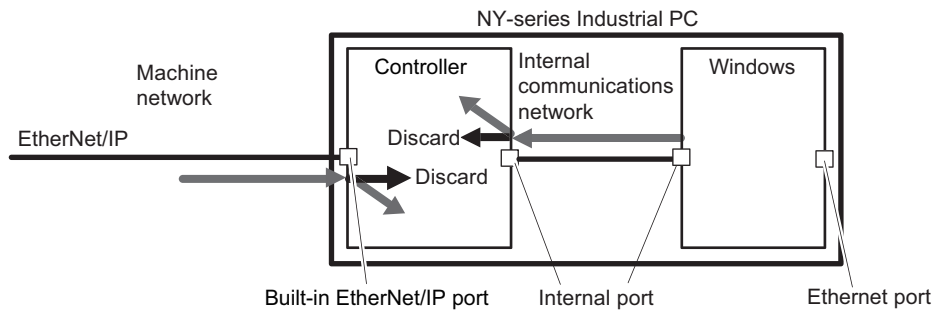
● IP Forward Function

This function transfers IP packets received from a network to another network.

It allows the Controller in an NY-series Industrial PC to relay IP packets between the internal communications network and machine networks. IP packets are transferred between the internal ports and the built-in EtherNet/IP port on the Controller to be relayed to a different network.



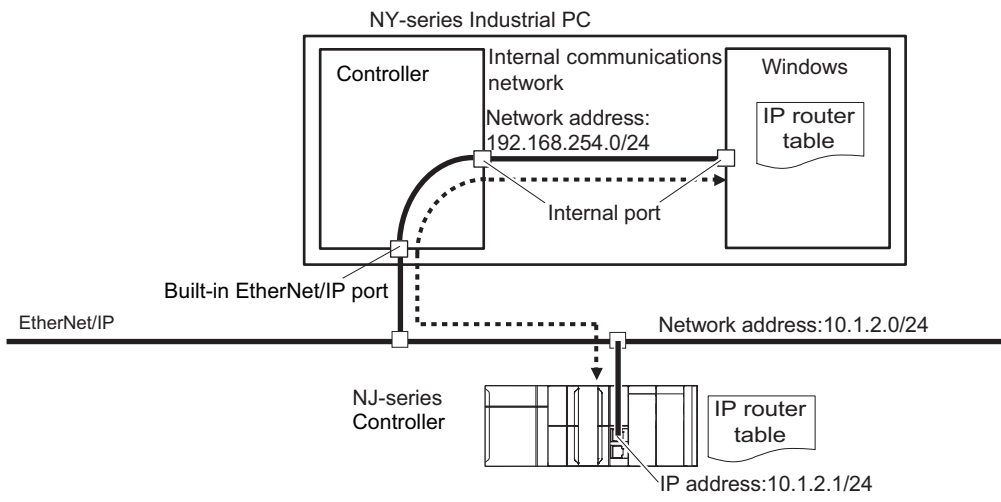
If the internal communications network needs to be divided from machine networks for security reasons, you can set this function to disable the transfer of IP packets. When the transfer of IP packets is disabled, the function discards IP packets that are not destined to the Controller.



Additional Information

You can route IP packets to EtherNet/IP devices on a network segment different from that where Windows in an NY-series Industrial PC is located.

In this case, you need the IP routing and default gateway settings in Windows.



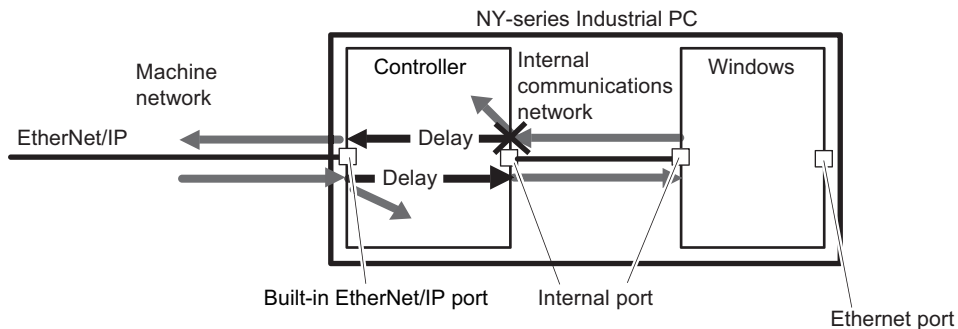
● Packet Filter Function

This function checks IP packets received from a network to determine whether they should be either received or relayed to another network.

It allows the Controller in an NY-series Industrial PC to limit the access to any machine network in which a Controller exists via the internal port, based on a condition predefined by the user.

The function checks the IP packets received by the internal port of the Controller through the internal communications network. It determines whether they can pass through the internal port based on a user-specified condition.

By default, IP packets from only networks whose source IP address is set to the IP address of the internal port will be received; otherwise, IP packets will be relayed to another network.

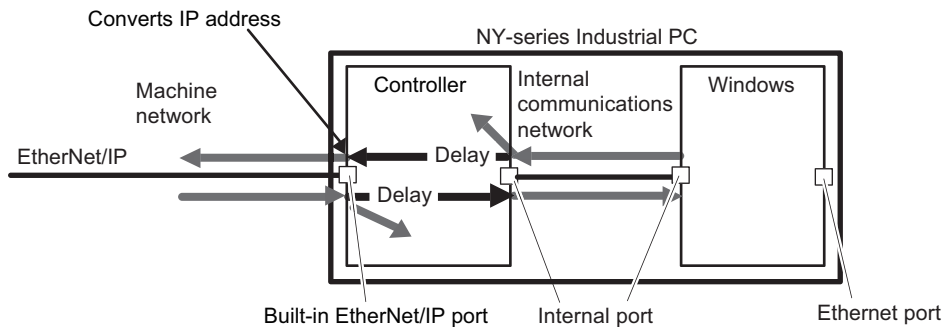


● NAT Function

This function converts the source IP address in IP packets to a different IP address when relaying IP packets from one network to another.

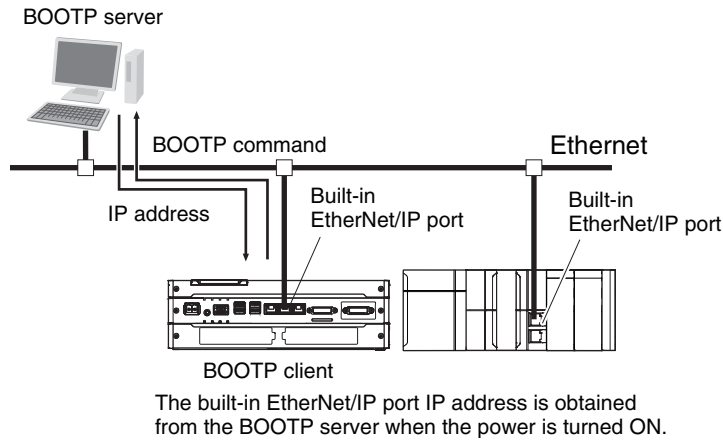
Enabling the NAT function allows the built-in EtherNet/IP port to automatically convert the source IP address in IP packets to its IP address.

You can disable the NAT function if there is no need to convert the source IP address in IP packets.



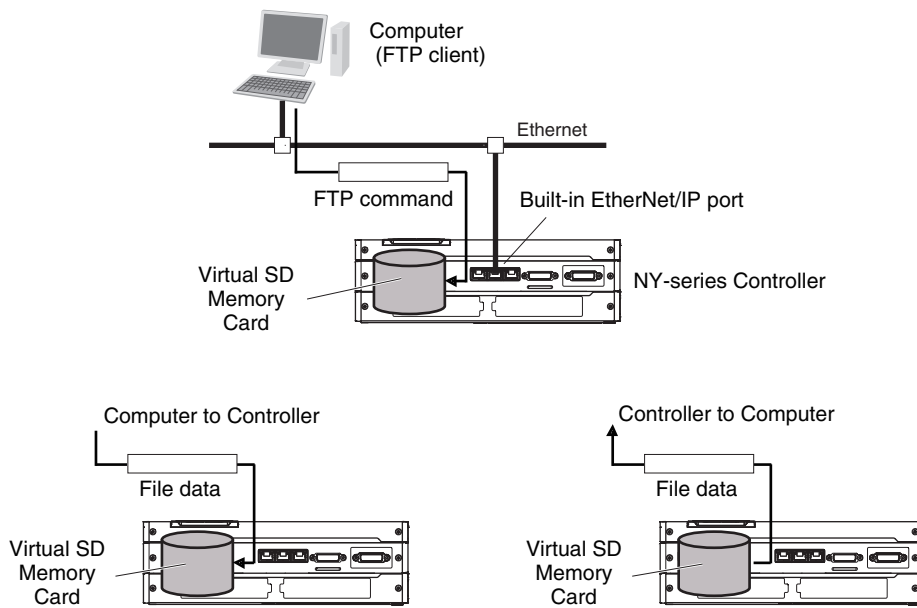
1-4-4 BOOTP Client

You set the built-in EtherNet/IP port in the BOOTP settings to use the BOOTP client to obtain settings, such as the built-in EtherNet/IP port IP address.



1-4-5 FTP Server

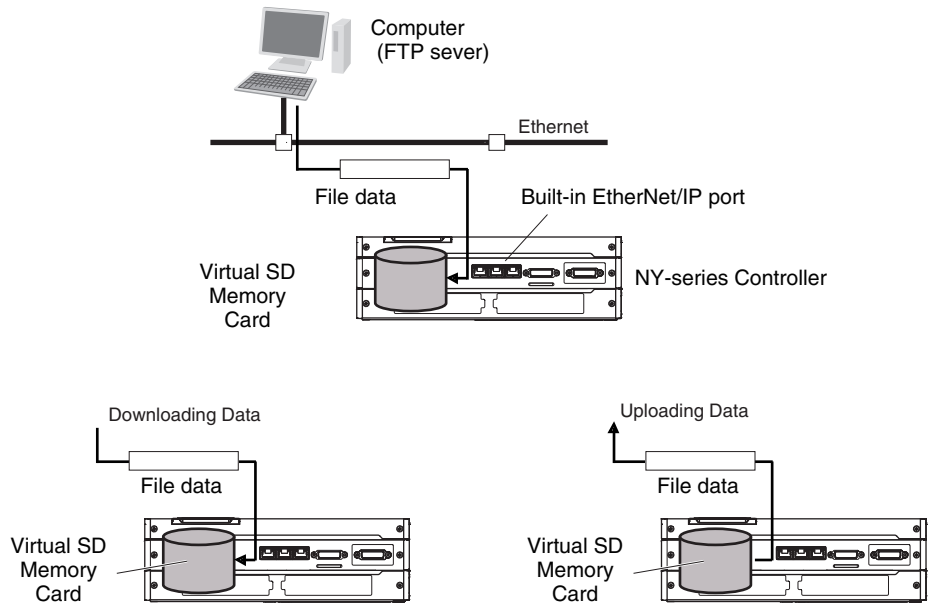
An FTP server is built into the built-in EtherNet/IP port so that files can be read from and written to the Virtual SD Memory Card in the Controller from computers at other Ethernet nodes. This makes it possible to exchange data files between a computer and the Controller with the computer as the FTP client and the Controller as the FTP server.



1-4-6 FTP Client

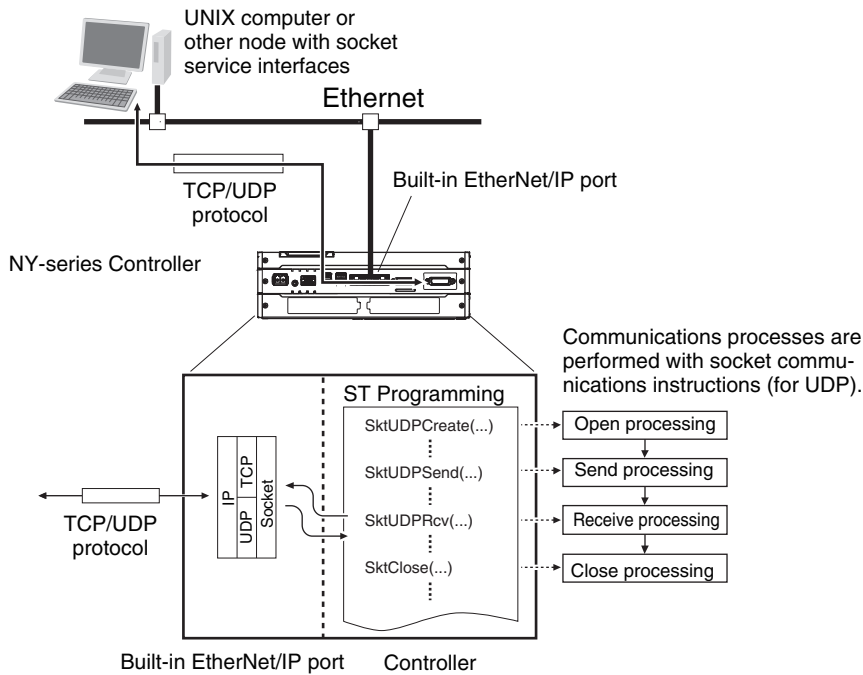
The built-in EtherNet/IP port contains an FTP client. With it, you can use FTP client communications instructions to transfer files between the Controller and computers on Ethernet.

This makes it possible to exchange data files between a computer and the Controller with the Controller as the FTP client and the computer as the FTP server.



1-4-7 Socket Service

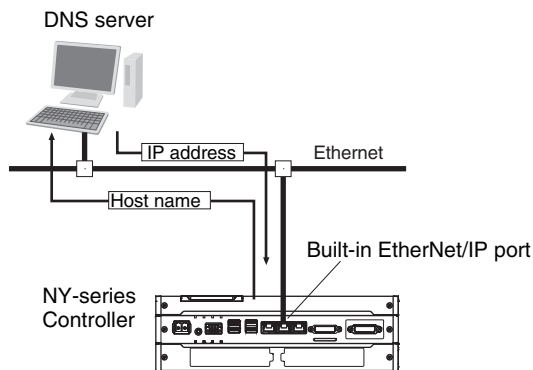
You can send data to and receive data from any node on Ethernet with the UDP or TCP protocol. To send/receive data with a socket service, you execute multiple socket communications instructions in sequence in an ST program to execute the required communications processes. After a connection with the other communications device is opened with an open instruction, the values of the variables that are specified for the send instruction are sent and the data that was received for a receive instruction is stored in the specified variables. The connection is closed with a close instruction, and communications end. For TCP, you can also read the socket status and received data. You can use a total of 16 TCP ports and UDP ports.



1-4-8 Specifying Host Names

You can directly specify IP addresses, but you can also use the host names instead of the IP addresses for SNMP managers, or the destinations of socket instructions and CIP communications instructions (DNS client or hosts settings).

Example: Setting Host Names on the DNS Server



Precautions for Correct Use

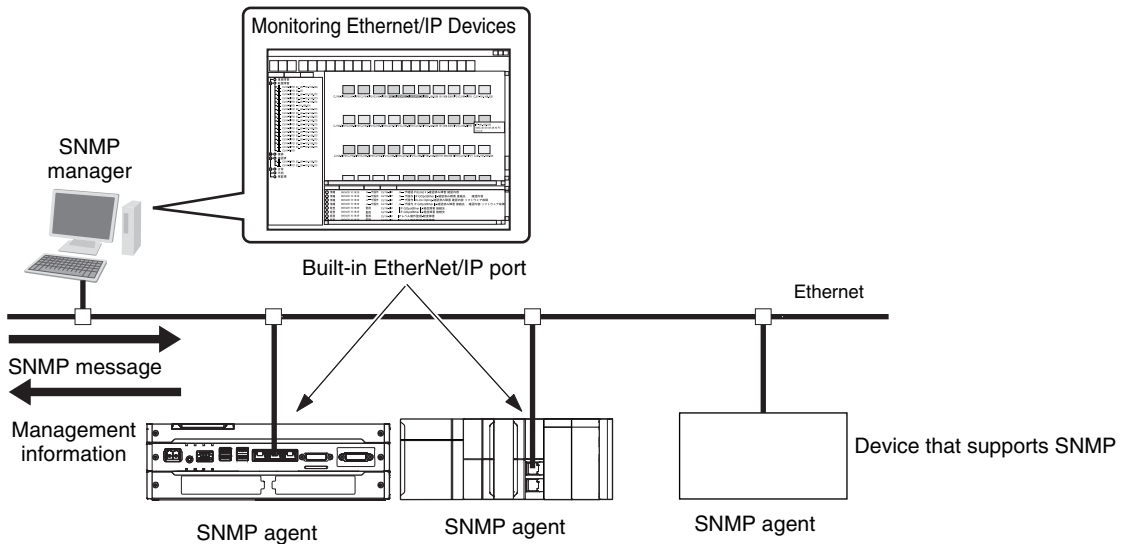
A DNS server is required to use the server host names for the DNS client.

1-4-9 SNMP Agent

The SNMP agent has the following functions.

SNMP Agent

The SNMP agent passes internal status information from the built-in EtherNet/IP port to network management software that uses an SNMP manager.

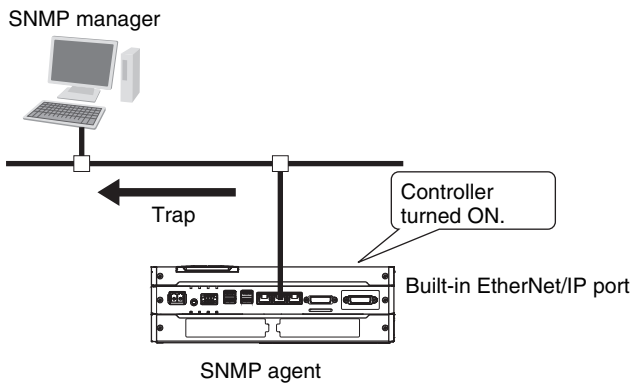


SNMP Traps

SNMP Trap

When specific conditions occur, the built-in EtherNet/IP port that is set as the SNMP agent sends status notification reports to the SNMP manager. The SNMP manager can learn about changes in status even without periodically monitoring of the built-in EtherNet/IP port. Status notification reports are sent under the following conditions.

- When the Controller is turned ON
- When links are established
- When an SNMP agent fails to be authorized



1-5 EtherNet/IP Communications Procedures

● Basic Operation

1 Wire the Ethernet network with twisted-pair cable.

Section 2 Installing Ethernet Networks



2 Set the built-in EtherNet/IP port IP address with the Sysmac Studio.

Section 4 Determining IP Addresses

1. Use the Sysmac Studio to create a new project.
2. Set the local IP address in one of the following ways:

- Defaults:

Built-in EtherNet/IP port 1	: 192.168.250.1 (subnet mask = 255.255.255.0)
Internal port	: 192.168.254.1 (subnet mask = 255.255.255.0)

- Set any IP address.
- Obtain from BOOTP server.



3 Perform a communications test with a PING command from a computer.

Section 6 Testing Communications



4 Use the Sysmac Studio to set the initial settings of the EtherNet/IP Function Module.

Section 5 Sysmac Studio Settings for the Built-in EtherNet/IP Port

Set the TCP/IP settings and Ethernet settings as required.

● Using Tag Data Links

1 Import the variable settings for the tags that were created on the Sysmac Studio to the Network Configurator.

7-2-4 Creating Tags and Tag Sets



2 Use the Network Configurator to create the tag data link table.

Section 7 Tag Data Link Functions

- Create the network configuration.
- Set the tags, tag sets, and connections.



3 Connect the Network Configurator online.



4 Download the tag data link setting.



5 Start the tag data links (the links starts automatically when power is turned ON).



6 Check operation.

- Check the built-in EtherNet/IP port indicators.
- Use the Sysmac Studio to check the communications status with the All Tag Data Link Communications Status system-defined variable.
- Use the monitor function of the Network Configurator to confirm that the tag data links are in normal operation.

1-3-2 Part Names and Functions

Section 14 Checking Communications Status of Network and Troubleshooting

● Using the CIP Message Communications Service

- CIP Communications Instructions

1 Execute CIP communications instructions in the user program.

Section 8 CIP Message Communications



2 Check operation.

- Use the Sysmac Studio to check the communications status with the end codes of the instruction (*Done*, *Error*, and *ErrorID*).

1-3-2 Part Names and Functions

Section 14 Checking Communications Status of Network and Troubleshooting

● Using the Socket Services

1 Execute the socket service instructions in the user program.

Section 9 Socket Service



2 Check operation.

- Use the Sysmac Studio to check the communications status with the end codes of the instruction (*Done*, *Error*, and *ErrorID*).

● Using the FTP Server

1 Use the Sysmac Studio to set the initial settings of the EtherNet/IP Function Module.

Section 10 FTP Server

- Set the FTP settings (enabling FTP, login name, and password).



2 Connect to the FTP server in NY-series Controller from an FTP client application.

- Input the FTP login name and password to log onto the built-in EtherNet/IP port.
- Check the event log to see if the FTP server started.



Precautions for Correct Use

To use the FTP server, you need to make the Virtual SD Memory Card settings in advance. Refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558) for information on making the Virtual SD Memory Card settings.

● Using the FTP Client

1 Execute the FTP client communications instructions in the user program.

Section 11 FTP Client



2 Check operation.

- Use the Sysmac Studio to check the communications status with the end codes of the instruction (*Done*, *Error*, and *ErrorID*).



Precautions for Correct Use

To use the FTP client, you need to make the Virtual SD Memory Card settings in advance. Refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558) for information on making the Virtual SD Memory Card settings.

● Using the SNMP Agent

1 Use the Sysmac Studio to set the initial settings of the EtherNet/IP Function Module.

Section 12 SNMP Agent

- Set the SNMP settings.
- Set the SNMP trap settings.



2 Check operation.

- Check the event log to see if the SNMP agent started.

● Using BOOTP

1 Use the Sysmac Studio to set the initial settings of the EtherNet/IP Function Module.

Section 5 Sysmac Studio Settings for the Built-in EtherNet/IP Port

- Set the BOOTP settings.



2 Check operation.

- Check the event log to see if BOOTP started.
- Check the Online system-defined variable.

2

Installing Ethernet Networks

2-1	Selecting the Network Devices	2-2
2-1-1	Recommended Network Devices	2-2
2-1-2	Ethernet Switch Types	2-3
2-1-3	Ethernet Switch Functions	2-3
2-1-4	Precautions for Ethernet Switch Selection	2-4
2-2	Network Installation	2-6
2-2-1	Basic Installation Precautions	2-6
2-2-2	Recommended Network Devices	2-6
2-2-3	Precautions When Laying Twisted-pair Cable	2-6
2-2-4	Precautions When Installing and Connecting Ethernet Switches	2-9
2-3	Connecting to the Network	2-10
2-3-1	Ethernet Connectors	2-10
2-3-2	Connecting the Cable	2-11

2-1 Selecting the Network Devices

2-1-1 Recommended Network Devices

The following table shows the devices recommended for use with the EtherNet/IP.


● Ethernet Switches

Manufacturer	Model	Description
OMRON	W4S1-03B	<ul style="list-style-type: none"> • Packet priority control (QoS): EtherNet/IP control data priority • Failure detection: Broadcast storm, LSI error detection • 10/100Base-TX • Auto negotiation • Number of ports: three for the W4S1-03B, or five each for the W4S1-05B and W4S1-05C • Failure detection output (W4S1-05C only)
	W4S1-05B	
	W4S1-05C	
Cisco Systems, Inc.	Consult the manufacturer. http://www.cisco.com/	
CONTEC USA Inc.	Consult the manufacturer. http://www.contec.com/	
Phoenix Contact USA	Consult the manufacturer. https://www.phoenixcontact.com	

● Twisted-pair Cables and Connectors

Applicable EtherNet/IP communications cables and connectors vary depending on the used baud rate. For 100Base-TX and 10Base-T, use an STP (shielded twisted-pair) cable of category 5 or higher. You can use either a straight or cross cable. For 1000Base-T, use an STP (shielded twisted-pair) cable (double shielding with aluminium tape and braiding) of category 5e or higher. You can use either a straight or cross cable.

"100Base-TX" in the "Product" column of the table below indicates that either 100Base-TX or 10Base-T can be used.

Product		Manufacturer	Model	
For 1000Base-T and 100Base-TX	Sizes and conductor pairs: AWG 24 × 4 pairs *1	Cables	Hitachi Metals, Ltd. NETSTAR-C5E SAB 0.5 × 4P CP	
			Kuramo Electric Co., Ltd. KETH-SB	
			SWCC Showa Cable Systems Co. Ltd. FAE-5004	
		RJ45 Connectors	Panduit Corporation MPS588	
For 100Base-TX	Sizes and conductor pairs: AWG 22 × 2 pairs *1	Cables	Kuramo Electric Co., Ltd. KETH-PSB-OMR	
			JMACS Japan Co., Ltd. PNET/B	
		RJ45 Assembly Connectors	OMRON 	XS6G-T421-1
	Sizes and conductor pairs: 0.5 mm × 4 pairs *1	Cables	Fujikura Ltd. F-LINK-E 0.5mm × 4P	
RJ45 Connectors		Panduit Corporation MPS588		

*1 We recommend that you use cables and connectors in above combinations.

2-1-2 Ethernet Switch Types

● Unmanaged Layer 2 (L2) Ethernet Switches

These Ethernet switches use the Ethernet MAC address to switch ports. Ordinary Ethernet switches have this function. Ethernet switch functions and settings cannot be changed.

● Managed Layer 2 (L2) Ethernet Switches

These Ethernet switches use the Ethernet MAC address to switch ports. Ethernet switch functions and settings can be changed with special software tools for Ethernet switches running on a network node. You can also collect analytical data. These Ethernet switches provide more-advanced functions than unmanaged layer 2 Ethernet switches.

2-1-3 Ethernet Switch Functions

This section describes the Ethernet switch functions that are important for an EtherNet/IP network. For a built-in EtherNet/IP port, consider whether the Ethernet switch supports these functions when you select the Ethernet switch.

- Multicast filtering
- QoS (Quality of Service) for TCP/UDP port numbers (L4)

● Multicast Filtering

Multicast filtering transfers multicast packets to the specific nodes only. This function is implemented in the Ethernet switch as IGMP snooping or GMRP. "Specific nodes" are nodes equipped with an IGMP client that have made transfer requests to the Ethernet switch. (OMRON built-in EtherNet/IP ports are equipped with an IGMP client.) When the Ethernet switch does not use multicast filtering, multicast packets are sent to all nodes, just like broadcast packets, which increases the traffic in the network. Settings must be made in the Ethernet switch to enable this function. There must be enough multicast filters for the network.

● QoS (Quality of Service) Function for TCP/UDP Port Numbers (L4)

This function controls the priority of packet transmissions so that packets can be sent with higher priority to a particular IP address or TCP (UDP) port. The TCP and UDP protocols are called transport layer protocols, leading to the name L4 (layer 4) QoS function. When tag data links and message communications are executed on the same network, tag data links can be sent at higher priority to prevent problems such as transmission delays due to message communications traffic and packet losses due to buffer overflow. Settings must be made in the Ethernet switch to enable this function and give higher priority to tag data link packets.

Support for the above two functions is as follows for the different types of Ethernet switches.

Ethernet Switch Types	Multicast filtering	L4 QoS	Remarks
Unmanaged L2 Ethernet switches	None	None	---
Managed L2 Ethernet switches	Provided.	Provided.	Both functions must be set with a special software tool.
OMRON W4S1-series Ethernet switches	None	Provided.	L4 QoS is set with a switch. No software tool is necessary.



Additional Information

If the Network Configurator is used to set the connection type in the connection settings to a multicast connection, multicast packets are used. If the connection type is set to a point-to-point connection, multicast packets are not used.

2-1-4 Precautions for Ethernet Switch Selection

The functions supported by the Ethernet switch may affect tag data link transmission delays and the settings in the Controller configurations and setup. In addition, if the Ethernet switch supports advanced functions, special settings are required for those functions. When you select an Ethernet switch, it is necessary to consider whether to select the Ethernet switch based on the kind and amount of communications you want to execute in the network. Refer to *13-2 Adjusting the Communications Load* to estimate the communications load for tag data links.

Selecting the Ethernet Switch Based on the Types of Network Communications

● Executing Tag Data Links Only

We recommend that you use an L2 Ethernet switch without multicast filtering or an L2 Ethernet switch with multicast filtering. An L2 Ethernet switch with multicast filtering prevents increased traffic due to unnecessary multicast packets, so the tag data links can operate at higher speed. If either of the following conditions exists, the amount traffic will be the same for both kinds of L2 Ethernet switches (with or without multicast filtering).

- The tag data links are set to share the same data with all nodes in the network. (The multicast packets are transferred to all nodes in the network, just like a broadcast.)
- The tag data link settings are all one-to-one (unicast) and multicast packets cannot be used.

If multicast filters are being used, settings must be made in the Ethernet switch. There must be enough multicast filters for the network being used.

● Executing Tag Data Links and Message Communications

We recommend an L2 Ethernet switch with multicast filtering and L4 QoS. If you set tag data links for higher-priority transmission, it is possible to prevent problems such as transmission delays due to message communications traffic and packet losses due to buffer overflow. You must make special settings in the Ethernet switch when using the multicast filtering function and L4 QoS function.

Selecting the Ethernet Switch Based on the Ethernet Switch's Supported Functions

● L2 Ethernet Switch without Multicast Filtering

We recommend this kind of Ethernet switch when only tag data links are executed and any of the following conditions is met.

- The tag data links are set to share the same data with all nodes in the network. (The multicast packets are transferred to all nodes in the network, just like a broadcast.)
- The tag data link settings are all one-to-one (unicast) and multicast packets cannot be used.
- There is little traffic in the tag data links.

No special settings are required for an L2 Ethernet switch without multicast filtering.

● L2 Ethernet Switch with Multicast Filtering

We recommend this kind of Ethernet switch when only tag data links are executed and the following condition is met.

- There are many 1:N links (where N represents some number of nodes in the network) in the tag data link settings, i.e., there are many multicast packets used, or there is heavy traffic in the tag data links.

Special settings are required for an L2 Ethernet switch with multicast filtering. There must be enough multicast filters for the network.

● L3 Ethernet Switch with Multicast Filtering and L4 QoS Functions

We recommend this kind of Ethernet switch when both tag data links and message communications are executed. If you set tag data links for higher-priority transmission, you can prevent problems such as transmission delays due to message communications traffic and packet losses due to buffer overflow. Special settings must be made in the Ethernet switch when using the multicast filtering function and L4 QoS function. There must be enough multicast filters for the network.

Selecting the Ethernet Switch Based on the Network Communication Speed

● Executing Tag Data Links at a Communication Speed Over 100 Mbps

If you will use data tag links with the following conditions, use an Ethernet switch with a multicast filter or an Ethernet switch that supports a communication speed of 1,000 Mbps.

- Multicast
- Communication speed over 100 Mbps

If there is an Ethernet device on the same network that communicates at a speed of 100 Mbps or less, the device may affect tag data link communication and cause tag data links to be broken, even if the device is not related to tag data link communication.



Precautions for Correct Use

- Ask the Ethernet switch manufacturer for setting procedures for the Ethernet switch.
- Install the Ethernet switch so that its environmental resistance specifications are not exceeded. Ask the Ethernet switch manufacturer for information on the environmental resistance of the Ethernet switch.

2-2 Network Installation

2-2-1 Basic Installation Precautions

- Take the greatest care when you install the Ethernet System. Be sure to follow ISO 8802-3 specifications. Be sure you understand them before attempting to install an Ethernet System.
- Unless you are already experienced in installation of communications systems, we strongly recommend that you employ a professional to install your system.
- Do not install Ethernet equipment near sources of noise. If a noisy environment is unavoidable, take adequate measures against noise interference, such as installation of network components in metal cases or the use of optical cable in the system.
- When using a shielded cable with the shields on both ends of the cable connected to connector hoods, ground loops induced by improper grounding methods may decrease noise immunity and cause device damage. To prevent ground loops caused by differences in potential between device grounding points, the reference potential between the devices must be stabilized. Design grounding appropriately so that noise current does not flow to ground lines between the devices. For grounding methods, refer to the *NY-series Industrial Box PC Hardware User's Manual* (Cat. No. W556) or *NY-series Industrial Panel PC Hardware User's Manual* (Cat. No. W557).
- To obtain information on laying EtherNet/IP cable, contact ODVA.
ODVA web site: <http://www.odva.org>
- When you install an EtherNet/IP network that combines an information network with the control system, and the communications load may be heavy due to tag data links, we recommend that you set up the network so that the load does not affect communications. For example, install the tag data links in a segment that is separate from the information network.

2-2-2 Recommended Network Devices

Refer to 2-1 *Selecting the Network Devices* for the devices recommended for use with the built-in EtherNet/IP port.

2-2-3 Precautions When Laying Twisted-pair Cable

Connecting the Shield to Connector Hoods

● Between an EtherNet/IP Port and an Ethernet Switch

Connect the shield to connector hoods as described below.

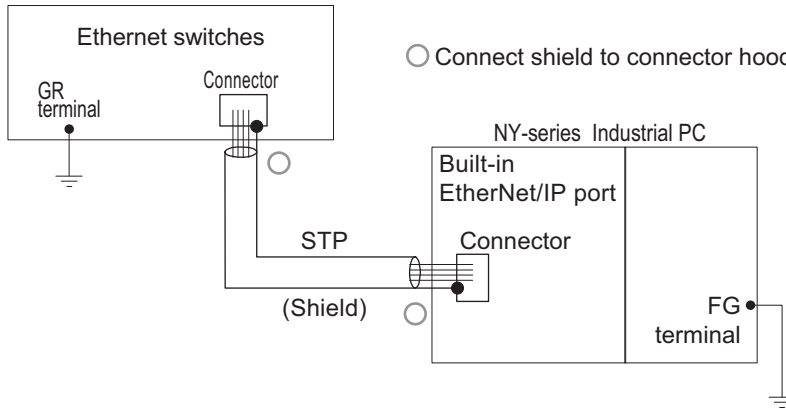
10Base-T	100Base-TX	1000Base-T
<ul style="list-style-type: none"> • Connect both ends or • Connect the Ethernet Switch side only. A clamp core must be attached to the EtherNet/IP port side of the cable. 		Connect both ends

- 10Base-T or 100Base-TX

Connect the cable shields to the connector hoods as described in either a) or b) below.

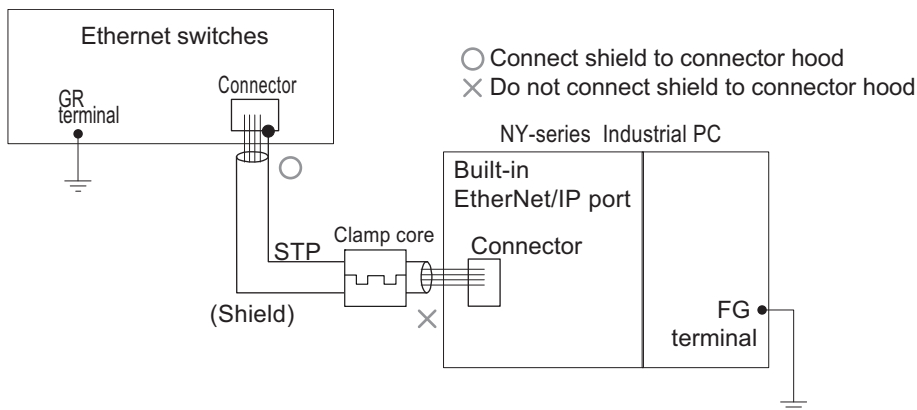
a) Connecting the shields at both ends of the cable

Connect the shields at both ends of the cables to connector hoods.



b) Connecting the shields on the Ethernet switch side only

Connect only the shield at the end of the cable on the Ethernet switch side to the connector hood. A clamp core must be attached at the end of the cable on the EtherNet/IP port side. For the recommended clamp core and attachment method, refer to *Recommended Clamp Core and Attachment Method*. To comply with EMC standards, it is mandatory that a clamp core be attached when connecting the shield only to the connector hood on the Ethernet switch side.

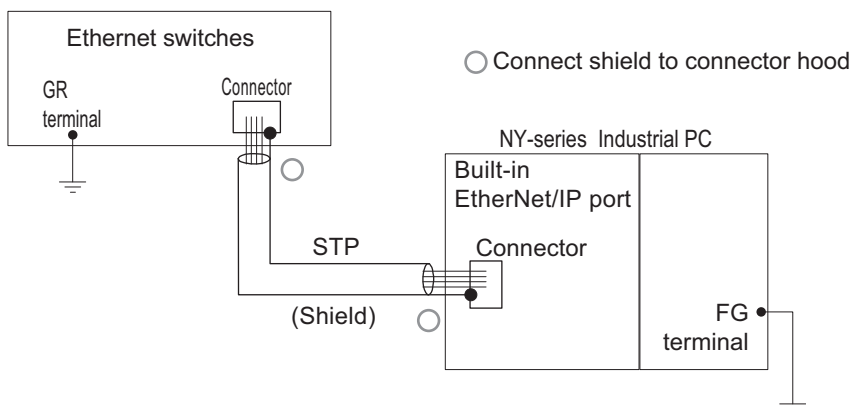


Additional Information

Noise immunity may be reduced and device damage may occur due to ground loops, which can occur due to improper shield connections and grounding methods. When using a baud rate of 100 Mbps or less, it may be possible to alleviate this problem by connecting only the Ethernet switch side as described in b), rather than connecting both ends as described in a).

• 1000Base-T

Connect the shields at the both ends of the cable to respective connector hoods. This connection is required with 1000Base-T to ensure compliance with EMC standards.



● **Between two Ethernet switches**

Regardless of which baud rate is used, check with the Ethernet switch manufacturers for information about installing the network between Ethernet switches, and in particular whether or not it is necessary to connect the cable shields to the connector hoods.

Other Precautions When Laying the Twisted-pair Cable

- Press the cable connector in firmly until it locks into place at both the Ethernet switch and the built-in EtherNet/IP port.
- Do not lay the twisted-pair cable together with high-voltage lines.
- Do not lay the twisted-pair cable near devices that generate noise.
- Do not lay the twisted-pair cable in locations subject to high temperatures or high humidity.
- Do not lay the twisted-pair cable in locations subject to excessive dirt and dust or to oil mist or other contaminants.

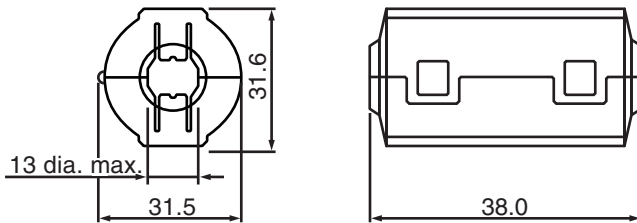
Recommended Clamp Core and Attachment Method

If you connect a shielded cable with only the Ethernet switch side connected to the connector hood, you must attach a clamp core on the built-in EtherNet/IP port side. The recommended clamp core and attachment method are given below.

● **Recommended clamp core**

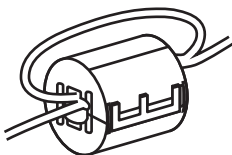
Manufacturer	Product	Model
NEC TOKIN	Clamp core	ESD-SR-250

ESD-SR-250 dimensions



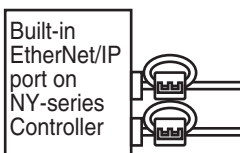
● **Recommended attachment method**

(1) **Attaching a clamp core to a communications cable**



Make two loops with the cable as shown.

(2) **Attaching a communications cable**



Attach close to the cable connection as shown.

2-2-4 Precautions When Installing and Connecting Ethernet Switches

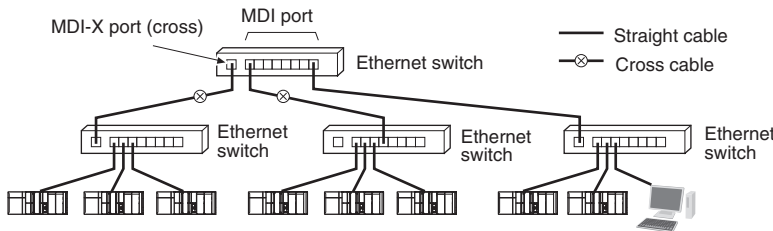
Precautions When Installing Ethernet Switches

- Do not ground the Ethernet switch in the same location as a drive-system component, such as an inverter.
- Always use a dedicated power supply for the Ethernet switch's power supply. Do not use the same power supply for other equipment, such as an I/O power supply, motor power supply, or control power supply.
- Before installation, check the Ethernet switch's environmental resistance specifications, and use an Ethernet switch that is appropriate for the ambient conditions. Contact the Ethernet switch manufacturer for details on Ethernet switch's environmental resistance specifications.

Ethernet Switch Connection Methods

- To connect Ethernet switches with twisted-pair cables, observe the followings:
 Connect an MDI port to an MDI-X port with a straight cable.
 Connect two MDI ports or two MDI-X ports with a cross cable.

Note It is very difficult to distinguish cross cables and straight cables by appearance. Incorrect cables will cause communications to fail. We recommend cascade connections with straight cables whenever possible.



- Some Ethernet switches can automatically distinguish between MDI and MDI-X. When this kind of Ethernet switch is used, straight cable can be used between Ethernet switches.



Precautions for Correct Use

Adjust the built-in EtherNet/IP port's link settings to match the communications settings of the connected Ethernet switch. If the settings do not match, the link will be unstable and prevent normal communications. The following table shows the allowed settings for each Ethernet switch communications mode. (Auto-Nego: Auto negotiation, Full: Full duplex, Half: Half duplex)

Ethernet switch	Built-in EtherNet/IP port					
	Auto-Nego	10 Mbps (fixed)		100 Mbps (fixed)		1,000 Mbps (fixed)
		Full	Half	Full	Half	Full
Auto-Nego	Best	---	OK	---	OK	---
10 Mbps (fixed)	Full	---	OK	---	---	---
	Half	OK	---	OK	---	---
100 Mbps (fixed)	Full	---	---	OK	---	---
	Half	OK	---	---	OK	---
1,000 Mbps (fixed)	Full	---	---	---	---	Best

Best = Recommended; OK = Allowed; --- = Not allowed.

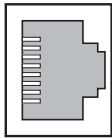
2-3 Connecting to the Network

2-3-1 Ethernet Connectors

The following standards and specifications apply to the connectors for the Ethernet twisted-pair cable.

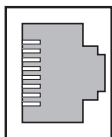
- Electrical specifications: Conforming to IEEE 802.3 standards.
- Connector structure: RJ45 8-pin Modular Connector (conforming to ISO 8877)
- For information on connecting shield wire to connector hoods, refer to *2-1-2 Ethernet Switch Types*.

10Base-T and 100Base-TX



Connector pin	Signal name	Abbr.	Signal direction
1	Transmission data +	TD+	Output
2	Transmission data –	TD–	Output
3	Reception data +	RD+	Input
4	Not used.	---	---
5	Not used.	---	---
6	Reception data –	RD–	Input
7	Not used.	---	---
8	Not used.	---	---

1000Base-T



Connector pin	Signal name	Abbr.	Signal direction
1	Communication data DA+	BI_DA+	Input/output
2	Communication data DA–	BI_DA–	Input/output
3	Communication data DB+	BI_DB+	Input/output
4	Communication data DC+	BI_DC+	Input/output
5	Communication data DC–	BI_DC–	Input/output
6	Communication data DB–	BI_DB–	Input/output
7	Communication data DD+	BI_DD+	Input/output
8	Communication data DD–	BI_DD–	Input/output

2-3-2 Connecting the Cable



Precautions for Correct Use

- Turn OFF the Controller's power supply before connecting or disconnecting Ethernet communications cable.
 - Allow extra space for the bending radius of the communications cable. For the CPU Unit dimensions when the communications cable is connected to the Unit, refer to the *NY-series Industrial Box PC Hardware User's Manual* (Cat. No. W556) or *NY-series Industrial Panel PC Hardware User's Manual* (Cat. No. W557). The required space depends on the communications cable and connector that are used. Consult the manufacturer or sales agent.
-

- 1 Lay the twisted-pair cable.
- 2 Connect the cable to the Ethernet switch.
- 3 Connect the twisted-pair cable to the connector on the built-in EtherNet/IP port. Be sure to press the connectors (both the Ethernet switch side and Ethernet side) until they lock into place.

3

System-defined Variables Related to the Built-in EtherNet/IP Port

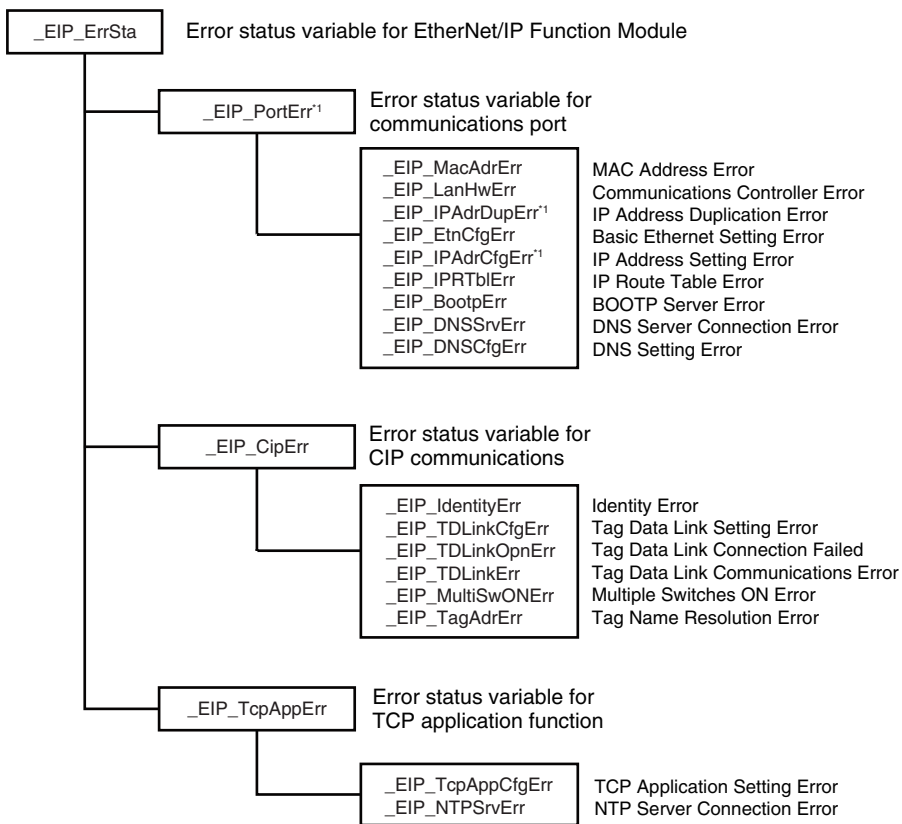
3-1	System-defined Variables Related to the Built-in EtherNet/IP Port	3-2
3-2	System-defined Variables	3-3
3-3	Specifications for Individual System-defined Variables	3-13

3-1 System-defined Variables Related to the Built-in EtherNet/IP Port

You can use the system-defined variables that are provided for the built-in EtherNet/IP port in programs to check the status of the built-in EtherNet/IP port.

● Checking for Errors in the Built-in EtherNet/IP Port

You can check for built-in EtherNet/IP port errors, internal port errors, Sysmac Studio setting errors, Network Configurator setting errors, TCP application errors (e.g., FTP or SNMP), etc. The following hierarchy is used. The system gives the error status at each level by logically ORing the error status information in the next lower level.



*1 Error status variables for errors related to NY-series Controllers are provided individually for communications port 1 and internal port 1. Refer to *Hierarchical Relationship of System-defined Variables Related to EtherNet/IP Errors in the NY-series Controller*.

3-2 System-defined Variables

The variables are described in the tables as shown below.

Variable name	Meaning	Function	Data type	Range of values	Reference
This is the system-defined variable name. The prefix gives the category name.	This is the meaning of the variable.	The function of the variable is described.	The data type of the variable is given.	The range of values that the variable can take is given.	The page of the individual system-defined variable specifications table is given.

● Functional Classification: EtherNet/IP Communications Errors

Variable name	Meaning	Function	Data type	Range of values	Reference
_EIP_ErrSta	Built-in EtherNet/IP Error	<p>This is the error status variable for the built-in EtherNet/IP port.</p> <p>NY-series Controllers: Represents the collective status of the following error flags.</p> <ul style="list-style-type: none"> • <i>_EIP1_PortErr</i> (Communications Port1 Error) • <i>_EIPIn1_PortErr</i> (Internal Port1 Error) • <i>_EIP_CipErr</i> (CIP Communications Error) • <i>_EIP_TcpAppErr</i> (TCP Application Communications Error) <p>Note Refer to information on the meanings of the error status bits at the end of this appendix for details.</p>	WORD	16#0000 to 16#00F0	page 3-13
_EIP_PortErr	Communications Port Error	<p>This is the error status variable for the communications port.</p> <p>NY-series Controllers: Represents the collective status of the following error flags.</p> <ul style="list-style-type: none"> • <i>_EIP1_MacAdrErr</i> (Port1 MAC Address Error) • <i>_EIP1_LanHwErr</i> (Port1 Communications Controller Error) • <i>_EIP1_EtnCfgErr</i> (Port1 Basic Ethernet Setting Error) • <i>_EIP1_IPAdrCfgErr</i> (Port1 IP Address Setting Error) • <i>_EIP1_IPAdrDupErr</i> (Port1 IP Address Duplication Error) • <i>_EIP1_BootpErr</i> (Port1 BOOTP Server Error) • <i>_EIP_DNSCfgErr</i> (DNS Setting Error) • <i>_EIP_DNSSrvErr</i> (DNS Server Connection Error) • <i>_EIP_IPRTblErr</i> (IP Route Table Error) <p>Note If a Link OFF Detected or Built-in EtherNet/IP Processing Error occurs, it is recorded in the event log and then the corresponding bit turns ON. Refer to information on the meanings of the error status bits at the end of this appendix for details.</p>	WORD	16#0000 to 16#00F0	page 3-14

3 System-defined Variables Related to the Built-in EtherNet/IP Port

Variable name	Meaning	Function	Data type	Range of values	Reference
_EIP1_PortErr	Communications Port1 Error	<p>This is the error status variable for the communications port.</p> <p>It represents the collective status of the following error flags.</p> <ul style="list-style-type: none"> • <i>_EIP1_MacAdrErr</i> (Port1 MAC Address Error) • <i>_EIP1_LanHwErr</i> (Port1 Communications Controller Error) • <i>_EIP1_EtnCfgErr</i> (Port1 Basic Ethernet Setting Error) • <i>_EIP1_IPAdrCfgErr</i> (Port1 IP Address Setting Error) • <i>_EIP1_IPAdrDupErr</i> (Port1 IP Address Duplication Error) • <i>_EIP1_BootpErr</i> (Port1 BOOTP Server Error) • <i>_EIP_DNSCfgErr</i> (DNS Setting Error) • <i>_EIP_DNSSrvErr</i> (DNS Server Connection Error) • <i>_EIP_IPRTblErr</i> (IP Route Table Error) <p>Note If a Link OFF Detected or Built-in EtherNet/IP Processing Error occurs, it is recorded in the event log and then the corresponding bit turns ON. Refer to information on the meanings of the error status bits at the end of this appendix for details.</p>	WORD	16#0000 to 16#00F0	page 3-14
_EIPIn1_PortErr	Internal Port1 Error	<p>This is the error status variable for the internal port 1.</p> <p>It represents the collective status of the following error flags.</p> <ul style="list-style-type: none"> • <i>_EIPIn1_IPAdrCfgErr</i> (Internal Port1 IP Address Setting Error) • <i>_EIP1_IPAdrDupErr</i> (Internal Port1 IP Address Duplication Error) • <i>_EIP_DNSCfgErr</i> (DNS Setting Error) • <i>_EIP_DNSSrvErr</i> (DNS Server Connection Error) • <i>_EIP_IPRTblErr</i> (IP Route Table Error) <p>Note If a Link OFF Detected or Built-in EtherNet/IP Processing Error occurs, it is recorded in the event log and then the corresponding bit turns ON. Refer to information on the meanings of the error status bits at the end of this appendix for details.</p>	WORD	16#0000 to 16#00F0	page 3-15
_EIP_CipErr	CIP Communications Error	<p>This is the error status variable for CIP communications.</p> <p>Note If a Tag Name Resolution Error occurs, it is recorded in the event log and this variable changes to TRUE. Refer to information on the meanings of the error status bits at the end of this appendix for details.</p>	WORD	16#0000 to 16#00F0	page 3-15

Variable name	Meaning	Function	Data type	Range of values	Reference
_EIP_TcpAppErr	TCP Application Communications Error	<p>This is the error status variable for TCP application communications.</p> <p>It represents the collective status of the following error flags.</p> <ul style="list-style-type: none"> • <i>_EIP_TcpAppCfgErr</i> (TCP Application Setting Error) • <i>_EIP_NTPSrvErr</i> (NTP Server Connection Error) <p>Note Refer to information on the meanings of the error status bits at the end of this appendix for details.</p>	WORD	16#0000 to 16#00F0	page 3-15
_EIP_MacAdrErr	MAC Address Error	<p>NY-series Controller: Indicates that an error occurred when the MAC address was read on the communications port 1 at startup.</p> <p>TRUE: Error FALSE: Normal</p>	BOOL	TRUE or FALSE	page 3-15
_EIP1_MacAdrErr	Port1 MAC Address Error	<p>Indicates that an error occurred when the MAC address was read on the communications port 1 at startup.</p> <p>TRUE: Error FALSE: Normal</p>	BOOL	TRUE or FALSE	page 3-16
_EIP_LanHwErr	Communications Controller Error	<p>NY-series Controller: Indicates that a communications controller failure occurred on the communications port 1.</p> <p>TRUE: Failure FALSE: Normal</p>	BOOL	TRUE or FALSE	page 3-16
_EIP1_LanHwErr	Port1 Communications Controller Error	<p>Indicates that a communications controller failure occurred on the communications port 1.</p> <p>TRUE: Failure FALSE: Normal</p>	BOOL	TRUE or FALSE	page 3-16
_EIP_EtnCfgErr	Basic Ethernet Setting Error	<p>NY-series Controller: Indicates that the Ethernet communications speed setting (Speed/Duplex) for the communications port 1 is incorrect. Or, a read operation failed.</p> <p>TRUE: Setting incorrect or read failed FALSE: Normal</p>	BOOL	TRUE or FALSE	page 3-16
_EIP1_EtnCfgErr	Port1 Basic Ethernet Setting Error	<p>Indicates that the Ethernet communications speed setting (Speed/Duplex) for the communications port 1 is incorrect. Or, a read operation failed.</p> <p>TRUE: Setting incorrect or read failed FALSE: Normal</p>	BOOL	TRUE or FALSE	page 3-16
_EIP_IPAdrCfgErr	IP Address Setting Error	<p>NY-series Controller: Indicates the IP address setting errors for the communications port 1.</p> <p>TRUE:</p> <ul style="list-style-type: none"> • There is an illegal IP address setting. • A read operation failed. • The IP address obtained from the BOOTP server is inconsistent. <p>FALSE: Normal</p>	BOOL	TRUE or FALSE	page 3-17

3 System-defined Variables Related to the Built-in EtherNet/IP Port

Variable name	Meaning	Function	Data type	Range of values	Reference
_EIP1_IPAdrCfgErr	Port1 IP Address Setting Error	Indicates the IP address setting errors for the communications port 1. TRUE: <ul style="list-style-type: none"> There is an illegal IP address setting. A read operation failed. The IP address obtained from the BOOTP server is inconsistent. FALSE: Normal	BOOL	TRUE or FALSE	page 3-17
_EIPIn1_IPAdrCfgErr	Internal Port1 IP Address Setting Error	Indicates the IP address setting errors for the internal port 1. TRUE: <ul style="list-style-type: none"> There is an illegal IP address setting. A read operation failed. The IP address obtained from the BOOTP server is inconsistent. FALSE: Normal	BOOL	TRUE or FALSE	page 3-17
_EIP_IPAdrDupErr	IP Address Duplication Error	NY-series Controller: Indicates that the same IP address is assigned to more than one node for the communications port 1. TRUE: Duplication occurred. FALSE: Other than the above.	BOOL	TRUE or FALSE	page 3-17
_EIP1_IPAdrDupErr	Port1 IP Address Duplication Error	Indicates that the same IP address is assigned to more than one node for the communications port 1. TRUE: Duplication occurred. FALSE: Other than the above.	BOOL	TRUE or FALSE	page 3-18
_EIPIn1_IPAdrDupErr	Internal Port1 IP Address Duplication Error	Indicates that the same IP address is assigned to more than one node for the internal port 1. TRUE: Duplication occurred. FALSE: Other than the above.	BOOL	TRUE or FALSE	page 3-18
_EIP_DNSCfgErr	DNS Setting Error	Indicates that the DNS or hosts settings are incorrect. Or, a read operation failed. TRUE: Setting incorrect or read failed FALSE: Normal	BOOL	TRUE or FALSE	page 3-18
_EIP_BootpErr	BOOTP Server Error	NY-series Controller: Indicates that a BOOTP server connection failure occurred on the communications port 1. TRUE: There was a failure to connect to the BOOTP server (timeout). FALSE: The BOOTP is not enabled, or BOOTP is enabled and an IP address was normally obtained from the BOOTP server.	BOOL	TRUE or FALSE	page 3-18
_EIP1_BootpErr	Port1 BOOTP Server Error	Indicates that a BOOTP server connection failure occurred on the communications port 1. TRUE: There was a failure to connect to the BOOTP server (timeout). FALSE: The BOOTP is not enabled, or BOOTP is enabled and an IP address was normally obtained from the BOOTP server.	BOOL	TRUE or FALSE	page 3-18

Variable name	Meaning	Function	Data type	Range of values	Reference
_EIP_IPRTblErr	IP Route Table Error	NY-series Controller: Indicates that the default gateway settings or IP router table settings are incorrect. Or, a read operation failed. TRUE: Setting incorrect or read failed FALSE: Normal	BOOL	TRUE or FALSE	page 3-18
_EIP_IdentityErr	Identity Error	NY-series Controller: Indicates that the identity information for CIP communications 1 (which you cannot overwrite) is incorrect. Or, a read operation failed. TRUE: Setting incorrect or read failed FALSE: Normal	BOOL	TRUE or FALSE	page 3-19
_EIP_TDLnkCfgErr	Tag Data Link Setting Error	NY-series Controller: Indicates that the tag data link settings for CIP communications 1 are incorrect. Or, a read operation failed. TRUE: Setting incorrect or read failed FALSE: Normal	BOOL	TRUE or FALSE	page 3-19
_EIP_TDLnkOpnErr	Tag Data Link Connection Failed	NY-series Controller: Indicates that establishing a tag data link connection for CIP communications 1 failed. TRUE: Establishing a tag data link connection failed due to one of the following causes. <ul style="list-style-type: none"> The information registered for a target node in the tag data link parameters is different from the actual node information. There was no response from the remote node. FALSE: Other than the above.	BOOL	TRUE or FALSE	page 3-19
_EIP_TDLnkErr	Tag Data Link Communications Error	NY-series Controller: Indicates that a timeout occurred in a tag data link connection for CIP communications 1. TRUE: A timeout occurred. FALSE: Other than the above.	BOOL	TRUE or FALSE	page 3-19
_EIP_TagAdrErr	Tag Name Resolution Error	NY-series Controller: Indicates that tag resolution for CIP communications 1 failed (i.e., the address could not be identified from the tag name). TRUE: Tag resolution failed (i.e., the address could not be identified from the tag name). The following causes are possible. <ul style="list-style-type: none"> The size of the network variable is different from the tag settings. The I/O direction that is set in the tag data link settings does not agree with the I/O direction of the variable in the Controller. There is no network variable in the Controller that corresponds to the tag setting. FALSE: Other than the above.	BOOL	TRUE or FALSE	page 3-20
_EIP_MultiSwONerr	Multiple Switches ON Error	NY-series Controller: Indicates that more than one switch turned ON at the same time in CIP communications 1. TRUE: More than one data link start/stop switch changed to TRUE at the same time. FALSE: Other than the above.	BOOL	TRUE or FALSE	page 3-20

Variable name	Meaning	Function	Data type	Range of values	Reference
_EIP_TcpAppCfgErr	TCP Application Setting Error	TRUE: At least one of the set values for a TCP application (FTP, NTP, SNMP) is incorrect. Or, a read operation failed. FALSE: Normal	BOOL	TRUE or FALSE	page 3-20
_EIP_NTPSrvErr	NTP Server Connection Error	Always FALSE for an NY-series Controller.	BOOL	TRUE or FALSE	page 3-20
_EIP_DNSSrvErr	DNS Server Connection Error	TRUE: The DNS client failed to connect to the server (timeout). FALSE: DNS is not enabled. Or, DNS is enabled and the connection was successful.	BOOL	TRUE or FALSE	page 3-20



Additional Information

Hierarchical Relationship of System-defined Variables Related to EtherNet/IP Errors in the NY-series Controller

The system-defined variables that are related to EtherNet/IP errors have the following hierarchical relationship. For example, if the value of any of the *_EIP1_PortErr*, *_EIPIn1_PortErr*, and *_EIP_TcpAppErr* variables in the second level is TRUE, then the *_EIP_ErrSta* variable in the first level also changes to TRUE. Therefore, you can check the values of system-defined variables in a higher level to see if an error has occurred for a variable in a lower level.

Level 1		Level 2		Level 3	
Variable	Name	Variable	Name	Variable	Name
_EIP_ErrSta	Built-in EtherNet/IP Error	_EIP1_PortErr	Communications Port1 Error	_EIP1_MacAdrErr	Port1 MAC Address Error
				_EIP1_LanHwErr	Port1 Communications Controller Error
				_EIP1_EtnCfgErr	Port1 Basic Ethernet Setting Error
				_EIP1_IPAdrCfgErr	Port1 IP Address Setting Error
				_EIP1_IPAdrDupErr	Port1 IP Address Duplication Error
				_EIP1_BootpErr	Port1 BOOTP Server Error
				_EIP_DNSCfgErr	DNS Setting Error
				_EIP_DNSSrvErr	DNS Server Connection Error
		_EIP_IPRTblErr	IP Route Table Error		
		_EIPIn1_PortErr	Internal Port1 Error	_EIPIn1_IPAdrCfgErr	Internal Port1 IP Address Setting Error
				_EIPIn1_IPAdrDupErr	Internal Port1 IP Address Duplication Error
				_EIP_DNSCfgErr	DNS Setting Error
				_EIP_DNSSrvErr	DNS Server Connection Error
		_EIP_TcpAppErr	TCP Application Communications Error	_EIP_TcpAppCfgErr	(TCP Application Setting Error)
				_EIP_NTPSrvErr	NTP Server Connection Error

Note You can access the same values of the system-defined variables whose variable names with *_EIP1* and the system-defined variables whose variable names with *_EIP*. For example, you can access the same values of *_EIP1_PortErr* (Communications Port1 Error) and *_EIP_PortErr* (Communications Port Error).

● Meanings of Error Status Bits

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORD			-	-	-	-	-	-					-	-	-	-

Bit	Meaning
15	Reserved)
14	Collective slave error status: This bit indicates if a Controller error was detected for levels (e.g., a Unit, slave, axis, or axes group) that are lower than the event source (i.e., for a function module). TRUE: A Controller error has occurred at a lower level. FALSE: A Controller error has not occurred at a lower level. (Valid for <i>_MC_ErrSta</i> , and <i>_EC_ErrSta</i> .)
8 to 13	Reserved.
7	This bit indicates whether a major fault level Controller error has occurred. TRUE: A major fault level Controller error has occurred. FALSE: A major fault level Controller error has not occurred.
6	This bit indicates whether a partial fault level Controller error has occurred. TRUE: A partial fault level Controller error has occurred. FALSE: A partial fault level Controller error has not occurred.
5	This bit indicates whether a minor fault level Controller error has occurred. TRUE: A minor fault level Controller error has occurred. FALSE: A minor fault level Controller error has not occurred.
4	This bit indicates whether an observation level Controller error has occurred. TRUE: An observation level Controller error has occurred. FALSE: An observation level Controller error has not occurred.
0 to 3	Reserved.

Note Bits 14 and 15 are never TRUE for the built-in EtherNet/IP port.



Precautions for Correct Use

Do not use *_ErrSta* (Controller Error Status) in the user program. There may be a delay in updating them and concurrency problems may occur with function module error status. Use this variable only to access status through communications from an external device.

● Functional Classification: EtherNet/IP Communications Status

You can check the status of the built-in EtherNet/IP port (e.g., communications status).

Variable name	Meaning	Function	Data type	Range of values	Reference
<i>_EIP_EtnOnlineSta</i>	Online	NY-series Controller: Indicates that the built-in EtherNet/IP port's communications can be used via the communications port 1 (that is, the link is ON, IP address is defined, and there are no errors). TRUE: The built-in EtherNet/IP port's communications can be used. FALSE: The built-in EtherNet/IP port's communications is disabled due to an error in initial processing, restart processing, or link OFF status.	BOOL	TRUE or FALSE	page 3-21

3 System-defined Variables Related to the Built-in EtherNet/IP Port

Variable name	Meaning	Function	Data type	Range of values	Reference
_EIP1_EtnOnlineSta	Port1 Online	Indicates that the built-in EtherNet/IP port's communications can be used via the communications port 1 (that is, the link is ON, IP address is defined, and there are no errors). TRUE: The built-in EtherNet/IP port's communications can be used. FALSE: The built-in EtherNet/IP port's communications is disabled due to an error in initial processing, restart processing, or link OFF status.	BOOL	TRUE or FALSE	page 3-21
_EIPIn1_EtnOnlineSta	Internal Port1 Online	Indicates that the built-in EtherNet/IP port's communications can be used via the internal port 1 (that is, the link is ON, IP address is defined, and there are no errors.) TRUE: The built-in EtherNet/IP port's communications can be used. FALSE: The built-in EtherNet/IP port's communications is disabled due to an error in initial processing, restart processing, or link OFF status.	BOOL	TRUE or FALSE	page 3-21
_EIP_TDLINKRunSta	Tag Data Link Communications Status	NY-series Controller: Indicates that at least one connection is in normal operation in CIP communications 1. TRUE: Normal operation FALSE: Other than the above.	BOOL	TRUE or FALSE	page 3-21
_EIP_TDLINKAllRunSta	All Tag Data Link Communications Status	NY-series Controller: Indicates that all tag data links are communicating in CIP communications 1. TRUE: Tag data links are communicating in all connections as the originator. FALSE: An error occurred in at least one connection.	BOOL	TRUE or FALSE	page 3-21
_EIP_RegTargetSta [255]	Registered Target Node Information	NY-series Controller: Gives a list of nodes for which built-in EtherNet/IP connections are registered for CIP communications 1. This variable is valid only when the built-in EtherNet/IP port is the originator. <i>Array[x]</i> is TRUE: The connection to the node with a target node ID of x is registered. <i>Array[x]</i> is FALSE: The connection to the node with a target node ID of x is not registered.	ARRAY [0..255] OF BOOL	TRUE or FALSE	page 3-22
_EIP_EstbTargetSta [255]	Normal Target Node Information	NY-series Controller: Gives a list of nodes that have normally established EtherNet/IP connections for CIP communications 1. <i>Array[x]</i> is TRUE: The connection to the node with a target node ID of x was established normally. <i>Array[x]</i> is FALSE: The connection to the node with a target node ID of x was not established, or an error occurred.	ARRAY [0..255] OF BOOL	TRUE or FALSE	page 3-22

Variable name	Meaning	Function	Data type	Range of values	Reference
_EIP_TargetPLC-ModeSta [255]	Target PLC Operating Mode	<p>NY-series Controller: Shows the operating status of the target node Controllers that are connected for CIP communications 1, with the built-in EtherNet/IP port as the originator.</p> <p>The array elements are valid only when the corresponding Normal Target Node Information is TRUE. If the corresponding Normal Target Node Information is FALSE, the Target Node Controller Operating Information indicates the previous operating status.</p> <p><i>Array[x]</i> is TRUE: This is the operating state of the target Controller with a node address of x.</p> <p><i>Array[x]</i> is FALSE: Other than the above.</p>	ARRAY [0..255] OF BOOL	TRUE or FALSE	page 3-22
_EIP_TargetPLCErr [255]	Target PLC Error Information	<p>NY-series Controller: Shows the error status (logical OR of fatal and non-fatal errors) of the target node Controllers that are connected for CIP communications 1, with the built-in EtherNet/IP ports as the originator. The array elements are valid only when the corresponding Normal Target Node Information is TRUE. The immediately preceding value is retained if this variable is FALSE.</p> <p><i>Array[x]</i> is TRUE: A fatal or non-fatal error occurred in the target Controller with a target node ID of x.</p> <p><i>Array[x]</i> is FALSE: Other than the above.</p>	ARRAY [0..255] OF BOOL	TRUE or FALSE	page 3-22
_EIP_TargetNodeErr [255]	Target Node Error Information	<p>NY-series Controller: Indicates that the connection for the Registered Target Node Information for CIP communications 1 was not established or that an error occurred in the target Controller.</p> <p>The array elements are valid only when the Registered Target Node Information is TRUE.</p> <p><i>Array[x]</i> is TRUE: A connection was not normally established with the target node for a target node ID of x (the Registered Target Node Information is TRUE and the Normal Target Node Information is FALSE), or a connection was established with the target node but an error occurred in the target Controller.</p> <p><i>Array[x]</i> is FALSE: The target node is not registered for a target node ID of x (the Registered Target Node Information is FALSE), or a connection was normally established with the target node (the Registered Target Node Information is TRUE and the Normal Target Node Information is TRUE). An error occurred in the target Controller (the Target PLC Error Information is TRUE).</p>	ARRAY [0..255] OF BOOL	TRUE or FALSE	page 3-23
_EIP_NTPResult	NTP Operation Information	---	_sNTP_RESULT		page 3-23

Variable name	Meaning	Function	Data type	Range of values	Reference
.ExecTime	NTP Last Operation Time	NY-series Controller: No change from the initial value.	DATE_AND_TIME	Depends on data type.	page 3-23
.ExecNormal	NTP Operation Result	NY-series Controller: No change from the initial value.	BOOL	TRUE or FALSE	page 3-23



Additional Information

Communications Status with Target Node

The communications status with the target node of an NY-series Controller is shown by the combination of the values of four system-defined variables.

- *_EIP_RegTargetSta* (Registered Target Node Information)
- *_EIP_EstbTargetSta* (Normal Target Node Information)
- *_EIP_TargetPLCErr* (Target PLC Error Information)
- *_EIP_TargetNodeErr* (Target Node Error Information)

Value of <i>_EIP_RegTargetSta</i>	Value of <i>_EIP_EstbTargetSta</i>	Value of <i>_EIP_TargetPLCErr</i>	Value of <i>_EIP_TargetNodeErr</i>	Communications status with target node
TRUE	TRUE	FALSE	FALSE	A connection with the target node was established normally and there is no error in the target PLC.
		TRUE	TRUE	A connection with the target node was established but there is an error in the target PLC.
	FALSE	Disabled	TRUE	A connection with the target node was not established normally.
FALSE	Disabled	Disabled	Disabled	The information is not valid because the target node is not registered.

● **Functional Classification: EtherNet/IP Communications Switches**

You can start and stop tag data links.

Variable name	Meaning	Function	Data type	Range of values	Reference
<i>_EIP_TDLINKStartCmd</i>	Tag Data Link Communications Start Switch	NY-series Controller: Change this variable to TRUE to start tag data links for CIP communications 1. It automatically changes back to FALSE after tag data link operation starts. Note Do not force this switch to change to FALSE from the user program or from the Sysmac Studio. It changes to FALSE automatically.	BOOL	TRUE or FALSE	page 3-23
<i>_EIP_TDLINKStopCmd</i>	Tag Data Link Communications Stop Switch	NY-series Controller: Change this variable to TRUE to stop tag data links for CIP communications 1. It automatically changes back to FALSE after tag data link operation stops. Note Do not force this switch to change to FALSE from the user program or from the Sysmac Studio. It changes to FALSE automatically.	BOOL	TRUE or FALSE	page 3-23

3-3 Specifications for Individual System-defined Variables

The specifications for each system-defined variable are given as described below.

Variable name	This is the system-defined variable name. The prefix gives the category name.		Members	The member names are given for structure variables.	
Meaning	This is the meaning of the variable.		Global/local	Global: Global variable, Local: Local variable	
Function	The function of the variable is described.				
Data type	The data type of the variable is given.		Range of values	The range of values that the variable can take is given.	
R/W access	R: Read only, RW: Read/write	Retained	The Retain attribute of the variable is given.	Network Publish	The Network Publish attribute of the variable is given.
Usage in user program	Whether you can use the variable directly in the user program is specified.	Related instructions	The instructions that are related to the variable are given. If you cannot use the variable directly in the user program, the instructions that access the variable are given.		

● Functional Classification: EtherNet/IP Communications Errors

Variable name	_EIP_ErrSta				
Meaning	Built-in EtherNet/IP Error		Global/local	Global	
Function	<p>This is the error status variable for the built-in EtherNet/IP port.</p> <p>NY-series Controllers: Represents the collective status of the following error flags.</p> <ul style="list-style-type: none"> • <i>_EIP1_PortErr</i> (Communications Port1 Error) • <i>_EIPIn1_PortErr</i> (Internal Port1 Error) • <i>_EIP_CipErr</i> (CIP Communications Error) • <i>_EIP_TcpAppErr</i> (TCP Application Communications Error) <p>Note Refer to <i>Meanings of Error Status Bits</i> for the meanings of the error status bits.</p>				
Data type	WORD		Range of values	16#0000 to 16#00F0	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	<p>You can access this variable from the user program with the following instruction.</p> <ul style="list-style-type: none"> • GetEIPError 		

3 System-defined Variables Related to the Built-in EtherNet/IP Port

Variable name	_EIP_PortErr				
Meaning	Communications Port Error			Global/local	Global
Function	<p>This is the error status variable for the communications port.</p> <p>NY-series Controllers: Represents the collective status of the following error flags.</p> <ul style="list-style-type: none"> • <i>_EIP1_MacAdrErr</i> (Port1 MAC Address Error) • <i>_EIP1_LanHwErr</i> (Port1 Communications Controller Error) • <i>_EIP1_EtnCfgErr</i> (Port1 Basic Ethernet Setting Error) • <i>_EIP1_IPAdrCfgErr</i> (Port1 IP Address Setting Error) • <i>_EIP1_IPAdrDupErr</i> (Port1 IP Address Duplication Error) • <i>_EIP1_BootpErr</i> (Port1 BOOTP Server Error) • <i>_EIP_DNSCfgErr</i> (DNS Setting Error) • <i>_EIP_DNSSrvErr</i> (DNS Server Connection Error) • <i>_EIP_IPRTblErr</i> (IP Route Table Error) <p>Note If a Link OFF Detected or Built-in EtherNet/IP Processing Error occurs, it is recorded in the event log and then corresponding bit turns ON. Refer to <i>Meanings of Error Status Bits</i> for the meanings of the error status bits.</p>				
Data type	WORD			Range of values	16#0000 to 16#00F0
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	<p>You can access this variable from the user program with the following instruction.</p> <ul style="list-style-type: none"> • GetEIPError 		

Variable name	_EIP1_PortErr				
Meaning	Communications Port1 Error			Global/local	Global
Function	<p>This is the error status variable for the communications port 1.</p> <p>It represents the collective status of the following error flags.</p> <ul style="list-style-type: none"> • <i>_EIP1_MacAdrErr</i> (Port1 MAC Address Error) • <i>_EIP1_LanHwErr</i> (Port1 Communications Controller Error) • <i>_EIP1_EtnCfgErr</i> (Port1 Basic Ethernet Setting Error) • <i>_EIP1_IPAdrCfgErr</i> (Port1 IP Address Setting Error) • <i>_EIP1_IPAdrDupErr</i> (Port1 IP Address Duplication Error) • <i>_EIP1_BootpErr</i> (Port1 BOOTP Server Error) • <i>_EIP_DNSCfgErr</i> (DNS Setting Error) • <i>_EIP_DNSSrvErr</i> (DNS Server Connection Error) • <i>_EIP_IPRTblErr</i> (IP Route Table Error) <p>Note If a Link OFF Detected or Built-in EtherNet/IP Processing Error occurs, it is recorded in the event log and then corresponding bit turns ON. Refer to <i>Meanings of Error Status Bits</i> for the meanings of the error status bits.</p>				
Data type	WORD			Range of values	16#0000 to 16#00F0
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	<p>You can access this variable from the user program with the following instruction.</p> <ul style="list-style-type: none"> • GetEIPError 		

Variable name	_EIPIn1_PortErr				
Meaning	Internal Port1 Error		Global/local	Global	
Function	This is the error status variable for the internal port 1. It represents the collective status of the following error flags. <ul style="list-style-type: none"> • <i>_EIPIn1_IPAdrCfgErr</i> (Internal Port1 IP Address Setting Error) • <i>_EIP1_IPAdrDupErr</i> (Internal Port1 IP Address Duplication Error) • <i>_EIP_DNSCfgErr</i> (DNS Setting Error) • <i>_EIP_DNSSrvErr</i> (DNS Server Connection Error) • <i>_EIP_IPRTblErr</i> (IP Route Table Error) <p>Note If a Link OFF Detected or Built-in EtherNet/IP Processing Error occurs, it is recorded in the event log and then corresponding bit turns ON. Refer to <i>Meanings of Error Status Bits</i> for the meanings of the error status bits.</p>				
Data type	WORD		Range of values	16#0000 to 16#00F0	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	You can access this variable from the user program with the following instruction. <ul style="list-style-type: none"> • GetEIPError 		

Variable name	_EIP_CipErr				
Meaning	CIP Communications Error		Global/local	Global	
Function	This is the error status variable for CIP communications. <p>Note If a Tag Name Resolution Error occurs, it is recorded in the event log and this variable changes to TRUE. Refer to <i>Meanings of Error Status Bits</i> for the meanings of the error status bits.</p>				
Data type	WORD		Range of values	16#0000 to 16#00F0	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	You can access this variable from the user program with the following instruction. <ul style="list-style-type: none"> • GetEIPError 		

Variable name	_EIP_TcpAppErr				
Meaning	TCP Application Communications Error		Global/local	Global	
Function	This is the error status variable for TCP application communications. It represents the collective status of the following error flags. <ul style="list-style-type: none"> • <i>_EIP_TcpAppCfgErr</i> (TCP Application Setting Error) • <i>_EIP_NTPSrvErr</i> (NTP Server Connection Error) <p>Note Refer to <i>Meanings of Error Status Bits</i> for the meanings of the error status bits.</p>				
Data type	WORD		Range of values	16#0000 to 16#00F0	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	You can access this variable from the user program with the following instruction. <ul style="list-style-type: none"> • GetEIPError 		

Variable name	_EIP_MacAdrErr				
Meaning	MAC Address Error		Global/local	Global	
Function	NY-series Controller: Indicates that an error occurred when the MAC address was read on the communications port 1 at startup. TRUE: Error FALSE: Normal				
Data type	BOOL		Range of values	TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

3 System-defined Variables Related to the Built-in EtherNet/IP Port

Variable name	_EIP1_MacAdrErr				
Meaning	Port1 MAC Address Error	Global/local		Global	
Function	Indicates that an error occurred when the MAC address was read on the communications port 1 at startup. TRUE: Error FALSE: Normal				
Data type	BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_LanHwErr				
Meaning	Communications Controller Error	Global/local		Global	
Function	NY-series Controller: Indicates that a communications controller failure occurred on the communications port 1. TRUE: Failure FALSE: Normal				
Data type	BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP1_LanHwErr				
Meaning	Port1 Communications Controller Error	Global/local		Global	
Function	Indicates that a communications controller failure occurred on the communications port 1. TRUE: Failure FALSE: Normal				
Data type	BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_EtnCfgErr				
Meaning	Basic Ethernet Setting Error	Global/local		Global	
Function	NY-series Controller: Indicates that the Ethernet communications speed setting (Speed/Duplex) for the communications port 1 is incorrect. Or, a read operation failed. TRUE: Setting incorrect or read failed FALSE: Normal				
Data type	BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP1_EtnCfgErr				
Meaning	Port1 Basic Ethernet Setting Error	Global/local		Global	
Function	Indicates that the Ethernet communications speed setting (Speed/Duplex) for the communications port 1 is incorrect. Or, a read operation failed. TRUE: Setting incorrect or read failed FALSE: Normal				
Data type	BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_IPAdrCfgErr				
Meaning	IP Address Setting Error		Global/local	Global	
Function	NY-series Controller: Indicates the IP address setting errors for the communications port 1. TRUE: <ul style="list-style-type: none"> • There is an illegal IP address setting. • A read operation failed. • The IP address obtained from the BOOTP server is inconsistent. FALSE: Normal				
Data type	BOOL		Range of values	TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP1_IPAdrCfgErr				
Meaning	Port1 IP Address Setting Error		Global/local	Global	
Function	Indicates the IP address setting errors for the communications port 1. TRUE: <ul style="list-style-type: none"> • There is an illegal IP address setting. • A read operation failed. • The IP address obtained from the BOOTP server is inconsistent. FALSE: Normal				
Data type	BOOL		Range of values	TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIPIn1_IPAdrCfgErr				
Meaning	Internal Port1 IP Address Setting Error		Global/local	Global	
Function	Indicates the IP address setting errors for the internal port 1. TRUE: <ul style="list-style-type: none"> • There is an illegal IP address setting. • A read operation failed. • The IP address obtained from the BOOTP server is inconsistent. FALSE: Normal				
Data type	BOOL		Range of values	TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_IPAdrDupErr				
Meaning	IP Address Duplication Error		Global/local	Global	
Function	NY-series Controller: Indicates that the same IP address is assigned to more than one node for the communications port 1. TRUE: Duplication occurred. FALSE: Other than the above.				
Data type	BOOL		Range of values	TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

3 System-defined Variables Related to the Built-in EtherNet/IP Port

Variable name	_EIP1_IPAdrDupErr				
Meaning	Port1 IP Address Duplication Error			Global/local	Global
Function	Indicates that the same IP address is assigned to more than one node for the communications port 1. TRUE: Duplication occurred. FALSE: Other than the above.				
Data type	BOOL			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIPIn1_IPAdrDupErr				
Meaning	Internal Port1 IP Address Duplication Error			Global/local	Global
Function	Indicates that the same IP address is assigned to more than one node for the internal port 1. TRUE: Duplication occurred. FALSE: Other than the above.				
Data type	BOOL			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_DNSCfgErr				
Meaning	DNS Setting Error			Global/local	Global
Function	Indicates that the DNS or hosts settings are incorrect. Or, a read operation failed. TRUE: Setting incorrect or read failed FALSE: Normal				
Data type	BOOL			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_BootpErr				
Meaning	BOOTP Server Error			Global/local	Global
Function	NY-series Controller: Indicates that a BOOTP server connection failure occurred on the communications port 1. TRUE: There was a failure to connect to the BOOTP server (timeout). FALSE: The BOOTP is not enabled, or BOOTP is enabled and an IP address was normally obtained from the BOOTP server.				
Data type	BOOL			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP1_BootpErr				
Meaning	Port1 BOOTP Server Error			Global/local	Global
Function	Indicates that a BOOTP server connection failure occurred on the communications port 1. TRUE: There was a failure to connect to the BOOTP server (timeout). FALSE: The BOOTP is not enabled, or BOOTP is enabled and an IP address was normally obtained from the BOOTP server.				
Data type	BOOL			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_IPRTblErr				
Meaning	IP Route Table Error			Global/local	Global
Function	NY-series Controller: Indicates that the default gateway settings or IP router table settings are incorrect. Or, a read operation failed. TRUE: Setting incorrect or read failed FALSE: Normal				

Data type	BOOL			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	<u>EIP_IdentityErr</u>				
Meaning	Identity Error			Global/local	Global
Function	NY-series Controller: Indicates that the identity information for CIP communications 1 (which you cannot overwrite) is incorrect. Or, a read operation failed. TRUE: Setting incorrect or read failed FALSE: Normal				

Data type	BOOL			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	<u>EIP_TDLinkCfgErr</u>				
Meaning	Tag Data Link Setting Error			Global/local	Global
Function	NY-series Controller: Indicates that the tag data link settings for CIP communications 1 are incorrect. Or, a read operation failed. TRUE: Setting incorrect or read failed FALSE: Normal				

Data type	BOOL			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	<u>EIP_TDLinkOpnErr</u>				
Meaning	Tag Data Link Connection Failed			Global/local	Global
Function	NY-series Controller: Indicates that establishing a tag data link connection for CIP communications 1 failed. TRUE: Establishing a tag data link connection failed due to one of the following causes. <ul style="list-style-type: none"> •The information registered for a target node in the tag data link parameters is different from the actual node information. •There was no response from the remote node. FALSE: Other than the above.				

Data type	BOOL			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	<u>EIP_TDLinkErr</u>				
Meaning	Tag Data Link Communications Error			Global/local	Global
Function	NY-series Controller: Indicates that a timeout occurred in a tag data link connection for CIP communications 1. TRUE: A timeout occurred. FALSE: Other than the above.				

Data type	BOOL			Range of values	TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

3 System-defined Variables Related to the Built-in EtherNet/IP Port

Variable name	_EIP_TagAdrErr				
Meaning	Tag Name Resolution Error		Global/local	Global	
Function	<p>NY-series Controller: Indicates that tag resolution for CIP communications 1 failed (i.e., the address could not be identified from the tag name).</p> <p>TRUE: Tag resolution failed (i.e., the address could not be identified from the tag name). The following causes are possible.</p> <ul style="list-style-type: none"> •The size of the network variable is different from the tag settings. •The I/O direction that is set in the tag data link settings does not agree with the I/O direction of the variable in the Controller. •There is no network variable in the Controller that corresponds to the tag setting. <p>FALSE: Other than the above.</p>				
Data type	BOOL		Range of values	TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_MultiSwONErr				
Meaning	Multiple Switches ON Error		Global/local	Global	
Function	<p>NY-series Controller: Indicates that more than one switch turned ON at the same time in CIP communications 1.</p> <p>TRUE: More than one data link start/stop switch changed to TRUE at the same time.</p> <p>FALSE: Other than the above.</p>				
Data type	BOOL		Range of values	TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_TcpAppCfgErr				
Meaning	TCP Application Setting Error		Global/local	Global	
Function	<p>TRUE: At least one of the set values for a TCP application (FTP, NTP, SNMP) is incorrect. Or, a read operation failed.</p> <p>FALSE: Normal.</p>				
Data type	BOOL		Range of values	TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_NTPSrvErr				
Meaning	NTP Server Connection Error		Global/local	Global	
Function	Always FALSE for an NY-series Controller.				
Data type	BOOL		Range of values	TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_DNSSrvErr				
Meaning	DNS Server Connection Error		Global/local	Global	
Function	<p>TRUE: The DNS client failed to connect to the server (timeout).</p> <p>FALSE: DNS is not enabled. Or, DNS is enabled and the connection was successful.</p>				
Data type	BOOL		Range of values	TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

● **Functional Classification: EtherNet/IP Communications Status**

Variable name	_EIP_EtnOnlineSta				
Meaning	Online	Global/local		Global	
Function	NY-series Controller: Indicates that the built-in EtherNet/IP port's communications can be used via the communications port 1 (that is, the link is ON, IP address is defined, and there are no errors). TRUE: The built-in EtherNet/IP port's communications can be used. FALSE: The built-in EtherNet/IP port's communications is disabled due to an error in initial processing, restart processing, or link OFF status.				
Data type	BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP1_EtnOnlineSta				
Meaning	Port1 Online	Global/local		Global	
Function	Indicates that the built-in EtherNet/IP port's communications can be used via the communications port 1 (that is, the link is ON, IP address is defined, and there are no errors). TRUE: The built-in EtherNet/IP port's communications can be used. FALSE: The built-in EtherNet/IP port's communications is disabled due to an error in initial processing, restart processing, or link OFF status.				
Data type	BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIPIn1_EtnOnlineSta				
Meaning	Internal Port1 Online	Global/local		Global	
Function	Indicates that the built-in EtherNet/IP port's communications can be used via the internal port 1 (that is, the link is ON, IP address is defined, and there are no errors.) TRUE: The built-in EtherNet/IP port's communications can be used. FALSE: The communications of the built-in EtherNet/IP port's internal port 1 is disabled due to an error in initial processing, restart processing, or link OFF status.				
Data type	BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_TDLinkRunSta				
Meaning	Tag Data Link Communications Status	Global/local		Global	
Function	NY-series Controller: Indicates that at least one connection is in normal operation in CIP communications 1. TRUE: Normal operation FALSE: Other than the above.				
Data type	BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_TDLinkAllRunSta				
Meaning	All Tag Data Link Communications Status	Global/local		Global	
Function	NY-series Controller: Indicates that all tag data links are communicating in CIP communications 1. TRUE: Tag data links are communicating in all connections as the originator. FALSE: An error occurred in at least one connection.				
Data type	BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

3 System-defined Variables Related to the Built-in EtherNet/IP Port

Variable name	<u>_EIP_RegTargetSta</u> [255]				
Meaning	Registered Target Node Information	Global/local		Global	
Function	<p>NY-series Controller: Gives a list of nodes for which built-in EtherNet/IP connections are registered for CIP communications 1.</p> <p>This variable is valid only when the built-in EtherNet/IP port is the originator.</p> <p><i>Array[x]</i> is TRUE: The connection to the node with a target node ID of x is registered.</p> <p><i>Array[x]</i> is FALSE: The connection to the node with a target node ID of x is not registered.</p>				
Data type	ARRAY [0..255] OF BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	<u>_EIP_EstbTargetSta</u> [255]				
Meaning	Normal Target Node Information	Global/local		Global	
Function	<p>NY-series Controller: Gives a list of nodes that have normally established EtherNet/IP connections for CIP communications 1.</p> <p><i>Array[x]</i> is TRUE: The connection to the node with a target node ID of x was established normally.</p> <p><i>Array[x]</i> is FALSE: The connection to the node with a target node ID of x was not established, or an error occurred.</p>				
Data type	ARRAY [0..255] OF BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	<u>_EIP_TargetPLCModeSta</u> [255]				
Meaning	Target PLC Operating Mode	Global/local		Global	
Function	<p>NY-series Controller: Shows the operating status of the target node Controllers that are connected for CIP communications 1, with the built-in EtherNet/IP port as the originator.</p> <p>The array elements are valid only when the corresponding Normal Target Node Information is TRUE. If the corresponding Normal Target Node Information is FALSE, the Target Node Controller Operating Information indicates the previous operating status.</p> <p><i>Array[x]</i> is TRUE: This is the operating state of the target Controller with a node address of x.</p> <p><i>Array[x]</i> is FALSE: Other than the above.</p>				
Data type	ARRAY [0..255] OF BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	<u>_EIP_TargetPLCErr</u> [255]				
Meaning	Target PLC Error Information	Global/local		Global	
Function	<p>NY-series Controller: Shows the error status (logical OR of fatal and non-fatal errors) of the target node Controllers that are connected for CIP communications 1, with the built-in EtherNet/IP ports as the originator. The array elements are valid only when the corresponding Normal Target Node Information is TRUE. The immediately preceding value is retained if this variable is FALSE.</p> <p><i>Array[x]</i> is TRUE: A fatal or non-fatal error occurred in the target Controller with a target node ID of x.</p> <p><i>Array[x]</i> is FALSE: Other than the above.</p>				
Data type	ARRAY [0..255] OF BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_TargetNodeErr [255]				
Meaning	Target Node Error Information	Global/local		Global	
Function	<p>NY-series Controller: Indicates that the connection for the Registered Target Node Information for CIP communications 1 was not established or that an error occurred in the target Controller.</p> <p>The array elements are valid only when the Registered Target Node Information is TRUE.</p> <p><i>Array[x]</i> is TRUE: A connection was not normally established with the target node for a target node ID of x (the Registered Target Node Information is TRUE and the Normal Target Node Information is FALSE), or a connection was established with the target node but an error occurred in the target Controller.</p> <p><i>Array[x]</i> is FALSE: The target node is not registered for a target node ID of x (the Registered Target Node Information is FALSE), or a connection was normally established with the target node (the Registered Target Node Information is TRUE and the Normal Target Node Information is TRUE). An error occurred in the target Controller (the Target PLC Error Information is TRUE).</p>				
Data type	ARRAY [0..255] OF BOOL	Range of values		TRUE or FALSE	
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_NTPResult		Member name	.ExecTime	
Meaning	NTP Last Operation Time		Global/local		Global
Function	NY-series Controller: No change from the initial value.				
Data type	Structure: _sNTP_RESULT Members: DATE_AND_TIME		Range of values		Depends on data type.
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Not possible.	Related instructions	You can read the contents of this variable with the GetNTPStatus instruction.		

Variable name	_EIP_NTPResult		Member name	.ExecNormal	
Meaning	NTP Operation Result		Global/local		Global
Function	NY-series Controller: No change from the initial value.				
Data type	BOOL		Range of values		TRUE or FALSE
R/W access	R	Retained	Not retained.	Network Publish	Published.
Usage in user program	Not possible.	Related instructions	You can read the contents of this variable with the GetNTPStatus instruction.		

● Functional Classification: EtherNet/IP Communications Switches

Variable name	_EIP_TDLinkStartCmd				
Meaning	Tag Data Link Communications Start Switch	Global/local		Global	
Function	<p>NY-series Controller: Change this variable to TRUE to start tag data links for CIP communications 1. It automatically changes back to FALSE after tag data link operation starts.</p> <p>Note Do not force this switch to change to FALSE from the user program or from the Sysmac Studio. It changes to FALSE automatically.</p>				
Data type	BOOL	Range of values		TRUE or FALSE	
R/W access	R/W	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

Variable name	_EIP_TDLinkStopCmd				
Meaning	Tag Data Link Communications Stop Switch	Global/local		Global	
Function	<p>NY-series Controller: Change this variable to TRUE to stop tag data links for CIP communications 1. It automatically changes back to FALSE after tag data link operation stops.</p> <p>Note Do not force this switch to change to FALSE from the user program or from the Sysmac Studio. It changes to FALSE automatically.</p>				
Data type	BOOL	Range of values		TRUE or FALSE	
R/W access	R/W	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

4

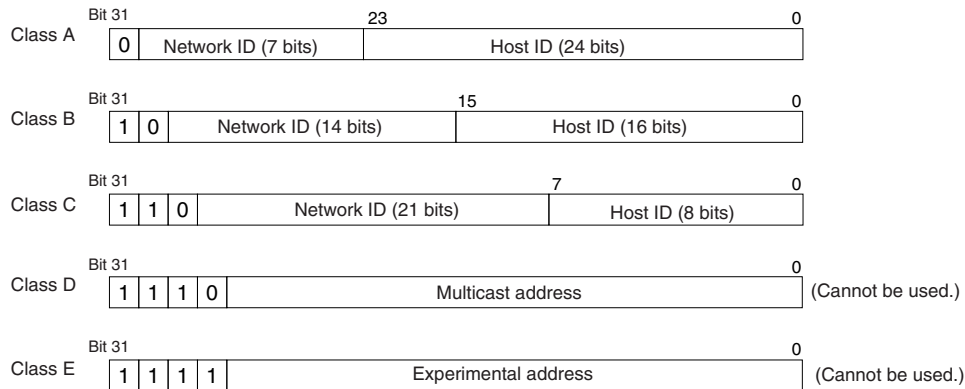
Determining IP Addresses

4-1	IP Addresses	4-2
4-1-1	IP Address Configuration	4-2
4-1-2	Allocating IP Addresses	4-3
4-1-3	Subnet Masks	4-3
4-1-4	CIDR	4-4
4-2	Controller IP Address Settings	4-5
4-2-1	Determining IP Addresses	4-5
4-2-2	Setting IP Addresses	4-6
4-2-3	Online Connection	4-8
4-2-4	Checking the Current IP Address	4-10
4-3	Private and Global Addresses	4-11
4-3-1	Private and Global Addresses	4-11
4-3-2	Using a Private Address for the Built-in EtherNet/IP Port	4-12
4-3-3	Using a Global Address for the Built-in EtherNet/IP Port	4-13

4-1 IP Addresses

4-1-1 IP Address Configuration

IP addresses are made up of 32 bits of binary data that specify the network number (net ID) and host number (host ID). The network number identifies the network, and the host number identifies the node (or host) on the network. IP addresses are divided into three classes, A, B, and C, so that the address system can be selected according to the scale of the network. (Classes D and E are not used.)



The number of networks in each class and the number of hosts possible on the network differ according to the class.

Class	Number of networks	Number of hosts
Class A	Small	$2^{24}-2$ max. (16,777,214 max.)
Class B	Medium	$2^{16}-2$ max. (65,534 max.)
Class C	Large	2^8-2 max. (254 max.)

The 32 bits of binary data in an IP address are divided into four sections of eight bits each. IP addresses are represented by the decimal equivalent of each of the four octets in the 32-bit address, each separated by a period.

For example, the binary address 10000010 00111010 00010001 00100000 would be represented as 130.58.17.32.

As shown in the above table, node 1 and node 2 have the same network address, which means these nodes belong to the same network.

4-1-4 CIDR

CIDR, or classless interdomain routing, is used to assign IP addresses that do not use classes. IP addresses that use classes are separated into blocks according to network IDs and host IDs, resulting in inefficient usage of IP address space. CIDR does not use classes, so IP address space can be divided as required to more efficiently use IP address space. For example, using a subnet mask setting with CIDR enables building a horizontally distributed network exceeding 254 nodes even if a class C address block (e.g., 192, 168...) is used.

Subnet Mask Range
192.0.0.0 to 255.255.255.252

4-2 Controller IP Address Settings

4-2-1 Determining IP Addresses

Use one of the following methods to set the IP address for the built-in EtherNet/IP port on the NY-series Controller. The IP address for the internal port can be set only by using a user-specified address.

Setting a User-specified IP Address

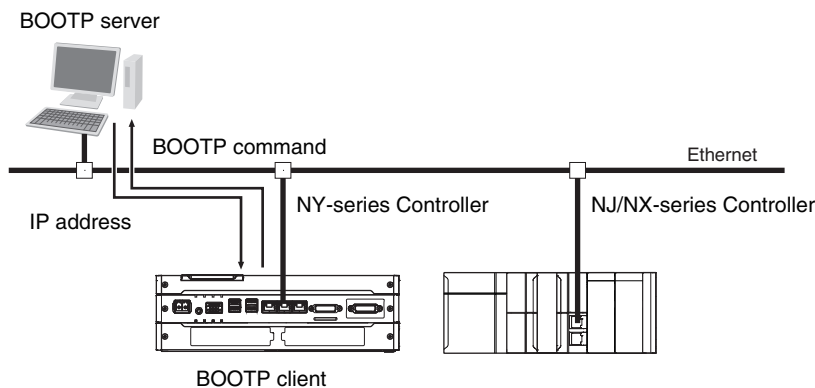
If you need to change the default IP address of the built-in EtherNet/IP port or if you need to use the built-in EtherNet/IP port with another EtherNet/IP node, set the IP address to the required value.

For an NX-series CPU Unit, you cannot set IP addresses that make two built-in EtherNet/IP ports belong to the same network.

Automatically Obtaining the IP Address from the BOOTP Server

There are two methods to automatically obtain an IP address.

- Obtain the IP address from the BOOTP server each time the power is turned ON.
- Obtain the IP address from the BOOTP server once when the power is turned ON and then do not allow it to change.



4-2-2 Setting IP Addresses

Use the Sysmac Studio to set the IP addresses of the built-in EtherNet/IP port and the internal port.



Additional Information

You can also set these IP addresses by using the Industrial PC Support Utility, instead of the Sysmac Studio.

Refer to the *NY-series Industrial Panel PC / Industrial Box PC Setup User's Manual* (Cat. No. W568) for information on setting the IP addresses with the Industrial PC Support Utility.

1 Select the setting method for IP addresses.

Make the following settings on the TCP/IP Settings Display of the Built-in EtherNet/IP Port Settings Tab Page in the Controller Setup to set the local IP address.

Used to set a user-specified IP address.

Used to obtain the IP address from the BOOTP server each time the power is turned ON.

Used to obtain the IP address from the BOOTP server once and then not change it.



Precautions for Correct Use

You cannot set IP addresses that make the built-in EtherNet/IP port and the internal port belong to the same network.

- 2 Connect the Sysmac Studio to the NY-series Controller via a USB connection or the Ethernet network.
- 3 Connect the Sysmac Studio online to the NY-series Controller. Refer to *4-2-3 Online Connection* for the procedure to connect online.
- 4 Use one of the following methods to download the IP addresses that were set on the Sysmac Studio to the NY-series Controller.
 - Go online with the Controller, and then select **Synchronization** from the Controller Menu. The data on the computer and the data in the physical Controller are compared automatically.
 - Click the **Transfer to Controller** Button.

Note Use the Synchronization Menu of the Sysmac Studio to upload and download data.

- 5** After the IP address settings are downloaded, the IP address is reflected in the NY-series Controller as follows:

Setting a User-specified IP Address

After the IP address settings are downloaded, the set IP address is automatically saved in the Controller.

Obtaining the IP Address from the BOOTP Server Each Time the Power Is Turned ON

After the data is downloaded, the IP address from the BOOTP server is automatically saved in the Controller. Each time the power supply is turned ON, the IP address from the BOOTP server is automatically saved in the Controller.

Obtaining the IP Address from the BOOTP Server Once When the Power Is Turned ON and Then Not Allow It to Change

After the I/O address is downloaded, the IP address from the BOOTP server is automatically saved in the Controller and then the same address is used.



Additional Information

- The TCP/IP Settings Display is not updated even if the IP address is obtained normally from the BOOTP server. To check the IP address that was obtained from the BOOTP server on the TCP/IP Display, upload the project from the NY-series Controller.
- If you cannot obtain the IP address from the BOOTP server, the *Fix at the IP address obtained from BOOTP server* Option is selected on the TCP/IP Display. Select *Fixed Setting* in the IP Address Settings and manually set the IP address, subnet mask, and default gateway. The Controller will continue the request to obtain the IP address even if it fails to connect to the BOOTP server.
- If the Controller power supply is turned ON when the IP address was not normally obtained from the BOOTP server, the setting remains at *Fix at the IP address obtained from BOOTP server*.
- After the IP address is obtained from the BOOTP server, the built-in EtherNet/IP port IP address setting is automatically set to *Fixed setting*. Therefore, the IP address will not match when the program is verified on the Sysmac Studio.

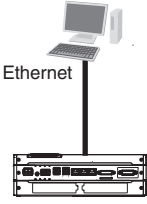
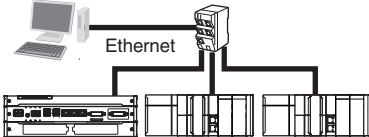
4-2-3 Online Connection

Connect the Sysmac Studio online to the Controller.

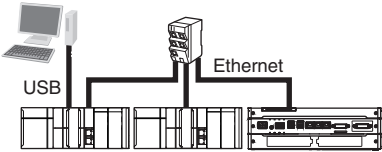
Types of Connections between the Controller and Computer That Runs the Sysmac Studio

The Controller and the computer that runs Sysmac Studio are connected as shown below via USB or Ethernet:

Ethernet Connection

Direct Connection via Ethernet (1:1 Connection with AutoIP)	Ethernet Connection via Hub (1: N Connection)
 <p>The diagram shows a computer on the left connected to a controller on the right by a single vertical line labeled 'Ethernet'.</p> <p>Note 1 An Ethernet switch is not necessarily required. 2 You can use a straight or cross Ethernet cable to connect.</p>	 <p>The diagram shows a computer on the left connected to a central hub. From the hub, multiple lines labeled 'Ethernet' connect to a row of three controllers on the right.</p> <p>Note An Ethernet switch is required to connect. Refer to 2-1-4 Precautions for Ethernet Switch Selection for details.</p>

Connection from USB Across Ethernet

Remote Connection via USB (1:N Connection)
 <p>The diagram shows a computer on the left connected to a controller on the right via a USB cable. An Ethernet hub is positioned between them, connected to both the computer and the controller. The connection from the computer to the hub is labeled 'USB', and the connection from the hub to the controller is labeled 'Ethernet'.</p> <p>Note 1 An NJ/NX-series Controller with a USB port is required to connect. 2 An Ethernet switch is required to connect. Refer to 2-1-4 Precautions for Ethernet Switch Selection for details.</p>



Additional Information

- Auto IP automatically assigns IP addresses in Windows 98 and later operating systems. Unique IP addresses are automatically assigned from the address 169.254.0.0 to 169.254.255.255.
- If the IP address of the connected EtherNet/IP port is changed when the Sysmac Studio is connected online via a built-in EtherNet/IP port, a timeout will occur in the Sysmac Studio. In the case, switch the Sysmac Studio status to offline, change to the IP address of the connected built-in EtherNet/IP port, and then switch back the Sysmac Studio status to online. This will allow you to reconnect.



Precautions for Correct Use

If there is more than one node with the same IP address in the EtherNet/IP network, the built-in EtherNet/IP port will connect to the node that is detected first. An IP Address Duplication Error will not occur.

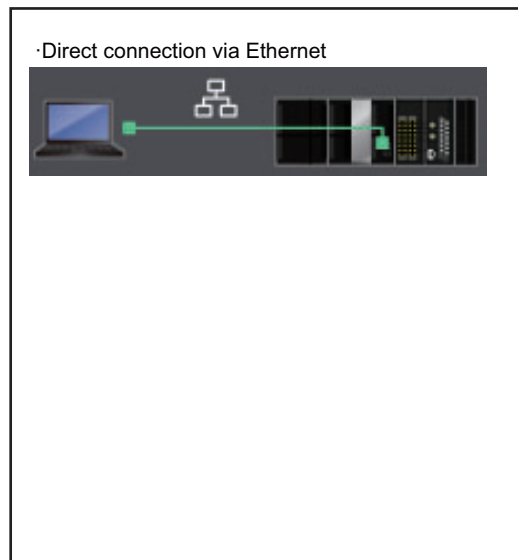
Online Connection Procedure

Connect the Controller and the computer that runs the Sysmac Studio via Ethernet, and then perform the following procedure.

- 1 Select **Controller - Communications Setup** and click the **OK** Button in the Sysmac Studio Project Window.

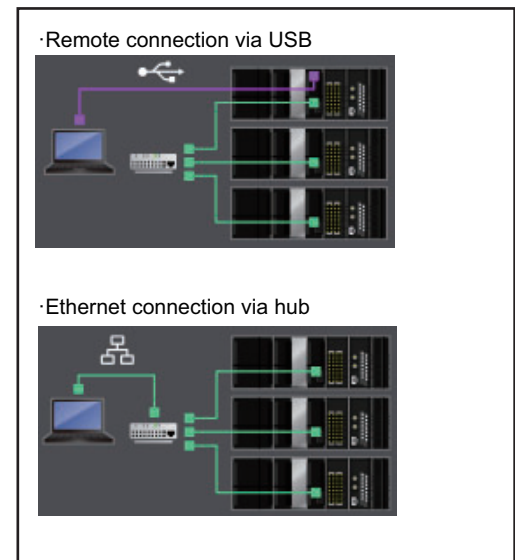
1:1 Connection

Direct Connection



1:N Connection

Ethernet Connection



Additional Information

If there is an error in the set IP address, an IP Address Setting Error is recorded in the event log.



Precautions for Correct Use

- If the IP address is duplicated or not set correctly, communications are not possible via the EtherNet/IP network.
- The IP address range shown below is used by the system and cannot be specified.
169.254.0.0 to 169.254.255.255
192.168.255.0 to 192.168.255.255
- Due to Ethernet restrictions, you cannot specify the IP addresses that are described below.
 - An IP address that is all 0's or all 1's
 - IP addresses that start with 127, 0, or 255 (decimal)
 - IP addresses that have a host ID that is all 0's or all 1's
 - Class-D IP addresses (224.0.0.0 to 239.255.255.255)
 - Class-E IP addresses (240.0.0.0 to 255.255.255.255)

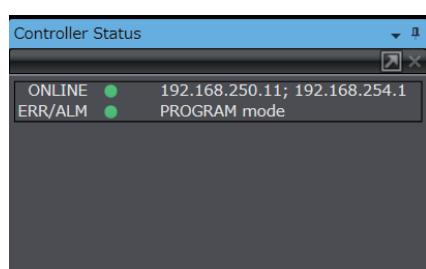
Connecting from a Saved Project

The connection configuration that is set (EtherNet/IP) is saved in the project. (The file is xxx.smc.) If you open a saved project on the Sysmac Studio, you can connect to the EtherNet/IP network without redoing the settings.

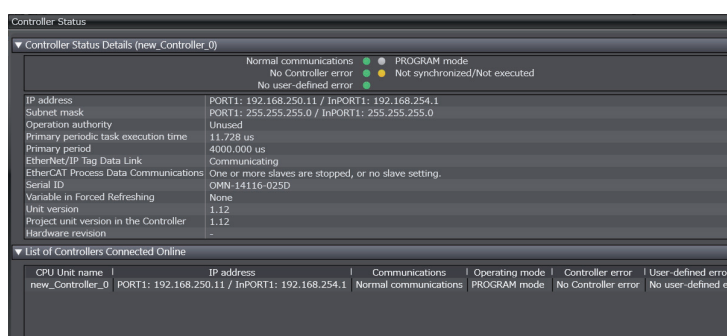
4-2-4 Checking the Current IP Address

The current IP address can be confirmed in the Controller Status Pane of the Sysmac Studio either when it is set manually or obtained from the BOOTP server.

- Basic Controller Status Pane



- Controller Status Pane with Details



Additional Information

- If you obtain the IP address from the BOOTP server, you can check the obtained IP address by synchronizing and uploading the controller settings from the Sysmac Studio.
- If the IP address of the built-in EtherNet/IP port is not registered due to the following reasons, the IP address field shows “0.0.0.0”.
 - The IP address was not obtained from the BOOTP server.

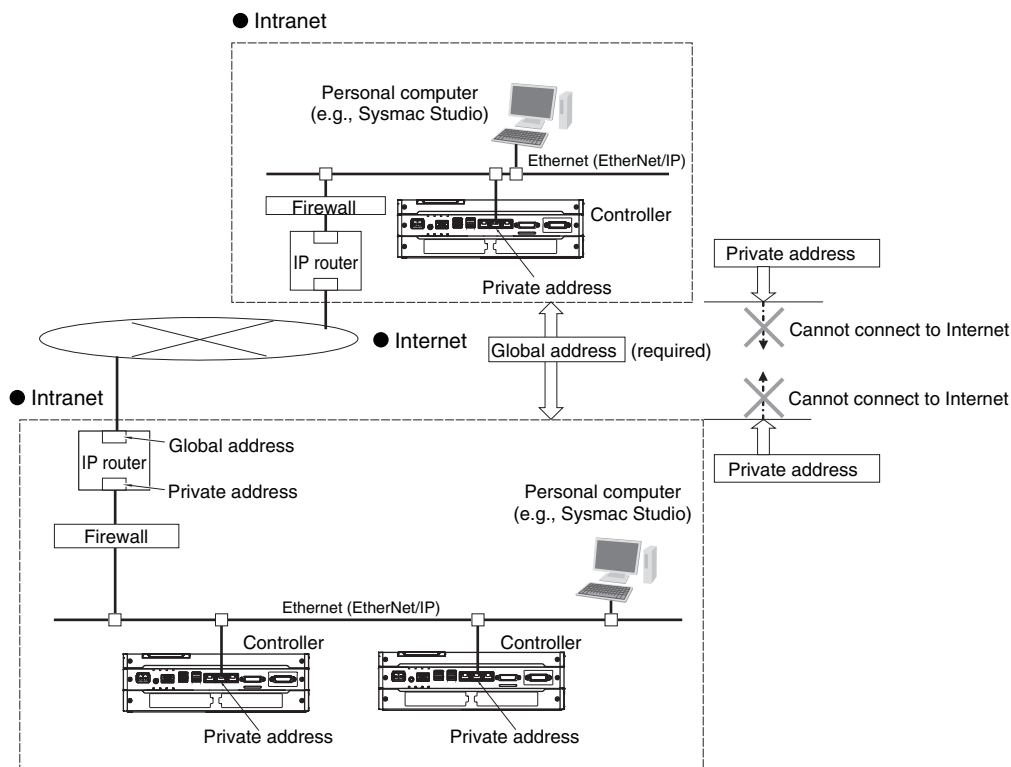
4-3 Private and Global Addresses

4-3-1 Private and Global Addresses

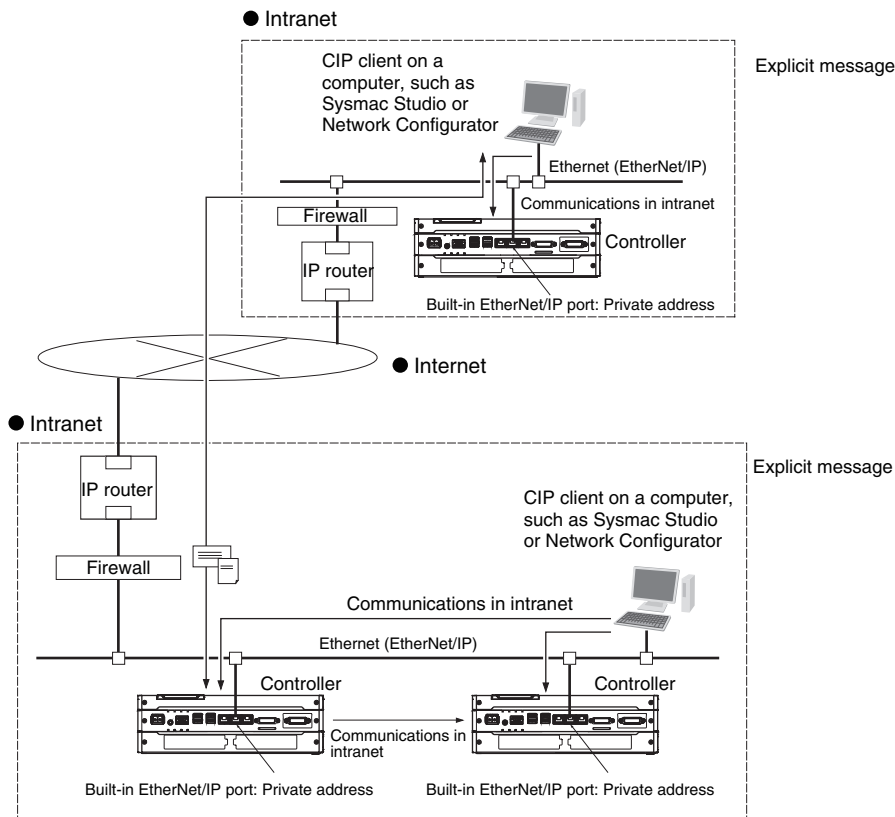
There are two kinds of IP addresses, private and global.

IP address	Description
Global address	These are IP addresses that connect directly to the Internet. Allocated by application to NIC, each address is unique in the world, and as many as 4.3 billion can be allocated world-wide.
Private address	These are IP addresses for Intranet (LAN) use. Direct connection to the Internet is not possible. Frames that include private IP addresses are restricted by the router from being sent outside the LAN.

Generally, as shown below, global addresses in the intranet are allocated only to IP routers (such as broadband routers) interfaced with the Internet. All other nodes in the intranet, which includes the built-in EtherNet/IP port, are allocated private addresses.



4-3-2 Using a Private Address for the Built-in EtherNet/IP Port



● Conditions for Communications Applications

If the built-in EtherNet/IP port uses a private address, you can use explicit message communications service under the following conditions.

- The explicit message communications service can be executed on the intranet between built-in EtherNet/IP ports with private addresses only.
- A device such as a personal computer (CIP applications including the Network Configurator) cannot connect online and communicate over the Internet with a built-in EtherNet/IP port that has a private address. Explicit message communications are also not possible over the Internet between built-in EtherNet/IP ports with private addresses.

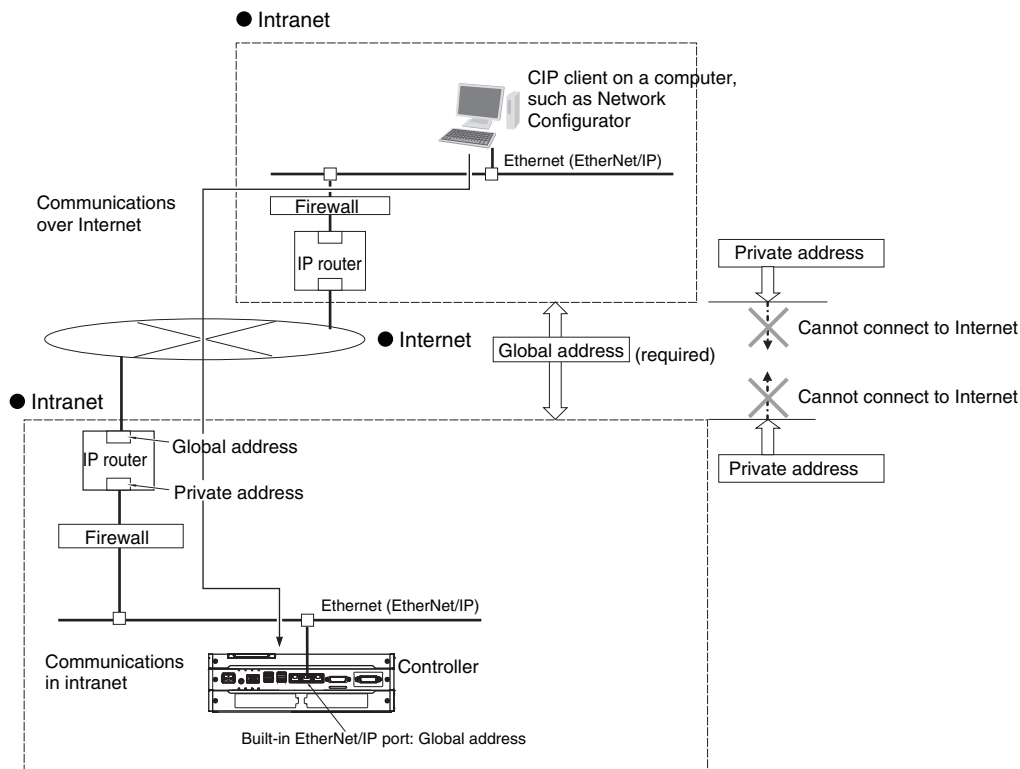


Precautions for Correct Use

Network Security and Firewalls

To set up an intranet through a global address involves network security considerations. Be sure to consult with a network specialist in advance and consider installation of a firewall. After a firewall has been set up by a communications company technician, there may be some applications that cannot be used. Be sure to check first with the communications company technician.

4-3-3 Using a Global Address for the Built-in EtherNet/IP Port



● Conditions for Communications Applications

You can use the explicit message communications service over the Internet under the following conditions.

- A device such as a personal computer (a CIP application including the Network Configurator) can connect online and communicate over the Internet with a built-in EtherNet/IP port that has a global address.
- The TCP port number (44818) or UDP port number (44818) that is used for EtherNet/IP cannot be used because it is prohibited by a firewall in the communications path.



Precautions for Correct Use

Network Security and Firewalls

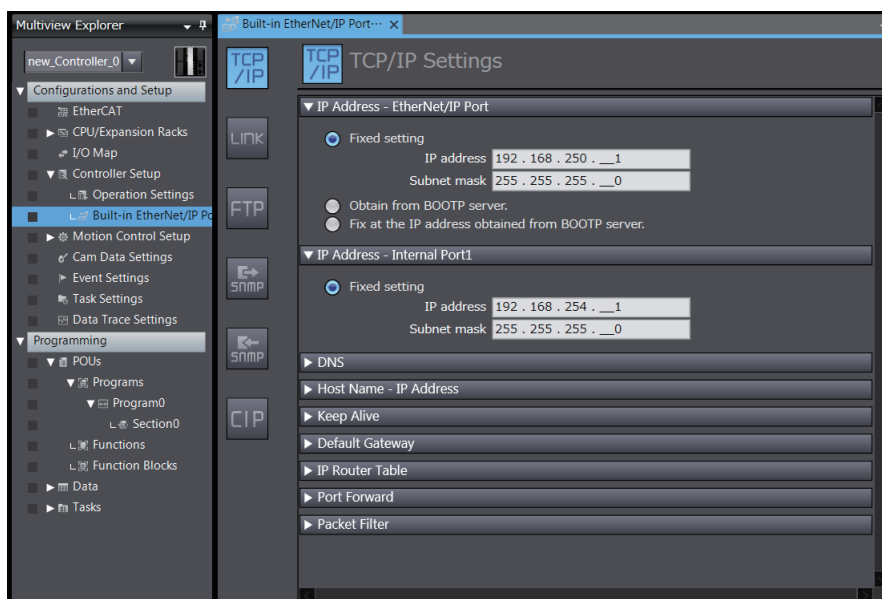
To set a global IP address for a built-in EtherNet/IP port involves network security considerations. It is recommended that the user contract with a communications company for a dedicated line, rather than for a general line such as a broadband line. Also, be sure to consult with a network specialist and consider security measures such as a firewall. After a firewall has been set up by a communications technician, there may be some applications that cannot be used. Be sure to check first with the communications technician.

5

Sysmac Studio Settings for the Built-in EtherNet/IP Port

5-1	TCP/IP Settings Display	5-2
5-2	Link Settings Display	5-6
5-3	FTP Display	5-7
5-4	SNMP Settings Display	5-8
5-5	SNMP Trap Settings Display	5-10
5-6	CIP Settings Display	5-12

5-1 TCP/IP Settings Display



● IP Address - EtherNet/IP Port

Setting	Description	Default
IP address setting method	Select one of the following IP address setting methods for the built-in EtherNet/IP port. <ul style="list-style-type: none"> Fixed setting Obtain from BOOTP server. Fix at the IP address obtained from BOOTP server. 	Fixed setting
IP address* ²	Set the IP address for the built-in EtherNet/IP port.* ³	Built-in EtherNet/IP port 1: 192.168.250.1 Built-in EtherNet/IP port 2: 192.168.251.1
Subnet mask* ²	Set the subnet mask for the built-in EtherNet/IP port.	255.255.255.0

*1 These settings are required if you set IP address setting method to *Fixed setting*.

*2 Refer to 4-2 *Controller IP Address Settings* for details on setting IP addresses.

● IP Address - Internal Port

Setting	Description	Default
IP address	Set the IP address for the internal port.* ¹	192.168.254.1
Subnet mask* ¹	Set the subnet mask for the internal port.	255.255.255.0

*1 Refer to 2-1-4 *Precautions for Ethernet Switch Selection* for details on setting IP addresses.



Precautions for Correct Use

You cannot set IP addresses that make the built-in EtherNet/IP port and the internal port belong to the same network.

● DNS

Setting	Description	Default
DNS	Enable using DNS if a DNS is used to resolve host names when host names are specified for the remote communications nodes in CIP communications and socket instructions. A DNS server is required to use DNS.	Do not use.
Priority DNS server*1	Set the IP address of the DNS server. You can set priority and secondary IP addresses.	None
Secondary DNS server		None
Domain name*1	Sets the domain name of the domain to which the built-in EtherNet/IP port belongs. (Single-byte alphanumeric characters, dots, and hyphens: 48 characters max.)	None

*1 These settings are required when DNS is used.

● Host Name - IP Address

Setting	Description	Default
Host Name	Addresses are converted according to this setting when host names are used to specify remote communications nodes. They can be set even if DNS is not used. You can set up to six host names. (You can use up to 200 single-byte alphanumeric characters, dots, and hyphens with up to 63 single-byte alphanumeric characters between dots.)	None
IP Address	Set the IP addresses of the registered hosts.	None

● Keep Alive

Setting	Description	Default
Keep Alive	Set whether to use the remote node keep alive function of connected servers and clients (such as socket services, FTP server, Sysmac Studio, and TCP). If the keep-alive function is used and the remote node does not respond during the set keep-alive monitoring time, the connection is disconnected. The connection is left open if the power supply to the remote node is turned OFF without warning. Use the keep-alive function whenever possible. <ul style="list-style-type: none"> • Use. • Do not use. 	Use.
Keep Alive Monitoring Time	The connection is disconnected if the keep-alive function is used and the remote node does not respond during the monitoring time set in the <i>Keep Alive Monitoring Setting</i> . Setting range: 1 to 65,535 (seconds)	300
Linger option	Set whether to specify the <i>Linger Option</i> for connections to TCP or socket services. If the <i>Linger Option</i> is specified, the port number is immediately opened even before the port number is released after the socket closes (approx. 1 minute). <ul style="list-style-type: none"> • Specify. • Do not specify. 	Do not specify.

● **Default Gateway**

Setting	Description	Default
Default gateway	Set the IP address of the default gateway for the built-in EtherNet/IP port. This setting is not required when the default gateway is not used.*1	None

*1 Even if you select *Fixed setting* for both EtherNet/IP port and internal port, you can only set the default gateway for one of the ports.

● **IP Router Table**

Parameter	Description	Default
Destination IP address	Set these settings when the built-in EtherNet/IP port is used for tag data links or CIP message communications with nodes on other IP network segments via an IP router. Set them also to use the IP routing function of the built-in EtherNet/IP port as an IP router. You can set up to 128 combinations of IP addresses and gateway addresses. Specify 0 for the host portions of the IP addresses.	None
Destination mask IP address		
Gateway address		None

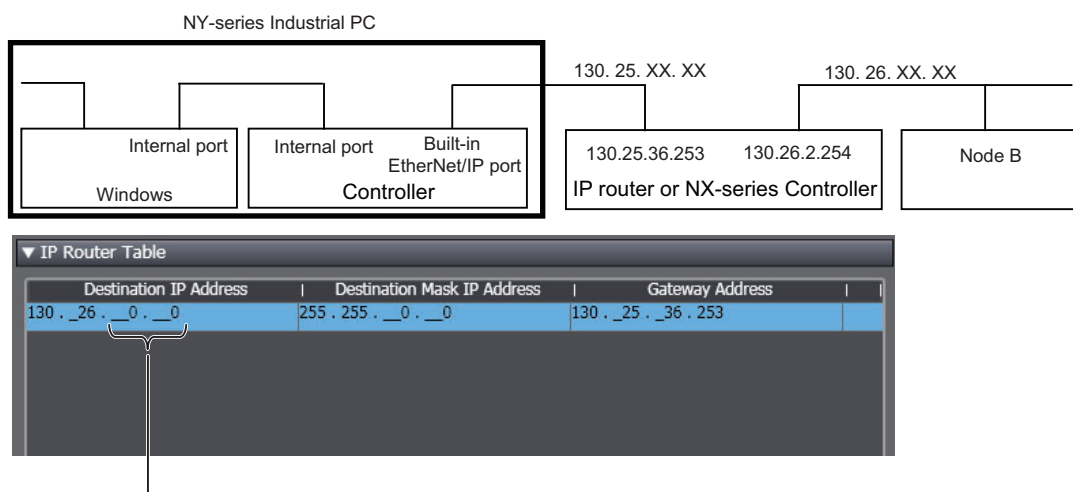


Additional Information

IP Router Table Setting Example

To enable the Controller in an NY-series Industrial PC to perform Ethernet communications with node B to send a CIP message etc. through an IP router, set the IP router table of the Controller as follows.

When you set the IP router table, Controller sends packets to the gateway IP address (130.25.36.253) if communications instructions are executed on Controller and addressed to node B.



The host fields are set to 0 in the destination IP address.

On the other hand, for Windows in an NY-series Industrial PC to perform Ethernet communications with node B to send a CIP message etc. through the Controller in an NY-series Industrial PC and an IP router, you need to set the default gateway and other settings for IP routing in Windows, in addition to the IP router table of the Controller.

● Port Forward

Setting	Description	Default
IP Forward	Specify whether to relay IP packets between the built-in EtherNet/IP port and the internal port on the Controller.	Use
NAT	Specify whether to convert the source IP address to the IP address for the built-in EtherNet/IP port if packets from Windows are relayed via the internal port to a network on the built-in EtherNet/IP port side.	Use

● Packet Filter

Setting	Description	Default
Packet Filter	Specify whether to set conditions for the IP packets received by the internal port.	Use ^{*1}
Pass Frame	Specify the conditions for the IP packets permitted to be received by the internal port by setting the items shown below. You can set a maximum of 32 conditions for the IP packets permitted to be received. This setting is valid only if you set <i>Packet Filter</i> to <i>Use</i> .	---
Specification method	Select from the following the specification method for the IP packets permitted to be received. <ul style="list-style-type: none"> IP address specification^{*2} any Interface network^{*3} 	Interface network
IP address ^{*4}	Set the IP address for which you permit reception.	None
Subnet mask ^{*4}	Set the subnet mask for which you permit reception.	None
Protocol	Select <i>any</i> , <i>tcp</i> , <i>udp</i> , <i>igmp</i> , or <i>icmp</i> .	any
Range specification	If you set <i>Protocol</i> to <i>tcp</i> or <i>udp</i> , specify whether to set a range of port numbers for which you permit reception. <ul style="list-style-type: none"> Selected Range specification is enabled. Specify a range by setting the items <i>Port A</i> and <i>Port B</i>. No check Range specification is disabled. Set an individual port number in the item <i>Port A</i>. 	No check
Port A	Set the starting IP address of the range for which you permit reception.	None
Port B	Set the ending IP address of the range for which you permit reception. You need to set this only if you selected <i>Range specification</i> .	None

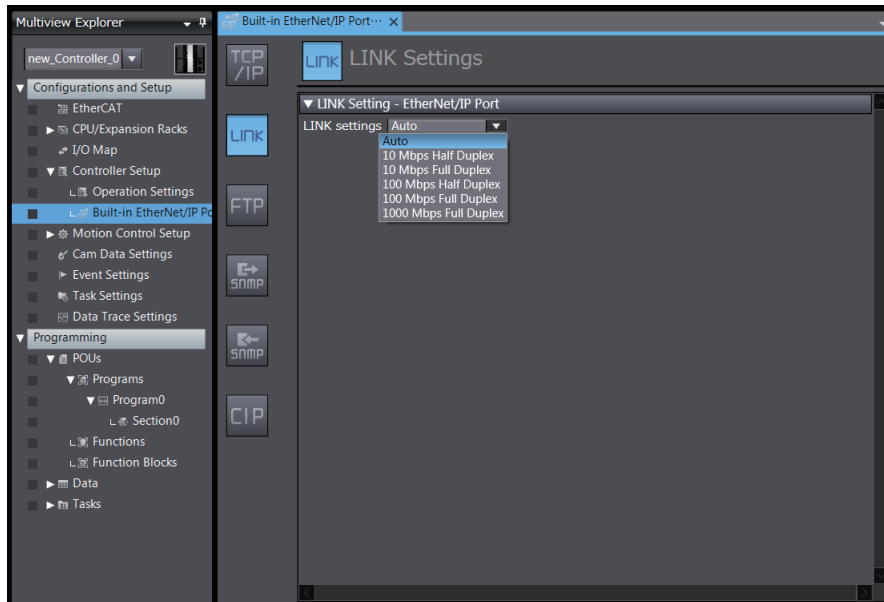
*1 By default, only the reception of IP packets transmitted from a network to which the internal port belongs (on the Windows side) is permitted.

*2 Specify the IP address by setting the items IP address and Subnet mask.

*3 This indicates a network to which the IP address for the internal port belongs. Unlike IP address specification, you do not need to change this setting even if you change the IP address for the internal port after setting this.

*4 You need to set this item only if you set Specification method to IP address specification.

5-2 Link Settings Display

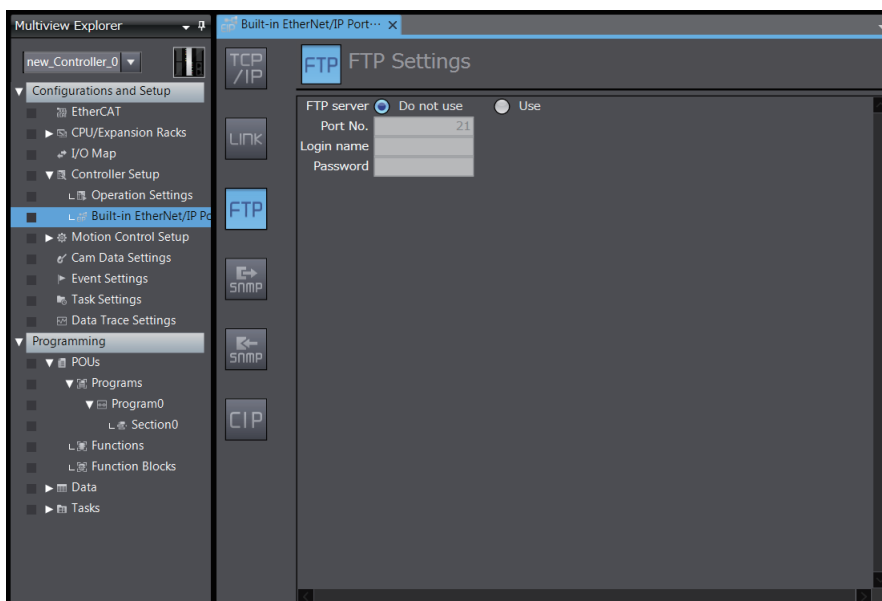


● LINK settings - EtherNet/IP Port

Set for each built-in EtherNet/IP port.

Setting	Description	Default
LINK settings	Set the baud rate for the built-in EtherNet/IP port. <ul style="list-style-type: none"> • Auto • 10 Mbps Half Duplex • 10 Mbps Full Duplex • 100 Mbps Half Duplex • 100 Mbps Full Duplex • 1000 Mbps Full Duplex 	Auto

5-3 FTP Display



Setting	Description	Default
FTP server	Specify whether to use the FTP server. FTP connections from external devices will not be possible if the <i>Do not use</i> Option is selected.	Do not use.
Port No.*1*3	Set the FTP port number of the built-in EtherNet/IP port. It is normally not necessary to change this setting. The FTP control port is set here. The FTP data transfer port is always port 20.	21
Login name*1	Set the login name to externally connect to the built-in EtherNet/IP port via FTP. (You can use up to 12 alphanumeric characters.)*2	None
Password*1	Set the password to externally connect to the built-in EtherNet/IP port via FTP. (You can use 8 to 32 alphanumeric characters.)*2	None

*1 These settings are required to use the FTP server.

*2 The login name and password are case sensitive.

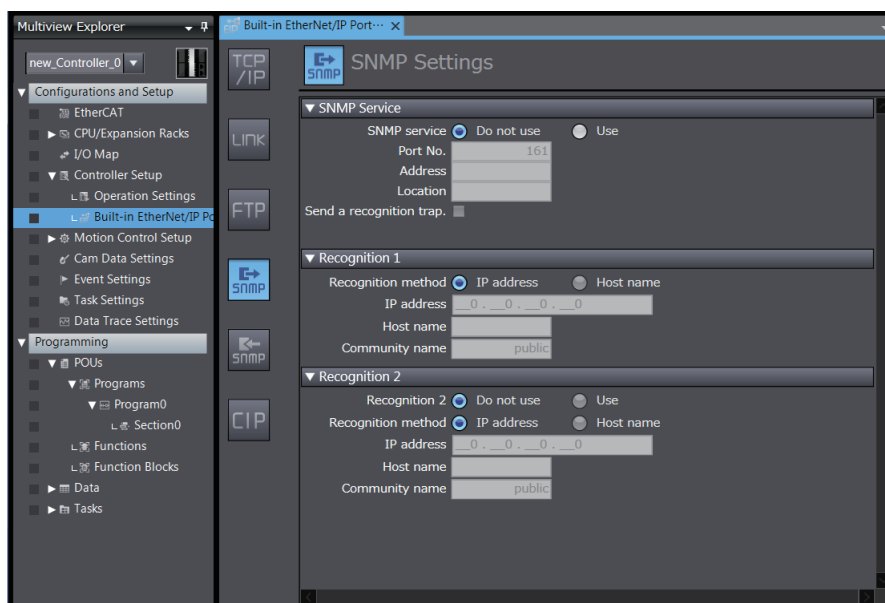
*3 The following ports are used by the system and cannot be set by the user: 20, 23, 25, 80, 110, 9610, 9900, and 44818.



Additional Information

Refer to *Section 10 FTP Server* for details on the FTP server.

5-4 SNMP Settings Display



● SNMP Service

Setting	Description	Default
SNMP service	Specify whether to use the SNMP monitor service.*1 If not using the SNMP monitor service is specified, an SNMP manager cannot connect from an external device.	Do not use.
Port No.*2	Set the port number to use to connect to the SMTP server that is used to connect from an SNMP manager. This setting does not normally need to be changed.	161
Address	Set the communications device administrator name and installation location as text information. You do not necessarily have to input all items. This information is read by the SNMP manager. (You can input up to 255 single-byte alphanumeric characters for each item.)	None
Location		None
Send a recognition trap	Set whether to send an authentication trap. If you select <i>Send a recognition trap</i> and there is access from an SNMP manager that is not set in Recognition 1 or Recognition 2, an authentication trap is sent to the SNMP manager. If you select <i>Send a recognition trap</i> , specify the SNMP trap settings on the SNMP Trap Tab Page.	Do not use.

*1 If you specify to use the SNMP trap, you also have to set the recognition 1 and 2 as described below.

*2 The following ports are used by the system and cannot be set by the user: 25, 53, 68, 110, 2222, 2223, 2224, 9600, and 44818.



Additional Information

Refer to *Section 12 SNMP Agent* for details on the SNMP service.

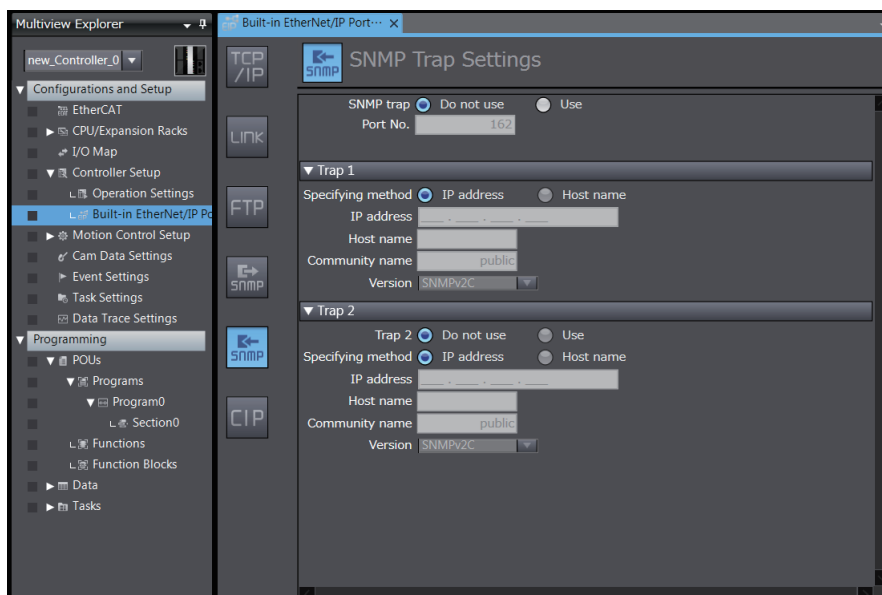
● Recognition 1

Setting	Description	Default
Recognition method	Set the method to use to specify SNMP managers for which access is permitted. <ul style="list-style-type: none"> • IP address • Host name Make these settings to permit access by only certain SNMP managers. Access is not allowed unless an IP address or host name is set.	IP address
IP address	Set the IP address of the SNMP manager. If the default setting of 0.0.0.0 is used, access is permitted from all SNMP managers. (Set this setting if the recognition method in the recognition 1 settings is set to the <i>IP address</i> Option.)	None
Host name	Set the host name of the SNMP manager. (Set this setting if the recognition method in the recognition 1 settings is set to the <i>Host name</i> Option.) (You can use up to 200 single-byte alphanumeric characters, dots, and hyphens with up to 63 single-byte alphanumeric characters between dots.)	None
Community name	Set the community name to enable the SNMP manager to access information from the built-in EtherNet/IP port. (Single-byte alphanumeric characters, dots, and hyphens: 255 characters max.)	public

● Recognition 2

Setting	Description	Default
Recognition 2	Specify whether to use the recognition 2 settings. <ul style="list-style-type: none"> • Use. • Do not use. 	Do not use.
Recognition method	Set the method to use to specify SNMP managers for which access is permitted. <ul style="list-style-type: none"> • IP address • Host name Make these settings to permit access by only certain SNMP managers. Access is not allowed unless an IP address or host name is set.	IP address
IP address	Set the IP address of the SNMP manager. If the default setting of 0.0.0.0 is used, access is permitted from all SNMP managers. (Set this setting if the recognition method in the recognition 2 settings is set to the <i>IP address</i> Option.)	None
Host name	Set the host name of the SNMP manager. (Set this setting if the recognition method in the recognition 2 settings is set to the <i>Host name</i> Option.) (You can use up to 200 single-byte alphanumeric characters, dots, and hyphens with up to 63 single-byte alphanumeric characters between dots.)	None
Community name	Set the community name to enable the SNMP manager to access information from the built-in EtherNet/IP port. (Single-byte alphanumeric characters, dots, and hyphens: 255 characters max.)	public

5-5 SNMP Trap Settings Display



● SNMP Trap

Setting	Description	Default
SNMP trap	Specify whether to use the SNMP trap (network error detection). ^{*1} If the SNMP trap service is not used, SNMP traps are not sent to the SNMP manager.	Do not use.
Port No. ^{*2}	Set the port number to use to connect to the SNMP server. It is normally not necessary to change this setting.	162

*1 If you specify to use the SNMP trap, you also have to set the trap 1 and 2 as describe below.

*2 The following ports are used by the system and cannot be set by the user: 25, 53, 68, 110, 2222, 2223, 2224, 9600, and 44818.



Additional Information

Refer to *12-1-1 Overview* for details on the SNMP trap.

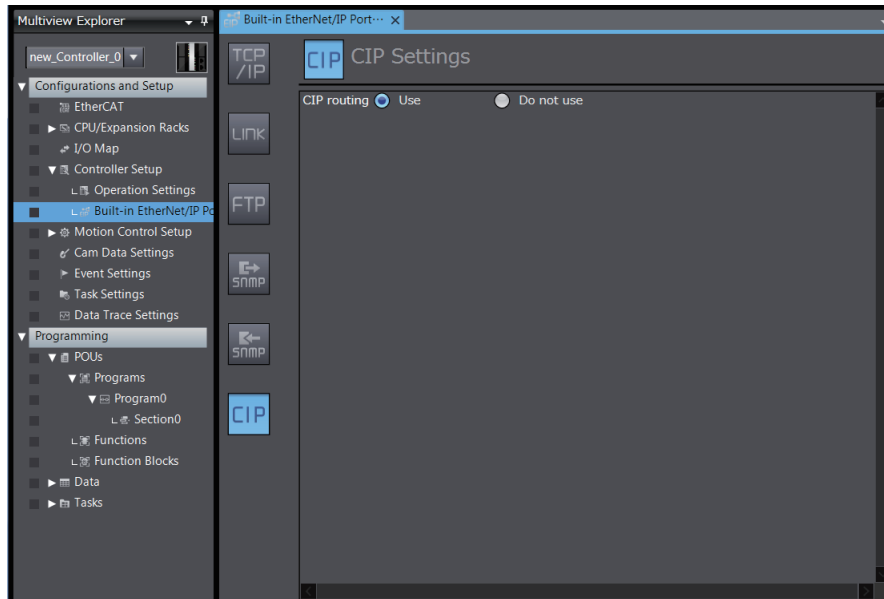
● Trap 1

Setting	Description	Default
Specifying method	Set the specifying method for the SNMP manager destination for SNMP traps. <ul style="list-style-type: none"> • IP address • Host name 	IP address
IP address	Set the IP address of the SNMP manager. (Set this setting if the specifying method in the trap 1 settings is set to the <i>IP address</i> Option.)	None
Host name	Set the host name of the SNMP manager. (Set this setting if the specifying method in the trap 1 settings is set to the <i>Host name</i> Option.) (You can use up to 200 single-byte alphanumeric characters, dots, and hyphens with up to 63 single-byte alphanumeric characters between dots.)	None
Community name	Set the community name. (You can use up to 255 single-byte alphanumeric characters.)	public
Version	Set the version of the SNMP manager. <ul style="list-style-type: none"> • SNMP version 1: SNMPv1 • SNMP version 2C: SNMPv2C 	SNMPv1

● Trap 2

Setting	Description	Default
Trap 2	Specify whether to use the trap 2 settings. <ul style="list-style-type: none"> • Use. • Do not use. 	Do not use.
Specifying method	Set the specifying method for the SNMP manager destination for SNMP traps. <ul style="list-style-type: none"> • IP address • Host name 	IP address
IP address	Set the IP address of the SNMP manager. (Set this setting if the specifying method in the trap 2 settings is set to the <i>IP address</i> Option.)	None
Host name	Set the host name of the SNMP manager. (Set this setting if the specifying method in the trap 2 settings is set to the <i>Host name</i> Option.) (You can use up to 200 single-byte alphanumeric characters, dots, and hyphens with up to 63 single-byte alphanumeric characters between dots.)	None
Community name	Set the community name. (You can use up to 255 single-byte alphanumeric characters.)	public
Version	Set the version of the SNMP manager. <ul style="list-style-type: none"> • SNMP version 1: SNMPv1 • SNMP version 2C: SNMPv2C 	SNMPv1

5-6 CIP Settings Display



● CIP Settings

Setting	Description	Default
CIP routing	Specify whether to route the CIP messages received by the built-in EtherNet/IP port or internal port to other ports. If you select <i>Do not use</i> , CIP messages that you receive will be discarded.	Use



Testing Communications

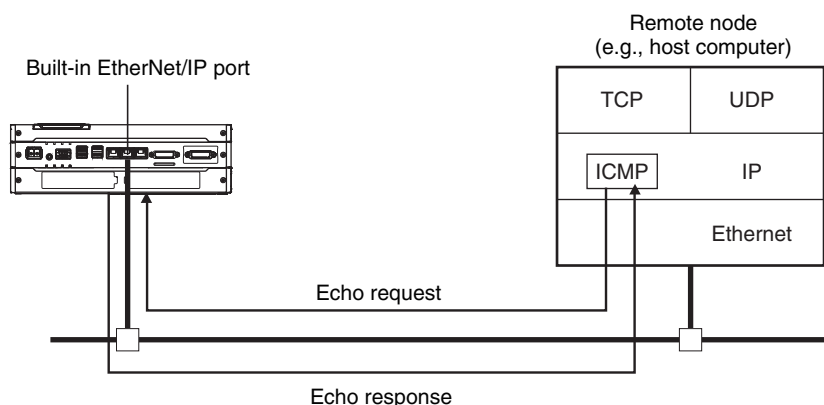
6-1	Testing Communications	6-2
6-1-1	PING Command	6-2
6-1-2	Using the PING Command	6-2
6-1-3	Host Computer Operation	6-3

6-1 Testing Communications

If the basic settings (in particular the IP address and subnet mask) have been made correctly for the built-in EtherNet/IP port, then it is possible to communicate with nodes on the EtherNet/IP network. This section describes how to use the PING command to test communications with the built-in EtherNet/IP port.

6-1-1 PING Command

The PING command sends an echo request packet to a remote node and receives an echo response packet to confirm that the remote node communications are normal. The PING command uses the ICMP echo request and responses. The echo response packet is automatically returned in the ICMP. The PING command is normally used to check the connections of remote nodes when you set up a network. The built-in EtherNet/IP port supports both the ICMP echo request and response functions. If the remote node returns a normal response to the PING command, then the nodes are physically connected correctly and Ethernet node settings are correct.



6-1-2 Using the PING Command

The built-in EtherNet/IP port automatically returns the echo response packet in response to an echo request packet sent by another node (e.g., host computer).

6-1-3 Host Computer Operation

The PING command can be executed from the host computer to send an echo request packet to a built-in EtherNet/IP port. The following example shows how to use the PING command in the host computer.

Application Method

Input the following command at the host computer's prompt (\$):

```
$ ping IP_address (host_name)
```

The destination is specified by its IP address or host name.



Additional Information

The PING command is not supported by some host computers.

Application Example

In this example, a PING command is sent to the node at IP address 130.25.36.8. The "\$" in the example represents the host computer prompt.

● Normal Execution

```
$ ping 130.25.36.8 ← Executes the PING command.
PING 130.25.36.8: 56 data bytes
64 bytes from 130.25.36.8: icmp_seq=0. time=0. ms
64 bytes from 130.25.36.8: icmp_seq=0. time=0. ms
: : : :
64 bytes from 130.25.36.8: icmp_seq=0. time=0. ms ← Press the Ctrl+C Keys to cancel execution.

---- 130.25.36.8 PING Statistics ----
9 packets transmitted, 9 packets received, 0% packets loss
round-trip (ms) min/avg/max = 0/1/16
$
```

● Error

```
$ ping 130.25.36.8 ← Executes the PING command.
PING 130.25.36.8: 56 data bytes
← Press the Ctrl+C Keys to cancel execution.

---- 130.25.36.8 PING Statistics ----
9 packets transmitted, 0 packets received, 100% packets loss
$
```

Refer to the command reference manual for your computer's OS for details on using the PING command.

7

Tag Data Link Functions

7-1	Introduction to Tag Data Links	7-2
7-1-1	Tag Data Links	7-2
7-1-2	Data Link Data Areas	7-3
7-1-3	Tag Data Link Functions and Specifications	7-6
7-1-4	Overview of Operation	7-7
7-1-5	Starting and Stopping Tag Data Links	7-9
7-1-6	Controller Status	7-9
7-1-7	Concurrency of Tag Data Link Data	7-12
7-2	Setting Tag Data Links	7-16
7-2-1	Starting the Network Configurator	7-16
7-2-2	Tag Data Link Setting Procedure	7-18
7-2-3	Registering Devices	7-19
7-2-4	Creating Tags and Tag Sets	7-21
7-2-5	Connection Settings	7-32
7-2-6	Creating Connections Using the Wizard	7-42
7-2-7	Creating Connections by Device Dragging and Dropping	7-45
7-2-8	Connecting the Network Configurator to the Network	7-48
7-2-9	Downloading Tag Data Link Parameters	7-53
7-2-10	Uploading Tag Data Link Parameters	7-56
7-2-11	Verifying the Tag Data Links	7-59
7-2-12	Starting and Stopping Tag Data Links	7-63
7-2-13	Clearing the Device Parameters	7-64
7-2-14	Saving the Network Configuration File	7-66
7-2-15	Reading a Network Configuration File	7-68
7-2-16	Checking Connections	7-69
7-2-17	Changing Devices	7-71
7-2-18	Displaying Device Status	7-72
7-3	Ladder Programming for Tag Data Links	7-73
7-3-1	Ladder Programming for Tag Data Links	7-73
7-3-2	Status Flags Related to Tag Data Links	7-77
7-4	Tag Data Links with Other Models	7-78

7-1 Introduction to Tag Data Links

7-1-1 Tag Data Links

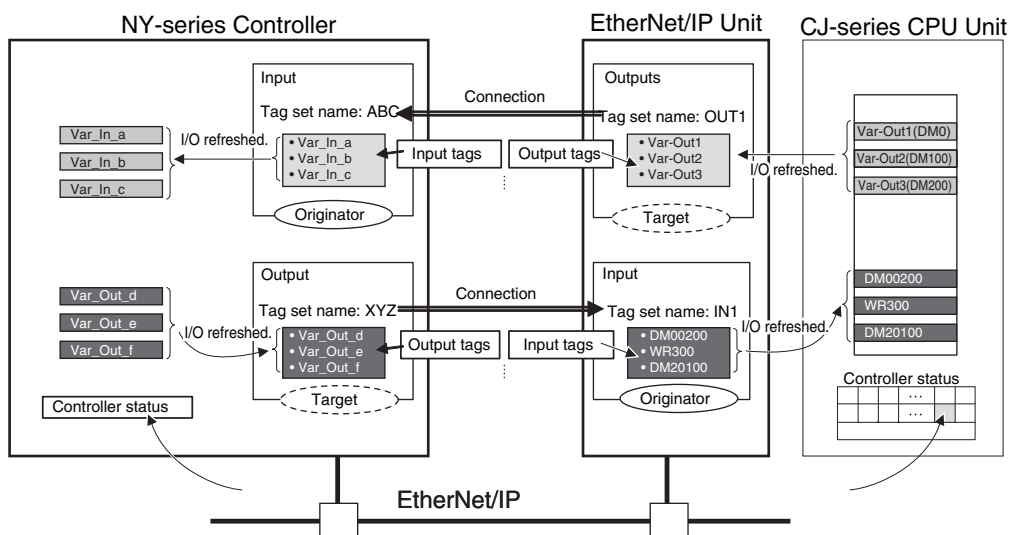
Tag data links enable cyclic tag data exchanges on an EtherNet/IP network between Controllers or between Controllers and other devices. Variables are assigned to tags. The settings for tag data links are made with the Network Configurator. Refer to *7-2 Setting Tag Data Links* for information on how to make the settings.



Additional Information

You can also use the Sysmac Studio to set the tag data links. Refer to *A-2 Use the Sysmac Studio to Set the Tag Data Links (EtherNet/IP Connections)* for details on setting the tag data links in the Sysmac Studio.

With a tag data link, one node requests the connection of a communications line to exchange data with another node. The node that requests the connection is called the originator, and the node that receives the request is called the target.



For communications between Controllers, the connection information is set in the built-in EtherNet/IP port of the Controller that receives (consumes) the data (i.e., the originator).



Additional Information

For communications between a Controller and an I/O device, the connection information is set in the built-in EtherNet/IP port that is the originator. If an I/O device is used, the Network Configurator must have an EDS file installed that includes connection information for the I/O device. Refer to *A-3 EDS File Management* for the installation procedure.

The output words and input words for each node for which data is exchanged are set in the connection information. These words are called the output tag set and input tag set. A tag set must specify at least one tag. The size of the data for data exchange is the total size of the tags included in the tag set. The size of the output tag set and the size of the input tag set must match.

7-1-2 Data Link Data Areas

Tags

A tag is a unit that is used to exchange data with tag data links. Data is exchanged between the local network variables and remote network variables specified in the tags or between specified I/O memory areas.



Precautions for Correct Use

To maintain concurrency in the values of network variables that are assigned to tags, you must set refreshing tasks. Refer to *7-1-7 Concurrency of Tag Data Link Data* for details.

Tag Sets

When a connection is established, from 1 to 8 tags (including Controller status) is configured as a tag set. Each tag set represents the data that is linked for a tag data link connection. Tag data links are therefore created through a connection between one tag set and another tag set. A tag set name must be set for each tag set.

Note A connection is used to exchange data as a unit within which data concurrency is maintained. Thus, data concurrency is maintained for all the data exchanged for the tags in one data set.

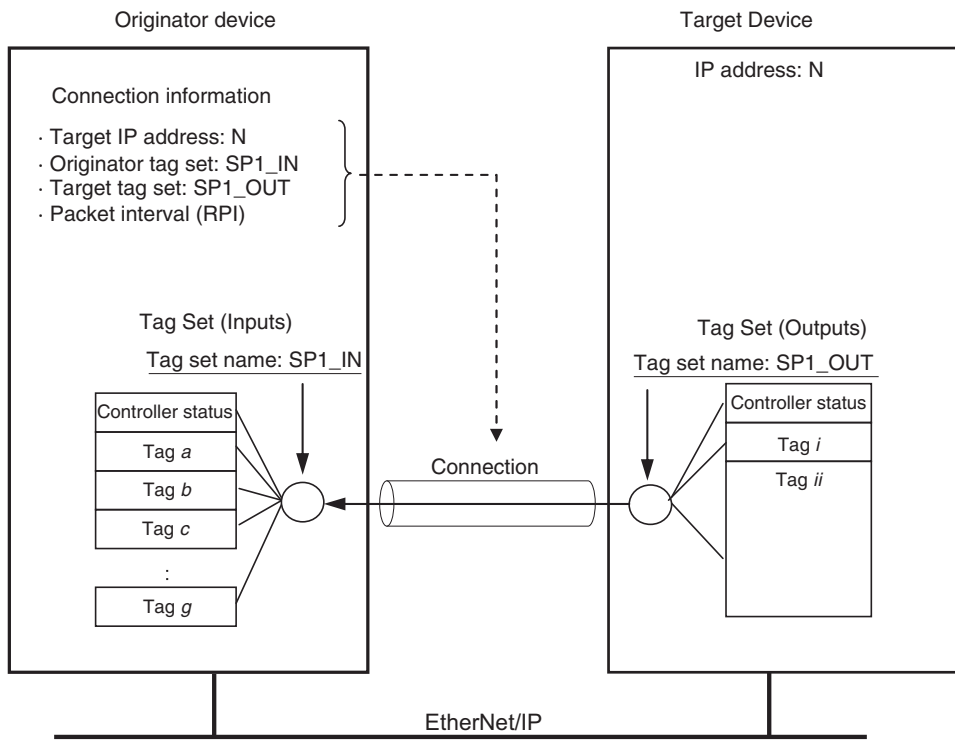


Precautions for Correct Use

Data for the tags is exchanged in the order that the tags are registered in the tag sets. Register the tags in the same order in the input and output tag sets.

● **Example**

In the following example, input tags *a* to *g* at the originator are a tag set named *SP1_IN* and output tags *i* and *ii* are a tag set named *SP1_OUT*. A connection is set between these two tag sets.



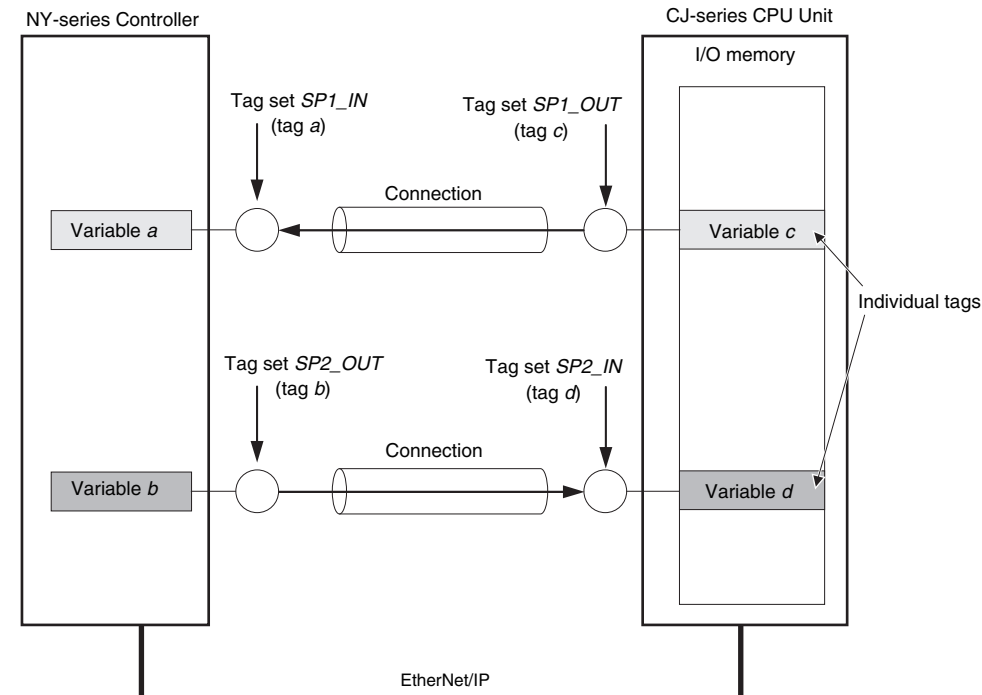
There are both input (consume) and output (produce) tag sets. Each tag set can contain only input tags or only output tags. The same input tag cannot be included in more than one input tag set.

● Number of Tags in Tag Sets

You can set any tag sets containing one or more tags for the input and output tag sets for one connection. For example, you can set a tag set with one tag for the input tag set and set a tag set with more than one tag for the output tag set.

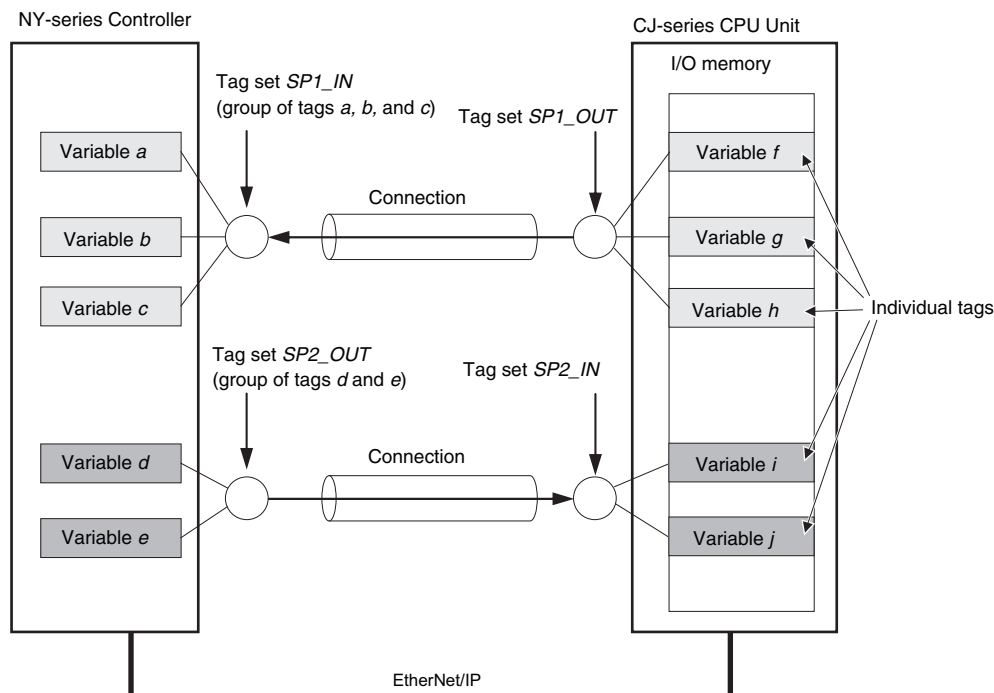
Tag Sets with Only One Tag Each

With basic Network Configurator procedures, each tag set contains only one tag.



Tag Sets Each with Multiple Tags

As shown below, tags can be grouped. You can place up to eight tags (up to 722 words in total) in one tag set.



Note To enable a connection, each tag set must include only input tags or only output tags. (Both input and output tags cannot be included in the same tag set.)

7-1-3 Tag Data Link Functions and Specifications

The tag data link and performance specifications of the NY-series Controller are given below.

Item	Specification	
	NY5□2-□□□□	
	Unit version 1.12 or later	
Communications type	Standard EtherNet/IP implicit communications (connection-type cyclic communications)	
Setting method	After you have set the tags, tag sets, and connections with the Network Configurator, you must download the tag data link parameters to all devices on the EtherNet/IP network. You can export the network variables that you created on the Sysmac Studio to a CSV file. You can then import the file to the Network Configurator and assign the network variables to tags. After the parameters are downloaded, the EtherNet/IP ports are restarted to start the tag data links.	
Tags	Supported variable types	You can specify the following network variables as tags. *1, *2 • Global variables
	Maximum number of words per tag	722 words (1444 bytes)
	Maximum number of tags	256*3
Tag sets	Maximum number of tags per tag set	8 (7 when Controller status is included)
	Maximum number of words per tag set	722 words (1444 bytes)
	Maximum number of tag sets	128
Connections	Maximum number of connections per Unit: 128	
Connection type	Each connection can be set for 1-to-1 (unicast) or 1-to-N (multi-cast) communications.	
Packet interval (RPI)	1 to 10,000 ms in 1-ms increments	
	The packet interval can be set separately for each connection.	
Permissible communications band (pps)	20,000 pps	
	Note The heartbeat is included.	

*1 You can import network variables created in the Sysmac Studio to the Network Configurator as tags. However, variables with a Network Publish attribute that have variable names that are the same as the I/O memory address notation, such as "0000" and "H0000" are not exported to CSV files.

*2 The following table lists the variables that you can specify as tags.

Type	Example	Specification	
Variables with basic data types	aaa	Supported.	
Enumerated variables	bbb	Supported.	
Array variables	Arrays	ccc	Supported.
	Elements	ccc[2]	Supported.
Structure variables	Structures	ddd	Supported.
	Members	ddd.xxx	Supported.
Union variables	Unions	eee	Supported.
	Members	eee.yyy	Supported.

*3 The maximum number of tags is given for the following conditions.

- All tag sets contain eight tags.
- The maximum number of tag sets (32) is registered.

7-1-4 Overview of Operation

In this manual, the connection information that is set is called tag data link parameters. This section describes how to set tag data links with the Sysmac Studio and the Network Configurator.

Setting Network Variables (Sysmac Studio)

First, create any variables that you want to use for tag data links as network variables in the Sysmac Studio.

- 1** Set the Network Publish attribute to Input or Output in the Global Variable Table for the variables you want to use for tag data links (i.e., as tags).
- 2** To maintain concurrency in tag data within a tag set, set all tags (i.e., variables with a Network Publish attribute) within the same tag set as follows:
Set a refreshing task for variables with a Network Publish attribute to maintain concurrency as described below for tag data link data.
Refer to *7-1-7 Concurrency of Tag Data Link Data* for details on the concurrency of tag data link data.
 - Maintain concurrency in the tag data in a tag set.
 - The timing of updating network variables that are assigned to tags is synchronized with the execution period of the program that accesses the network variables.



Precautions for Correct Use

You cannot use the following notation, which specifies an I/O memory address, in the variable name of any variable used in a tag data link.

- 1) Variable names that contain only single-byte numerals (Example: 001)
- 2) Variable names with the following single-byte letters (uppercase or lowercase) followed by single-byte numerals
 - H (Example: H30)
 - W (Example: w30)
 - D (Example: D100)
 - E0_ to E18_

Setting and Downloading Tag Data Link Parameters (Network Configurator or Sysmac Studio)

The tag data link parameters (e.g., connection information) that are described below are created with the Network Configurator or the Sysmac Studio, and then the parameters are downloaded to all originator devices on the EtherNet/IP network. When the tag data links are used on the built-in EtherNet/IP port, use the Network Configurator to make the following settings.



Additional Information

In the setting of the following tag data link parameter, the specifications of the settable numbers and the ranges differ depending on the Controller or the version of the Controller. For details, refer to *1-3-1 Specifications*.

- 1** Creating the Configuration Information
Register the EtherNet/IP ports and EtherNet/IP Units to create the connections that define the tag data links. Refer to *7-2-3 Registering Devices* for details.
- 2** Setting Tags

Create Controller variables for input (consume) tags and output (produce) tags. You can import and export network variables that are created on the Sysmac Studio to CSV files. This allows you to register them as tags on the Network Configurator. Output tags can be defined to clear output data to 0 or to hold the output data from before the error when a fatal error occurs in the Controller.

3 Setting Tag Sets

Create output tag sets and input tag sets and assign tags to them. You can specify the Controller status that indicates the Controller's operating status (operating information and error information) in a tag set.

4 Setting Connections

Link the output tag sets for the target device and the input tag sets for the originator device as connections.

● Connection Setting Parameters

The connection settings in step 4 above have the following setting parameters.

Setting the Requested Packet Interval (RPI)

The RPI (Requested Packet Interval) is the I/O data refresh cycle on the Ethernet line when tag data links are established. With EtherNet/IP, data is exchanged on the communications line at the RPI that is set for each connection, regardless of the number of nodes.

With the built-in EtherNet/IP port, you can set RPI for each connection.

Setting Multi-cast and Unicast Communications

You can select a multi-cast connection or unicast (point-to-point) connection as the connection type in the tag data link connection settings. With a multi-cast connection, you can send an output tag set in one packet to multiple nodes and make allocations to the input tag sets. A unicast connection separately sends one output tag set to each node, and so it sends the same number of packets as the number of input tag sets. Therefore, multi-cast connections can decrease the communications load if one output tag set is sent to multiple nodes. To use a multi-cast connection and send an output tag set in one packet to multiple nodes, the following settings for the receiving node must be the same as the settings of the sending node: the connection type (multi-cast), the connection I/O types, packet internals (RPI), and timeout values.



Precautions for Correct Use

- The performance of communications devices is limited to some extent by the limitations of each product's specifications. Consequently, there are limits to the packet interval (RPI) settings. Refer to *13-2 Adjusting the Communications Load* and set an appropriate packet interval (RPI).
- If multi-cast connections are used, however, use an Ethernet switch that has multi-cast filtering, unless the tag set is received by all nodes in the network. If an Ethernet switch without multi-cast filtering is used, the multi-cast packets are broadcast to the entire network, and so packets are sent to nodes that do not require them, which will cause the communications load on those nodes to increase.
- If you will use data tag links with multicast at a communication speed over 100 Mbps, use an Ethernet switch that supports a communication speed of 1000 Mbps. If there is an Ethernet device on the same network that communicates at a speed of 100 Mbit/sec or less, the device may affect tag data link communication and cause tag data links to be broken, even if the device is not related to tag data link communication.



Additional Information

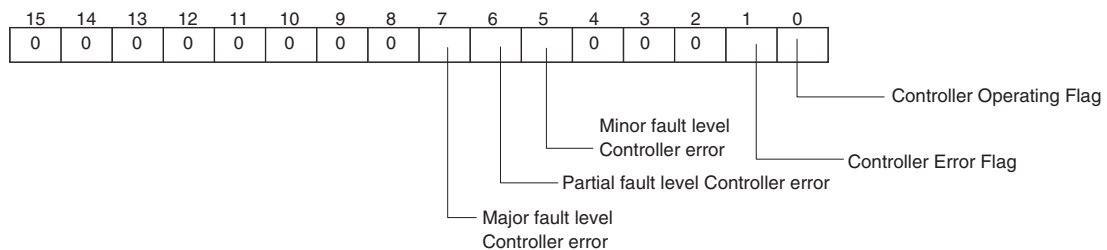
- To calculate the number of connections of each connection type, refer to *13-1-2 Calculating the Number of Connections*.
- If the maximum number of connections is exceeded, you must review the number of connections for the built-in EtherNet/IP port, or the number of nodes. When you use an NJ-series CPU Unit, you can also consider adding EtherNet/IP Units.

7-1-5 Starting and Stopping Tag Data Links

Tag data links are automatically started when the data link parameters are downloaded from the Network Configurator and the power supply to the NY-series Controller is turned ON. Thereafter, you can start and stop tag data links for the entire network or individual devices from the Network Configurator. Starting and stopping tag data links for individual devices must be performed for the originator. Furthermore, you can use system-defined variables to start and stop the entire network. Refer to *7-2-12 Starting and Stopping Tag Data Links* for details.

7-1-6 Controller Status

You can include the Controller status as a member of a tag set in the data sent and received. The Controller status is a set of flags that indicate the operating status of the Controller (operating information, error information, Controller error level). If the Controller status is specified as an output (send) tag, the Controller status is added to the start of the tag set in the following format. (Select the *Include* Option for the *Controller Status* in the upper right of the Edit Tag Set Dialog Box.)

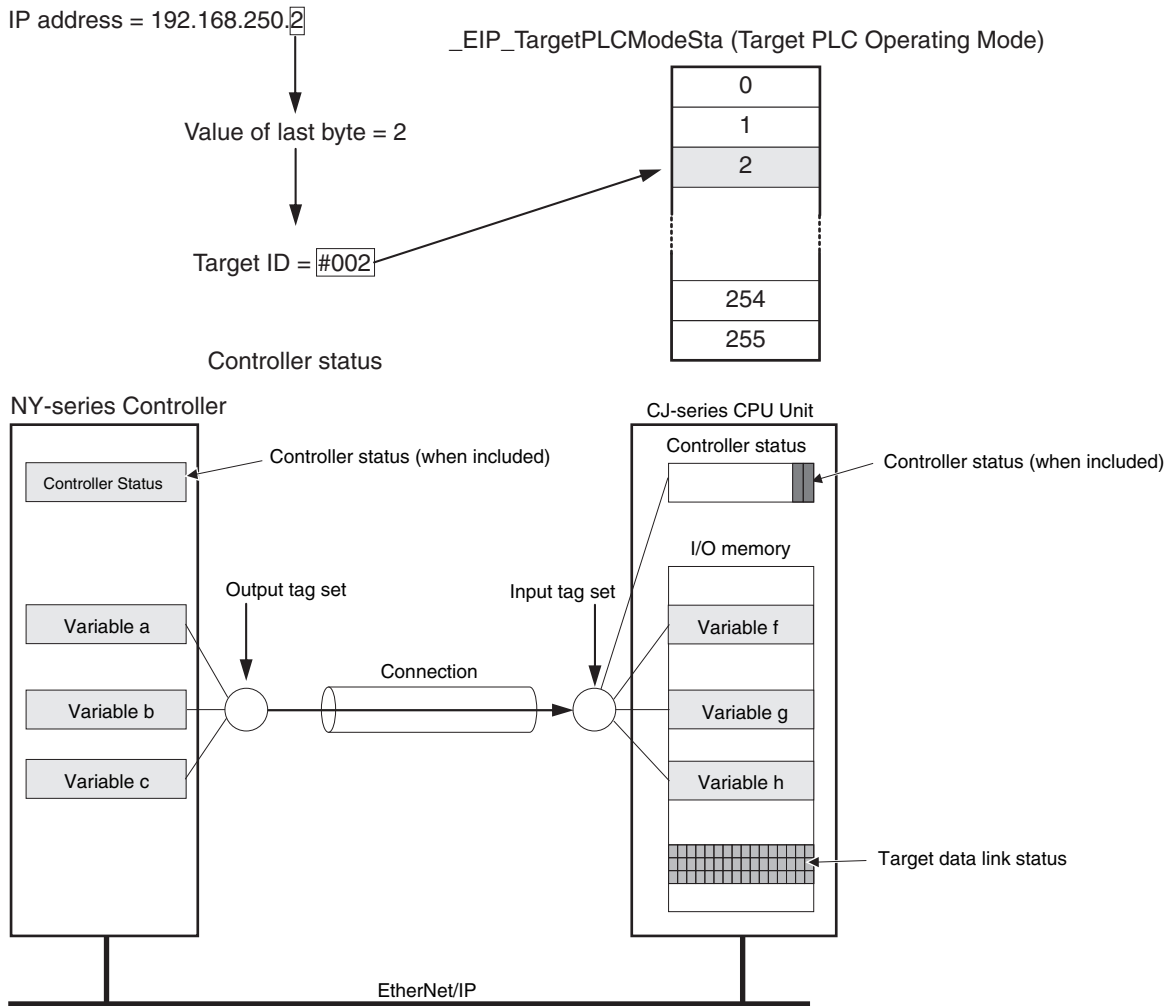


- * Of the flags in bits 5 to 7 that indicate the current error level, only the flag for the highest error level changes to TRUE. For example, if a minor fault level Controller error and a major fault level Controller error occur at the same time, only the flag for the major fault level Controller error (bit 7) will change to TRUE and the flag for the minor fault level Controller error (bit 5) will remain FALSE.

To receive the Controller status, specify the Controller status for the In - Consume Tab Page in the dialog box used to edit the receive tag set. (Select the *Include* Option for the *Controller Status* in the upper right of the Edit Tag Set Dialog Box.) When a tag data link is started, the contents of the Controller status is stored in the system variables that are given below.

- Target PLC Operating Mode
`_EIP_TargetPLCModeSta`
- Target PLC Error Information
`_EIP_TargetPLCErr`

Example: Using an NY-series Controller to send the Target PLC Operating Mode of the Target Node with an IP Address of 192.168.250.2



Additional Information

The target ID may be duplicated depending on the IP addresses of the target nodes. In this case, it is necessary to change the target ID with the Network Configurator so that the same address is not used twice. For information on how to change the target node ID, refer to step 4 under 7-2-5 *Connection Settings*.

When you use multiple connections to communicate with one specific node, the information in the Controller status is stored in the following variables if the Controller status is specified in the input tags and the output tags for all connections.

Controller status	Variable name	Description of operation
Controller Operating Flag	Target PLC Operating Mode EIP_TargetPLCModeSta	This flag shows the operation information of the Controller at the target node. When the Built-in EtherNet/IP Port Is the Originator of the Connection The array element that corresponds to the target ID at the target is TRUE when all information for all connections of the relevant target node is shows operating status. You can change the target ID of the IP address from the Network Configurator. This status information is enabled when the Controller status is included in the communications data in both the originator and target node. This variable is updated when necessary.
Controller Error Flag	Target PLC Error Information _EIP_TargetPLCErr	This variable shows the error status (logical OR of fatal and non-fatal errors) of the target node Controllers. When the Built-in EtherNet/IP Port Is the Originator of the Connection You can change the target ID of the IP address from the Network Configurator. The Controller Error Flags are enabled when the Controller status is included in the communications data for both the originator and target. This variable is updated when necessary.



Additional Information

Even if you specify including the Controller status in the output (produce) tags, you do not necessarily need to include it in the input (consume) tags. If you do not include the Controller status in an input (consume) tag, the contents of the Controller status is not updated in the Target PLC Operating Mode and Target PLC Error Information variables, but it is sent in the input (consume) tag. Therefore, you can use the Controller status data that was received in the input (consume) tag as receive data.

7-1-7 Concurrency of Tag Data Link Data

To maintain the concurrency of data in a tag data link, you must set a refreshing task for each network variable that is assigned to a tag.

- Maintain concurrency in the tag data in a tag set.
- The timing of updating network variables that are assigned to tags is synchronized with the execution period of the program that accesses the network variables.



Additional Information

A refreshing task maintains concurrency of the value of a global variable from all tasks that access that global variable. This is achieved by specifying a single task that can write to that global variable and not allowing any other task to write to that global variable. For details on refreshing tasks, refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558).

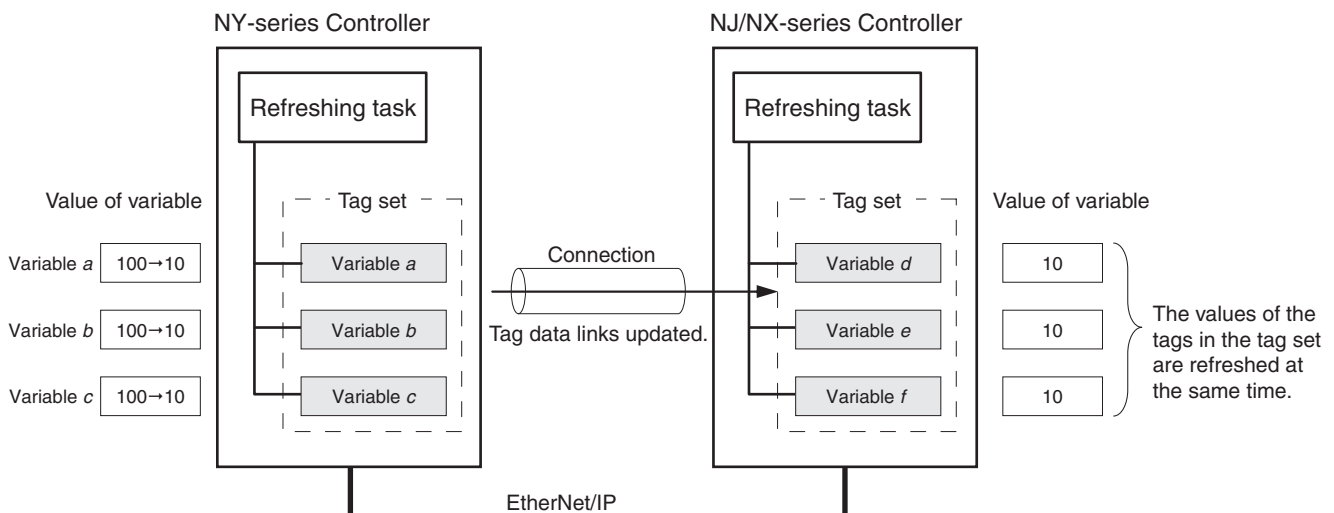
● Maintaining Concurrency in the Tag Data in a Tag Set

To maintain concurrency in the values of multiple tags in a tag set, the tags (variables with a Network Publish attribute) must satisfy the following four conditions.

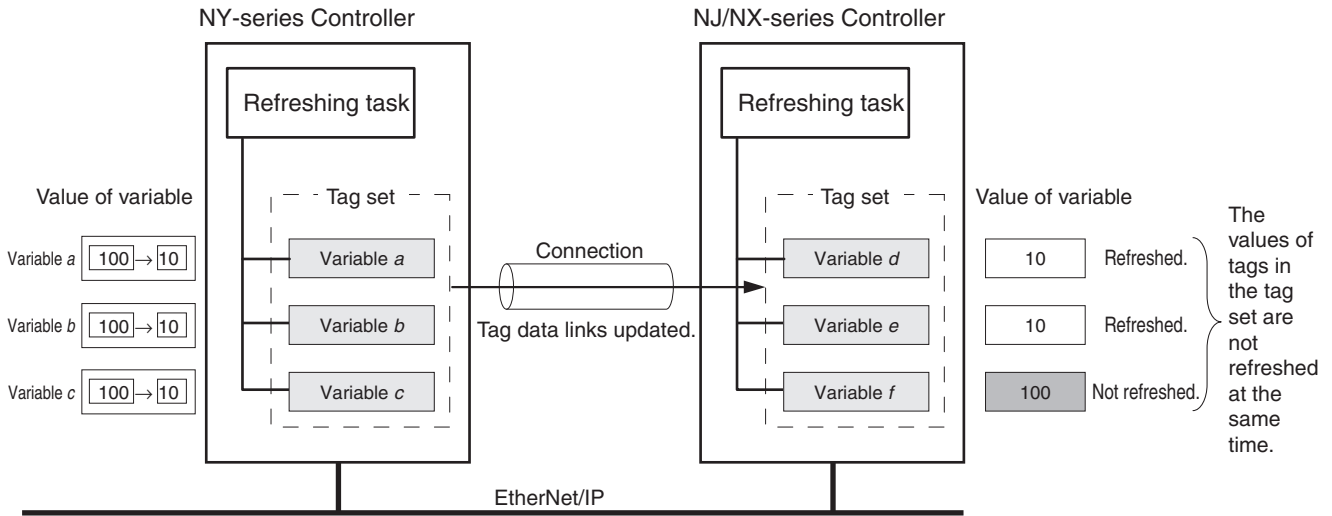
- (1) **The tags must be assigned to the same tag set (connection).**
- (2) **A refreshing task must be set for the network variables assigned to the tags, and the refreshing task must be the same for all tags in the tag set.***
- (3) **The variable access time set for each task must be set to a higher value than is required to transfer the tag data. Refer to 13-3-3 Relationship between Task Periods and Packet Intervals (RPIs) for details on the variable access time and data transfer.**

* If you set a refreshing task for a network variable, you must set a variable access time to allocate enough time to access the network variable from outside of the Controller.

- Setting Refreshing Tasks for Tags (Network Variables)
Concurrency of the tags in the tag set is maintained.



- Not Setting Refreshing Tasks for Tags (Network Variables)
Concurrency of the tags in the tag set is not maintained.



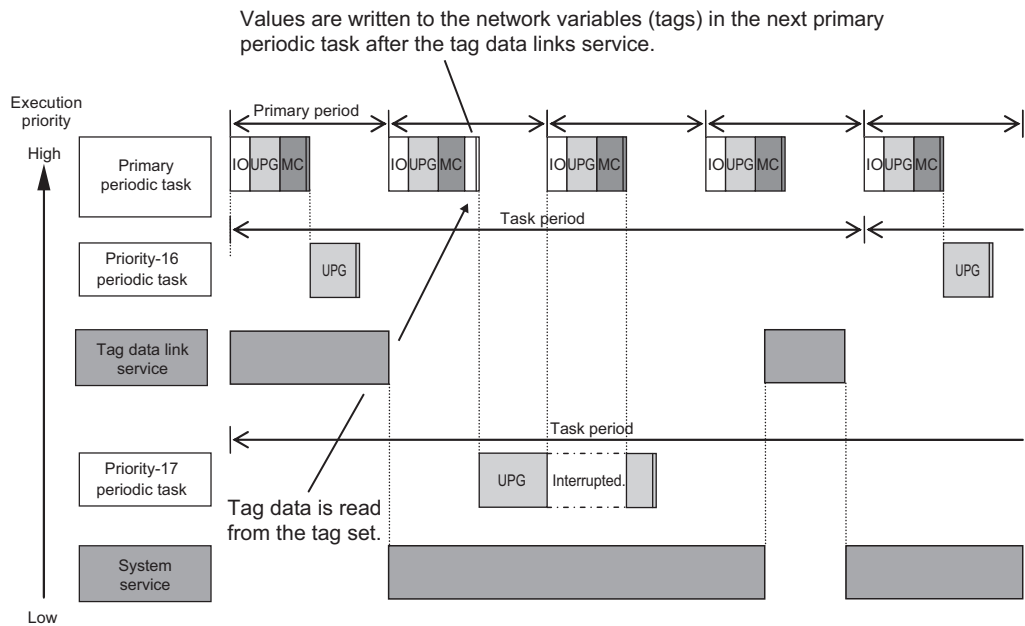
● **Synchronizing the Update Timing of Network Variables (Tags) with the User Program Execution Period**

To have the values of network variables (tags) updated to the latest tag data values each time the user program that accesses those network variables is executed, set the refreshing task for the network variables (tags) to the same type of the task as for the user program that accesses the network variables (tags).

The difference between the operation of tags with a refreshing task that is the same as the user program and tags without a refreshing task is described below.

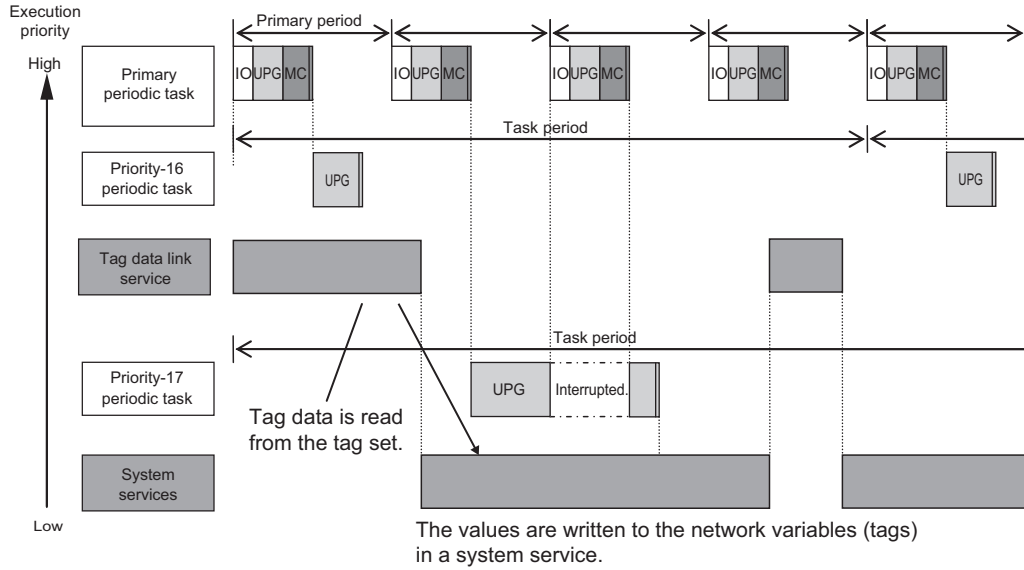
(1) **Tag (network variable) with a refreshing task**

In this example, the primary periodic task is set as the refreshing task. Values are written to the network variables (tags) in the next primary periodic task after the tag data link service.



(2) Tag (network variable) without a refreshing task

The values are written to the variables in a system service.



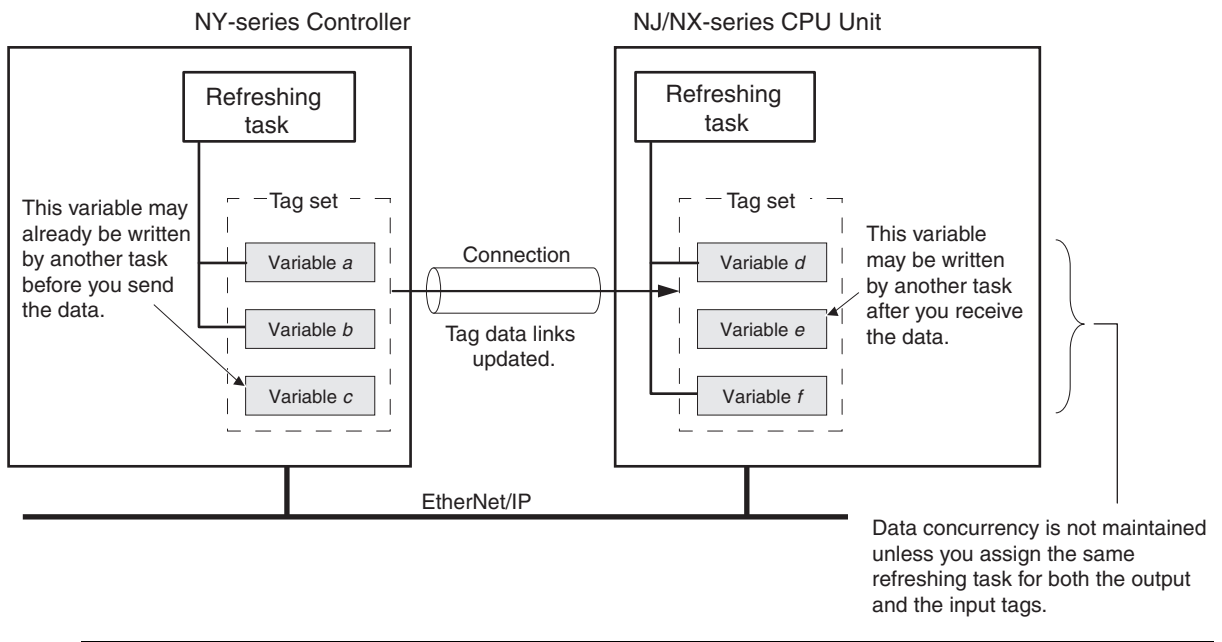
Additional Information

Relationship between Refreshing Tasks and Data Concurrency in Tag Data Links

If you do not specify a refreshing task for global variables in tag data links, the following may occur.

1. When the data is sent for the output tag set, another task may have already written different values before that data is sent, depending on the timing of the task.
2. When data is received by an input tag set, another task may write different values after that data is received, depending on the timing of the task.

Therefore, to maintain the concurrency of data in tag data links, you must specify the same refreshing task on both the output Controller and the input Controller.



● Required Processing Time to Maintain Concurrency

When you set a refreshing task for tags (network variables) to maintain the concurrency of data link data, the processing time required for that specified task increases. Due to this increase in task processing time, the refreshing of tag data link data may not occur during the packet interval (RPI) period set for each connection. Therefore, you need to adjust the packet interval (RPI) settings to match the period of the task specified as the refreshing task. Refer to *13-3-3 Relationship between Task Periods and Packet Intervals (RPIs)* for details.

● Task Setup Procedure

- (1) **Set the global variables for which to specify a refreshing task, and set the refreshing tasks and accessing tasks in the Settings for Exclusive Control of Variables in Tasks in the Task Setup on the Sysmac Studio.**
- (2) **Set the variable access time for each refreshing task.**

For details, refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558).

7-2 Setting Tag Data Links



Additional Information

You can also use the Sysmac Studio to set the tag data links. Refer to *A-2 Use the Sysmac Studio to Set the Tag Data Links (EtherNet/IP Connections)* for details on setting the tag data links in the Sysmac Studio.

7-2-1 Starting the Network Configurator

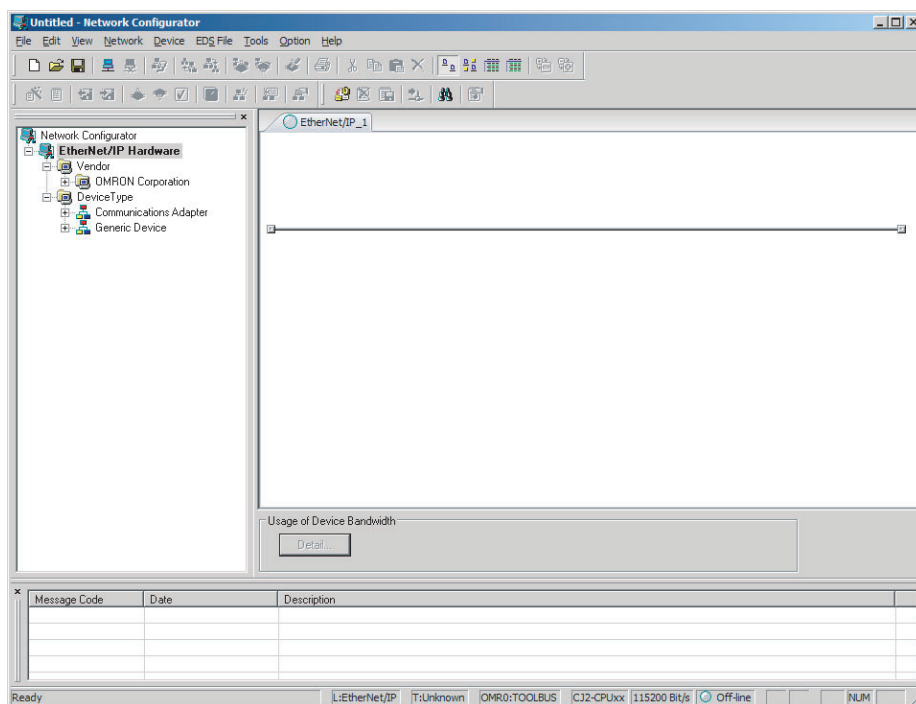
Procedure

Tag data links are set from the Network Configurator. Use the following procedure to start the Network Configurator.

- **Using the Windows Start Menu**

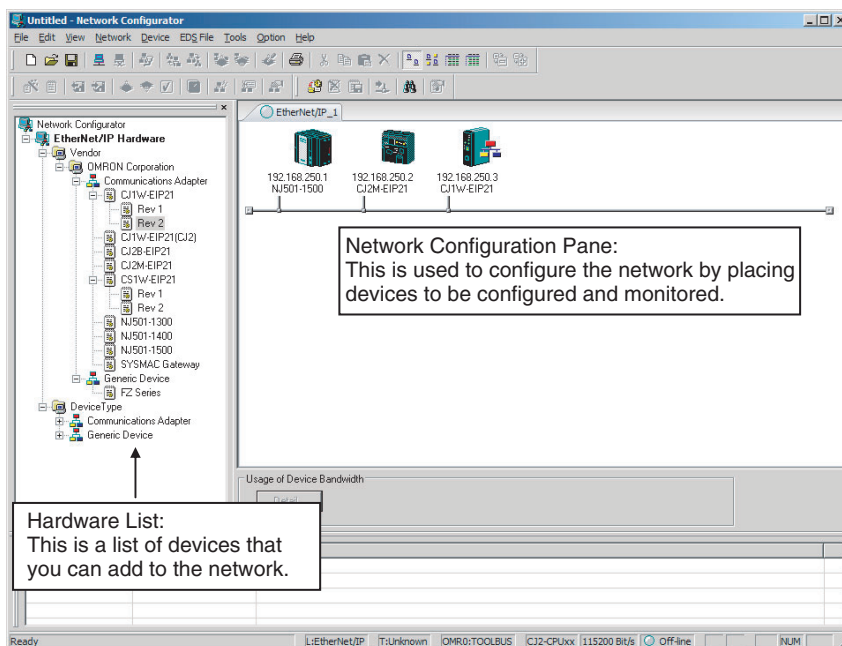
To start the Network configurator, select **OMRON - Sysmac Studio - Network Configurator for EtherNet/IP - Network Configurator** from the Windows Start Menu.

When the Network Configurator starts, the following window is displayed.

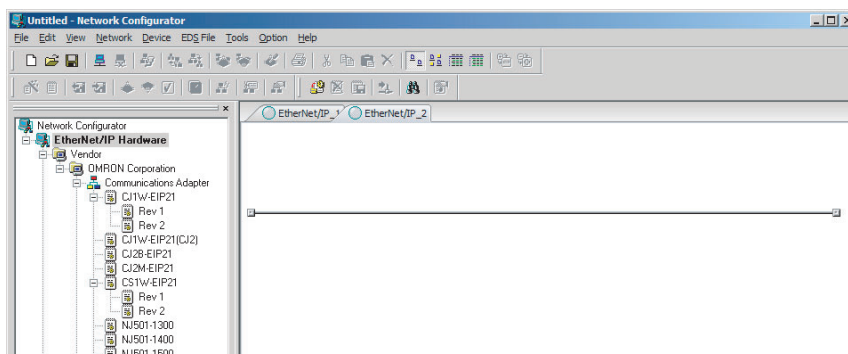


Main Window

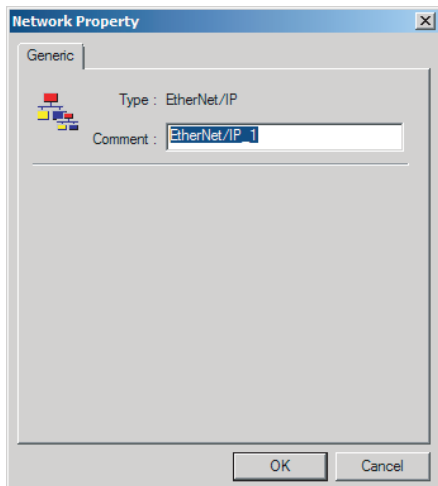
The Main Window consists of a Hardware List and a Network Configuration Pane, as shown in the following diagram.



To manage two or more networks, you can select **Network - Add** to add a new Network Configuration Pane. You can add a new Network Configuration Pane.



To change the name displayed in the Network Tab Page, select **Network - Property**. You can change the name set in the Comment Field of the Network Property Dialog Box.



7-2-2 Tag Data Link Setting Procedure

This section describes the procedure to set tag data links (i.e., connection information). For data links between Controllers, the connection information is set only in the originator, i.e., the node that receives data.

1 Create the network configuration.

- (1) **Register all built-in EtherNet/IP ports for which to create connections in the EtherNet/IP Network Configuration Pane. (Refer to 7-2-3 Registering Devices.)**

* If a system has already been installed, connect online to the EtherNet/IP network and upload the network configuration. (Refer to 7-2-10 Uploading Tag Data Link Parameters.)



2 Create the tag and tag set connections.

Set the connections with one of the following methods.

(1) **Basic Operation:**

1. Create tags and tag sets for all registered devices (built-in EtherNet/IP ports). (Refer to 7-2-4 Creating Tags and Tag Sets.)
2. Create a connection for the originator device (i.e., the registered device that receives data as input data). (Refer to 7-2-5 Connection Settings.)

- (2) **Create the connections by dragging registered devices. (Refer to 7-2-10 Uploading Tag Data Link Parameters.)**



3 Download the tag data link parameters. (Refer to 7-2-9 Downloading Tag Data Link Parameters.)



4 Make sure that the tag data links are operating normally by using the indicators for the built-in EtherNet/IP port (refer to *NYseries Troubleshooting Manual* (Cat. No.W564)) and the Network Configurator monitor functions. (Refer to 13-1 Communications System.)



5 Make sure that the output tag data is updated in the input tags by using the Sysmac Studio's Watch Tab Page.

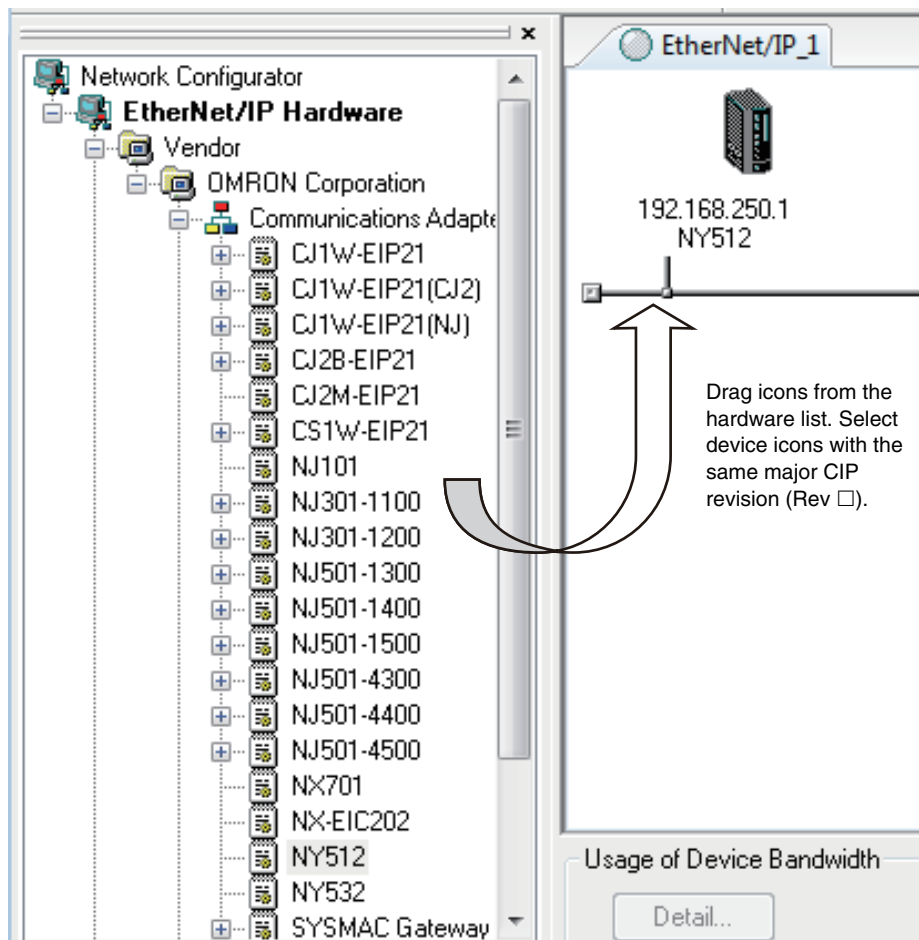
Note Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for the procedure.

7-2-3 Registering Devices

Register all of the devices required in the equipment (such as EtherNet/IP Units performing tag data links) in the network configuration.

- 1 Register the devices that will participate in the tag data links by dragging the devices from the Hardware List and dropping them in the Network Configuration Pane on the right. (To drag and drop an icon, click and hold the left mouse button over the icon, move the icon to the destination, and release the mouse button.)

You can also select a device in the Hardware List and press the **Enter** Key to register it. The icon of the device is displayed in the Network Configuration Pane, as shown in the following diagram.



The device names and major CIP revisions (Rev □) are displayed in the hardware list. For NY-series Controller, device names and major CIP revisions are as shown in the following table.

Device name in hardware list	Unit version	CIP revisions	
		Major revision	Revision name in hardware list
NY512	Unit version 1.12 or later	2	None
NY532	Unit version 1.12 or later	2	None



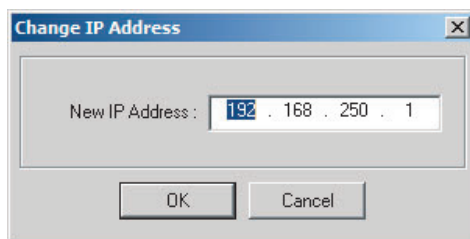
Precautions for Correct Use

Make sure that you select the devices with the same device names and the same major CIP revisions as the devices that you use in actual operation. The following will occur if any device names or CIP revisions are different when you attempt to download tag data link parameters on the Network Configurator.

- If a device name is different, an error message “Specified device can not be accessed, or wrong device type.” will be displayed and the download will fail.
- If a revision is different, “Wrong unit revision.” will be displayed and the download will fail.

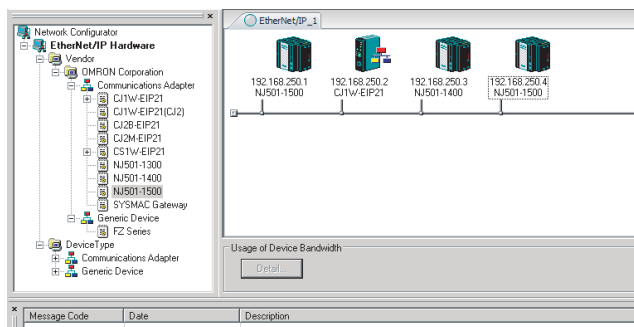
The above also applies when uploading or comparing tag data link parameters. In any of the above cases, refer to 7-2-17 *Changing Devices* and change the device.

- 2 Right-click the registered device’s icon to display the pop-up menu, and select **Change UNKNOWN Address**.



- 3 Set the IP address to match the node address (IP address) actually used in the device and click the **OK** Button.

- 4 Repeat steps 1 to 3, and register all of the devices that participate in the tag data links.



7-2-4 Creating Tags and Tag Sets

You must create the tag sets and set member tags required to create connections for a registered EtherNet/IP Unit. You can set the network variables used in control programs for tags.

This section first describes the basic procedure to create tags and tag sets (1, below). Then it explains how to import variables with a Network Publish attribute from the Sysmac Studio to the Network Configurator (2, below).

Then it explains how to effectively use network variables for tags.

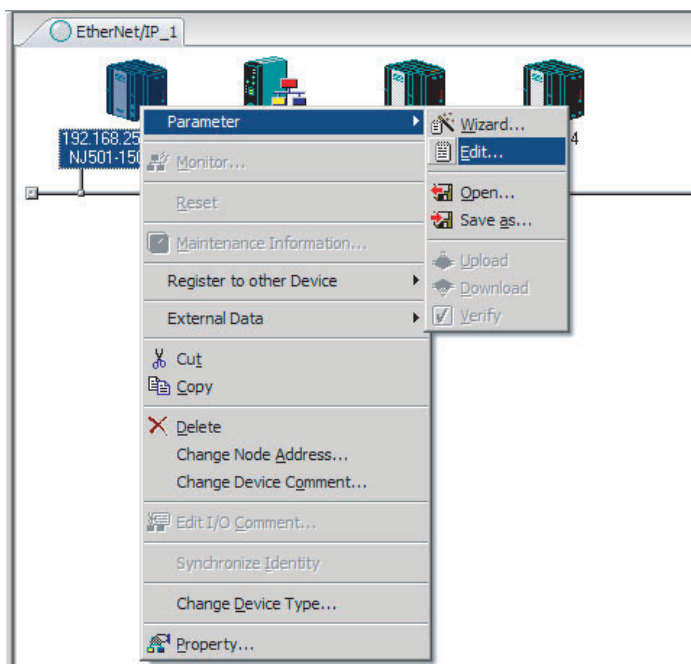
- (1) **Creating Tags and Tag Sets with the Network Configurator's Device Parameter Editing Function**
- (2) **Importing Variables with a Network Publish Attribute Created in the Sysmac Studio to the Network Configurator**

(1) Creating Tags and Tag Sets with the Network Configurator's Device Parameter Editing Function

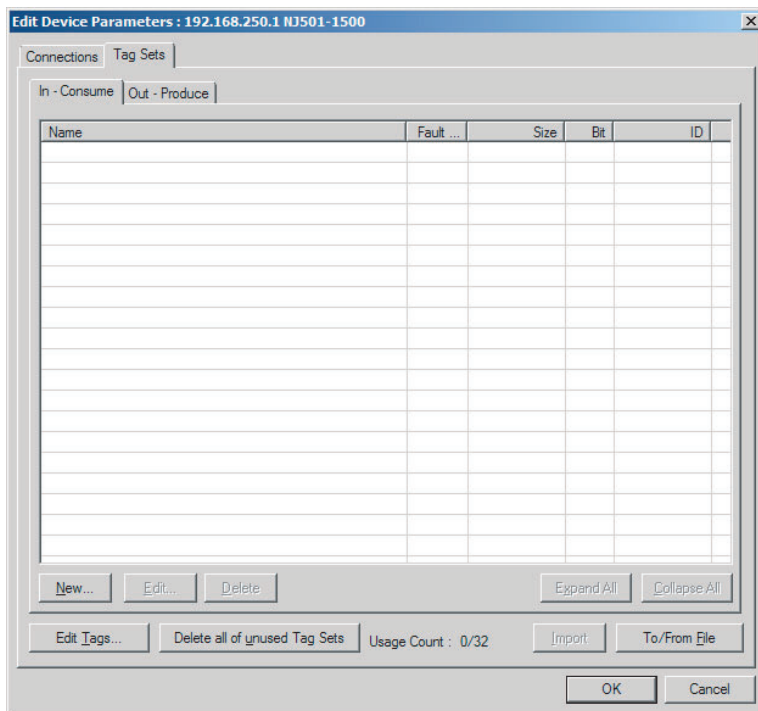
● Creating a Tag Set

- 1** Double-click the icon of the device for which to create a tag set, or right-click the icon to display the pop-up menu and select **Parameter – Edit**.

The Edit Device Parameters Dialog Box is displayed.



- 2 Click the **Tag Sets** Tab at the top of the Edit Device Parameters Dialog Box. There are two kinds of tag sets: input (consume) and output (produce).

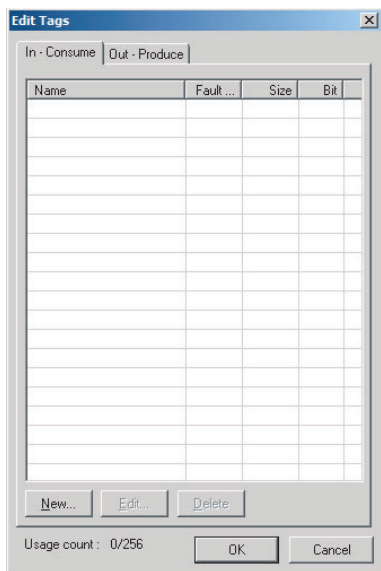


● **Creating and Adding Tags**

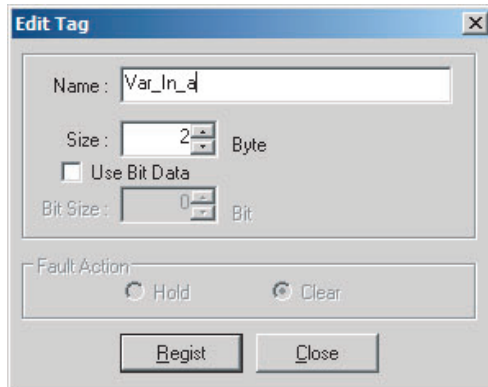
- 3 Click the **Edit Tags** Button.

The Edit Tags Dialog Box is displayed.

Register the input (consume) tags and output (produce) tags separately.



- 4** Click the **In - Consume** Tab, and then click the **New** Button. The Edit Tag Dialog Box is displayed.



- 5** Enter the variable name directly into the *Name* Box. (Example: Var_In_a)



Additional Information

- You can use the following characters in tag names. 0 to 9, A to Z, a to z, single-byte kana, _ (underbar), and multi-byte characters (e.g., Japanese)
- You cannot use the following characters in tag names. ! " # \$ & ' () * + , - . / : ; < = > ? @ [] ^ ` % spaces or text strings that start with numerals (0 to 9)
- The maximum length of a tag name is 255 bytes.
- Specify array variables, structure variables, and union variables as shown below.
 - Specifying array elements: *array [2][3]* (or *array [2,3]*) and *array [2][3][4]* (or *array [2,3,4]*)
 - Specifying structure members: *Struct.member* (Separate the member name with a period.)
 - Specifying union members: *Union.member* (Separate the member name with a period.)



Precautions for Correct Use

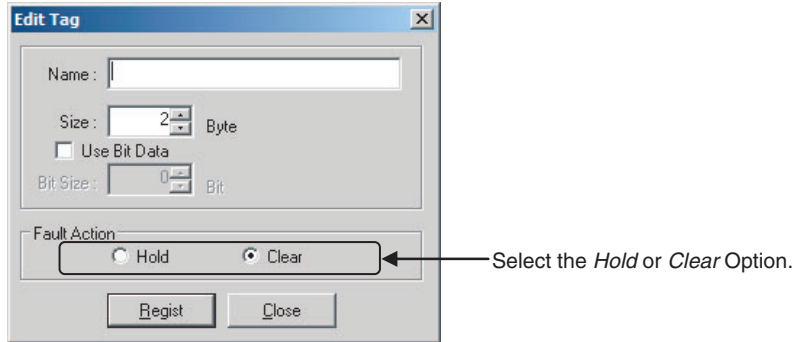
You cannot use the following notation, which specifies an I/O memory address, in the variable name of any variable used in a tag data link.

- 1) Variable names that contain only single-byte numerals (Example: 001)
- 2) Variable names with the following single-byte letters (uppercase or lowercase) followed by single-byte numerals
 - H (Example: H30)
 - W (Example: w30)
 - D (Example: D100)
 - E0_ to E18_

- 6** Input the size of the tag in bytes in the *Size* Field. Input the tag size so that it is the same as the data type size of the variable. Select the *Use Bit Data* Check Box and change the bit size to 1 to use BOOL variables.
- 7** Click the **Register** Button to register the tag. If an I/O memory address is specified for a tag name, the Edit Tag Dialog Box is displayed with the next consecutive address as the tag name for the next tag. After you have registered all of the tags, click the **Close** Button.
- 8** Click the **Out - Produce** Tab, and then click the **New** Button. The Edit Tag Dialog Box is displayed. Input the output tag in the same way. Use the Fault Action setting of the output (produce) tag to specify whether to clear the output data or continue to send it when a major fault occurs in the Controller.

The Fault Action setting is not required for input (consume) tag sets.

- Retain output for major fault: *Hold* (default)
Output data maintains its previous status even after a major fault occurs.
- Clear output at major fault: *Clear*
Output data is cleared to 0 when a major fault occurs.

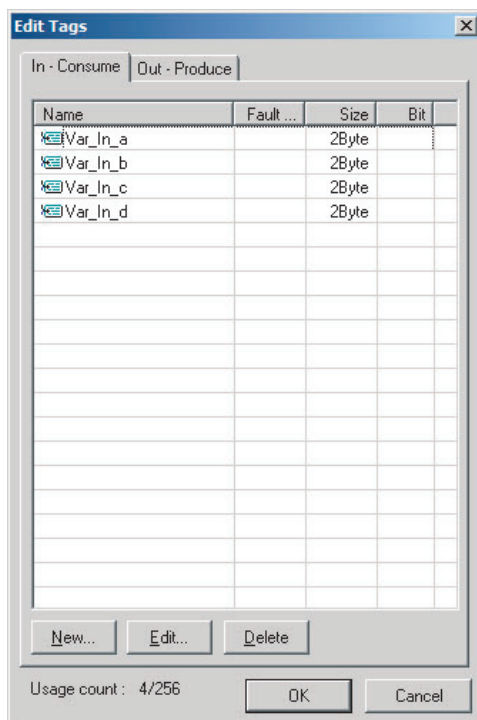


Precautions for Correct Use

Connections are cut off if any of the following errors occurs in the CPU Unit that is the originator while tag data links are active.

- Major fault level Controller error
- Partial fault level Controller error

- 9** After you register all of the required tags, click the **OK** Button in the Edit Tags Dialog Box.



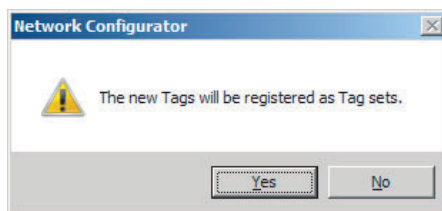


Precautions for Correct Use

Make the following settings to refresh all of the tag data in the same tag set at the same time.

- Use the Sysmac Studio to specify, in advance, the same refreshing task for all of the variables that are assigned to tags in the tag set.
- Do not place tag variables that have AT specifications in I/O memory and tag variables that do not have AT specifications in the same tag set.

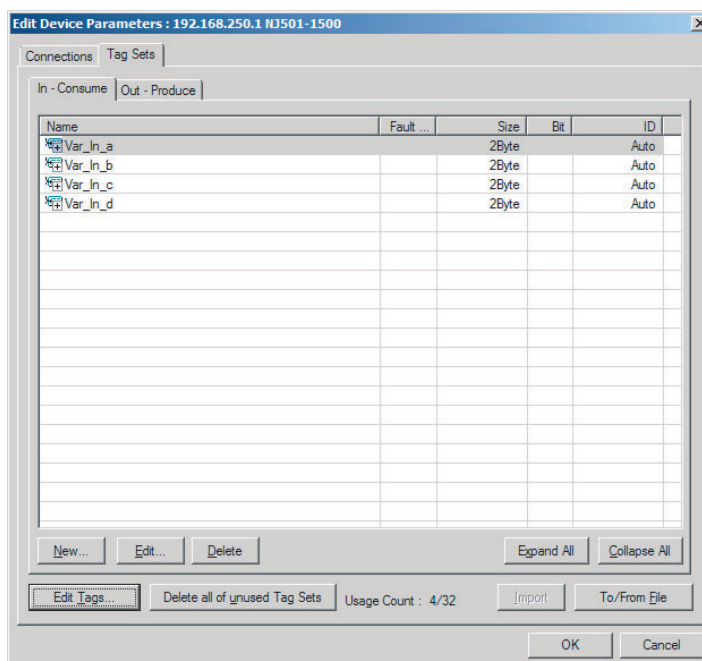
- 10** At this point, a confirmation dialog box is displayed to check whether the registered tag names are used as the tag set names. A tag set can contain up to eight tags, but tag sets are registered with one tag per tag set if the tag names are registered as tag set names. In this case, click the **Yes** Button.




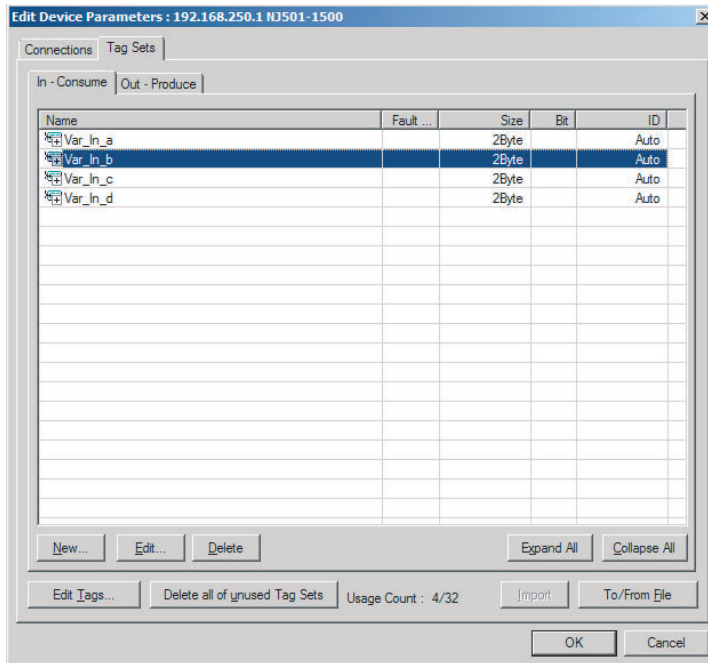
If the **No** Button is clicked, you can add more tags to the tag set. Refer to step 18 for details on how to register new tags first and add more tags to the tag set later.

● Changing and Registering Tag Sets

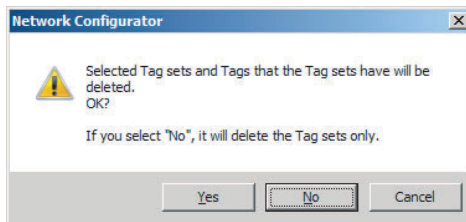
- 11** The following dialog box is displayed when the tags in the Edit Tags Dialog Box are registered directly as tag sets.



- 12** If an input tag is already registered in an input tag set, and you want to change its registration to a different input tag set, it is necessary to delete the tag from the tag set in which it was originally registered. Open the Edit Device Parameters Dialog Box, select the tag set containing the tag that you want to delete on the Tag Sets Tab Page, and click the **Delete** Button in the Edit Tags Dialog Box. (If there are other tags registered in that tag set, it is possible to delete just one tag by selecting the tag that you want to delete in the Edit Tag Set Dialog Box and clicking the  Button.)

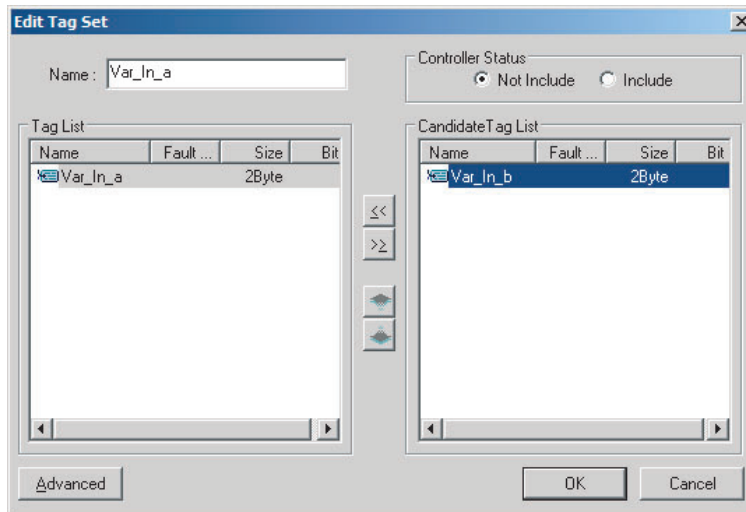


A confirmation message is displayed.



If the **No** Button is clicked, only the tag set is deleted. Click the **No** Button.

- 13** To edit a registered tag set and add tags, either double-click the tag set, or select the tag set and click the **Edit** Button. The Edit Tag Set Dialog Box is displayed.



The Tag List on the left side of the dialog box shows the tags that are already registered, and the Candidate Tag List on the right side of the dialog box shows the other tags that are not registered yet. To add a tag, select it in the Candidate Tag List and click the **<<** Button.

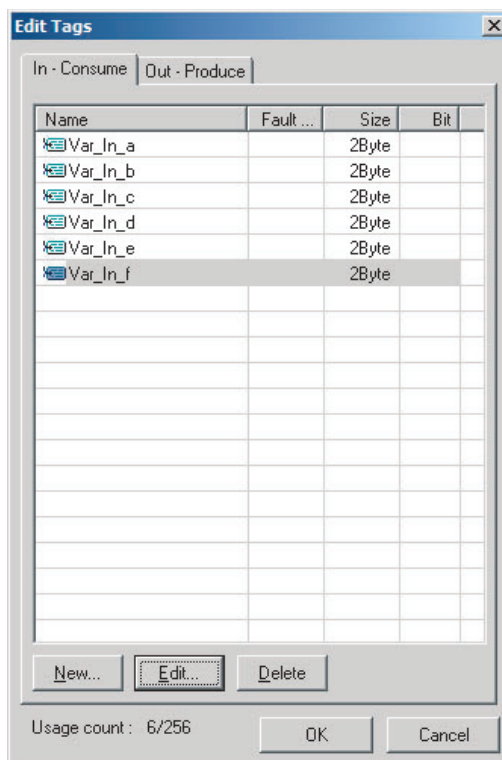
- 14** To include the Controller status in the tag set, select the *Include* Option at the upper-right corner of the dialog box.



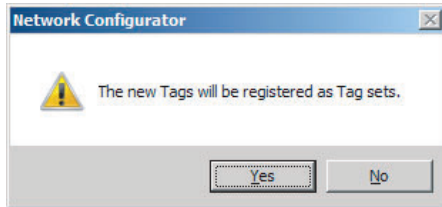
- 15** To confirm a change, click the **OK** Button in the Edit Tag Set Dialog Box.

- 16** Click the **OK** Button in the Edit Device Parameters Dialog Box.

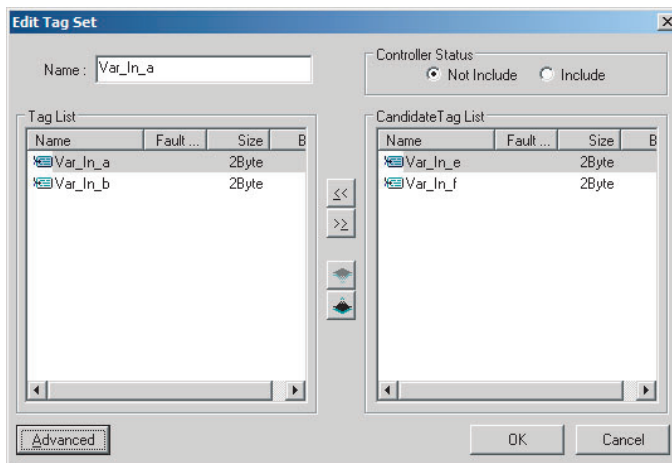
- 17** If you want to just add new tags and register the tag set, first register the tags with steps 1 to 9. In this example, input tags Var_In_e, Var_In_f are newly added.



- 18** After you register all of the required tags, click the **OK** Button at the bottom of the Edit Tags Dialog Box.
- 19** At this point, a confirmation dialog box is displayed to check whether the registered tag names are used as the tag set names. Tags are just added in this case, so click the **No** Button. Just the tags are registered. The tags are not registered as tag sets.

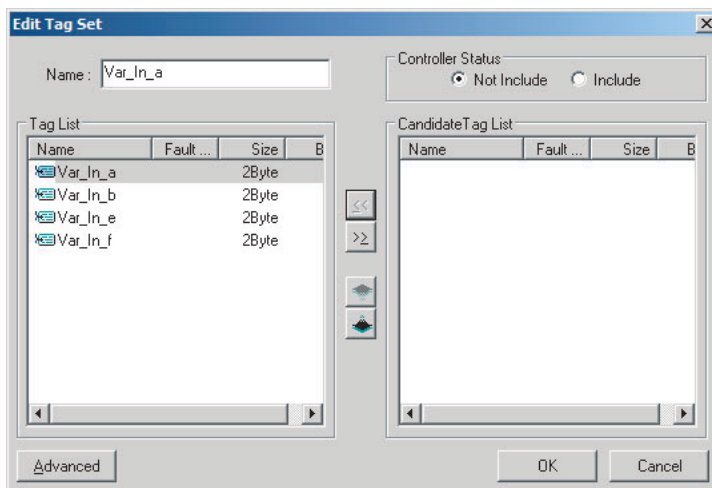




- 20** To register the newly added tags in a tag set, either double-click the desired tag set, or select the tag set and click the **Edit** Button.



The Tag List on the left side of the dialog box shows the tags that are already registered, and the Candidate Tag List on the right side of the dialog box shows the other tags that are not registered yet.

- 21** Select the tags that you want to add from the Candidate Tag List and click the **<<** Button.



You can register up to eight tags in a tag set. (If you include the Controller status in the tag set, you can register up to only seven tags, and two bytes are added to the size.) Data is sent and received in the order it is displayed in the tag list. To change the order of a tag, select the tag and click the Up and Down Buttons ( ).

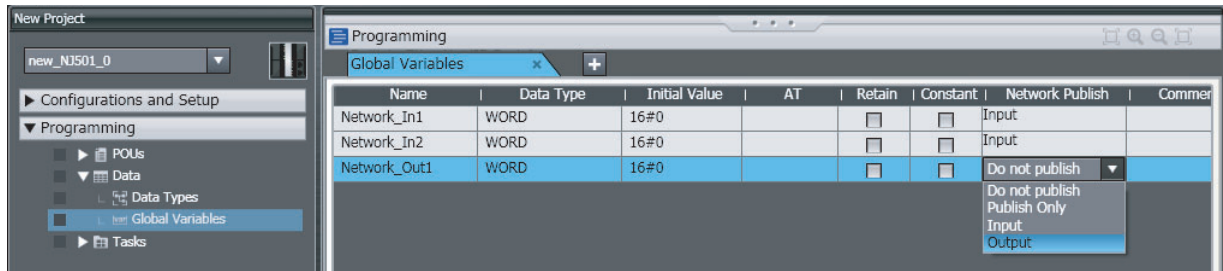
- 22** To confirm the changes, click the **OK** Button at the bottom of the Edit Tag Set Dialog Box.
- 23** Click the **OK** Button in the Edit Device Parameters Dialog Box.

(2) Importing Variables with a Network Publish Attribute Created on the Sysmac Studio to the Network Configurator

You can create network variables in the Sysmac Studio and import these variables to the Network Configurator to assign them to tags and tag sets. Use the following procedure.

● Creating Global Variables on the Sysmac Studio

- 1 Create a global variable with the Global Variable Editor of the Sysmac Studio and select *Input* or *Output* for the Network Publish attribute of the variable. Save the project when you are finished.



- 2 Select **Export Global Variables - Network Configurator...** from the Tools Menu.

Any global variables with *Input* or *Output* set for the Network Publish attribute are imported from the csv file for the import procedure described below (*Importing to the Network Configurator*).

● Importing to the Network Configurator



Precautions for Correct Use

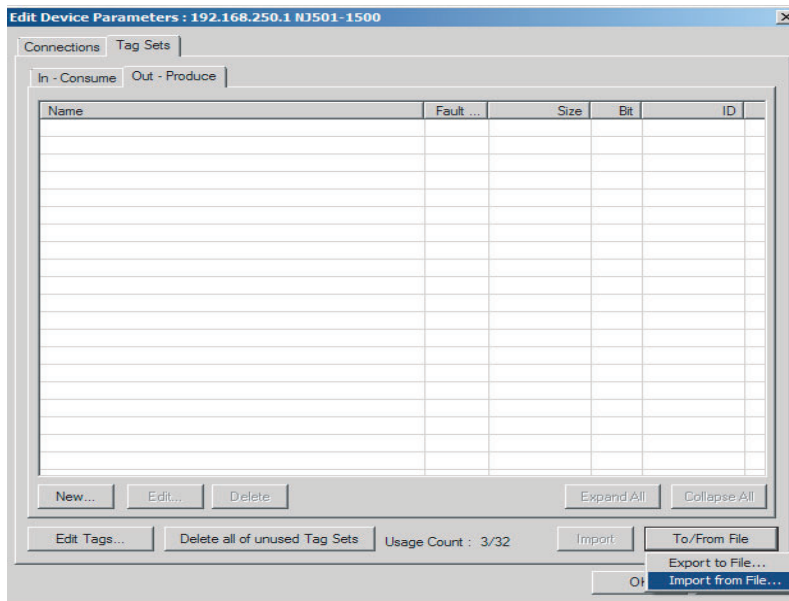
Variables with a Network Publish attribute that have variable names that are the same as the I/O memory address notation, such as “0000” and “H0000” are not exported to CSV files.

- 1) Variable names that contain only single-byte numerals (Example: 001)
- 2) Variable names with the following single-byte letters (uppercase or lowercase) followed by single-byte numerals
 - H (Example: H30)
 - W (Example: w30)
 - D (Example: D100)
 - E0_ to E18_ (Example: EA_100)

- 1 Double-click the icon of the device registered in the Network Configurator for which you want to import the variable with a Network Publish attribute to display the Edit Device Parameters Dialog Box.

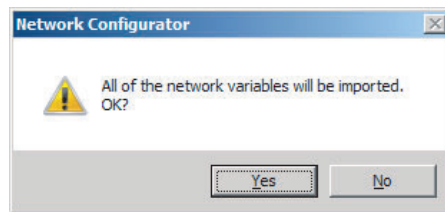
Right-click the icon to display the pop-up menu, and select **Device - Parameter - Edit**.

- 2** Click the **Tag Sets** Tab at the top of the Edit Device Parameters Dialog Box. Select **Import from File** from the **To/From File** Button.

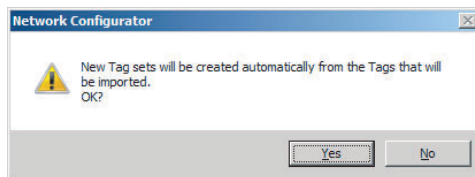


A confirmation dialog box is displayed that asks you how you want to import the variables as shown below.

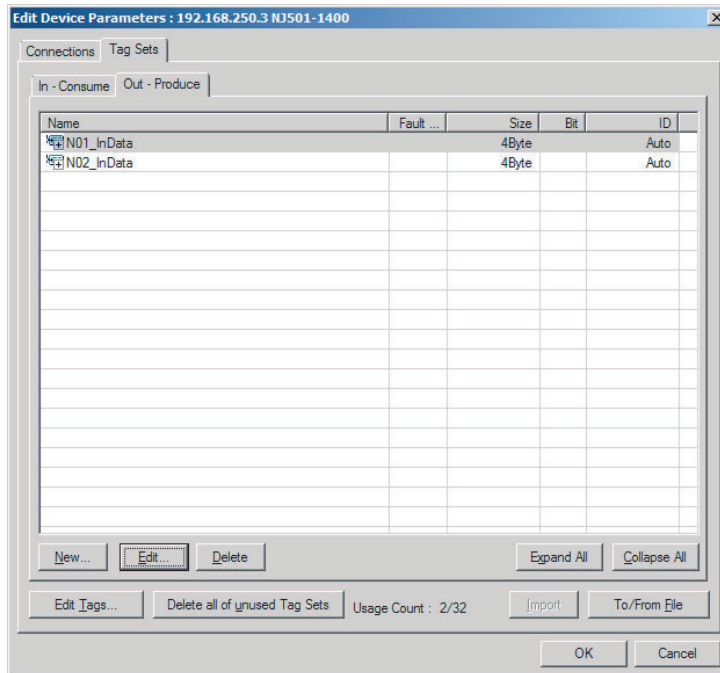
- To import all variables with a Network Publish attribute, click the **Yes** Button. To import only some of these variables, click the **No** Button.



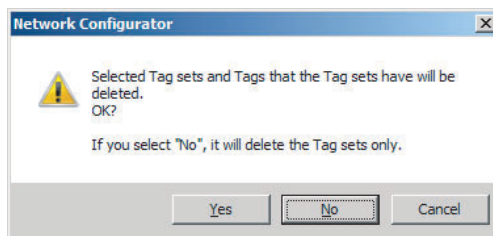
After you import the variables to the tags, click the **Yes** Button to automatically create tag sets, or click the **No** Button to set up tag sets manually.



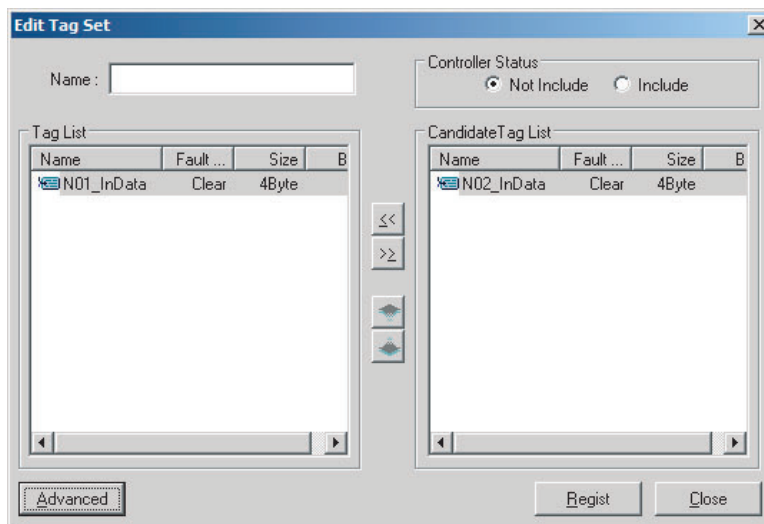
The variables will be imported as shown below on the Tag Sets Tab Page. Each variable will be imported into a different tag set and the device parameters will be automatically edited. (The variable name will be used for the tag set name.)



To place more than one input variable (input tag) imported from the Sysmac Studio into one tag set, you must delete the input tags that were registered. Select the tag set containing the variables you want to put into a tag set, then click the **Delete** Button. A confirmation dialog box is displayed to confirm that you want to delete the selected tag set and the tags contained in that tag set. You only want to delete the tag set, so click the **No** Button.



Click the **New** Button to create a new tag set. To place more than one tag in an existing tag set, double-click the tag set, or select it and click the **Edit** Button. The Edit Tag Set Dialog Box is displayed. Imported tags that are not registered in another tag set are displayed in the Candidate Tag List on the right. Click the Button to add tags individually.



- 3 You can change tag set names in this dialog box. To confirm a change, click the **Register** Button in the Edit Tag Set Dialog Box.
- 4 Perform steps 1 to 3 for all the devices to import variables and to create tag sets.

7-2-5 Connection Settings

After you create the tag sets, click the **Connections** Tab at the top of the Edit Device Parameters Dialog Box, and set the following connection information.

- The target devices with which connections are opened
- The connection type (multi-cast or unicast)
- The length of the packet intervals (RPI)
- Connection name (optional)

Make the connections settings in the originator only. The connections settings are not necessary in the target device.



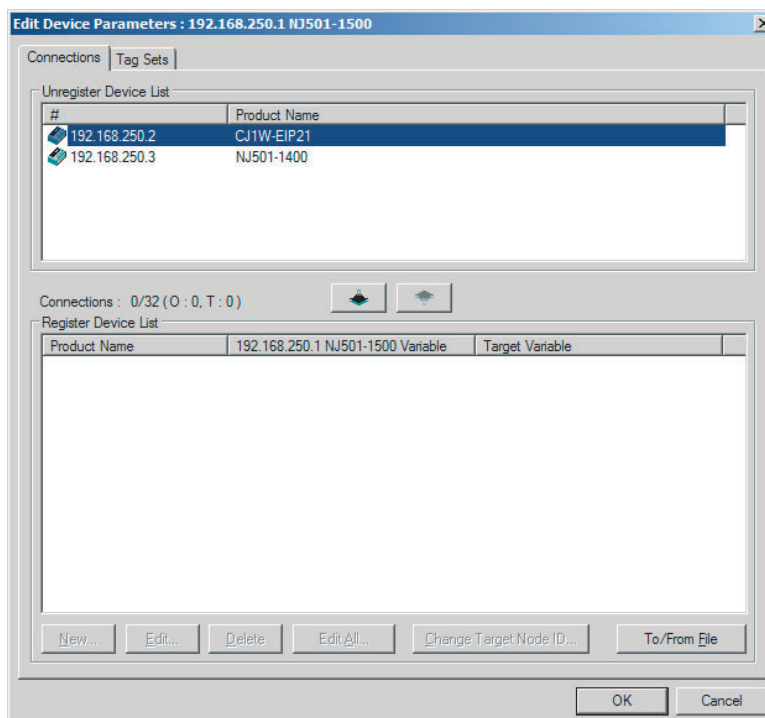
Precautions for Correct Use


Make the connections settings after you create tag sets for all of the devices involved in tag data links.

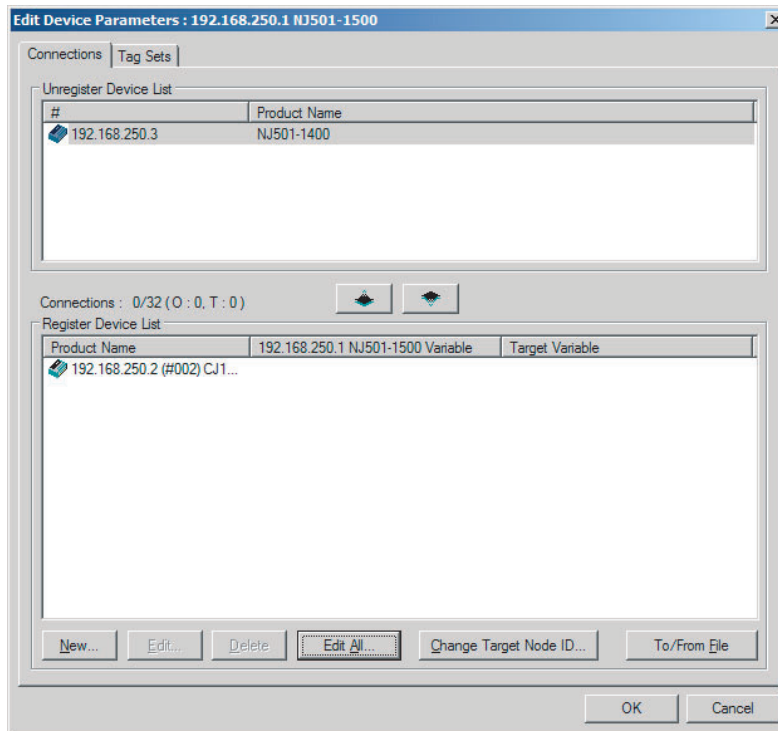
Connection Settings (Connections Tab Page)

● Registering Devices in the Register Device List

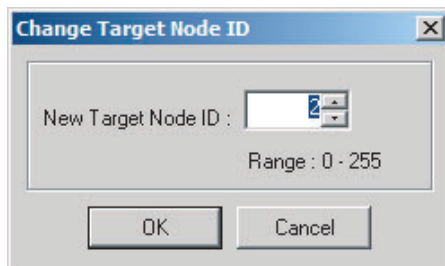
- 1 Double-click the icon of the device for which to make originator settings in the Network Configuration Pane of the Network Configurator. The Edit Device Parameters Dialog Box is displayed. Right-click the icon to display the pop-up menu, and select **Parameter – Edit**.
- 2 Click the **Connections** Tab in the Edit Device Parameters Dialog Box. All of the devices registered in the network (except the local node) are displayed.



- 3 In the Unregister Device List, click the target device that requires connection settings so its color changes to gray, and click the  Button. The selected target device is displayed in the Register Device List, as shown in the following diagram.



- 4 Target node IDs are assigned to the devices that are registered in the Register Device List. The target node ID serves as the bit array position for the following variables in the originator Controller: Target Node Controller Mode, Target Node Controller Error Information, Target Node Error Information, Registered Target Node Information, and Normal Target Node Information. By default, the target ID is automatically set to the rightmost 8 bits of the IP address. In the example above, the target device's IP address is 192.168.250.2, so the device number is #002. If a target node ID is duplicated and you want to change the device number, click the **Change Target Node ID** Button and change the target ID.



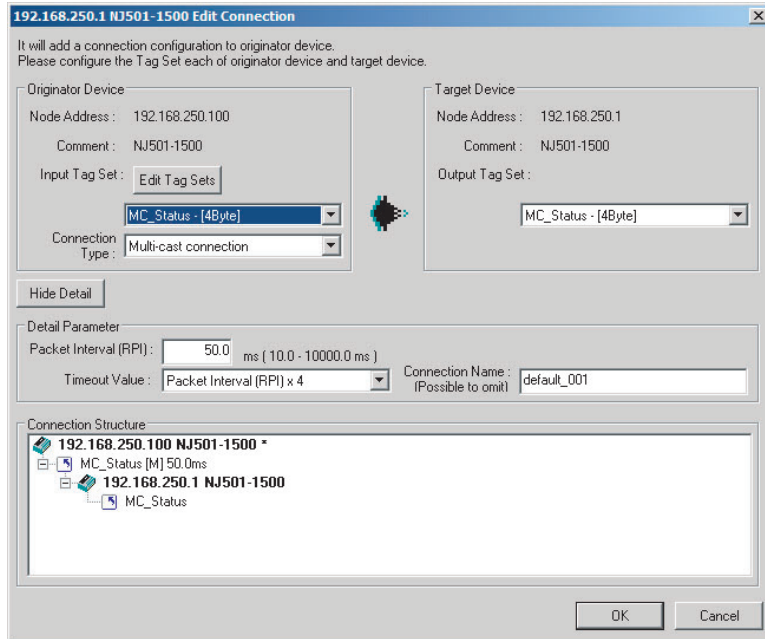
● **Editing Settings for Individual Connections**

You can edit each connection separately.

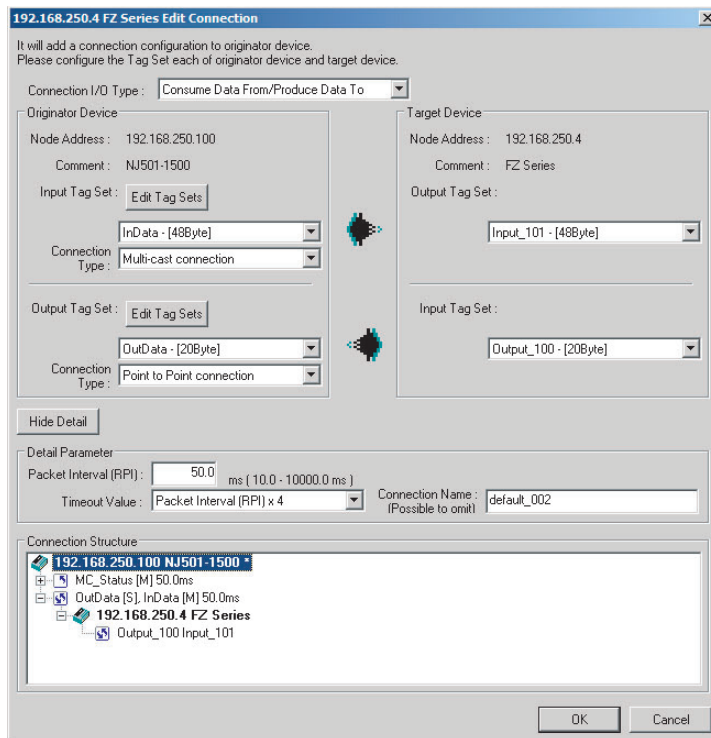
Note Refer to the following section for information on how to perform batch editing in a table format.

- 1 Click the **Connections** Tab and then click the **New** Button. The following Edit Connection Dialog Box is displayed according to the type of device that is selected.

Using Built-in EtherNet/IP Ports as Targets (for Input Only)



Using Other EtherNet/IP Devices as Targets (for Settings Other Than Input Only)



The settings are as follows:

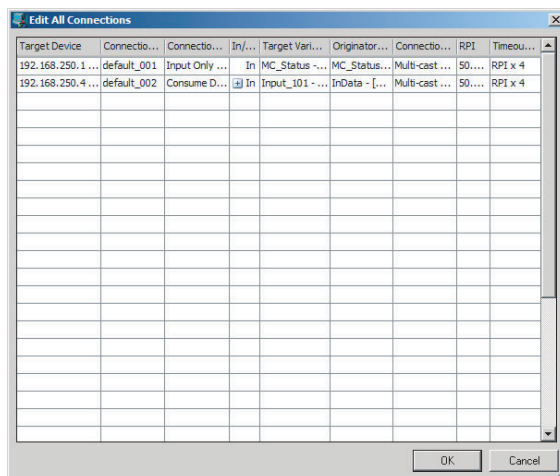
Setting	Description
Connection I/O Type	Select Input Only (tag type) to use tag data links with a CS1W-EIP21, CJ1W-EIP21, CJ2B-EIP21, CJ2M-EIP21, CJ1W-EIP21(CJ2), CJ1W-EIP21(NJ), NX701, NJ501-□□□□, NJ301-□□□□, NJ101, or NY5□2. When you create tag data links for other devices, select the connection I/O type specified in that device's EDS file. Use the Input Only (ID type) setting when another company's node is the originator and does not support connection settings with a Tag type setting.
Connection Type	Select whether the data is sent in multi-cast or unicast (point-to-point) format. The default setting is multi-cast. <ul style="list-style-type: none"> Multi-cast connection: Select when the same data is shared by multiple nodes. This setting is usually used. Point-to-Point connection: Select when the same data is not shared by multiple nodes. In a unicast transmission, other nodes are not burdened with an unnecessary load. <p>Note Refer to 7-1-4 <i>Overview of Operation</i> for details on using multi-cast and unicast connections, and counting the number of connections.</p>
The Connection Structure Area and the following items are not displayed if the Hide Detail Button is clicked.	
Packet Interval (RPI)	Set the data update cycle (i.e., the packet interval) of each connection between the originator and target. The default setting is 50 ms (i.e., data is updated once every 50 ms). Set the RPI to between 1 and 10,000 ms in 1-ms increments.
Timeout Value	Set the time until a connection timeout is detected. The timeout value is set as a multiple of the packet interval (RPI) and can be set to 4, 8, 16, 32, 64, 128, 256, or 512 times the packet interval. The default setting is 4 times the packet interval (RPI).
Connection Name	Set a name for the connection. (32 single-byte characters max.)

2 After you make all of the settings, click the **OK** Button.

● Editing Settings for All Connections

You can edit the connection settings between the originator and all of the target devices selected in the Register Device List together in a table.

1 Click the **Connections** Tab, and then click the **Edit All** Button. The following Edit All Connections Dialog Box is displayed.



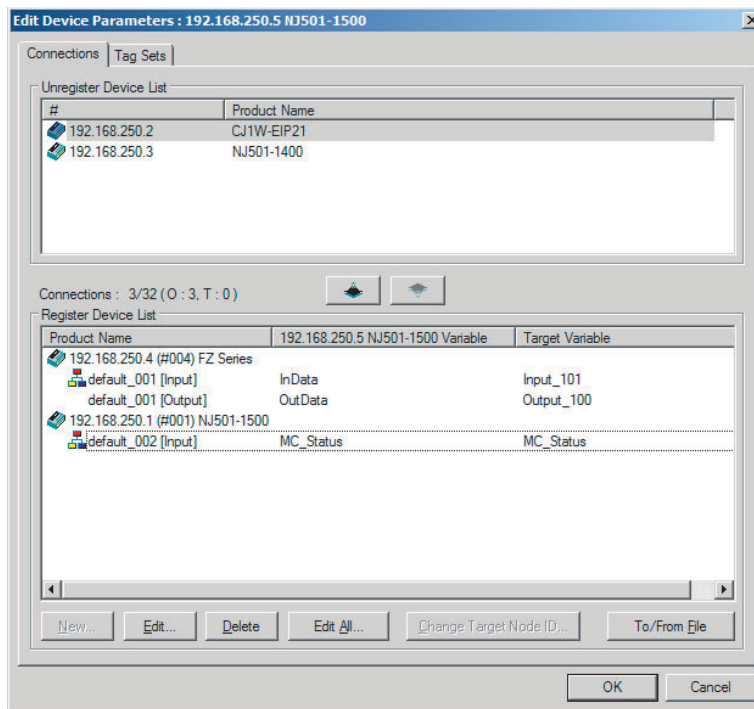
The settings are as follows:

Setting	Description
Target Device	Select the target device.
Connection Name	Any name can be given to the connection (32 single-byte characters max.). If this field is left blank, a default name is assigned. The connection name is used as a comment.
Connection I/O Type	Select InputOnly (tag type) to use tag data links with a CJ1W-EIP21(CJ2), CJ1W-EIP21(NJ), NX701, NJ501-□□□□, NJ301-□□□□, NJ101, or NY5□2. When you create tag data links for other devices, select the connection I/O type specified in that device's EDS file. Use the Input Only (ID type) setting when another company's node is the originator and does not support connection settings with a Tag type setting.
In/Out	The connections I/O is automatically displayed based on the selected connection. <ul style="list-style-type: none"> • Input Only: Just <i>In</i> is displayed.
Target Variable	Select the target node's tag set to assign it. <ul style="list-style-type: none"> • In: Select the target's output (produce) tag set. • Out: Select the target's input (consume) tag set.
Originator Variable	Select the originator node's tag set to assign it. <ul style="list-style-type: none"> • In: Select the originator's input (consume) tag set. • Out: Select the originator's output (produce) tag set.
Connection Type	Select whether the data is sent in multi-cast or unicast (point-to-point) form. The default setting is multi-cast. <ul style="list-style-type: none"> • Multi-cast connection: Select when the same data is shared by multiple nodes. This setting is usually used. • Point-to-point connection: Select when the same data is not shared by multiple nodes. In a unicast connection, other nodes are not burdened with an unnecessary load. <p>Note Refer to <i>7-1-4 Overview of Operation</i> for details on using multi-cast and unicast connections, and counting the number of connections.</p>
RPI	Set the data update cycle (i.e., the packet interval) of each connection between the originator and target. The default setting is 50 ms (i.e., data is updated once every 50 ms). Set the RPI to between 1 and 20,000 ms in 1-ms increments.
Timeout Value	Set the time until a connection timeout is detected. The timeout value is set as a multiple of the packet interval (RPI) and can be set to 4, 8, 16, 32, 64, 128, 256, or 512 times the packet interval. The default setting is 4 times the packet interval (RPI).

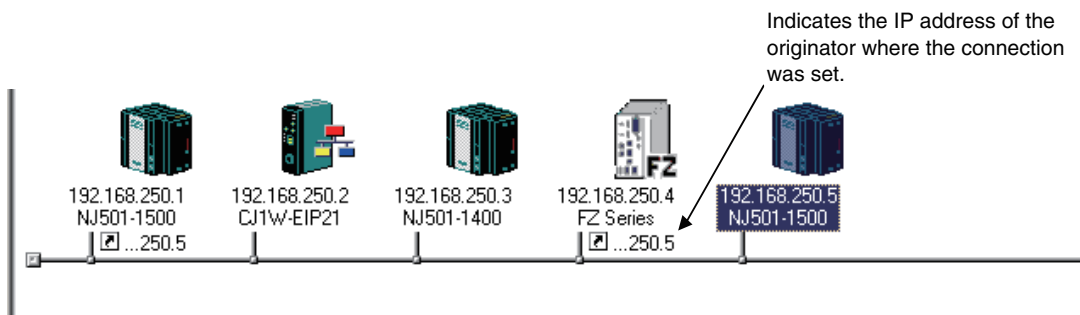
2 After you make all of the settings, click the **OK** Button.

● Confirming the Connections Settings

- 1 An overview of the connections that were set in the Register Device List is displayed in the Connections Tab Page.



- 2 Click the **OK** Button. The following kind of diagram is displayed.



- 3 Repeat the connections setting procedure until all of the connections are set.

Precautions for Correct Use

After you have made all of the settings, always click the **OK** Button before you close the Edit Device Parameters Dialog Box. If the **Cancel** Button is clicked and the dialog box is closed, the new settings are discarded.

- 4 If the tag set's size is changed in either the originator or target after the connection was set, the size will not match the other node and a parameter data mismatch will occur. In this case, if you change the connection settings, be sure to check the connections. (Refer to 7-2-16 *Checking Connections*.)

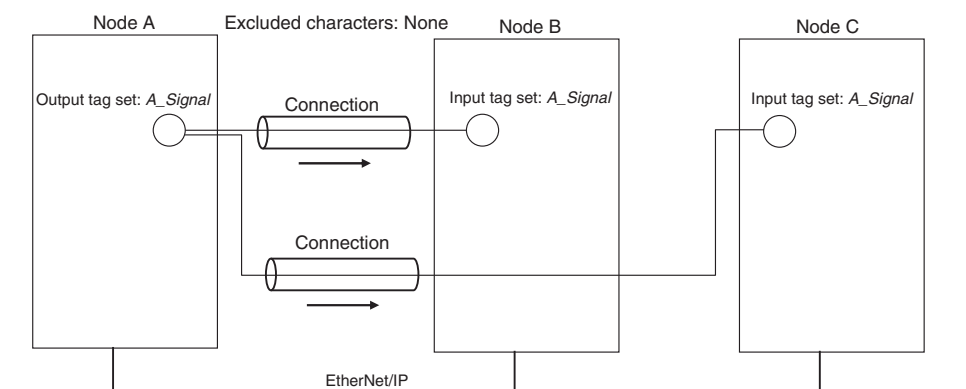
Automatically Setting Connections (Network - Auto Connection)

You can use automatic detection of the tag set names that are set for devices to automatically set connections between input and output tag sets with the same name (or the same names excluding specified ellipses). Connections are automatically set under the following conditions.

Output tag set names for connection setting	Except for specified ellipses, the output tag set name must be the same as the input tag set name. Ellipses can be set for the beginning or end of tag set names.
Input tag set names for connection settings	Except for specified ellipses, the input tag set name must be the same as the output tag set name. Ellipses can be set for the beginning or end of tag set names.
Connection type	The connection type must be Input Only. Multi-cast or single-cast connections can be specified for a connection.
RPI	The default setting is used.
Timeout	The default setting is used.

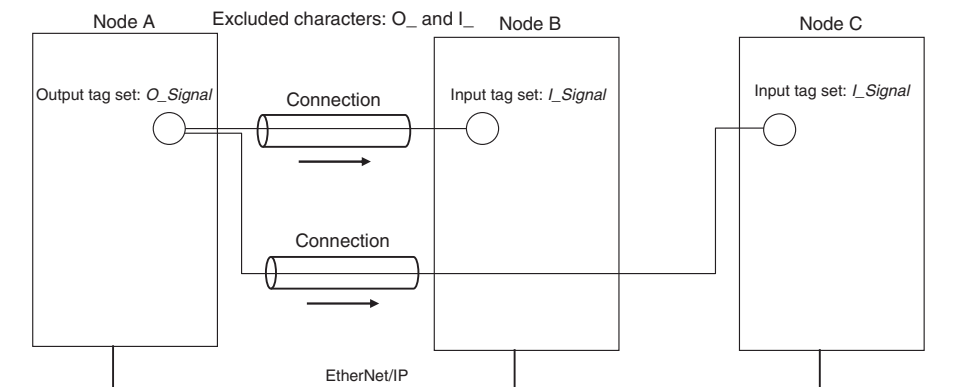
Example 1: Automatic Connections with the Same Tag Set Names

The following connections are automatically set with the same tag set name (*A_Signal*) if there is an output (produce) tag set named *A_Signal* at node A and input (consume) tag sets named *A_Signal* at nodes B and C.



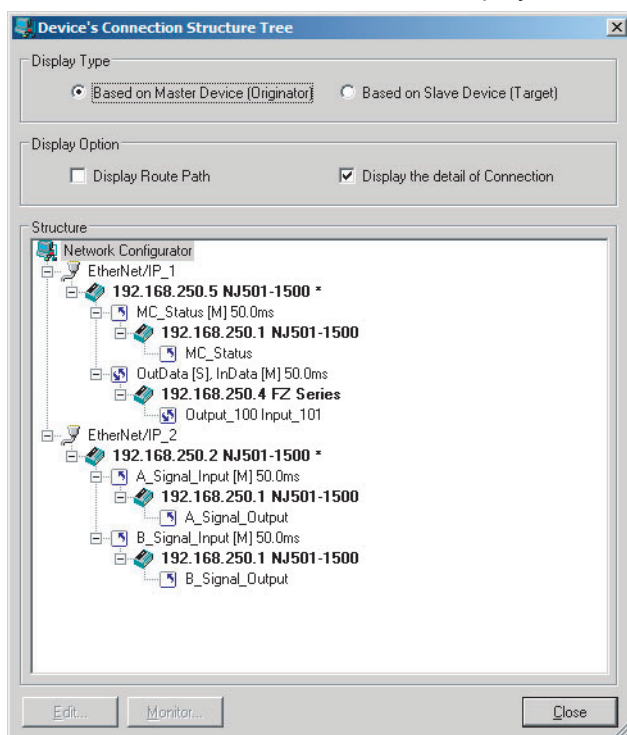
Example 2: Automatic Connections with the Ellipses

The following connections are automatically set with the same tag set name (*Signal*) if there is an output (produce) tag set named *O_Signal* at node A and input (consume) tag sets named *I_Signal* at nodes B and C, and "O_" and "I_" are set as forward ellipses.



- 1 Set the same tag set names for the output and input tag sets for the connection. The tag set names can also include forward and backward ellipses.

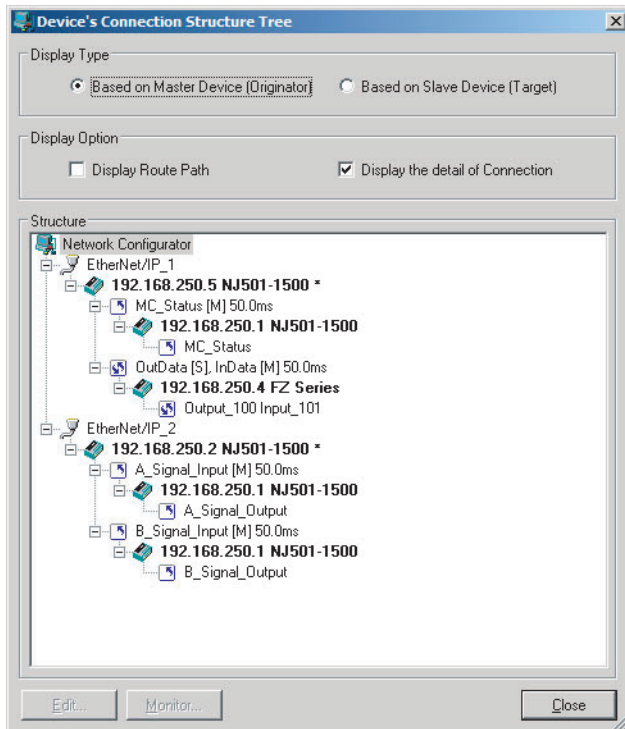
- 4** A device connection structure tree is displayed when processing is completed.



- 5** Use the device connection structure tree as required to change the RPI and timeout settings.

Device Connection Structure Tree

Connection settings can be displayed on the network configuration. Select **View Device's Connection Structure Tree** from the Network Menu.



- You can use the *Display the detail of Connection* Check Box to switch between device-level and connection-level displays of tag data link communications.
- An asterisk is displayed after the device name of the originator set for the connection.
- The Edit Device Parameters Dialog Box is displayed if you select a connection and click the **Edit** Button. You can edit the connections in this dialog box.

7-2-6 Creating Connections Using the Wizard

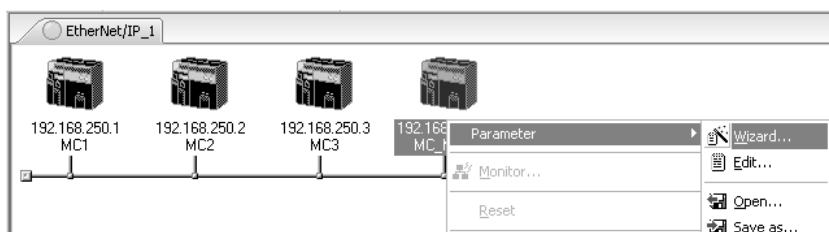
You can use the Network Configurator's Wizard to easily create connections between OMRON PLCs following the instructions provided by the Wizard.

Note The Wizard can be used only with the following OMRON EtherNet/IP devices.

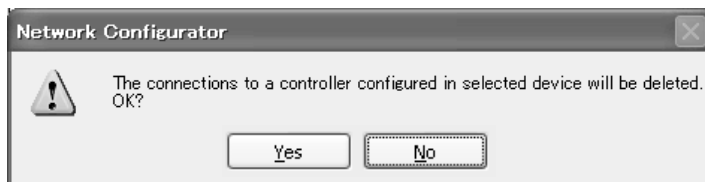
Device name	Remarks
CJ1W-EIP21(NJ)	CJ1W-EIP21 mounted to NJ-series Controller
CJ1W-EIP21	CJ1W-EIP21 mounted to CJ1 CPU Unit
CJ1W-EIP21(CJ2)	CJ1W-EIP21 mounted to CJ2 CPU Unit
CJ2B-EIP21	Built-in EtherNet/IP port in CJ2H CPU Unit
CJ2M-EIP21	Built-in EtherNet/IP port in CJ2M CPU Unit
CS1W-EIP21	CJ1W-EIP21 mounted to CS1 CPU Unit
NX701	Built-in EtherNet/IP port on NX-series CPU Unit
NJ301-□□□□ NJ501-□□□□ NJ101	Built-in EtherNet/IP port on NJ-series CPU Unit
NY5□2	Built-in EtherNet/IP port on NY-series Controller

Use the following procedure to create connections (i.e., data links) with the Wizard.

- 1** Set tags and tag sets for all devices before starting the Wizard. Refer to 7-2-4 *Creating Tags and Tag Sets* for the setting procedure.
- 2** For tag data links between OMRON PLCs, a connection is created in the PLC (i.e., the originator device) that receives data as input data.
First, select the registered device for which you want to create a connection in the Network Configuration Window of the Network Configurator, and then select **Device - Parameters - Wizard** from the menus.

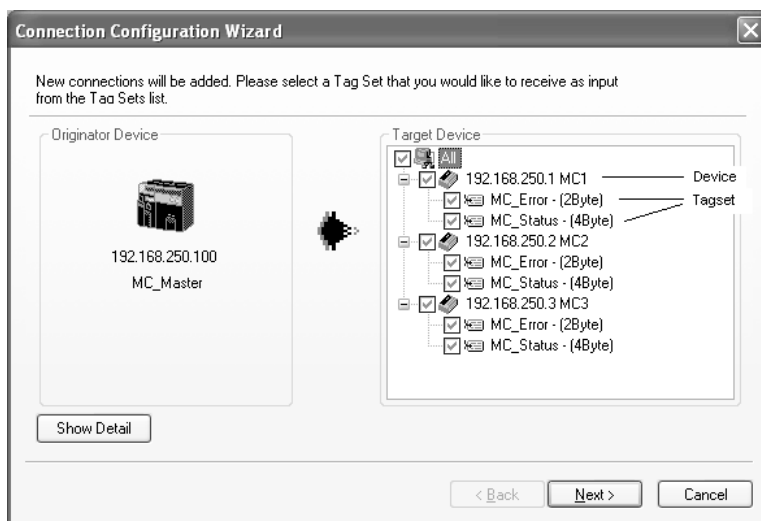


The following dialog box will be displayed before the Wizard starts.



Click the **Yes** Button to delete the connections that have been set with OMRON PLCs before starting the Wizard.

- 3 Create the connection following the instructions that are given by the Wizard after the Wizard starts. (See the following figure.)

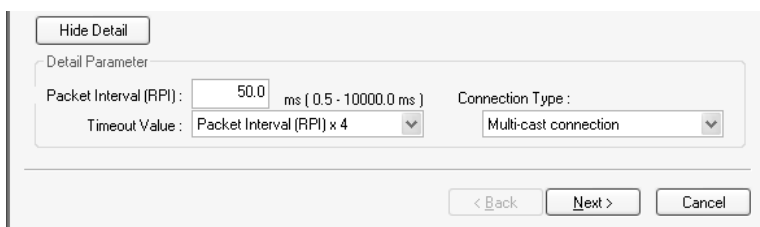


- 4 A list of tag sets is displayed on the right side of the Wizard Dialog Box with target devices that support receiving input data. Select the tag sets that you want to receive at the originator device. The following tables describes the meanings of the icons and check marks displayed in the tag set list.

Icon	Display position	Status
<input checked="" type="checkbox"/>	All	All output tag sets for all devices are selected.
	Device	All output tag sets for the applicable device are selected.
	Tag set	The applicable output tag sets are selected. These are the tag sets that will be set in the connection.
<input checked="" type="checkbox"/>	All	All or some output tag sets for some devices are selected.
	Device	Some output tag sets for applicable devices are selected.
<input type="checkbox"/>	All	All output tag sets for all devices are not selected.
	Device	All output tag sets for applicable devices are not selected.
	Tag set	The applicable output tag sets are not selected. The connections for this tag set will be deleted.
<input type="checkbox"/>	Device	No applicable tag sets.

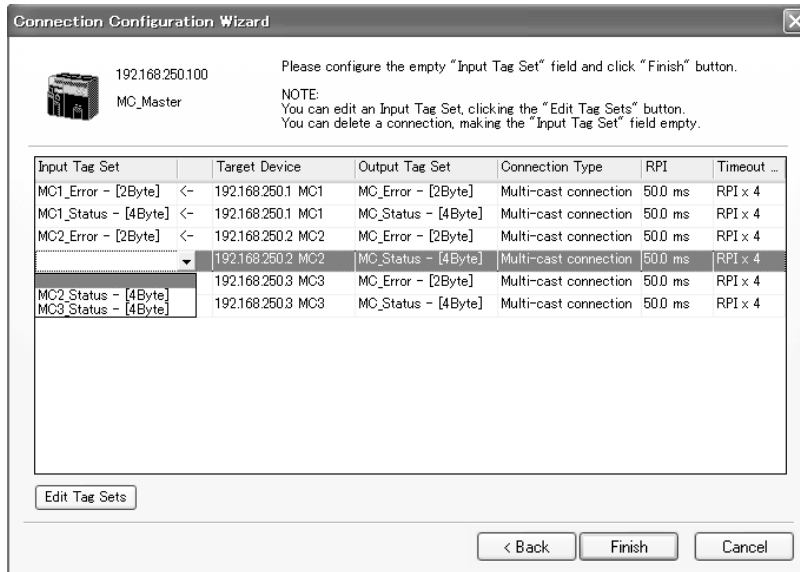
Note Tag sets that are used in connections that are already set are not displayed.

The following display will appear when you click the **Show Detail** Button.



The specified values for detailed parameters will be displayed. Change the values as required. The connection name cannot be set. They are automatically created using the following rule. default_N (where N is a 3-digit number (001, 002, etc.) starting from 1)

- 5** Click the **Next** Button to switch to the table in the following Wizard Dialog Box. Follow the instructions to select the input tag set of the originator device that receives the output tag set of the target device from the list box.

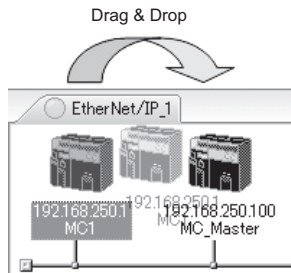


- The blank area in the Input Tag Set Column is the connection that you are creating.
 - The rows in which there are input tag sets are connections that are already set.
 - To prevent duplicate settings, input tag sets that have been used are not displayed in the list box for input tag sets.
 - If there is no applicable input tag set, you can edit a tag set or create a new one by using the **Edit Tag Sets** Button and **Edit Tag** Button.
- 6** Once the input tag set settings have been completed, click the **Finish** Button. You can check the set connection by selecting **Network - View Devices Connection Structure Tree** from the menus.
- The Wizard can be ended even if the input tag set includes a blank row. In that case, a connection is not created for the blank row.
 - You can delete a connection by deleting the input tag sets that were previously set.

7-2-7 Creating Connections by Device Dragging and Dropping

You can create a connection to the originator by dragging a target device and dropping it at the originator device. Network Configurator version 3.10 or higher is required to drag and drop devices to make connections.

Example: Drag the target device at 192.168.250.1 and drop it at the originator device at 192.168.250.100.



Additional Information

The EtherNet/IP originator device (i.e., a device in which connections can be set) must be one of the following OMRON EtherNet/IP devices.

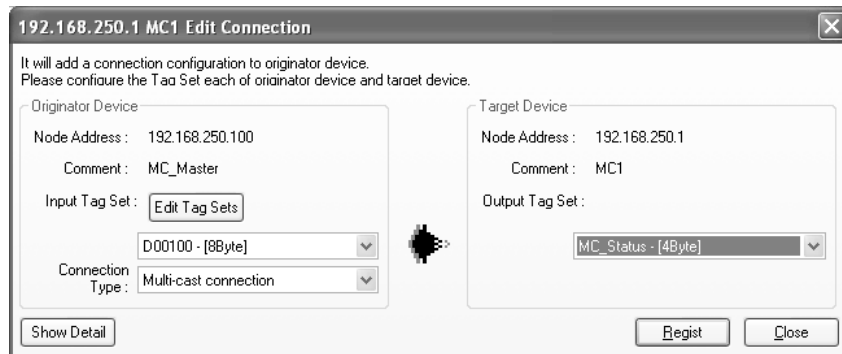
Device name	Remarks
CJ1W-EIP21(NJ)	CJ1W-EIP21 mounted to NJ-series CPU Unit
CJ1W-EIP21	CJ1W-EIP21 mounted to CJ1 CPU Unit
CJ1W-EIP21(CJ2)	CJ1W-EIP21 mounted to CJ2 CPU Unit
CJ2B-EIP21	Built-in EtherNet/IP port in CJ2H CPU Unit
CJ2M-EIP21	Built-in EtherNet/IP port in CJ2M CPU Unit
CS1W-EIP21	CJ1W-EIP21 mounted to CS1 CPU Unit
NX701	Built-in EtherNet/IP port on NX-series CPU Unit
NJ301-□□□□	Built-in EtherNet/IP port on NJ-series CPU Unit
NJ501-□□□□	
NJ101	
NY5□2	Built-in EtherNet/IP port on NY-series Controller

Use the following procedure to create connections (i.e., data links) by dragging and dropping devices.

- 1** Set the tags and tag sets for the target device that will be dragged.
 - (1) Refer to **7-2-4 Creating Tags and Tag Sets** for information on creating the settings if the target is one of the OMRON EtherNet/IP devices given above.
 - (2) If the target is another EtherNet/IP device, refer to the manual of that device and perform settings as required.

2 A dialog box as in the following figure for connection allocation will be displayed when you drag the target device and drop it at the OMRON EtherNet/IP device.

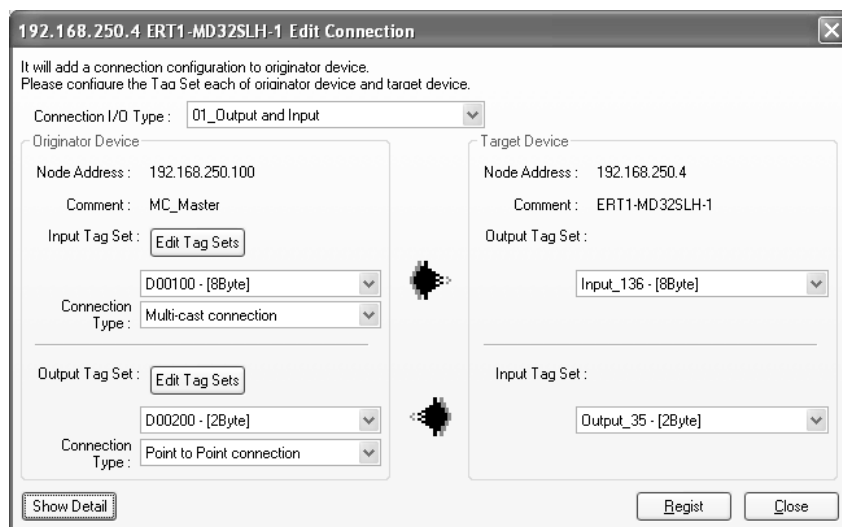
(1) Using One of the Above OMRON EtherNet/IP Devices As Target



Select the output tag set from Target Device Area on the right side of the Edit Connection Dialog Box, and then select the input tag set to receive the output tag set in the Originator Device Area on the left.

- If there is no applicable input tag set at the originator, you can create a new one by using the **Edit Tag Sets** Button and **Edit Tag** Button.

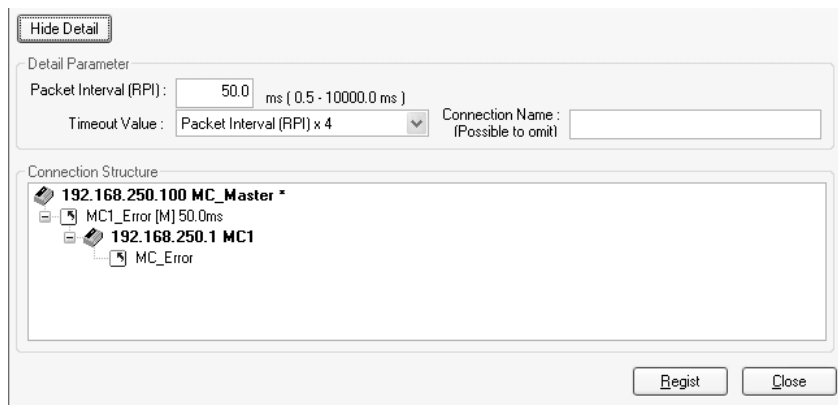
(2) Using Other EtherNet/IP Devices as Target



The connection I/O type list box in the upper part of the Connection Settings Dialog Box displays the connection I/O types that can be selected. Select the connection I/O type according to your application.

- The connection I/O types that can be selected depend on the target device.
- Items that can be selected will depend on the connection I/O type that is selected.
- Select the output, input, or both output and input tag sets at the target and specify the corresponding input, output, or both input and output tag sets at the originator.
- If there is no applicable tag set at the originator, you can create a new one by using the **Edit Tag Sets** Button and **Edit Tag** Button.

The following display will appear when you click the **Show Detail** Button.



The specified values for detailed parameters will be displayed. Change the values as required. Connection names are automatically created using the following rule.

default_N (where N is a 3-digit number (001, 002, etc.) starting from 1)



Additional Information

The following dialog box will be displayed if a target device that does not have I/O data is dropped.



Before dropping again, refer to the manual of the applicable device and create the I/O data (i.e., output tag sets) required to create a connection.

- 3 After you have set all of the connection, click the **Regist** Button to create the connection. When creating the connection has been completed, the input tag set and output tag set will be blank. Next, you can continue to create connections by selecting the connection I/O type and setting a tag set.

7-2-8 Connecting the Network Configurator to the Network

This section describes how to Connect the Network Configurator online.

Connecting through Ethernet

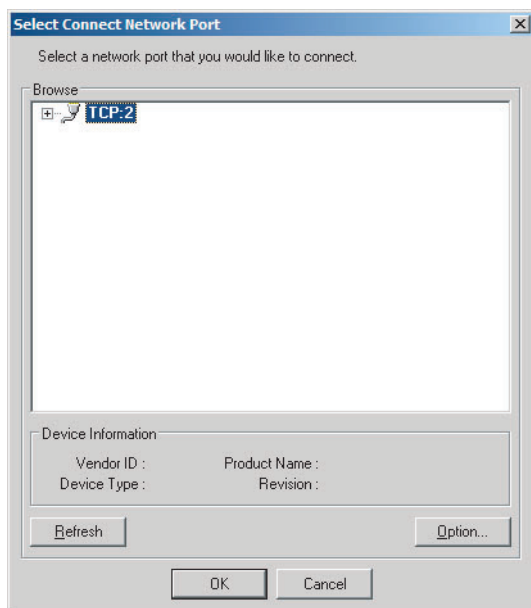
Connect to the built-in EtherNet/IP port on the Controller via an Ethernet switch.



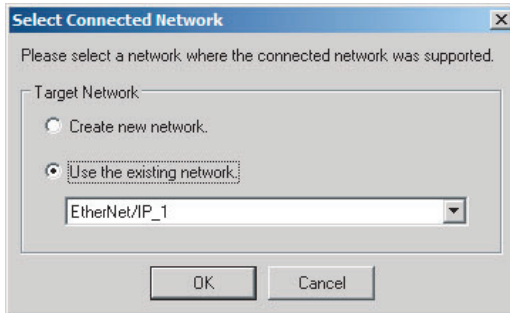
Precautions for Correct Use

The first time you connect via Ethernet with Windows XP (SP2 or higher), Windows Vista, or Windows 7, you must change the Windows firewall settings. For the procedure, refer to *A-4 Precautions for Using the Network Configurator on Windows XP, Windows Vista, or Windows 7 or Higher*.

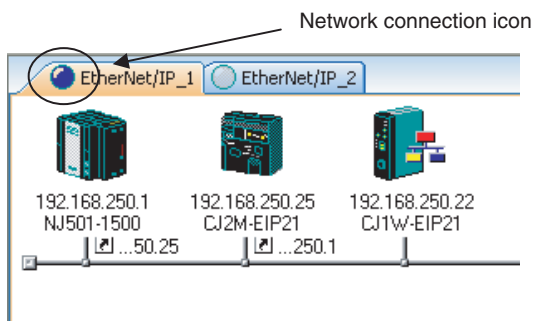
- 1 Select **Option - Select Interface - Ethernet I/F**.
- 2 Select **Network - Connect**. If there are multiple Ethernet interfaces on the computer, the Select Connect Network Port Dialog Box is displayed. Select the interface to connect, and press the **OK** Button.
The following dialog box is displayed.



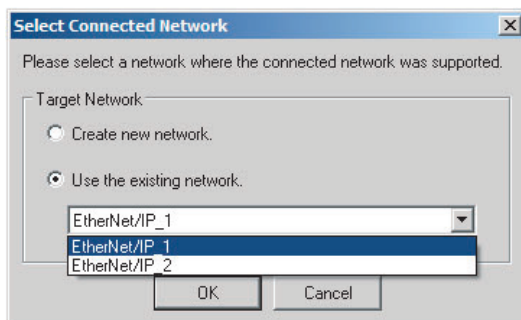
- 3** Click the **OK** Button.
Select the network to connect.



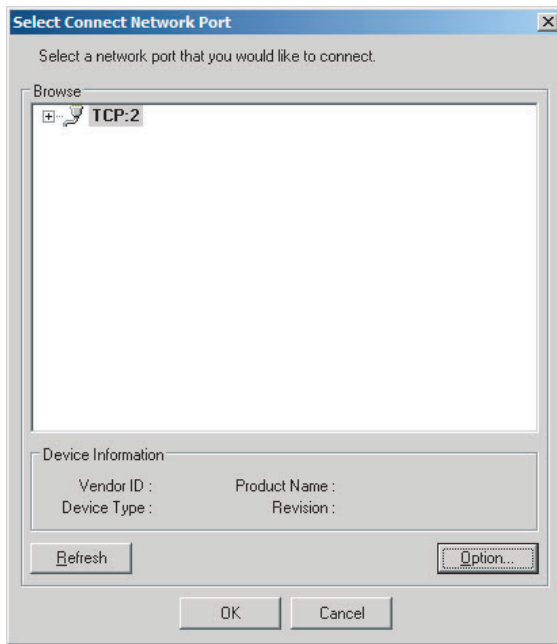
The Network Configurator will connect to the EtherNet/IP network. If the Network Configurator goes online normally, “Online” is displayed in the status bar at the bottom of the window. The network connection icon is displayed in blue on the Network Tab Page in which the Network Configurator is connected.



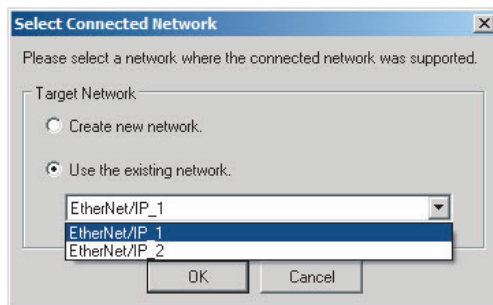
Select **Network - Change Connect Network** to switch the connected network.



4 The following dialog box is displayed.

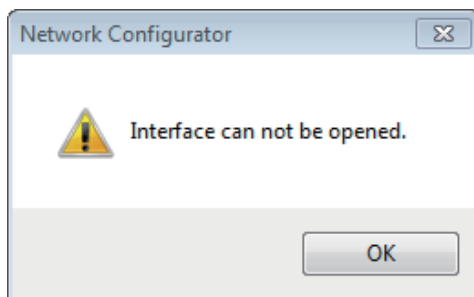


5 Click the **OK** Button.
Select the network to connect to.



Additional Information

If the following dialog box appears in the Network Configurator when you go online with an NY-series Controller, refer to the following table for possible causes and corrections.



Assumed cause	Correction
The cable is not connected correctly.	Check if the cable is disconnected or loose.
Connection with the Controller is blocked due to the firewall settings.	If connection with the Controller is blocked due to the firewall settings, disable the blocking. For the firewall settings, refer to <i>A-4 Precautions for Using the Network Configurator on Windows XP, Windows Vista, or Windows 7 or Higher.</i>

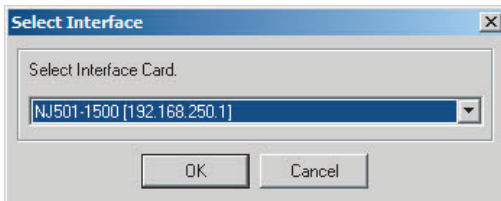
Direct Connection via Ethernet to Built-in EtherNet/IP Port

Use the following procedure to directly connect to the built-in EtherNet/IP port via Ethernet. You can connect to the built-in EtherNet/IP port even if the IP address has not been set on the computer.

- 1 Select the communications interface.
Select **Option - Select Interface - NJ/NX/NY Series Ethernet Direct I/F.**

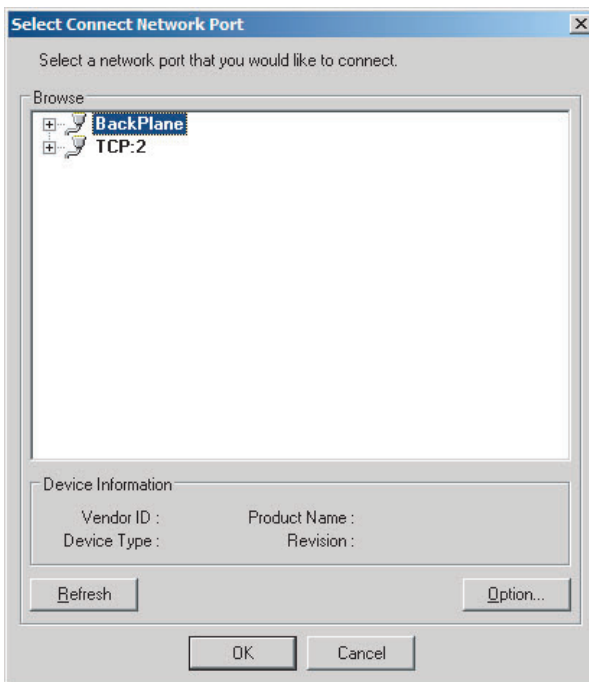
- 2 Select **Network - Connect** and click the **OK** Button.

The Setup Interface Dialog Box is displayed if there are several Controllers that you can connect to.



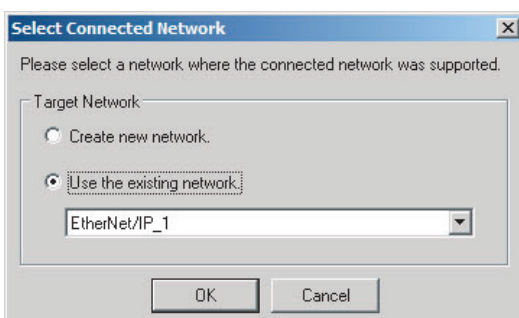
- 3 Select the Interface Card to connect and click the **OK** Button.

Select from the options displayed as **Controller model (IP number)**. The following dialog box is displayed.



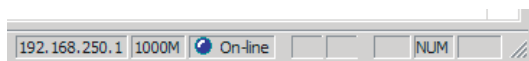
- 4 Select **TCP:2** and then click the **OK** Button.

The following dialog box is displayed.



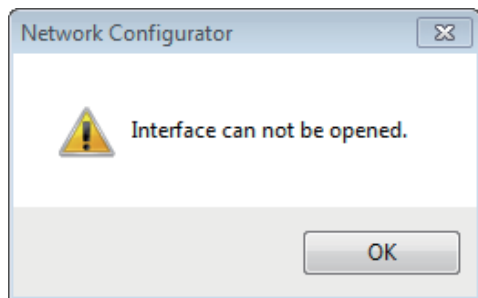
5 Select the network to connect.

The Network Configurator will connect to the EtherNet/IP network. If the Network Configurator goes online normally, “On-line” is displayed in the status bar at the bottom of the window.



Additional Information

If the following dialog box appears in the Network Configurator when you go online with an NY-series Controller, refer to the following table for possible causes and corrections.



Assumed cause	Correction
The cable is not connected correctly.	Check if the cable is disconnected or loose.
Connection with the Controller is blocked due to the firewall settings.	If connection with the Controller is blocked due to the firewall settings, disable the blocking. For the firewall settings, refer to <i>A-4 Precautions for Using the Network Configurator on Windows XP, Windows Vista, or Windows 7 or Higher.</i>

7-2-9 Downloading Tag Data Link Parameters

To make tag data links, you must download tag data link parameters, such as tag set settings and connection settings, to all devices in the EtherNet/IP network. When the download operation is executed, the tag data link parameters are transferred to the EtherNet/IP Units that require the settings.

The following procedure shows how to download the tag data link parameters. Refer to 7-2-8 *Connecting the Network Configurator to the Network* for information on how to Connect the Network Configurator online.



Precautions for Correct Use

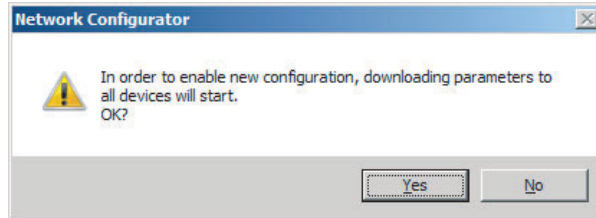
- If the node addresses (IP addresses) are not set correctly, you may connect to the wrong Controller and set incorrect device parameters. Download data only after you confirm that you are connected to the correct Controller.
- If incorrect tag data link parameters are set, it may cause equipment to operate unpredictably. Even when the correct tag data link parameters are set, make sure that there will be no effect on equipment before you transfer the data.
- When network variables are used in tag settings, a connection error will result if the variables are not also set in the Controller. Before downloading the tag data link parameters, check to confirm that the network variables are set in the Controller. Check whether the network variable, tag, and connection settings are correct. On the Connection and Tag Status Tab Pages described in 14-1-1 *Starting the Device Monitoring*.
- If a communications error occurs, the output status depends on the specifications of the device being used. When a communications error occurs for a device that is used along with output devices, check the operating specifications and implement safety countermeasures.
- The built-in EtherNet/IP port is automatically restarted after the parameters are downloaded. This restart is required to enable the tag set and connection information. Before you download the parameters, check to confirm that problems will not occur with the equipment when the port is restarted.
- Make sure that the major CIP revision of the device registered with the Network Configurator is the same as the major CIP revision of the NY-series Controller that you use. If major CIP revisions are not the same, the parameters may not be downloaded. To determine whether downloading is possible, refer to 7-2-3 *Registering Devices*.
- Do not disconnect the Ethernet cable during the parameter download.
- Tag data links (data exchange) between relevant nodes is stopped during a download. Before you download data in RUN mode, make sure that it will not affect the controlled system. Also implement interlocks on data processing in ladder programming that uses tag data links when the tag data links are stopped or a tag data link error occurs.

- 1 Connect the Network Configurator online.
- 2 There are two ways to download the parameters.

Downloading to All Devices in the Network

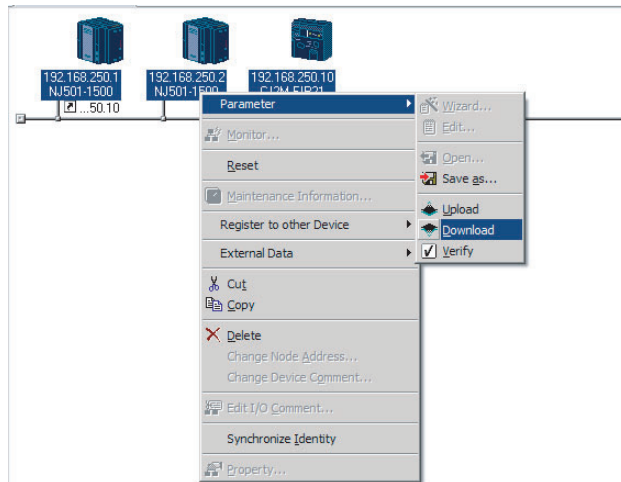
Select **Network - Download**.

The following dialog box is displayed.

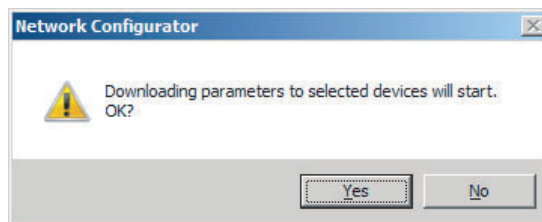


Downloading Individually to Particular Devices

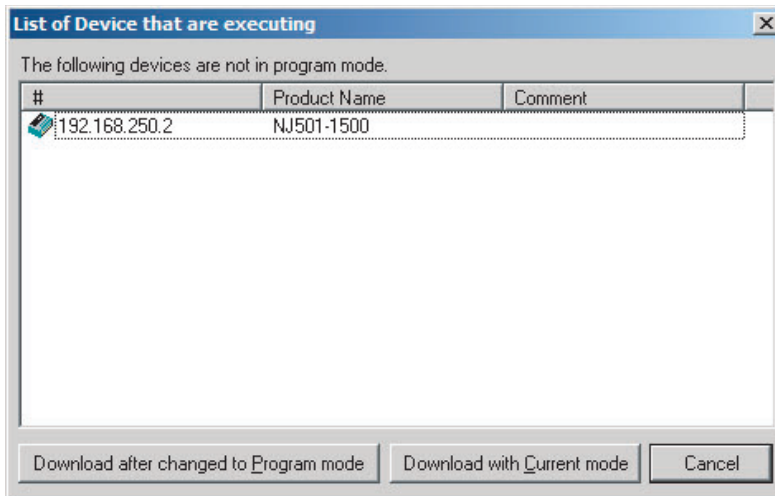
Select the icon of the EtherNet/IP Unit to which you want to download. To select multiple nodes, hold down the **Shift** Key or the **Ctrl** Key while you click the icons. (In the following example, 2 nodes are selected: 192.168.250.1 and 192.168.250.2.) Right-click the icon to display the pop-up menu, and select **Parameter - Download**.



The following dialog box is displayed.



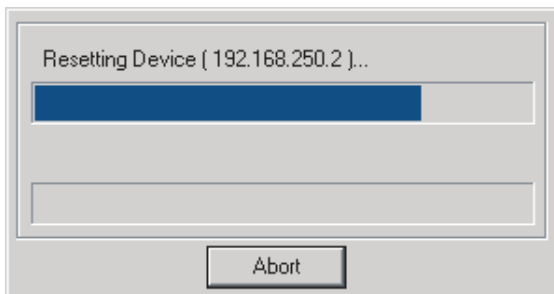
- 3 Click the **Yes** Button to download the tag data link parameters to the EtherNet/ IP Unit. The following dialog box is displayed if any of the Controllers is not in PROGRAM mode.



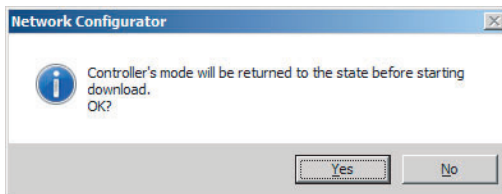
If the **Download after changed to Program mode** Button is clicked, all Controllers are changed to PROGRAM mode and the parameters are downloaded. Confirm safety for all controlled equipment before you change the Controllers to PROGRAM mode. You can restore the operating modes after the parameters are downloaded.

You can click the **Download with Current mode** Button to download the parameters even when one or more Controllers is in RUN mode. The **Download with Current mode** Button is disabled if the EtherNet/IP Unit does not support this function (e.g., revision 1 of CJ1W-EIP21 or CS1W-EIP21).

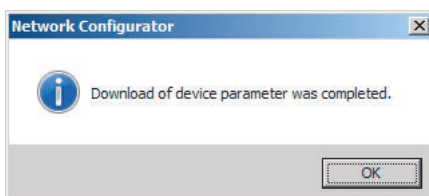
During the download, the following progress monitor is displayed to show the progress of the download.



If the operating mode of one or more Controllers was changed to download the parameters, you can return the Controllers to the previous operating modes. If the **No** Button is clicked, the Controllers remain in PROGRAM mode.



- 4 The following dialog box is displayed to show that the download was completed.



7-2-10 Uploading Tag Data Link Parameters

You can upload tag data link parameters (such as the tag set settings and connection settings) from EtherNet/IP Units in the EtherNet/IP network. The following procedure shows how to upload the parameters. For details on how to connect to the network from the Network Configurator, refer to *7-2-8 Connecting the Network Configurator to the Network*.



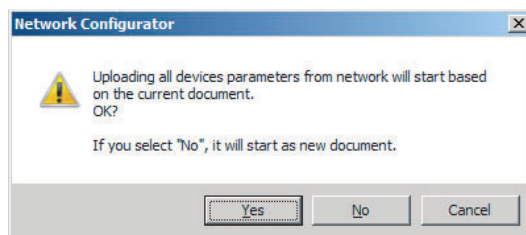
Precautions for Correct Use

Make sure that the major CIP revision of the device registered with the Network Configurator is the same as the major CIP revision of the Controller that you use. If the major CIP revisions are not the same, the parameters may not be uploaded. To determine whether uploading is possible, refer to *7-2-3 Registering Devices*.

There are two ways to upload the parameters.

Uploading from All Devices in the Network

- 1 Connect the Network Configurator online, and then select **Upload** from the Network Menu.
- 2 The following dialog box is displayed.



Clicking the Yes Button:

The tag data link parameters in the current project are uploaded.

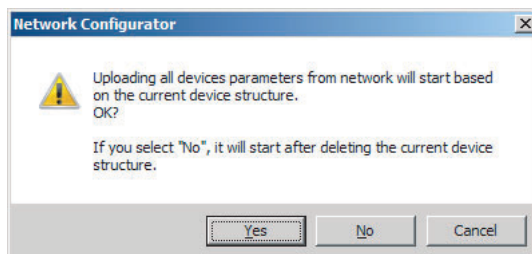
Clicking the No Button:

You open a new project to upload the tag data link parameters. The current project is closed.

Clicking the Cancel Button:

The upload operation is cancelled. The upload is not performed.

- 3 If you click the **Yes** Button in step 2, the following dialog box is displayed.



Clicking the Yes Button:

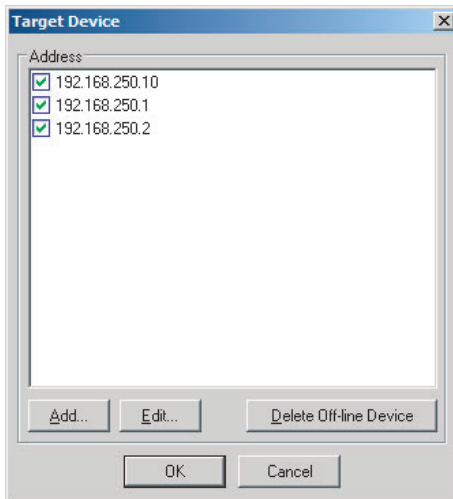
Parameters are uploaded only from the devices registered in the Network Configuration Pane. Parameters are not uploaded from devices that are not registered in the Network Configuration Pane.

Clicking the No Button:

Performing a Batch Upload over the Network

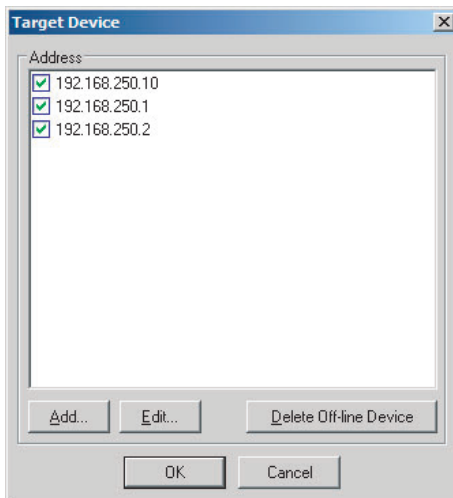
Parameters are uploaded from all devices on the network. The current Network Configuration Information will be lost.

The following dialog box will be displayed. Select the devices for which to upload parameters and click the **OK** Button.

**Clicking the Cancel Button:**

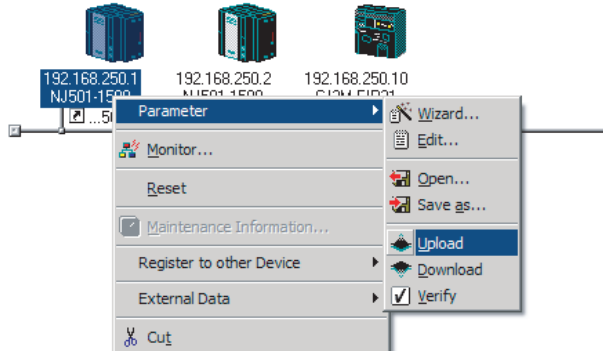
The upload operation is cancelled. The upload is not performed.

- 4** If you click the **No** Button in step 2, the following dialog box is displayed. Select the devices for which to upload parameters and click the **OK** Button.

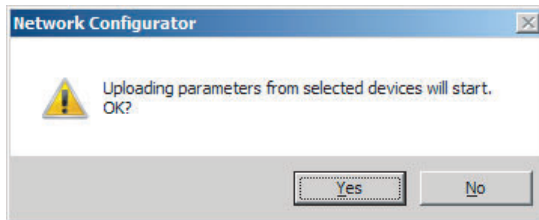


Uploading Individually from Particular Devices

- 1 Connect the Network Configurator online and select the icon of the EtherNet/IP Unit from which you want to upload the parameters. To select multiple nodes, press and hold the **Shift** Key or the **Ctrl** Key while you select additional icons. (In the following example, 2 nodes are selected: 192.168.250.1 and 192.168.250.2.)
Right-click the icon to display the pop-up menu, and select **Parameter - Upload**.

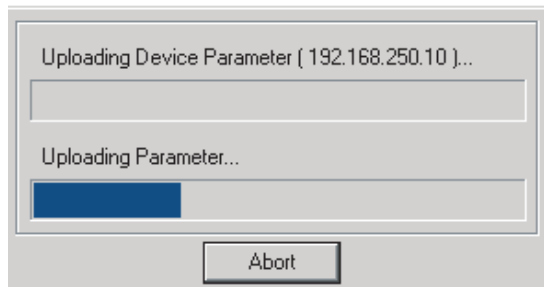


- 2 The following dialog box is displayed.

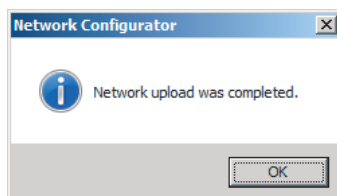


Click the **Yes** Button or the **No** Button.

- 3 During the upload, the following progress monitor is displayed to show the progress of the upload.



- 4 The following dialog box is displayed to show that the upload was completed.



7-2-11 Verifying the Tag Data Links

Tag data link parameters (such as the tag set settings and connection settings) can be compared with the parameters of the built-in EtherNet/IP ports in the EtherNet/IP network. The following procedure shows how to compare the parameters. For details on how to connect to the network from the Network Configurator, refer to *7-2-8 Connecting the Network Configurator to the Network*.



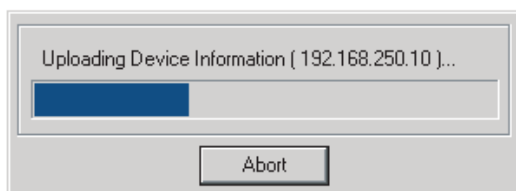
Precautions for Correct Use

Make sure that the major CIP revision of the device registered with the Network Configurator is the same as the major CIP revision of the Controller that you use. If the major CIP revisions are not the same, the parameters may not be compared. To determine whether comparison is possible, refer to *7-2-3 Registering Devices*.

Verifying the Network Configuration

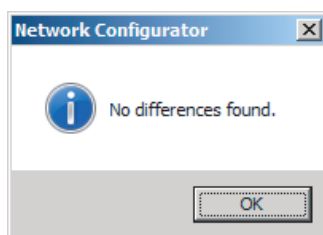
You can use the following procedure to compare the list of registered devices in the Network Configuration Pane with the devices connected on the EtherNet/IP network, and check the IP addresses and device types. This function does not verify device parameters.

- 1 Connect the Network Configurator online.
- 2 Select **Network - Verify Structure**.
The following progress monitor is displayed to show the progress as data is read from the network and compared.

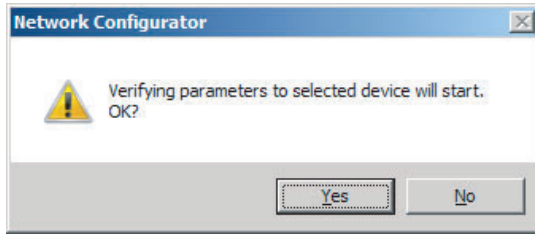


- 3 The results of the comparison between the network configuration file and data from the network are displayed as shown below.

Differences Not Found in the Comparison



- 5** If multiple nodes have been selected and compared, the following message is displayed. Click the **Yes** Button.



The comparison results are displayed in order of the selected nodes.

7-2-12 Starting and Stopping Tag Data Links

Automatically Starting Tag Data Links

Tag data links are automatically started immediately after the data link parameters are downloaded from the Network Configurator. (They are automatically started after the CPU Unit's power is turned ON or the Unit is restarted.)



Additional Information

A Tag Data Link Connection Timeout error will occur if a connection is not established with the target device within 1 minute after the tag data links are started in operation as the originator device. Reconnection processing is continued periodically even after this error occurs to automatically recover. If the application environment allows this error to be ignored, such as when a target device is started later than the originator device, you can change the event level to the observation level.

Starting and Stopping Tag Data Links for the Entire Network

You can start and stop tag data links for the entire network from the user program or from the Network Configurator.



Precautions for Correct Use

Use the same method (i.e., either the user program or the Network Configurator) to both start and stop tag data links. For example, if you use the `_EIP_TDLINKStopCmd` (Tag Data Link Communications Stop Switch) system-defined variable to stop tag data links, you cannot start them from the Network Configurator.

● Using Commands in the User Program

You can change the corresponding elements in the following system-defined variables to TRUE in the user program to start and stop tag data links for individual devices. (Refer to *Section 3 System-defined Variables Related to the Built-in EtherNet/IP Port*.)

- `_EIP_TDLINKStartCmd` (Tag Data Link Communications Start Switch)
- `_EIP_TDLINKStopCmd` (Tag Data Link Communications Stop Switch)



Additional Information

- Change the Tag Data Link Start Switch to TRUE, while the Tag Data Link Communications Stop Switch is FALSE. If the Tag Data Link Stop Switch is TRUE, the tag data links do not start even if the Tag Data Link Start Switch is changed to TRUE. Furthermore, if the Tag Data Link Start Switch and the Tag Data Link Stop Switch are both TRUE, an error occurs, the Multiple Switches TRUE Error system-defined variable changes to TRUE, and the event is recorded in the event log.
- After you start the tag data links, do not force the Tag Data Link Start Switch to change to FALSE from the user program or from the Sysmac Studio. It will change to FALSE automatically.

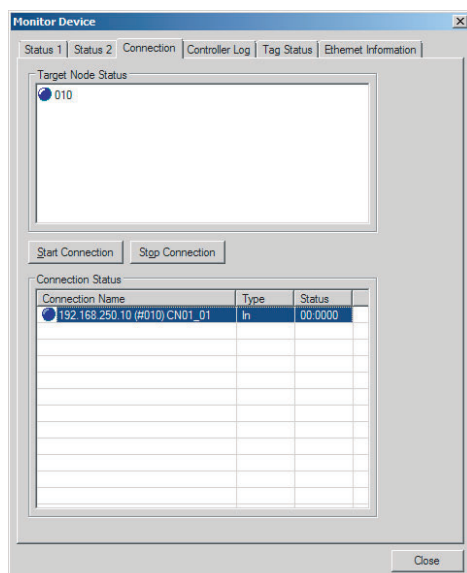
● Using the Network Configurator

You can select **I/O Connection - Start** or **I/O Connection - Stop** from the Network Menu to start and stop tag data links for individual devices.

Starting and Stopping Tag Data Links for Individual Devices

● Using the Network Configurator

You can start and stop tag data links for individual devices using the Connection Tab Page in the Monitor Device Dialog Box. This applies only to tag data links for which the device is the originator. Select **Monitor** from the Device Menu to access the Monitor Device Dialog Box. When using an NX-series CPU Unit, you can individually start and stop the tag data links of the built-in EtherNet/IP ports 1 and 2 connected to the Network Configurator.



Start Connection Button:

Starts all connections for which the device is the originator.

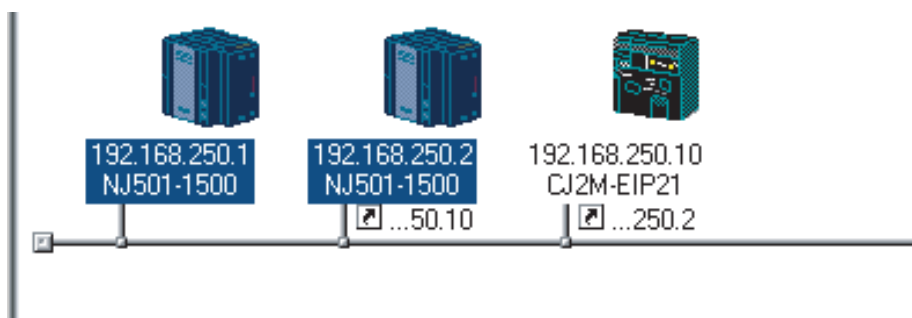
Stop Connection Button:

Stops all connections for which the device is the originator.

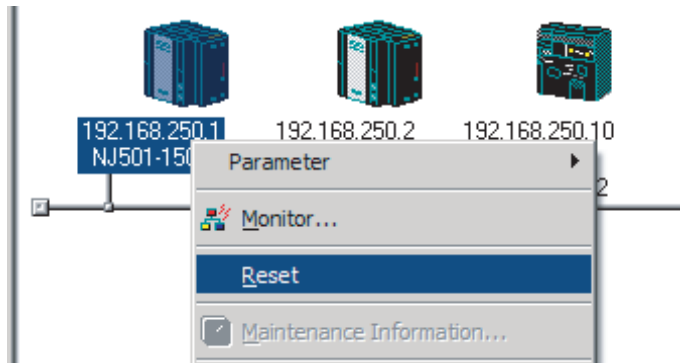
7-2-13 Clearing the Device Parameters

You can clear the tag data link settings (or return them to their factory settings) that are saved in the registered EtherNet/IP device. The following procedure shows how to clear the tag data link settings. For details on how to connect to the network from the Network Configurator, refer to *7-2-8 Connecting the Network Configurator to the Network*.

- 1** Connect the Network Configurator online.
- 2** Select the icon of the device from which you want to clear the device parameters. In the following example, 2 nodes are selected: 192.168.250.1 and 192.168.250.2. To select multiple nodes, press and hold the **Shift** Key while you select additional icons.



- 3** Select **Device - Reset**. You can also right-click the icon and select **Reset** from the popup menu.

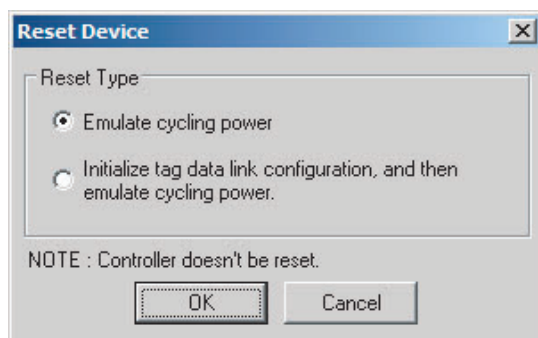


- 4** The following dialog box is displayed.



Yes Button:

The following dialog box is displayed.



Select the *Initialize tag data link configuration, and then emulate cycling power* Option and then click the **OK** Button.

 **Precautions for Correct Use**

The Controller is not restarted. Both the built-in EtherNet/IP port and internal port are reset.

No Button:

The tag data link settings will not be cleared and the built-in EtherNet/IP port and internal port will not be reset.



Additional Information

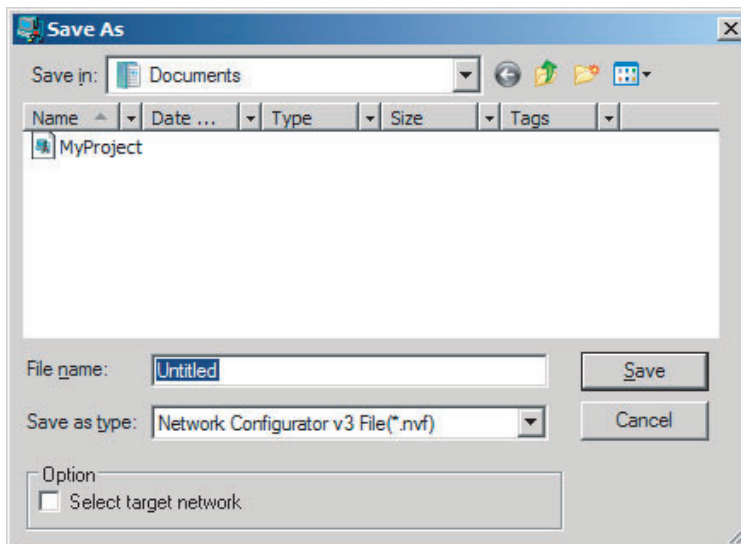
You can also execute the Reset service of the Identity object for the Controller to clear the tag data link settings. The procedures to execute the service from the Network Configurator is given below.

- (1) Connect the Network Configurator to the network.
- (2) Select **Tool - Setup Parameters** on the main window.
Then the dialog box for the general parameter settings are displayed.
- (3) Specify the target device and message to send.
 - Target Node Address: Enter the IP address of the target device.
 - Service: Select *Reset*.
 - Class: Enter *01*.
 - Instance: Enter *01*.
 - Attribute: Enter *00*.
 - Data: Enter *02*.
- (4) Click the **Send** Button.

7-2-14 Saving the Network Configuration File

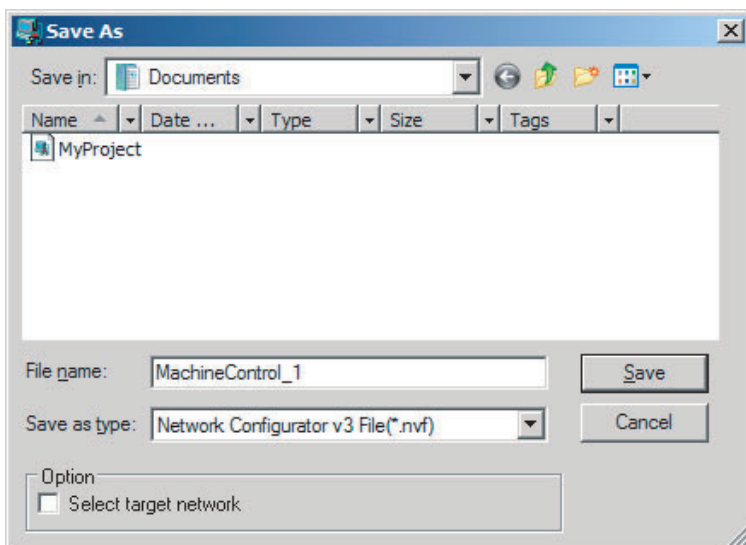
You can save device parameters set in the Network Configurator or device parameters uploaded from the network in a network configuration file.

- 1** Select **File - Save As**.
The following dialog box is displayed.




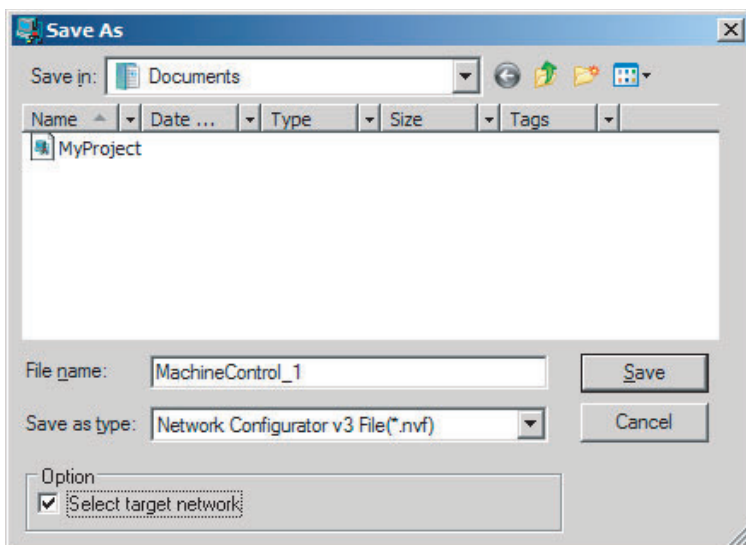
Untitled.nvf is displayed as the default file name.

- 2** Input the file name, and then click the **Save** Button.

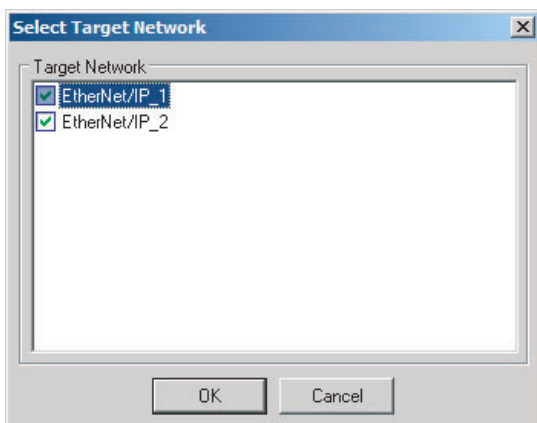


This completes the network configuration file save operation.

- 3** When the network configuration is changed later, you can overwrite the existing network configuration file if you select **File - Save** or click the  Button.
- 4** You can select the *Select target network* Check Box in the Option Area to save a network configuration file with only the required networks.




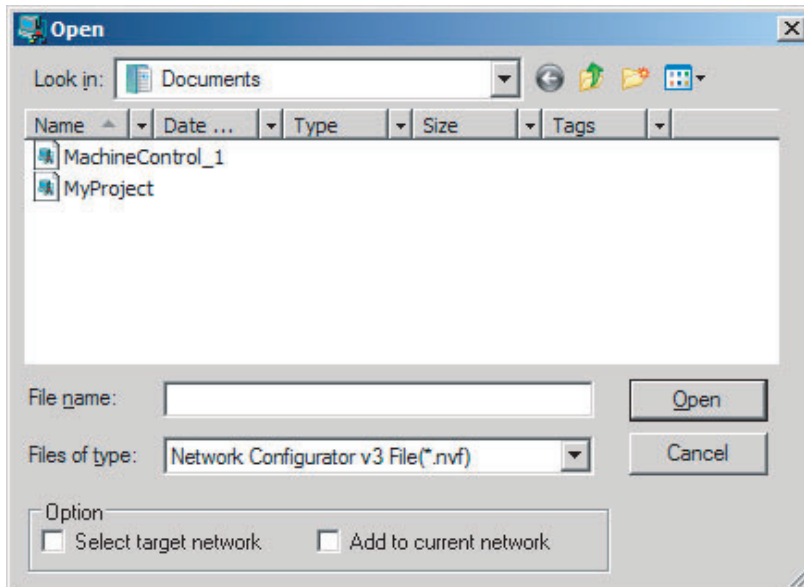
Select the check boxes of the networks to save and click the **OK** Button.



7-2-15 Reading a Network Configuration File

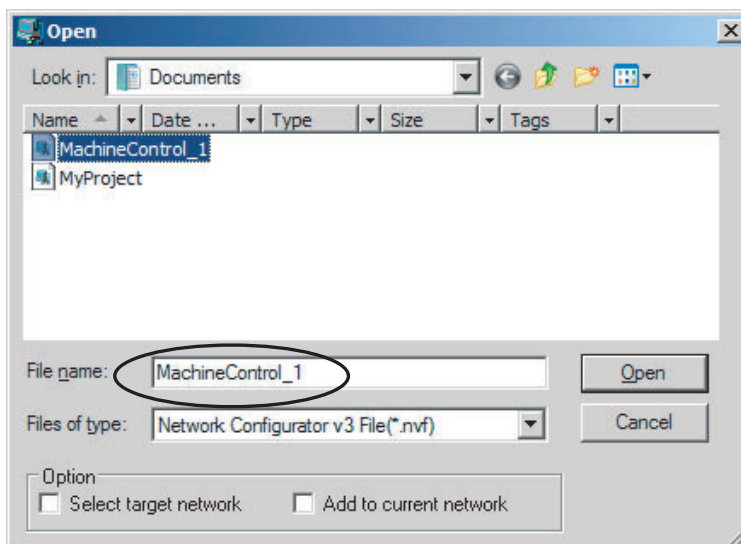
You can read a previously saved network configuration file into the Network Configurator.

- 1 Select **File - Open** or click the  Button.
The following dialog box is displayed.



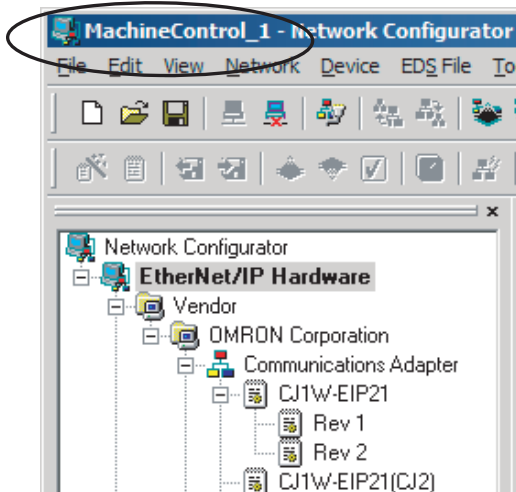
If the network configuration file that you want to read is not displayed, change to another folder.

- 2 If you select the network configuration file that you want to read, that file name is displayed in the *File name* Field.



- 3 Click the **Open** Button to read the network configuration file.

- 4 The Network Configurator's Title Bar will display the name of the file that was read.



- 5 Select any of the options as necessary. The options are listed below.

Setting	Description
Select target network	Allows you to select specific networks from the network configuration and open them.
Add to current document	Allows you to add the networks from the network configuration file that is currently open to the current configuration file.



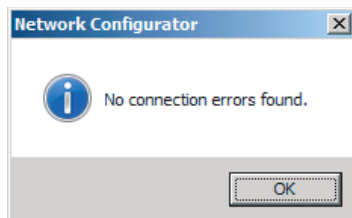
Additional Information

The save format will depend on the Network Configurator version. You can import configuration files (*.ncf) created with the Network Configurator for EtherNet/IP (version 2 or lower) if you select **External Data - Import** from the File Menu.

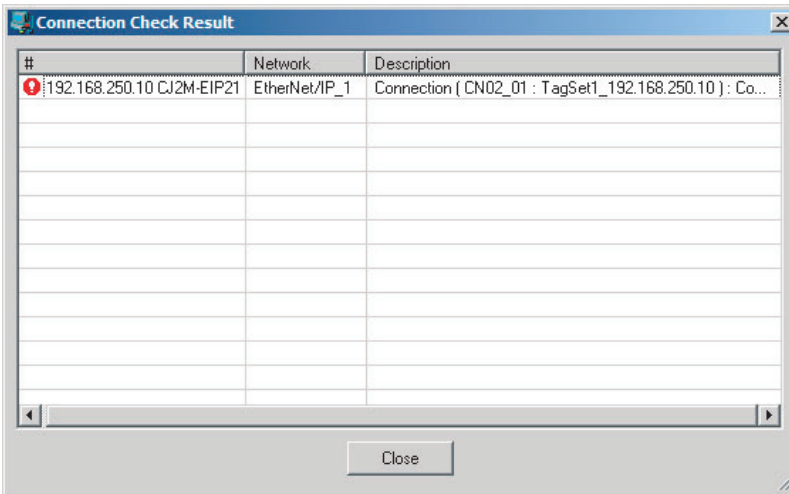
7-2-16 Checking Connections



You can check the consistency of connection parameters for network configuration files with device parameters that were set with the Network Configurator or device parameters uploaded from the network.

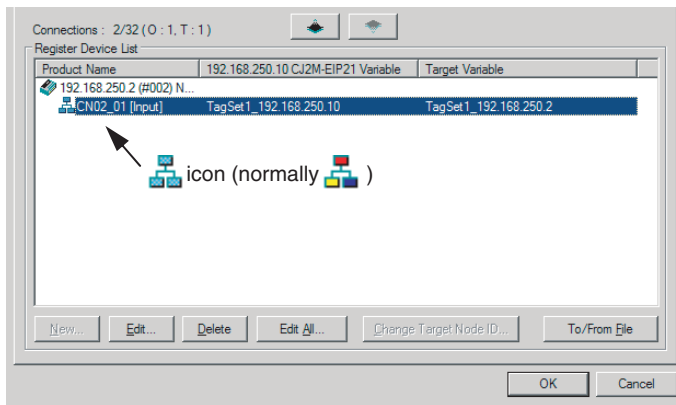
- 1 Select **Check Connection** from the Network Menu.
The following dialog box is displayed if parameters are normal.



The following dialog box is displayed if there are parameter errors. Check the displayed details and review the settings.



If an inconsistency is found, open the originator's Edit Device Parameter Dialog Box and click the **Connection** Tab. The inconsistent connection in the Register Device List is displayed with a  icon (instead of the normal  icon). To change the connection setting and select a different target variable, select the connection as shown below and click the **Edit** Button.



7-2-17 Changing Devices

You can change devices that are registered in a network configuration with the Network Configurator. Select **Change Device** from the Device Menu to display a list of the possible devices to change to. Select the desired device. You can change a device only when there is complete or upward compatibility with the device.

Device Changes

Model after change		CJ1W-EIP21(NJ)		NJ501-□□□□ NJ301-□□□□ NJ101		NX701	NY5□2
Model before change	CIP Rev	Rev2	Rev3	Rev1	Rev2	Rev2	Rev2
CJ1W-EIP21(NJ)	Rev2	---	Yes	*	*	Yes	*
	Rev3	Yes	---	*	*	Yes	*
NJ501-□□□□ NJ301-□□□□ NJ101	Rev1	Yes	Yes	---	Yes	No	No
	Rev2	Yes	Yes	Yes	---	Yes	Yes
NX701	Rev2	Yes	Yes	No	*	---	*
NY5□2	Rev2	Yes	Yes	No	*	Yes	---

Yes: Can be changed.

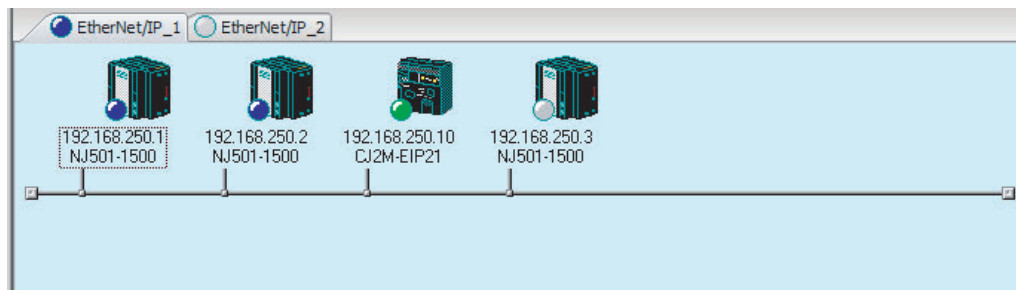
No: Cannot be changed.







* Cannot be changed in any of the following cases:

- The number of I/O connections, number of tags, number of tag sets, or size of one tag set exceeds the permissible settings for the device after the change.
- RPI exceeds the permissible settings or is set in 0.5-ms increments (such as 10.5 ms).

7-2-18 Displaying Device Status

Device status is displayed using the following icons in Maintenance Mode. To enter Maintenance Mode, select **Large Icons - Maintenance Mode** from the View Menu.



Icon	Status
 (white)	Offline
 (gray)	Default (including no Controller Configurations and Setup)
 (green)	Idle (including when CPU Unit of Controller is in PROGRAM mode)
 (blue)	Normal communications state (including when Controller is in RUN mode)
 (yellow)	Warning status (including when there is a partial fault or non-fatal error)
 (red)	Alarm status (including when there is a major fault or fatal error in the Controller)

7-3 Ladder Programming for Tag Data Links

7-3-1 Ladder Programming for Tag Data Links

If data in the ladder program is linked by tag data links, add conditions 1 to 3 in the ladder program for that data. If you want to use target node Controller information in the input conditions, add conditions 4 and 5.

- **Conditions to enable the NY-series Built-in EtherNet/IP Port's tag data links:**
 - (1) The following error bits in the Built-in EtherNet/IP Error variable (`_EIP_ErrSta`) are FALSE.
Major fault: Bit 7,
Partial fault: Bit 6,
and Minor fault: Bit 5
 - (2) Also, the Online variable (`_EIP_EtnOnlineSta`)*1 is TRUE.
- **Conditions showing that connections are established with the target device, and tag data links are operating:**
 - (3) The bit corresponding to the target node address in the Normal Target Node Information variable (`_EIP_EstbTargetSta`) is TRUE.
- **Condition for the Controller operating mode (operating or stopped) of the target node (valid for OMRON Controllers only):**
 - (4) The bit corresponding to the target node address in the Target PLC Operating Mode (`_EIP_TargetPLCModeSta`) is TRUE.
- **Condition for the Controller error status (fatal or non-fatal error) of the target node (valid for OMRON Controllers only):**
 - (5) The bit corresponding to the target node address in the Target PLC Error Information (`_EIP_TargetPLCErr`) is FALSE.

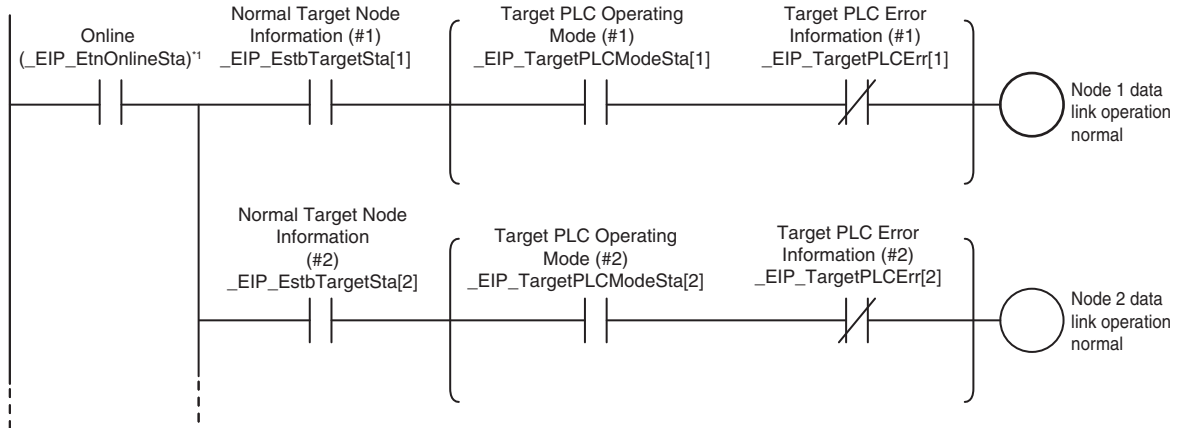
When you want to use the Target Node Controller Error Flag, the Controller status must be included in the tag sets for both the originator and target. Include the Controller status by using the Network Configurator to select the Include Options in the Edit Tag Set Dialog Boxes.

*1 Replace this with the following variable name, depending on which port you use.
Built-in EtherNet/IP port: `_EIP1_EtnOnlineSta`
Internal port: `_EIPIn1_EtnOnlineSta`

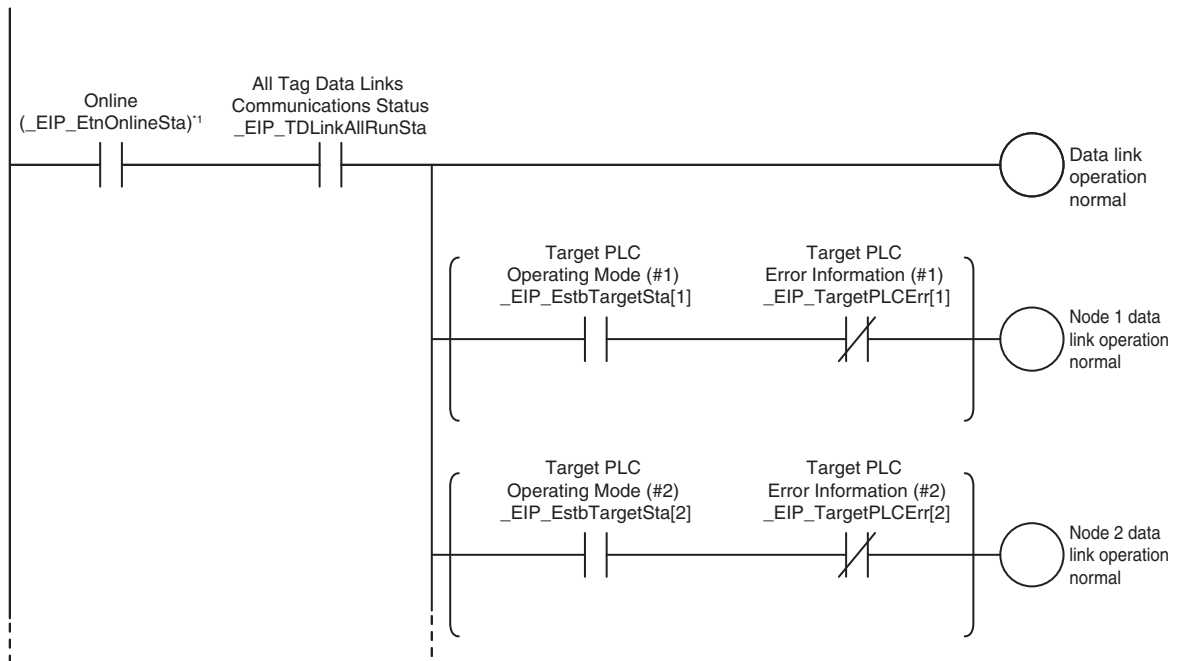
● **Programming Example to Detect Normal Operation**

The following programming can be used to confirm that normal communications are being performed for each target node. If the Controller status is included in the tag data, the status of the Controller can also be detected.

• **Programming Example 1 to Detect Normal Operation**



• **Programming Example 2 to Detect Normal Operation**



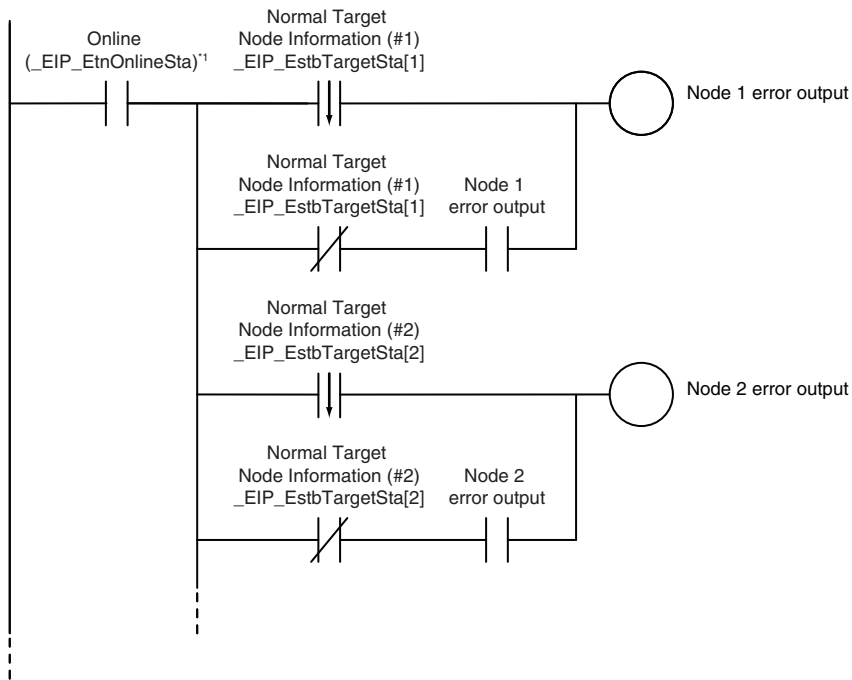
*1 Replace this with the following variable name, depending on which port you use.

Built-in EtherNet/IP port: *_EIP1_EtnOnlineSta*

Internal port: *_EIPIn1_EtnOnlineSta*

● **Programming Example to Detect Errors**

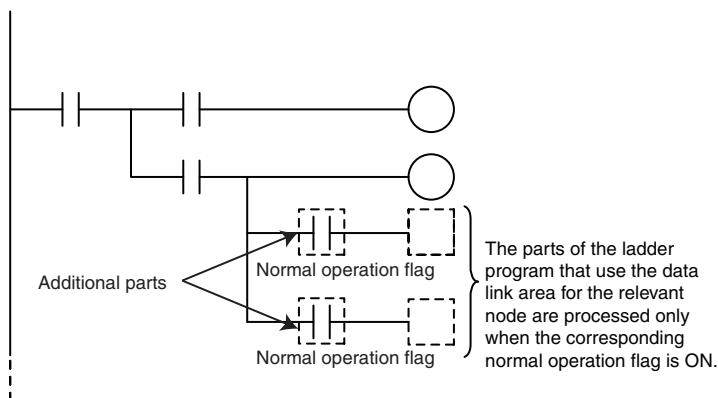
The following programming can be used to check for tag data link errors for each target node. This programming is used to detect errors only after the data links for all nodes have started normally.



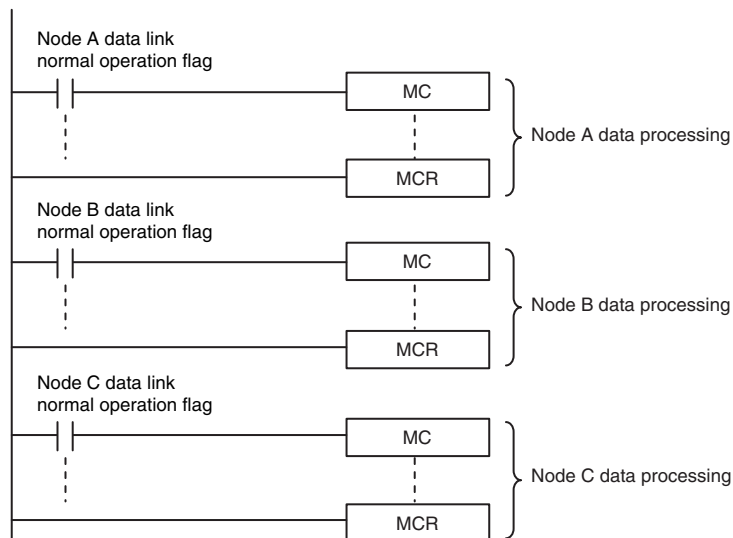
*1 Replace this with the following variable name, depending on which port you use.
 Built-in EtherNet/IP port: `_EIP1_EtnOnlineSta`
 Internal port: `_EIPIn1_EtnOnlineSta`

● **Data Processing Programming Example**

- The following type of programming can be used to process data only when the data links are operating normally.



- You can use MC - MCR instructions and JMP instructions to process data only when the data links are operating normally as shown below.



Precautions for Correct Use

Even if an error occurs in communications with a target device, the input data from the target device will remain stored in words allocated in memory to the local node. To prevent malfunctions, write the user program so that no input processing is performed when the following Built-in EtherNet/IP Error (*_EIP_ErrSta*) bits are TRUE.

- Major fault: Bit 7
- Partial fault: Bit 6
- Minor fault: Bit 5

7-3-2 Status Flags Related to Tag Data Links

The status of the tag data links is reflected in the following system-defined variables.

Variable name	Description
_EIP_TargetPLCModeSta[255] (Target PLC Operating Mode) * Corresponds to the Controller Operating Flag in the Controller status.	This variable shows the operating status of the target node Controllers that are connected with the built-in EtherNet/IP port as the originator. The information in this area is valid only when the corresponding Normal Target Node Information is TRUE. If the corresponding Normal Target Node Information is FALSE, the Target Node Controller Operating Information indicates the previous operating status. <i>Array[x]</i> is TRUE: The target Controller with a node address of x is in operating status. <i>Array[x]</i> is FALSE: Other than the above.
_EIP_TargetNodeErr[255] (Target Node Error Information) * Corresponds to the Controller Error Flag in the Controller status.	This variable indicates that the connection for the Registered Target Node Information was not established or that an error occurred in the target Controller. The information in this area is valid only when the Registered Target Node Information is TRUE. <i>Array[x]</i> is TRUE: The Registered Target Node Information for a node address of x is TRUE, and the Normal Target Node Information is FALSE or the Target PLC Error Information is TRUE. <i>Array[x]</i> is FALSE: When the Registered Target Node Information for a node address of x is FALSE, or when the Registered Target Node Information is TRUE, the Normal Target Node Error Information is TRUE, and the Target PLC Error Information is FALSE.
_EIP_EstbTargetSta[255] (Normal Target Node Information) * This status is not included in the Controller status.	This variable gives a list of nodes that have normally established EtherNet/IP connections. <i>Array[x]</i> is TRUE: The connection to the node with a node address of x is established normally. <i>Array[x]</i> is FALSE: A connection is not established or an error has occurred.

7-4 Tag Data Links with Other Models

The performance of tag data links depends on the Controller and EtherNet/IP Unit model as shown below. When you use tag data links between the built-in EtherNet/IP port on the NY-series Controller and another Controller or EtherNet/IP Unit, use tag data link settings of the Unit with the lower communications performance.

● Differences in Tag Data Link Performance Specifications

Item		NY-series Controller	NX-series Controller	NJ-series CPU Unit		CJ2M-CPU3□		CS1W-EIP21, CJ1W-EIP21, or CJ2H-CPU6□-EIP
				Unit version		2.0	2.1 or later	
				Unit version 1.00 to 1.02	Unit version 1.03 or later			
Tags	Total size of all tags	92,416 words	184,832 words (total of 369,664 words with two ports)	9,600 words		640 words		184,832 words
	Maximum size of 1 tag	722 words (721 words when the tag set includes the Controller status)		300 words (299 words when the tag set includes the Controller status)		20 words (19 words when the tag set includes the Controller status)	640 words (639 words when the tag set includes the Controller status)	722 words (721 words when the tag set includes the Controller status)
	Number of registrable tags	256*3	256 (total of 512 tags with two ports)*3	256*3		32		256
Tag sets	Maximum size of 1 tag set	722 words (721 words when the tag set includes the Controller status)		300 words (299 words when the tag set includes the Controller status)		20 words (19 words when the tag set includes the Controller status)	640 words (639 words when the tag set includes the Controller status)	722 words (721 words when the tag set includes the Controller status)
	Number of tags per tag set	8 (7 tags when the tag set includes the Controller status) Note Input and output variables cannot be combined.						
	Number of registrable tag sets	128	256 (total of 512 tags with two ports)	32		32		256
Connections	Number of connections	128	256 (total of 512 connections with two ports)	32		32		256
	Maximum data size per connection	722 words*2 (Data concurrency is maintained within each connection.)	722 words*2 (Data concurrency is maintained within each connection.)	300 words (Refer to 7-1-7 <i>Concurrency of Tag Data Link Data</i> for information on the conditions to maintain concurrency in the data for one connection.)		20 words	640 words	252 words or 722 words*2 (Data concurrency is maintained within each connection.)
Packet intervals (RPIs)		1 to 10,000 ms in 1-ms increments	0.5 to 10,000 ms in 0.5-ms increments	10 to 10,000 ms in 1-ms increments	1 to 10,000 ms in 1-ms increments	1 to 10,000 ms in 0.5-ms increments		0.5 to 10,000 ms in 0.5-ms increments
Communications bandwidth used (pps)*1		20,000 pps	40,000 pps	1,000 pps	3,000 pps	3,000 pps		12,000 pps

*1 Here, pps means "packets per second" and indicates the number of packets that can be processed in one second.

- *2 To use a data size of 505 bytes or more, the system must support a large forward open (an optional CIP specification). The SYSMAC CS/CJ-series Units support Large_Forward_Open, but before you connect to nodes of other companies, confirm that those devices also support it.
- *3 The maximum number of tags is given for the following conditions.
- All tag sets contain eight tags.
 - The maximum number of tag sets (32) is registered.

● Specifying Tags

You can specify where to assign a tag either with a variable or with a I/O memory address. However, some Controllers may not support both of these methods. Communications with the devices are possible regardless of whether the remote node tags are set using I/O memory addresses or network variables.

The supported tag specification methods for each Controller are listed in the table below.

Yes: Supported, No: Not supported

Controller	EtherNet/IP Unit	Name in Hardware List of Network Configurator	Network variable name specification	I/O memory address specification
NY-series Controller	---	NY512 NY532	Yes	No
NX-series Controller	---	NX701	Yes	No
NJ-series Controller	---	NJ501-□□□□ NJ301-□□□□ NJ101	Yes	Yes*
	CJ1W-EIP21	CJ1W-EIP21(NJ)	Yes	Yes*
CJ2H-CPU6□-EIP	---	CJ2B-EIP21	Yes	Yes
	CJ1W-EIP21	CJ1W-EIP21(CJ2)	Yes	Yes
CJ2H-CPU6□	CJ1W-EIP21	CJ1W-EIP21(CJ2)	No	Yes
CJ2M-CPU3□	---	CJ2M-EIP21	Yes	Yes
	CJ1W-EIP21	CJ1W-EIP21(CJ2)	Yes	Yes
CJ2M-CPU1□	CJ1W-EIP21	CJ1W-EIP21(CJ2)	No	Yes
CJ1 CPU Unit	CJ1W-EIP21	CJ1W-EIP21	No	Yes
CS1 CPU Unit	CS1W-EIP21	CS1W-EIP21	No	Yes

* To specify an I/O memory address for a tag, do not specify the I/O memory address for the tag directly. Instead, create a variable, set an AT specification of the I/O memory address on the Sysmac Studio, and then specify the variable with the AT specification for the tag.

8

CIP Message Communications

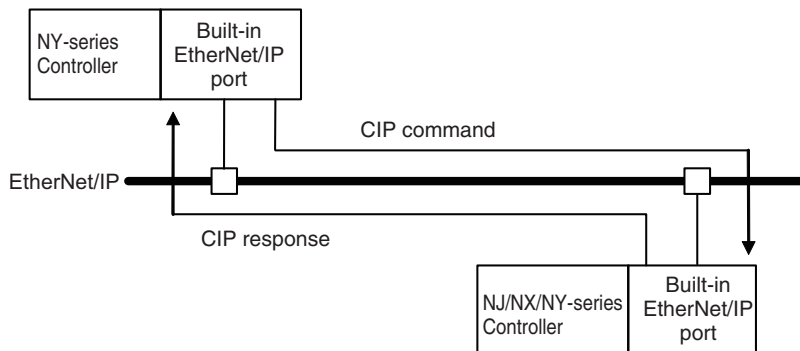
8-1	Overview of the CIP Message Communications Service	8-3
8-1-1	Overview of the CIP Message Communications Service	8-3
8-1-2	Message Communications Service Specifications	8-3
8-2	CIP Messaging Communications Client Function	8-4
8-2-1	Overview	8-4
8-2-2	CIP Communications Instructions	8-5
8-2-3	Using CIP Communications Instructions	8-6
8-2-4	Route Path	8-7
8-2-5	Request Path (IOI)	8-10
8-2-6	Service Data and Response Data	8-14
8-2-7	Sample Programming for CIP Connectionless (UCMM) Message Communications	8-15
8-2-8	Sample Programming for CIP Connection (Class 3) Message Communications	8-20
8-2-9	Operation Timing	8-27
8-2-10	Response Codes	8-28
8-3	CIP Communication Server Functions	8-32
8-3-1	CIP Message Structure for Accessing CIP Objects	8-33
8-3-2	CIP Message Structure for Accessing Variables	8-33
8-4	Specifying Request Path	8-34
8-4-1	Examples of CIP Object Specifications	8-34
8-4-2	Examples of Variable Specifications	8-34
8-4-3	Logical Segment	8-35
8-4-4	Data Segment	8-36
8-4-5	Specifying Variable Names in Request Paths	8-37

8-5 CIP Object Services	8-40
8-5-1 CIP Objects Sent to the Built-in EtherNet/IP Port	8-40
8-5-2 Identity Object (Class ID: 01 Hex)	8-40
8-5-3 TCP/IP Interface Object (Class ID: F5 hex)	8-44
8-5-4 Ethernet Link Object (Class ID: F6 Hex)	8-47
8-5-5 Controller Object (Class ID: C4 Hex)	8-52
8-6 Read and Write Services for Variables	8-53
8-6-1 Read Services for Variables	8-53
8-6-2 Write Service for Variables	8-55
8-7 Variable Data Types	8-57
8-7-1 Data Type Codes	8-57
8-7-2 Common Format	8-58
8-7-3 Elementary Data Types	8-58
8-7-4 Derived Data Types	8-60

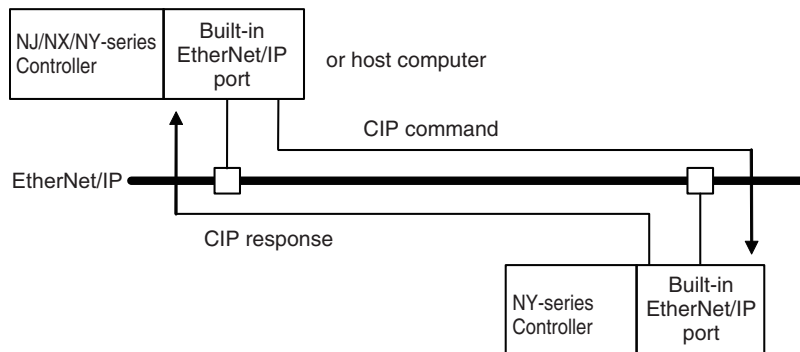
8-1 Overview of the CIP Message Communications Service

8-1-1 Overview of the CIP Message Communications Service

CIP commands can be sent to devices on the EtherNet/IP network whenever they are required. You execute CIP_SEND instructions in a program in the NY-series Controller to send CIP commands, such as those to read and write data and to receive the responses.



You can use CIP messages from the client to read and write memory in the Controller with the server without adding any special programming to the user program of the Controller with the server.



8-1-2 Message Communications Service Specifications

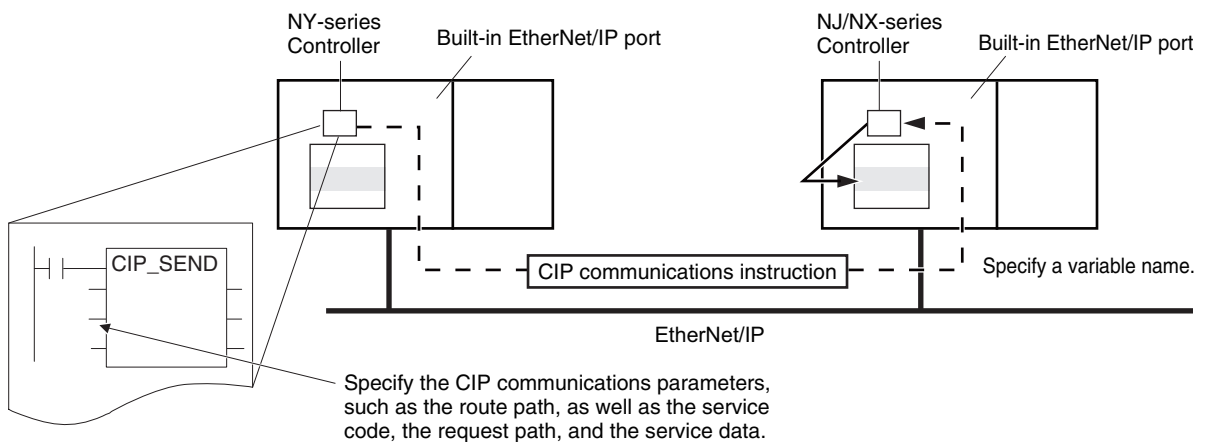
Item		Specification
Message type		Either of the following can be selected. CIP UCMM connectionless messages CIP class 3 connection messages
Execution method		CIPSend (Send Explicit Message Class 3) instruction or CIPUC-MMSSend (Send Explicit Message UCMM) instruction
Data contents		Sending required CIP commands and receiving responses
Communications parameters		Message type, timeout value, and root path specification
Maximum length per connection	Non-connection type (UCMM)	502 bytes
	Connection type (class 3)	<ul style="list-style-type: none"> Using Forward_Open 502 bytes Using Large_Forward_Open 8192 bytes

8-2 CIP Messaging Communications Client Function

8-2-1 Overview

The CIP Messaging Communications Client function is available for an NY-series Controller. This function allows you to request a service by sending CIP messages to an external device and specifying a self-contained object of a device supporting the CIP Communication Server functions.

The NY-series Controller executes CIP communications instructions in the user program and send CIP messages. In this way, you can use CIP messages to execute instructions such as reading and writing variables (data) of the NJ/NX-series Controller on the EtherNet/IP network.



8-2-2 CIP Communications Instructions

The following CIP communications instructions are available. For details on CIP communications instructions, refer to the *NY-series Instructions Reference Manual* (Cat. No. W560).

Instruction	Name	Description	Communications method
CIPUCMMRead	Read Variable UCMM Explicit	Reads the value of a variable with a Network Publish attribute from the specified remote Controller on the CIP network and stores the value in a variable at the local Controller.	CIP UCMM connectionless messages
CIPUCMMWrite	Write Variable UCMM Explicit	Writes the value of a variable at the local controller to a variable with a Network Publish attribute at the specified remote Controller on the CIP network.	
CIPUCMMSend	Send Explicit Message UCMM	Sends a specified CIP command to the specified remote Controller on the CIP network. Refer to <i>8-2-10 Response Codes</i> and <i>8-5 CIP Object Services</i> for information on the service codes and response codes that are used with the NY-series Controllers.	
CIPOpen	Open CIP Class 3 Connection (Large_Forward_Open)	Opens a CIP class 3 connection (Large_Forward_Open) with the specified remote node.	CIP class 3 connection message
CIPOpenWithDataSize	Open CIP Class 3 Connection with Specified Data Size	Opens a CIP class 3 connection with the specified remote node that allows class 3 explicit messages of the specified data length or shorter to be sent and received.	
CIPRead	Read Variable Class 3 Explicit	Reads the value of a variable with a Network Publish attribute from the specified remote Controller on the CIP network and stores the value in a variable at the local Controller.	
CIPWrite	Write Variable Class 3 Explicit	Writes the value of a variable at the local controller to a variable with a Network Publish attribute at the specified remote Controller on the CIP network.	
CIPSend	Send Explicit Message Class 3	Sends a specified class-3 CIP command to the specified remote Controller on the CIP network. Refer to <i>8-2-10 Response Codes</i> and <i>8-5 CIP Object Services</i> for information on the service codes and response codes that are used with the NY-series Controllers.	
CIPClose	Close CIP Class 3 Connection	This instruction closes the CIP class 3 connection that is specified by the handle.	

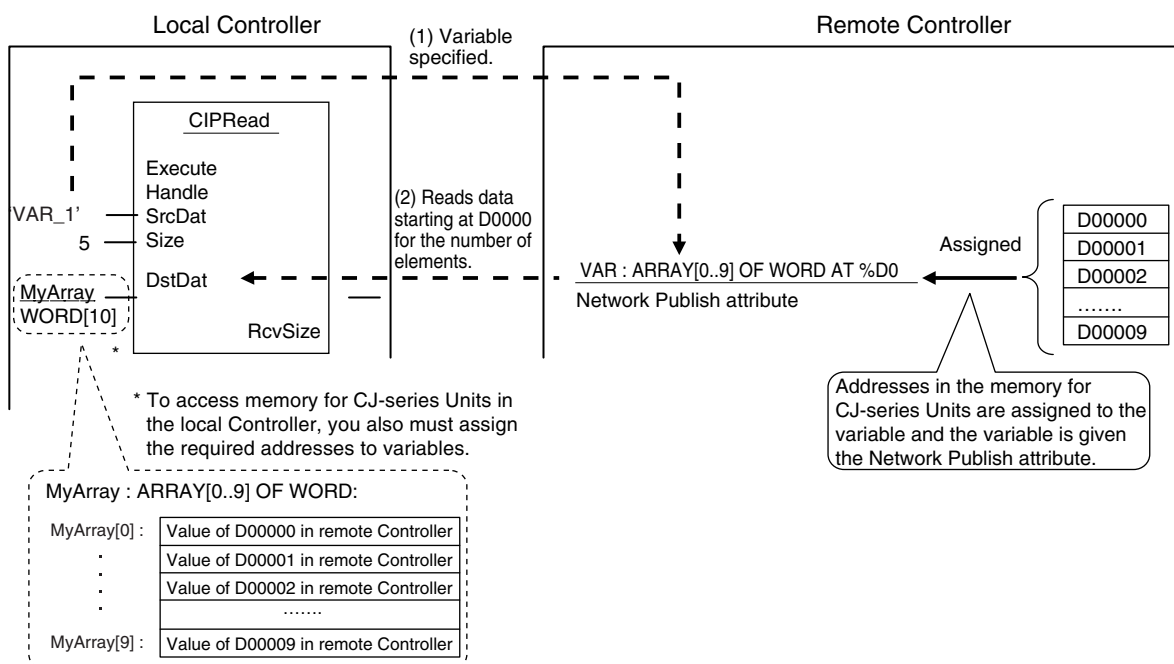
8-2-3 Using CIP Communications Instructions

CIP message communications include the following processes. If CIP class 3 connections are used, the open and close processes are required before and after the data is sent and received.

Process	Description	Instruction
Open process (only for CIP class 3 connections)*1	Execute this process before you use a CIP message. Open processing is continued until a CIP class 3 connection is established.	CIPOpen CIPOpenWithDataSize
Sending and receiving variable data*2	This process is used to read and write data for specified variables with the Network Publish attributes.	CIPUCMMRead CIPUCMMWrite CIPRead CIPWrite
Sending CIP commands	You can set the required CIP command.	CIPUCMMSend CIPSend
Close process (only for CIP class 3 connections)	This process closes the connection.	CIPClose

*1 The maximum number of connection handles that you can obtain simultaneously by the opening process is 32. Even if a connection is broken for a timeout, the handle is not released. Execute the CIPClose instruction.

*2 Addresses in memory for CJ-series Units (e.g., D0000) cannot be specified directly.



Precautions for Correct Use

You can execute up to 32 CIP communications instructions at the same time regardless of the instruction types.

Use exclusive control in the user program so that the number of CIP communications instructions executed at the same time does not exceed the above numbers.

8-2-4 Route Path

The route path indicates the path from the local Controller to the remote Controller on the network. Routing is performed for CIP communications instructions based on route paths.

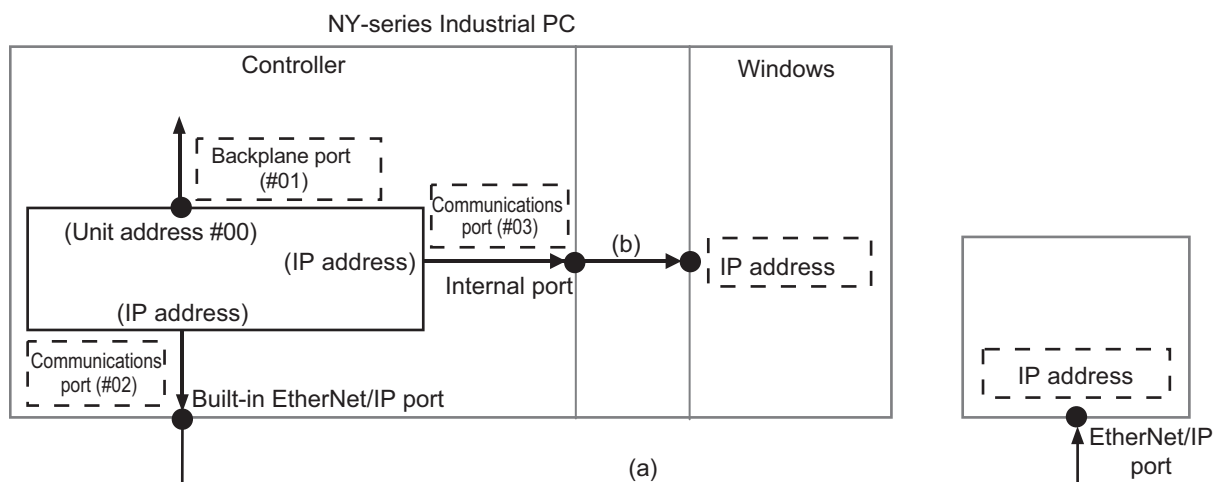
Route Path Notation

The EPATH data type is used to give route paths. The basic format is shown below.

Network_type_number \ Remote_address

The network type number and the destination address for the Controller in an NY-series Industrial PC are determined as shown in the following table.

Network type	Route path notation	
	Network type number (hexadecimal)	Destination address (hexadecimal)
Backplane port	#01	Remote Unit address (Refer to <i>Additional Information</i> below.)
Built-in EtherNet/IP port	#02	IP address
Internal Port	#03	



(1) When Routing the Output from the Controller in an Industrial PC to a Node on the EtherNet/IP Network

Output the command to the built-in EtherNet/IP port. Specify the IP address as the address of the remote node.

(2) When Routing the Output from the Controller to Windows in an Industrial PC

Output the command to the internal port. Specify the IP address as the address of the remote node.



Additional Information

Unit Addresses

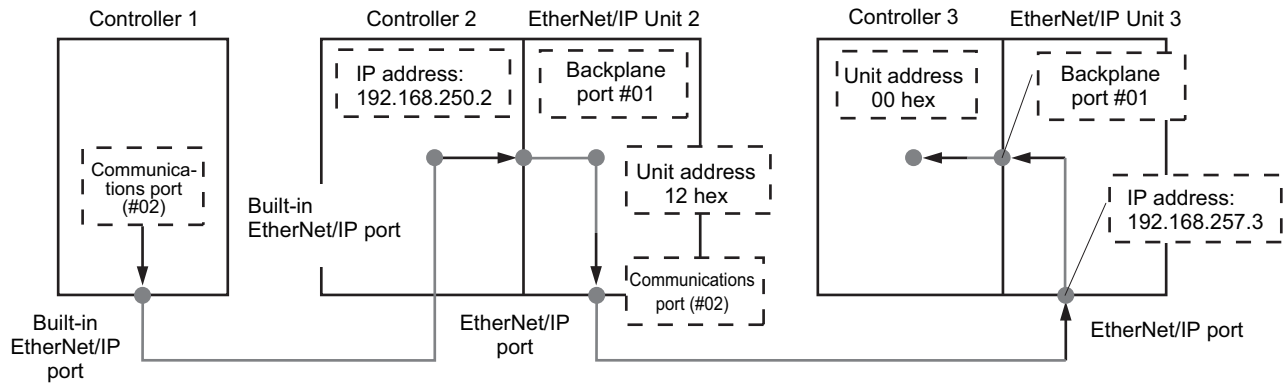
A unit address is used to discriminate between several devices connected to a single node on a network.

Set the unit address as shown below.

- Controller: 00 hex

Accessing via a Relay Node

Example: Communicating from Controller 1 to Controller 3 via Controller 2



(1) Controller 1 to Controller 2

- Network type number: "#02" (Output the command via EtherNet/IP port.)
- Remote address: Specify the remote IP address.

(2) Controller 2 to EtherNet/IP Unit 2

- Network type number: "#01" (Output the command via internal backplane port.)
- Remote address: "#12" (Unit address of EtherNet/IP Unit (Unit number: 2+10 hex = 12 hex))

(3) EtherNet/IP Unit 2 to EtherNet/IP Unit 3

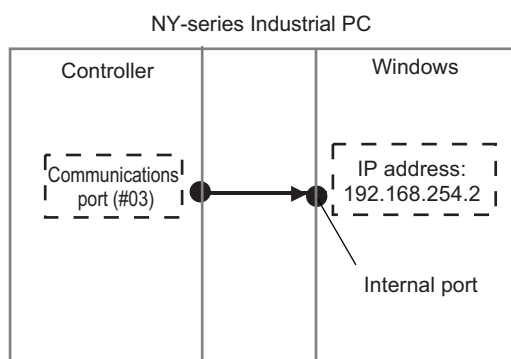
- Network type number: "#02" (Output the command via EtherNet/IP port.)
- Remote address: Specify the remote IP address.

(4) EtherNet/IP Unit 3 to Controller 3

- Network type number: "#01" (Output the command via internal backplane port.)
- Remote address: "#00" (unit address of Controller)

Route path : 02\192.168.250.2\01\#12\02\192.168.257.3\01\#00
 (1) (2) (3) (4)

● When Routing the Output from the Controller to Windows in an Industrial PC



- Network type number: "#03" (Output the command via internal port.)
- Destination address: Specify the destination IP address
- Route path: 03\192.168.254.2

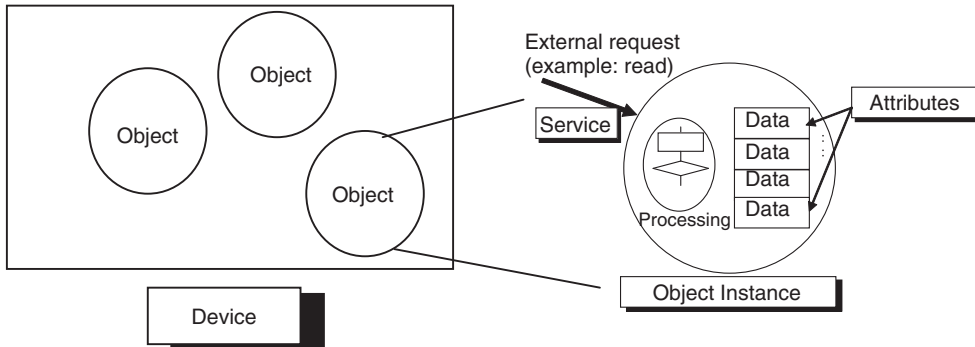
8-2-5 Request Path (IOI)

A request path indicates an object of a device on the network.

A CIP communications instruction uses the request path to access an object of a device.

Overview of Request Path

In the CIP world, each device is modeled as a collection of objects. An Object abstractly represents the specific configuration elements of a device.

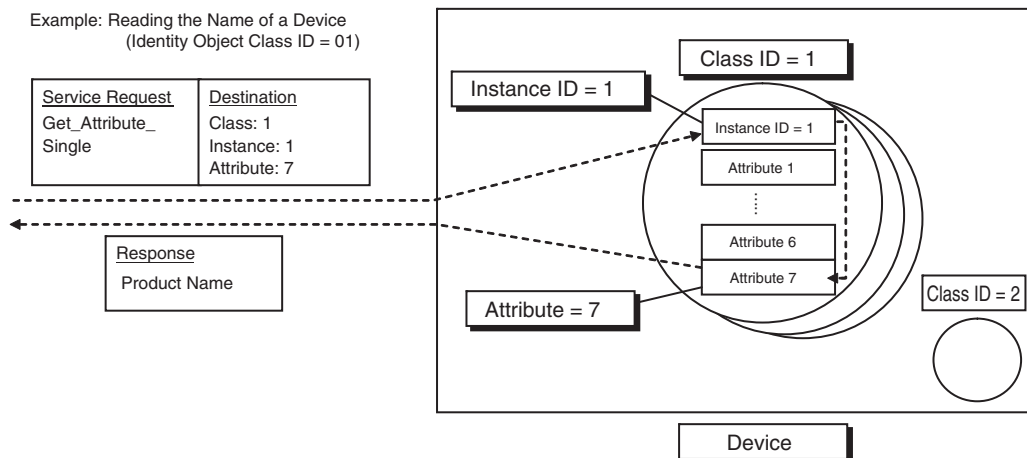


In the CIP Common Specification, “Object,” “Class,” “Instance,” “Attribute” and “Service” are defined as follows: (Source: CIP Common Specification)

Term	Definition
Object	An abstract representation of a particular component within a device.
Class	A set of objects that all represent the same kind of system component.
Instance	A specific and real (physical) occurrence of an object.
Attribute	A description of an externally visible characteristic or feature of an object.
Service	A request from an external object (e.g., to read data).

You use the Class ID, Instance ID, and Attribute ID to access an object.

You specify these three IDs to designate an object in a device. When you make a request from an external device for a service, you must specify the Class ID, Instance ID, and Attribute ID. (The Instance ID and Attribute ID are not required for some services.)



These are called Internal Object Identifiers (IOI) because they identify the Class ID, Instance ID, and Attribute ID within the device. Refer to 8-5 CIP Object Services for the class ID, instance ID, attribute ID, and service code for each object.

Providing the Structure Variables to Input Request Paths

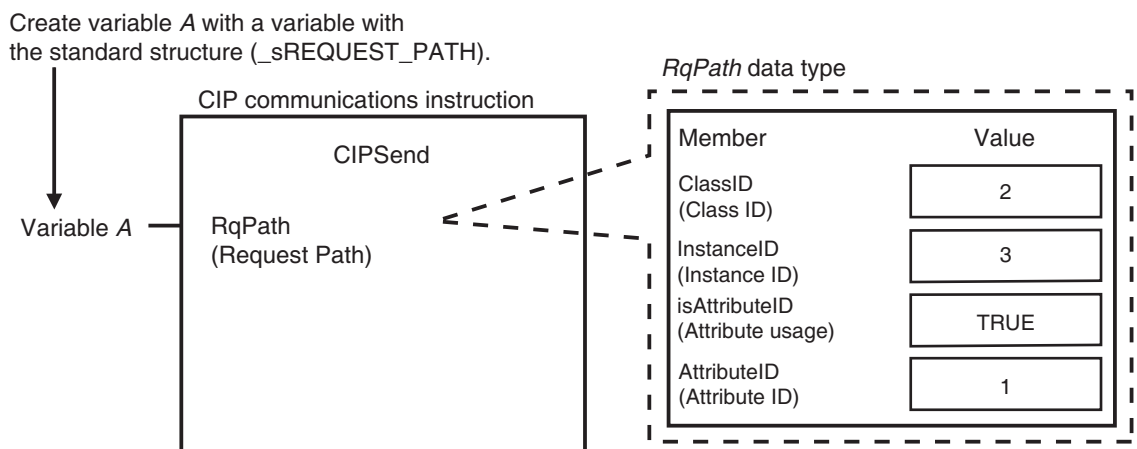
For a CIP communications instruction, you prepare a variable to store the request path. In this variable, you specify the object to access with the user program.

A structure in which the Class ID, Instance ID and Attribute ID are specified is provided for the data type of a variable for a request path.

There are two types of structures: standard structure (`_sREQUEST_PATH`) and extension structure (`_sREQUEST_PATH_EX`). When you use an extension structure, it is possible to specify the size according to the size of values of the Class ID, Instance ID and Attribute ID of the object that you access. When you use a standard structure, the size is always set to 16 bits.

● When a Standard Structure Variable Is Used

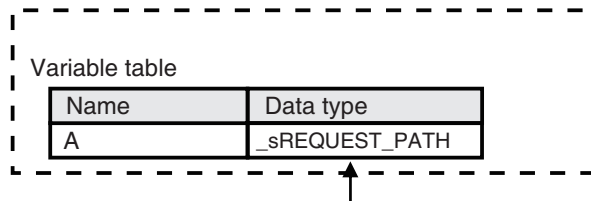
Example: Using a standard structure variable to input values into *RqPath* (Request Path) for the CIPSend instruction



1 Create a standard structure variable.

To use a standard structure variable to input values into *RqPath* (Request Path) for a CIP communications instruction, first you need to create a standard structure user-defined variable.

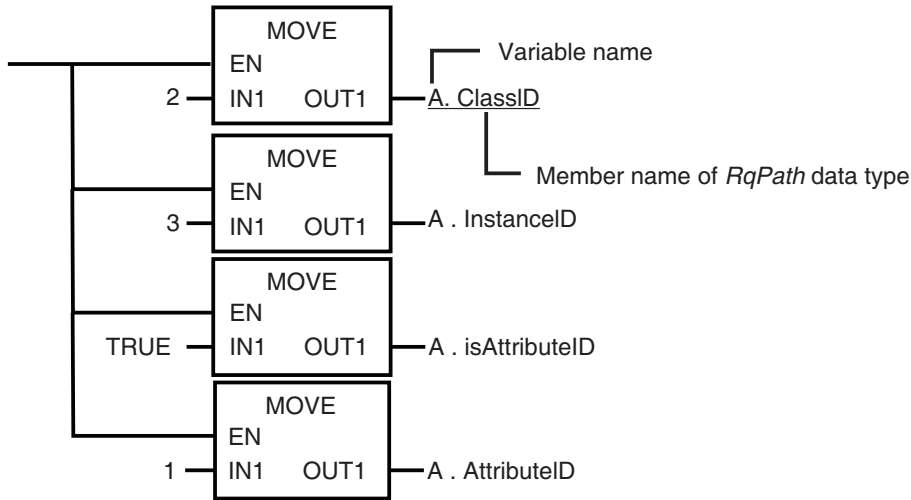
When you create a variable in a variable table, select the pre-registered standard structure (`sREQUEST_PATH`) for a CIP communications instruction.



Select a standard structure for the data type of variable *A*.

2 Input a value for each standard structure variable member.

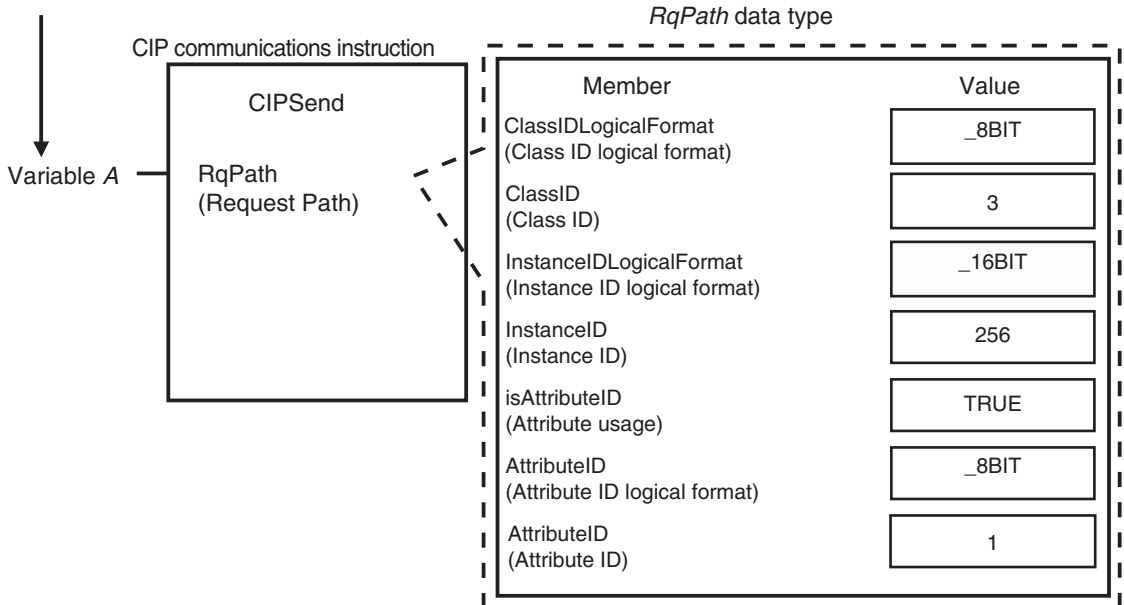
Input the following values into the communications parameters that were registered as members of the standard structure variable.



● **When an Extension Structure Variable Is Used**

Example: Using an extension structure variable to input values into *RqPath* (Request Path) for the CIPSend instruction

Create variable *A* with a variable with the extension structure (`_sREQUEST_PATH_EX`).



1 Create an extension structure variable.

To use an extension structure variable to input values into *RqPath* (Request Path) for a CIP communications instruction, first you need to create an extension structure user-defined variable.

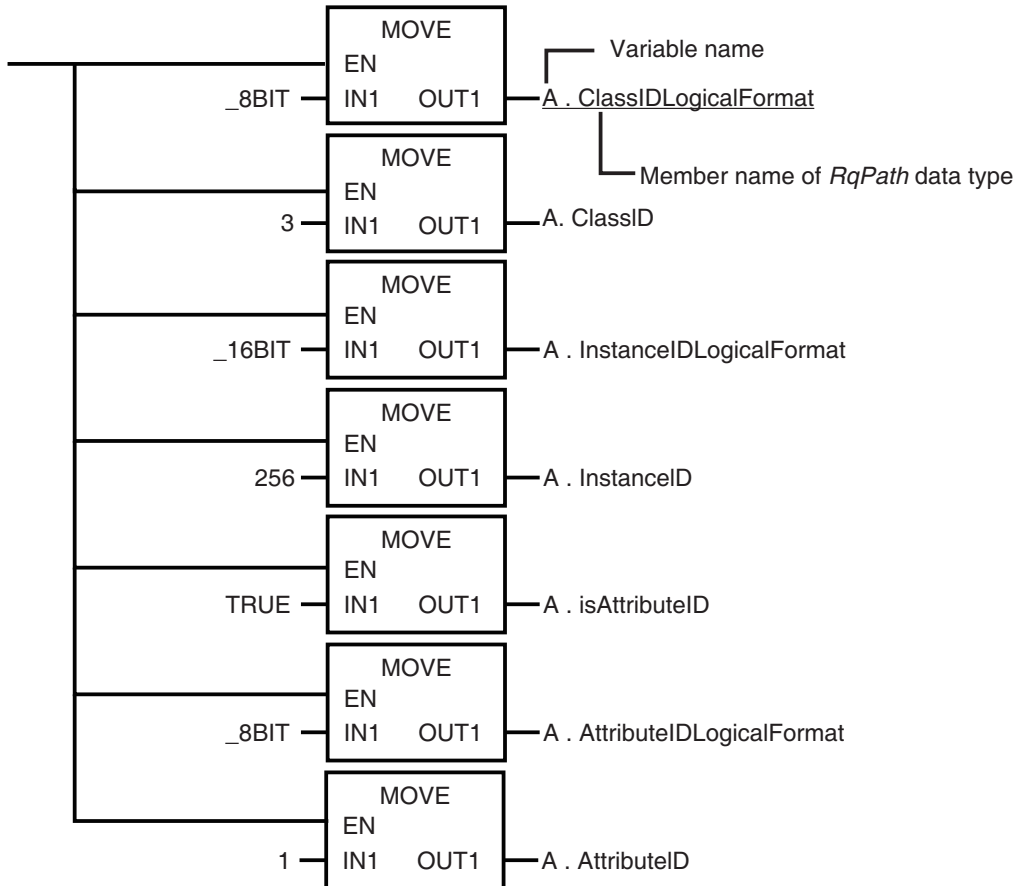
When you create a variable in a variable table, select the pre-registered extension structure (`_sREQUEST_PATH_EX`) for a CIP communications instruction.

Variable table	
Name	Data type
A	_sREQUEST_PATH_EX

Select an extension structure for the data type of variable A.

2 Input a value for each extension structure variable member.

Input the following values into the communications parameters that were registered as members of the extension structure variable.



8-2-6 Service Data and Response Data

CIP communications instructions send and receive data that is stored in array variables.

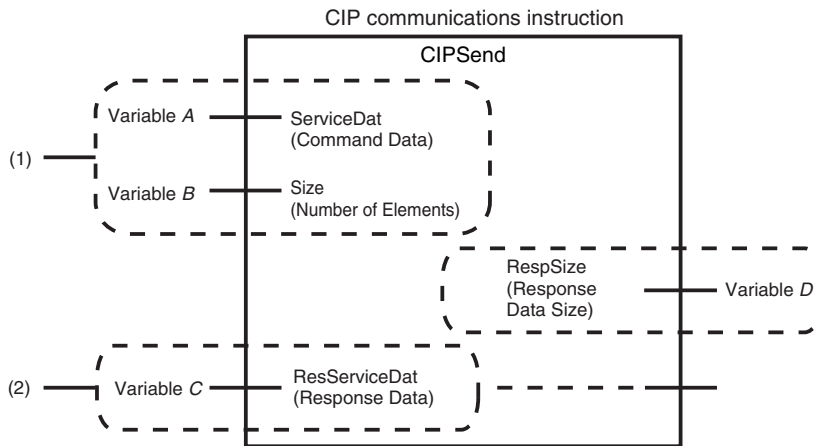
Preparing Array Variables to Input and Output Service Data and Response Data

This section describes the array variables for storing service data and response data that CIP communications instructions send and receive.

● Creating Array Variables

To input a value into the array variable of a CIP communications instruction, you must create a variable with the same configuration as the array variable in advance.

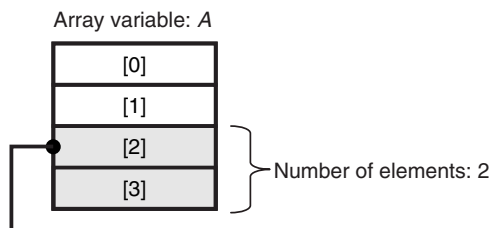
Example: Creating a Variable to Input Data to the CIPSend Instruction Array Variables



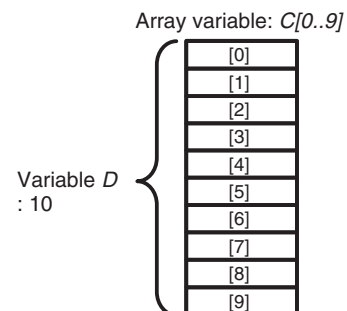
(1) Input the service data to send.

The data to send is stored in array variable A. If only certain elements are specified in array variable A, specify the number of elements in variable B.

(2) The data that is received is stored in variable C. The byte size of the data that was actually received is stored in variable D.



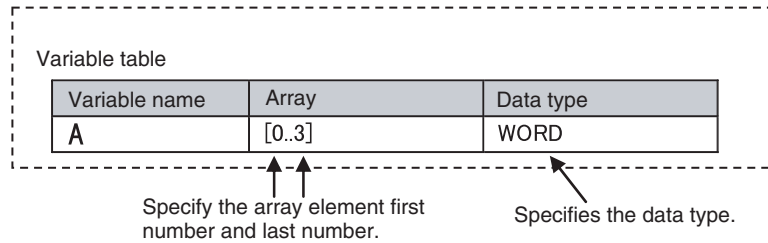
If the service data (*ServiceDat*) is *Array[2]* and number of elements (*Size*) = 2, *Array[2]* and *Array[3]* are sent.



Use the following procedure to create a variable in the variable table.

- 1** Select the *Array* Check Box.
- 2** Specify the element first number, the element last number, and the data type.

Example: UINT Array



● CIP Communications Instructions That Use Array Variables

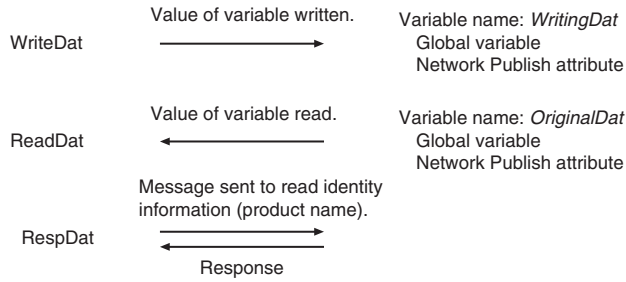
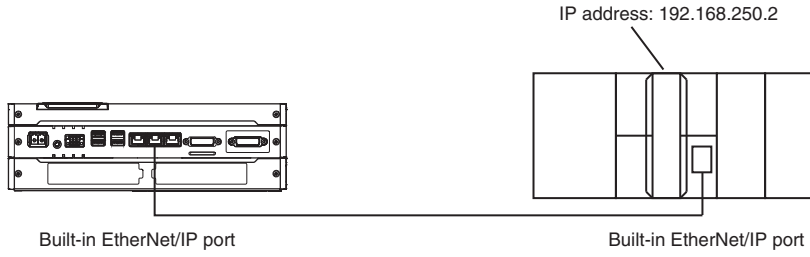
Instruction	Structure variable name		
	Input variable	In-out variable	Output variable
CIPRead	---	---	DstDat (Read Data)
CIPWrite	SrcDat (Write Data)	---	---
CIPSend	ServiceDat (Command Data)	RespServiceDat (Response Data)	---

8-2-7 Sample Programming for CIP Connectionless (UCMM) Message Communications

This sample uses CIP UCMM messages to write a variable, read a variable, and send a message. The Controllers are connected to an EtherNet/IP network. The IP address of the remote node is 192.168.250.2. The following procedure is used.

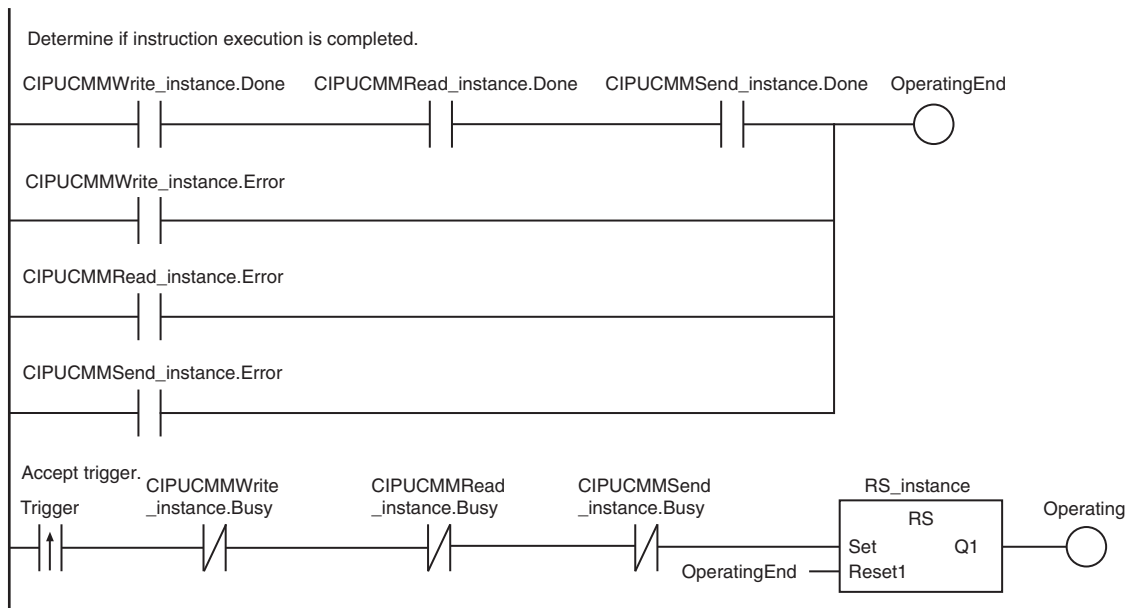
- 1** The CIPUCMMWrite instruction is used to write the value of a variable at a remote node. The variable name at the remote node is *WritingDat* and the contents of the *WriteDat* is written to it. *WritingDat* must be defined as a global variable at the remote node and the Network Publish attribute must be set.
- 2** The CIPUCMMRead instruction is used to read the value of a variable at a remote node. The value of the variable *OriginalDat* at the other node is read and the read value is stored in the *ReadDat* variable. *OriginalDat* must be defined as a global variable at the remote node and the Network Publish attribute must be set.
- 3** The CIPUCMMSend instruction is used to send an explicit message to a remote node. The contents of the message is to read identity information (product name). The class ID, instance ID, attribute ID, and service code are as follows. The response data is stored in the *RespDat* variable.

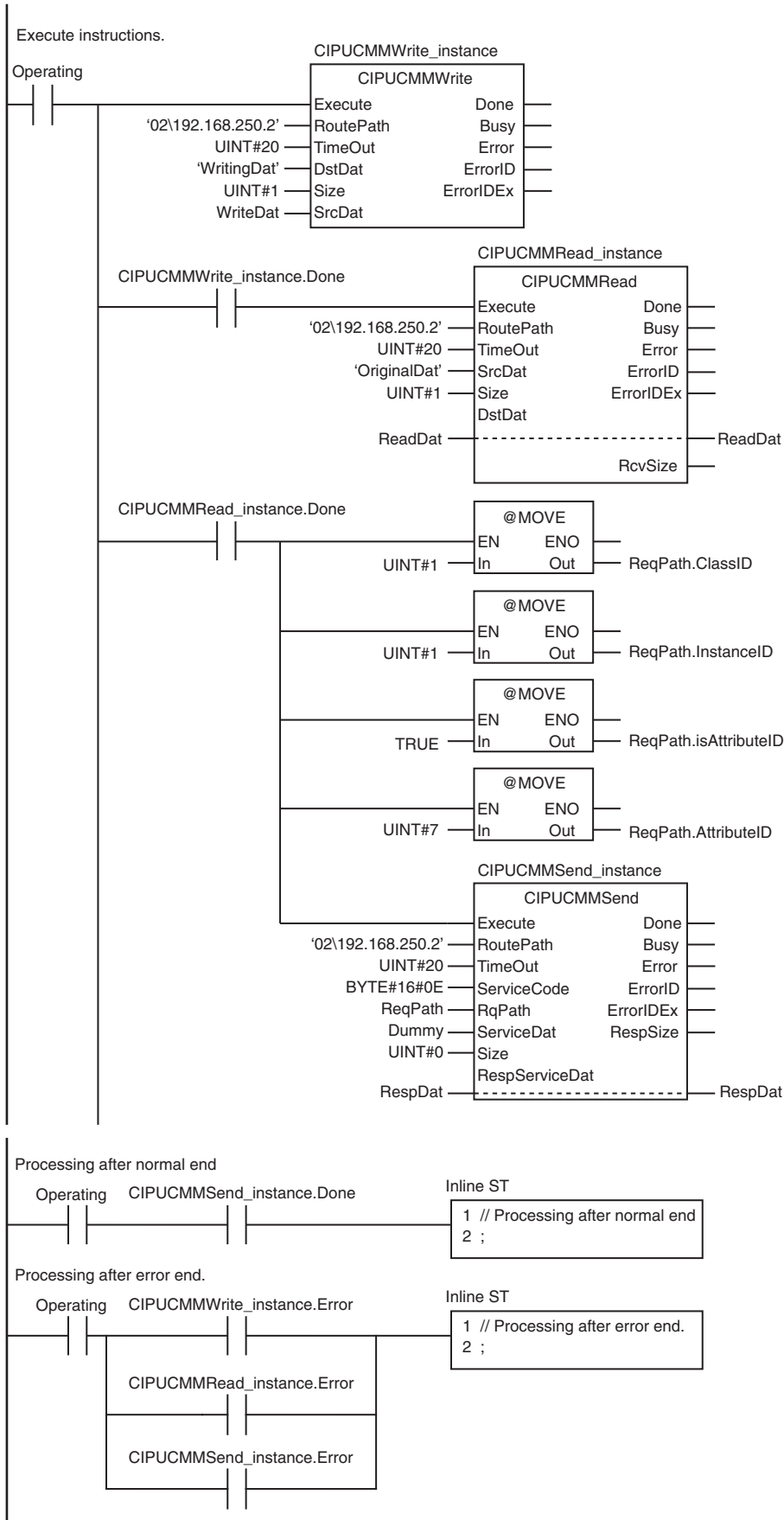
Item	Value
Class ID	1
Instance ID	1
Attribute ID	7
Service code	16#0E



LD

Variable	Data type	Initial value	Comment
OperatingEnd	BOOL	False	Processing finished.
Trigger	BOOL	False	Execution condition
Operating	BOOL	False	Processing
WriteDat	INT	1234	Source data
ReadDat	INT	0	Read data
ReqPath	_sREQUEST_PATH	(ClassID:=0, InstanceID:=0, isAttributeID:=False, AttributeID:=0)	Request path
RespDat	ARRAY[0..10] OF BYTE	[11(16#0)]	Response data
Dummy	BYTE	16#0	Dummy
RS_instance	RS		
CIPUCMMWrite_instance	CIPUCMMWrite		
CIPUCMMRead_instance	CIPUCMMRead		
CIPUCMMSend_instance	CIPUCMMSend		





ST

Internal Variables	Variable	Data type	Initial value	Comment
	Trigger	BOOL	False	Execution condition
	DoUCMMTrigger	BOOL	False	Processing
	Stage	INT	0	Status change
	WriteDat	INT	1234	Write data
	ReadDat	INT	0	Read data
	ReqPath	_sREQUEST_PATH	(ClassID:=0, InstanceID:=0, isAttributeID:=False, AttributeID:=0)	Request path
	RespDat	ARRAY[0..10] OF BYTE	[11(16#0)]	Response data
	Dummy	BYTE	16#0	Dummy
	CIPUCMMWrite_instance	CIPUCMMWrite		
	CIPUCMMRead_instance	CIPUCMMRead		
	CIPUCMMSend_instance	CIPUCMMSend		

External Variables	Variable	Data type	Constant	Comment
	_EIP_EtnOnlineSta*1	BOOL	<input checked="" type="checkbox"/>	Online

*1 Replace the variable name with `_EIP1_EtnOnlineSta` (Port1 Online) or `_EIPIn1_EtnOnlineSta` (Internal Port1 Online), depending on which communications port you use.

// Start sequence when *Trigger* changes to TRUE.

IF ((Trigger=TRUE) AND (DoUCMMTrigger=FALSE) AND (_EIP_EtnOnlineSta=TRUE))

THEN

DoUCMMTrigger :=TRUE;

Stage :=INT#1;

CIPUCMMWrite_instance(

Execute :=FALSE, // Initialize instance.

SrcDat :=WriteDat); // Dummy

CIPUCMMRead_instance(// Initialize instance.

Execute :=FALSE, // Dummy

DstDat :=ReadDat); // Dummy

CIPUCMMSend_instance(

Execute :=FALSE, // Initialize instance.

ServiceDat := Dummy, // Dummy

RespServiceDat :=RespDat); // Dummy

END_IF;

IF (DoUCMMTrigger=TRUE) THEN

CASE Stage OF

1 : // Request writing value of variable.

CIPUCMMWrite_instance(

Execute :=TRUE,

RoutePath :='02\192.168.250.2', // Route path

TimeOut :=UINT#20, // Timeout value

DstDat :='WritingDat', // Source variable name

Size :=UINT#1, // Number of elements to write

SrcDat :=WriteDat); // Write data

```

IF (CIPUCMMWrite_instance.Done=TRUE) THEN
    Stage      :=INT#2;           // Normal end
ELSIF (CIPUCMMWrite_instance.Error=TRUE) THEN
    Stage      :=INT#10;         // Error end
END_IF;
2 :           // Request reading value of variable.
CIPUCMMRead_instance(
    Execute     :=TRUE,
    RoutePath   :='02\192.168.250.2', // Route path
    TimeOut     :=UINT#20,         // Timeout value
    SrcDat      :='OriginalDat',   // Source variable name
    Size        :=UINT#1,         // Number of elements to read
    DstDat      :=ReadDat);       // Read data

IF (CIPUCMMRead_instance.Done=TRUE) THEN
    Stage      :=INT#3;           // Normal end
ELSIF (CIPUCMMRead_instance.Error=TRUE) THEN
    Stage      :=INT#40;         // Error end
END_IF;
3 :           // Send message
ReqPath.ClassID:=UINT#01;
ReqPath.InstanceID:=UINT#01;
ReqPath.isAttributeID:=TRUE;
ReqPath.AttributeID:=UINT#07;
CIPUCMMSend_instance(
    Execute     :=TRUE,
    RoutePath   :='02\192.168.250.2', // Route path
    TimeOut     :=UINT#20,         // Timeout time
    ServiceCode :=BYTE#16#0E,     // Service code
    RqPath      :=ReqPath,        // Request path
    ServiceDat  :=Dummy,         // Service data
    Size        :=UINT#0,        // Number of elements
    RespServiceDat :=RespDat);   // Response data

IF (CIPUCMMSend_instance.Done=TRUE) THEN
    Stage      :=INT#0;           // Normal end
ELSIF (CIPUCMMSend_instance.Error=TRUE) THEN
    Stage      :=INT#30;         // Error end
END_IF;
0 :           // Processing after normal end
DoUCMMTrigger:=FALSE;
Trigger       :=FALSE;

ELSE           // Processing after error end
    DoUCMMTrigger:=FALSE;
    Trigger       :=FALSE;
END_CASE;
END_IF;

```

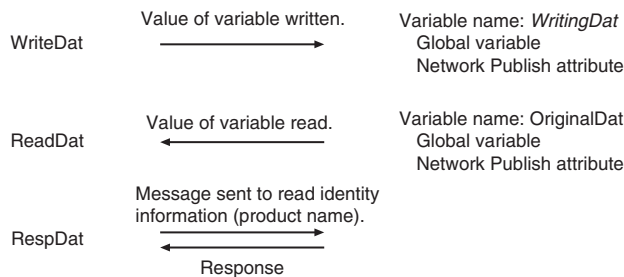
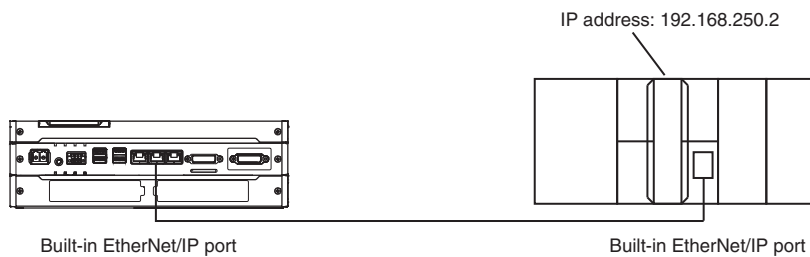
8-2-8 Sample Programming for CIP Connection (Class 3) Message Communications

This sample uses CIP class 3 messages to write a variable, read a variable, and send a message. The Controllers are connected to an EtherNet/IP network. The IP address of the remote node is 192.168.250.2. The following procedure is used.

- 1** The CIPOpen is used to open a class 3 connection (Large_Forward_Open). The timeout time is 2 s.
- 2** The CIPWrite instruction is used to write the value of a variable at a remote node. The variable name at the remote node is *WritingDat* and the contents of the *WriteDat* is written to it. *WritingDat* must be defined as a global variable at the remote node and the Network Publish attribute must be set.
- 3** The CIPRead instruction is used to read the value of a variable at a remote node. The value of the variable *OriginalDat* at the other node is read and the read value is stored in the *ReadDat* variable. *OriginalDat* must be defined as a global variable at the remote node and the Network Publish attribute must be set.
- 4** The CIPSend instruction is used to send an explicit message to a remote node. The contents of the message is to read identity information (product name). The class ID, instance ID, attribute ID, and service code are as follows: The response data is stored in the RespDat variable.

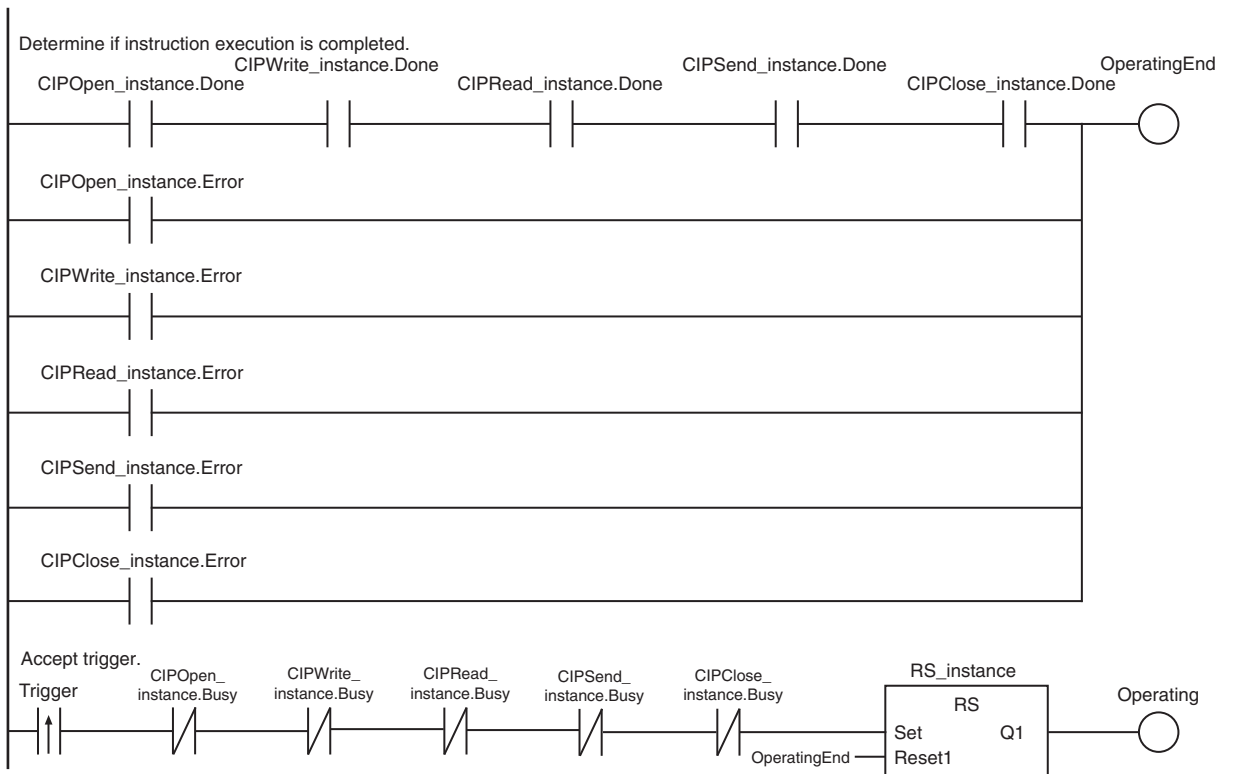
Item	Value
Class ID	1
Instance ID	1
Attribute ID	7
Service code	16#0E

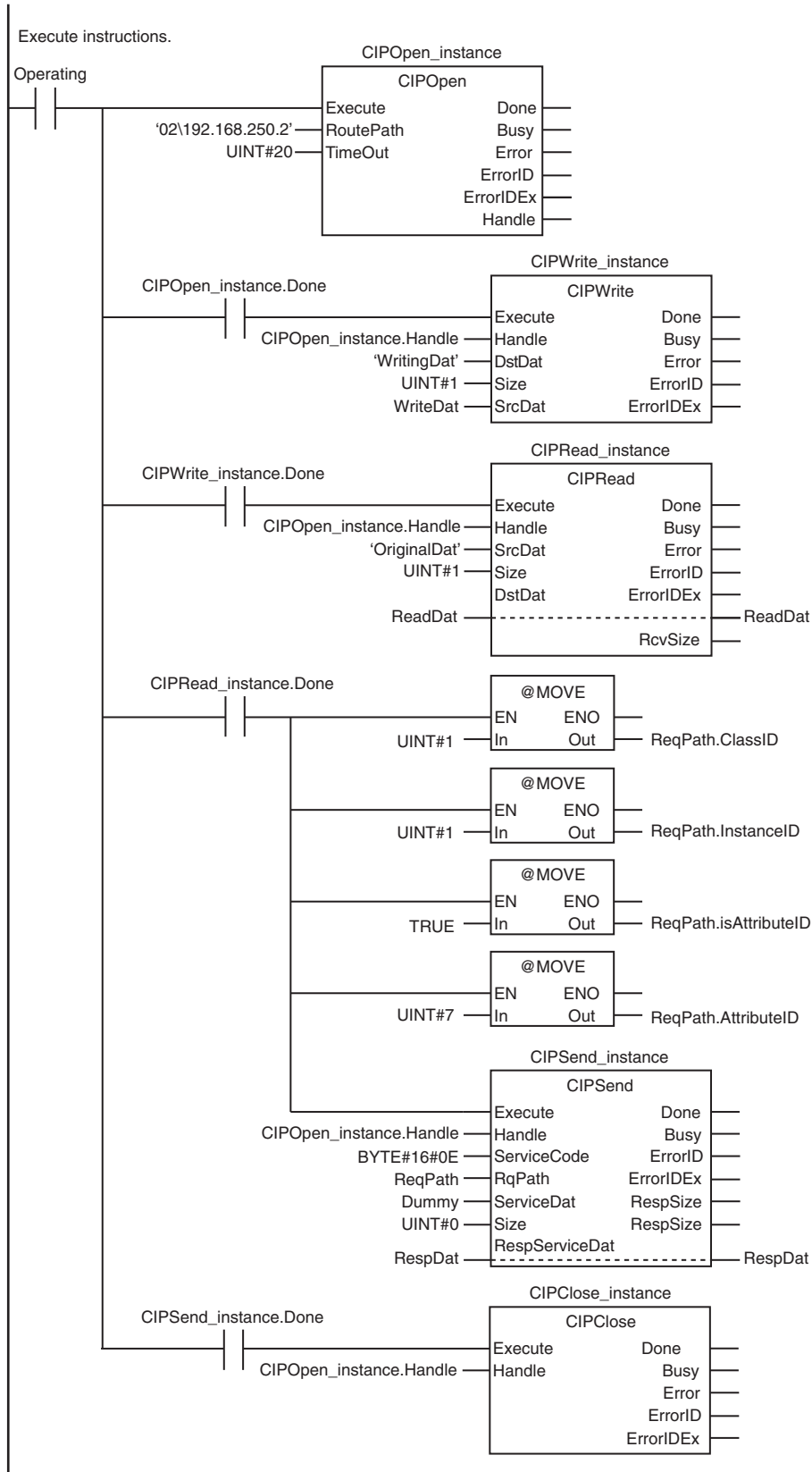
- 5** The CIPClose instruction is used to close the class 3 connection.



LD

Variable	Data type	Initial value	Comment
OperatingEnd	BOOL	False	Processing finished.
Trigger	BOOL	False	Execution condition
Operating	BOOL	False	Processing
WriteDat	INT	1234	Source data
ReadDat	INT	0	Read data
ReqPath	_sREQUEST_PATH	(ClassID:=0, InstanceID:=0, isAttributeID:=False, AttributeID:=0)	Request path
RespDat	ARRAY[0..10] OF BYTE	[11(16#0)]	Response data
Dummy	BYTE	16#0	Dummy
RS_instance	RS		
CIPOpen_instance	CIPOpen		
CIPWrite_instance	CIPWrite		
CIPRead_instance	CIPRead		
CIPSend_instance	CIPSend		
CIPClose_instance	CIPClose		





ST

Internal Variables	Variable	Data type	Initial value	Comment
	Trigger	BOOL	False	Execution condition
	DoCIPTrigger	BOOL	False	Processing
	Stage	INT	0	Status change
	WriteDat	INT	1234	Write data
	ReadDat	INT	0	Read data
	ReqPath	_sREQUEST_PATH	(ClassID:=0, InstanceID:=0, isAttributeID:=False, AttributeID:=0)	Request path
	RespDat	ARRAY[0..10] OF BYTE	[11(16#0)]	Response data
	Dummy	BYTE	16#0	Dummy
	CIPOpen_instance	CIPOpen		
	CIPWrite_instance	CIPWrite		
	CIPRead_instance	CIPRead		
	CIPSend_instance	CIPSend		
	CIPClose_instance	CIPClose		

External Variables	Variable	Data type	Constant	Comment
	_EIP_EtnOnlineSta*1	BOOL	<input checked="" type="checkbox"/>	Online

*1 Replace the variable name with `_EIP1_EtnOnlineSta` (Port1 Online) or `_EIPIn1_EtnOnlineSta` (Internal Port1 Online), depending on which communications port you use.

```
// Start sequence when Trigger changes to TRUE.
IF ((Trigger=TRUE) AND (DoCIPTrigger=FALSE) AND (_EIP_EtnOnlineSta=TRUE))THEN
  DoCIPTrigger:=TRUE;
  Stage          :=INT#1;
  CIPOpen_instance(Execute:=FALSE);           // Initialize instance.
  CIPWrite_instance(
    Execute      :=FALSE,                     // Initialize instance.
    SrcDat       :=WriteDat);                 // Dummy
  CIPRead_instance(
    Execute      :=FALSE,                     // Initialize instance.
    DstDat       :=ReadDat);                 // Dummy
  CIPSend_instance(
    Execute      :=FALSE,                     // Initialize instance.
    ServiceDat   := Dummy,                   // Dummy
    RespServiceDat :=RespDat);               // Dummy
  CIPClose_instance(Execute:=FALSE);         // Initialize instance.
END_IF;
```

```

IF (DoCIPTrigger=TRUE) THEN
CASE Stage OF
1 :                                     // Open CIP Class 3 Connection (Large_Forward_Open)
  CIPOpen_instance(
    Execute   :=TRUE,
    TimeOut   :=UINT#20,                // Timeout time: 2.0 s
    RoutePath :='02\192.168.250.2');    // Route path

  IF (CIPOpen_instance.Done=TRUE) THEN
    Stage     :=INT#2;                  // Normal end
  ELSIF (CIPOpen_instance.Error=TRUE) THEN
    Stage     :=INT#10;                 // Error end
  END_IF;

2 :                                     // Request writing value of variable.
  CIPWrite_instance(
    Execute   :=TRUE,
    Handle    :=CIPOpen_instance.Handle, // Handle
    DstDat    :='WritingDat',           // Source variable name
    Size      :=UINT#1,                 // Number of elements to write
    SrcDat    :=WriteDat);              // Write data

  IF (CIPWrite_instance.Done=TRUE) THEN
    Stage     :=INT#3;                  // Normal end
  ELSIF (CIPWrite_instance.Error=TRUE) THEN
    Stage     :=INT#20;                 // Error end
  END_IF;

3 :                                     // Request reading value of variable.
  CIPRead_instance(
    Execute   :=TRUE,
    Handle    :=CIPOpen_instance.Handle, // Handle
    SrcDat    :='OriginalDat',          // Source variable name
    Size      :=UINT#1,                 // Number of elements to read
    DstDat    :=ReadDat);              // Read data

  IF (CIPRead_instance.Done=TRUE) THEN
    Stage     :=INT#4;                  // Normal end
  ELSIF (CIPRead_instance.Error=TRUE) THEN
    Stage     :=INT#30;                 // Error end
  END_IF;

```

```

4 :                                     // Send message
  ReqPath.ClassID :=UINT#01;
  ReqPath.InstanceID :=UINT#01;
  ReqPath.isAttributeID:=TRUE;
  ReqPath.AttributeID :=UINT#07;
  CIPSend_instance(
    Execute :=TRUE,
    Handle :=CIPOpen_instance.Handle, // Handle
    ServiceCode:=BYTE#16#0E, // Service code
    RqPath :=ReqPath, // Request path
    ServiceDat :=Dummy, // Service data
    Size :=UINT#0, // Number of elements
    RespServiceDat:=RespDat); // Response data

  IF (CIPSend_instance.Done=TRUE) THEN
    Stage :=INT#5; // Normal end
  ELSIF (CIPSend_instance.Error=TRUE) THEN
    Stage :=INT#40; // Error end
  END_IF;

5 :                                     // Request closing CIP class 3 connection.
  CIPCclose_instance(
    Execute :=TRUE,
    Handle :=CIPOpen_instance.Handle); // Handle

  IF (CIPCclose_instance.Done=TRUE) THEN
    Stage :=INT#0;
  ELSIF (CIPCclose_instance.Error=TRUE) THEN
    Stage :=INT#50;
  END_IF;

0:                                     // Processing after normal end
  DoCIPTrigger:=FALSE;
  Trigger :=FALSE;

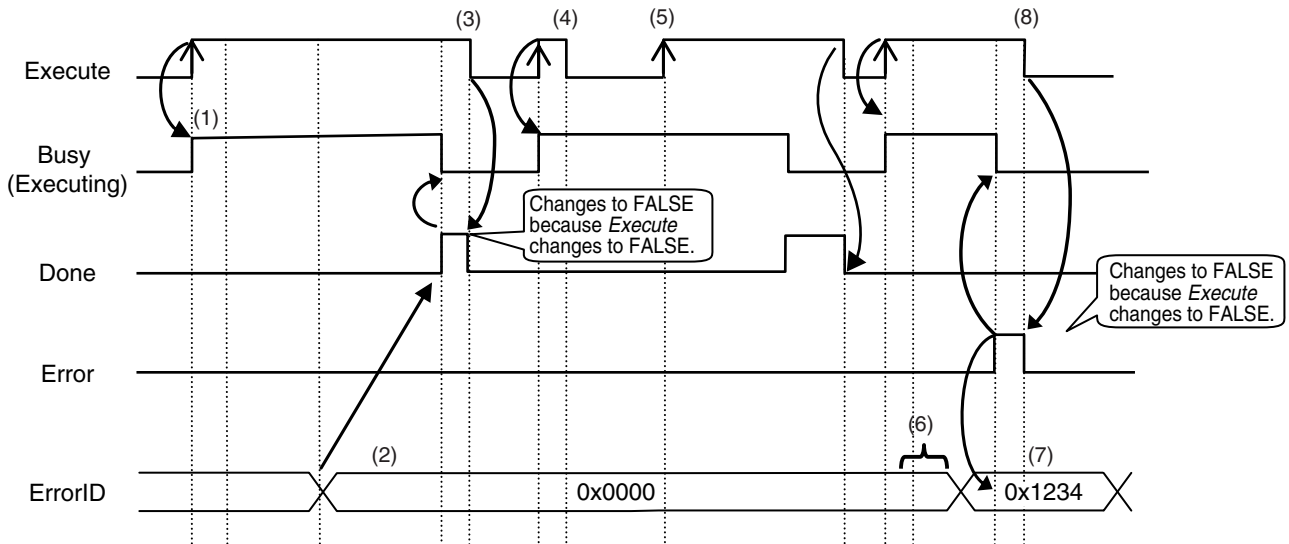
  ELSE // Processing after error end
    DoCIPTrigger :=FALSE;
    Trigger :=FALSE;
  END_CASE;
END_IF;

```

8-2-9 Operation Timing

Output Variable Operation and Timing

You can monitor the values of the output variables to determine the status throughout instruction execution. The following timing chart shows the operation of the output variables.



- 1** When *Execute* changes to TRUE, the instruction is executed and *Busy* changes to TRUE.
- 2** After the results of instruction execution are stored in the output variables, *Done* changes to TRUE and *Busy* changes to FALSE.
- 3** When *Execute* changes to FALSE, *Done* returns to FALSE.
- 4** When *Execute* changes to TRUE again, *Busy* changes to TRUE.
- 5** *Execute* is ignored if it changes to TRUE during instruction executed (i.e., when *Busy* is TRUE).
- 6** If an error occurs, several retries are attempted internally. The error code in the *ErrorID* is not updated during the retries.
- 7** When a communications error occurs, *Error* changes to TRUE and the *ErrorID* is stored. Also, *Busy* and *Done* change to FALSE.
- 8** When *Execute* changes to FALSE, *Error* changes to FALSE.

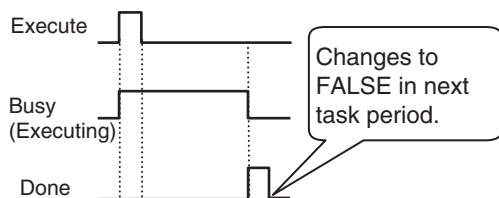


Precautions for Correct Use

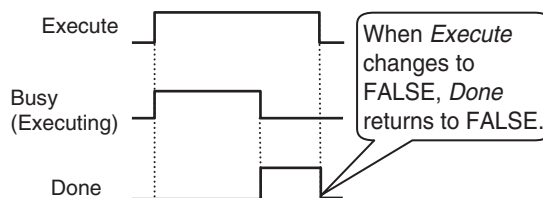
If *Execute* changes back to FALSE before *Done* changes to TRUE, *Done* stays TRUE for only one task period. (Example 1)

If you want to see if *Done* is TRUE at any time, make sure to keep *Execute* TRUE until you confirm that *Done* is TRUE. If *Execute* is TRUE until *Done* changes to TRUE, *Done* stays TRUE until *Execute* changes to FALSE. (Example 2)

Example 1



Example 2

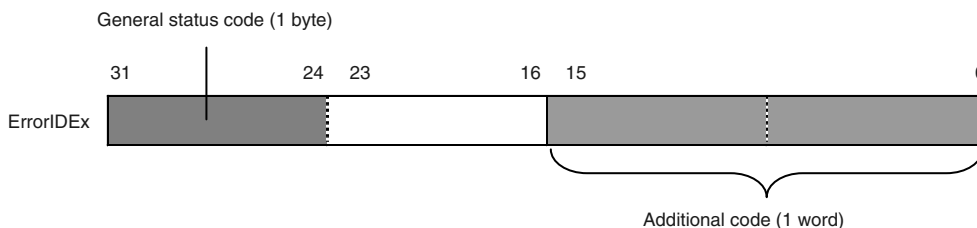


8-2-10 Response Codes

This section describes the response codes stored in the output variable *ErrorIDEx* if an error occurs during the execution of a CIP message communications instruction.

General Status Codes

As response codes, general codes are stored in the *ErrorIDEx* output variable (DWORD data) after execution of a CIP communications instruction is completed. If an additional code is added, the additional code is also stored.



General status code (hex)	Status name	Description of status
00	Success	Service was successfully performed by the object specified.
01	Connection failure	A connection related to service failed along the connection path.
02	Resource unavailable	Resources needed for the object to perform the requested service were unavailable.
03	Invalid parameter value	See Status Code 20 hex.
04	Path segment error	The path segment identifier or the segment syntax was not understood by the processing node. Path processing stops when a path segment error occurs.
05	Path destination unknown	The path is referencing an object class, instance, or structure element that is not known or is not contained in the processing node. Path processing stops when a Path Destination Unknown Error occurs.
06	Partial transfer	Only part of the expected data was transferred.
07	Connection lost	The message connection was lost.

General status code (hex)	Status name	Description of status
08	Service not supported	The requested service was not supported or was not defined for this object class/instance.
09	Invalid attribute value	Invalid attribute data was detected.
0A	Attribute list error	An attribute in the Get_Attribute_List or Set_Attribute_List response has a non-zero status.
0B	Already in requested mode/state	The object is already in the mode/state being requested by the service.
0C	Object state conflict	The object cannot perform the requested service in its current mode/state.
0D	Object already exists	The requested instance of object to be created already exists.
0E	Attribute not settable	A request to modify a non-modifiable attribute was received.
0F	Privilege violation	A permission/privilege check failed.
10	Device state conflict	The device's current mode/state prohibits the execution of the requested service.
11	Reply data too large	The data to be transmitted in the response buffer is larger than the allocated response buffer.
12	Fragmentation of a primitive value	The service specified an operation that is going to fragment a primitive data value, i.e. half a REAL data type.
13	Not enough data	The requested service did not supply enough data to perform the specified operation.
14	Attribute not supported	The attribute specified in the request is not supported.
15	Too much data	The service supplied more data than was expected.
16	Object does not exist	An object that does not exist was specified for the requested service.
17	Service fragmentation sequence not in progress	The fragmentation sequence for this service is not currently active for this data.
18	No stored attribute data	The attribute data of this object was not saved prior to the requested service.
19	Store operation failure	The attribute data of this object was not saved due to a failure during the attempt.
1A	Routing failure (request packet too large)	The service request packet was too large for transmission on a network in the path to the destination. The routing device was forced to abort the service.
1B	Routing failure (response packet too large)	The service response packet was too large for transmission on a network in the path from the destination. The routing device was forced to abort the service.
1C	Missing attribute list entry data	The service did not supply an attribute in a list of attributes that was needed by the service to perform the requested behavior.
1D	Invalid attribute value list	The service is returning the list of attributes supplied with status information for those attributes that were invalid.
1E	Embedded service error	An embedded service resulted in an error.
1F	Vendor specific error	A vendor-specific error occurred. The Additional Code Field of the error response defines the error. This is a general error code that is used only for errors that do not correspond to any of the error codes in this table and are not in an object class definition.
20	Invalid parameter	A parameter for the requested service is invalid. This code is used when a parameter does not meet the requirements of the specification and/or the requirements defined in an application object specification.

General status code (hex)	Status name	Description of status
21	Write-once value or medium already written	An attempt was made to write to a write-once medium (e.g. WORM drive or PROM) that was previously written or cannot be changed.
22	Invalid Reply Received	An invalid reply was received. (For example, the reply service code does not match the request service code, or the reply message is shorter than the minimum expected reply size.) This status code is used for other causes of invalid replies.
23-24		Reserved by CIP for future extensions.
25	Key Failure in path	The key segment that was included as the first segment in the path does not match the destination module. The object specific status must indicate which part of the key check failed.
26	Path Size Invalid	The size of the path that was sent with the service request is either too large or too small for the request to be routed to an object.
27	Unexpected attribute in list	An attempt was made to set an attribute that is not able to be set at this time.
28	Invalid Member ID	The member ID specified in the request does not exist in the specified class, instance, and attribute.
29	Member not settable	A request to modify a non-modifiable member was received.
2A	Group 2 only server general failure	This error code is reported only by group 2 only servers with 4K or less of code space and only in place of <i>Service not supported</i> , <i>Attribute not supported</i> , or <i>Attribute not settable</i> .
2B-CF		Reserved by CIP for future extensions.
D0-FF	Reserved for Object Class and service errors	This range of error codes is to be used to indicate object class-specific errors. This code range is used only when none of the error codes in this table accurately reflect the error that occurred. The additional code field is used to describe the general error code in more detail.

● Examples of Additional Status When General Status Is 01 Hex

(Status of Connection Manager Object)

General Status (hex)	Additional Status (hex)	Description
01	0100	Connection in use or duplicate forward open.
01	0103	Transport class and trigger combination not supported.
01	0106	Ownership conflict.
01	0107	Connection not found at target application.
01	0108	Invalid connection type. There is a problem with either the connection type or priority of the connection.
01	0109	Invalid connection size.
01	0100	Device not configured.
01	0111	RPI not supported. May also indicate problem with connection time-out multiplier, or production inhibit time.
01	0113	Connection Manager cannot support any more connections.
01	0114	Either the vendor ID or the product code in the key segment does not match the device.
01	0115	Device type in the key segment does not match the device.

General Status (hex)	Additional Status (hex)	Description
01	0116	Major or minor revision information in the key segment does not match the device.
01	0117	Invalid connection point.
01	0118	Invalid configuration format.
01	0119	Connection request failed because there is no controlling connection currently open.
01	011A	Target application cannot support any more connections.
01	011B	RPI is smaller than the production inhibit time.
01	0203	Connection cannot be closed because the connection has timed out.
01	0204	Unconnected_Send service timed out while waiting for a response.
01	0205	Parameter error in Unconnected_Send service.
01	0206	Message too large for unconnected message service.
01	0207	Unconnected acknowledgement without reply.
01	0301	No buffer memory available.
01	0302	Network bandwidth not available for data.
01	0303	No tag filters available.
01	0304	Not configured to send real-time data.
01	0311	Port that was specified in port segment is not available.
01	0312	Link address that was specified in port segment is not available.
01	0315	Invalid segment type or segment value in path.
01	0316	Path and connection were not equal when closing the connection.
01	0317	Either the segment is not present or the encoded value in the network segment is invalid.
01	0318	Link address to self is invalid.
01	0319	Resources on secondary are unavailable.
01	031A	Connection is already established.
01	031B	Direct connection is already established.
01	031C	Others
01	031D	Redundant connection mismatch.
01	031E	There are no more reception resources available on the sending module.
01	031F	No connection resources exist for the target path.
01	0320-07FF	Vendor specific.

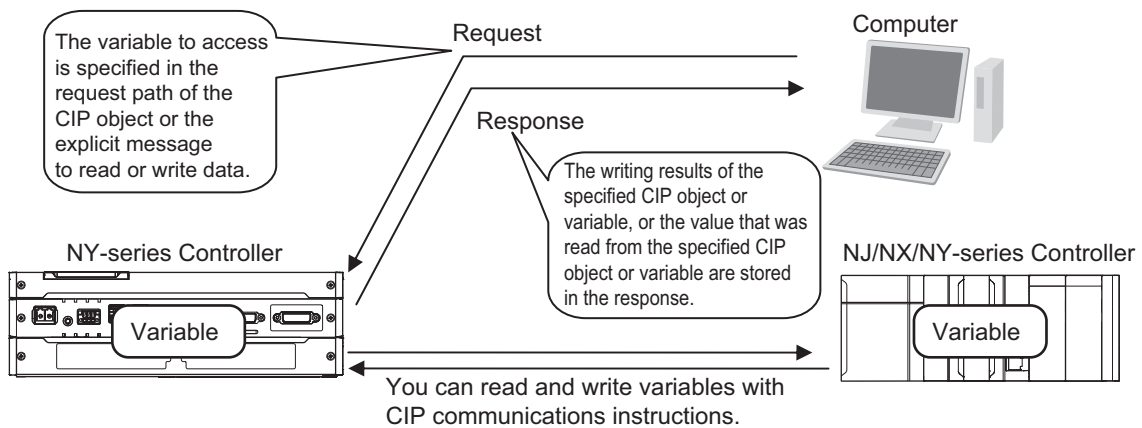
8-3 CIP Communication Server Functions

The CIP Communication Server functions are exclusively available for the NY-series Controllers. These functions execute services for a specified self-contained object in the Controller after receiving the CIP messages from external devices.

This section provides information on CIP message structure along with information about how to use the CIP message communications server functions to read and write the values of variables in programs such as the following:

- Programs running on a computer on the EtherNet/IP network
- Windows programs installed in an NY-series Industrial PC

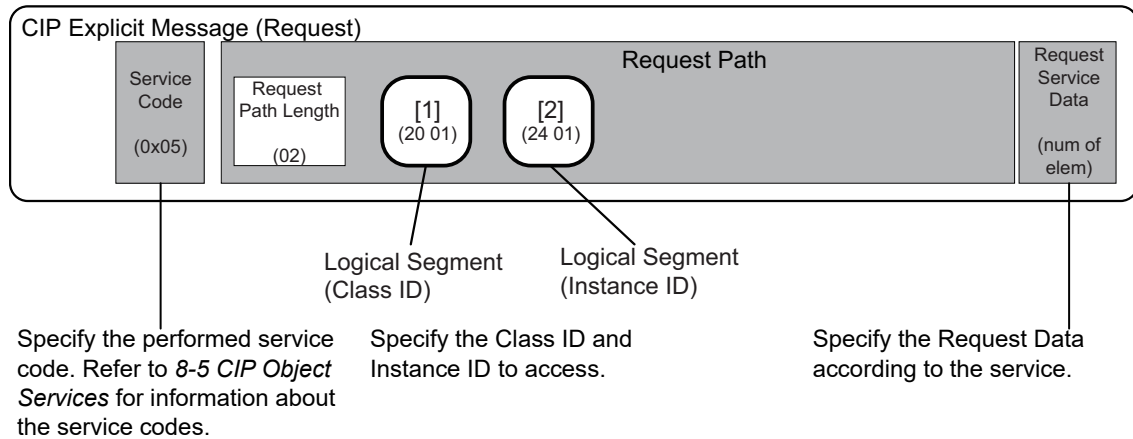
To read and write CIP objects or the values of variables between NJ/NX/NY-series Controllers, use the CIP communications instructions. Refer to *8-2 CIP Messaging Communications Client Function* for information about how to use CIP communications instructions for CIP message communications.



8-3-1 CIP Message Structure for Accessing CIP Objects

This section shows how to specify messages to access CIP objects. The CIP objects to access are expressed by connecting the segments that are defined in the CIP Common specifications in the request path field in a CIP explicit message.

Example: Performing the Reset service (0x05) to the Instance (01 hex) of the Identity object (class: 01 hex)



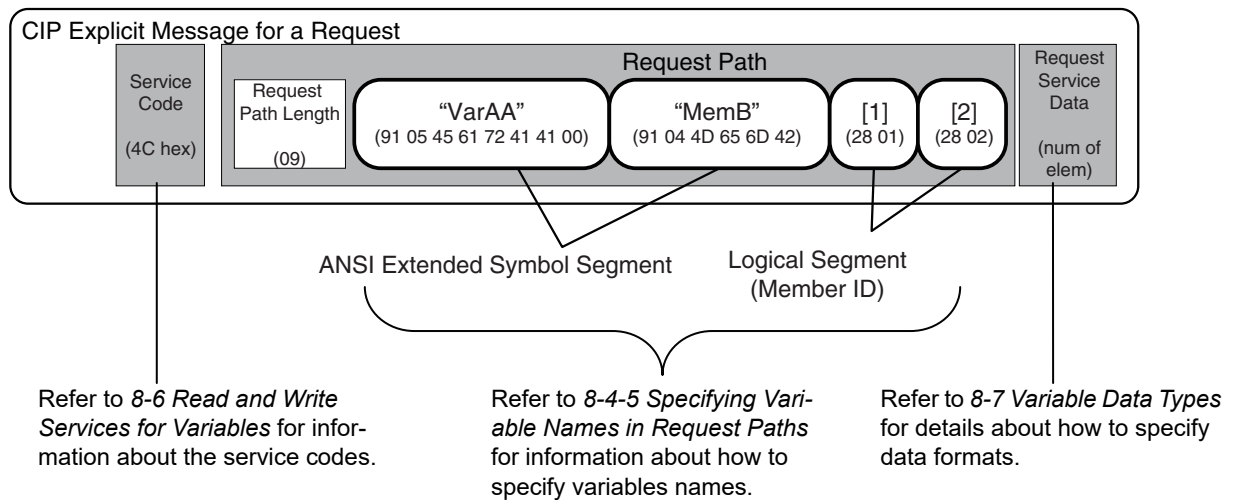
8-3-2 CIP Message Structure for Accessing Variables

This section shows how to specify messages to access variables. The variables to access are given by connecting the segments that are defined in the CIP Common specifications so that explicit message can be set in the request path field.

The following elements are combined to make the specification.

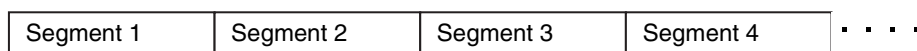
Specifying the variable to access: The elements are stored in the CIP segments and then joined to make the message.

Example: Reading the Present Value of One Member of the *VarAA.MemB[1.2]* Structure Variable
 Example for Using the CIP Read Data Service for a Variable Object

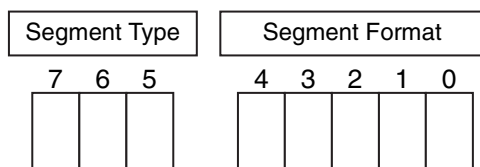


8-4 Specifying Request Path

The CIP object, variable name, structure member name, and array index are specified for the request path. In CIP, the EPATH data type is used for the request path. With this method, the request path is divided into segments and a value is assigned to each segment. The request path notation shows the path to the final destination when the data segments are joined together. Each segment includes the segment type information and the segment data.



The first byte gives the interpretation method for the segment. It consists of two parts; a 3-bit segment type and a 5-bit segment format.



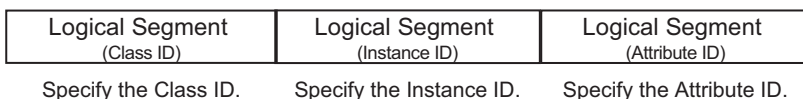
The segment type specifications are defined as follows in the CIP specifications.

Segment Type			Meaning
7	6	5	
0	0	0	Port Segment
0	0	1	Logical Segment
0	1	0	Network Segment
0	1	1	Symbolic Segment
1	0	0	Data Segment
1	0	1	Data Type
1	1	0	Data Type
1	1	1	Reserved

The specifications of segment format are different for each segment type. Use the segment format to request a service from a particular object of a particular device. Logical segments and data segments, which are needed to specify variables in CIP message communications, are described below.

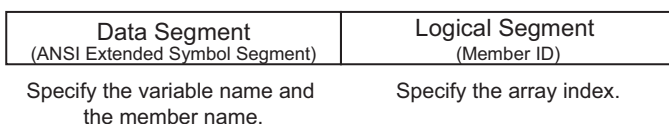
8-4-1 Examples of CIP Object Specifications

Logical Segments are joined to form the request path that specifies the object to access.



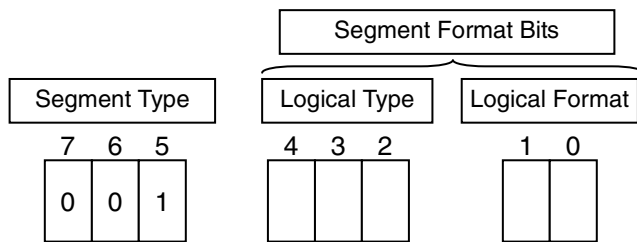
8-4-2 Examples of Variable Specifications

Segments are joined to form the request path that specifies the variable to access.



8-4-3 Logical Segment

A logical segment is used to give the range of the CIP Object or variable (array) in the request path.



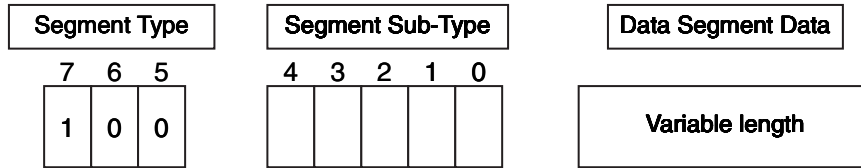
Logical Type			Meaning
4	3	2	
0	0	0	Class ID
0	0	1	Instance ID
0	1	0	Member ID
0	1	1	Connection Point
1	0	0	Attribute ID
1	0	1	Special (Do not use the logical addressing definition for the Logical Format.)
1	1	0	Service ID (Do not use the logical addressing definition for the Logical Format.)
1	1	1	Reserved

Logical Format		Meaning
1	0	
0	0	8 bit logical address
0	1	16 bit logical address
1	0	32 bit logical address
1	1	Reserved

An 8-bit or 16-bit logical address can be used for the class ID and attribute ID.
An 8-bit, 16-bit, or 32-bit logical address can be used for the instance ID.

8-4-4 Data Segment

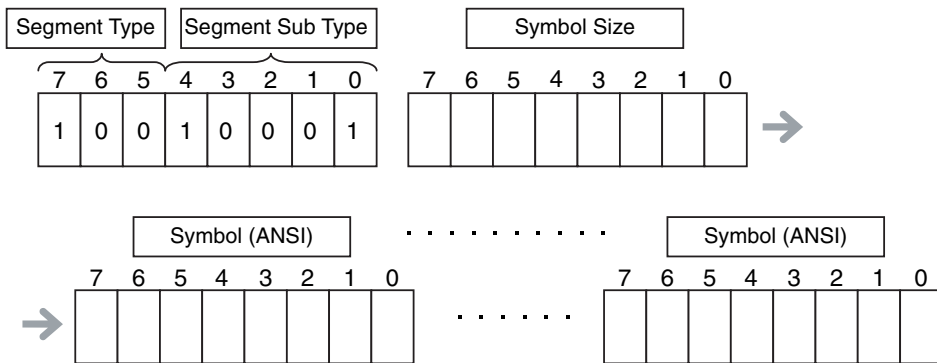
A data segment is used to give the specified variable name in the request path.



Segment Sub-Type					Meaning
4	3	2	1	0	
0	0	0	0	0	Simple Data Segment
1	0	0	0	1	ANSI Extended Symbol Segment

A data segment is mainly used for an ANSI extended symbol segment. This segment sub-type is used to read and write the values of variables.

ANSI Extended Symbol Segment



8-4-5 Specifying Variable Names in Request Paths

Variable Names

A variable name is specified as a symbolic segment (ANSI extended symbol segment).

Variable Name Specification Format

BYTE	91 hex	ANSI Extended Symbol Segment
BYTE	Length in BYTE	Length of variable name in bytes
Array of octet	: <i>Variable_name</i>	Variable name encoded in UTF-8
	:	
Octet	(pad)	00 hex. One byte is padded if the variable name length is an odd number of bytes.

Variable Names

Variable names are encoded in UTF-8.

Structure Member Names

Structure member names are specified in the same way as variable names. (Store UTF-8 character codes in the ANSI extended symbol segment.)

Array Indices

Specify the array index in a logical segment that is set as a member ID. You can specify an array index ([x]) in a variable name.

Specification Method 1: 8-bit Index

BYTE	28 hex	Logical Segment (Member ID)
USINT	Index	Array index from 0 to 255

Specification Method 2: 16-bit Index

BYTE	29 hex	Logical Segment (Member ID)
octet	00 hex	Pad
UINT	Index (L) ----- (H)	Array index from 0 to 65,535

Range Specifications with the Num of Element Field

There is a Num of Element field in the request data for the variable read and variable write services. You can use these services to access the specified range of an array with the following specifications.

- Specify the first element in the range of elements to access in the array variable as the variable to read or write.
- Specify the number of elements to access in the Num of Element field.

Specification Examples

This example shows how to specify *VarAA.MemB[1.2]* for the following structure variable.

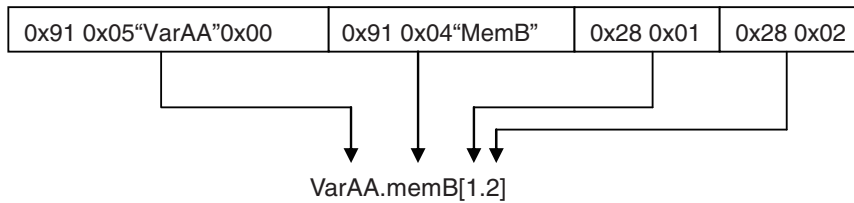
```
struct
{
    UINT      MemA;
    BOOL      MemB[10][10];
} VarAA;
```

Variable Name Specification Format

BYTE	91 hex	ANSI Extended Symbol Segment
BYTE	05 hex	Length of variable name in bytes
Array of octet	'V'	Variable name
	'a'	
	'r'	
	'A'	
	'A'	
Octet	00 hex	Pad
BYTE	91 hex	ANSI Extended Symbol Segment
BYTE	04 hex	Length of variable name in bytes
Array of octet	'M'	Variable name
	'e'	
	'm'	
	'B'	
BYTE	28 hex	Logical Segment (Member ID)
USINT	01 hex	Array index for first element
BYTE	28 hex	Logical Segment (Member ID)
USINT	02 hex	Array index for second element

The variable name that is specified in the symbolic segment (ANSI extended symbol segment) must be converted to a text string to pass it to the communications thread. The following conversion rules apply.

Specification Example for Structure Members and Array Elements



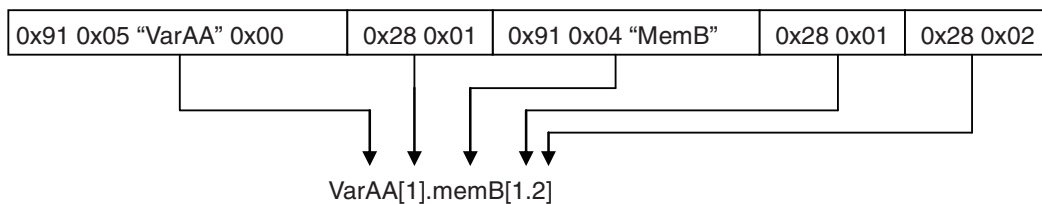
This example shows how to specify *VarAA[1].MemB[1.2]* for the following structure variable.

```
struct
{
    UINT      MemA;
    BOOL      MemB[10][10];
} VarAA[3];
```

Variable Name Specification Format

BYTE	91 hex	ANSI Extended Symbol Segment
BYTE	05 hex	Length of variable name in bytes
Array of octet	'V'	Variable name
	'a'	
	'r'	
	'A'	
	'A'	
Octet	00 hex	Pad
BYTE	28 hex	Logical Segment (Member ID)
USINT	01 hex	Array index
BYTE	91 hex	ANSI Extended Symbol Segment
BYTE	04 hex	Length of variable name in bytes
Array of octet	'M'	Variable name
	'e'	
	'm'	
	'B'	
BYTE	28 hex	Logical Segment (Member ID)
USINT	01 hex	Array index for first element
BYTE	28 hex	Logical Segment (Member ID)
USINT	02 hex	Array index for second element

Specification Example for Structure Array



8-5 CIP Object Services

This section shows services that specify the CIP object in the Request Path and access the CIP message server function of the NY-series Controllers.

8-5-1 CIP Objects Sent to the Built-in EtherNet/IP Port

The following CIP objects can be sent to an EtherNet/IP port.

Object name	Function	Reference
Identity object	<ul style="list-style-type: none"> Reads ID information from the Controller. Resets the built-in EtherNet/IP port and internal port. 	page 8-40
TCP/IP interface object	<ul style="list-style-type: none"> Writes and reads TCP/IP settings. 	page 8-44
Ethernet link object	<ul style="list-style-type: none"> Reads Ethernet settings. Reads Ethernet status. 	page 8-47
Controller object	<ul style="list-style-type: none"> Gets the Controller status. Changes the operating mode of the Controller. 	page 8-52

8-5-2 Identity Object (Class ID: 01 Hex)

This object reads the ID information of the Controller and resets the built-in EtherNet/IP port and internal port.

Service Codes

Specify the service to execute with the service code.

Service code	Parameter name	Description	Supported services	
			Classes	Instances
01 hex	Get_Attribute_All	Reads the values of the attributes.	Supported.	Supported.
0E hex	Get_Attribute_Single	Reads the value of the specified attribute.	Supported.	Supported.
05 hex	Reset	<p>Resets the built-in EtherNet/IP port and internal port. This parameter is used to reset the port when you change the IP address or other parameter settings and want to apply them. Input one of the following values for the <i>ServiceDat</i> input variable to the CIPSend instruction to specify the reset method.</p> <p>00 hex: Resets the port. 02 hex: Clears the saved tag data link settings and resets the port.</p>	Not supported.	Supported.

Class ID

Specify 01 hex.

Instance ID

Specify 00 or 01 hex.

Attribute ID

The attribute ID specifies the information to read.

● Class Attribute ID

The class attribute ID specifies the attribute of the entire object.

Attribute ID	Parameter name	Description	Attribute	Read data	
				Data type	Value
01 hex	Revision	Revision of the object	Read	UINT	0001 hex
02 hex	Max Instance	The maximum instance number	Read	UINT	0001 hex

● Instance Attribute ID

The instance attribute ID specifies the attribute of the instance.

Attribute ID	Parameter name	Description	Attribute	Read data	
				Data type	Value
01 hex	Vendor ID	Vendor ID	Read	UINT	002F hex
02 hex	Device Type	Device type	Read	UINT	000C hex
03 hex	Product Code	Product code	Read	UINT	Refer to (1) <i>Product Codes for Each Model</i> , below
04 hex	Revision	Device revision	Read	Struct	---
	Major Revision	Major revision	Read	USINT	Refer to (2) <i>Major and Minor CIP Revisions for Each Unit Version</i> , below
	Minor Revision	Minor revision	Read	USINT	
05 hex	Status	Status of the built-in EtherNet/IP port	Read	WORD	Refer to (3) <i>Status Details of the Built-in EtherNet/IP Port</i> , below
06 hex	Serial Number	Serial number	Read	UDINT	Set value
07 hex	Product Name	Product name	Read	STRING	Set value

(1) Product Codes for Each Model

Model	Product Code
NY512-□□□□	0685 hex
NY532-□□□□	0686 hex

(2) Major and Minor CIP Revisions for Each Unit Version

Unit version	CIP revisions	
	Major revision	Minor revision
Unit version 1.12 to 1.19	02 hex	05 hex
Unit version 1.21 or later		06 hex

(3) Status Details of the Built-in EtherNet/IP Port

Bit	Name	Description																														
0	Owned	Indicates when the built-in EtherNet/IP port has an open connection as the target of a tag data link.																														
1	Reserved	Always FALSE.																														
2	Configured	Tag data link settings exist.																														
3	Reserved	Always FALSE.																														
4 to 7	Extended Device Status	Indicates the status of the built-in EtherNet/IP port. <table border="1" data-bbox="619 517 1433 786"> <thead> <tr> <th>b7</th> <th>b6</th> <th>b5</th> <th>b4</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>A major fault occurred.</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>A timeout occurred in one or more target connections.</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>Indicates that there are no tag data link settings.</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>Indicates that one or more connections are performing communications normally.</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>Other than the above.</td> </tr> </tbody> </table>	b7	b6	b5	b4		0	1	0	1	A major fault occurred.	0	0	1	0	A timeout occurred in one or more target connections.	0	0	1	1	Indicates that there are no tag data link settings.	0	1	1	0	Indicates that one or more connections are performing communications normally.	0	1	1	1	Other than the above.
b7	b6	b5	b4																													
0	1	0	1	A major fault occurred.																												
0	0	1	0	A timeout occurred in one or more target connections.																												
0	0	1	1	Indicates that there are no tag data link settings.																												
0	1	1	0	Indicates that one or more connections are performing communications normally.																												
0	1	1	1	Other than the above.																												
8	Minor Recoverable Fault	TRUE when any of the following errors occurs. <ul style="list-style-type: none"> • IP Route Table Setting Error • DNS Server Connection Failed • Tag Data Link Setting Error • Tag Data Link Timeout • Tag Data Link Connection Timeout • FTP Server Setting Error • NTP Client Setting Error • SNMP Setting Error • NTP Server Connection Failed • Tag Name Resolution Error 																														
9	Minor Unrecoverable Fault	TRUE when the following error occurs. <ul style="list-style-type: none"> • Identity Error 																														
10	Major Recoverable Fault	TRUE when any of the following errors occurs. <ul style="list-style-type: none"> • IP Address Duplication Error • BOOTP Server Connection Error • Basic Ethernet Setting Error • IP Address Setting Error 																														
11	Major Unrecoverable Fault	TRUE when any of the following errors occurs. <ul style="list-style-type: none"> • Communications Controller Failure • MAC Address Error 																														
12 to 15	Reserved	Always FALSE.																														

Request Paths (IOIs) to Specify Objects

When you specify an object, specify the request path (IOI) for each service code as given below.

Service code		Class ID	Instance ID	Attribute ID
01 hex	Get_Attribute_All	01 hex	<ul style="list-style-type: none"> Specifying a service for a class: 00 hex Specifying a service for an instance: Always 01 hex 	Not required.
0E hex	Get_Attribute_Single			<ul style="list-style-type: none"> Reading a class attribute: 01 or 02 hex Reading an instance attribute: 01 to 07 hex
05 hex	Reset			Always 01 hex

8-5-3 TCP/IP Interface Object (Class ID: F5 hex)

This object is used to read and write settings such as the IP address, subnet mask, and default gateway.

Service Codes

Specify the service to execute with the service code.

Service code	Parameter name	Description	Supported services	
			Classes	Instances
01 hex	Get_Attribute_All	Reads the values of the attributes.	Supported.	Not supported.
0E hex	Get_Attribute_Single	Reads the value of the specified attribute.	Supported.	Supported.
10 hex	Set_Attribute_Single	Writes a value to the specified attribute. The built-in EtherNet/IP port restarts automatically after the value is written to the attribute. When the restart process is not completed and the next Set_Attribute_Single is executed, the general status "0C hex" (Object State Conflict) is returned.	Not supported.	Supported.

Class ID

Specify F5 hex.

Instance ID

Specify 00 to 02 hex.

00: Specify the class

01: Built-in EtherNet/IP port

02: Internal port

Attribute ID

The attribute ID specifies the information to read.

● Class Attribute ID

The class attribute ID specifies the attribute of the entire object.

Attribute ID	Parameter name	Description	Attribute	Read data	
				Data type	Value
01 hex	Revision	Revision of the object	Read	UINT	0004 hex
02 hex	Max Instance	The maximum instance number	Read	UINT	0002 hex
03 hex	Num of Instance	The number of object instance	Read	UINT	0002 hex

● Instance Attribute ID

The instance attribute ID specifies the attribute of the instance.

Attribute ID	Parameter name	Description	Attribute	Read/write data	
				Data type	Value
01 hex	Interface Configuration Status	Indicates the IP address settings status of the interface.	Read	DWORD	Bits 0 to 3: Interface Configuration Status: 0 = IP address is not set. (This includes when BOOTP is starting.) 1 = IP address is set. Bits 4 and 5: Reserved (always FALSE). Bit 6: AcdStatus: FALSE = IP address collisions have not been detected. TRUE = IP address collisions have been detected. Bits 7 to 31: Reserved (always FALSE).
02 hex	Configuration Capability	Indicates a Controller Configurations and Setup that can be set to the interface.	Read	DWORD	Bit 0: BOOTP Client: FALSE = Internal port TRUE = Built-in EtherNet/IP port Bit 1: DNS Client: Always TRUE. Bit 2: DHCP Client: Always FALSE. Bit 3: DHCP-DNS Update: Always FALSE. Bit 4: Configuration Settable: Always TRUE. Bit 5: Hardware Configurable: Always FALSE. Bit 6: Interface Configuration Change Requires Reset: Always FALSE. Bit 7: ACD Capable: Always TRUE. Bits 8 to 31: Reserved (always FALSE).
03 hex	Configuration Control	Sets the method used to set the IP address when the interface starts.	Read/Write	DWORD	Bits 0 to 3: IP Address Setting Method: 0 = Setting the static IP address. 1 = Setting by BOOTP. Bits 4: DNS Enable/Disable Setting: FALSE = DNS disabled. TRUE = DNS enabled. Bits 5 to 31: Reserved (always FALSE).
04 hex	Physical Link Object	The path to the link object in the physical layer.	Read	Struct	---
	Path size	The path size (WORD size).		UINT	0002 hex
	Path	The path to the link object in the physical layer (static).		EPATH	20 F6 24 02 hex = Internal port 20 F6 24 01 hex = Built-in EtherNet/IP port
05 hex	Interface Configuration	The interface settings.	Read/Write	Struct	---
	IP Address	IP address.		UDINT	Set value
	Network Mask	Subnet mask.		UDINT	Set value
	Gateway Address	The default gateway.		UDINT	Set value
	Nama Server	The primary name server.		UDINT	Set value
	Nama Server2	The secondary name server.		UDINT	Set value
	Domain Name	The domain name.		STRING	Set value*1
06 hex	Host Name	The host name (reserved).	Read/Write	STRING	Set value*2

*1 The value is the size of domain name (2 bytes) + domain name (48 bytes max.).

*2 The value is the size of host name (2 bytes) + host name (64 bytes max.).

Request Paths (IOIs) to Specify Objects

When you specify an object, specify the request path (IOI) for each service code as given below.

Service code		Class ID	Instance ID	Attribute ID
01 hex	Get_Attribute_All	F5 hex	<ul style="list-style-type: none"> Specifying a service for a class: 00 hex Specifying a service for an instance: 01 or 02 hex 	Not required.
0E hex	Get_Attribute_Single			<ul style="list-style-type: none"> Reading a class attribute: 01 or 02 hex
10 hex	Set_Attribute_Single			<ul style="list-style-type: none"> Reading and writing an instance attribute: 01 to 06 hex

8-5-4 Ethernet Link Object (Class ID: F6 Hex)

This object is used to set and read Ethernet communications and read Ethernet communications status information.

Service Codes

Specify the service to execute with the service code.

Service code	Parameter name	Description	Supported services	
			Classes	Instances
0E hex	Get_Attribute_Single	Reads the value of the specified attribute.	Supported.	Supported.
10 hex	Set_Attribute_Single	Writes a value to the specified attribute.	Supported.	Supported.
4C hex	Get_and_Clear	Specify <i>Attribute4</i> or <i>Attribute5</i> to reset the value of the attribute to 0.	Not supported.	Supported.

Class ID

Specify F6 hex.

Instance ID

Specify 00 to 02 hex.

00: Specify the class

01: Built-in EtherNet/IP port

02: Internal port

Attribute ID

The attribute ID specifies the information to read.

● Class Attribute ID

The class attribute ID specifies the attribute of the entire object.

Attribute ID	Parameter name	Description	Attribute	Read data	
				Data type	Value
01 hex	Revision	Revision of the object	Read	UINT	0004 hex
02 hex	Max Instance	The maximum instance number	Read	UINT	0002 hex
03 hex	Num of Instance	The number of object instance	Read	UINT	0002 hex

● Instance Attribute ID

The instance attribute ID specifies the attribute of the instance.

Attribute ID	Parameter name	Description	Attribute	Read/write data	
				Data type	Value
01 hex	Interface Speed	Gives the baud rate for the interface.	Read	UDINT	Reads the current value.
02 hex	Interface Flags	Gives the status of the interface.	Read	DWORD	Refer to (1) <i>Interface Flag Details</i> , below.

Attribute ID	Parameter name	Description	Attribute	Read/write data	
				Data type	Value
03 hex	Physical Address	Gives the MAC address of the interface.	Read	ARRAY [0...5] OF USINT	Reads the current value of the MAC address.
04 hex	Interface Counters	The number of packets sent/received through the interface.	Read	Struct	---
	In Octets	The number of octets received through the interface. This includes unnecessary multicast packets and discarded packets counted by <i>InDiscards</i> .		UDINT	Reads the current value.
	In Unicast Packets	The number of unicast packets received through the interface. This does not include discarded packets counted by <i>InDiscards</i> .		UDINT	Reads the current value.
	In NonUnicast Packets	The number of packets besides unicast packets received through the interface. This includes unnecessary multicast packets, but does not include discarded packets counted by <i>InDiscards</i> .		UDINT	Reads the current value.
	In Discards	The number of discarded incoming packets received through the interface.		UDINT	Reads the current value.
	In Errors	The number of incoming packets that had errors. This is not included in <i>InDiscards</i> .		UDINT	Reads the current value.
	In Unknown Protos	The number of incoming packets that were of an unknown protocol.		UDINT	Reads the current value.
	Out Octets	The number of octets sent through the interface.		UDINT	Reads the current value.
	Out Unicast Packets	The number of unicast packets sent through the interface.		UDINT	Reads the current value.
	Out NonUnicast Packets	The number of packets besides unicast packets sent through the interface.		UDINT	Reads the current value.
	Out Discards	The number of discarded sent packets.		UDINT	Reads the current value.
	Out Errors	The number of sent packets that had errors.		UDINT	Reads the current value.

Attribute ID	Parameter name	Description	Attribute	Read/write data	
				Data type	Value
05 hex	Media Counters	Media counters for the communications port.	Read	Struct	---
	Alignment Errors	Number of frames received that were not octets in length.		UDINT	Reads the current value.
	FCS Errors	Number of frames received that did not pass the FCS check.		UDINT	Reads the current value.
	Single Collisions	Number of frames sent successfully with only one collision.		UDINT	Reads the current value.
	Multiple Collisions	Number of frames sent successfully with two or more collisions.		UDINT	Reads the current value.
	SQE Test Errors	Number of times a SQE test error message was generated.		UDINT	Reads the current value.
	Deferred Transmissions	The number of frames for which the first attempt to send was delayed because the media was busy.		UDINT	Reads the current value.
	Late Collisions	The number of collisions detected in packets that were sent after 512 bit times.		UDINT	Reads the current value.
	Excessive Collisions	The number of frames that failed to be sent because of excessive collisions.		UDINT	Reads the current value.
	MAC Transmit Errors	The number of frames that failed to be sent due to an internal MAC sublayer transmission error.		UDINT	Reads the current value.
	Carrier Sense Errors	The number of times the carrier sense condition was lost or the number of times an assertion did not occur when an attempt was made to send the frame.		UDINT	Reads the current value.
	Frame Too Long	The number of frames received that exceeded the maximum allowed frame size.		UDINT	Reads the current value.
	MAC Receive Errors	The number of frames that could not be received through the interface due to an internal MAC sublayer reception error.		UDINT	Reads the current value.
06 hex	Interface Control	Control settings for the interface.	Read/Wri te	Struct	---
	Control Bits	<i>Auto Nego</i> for Ethernet communications that specifies full duplex.		WORD	Refer to (2) <i>Control Bit Details</i> , below.
	Forced Interface Speed	Gives the set value of the Ethernet baud rate.		UINT	Reads the set value.

Attribute ID	Parameter name	Description	Attribute	Read/write data	
				Data type	Value
0C hex	HC Interface Counters	The number of packets sent/received through the HC interface.	Read	Struct	---
	HCInOctets	The number of octets received through the interface. This counter is the 64-bit edition of In Octets.		ULINT	Reads the current value.
	HCInUnicastPkts	The number of unicast packets received through the interface. This counter is the 64-bit edition of In Ucast Packets.		ULINT	Reads the current value.
	HCInMulticastPkts	The number of multicast packets received through the interface.		ULINT	Reads the current value.
	HCInBroadcastPkts	The number of broadcast packets received through the interface.		ULINT	Reads the current value.
	HCOctets	The number of octets sent through the interface. This counter is the 64-bit edition of Out Octets.		ULINT	Reads the current value.
	HCOctetsUnicastPkts	The number of unicast packets sent through the interface.		ULINT	Reads the current value.
	HCOctetsMulticastPkts	The number of multicast packets sent through the interface.		ULINT	Reads the current value.
	HCOctetsBroadcastPkts	The number of broadcast packets sent through the interface.		ULINT	Reads the current value.
0D hex	HC Media Counters	Media counters for the communications port.	Read	Struct	---
	HCStatsAlignmentErrors	Number of frames received that were not octets in length. This counter is the 64-bit edition of Alignment Errors.		ULINT	Reads the current value.
	HCStatsFCSErrors	Number of frames received that did not pass the FCS check. This counter is the 64-bit edition of FCS Errors.		ULINT	Reads the current value.
	HCStatsInternalMac TransmitErrors	The number of frames that failed to be sent due to an internal MAC sublayer transmission error. This counter is the 64-bit edition of MAC Transmit Errors.		ULINT	Reads the current value.
	HCStatsFrameTooLongs	The number of frames received that exceeded the maximum allowed frame size. This counter is the 64-bit edition of Frame Too Long.		ULINT	Reads the current value.
	HCStatsMacReceiveErrors	The number of frames that could not be received through the interface due to an internal MAC sublayer reception error. This counter is the 64-bit edition of MAC Receive Errors.		ULINT	Reads the current value.
	HCStatsMacSymbolErrors	The number of frames that could not be received through the interface due to an internal MAC sublayer symbol error.		ULINT	Reads the current value.

(1) Interface Flag Details

Bit	Name	Description
0	LinkStatus	FALSE: The link is down. TRUE: The link is up.
1	Half/FullDuplex	FALSE: Half duplex TRUE: Full duplex
2 to 4	Negotiation Status	00 hex: Auto-negotiation is in progress. 01 hex: Auto-negotiation and speed detection failed. 02 hex: Auto-negotiation failed, but speed detection succeeded. 03 hex: Speed and duplex mode negotiation succeeded. 04 hex: Auto-negotiation was not attempted.
5	Manual Setting Requires Speed	Always FALSE: Changes can be applied automatically.
6	Local Hardware Fault	Always FALSE
7 to 31	Reserved	Always FALSE

(2) Control Bit Details

Bit	Name	Description
0	Auto-negotiate	FALSE: Auto-negotiation is disabled. TRUE: Auto-negotiation is enabled.
1	ForcedDuplex Mode	FALSE: Half duplex TRUE: Full duplex * When auto-negotiation is enabled (bit 0 is TRUE), this should always be FALSE.
2 to 16	Reserved	Always FALSE

Request Paths (IOIs) to Specify Objects

When you specify an object, specify the request path (IOI) for each service code as given below.

Service code	Class ID	Instance ID	Attribute ID
0E hex	F6 hex	<ul style="list-style-type: none"> Specifying a service for a class: 00 hex Specifying a service for an instance: 01 or 02 hex 	<ul style="list-style-type: none"> Reading a class attribute: 01 to 03 hex Reading and writing an instance attribute: 01 to 06 hex, 0C hex or 0D hex
10 hex			
4C hex			

8-5-5 Controller Object (Class ID: C4 Hex)

This object is used to get the status of the Controller or to change the operating mode of the Controller.

Service Codes

Specify the service to execute with the service code.

Service code	Parameter name	Description	Supported services	
			Classes	Instances
0E hex	Get_Attribute_Single	Reads the value of the specified attribute.	Supported.	Not supported.
10 hex	Set_Attribute_Single	Writes a value to the specified attribute.	Supported.	Not supported.
51 hex	Reset_System_Alarm_All	Resets all errors in the Controller.	Supported.	Not supported.

Class ID

Specify C4 hex.

Instance ID

Specify 00 hex.

● Class Attribute ID

The class attribute ID specifies the attribute (value) of the entire object.

Attribute ID	Parameter name	Description	Attribute	Read/write data	
				Data type	Value
01 hex	Revision	Revision of the object	Read	UINT	Always 0002 hex.
02 hex	Max Instance	The maximum instance number	Read	UINT	Always 0001 hex
64 hex	PLC Mode	This can be used to read and modify the Controller operating mode.	Read/Wri te	UINT	Specify this when you want to write to an attribute. 0000 hex: PROGRAM mode 0004 hex: RUN mode
65 hex	PLC Error Status	Indicates when there is a Controller error. Changes to TRUE when a fatal or non-fatal error occurs.	Read	UINT	0000 hex: There is no Controller error. 0001 hex: There is a Controller error.
66 hex	PLC Model	Indicates the model of the Controller. The length is always 2 bytes for the size + 20 bytes for the name. Unused area is padded with spaces.	Read	STRING	

● Instance Attribute ID

None

Request Paths (IOIs) to Specify Objects

When you specify an object, specify the request path (IOI) for each service code as given below.

Service code	Class ID	Instance ID	Attribute ID
0E hex	C4 hex	00 hex	Specifies the attribute of the class to read or write: 01 hex, 02 hex, or 64 to 66 hex
10 hex			

8-6 Read and Write Services for Variables

This section shows services that specify the CIP object in the Request Path and access the CIP message server function of the NY-series Controllers.

8-6-1 Read Services for Variables

Specify service code 4C hex to read the value of the variable that is specified by the request path.

Service Code: 4C Hex

● Request Data Format

Request Path Data
[Variable name specification]

Request Service Data			
+0	Num of Element (L)	UINT	Specify the number of elements to read for an array variable.
+1	(H)		

Response Service Data			
	Data Type	USINT	Data type of variable to read
	AddInfo Length	USINT	Additional information: Field length in bytes
	(AddInfo)		Additional information: CRC value of structure
	Actual data*		

* The actual data is stored in little-endian format.

Data Type	Code for data type of variable to read Refer to 8-7-1 Data Type Codes.
AddInfoLength	The size of the AddInfo area is stored only when accessing a structure variable. Set 02 hex for a structure variable. Otherwise, set 00 hex.
AddInfo	The CRC code of the structure definition is stored only when accessing a structure variable. In this case, the size of AddInfo will be 2 bytes.
Actual data	The actual data is stored in little-endian format. If 0001 hex is specified for an array, the actual data is stored in the same format as when you access a variable with the data type of the elements of the array.

Response Codes

CIP status	Meaning	Add status	Cause
00	SUCCESS	---	The service ended normally.
02	RESOURCE_UNAVAILABLE	---	The internal processing buffer is not available.
04	PATH_SEGMENT_ERROR	---	The request path specification is not correct.
05	PATH_DESTINATION_UNKNOWN	---	The variable specification is not correct.
0C	OBJECT_STATE_CONFLICT	8010	Downloading, starting up
		8011	There is an error in tag memory.
11	REPLY_DATA_TOO_LARGE	---	The response exceeds the maximum response length.
13	NOT_ENOUGH_DATA	---	The data length is too short for the specified service.
15	TOO_MUCH_DATA	---	The data length is too long for the specified service.
1F	VENDOR_SPECIFIC_ERROR	0102, 2104	An attempt was made to read an I/O variable that cannot be read.
		0104, 1103	The specified address and size exceed a segment boundary.
		8001	An internal error occurred.
		8007	An inaccessible variable was specified.
		8031	An internal error occurred. (A memory allocation error occurred.)
20	INVALID_PARAMETER	8009	A segment type error occurred.
		800F	There is an inconsistency in data length information in the request data.
		8017	More than one element was specified for a variable that does not have elements.
		8018	Zero elements or data that exceeded the range of the array was specified for an array.
		8023	An internal error occurred. (An illegal command format was used.)
		8024	An internal error occurred. (An illegal command length was used.)
		8025	An internal error occurred. (An illegal parameter was used.)
		8027	An internal error occurred. (A parameter error occurred.)
		8028	<ul style="list-style-type: none"> • An attempt was made to write an out-of-range value for a variable for which a sub-range is specified. • An attempt was made to write an undefined value to an enumeration variable.

8-6-2 Write Service for Variables

Specify service code 4D hex to write the value of the variable that is specified by the request path.

Request Data Format for Writing a Variable

Request Path Data
[Variable name specification]

Request Service Data			
Data Type		USINT	Data type of variable to write
AddInfo Length		USINT	Additional information: Field length in bytes
(AddInfo)			Additional information: CRC value of structure
Num of Element	(L)	UINT	
	(H)		
Actual data*			

Response Service Data
There is no response service data.

* Data to write: Store the data to write in little-endian format

Data Type	Code for data type of variable to write Refer to 8-7 Variable Data Types.
AddInfoLength	Specify the size of the AddInfo area only when accessing a structure variable. Set 02 hex for a structure variable. Otherwise, set 00 hex.
AddInfo	The CRC code of the structure definition is specified only when accessing a structure variable. In this case, the size of AddInfo will be 2 bytes.
NumOfElement	Specifying the number of elements in the array. Do not specify 0000 hex (an error will occur). For variables other than arrays, set 0001 hex.
Actual data	Specify the actual data in little-endian format. If 0001 hex is specified for an array, specify the actual data in the same format as when you access a variable with the data type of the elements of the array.

Response Codes

CIP status	Meaning	Add status	Cause
00	SUCCESS	---	The service ended normally.
02	RESOURCE_UNAVAILABLE	---	The internal processing buffer is not available.
04	PATH_SEGMENT_ERROR	---	The request path specification is not correct.
05	PATH_DESTINATION_UNKNOWN	---	The link was followed to the end, but the variable was not found.
0C	OBJECT_STATE_CONFLICT	8010	Downloading, starting up
		8011	There is an error in tag memory.
13	NOT_ENOUGH_DATA	---	The data length was too short for the specified service.
15	TOO_MUCH_DATA	---	The data length was too long for the specified service.
1F	VENDOR_SPECIFIC_ERROR	0102, 2103	An attempt was made to write a constant or read-only variable.
		0104, 1103	The specified address and size exceed a segment boundary.
		8001	An internal error occurred. (An information inconsistency was detected in the interface in the Module.)
		8007	An inaccessible variable was specified.
		8029	A region that all cannot be accessed at the same time was specified for <i>Simple-DataSegment</i> .
		8031	An internal error occurred. (A memory allocation error occurred.)
20	INVALID_PARAMETER	8009	A segment type error occurred.
		800F	There is an inconsistency in data length information in the Request Data.
		8017	More than one element was specified for a variable that does not have elements.
		8018	Zero elements or data that exceeded the range of the array was specified for an array.
		8021	A value other than 0 or 2 was specified for an AddInfo area.
		8022	The data type that is specified in the request service data does not agree with the tag information. The AddInfo Length in the request service data is not 0.
		8023	An internal error occurred. (An illegal command format was used.)
		8024	An internal error occurred. (An illegal command length was used.)
		8025	An internal error occurred. (An illegal parameter was used.)
		8027	An internal error occurred. (A parameter error occurred.)
		8028	<ul style="list-style-type: none"> • An attempt was made to write an out-of-range value for a variable for which a subrange is specified. • An attempt was made to write an undefined value to an enumeration variable.

8-7 Variable Data Types

This section provides the data types of variables that can be used with CIP message communications.

8-7-1 Data Type Codes

The following codes are given to variable data types.

Data Type	Code(Hex)	Group*
Boolean (bit)	C1	CIP Common
SINT (1-byte signed binary)	C2	CIP Common
INT (1-word signed binary)	C3	CIP Common
DINT (2-word signed binary)	C4	CIP Common
LINT (4-word signed binary)	C5	CIP Common
USINT (1-byte unsigned binary)	C6	CIP Common
UINT (1-word unsigned binary)	C7	CIP Common
UDINT (2-word unsigned binary)	C8	CIP Common
ULINT (4-word unsigned binary)	C9	CIP Common
REAL (2-word floating point)	CA	CIP Common
LREAL (4-word floating point)	CB	CIP Common
STRING	D0	CIP Common
BYTE (1-byte hexadecimal)	D1	CIP Common
WORD (1-word hexadecimal)	D2	CIP Common
DWORD (2-word hexadecimal)	D3	CIP Common
TIME (8-byte data)	DB	CIP Common
LWORD (4-word hexadecimal)	D4	CIP Common
Abbreviated STRUCT	A0	CIP Common
STRUCT	A2	CIP Common
ARRAY	A3	CIP Common
UINT BCD (1-word unsigned BCD)	04	Vendor Specific
UDINT BCD (2-word unsigned BCD)	05	Vendor Specific
ULINT BCD (4-word unsigned BCD)	06	Vendor Specific
ENUM	07	Vendor Specific
DATE_NSEC	08	Vendor Specific
TIME_NSEC	09	Vendor Specific
DATE_AND_TIME_NSEC	0A	Vendor Specific
TIME_OF_DAY_NSEC	0B	Vendor Specific
Union	0C	Vendor Specific

* "CIP Common" indicates codes that are defined in the CIP Common Specifications. "Vendor Specific" indicates codes that are assigned by OMRON.

8-7-2 Common Format

The basic format on the data line is shown below.

Data Format

USINT	Data Type
USINT	AddInfo Length (AddInfo)
UINT	Num of Element (L)
	(H)
	Actual data

Refer to *Data Type Codes* on page 8-57 for specific values.
 Additional information: Field length in bytes
 Additional information: CRC value of structure or other information
 This field exists only in the parameters for the variable write service.

8-7-3 Elementary Data Types

Fixed-length Byte Data

Applicable data types: BYTE, USINT, and SINT

Data Format

USINT	Data Type
USINT	00h
UINT	Num of Elem (L)
	(H)
USINT	Data
USINT	0 (padding)

01 hex
00 hex

00 hex

Fixed-length 2-byte Data

Applicable data types: INT, UINT, UINT BCD, and WORD

Data Format

USINT	Data Type
USINT	00h
UINT	Num of Elem (L)
	(H)
	Data (L)
	(H)

01 hex
00 hex

Fixed-length 4-byte Data

Applicable data types: DINT, UDINT, UDINT BCD, REAL, and DWORD

Data Format

USINT	Data Type
USINT	00h
UINT	Num of Elem (L)
	(H)
	Data (LL)
	(LH)
	(HL)
	(HH)

01 hex
00 hex

Fixed-length 8-byte Data

Applicable data types: LINT, ULINT, ULINT BCD, LREAL, and LWORD

Data Format

USINT	Data Type	
USINT	00 hex	
UINT	Num of Elem	(L)
		(H)
	Data (Least-significant byte)	01 hex
	:	00 hex
	:	
	:	
	:	
	:	
	:	
	(Most-significant byte)	

Boolean Data

Data Format

USINT	Data Type	C1 hex
USINT	00 hex	
UINT	Num of Elem	(L)
		(H)
		01 hex
		00 hex
USINT	Status	01 hex: TRUE, 00 hex : FALSE
USINT	Forced set/reset information*	01 hex: Forced, 00 hex: Not forced

* Specify 0 when writing data.

8-7-4 Derived Data Types

Arrays and structures are handled as derived data types.

Accessing One Member

The data format for accessing one element of an array or one member of a structure is the same as the data format for the corresponding elementary data type.

Example: If you specify Var[5] to access a variable defined with UINT Var[10], use the same data format as for UINT data.

Accessing More Than One Element at the Same Time

● Arrays

- Accessing an Entire Array

If you access an array variable without specifying an element, the entire array is accessed. The following data format is used.

Data Format

USINT	Data Type	Data type of array elements (A1 hex is not used.)
USINT	00 hex	
UINT	Num of Elem	Gives the number of elements in the array.
	(L)	
	(H)	
	Data	The actual data for the elements of the array are given in order in the same format as when the elements are accessed individually.*
	:	
	Data	

* For STRING data, the output format differs from the format when accessing individual elements in the following ways.

- There are no fields for the text string lengths. Only the text strings (including NULL) are given.
- The transferred data length is not the combined lengths of the text strings, but the memory size that is allocated to the STRING variable.
- Upper and lower bytes are reversed.

Example: The outputs will be as follows for a STRING array named s that has two elements (with the data quantity around elements is set to 4 bytes) when s[0] is "ab" and s[1] is "d".

Individual [0]: D0 00 03 00 61 62 63 (hex)

Entire array: D0 00 62 61 ?? 00 00 64 ?? ?? (hex) (?: Invalid data.)

- Handling Multi-dimensional Array

Elements for a multi-dimensional array are given in order from the deepest elements. For example, the data is read in the following format when *Var* is specified for a variable defined with *UINT Var[2][2]*.

Data Format

USINT	C7 hex	Data type code for UINT
USINT	00 hex	
UINT	Value of <i>Var[0][0]</i>	(L)
		(H)
UINT	Value of <i>Var[0][1]</i>	(L)
		(H)
UINT	Value of <i>Var[1][0]</i>	(L)
		(H)
UINT	Value of <i>Var[1][1]</i>	(L)
		(H)

The following data format is used for a BOOL array (using *BOOL b[2][3]* as an example).

Data Format

USINT	C1 hex (data type code for BOOL)							
USINT	00 hex							
(WORD)	rsv	rsv	b[1][2]	b[1][1]	b[1][0]	b[0][2]	b[0][1]	b[0][0]
	rsv	rsv	rsv	rsv	rsv	rsv	rsv	rsv

- Exceptions When Specifying the Num of Element Field

The following data format is used if a specification is made in the Num of Element field for a BOOL array. (Refer to 8-4-5 *Specifying Variable Names in Request Paths* for information on the Num of Element field.) The status (TRUE/FALSE) is given in order for each element of the BOOL variable.

Data Format

USINT	Data Type	C1 hex
USINT	00 hex	
UINT	Num of Elem	(L)
		(H)
USINT	Status	01 hex: TRUE, 00 hex: FALSE
:	:	
USINT	Status	

- Structure Variables**

- Accessing an Entire Structure

If a structure variable is specified, it is treated as an access request for all of the members of the structure.

Data Format

USINT	Data Type	A0 Hex (Abbreviated STRUCT)
USINT	02 hex	
UINT	CRC	(L)
		(H)
UINT	Num of Elem	(L)
		(H)
:	:	01 hex
:	Data	00 hex
:	:	

9

Socket Service

9-1	Basic Knowledge on Socket Communications	9-2
9-1-1	Sockets	9-2
9-1-2	Port Numbers for Socket Services	9-2
9-2	Basic Knowledge on Protocols	9-3
9-2-1	Differences between TCP and UDP	9-3
9-2-2	Fragmenting of Send Data	9-5
9-2-3	Data Reception Processing	9-7
9-2-4	Broadcasting	9-10
9-3	Overview of Built-in EtherNet/IP Port Socket Services	9-11
9-3-1	Overview	9-11
9-3-2	Procedure	9-11
9-4	Settings Required for the Socket Services	9-12
9-5	Socket Service Instructions	9-13
9-6	Details on Using the Socket Services	9-14
9-6-1	Using the Socket Services	9-14
9-6-2	Procedure to Use Socket Services	9-15
9-6-3	Timing Chart for Output Variables Used in Communications	9-17
9-6-4	UDP Sample Programming	9-19
9-6-5	TCP Sample Programming	9-24
9-7	Precautions in Using Socket Services	9-30
9-7-1	Precautions for UDP and TCP Socket Services	9-30
9-7-2	Precautions for UDP Socket Services	9-30
9-7-3	Precautions for TCP Socket Services	9-30

9-1 Basic Knowledge on Socket Communications

9-1-1 Sockets

A socket is an interface that allows you to directly use TCP or UDP functions from the user program. In a host computer (e.g., personal computer), sockets are provided in the form of a C language interface library. If you load the library, you can program communications via TCP and UDP in the user program. In a UNIX computer, a socket interface is provided in the format of system calls. For the built-in EtherNet/IP port, you execute instructions in the user program to use sockets. You can use these communications services to send and receive any data to and from remote nodes, i.e., between host computers and Controllers or between Controllers. The built-in EtherNet/IP port provides a UDP socket service and a TCP socket service.

9-1-2 Port Numbers for Socket Services

Ports 0 to 1023 used for TCP/IP are reserved as well-known ports. In addition, ports 1024 to 49151 are reserved as registered ports by the protocols that are used. Therefore, we recommend that you use port numbers 49152 to 65535 for applications other than the protocols that are registered with the socket service.

You cannot specify port number 0 for the built-in EtherNet/IP port. Also, make sure that the following ports, which are used by the built-in EtherNet/IP port, are not used for other purposes.

Application	UDP	TCP
FTP server	---	20, 21*
DNS	53	---
BOOTP client	68	---
HTTP server	---	80
NTP server	123*	---
SNMP Agent	161*	---
SNMP trap	162*	---
UDP	9600	---
TCP	---	9600*
Used by system.	2222, 2223, and 2224	9610, 9900
CIP messages	44818	44818

* These are the default port numbers. You can change the port numbers in the settings.

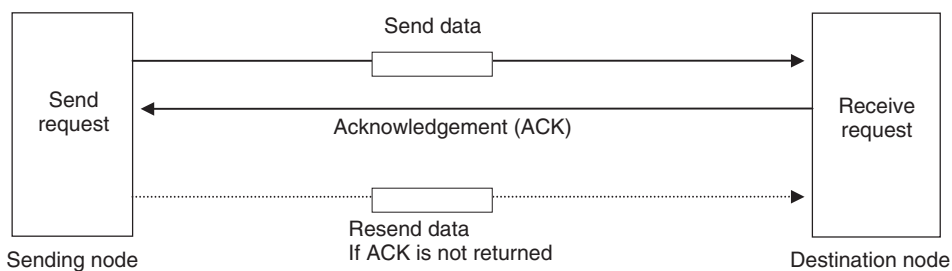
9-2 Basic Knowledge on Protocols

9-2-1 Differences between TCP and UDP

The TCP and UDP functions used on socket services differ as shown below.

TCP Communications

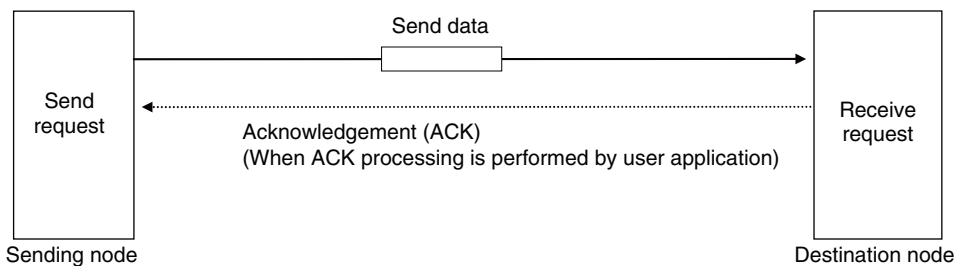
The following operations are performed each time data is sent to ensure that it reaches the destination node. The destination node returns an acknowledgement (ACK) when data is received normally. The sending node sends the next data after ACK is returned. It resends the same data if ACK is not returned within a certain length of time.



In TCP, the remote IP address and the remote TCP port number are specified when a request is made to open a socket. The variables that store the data to send are specified when the send request is made.

UDP Communications

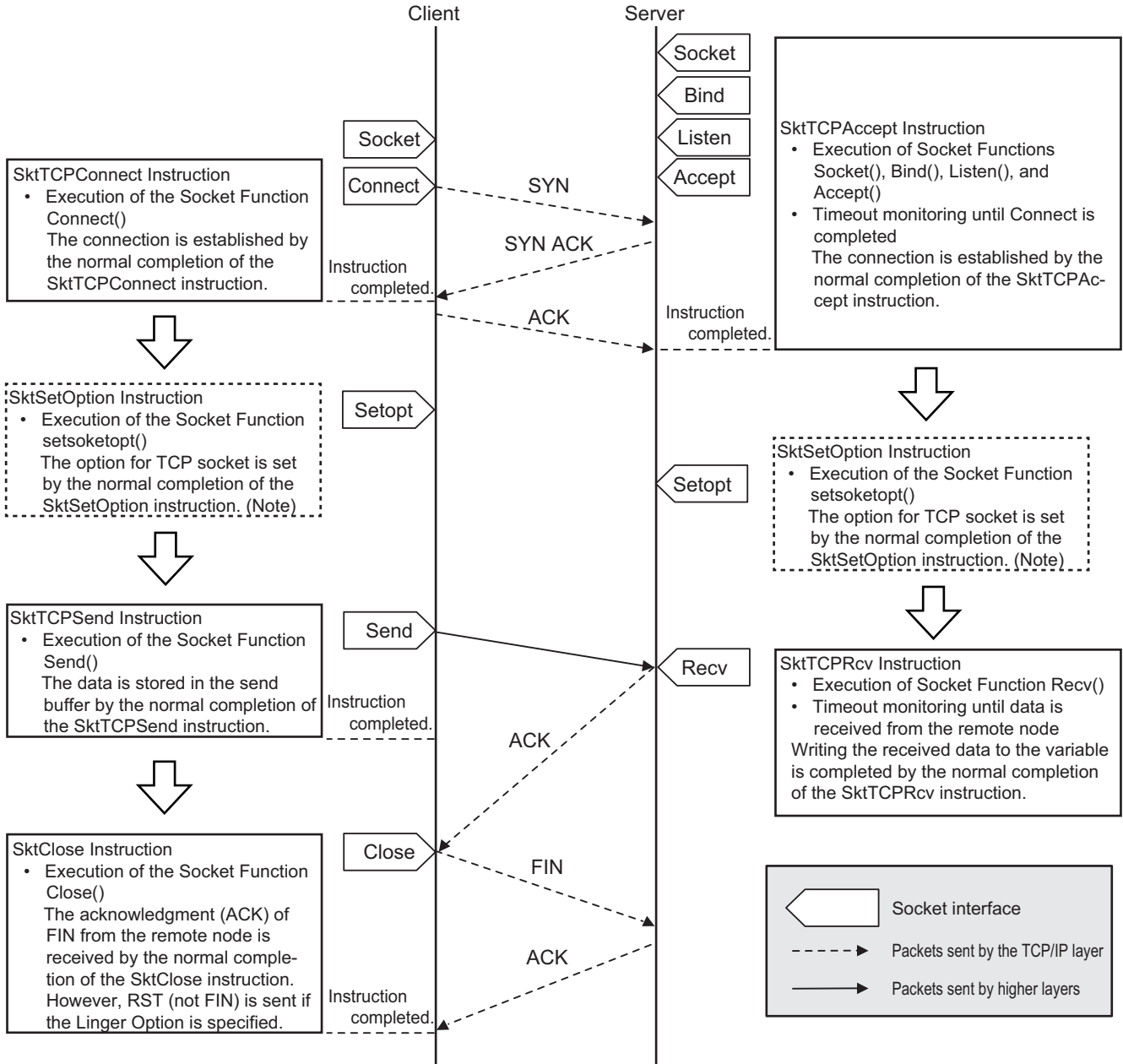
Data is simply sent to the destination node and no acknowledgement or resends are performed like they are for TCP. To increase the reliability of communications, the user application must perform data resend processing.



In UDP, the remote IP address and the remote UDP port number are not specified when a request is made to open a socket. The variables that store the remote IP address, the remote UDP port number, and the data to send are specified when the send request is made. (The received data is stored in the response data to specify which UDP port was used from which IP address.) Furthermore, once a socket is open in UDP, communications with other remote nodes is possible even if the socket is not closed.

TCP Communications Procedure

You execute socket communications instructions in sequence to perform TCP communications for the built-in EtherNet/IP port.



Note Set the socket option as required. Refer to the *NY-series Instructions Reference Manual (Cat. No. W560)* for the socket option.

9-2-2 Fragmenting of Send Data

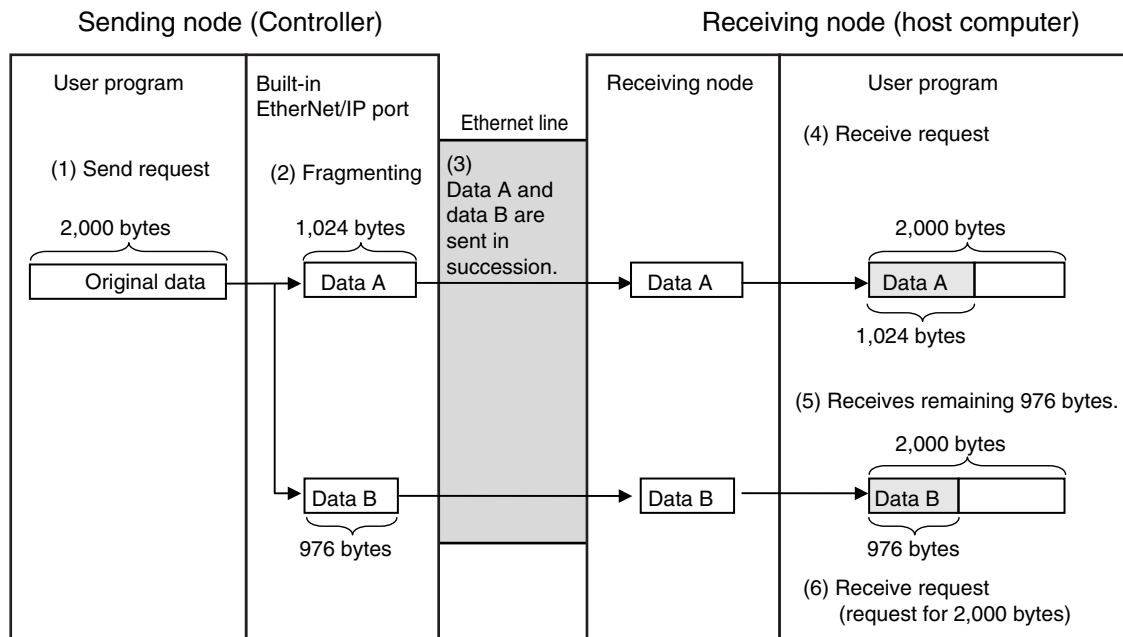
The receive buffer for the built-in EtherNet/IP port is a maximum of 9,000 bytes per socket handle. If any data that is larger than 9,000 bytes is received, any data over 9,000 bytes is discarded.

Up to 2,000 bytes can be received for a single request. In this case, the data is sent in fragments as described below.

Using TCP

The conditions shown in the following figure occur when data is sent in fragments in TCP communications.

- (1) A send request is sent from the user program at the sending node. It specifies a variable with a data length of 2,000 bytes.
- (2) The built-in EtherNet/IP port separates the send data into 1,024 bytes in data A and 976 bytes in data B.
- (3) Data A and data B are sent in sequence by the sending node.
- (4) After data A is received, the remaining data B is received.



Data in TCP communications is delivered to the user program in a fragmented form, as shown above. The number of bytes of data that is received must be checked to confirm all data was received before the next receive request is made. (You can use the *RecvDataSize* output variable of the socket receive request instruction to check the received data.)



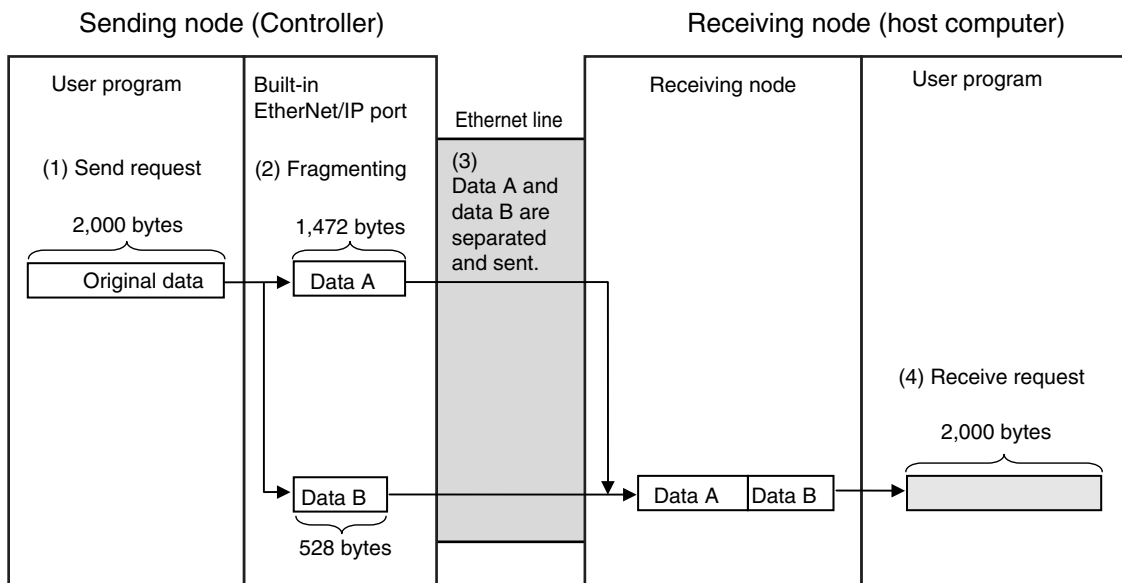
Additional Information

If TCP is used to send data to a different segment, the data is separated into 536-byte fragments.

Using UDP

The conditions shown in the following figure occur when data is sent in fragments in UDP communications.

- (1) A send request is sent from the user program at the sending node. It specifies a variable with a data length of 2,000 bytes.
- (2) The built-in EtherNet/IP port separates the send data into 1,472 bytes in data A and 528 bytes in data B.
- (3) Data A and data B are sent in sequence by the sending node.
- (4) Data A and data B are joined to restore the original send data, and the data is passed to the user program.



Because UDP communications are performed in datagram units as shown above, send data is returned to the original data format before it is passed to the user program.

9-2-3 Data Reception Processing

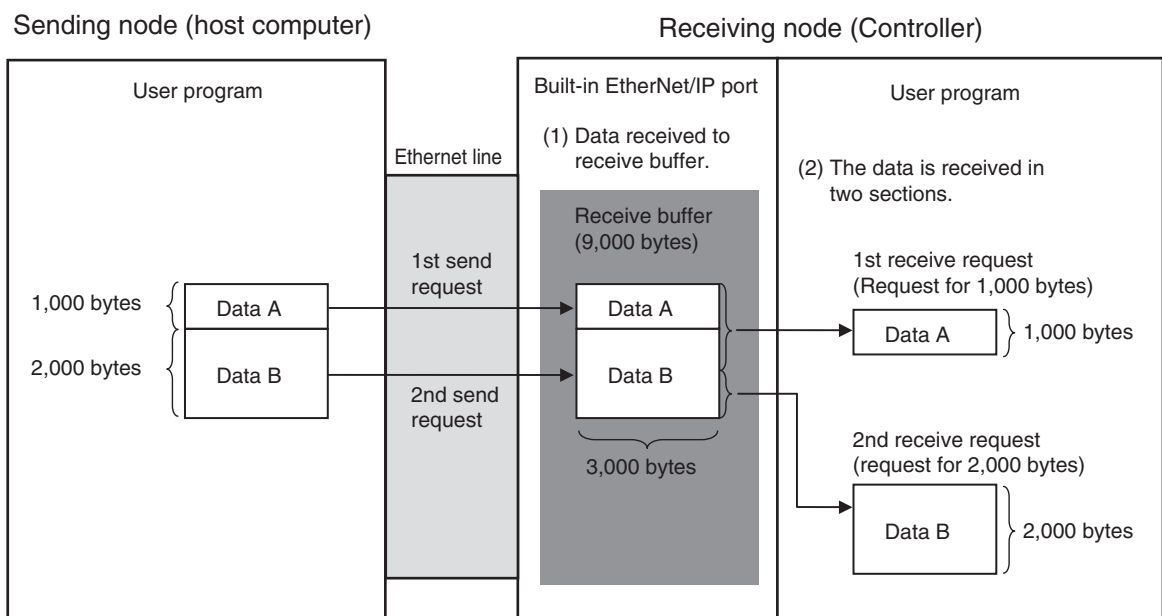
This section describes data reception processing for TCP and UDP.

● TCP Receive Processing

With TCP communications, the receive data that is stored in the receive buffer can be separated and received. (The receive buffer holds up to 9,000 bytes.) This can be used to receive data that exceeds the size that can be received with one data request (2,000 bytes), i.e., more than one receive request is used to receive all of the data. Also, all of the receive data can be received if there is less data in the receive buffer than the size of the variable that was specified in the receive request.

Example: Receiving 3,000 Bytes of Receive Data in Two Sections

1. The data that was sent from the sending node in two sends is stored in the receive buffer.
2. Two receive requests are used to receive all of the send data.



● UDP Receive Processing

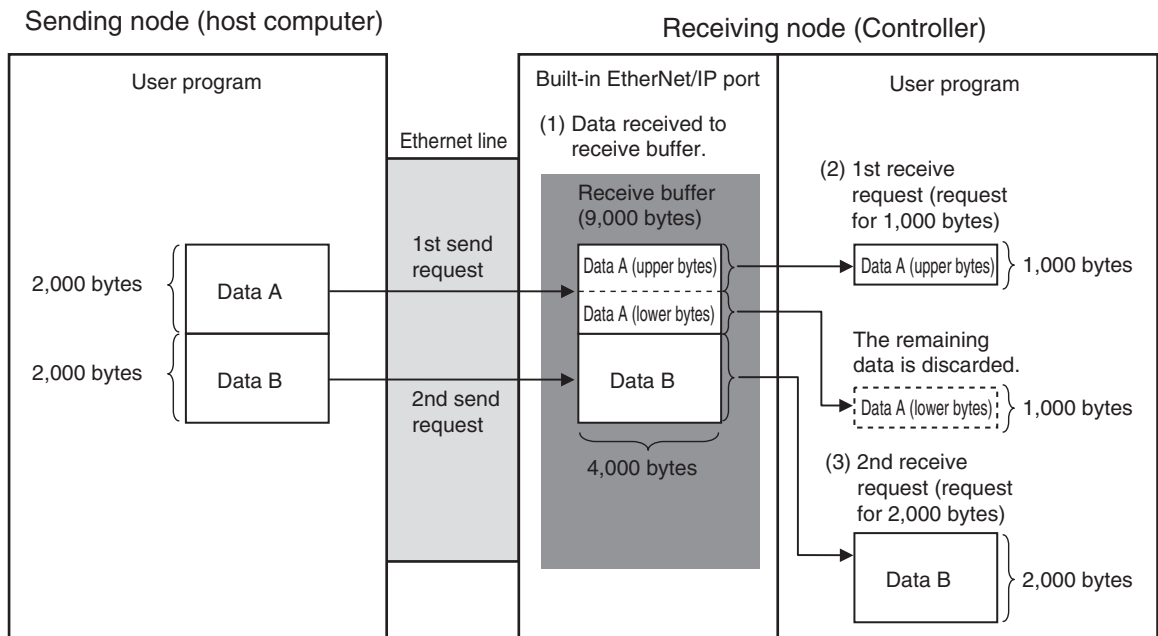
With UDP communications, the receive data that is stored in the receive buffer cannot be separated and received. (The receive buffer holds up to 9,000 bytes.) Therefore, the data that is sent for each send request must be received with one receive request. Because of this, attention must be given to the following at the receiving node when data is received.

- When the Size of the Variable Specified in the Receive Request Is Smaller Than the Data Size Sent with the Send Request

The received data that exceeds the size of the variable that is specified in the receive request is discarded. All of the receive data is received if there is less data in the receive buffer than the size of the variable that was specified in the receive request.

Example 1: Receive Request for 1,000 Bytes when 2,000 Bytes of Data Was Received

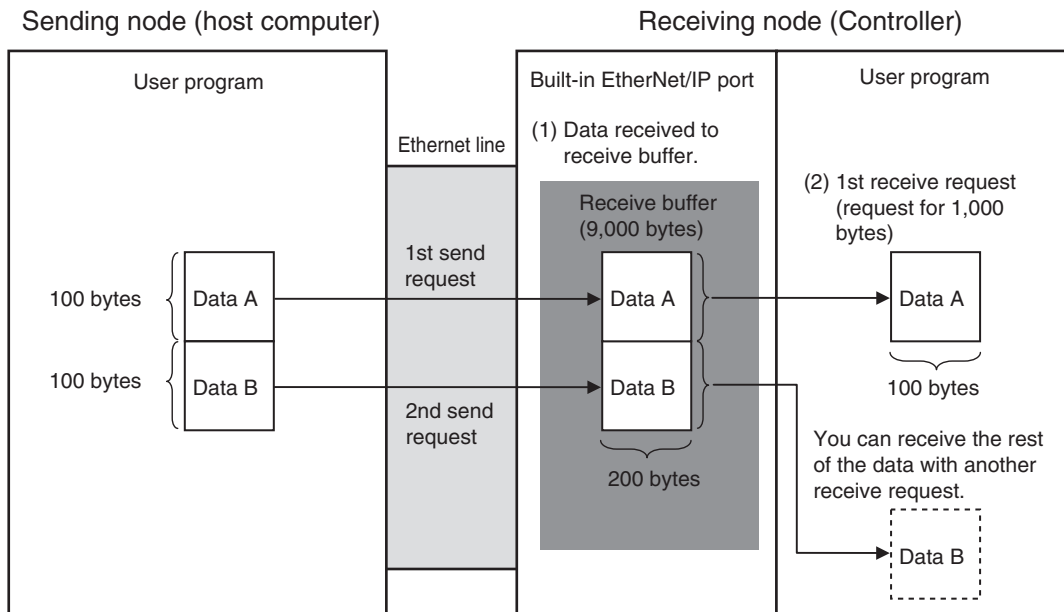
1. The data that was sent from the sending node in two sends is stored in the receive buffer.
2. If a receive request is made for 1,000 bytes for the first send data, the remaining 1,000 bytes is discarded.
3. If the next receive request is for 2,000 bytes, all of the send data for the second send is received.



- When There Is Only One Receive Request for Data Sent for Multiple Send Requests
 You cannot use just one receive request to receive the data that was sent for multiple send requests, regardless of the size of the data.

Example 2: Receive Request for 1,000 Bytes when 200 Bytes of Data Was Sent for Two Send Requests

1. The data that was sent from the sending node in two sends is stored in the receive buffer.
2. Even if a receive request is made for 2,000 bytes of data, only the 100 bytes that was sent for the first send request is received.



9-2-4 Broadcasting

You can set the destination IP address to a broadcast address for a UDP socket to broadcast data to the host and all nodes on the same network as the EtherNet/IP port. If there is a router on the network, packets are not sent beyond the router. You can broadcast up to 1,472 bytes of data. Data larger than 1,472 bytes cannot be broadcast.

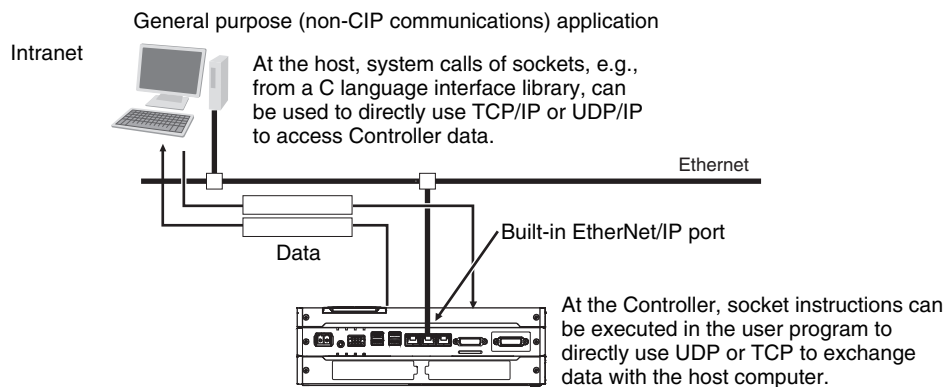
You can specify either of the two following types of broadcast addresses.

- Local Broadcast
If no destination IP address is specified, the following IP address is specified automatically.
Network segment: The network segment of the local IP address is set.
Host segment: All bits are set to 1.
- Global Broadcast
Specify this type when the IP address of the local node or the subnet to which the local node belongs is unknown. As shown below, every bit of the 32-bit address is set to 1.
255.255.255.255

9-3 Overview of Built-in EtherNet/IP Port Socket Services

9-3-1 Overview

Socket services on the built-in EtherNet/IP port are used to exchange data between Controllers and general-purpose applications that do not support CIP message communications. The Controller requests the socket service from the user program.



Overview of Socket Services with Socket Service Instructions

You can use socket service instructions to use the socket services. The maximum total number of UDP and TCP sockets that you can use is given in the following table.

UDP/TCP	Number of sockets
	NY-series Controller
UDP socket service	Total of 30 sockets
TCP socket service	

9-3-2 Procedure

1 Make the settings that are required for socket services.
Refer to *9-4 Settings Required for the Socket Services*.



2 Execute the socket service instructions from the user program.
Refer to *9-5 Socket Service Instructions*.

9-4 Settings Required for the Socket Services

Make the following settings in the Unit Setup to use the socket services.

Sysmac Studio Unit Settings Tab Page	Setting	Setting conditions
Setting	Local IP Address	Required
	Subnet Mask	Required
	TCP/IP Keep Alive	Optional (Change when the default setting of 5 minutes is unacceptable.)
	Linger Option	Optional



Additional Information

Make this setting in the TCP/IP Dialog Box. Refer to *5-1 TCP/IP Settings Display* for information on the TCP/IP Dialog Box.

9-5 Socket Service Instructions

You can use the following socket service instructions for socket services use. Refer to the *NY-series Instructions Reference Manual* (Cat. No. W560) for information on the socket service instructions.

UDP/TCP	Instruction	Socket service
UDP sockets	SktUDPCreate	Create UDP Socket instruction
	SktUDPRcv	UDP Socket Receive instruction
	SktUDPSend	UDP Socket Send instruction
TCP sockets	SktTCPAccept	Accept TCP Socket instruction
	SktTCPConnect	Connect TCP Socket instruction
	SktTCPRcv	TCP Socket Receive instruction
	SktTCPSend	TCP Socket Send instruction
	SktGetTCPStatus	Read TCP Socket Status instruction
Services for both UDP and TCP sockets	SktClose	Close TCP/UDP Socket instruction
	SktClearBuf	Clear TCP/UDP Socket Receive Buffer instruction
	SktSetOption	Set TCP Socket Option instruction



Precautions for Correct Use

You can execute a maximum of 32 socket service instructions at the same time. Use exclusive control in the user program so that no more than 32 socket service instructions are executed at the same time.

9-6 Details on Using the Socket Services

9-6-1 Using the Socket Services

The built-in EtherNet/IP port has the total number of sockets for TCP and UDP that is given in the following table.

UDP/TCP	Number of sockets
	NY-series Controller
UDP socket service	Total of 30 sockets
TCP socket service	

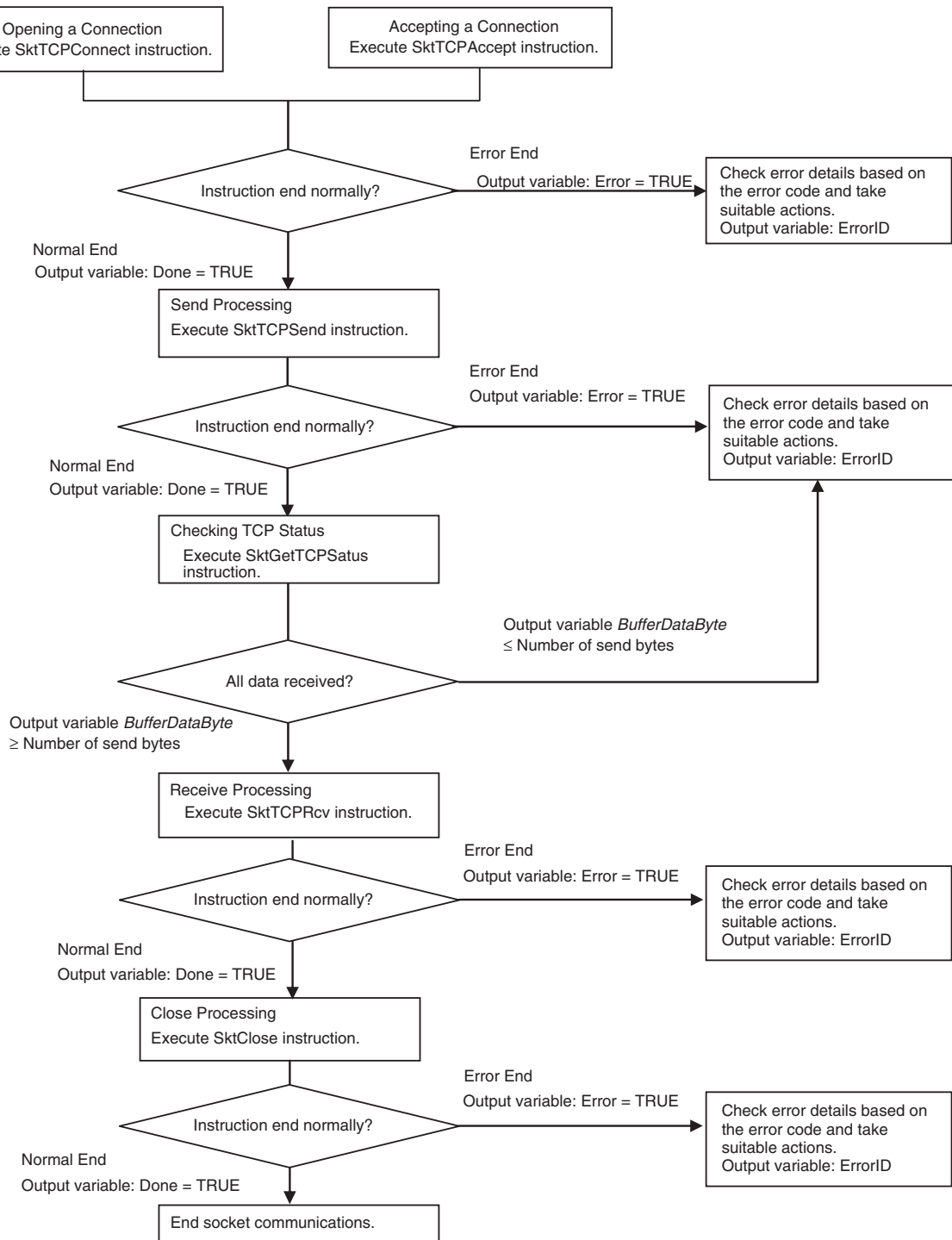
To use these sockets for communications, special ST instructions for sockets are executed to execute the following processes.

Open processing:	This process places the socket in a usable state. This is the first process that is executed when using socket services. With TCP, open processing is performed until a connection is established.
Close processing:	This process ends the use of the socket. With TCP, it closes the connection.
Send processing:	This process sends data from the socket.
Receive processing:	This process receives data from the socket.
Clear processing:	This process clears data in the receive buffer received from the remote node.

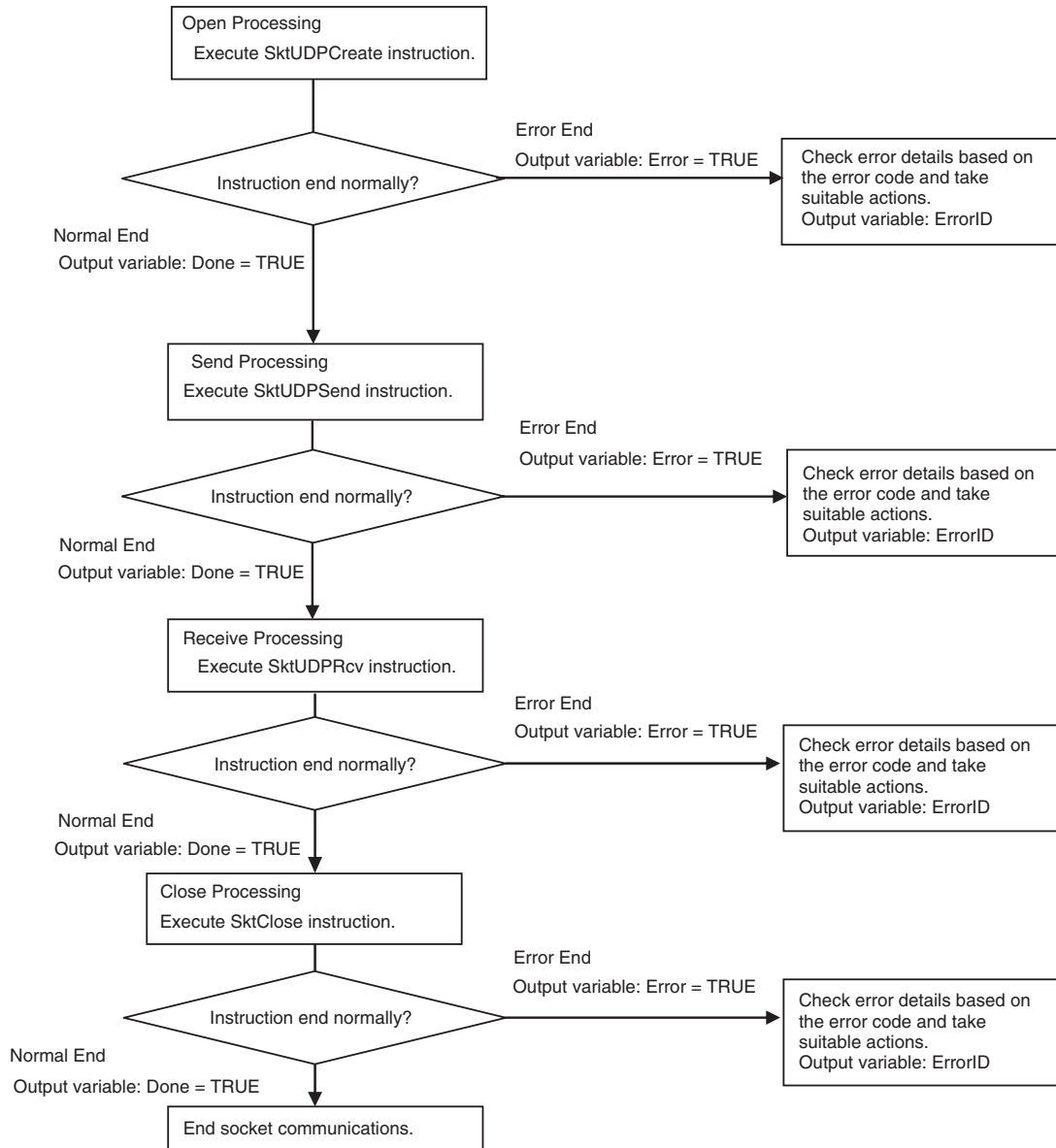
9-6-2 Procedure to Use Socket Services

You execute special instructions for sockets in sequence to use the socket services according to the procedure shown below. Use the values of the output variables for each instruction to confirm that each instruction is completed normally.

TCP



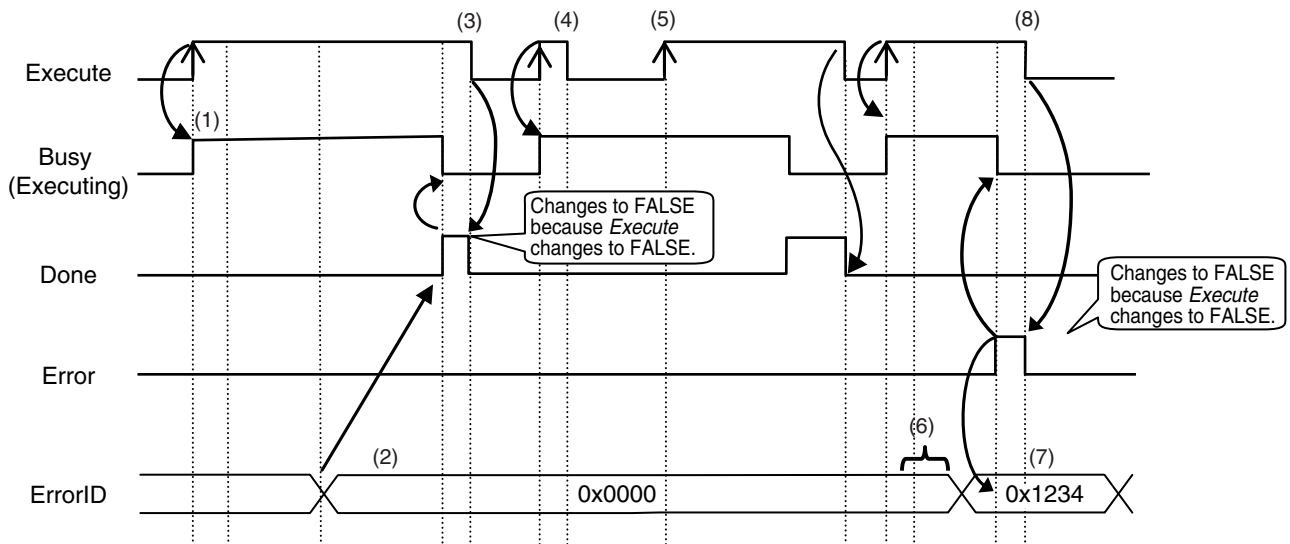
UDP



9-6-3 Timing Chart for Output Variables Used in Communications

Output Variable Operation and Timing

You can monitor the values of the output variables to determine the status throughout instruction execution. The following timing chart shows the operation of the output variables.



- 1** When *Execute* changes to TRUE, the instruction is executed and *Busy* changes to TRUE.
- 2** After the results of the instruction execution of the instruction are stored in the output variables, *Done* changes to TRUE and *Busy* changes to FALSE.
- 3** When *Execute* changes to FALSE, *Done* returns to FALSE.
- 4** When *Execute* changes to TRUE again, *Busy* changes to TRUE.
- 5** *Execute* is ignored if it changes to TRUE during instruction executed (i.e., when *Busy* is TRUE).
- 6** If an error occurs, several retries are attempted internally. The error code in the *ErrorID* is not updated during the retries.
- 7** When a communications error occurs, *Error* changes to TRUE and the *ErrorID* is stored. Also, *Busy* and *Done* change to FALSE.
- 8** When *Execute* changes to FALSE, *Error* changes to FALSE.

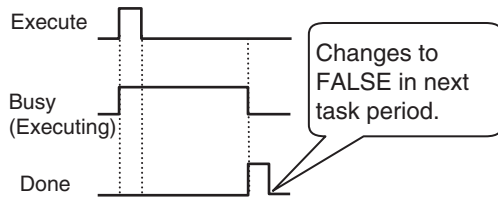


Precautions for Correct Use

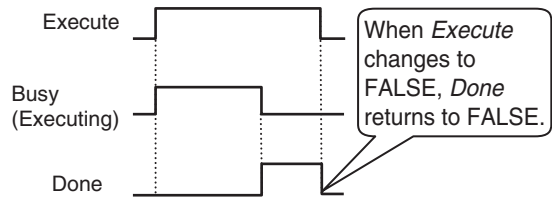
If *Execute* changes back to FALSE before *Done* changes to TRUE, *Done* stays TRUE for only one task period. (Example 1)

If you want to see if *Done* is TRUE at any time, make sure to keep *Execute* TRUE until you confirm that *Done* is TRUE. If *Execute* is TRUE until *Done* changes to TRUE, *Done* stays TRUE until *Execute* changes to FALSE. (Example 2)

Example 1

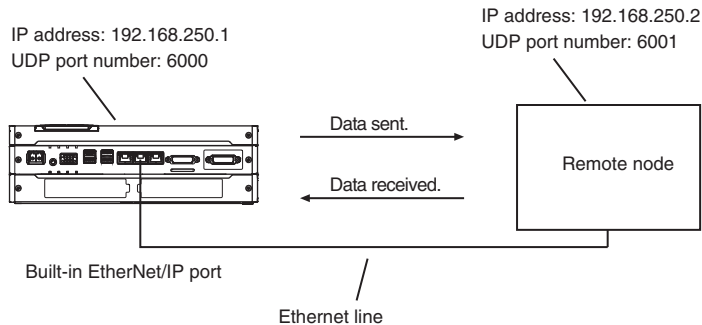


Example 2



9-6-4 UDP Sample Programming

In this sample, the UDP socket service is used for data communications between the NY-series Controller and a remote node.



The processing procedure is as follows:

- 1** The SktUDPCreate instruction is executed to request creation of a UDP socket.
- 2** Execute the SktUDPSend instruction to request sending. The data in *SendSocketDat[]* is sent.
- 3** The SktUDPRcv instruction is executed to request reception. The receive data is stored in *RcvSocketDat[]*.
- 4** The SktClose instruction is executed to close the socket.

ST

Internal Variables	Variable	Data type	Initial value	Comment
	Trigger	BOOL	False	Execution condition
	DoSendAndRcv	BOOL	False	Processing
	Stage	INT	0	Status change
	RcvSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	Receive data
	WkSocket	_sSOCKET	(Handle:=0, SrcAdr:=(PortNo:=0, IpAdr:=""), DstAdr:=(PortNo:=0, IpAdr:=""))	Socket
	SendSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	Send data
	SktUDPCreate_instance	SktUDPCreate		
	SktUDPSend_instance	SktUDPSend		
	SktUDPRcv_instance	SktUDPRcv		
	SktClose_instance	SktClose		

External Variables	Variable	Data type	Constant	Comment
	<u>_EIP_EtnOnlineSta</u> *1	BOOL	<input checked="" type="checkbox"/>	Online

*1 Replace the variable name with _EIP1_EtnOnlineSta (Port1 Online) or _EIPIn1_EtnOnlineSta (Internal Port1 Online), depending on which communications port you use.

```

// Start sequence when Trigger changes to TRUE.
IF((Trigger=TRUE) AND(DoSendAndRcv=FALSE) AND (_EIP_EtnOnlineSta=TRUE))THEN
  DoSendAndRcv :=TRUE;
  Stage        :=INT#1;
  SktUDPCreate_instance(Execute:=FALSE);      // Initialize instance.
  SktUDPSend_instance(                        // Initialize instance.
    Execute    :=FALSE,
    SendDat    :=SendSocketDat[0]);           // Dummy
  SktUDPRcv_instance(                        // Initialize instance.
    Execute    :=FALSE,
    RcvDat     :=RcvSocketDat[0]);           // Dummy
  SktClose_instance(Execute:=FALSE);         // Initialize instance.
END_IF;

IF (DoSendAndRcv=TRUE) THEN
  CASE Stage OF
  1 :          // Request creating socket.
    SktUDPCreate_instance(
      Execute :=TRUE,
      SrcUdpPort:=UINT#6000,                // Local UDP port number
      Socket   =>WkSocket);                // Socket

    IF (SktUDPCreate_instance.Done=TRUE) THEN
      Stage :=INT#2;                        // Normal end
    ELSIF (SktUDPCreate_instance.Error=TRUE) THEN
      Stage :=INT#10;                       // Error end
    END_IF;

  2 :          // Send request
    WkSocket.DstAdr.PortNo :=UINT#6001;
    WkSocket.DstAdr.IpAdr:='192.168.250.2';
    SktUDPSend_instance(
      Execute :=TRUE,
      Socket  :=WkSocket,                   // Socket
      SendDat :=SendSocketDat[0],          // Send data
      Size    :=UINT#2000);                // Send data size

    IF (SktUDPSend_instance.Done=TRUE) THEN
      Stage :=INT#3;                        // Normal end
    ELSIF (SktUDPSend_instance.Error=TRUE) THEN
      Stage :=INT#20;                       // Error end
    END_IF;
  END_CASE;
END_IF;

```

```

3 :                // Request receiving data.
  SktUDPRcv_instance(
    Execute   :=TRUE,
    Socket    :=WkSocket,           // Socket
    TimeOut   :=UINT#0,           // Timeout value
    Size      :=UINT#2000,        // Receive data size
    RcvDat    :=RcvSocketDat[0]); // Receive data

  IF (SktUDPRcv_instance.Done=TRUE) THEN
    Stage     :=INT#4;             // Normal end
  ELSIF (SktUDPRcv_instance.Error=TRUE) THEN
    Stage     :=INT#30;           // Error end
  END_IF;

4 :                // Request closing data.
  SktClose_instance(
    Execute   :=TRUE,
    Socket    :=WkSocket);        // Socket

  IF (SktClose_instance.Done=TRUE) THEN
    Stage     :=INT#0;           // Normal end
  ELSIF (SktClose_instance.Error=TRUE) THEN
    Stage     :=INT#40;         // Error end
  END_IF;

0 :                // Normal end
  DoSendAndRcv:=FALSE;
  Trigger      :=FALSE;

ELSE                // Interrupted by error.
  DoSendAndRcv:=FALSE;
  Trigger      :=FALSE;
END_CASE;

END_IF;

```

● Remote Node Programming

In this example, programming is also required in the remote node. The order of sending and receiving is reversed in comparison with the above procedure.

- 1** The SktUDPCreate instruction is used to request creating a UDP socket.
- 2** The SktUDPRcv instruction is executed to request reception. The receive data is stored in *RcvSocketDat[]*.
- 3** Execute the SktUDPSend instruction to request sending. The data in *SendSocketDat[]* is sent.
- 4** The SktClose instruction is executed to close the socket.

ST

Internal Variables	Variable	Data type	Initial value	Comment
	Trigger	BOOL	False	Execution condition
	DoSendAndRcv	BOOL	False	Processing
	Stage	INT	0	Status change
	RcvSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	Receive data
	WkSocket	_sSOCKET	(Handle:=0, SrcAdr:=(PortNo:=0, IpAdr:=""), DstAdr:=(PortNo:=0, IpAdr:=""))	Socket
	SendSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	Send data
	SkUDPCreate_instance	SkUDPCreate		
	SkUDPSend_instance	SkUDPSend		
	SkUDPRcv_instance	SkUDPRcv		
	SkClose_instance	SkClose		

External Variables	Variable	Data type	Constant	Comment
	_EIP_EtnOnlineSta*1	BOOL	<input checked="" type="checkbox"/>	Online

*1 Replace the variable name with `_EIP1_EtnOnlineSta` (Port1 Online) or `_EIPIn1_EtnOnlineSta` (Internal Port1 Online), depending on which communications port you use.

// Start sequence when *Trigger* changes to TRUE.

IF((Trigger=TRUE) AND (DoSendAndRcv=FALSE) AND (_EIP_EtnOnlineSta=TRUE))THEN

DoSendAndRcv :=TRUE;

Stage :=INT#1;

SkUDPCreate_instance(Execute:=FALSE); // Initialize instance.

SkUDPSend_instance(// Initialize instance.

Execute :=FALSE,

SendDat:=SendSocketDat[0]; // Dummy

SkUDPRcv_instance(// Initialize instance.

Execute :=FALSE,

RcvDat :=RcvSocketDat[0]; // Dummy

SkClose_instance(Execute:=FALSE); // Initialize instance.

END_IF;

IF (DoSendAndRcv=TRUE) THEN

CASE Stage OF

1 : // Request creating socket.

SkUDPCreate_instance(

Execute :=TRUE,

SrcUdpPort:=UINT#6001, // Local UDP port number

Socket =>WkSocket); // Socket

IF (SkUDPCreate_instance.Done=TRUE) THEN

Stage :=INT#2; // Normal end

ELSIF (SkUDPCreate_instance.Error=TRUE) THEN

Stage :=INT#10; // Error end

END_IF;


```

2 :                // Request receiving data.
  SktUDPRcv_instance(
    Execute   :=TRUE,
    Socket    :=WkSocket,           // Socket
    TimeOut   :=UINT#0,             // Timeout value
    Size      :=UINT#2000,         // Receive data size
    RcvDat    :=RcvSocketDat[0]);  // Receive data

  IF (SktUDPRcv_instance.Done=TRUE) THEN
    Stage     :=INT#3;              // Normal end
  ELSIF (SktUDPRcv_instance.Error=TRUE) THEN
    Stage     :=INT#20;             // Error end
  END_IF;

3 :                // Send request
  WkSocket.DstAdr.PortNo:=UINT#6000;
  WkSocket.DstAdr.IpAdr :='192.168.250.1';
  SktUDPSend_instance(
    Execute   :=TRUE,
    Socket    :=WkSocket,           // Socket
    SendDat   :=SendSocketDat[0],  // Send data
    Size      :=UINT#2000);        // Send data size

  IF (SktUDPSend_instance.Done=TRUE) THEN
    Stage     :=INT#4;              // Normal end
  ELSIF (SktUDPSend_instance.Error=TRUE) THEN
    Stage     :=INT#30;             // Error end
  END_IF;

4 :                // Request closing data.
  SktClose_instance(
    Execute   :=TRUE,
    Socket    :=WkSocket);         // Socket

  IF (SktClose_instance.Done=TRUE) THEN
    Stage     :=INT#0;              // Normal end
  ELSIF (SktClose_instance.Error=TRUE) THEN
    Stage     :=INT#40;             // Error end
  END_IF;

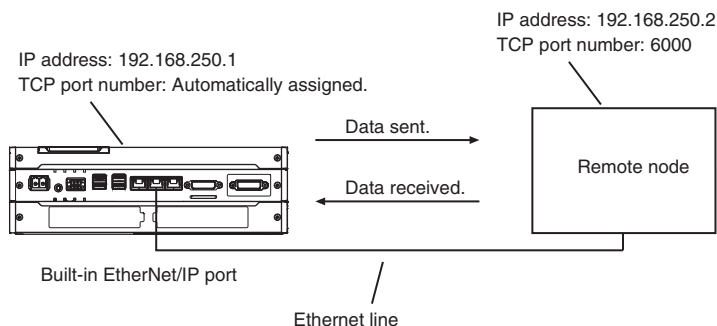
0 :                // Normal end
  DoSendAndRcv:=FALSE;
  Trigger      :=FALSE;
ELSE                // Interrupted by error.
  DoSendAndRcv:=FALSE;
  Trigger      :=FALSE;
END_CASE;

END_IF;

```

9-6-5 TCP Sample Programming

In this sample, the TCP socket service is used for data communications between the NY-series Controller and a remote node.



The processing procedure is as follows:

- 1** The SktTCPConnect instruction is executed to request connecting to the TCP port on the remote node.
- 2** The SktClearBuf instruction is executed to clear the receive buffer for a TCP socket.
- 3** The SktGetTCPStatus instruction is executed to read the status of a TCP socket.
- 4** The SktTCPSend instruction is executed to request sending data. The data in *SendSocketDat[]* is sent.
- 5** The SktTCPRcv instruction is executed to request receiving data. The receive data is stored in *RcvSocketDat[]*.
- 6** The SktClose instruction is executed to close the socket.

ST

Internal Variables	Variable	Data type	Initial value	Comment
	Trigger	BOOL	False	Execution condition
	DoTCP	BOOL	False	Processing
	Stage	INT	0	Status change
	RcvSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	Receive data
	WkSocket	_sSOCKET	(Handle:=0,SrcAdr:=(PortNo:=0, IpAdr:""), DstAdr:=(PortNo:=0, IpAdr:""))	Socket
	SendSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	Send data
	SktTCPConnect_instance	SktTCPConnect		
	SktClearBuf_instance	SktClearBuf		
	SktGetTCPStatus_instance	SktGetTCPStatus		
	SktTCPSend_instance	SktTCPSend		
	SktTCPRcv_instance	SktTCPRcv		
	SktClose_instance	SktClose		

External Variables	Variable	Data type	Constant	Comment
	_EIP_EtnOnlineSta*1	BOOL	<input checked="" type="checkbox"/>	Online

*1 Replace the variable name with *_EIP1_EtnOnlineSta* (Port1 Online) or *_EIPIn1_EtnOnlineSta* (Internal Port1 Online), depending on which communications port you use.

```

// Start sequence when Trigger changes to TRUE.
IF ((Trigger=TRUE) AND (DoTCP=FALSE) AND (_EIP_EtnOnlineSta=TRUE)) THEN
  DoTCP      :=TRUE;
  Stage      :=INT#1;
  SktTCPConnect_instance(Execute:=FALSE);    // Initialize instance.
  SktClearBuf_instance(Execute:=FALSE);      // Initialize instance.
  SktGetTCPStatus_instance(Execute:=FALSE);  // Initialize instance.
  SktTCPSend_instance(                       // Initialize instance.
    Execute  :=FALSE,
    SendDat  :=SendSocketDat[0]);           // Dummy
  SktTCPRcv_instance(                        // Initialize instance.
    Execute  :=FALSE,
    RcvDat   :=RcvSocketDat[0]);           // Dummy
  SktClose_instance(Execute:=FALSE);        // Initialize instance.
END_IF;

IF (DoTCP=TRUE) THEN
  CASE Stage OF
  1 :                               // Request a connection.
    SktTCPConnect_instance(
      Execute  :=TRUE,
      SrcTcpPort :=UINT#0,             // Local TCP port number: Automatically assigned.
      DstAdr    :='192.168.250.2',    // Remote IP address
      DstTcpPort :=UINT#6000,         // Destination TCP port number
      Socket    =>WkSocket);         // Socket

    IF (SktTCPConnect_instance.Done=TRUE) THEN
      Stage    :=INT#2;                // Normal end
    ELSIF (SktTCPConnect_instance.Error=TRUE) THEN
      Stage    :=INT#10;               // Error end
    END_IF;

  2 :                               // Clear receive buffer.
    SktClearBuf_instance(
      Execute  :=TRUE,
      Socket   :=WkSocket);           // Socket

    IF (SktClearBuf_instance.Done=TRUE) THEN
      Stage    :=INT#3;                // Normal end
    ELSIF (SktClearBuf_instance.Error=TRUE) THEN
      Stage    :=INT#20;               // Error end
    END_IF;
  END_CASE;
END_IF;

```

```

3 :                // Request reading status.
SkGetTCPStatus_instance(
    Execute :=TRUE,
    Socket  :=WkSocket);           // Socket

IF (SkGetTCPStatus_instance.Done=TRUE) THEN
    Stage   :=INT#4;               // Normal end
ELSIF (SkGetTCPStatus_instance.Error=TRUE) THEN
    Stage   :=INT#30;              // Error end
END_IF;

4 :                // Send request
SkTCPSend_instance(
    Execute :=TRUE,
    Socket  :=WkSocket,           // Socket
    SendDat :=SendSocketDat[0],   // Send data
    Size    :=UINT#2000);         // Send data size

IF (SkTCPSend_instance.Done=TRUE) THEN
    Stage   :=INT#5;               // Normal end
ELSIF (SkTCPSend_instance.Error=TRUE) THEN
    Stage   :=INT#40;              // Error end
END_IF;

5 :                // Request receiving data.
SkTCPRcv_instance(
    Execute :=TRUE,
    Socket  :=WkSocket,           // Socket
    TimeOut :=UINT#0,             // Timeout value
    Size    :=UINT#2000,         // Receive data size
    RcvDat  :=RcvSocketDat[0]);   // Receive data

IF (SkTCPRcv_instance.Done=TRUE) THEN
    Stage   :=INT#6;               // Normal end
ELSIF (SkTCPRcv_instance.Error=TRUE) THEN
    Stage   :=INT#50;              // Error end
END_IF;

6 :                // Request closing data.
SkClose_instance(
    Execute :=TRUE,
    Socket  :=WkSocket);           // Socket

IF (SkClose_instance.Done=TRUE) THEN
    Stage   :=INT#0;               // Normal end
ELSIF (SkClose_instance.Error=TRUE) THEN
    Stage   :=INT#60;              // Error end
END_IF;

```

```

0 :                // Normal end
  DoTCP            :=FALSE;
  Trigger          :=FALSE;

ELSE              // Interrupted by error.
  DoTCP            :=FALSE;
  Trigger          :=FALSE;
END_CASE;

END_IF;

```

● Remote Node Programming

In this example, programming is also required in the remote node. The order of sending and receiving is reversed in comparison with the above procedure.

- 1** The SktTCPAccept instruction is executed to request accepting a TCP socket.
- 2** The SktTCPRcv instruction is executed to request receiving data. The receive data is stored in *RcvSocketDat[]*.
- 3** The SktTCPSend instruction is executed to request sending data. The data in *SendSocketDat[]* is sent.
- 4** The SktClose instruction is executed to close the socket.

ST

Internal Variables	Variable	Data type	Initial value	Comment
	Trigger	BOOL	False	Execution condition
	DoTCP	BOOL	False	Processing
	Stage	INT	0	Status change
	RcvSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	Receive data
	WkSocket	_sSOCKET	(Handle:=0, SrcAdr:=(PortNo:=0,IpAdr:=""), DstAdr:=(PortNo:=0,IpAdr:=""))	Socket
	SendSocketDat	ARRAY[0..1999] OF BYTE	[2000(16#0)]	Send data
	SktTCPAccept_instance	SktTCPAccept		
	SktTCPSend_instance	SktTCPSend		
	SktTCPRcv_instance	SktTCPRcv		
	SktClose_instance	SktClose		

External Variables	Variable	Data type	Constant	Comment
	_EIP_EtnOnlineSta*1	BOOL	<input checked="" type="checkbox"/>	Online

- *1 Replace the variable name with *_EIP1_EtnOnlineSta* (Port1 Online) or *_EIPIn1_EtnOnlineSta* (Internal Port1 Online), depending on which communications port you use.

```

// Start sequence when Trigger changes to TRUE.
IF ((Trigger=TRUE) AND (DoTCP=FALSE) AND (_EIP_EtnOnlineSta=TRUE)) THEN
  DoTCP          :=TRUE;
  Stage          :=INT#1;
  SktTCPAccept_instance(Execute:=FALSE);      // Initialize instance.
  SktTCPSend_instance(                          // Initialize instance.
    Execute      :=FALSE,
    SendDat      :=SendSocketDat[0]);          // Dummy
  SktTCPRcv_instance(                           // Initialize instance.
    Execute      :=FALSE,
    RcvDat       :=RcvSocketDat[0]);           // Dummy
  SktClose_instance(Execute:=FALSE);          // Initialize instance.
END_IF;

IF (DoTCP=TRUE) THEN
  CASE Stage OF
  1 :                               // Request accepting a socket connection.
    SktTCPAccept_instance(
      Execute :=TRUE,
      SrcTcpPort:=UINT#6000,          // Local TCP port number
      TimeOut  :=UINT#0,              // Timeout value
      Socket   =>WkSocket);           // Socket

    IF (SktTCPAccept_instance.Done=TRUE) THEN
      Stage :=INT#2;                  // Normal end
    ELSIF (SktTCPAccept_instance.Error=TRUE) THEN
      Stage :=INT#10;                 // Error end
    END_IF;

  2 :                               // Request reception.
    SktTCPRcv_instance(
      Execute :=TRUE,
      Socket  :=WkSocket,              // Socket
      TimeOut :=UINT#0,                // Timeout value
      Size    :=UINT#2000,             // Receive data size
      RcvDat  :=RcvSocketDat[0]);      // Receive data

    IF (SktTCPRcv_instance.Done=TRUE) THEN
      Stage :=INT#3;                  // Normal end
    ELSIF (SktTCPRcv_instance.Error=TRUE) THEN
      Stage :=INT#20;                 // Error end
    END_IF;
  END_CASE;
END_IF;

```

```

3 :                // Send request
  SendSocketDat:=RcvSocketDat;
  SktTCPSend_instance(
    Execute   :=TRUE,
    Socket    :=WkSocket,           // Socket
    SendDat   :=SendSocketDat[0],   // Send data
    Size      :=UINT#2000);         // Send data size

  IF (SktTCPSend_instance.Done=TRUE) THEN
    Stage     :=INT#4;              // Normal end
  ELSIF (SktTCPSend_instance.Error=TRUE) THEN
    Stage     :=INT#30;             // Error end
  END_IF;

4 :                // Request closing data.
  SktClose_instance(
    Execute   :=TRUE,
    Socket    :=WkSocket);         // Socket

  IF (SktClose_instance.Done=TRUE) THEN
    Stage     :=INT#0;              // Normal end
  ELSIF (SktClose_instance.Error=TRUE) THEN
    Stage     :=INT#40;             // Error end
  END_IF;

0 :                // Normal end
  DoTCP      :=FALSE;
  Trigger    :=FALSE;
ELSE         // Interrupted by error.
  DoTCP      :=FALSE;
  Trigger    :=FALSE;
END_CASE;

END_IF;

```

9-7 Precautions in Using Socket Services

9-7-1 Precautions for UDP and TCP Socket Services

- Communications processing are sometimes delayed when multiple functions of the built-in EtherNet/IP port are used simultaneously or due to the contents of the user program.
- Communications efficiency is sometimes reduced by high communications traffic on the network line.
- The close processing for a close request instruction discards all of the buffered send and receive data for the socket. For example, send data from a send request instruction immediately before the close processing is sometimes not sent.
- After a socket is open, the built-in EtherNet/IP port provides a receive buffer of 9,000 bytes per TCP socket and 9,000 bytes per UDP socket to enable data to be received at any time. If the receive buffer is full, data received by that socket is discarded. Make sure that the user application always executes receive requests to prevent the internal buffer from becoming full.

9-7-2 Precautions for UDP Socket Services

- The destination IP address can be set to a broadcast address for a UDP socket to broadcast data to all nodes on the network. However, in this case, the maximum length of send data is 1,472 bytes. Data lengths broken into multiple fragments (1,473 bytes or more in UDP) cannot be sent.
- For UDP socket, controls to confirm the reliability of communications, such as the confirmation of send data, are not performed. To improve the reliability of communications when you use UDP sockets, make sure the user program confirms that data is sent and resends data when necessary.

9-7-3 Precautions for TCP Socket Services

- If the TCP socket is closed on the remote node without warning during communications (i.e., if the connection is closed), the socket at the local node must also be closed. You can use the Read TCP Socket Status instruction (SktGetTCPstatus) to see if the connection is closed. Immediately close the socket at the local node if the TCP socket at the remote node is closed.
- If the remote node's TCP socket closes without warning, the data to send may remain in the buffer at the local node. The remaining data is discarded in the local node's TCP close processing. The steps that are required in applications to avoid this include sending data from the sending node that permits closing and closing the socket only after checking the remote node.
- While open processing is performed for a TCP socket, a port that was closed first cannot be opened again for 60 seconds from the time the close processing is performed for the remote socket. However, this is not true if you specified 0 (automatic assignment by the Unit) as the port for the SktTCP-Connect instruction.
- You can use *Connect* from another socket to open a connection to a socket that was opened with *Accept*. A connection is not opened if you try to use *Connect* from another socket to open a connection to a socket that was opened with *Connect*. Also, a connection is not opened if you attempt to use *Accept* from another socket to open a socket that was opened with *Accept*. Furthermore, you cannot use *Connect* from more than one other node to establish multiple connections with a single TCP socket that was opened with *Accept* on the built-in EtherNet/IP port.
- You can use the keep-alive function for TCP sockets at the built-in EtherNet/IP port. The keep alive function checks whether a connection is normally established when no data is sent or received for a certain period on the communications line where the connection was established. The built-in EtherNet/IP port responds to checks from other nodes even if keep alive is not specified.

- For TCP sockets, the send data is resent up to 12 times if an acknowledgment (ACK) from the remote node is not received. The resend interval increases to between one second and 64 seconds for each resend.
- For TCP sockets, a connection request (SYN) is sent by performing an open connection. SYN is resent up to four time if an acknowledgment (SYN + ACK) from the remote node is not received. An error will occur if SYN + ACK is not received even 75 seconds elapsed since the open processing is performed.

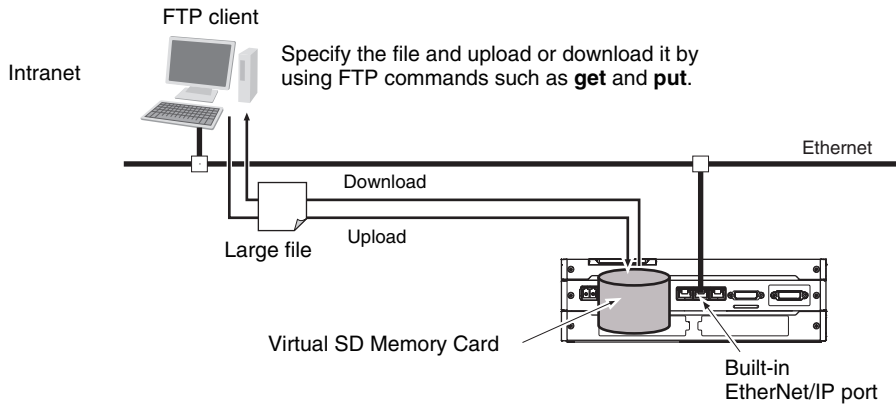
FTP Server

10-1 Overview and Specifications	10-2
10-1-1 Overview	10-2
10-1-2 Specifications	10-3
10-2 FTP Server Function Details	10-4
10-2-1 Supported Files	10-4
10-2-2 Connecting to the FTP Server	10-4
10-3 Using the FTP Server Function	10-7
10-3-1 Procedure	10-7
10-3-2 List of Settings Required for the FTP Server Function	10-7
10-4 FTP Server Application Example	10-8
10-5 Using FTP Commands	10-10
10-5-1 Table of Commands	10-10
10-5-2 Using the Commands	10-11
10-6 Using SD Memory Card Operations	10-16
10-6-1 File Types	10-16
10-6-2 Format of Variable Data	10-17
10-7 Application Example from a Host Computer	10-18

10-1 Overview and Specifications

10-1-1 Overview

The built-in EtherNet/IP port contains an FTP (file transfer protocol) server. You can therefore send FTP commands from an FTP client software application on a computer on an Ethernet network to upload and download large files on the Virtual SD Memory Card.



Precautions for Correct Use

To use a Virtual SD Memory Card, you need the settings for the SD Memory Card operations in the Controller. For information on setting the SD Memory Card operations, refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558).

10-1-2 Specifications

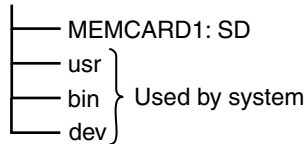
Item	Specification
Executable commands	<p>open: Connects the specified host FTP server.</p> <p>user: Specifies the user name for the remote FTP server.</p> <p>ls: Displays file names in the remote host.</p> <p>mls: Displays file names in the remote host.</p> <p>dir: Displays file names and details in the remote host.</p> <p>mdir: Displays file names and details in the remote host.</p> <p>rename: Changes a file name.</p> <p>mkdir: Creates a new directory in the working directory in the remote host.</p> <p>rmdir: Deletes a directory from the working directory in the remote host.</p> <p>cd: Changes the work directory in the remote host to the specified directory.</p> <p>pwd: Displays the work directory on the remote host.</p> <p>type: Specifies the data type of transferred files.</p> <p>get: Transfers the specified remote file to the local host.</p> <p>mget: Transfers the specified multiple remote files to the local host.</p> <p>put: Transfers the specified local file to the remote host.</p> <p>mput: Transfers the specified multiple local files to the remote host.</p> <p>delete: Deletes the specified file from the remote host.</p> <p>mdelete: Deletes the specified multiple files from the remote host.</p> <p>append: Uses the currently specified file data type and appends a local file to the remote host.</p> <p>close: Disconnects the FTP server.</p> <p>bye: Closes the FTP client.</p> <p>quit: Closes the FTP client.</p>
Protection	<p>Login name (up to 12 characters)</p> <p>Password consists of 8 to 32 characters</p>
Protocol used	FTP (port number: 20/TCP, 21/TCP)
Number of connections	6

10-2 FTP Server Function Details

10-2-1 Supported Files

The file system in the Controller that can be accessed by the built-in EtherNet/IP port includes files in any Virtual SD Memory Card in the NY-series Controller. The directory tree is shown below.

/: root



A connection is initially made to the root directory.



Additional Information

- The date of the MEMCARD1 directory displayed for **ls**, **dir**, or **mkdir** commands in the root directory is the date of the file system volume label.
- The login date is displayed for MEMCARD1 if a volume label has not been created.

10-2-2 Connecting to the FTP Server

Input the FTP login name and password to login to the built-in EtherNet/IP port from an FTP client application. Use the Built-in EtherNet/IP Port Settings in the Sysmac Studio to set the FTP login name and password.



Additional Information

When a general-purpose FTP application is used, you can use a graphical user interface similar to Explorer to transfer and read files.

● Login Name and Password Setting

The FTP login name and password are not set by default. Use the Built-in EtherNet/IP Port Settings to set any login name and password.

● Login Messages

Status	Message
Normal connection	220 xxx.xx.xx.xx FTP server ready. xxx.xx.xx.xx: Controller model (example: NY512-1500-131513F10)
Connected to maximum number of connections (6)	530 FTP server busy, Goodbye.

● Restrictions on Login Name and Password Setting

The following restrictions apply to login names and passwords.

- Only single-byte alphanumeric characters can be used in the login name and password. The login name and password are case sensitive.
- A login name consists of up to 12 characters.
- A password consists of 8 to 32 characters.
- Always set a password when you set a new login name. The login name will not be valid unless a password is set for it.
- The login name is invalid if the login name is not set or characters other than single-byte alphanumeric characters are used.

● FTP File Transfer Mode

FTP has two file transfer modes: ASCII mode and binary mode. Before you start to transfer files, use the **type** command (specifies the data type of transferred files) to select the required mode.

- To transfer a file in binary format: Select binary mode.
- To transfer a file in ASCII format: Select ASCII mode.

● Multiple Accesses to the Same File

Files accessed with the FTP server may be simultaneously accessed from multiple sources with communications commands from other FTP servers or programming instructions. Exclusive control is required to prevent multiple accesses. This is to prevent reading and writing the same file at the same time. The Controller automatically performs exclusive control as shown below only when the following combinations of instructions are used. In other cases, use file operation instructions (Change File Name, Copy File, etc.) or communications commands and perform exclusive control.

- Exclusive Control When Accessing the Same File on the Virtual SD Memory Card

			First access					
			Instructions*1		File Operations from the Sysmac Studio		FTP server	
			Reading	Writing	Reading	Writing	Reading	Writing
Later access	Instructions	Reading	Exclusive control is performed automatically, and an error occurs for the instruction that is executed later.		Exclusive control is not required.	Perform exclusive control.	Exclusive control is not required.	Perform exclusive control.
		Writing			Perform exclusive control.	Perform exclusive control.		
	File operations from the Sysmac Studio	Reading	Exclusive control is not required.	Perform exclusive control.	---	---	Exclusive control is not required.	
		Writing	Perform exclusive control.		---	---	Perform exclusive control.	
	FTP server	Reading	Exclusive control is not required.	Perform exclusive control.	Exclusive control is not required.	Perform exclusive control.	---	---
		Writing	Perform exclusive control.				---	---

*1 The instructions include the SD Memory Card operation instructions and the FTP client communications instructions.

- **Restrictions on Connection to FTP Server**

If you repeat connection to and disconnection from the FTP server frequently in a short period of time, your access may be restricted temporarily to protect the system. If you cannot connect to the FTP server, wait for 10 minutes and try again.

10-3 Using the FTP Server Function

10-3-1 Procedure

- 1** Make the basic settings.
Refer to *1-5 EtherNet/IP Communications Procedures* for basic operations.
- 2** Set up the FTP server on the Sysmac Studio. (Refer to *5-3 FTP Display*.)
- 3** Select **Built-in EtherNet/IP Port Settings** under **Configuration - Controller Setup** on the Sysmac Studio. Make the following settings on the FTP Settings Display.
 - FTP server
 - Port number
 - Login name
 - Password
- 4** Place the Controller online and transfer the settings to the Controller.
- 5** Connect to the built-in EtherNet/IP port from an FTP client.
- 6** Input the FTP login name and password that you set in the Built-in EtherNet/IP Port Settings to log in to the built-in EtherNet/IP port.

Note Once logged in, the ftp commands can be used, such as cd (Change Directory), and get (Obtain File).
- 7** After you are logged in, you can use the ftp commands, such as cd (Change Directory) and get (Obtain File) for the MEMCARD1 directory in the Virtual SD Memory Card in the Controller.
- 8** Close the connection.

10-3-2 List of Settings Required for the FTP Server Function

Make the following settings for the unit setup when the FTP server function is used.

Built-in EtherNet/IP Port Settings Tab Page on Sysmac Studio	Setting	Setting conditions	Reference
FTP	FTP server	Required	page 5-7
	Port No.	Optional* Note Required when changing the default value of 21.	
	Login name	Required*	
	Password	Required*	

* Settings are not required if the FTP server is not used.



Additional Information

Make the settings in the FTP Settings Dialog Box if the FTP server is used. Refer to *5-3 FTP Display* for information on the FTP Settings Dialog Box.

10-4 FTP Server Application Example

An example application of the FTP server when the login name is “user1” and the password is “password” is shown below.



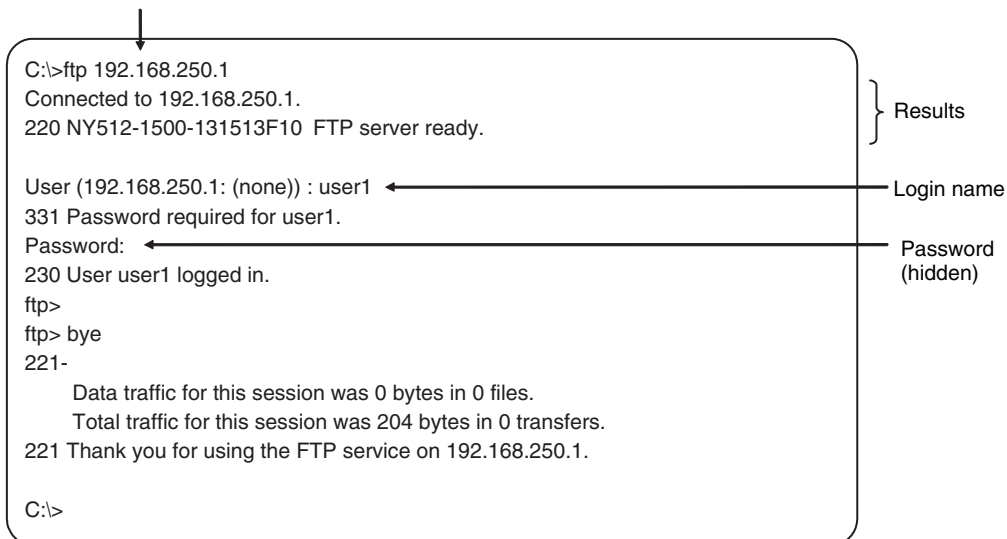
Additional Information

When a general-purpose FTP application is used, you can use a graphical user interface similar to Explorer to transfer and read files.

● Step

- 1** Turn ON the power supply to the Controller.
- 2** Connect to the FTP server from a computer on the Ethernet by entering the text that is underlined in the following diagram.

IP address of built-in EtherNet/IP port



- 3** Enter FTP commands (underlined in the following diagram) to read and write files. The following directory tree is used in this example.

```

/ (root directory)
├── MEMCARD1
│   └── ABC (subdirectory)
│       └── DEF.BIN (file)

```

```

ftp>ls ← File names read.
200 PORT command successful.
150 Opening ASCII mode data connection for 'file list'
usr
bin
MEMCARD1
dev
226 Transfer complete.
ftp:** bytes received in 0 seconds(**bytes/s)
ftp>cd MEMCARD1 ← Change to MEMCARD1 directory
250 CWD command successful.
ftp>get ABC/DEF.BIN ← Get DEF.BIN from ABC directory
200 PORT command successful.
150 opening ASCII mode data connection for 'ABC/DEF.BIN'(**bytes).
226 Transfer complete
**bytes received in *.*** seconds(**bytes/s)

```

Results

Results

Results

10-5 Using FTP Commands

This section describes the FTP commands which the host computer (FTP client) can send to the FTP server of the built-in EtherNet/IP port. The descriptions should also apply to most workstations, but slight differences may arise. Refer to your workstation's operation manuals for details.

10-5-1 Table of Commands

The FTP commands which can be sent to the built-in EtherNet/ IP port are listed in the following table.

Command	Description
open	Connects the specified host FTP server.
user	Specifies the user name for the remote FTP server.
ls	Displays file names in the remote host.
mls	Displays file names in the remote host.
dir	Displays file names and details in the remote host.
mdir	Displays file names and details in the remote host.
rename	Changes a file name.
mkdir	Creates a new directory in the working directory in the remote host.
rmdir	Deletes a directory from the working directory in the remote host.
cd	Changes the work directory in the remote host to the specified directory.
pwd	Displays the work directory on the remote host.
type	Specifies the data type of transferred files.
get	Transfers the specified remote file to the local host.
mget	Transfers the specified multiple remote files to the local host.
put	Transfers the specified local file to the remote host.
mput	Transfers the specified multiple local files to the remote host.
delete	Deletes the specified file from the remote host.
mdelete	Deletes the specified multiple files from the remote host.
append	Uses the file data type that is specified by the type command to append the local file to the remote host.
close	Disconnects the FTP server.
bye	Closes the FTP client.
quit	Closes the FTP client.

Note 1 "Remote host" refers to the built-in EtherNet/IP port.

2 A "remote file" is a file on the Virtual SD Memory Card in the Controller.

3 "Local host" refers to the host computer (FTP client).

4 "Local file" refers to a file on the host computer (FTP client).

10-5-2 Using the Commands

open

- **Format**

open [*IP_address* or *host_name_of_FTP_server*]

- **Function**

Connects the FTP server. Normally when the FTP client is booted, the FTP server IP address is specified to execute this command automatically.

user

- **Format**

user [*user_name*]

- **Function**

- Specifies the user name. Specify the FTP login name set in the built-in EtherNet/IP port system setup.
- The user name is automatically requested immediately after connection to the FTP server.

ls

- **Format**

ls [-l] [*remote_file_name* [*local_file_name*]]

- **Function**

- Displays the names of files on the remote host (on the Virtual SD Memory Card).
- Set the switch [-l] to display not only the file names but the creation dates and sizes as well. If the switch is not set, only the file names are displayed.
- Specify a file on the Virtual SD Memory Card for the remote file name.
- If a local file name is specified, the file information is stored in the specified file.

mls

- **Format**

mls *remote_file_name* *local_file_name*

- **Function**

- Displays a list of the names of files on the remote host (on the Virtual SD Memory Card).
- Specify the directory or file name on the Virtual SD Memory Card you wish to list for remote file name. Input an asterisk (*) to display a list of the current working directory.
- If a local file name is specified, the file information is stored in the specified file. Input a hyphen (-) to display a list of the remote host but not store the list of file names.

dir

- **Format**

`dir [remote_file_name [local_file_name]]`

- **Function**

- Displays the names, creation dates, and sizes of files on the remote host (on the Virtual SD Memory Card).
- It displays the same information as command `[ls -l]`.
- Specify a file on the Virtual SD Memory Card for the remote file name.
- If a local file name is specified, the file information is stored in the specified file.

mmdir

- **Format**

`mmdir remote_file_name local_file_name`

- **Function**

- Displays the names of files, subdirectories, creation dates, and sizes on the remote host (on the Virtual SD Memory Card).
- Specify the directory or file name on the Virtual SD Memory Card to list as the remote file name. Input a hyphen (-) to display a list of the current working directory.
- If a local file name is specified, the file information is stored in the specified file. Input a hyphen (-) to display a list of the remote host and not store the list of file names.

rename

- **Format**

`rename current_file_name new_file_name`

- **Function**

- Changes the specified current file name to the specified new file name.
- If the new file name is already used on the remote host (on the Virtual SD Memory Card), the existing file is overwritten by the file for which the name was changed.
- **rename** can be used only to change the file name. It cannot be used to move the file to a different directory.

mkdir

- **Format**

`mkdir directory_name`

- **Function**

- Creates a directory of the specified name at the remote host (on the Virtual SD Memory Card).
- An error will occur if a file or directory of the same name already exists in the working directory.

rmdir

- **Format**

rmdir *directory_name*

- **Function**

- Deletes the directory with the specified name from the remote host (from the Virtual SD Memory Card).
- The directory must be empty to delete it.
- An error will occur if the specified directory does not exist or is not empty.

pwd

- **Format**

pwd

- **Function**

- Displays the work directory on the remote host.

append

- **Format**

append *local_file_name* [*remote_file_name*]

- **Function**

Uses the file data type that is specified by the type command to append the local file to the remote host (on the Virtual SD Memory Card).

cd

- **Format**

cd [*directory_name*]

- **Function**

- Changes the remote host work directory to the specified remote directory.
- Files on the Virtual SD Memory Card are stored in the MEMCARD1 directory under the root directory (/).
- The root directory (/) is the directory that is used when you log onto the built-in EtherNet/IP port. The MEMCARD1 directory does not exist if a Virtual SD Memory Card is not set in the Controller.

type

- **Format**

type *data_type*

- **Function**

- Specifies the file data type.
- The following data types are supported:
 - ascii: Files are transferred as ASCII data.
 - binary (image): Files are transferred as binary data. The Controller handles binary files. Use the type command to specify binary transfers before you upload or download files. File contents cannot be guaranteed if transferred as ASCII data.
- The default file type is ASCII.

get

- **Format**

get *file_name* [*receive_file_name*]

- **Function**

- Transfers the specified remote file from the Virtual SD Memory Card to the local host.
- A receive file name can be used to specify the name of the file in the local host.

mget

- **Format**

mget *file_name*

- **Function**

- You can include wildcards (*) in the file name to transfer multiple remote files from the Virtual SD Memory Card to the local host.

put

- **Format**

put *file_name* [*destination_file_name*]

- **Function**

- Transfers the specified local file to the remote host (to the Virtual SD Memory Card).
- You can specify the destination file name to specify the name the file is stored under on the Virtual SD Memory Card.
- Any existing file with the same name in the remote host (on the Virtual SD Memory Card) is overwritten by the contents of the transferred file.

mput

- **Format**

mput *file_name*

- **Function**

- You can include wildcards (*) in the file name to transfer multiple local files to the remote host (to the Virtual SD Memory Card).

- Any existing file with the same name in the remote host (on the Virtual SD Memory Card) is overwritten by the contents of the transferred file.

delete

- **Format**

delete *file_name*

- **Function**

- Deletes the specified remote file (on the Virtual SD Memory Card).

mdelete

- **Format**

mdelete *file_name*

- **Function**

- You can include wildcards (*) in the file name to delete multiple remote files from the Virtual SD Memory Card.

close

- **Format**

close

- **Function**

- Disconnects the FTP server of the built-in EtherNet/IP port.

bye

- **Format**

bye

- **Function**

- Ends the FTP sessions.

quit

- **Format**

quit

- **Function**

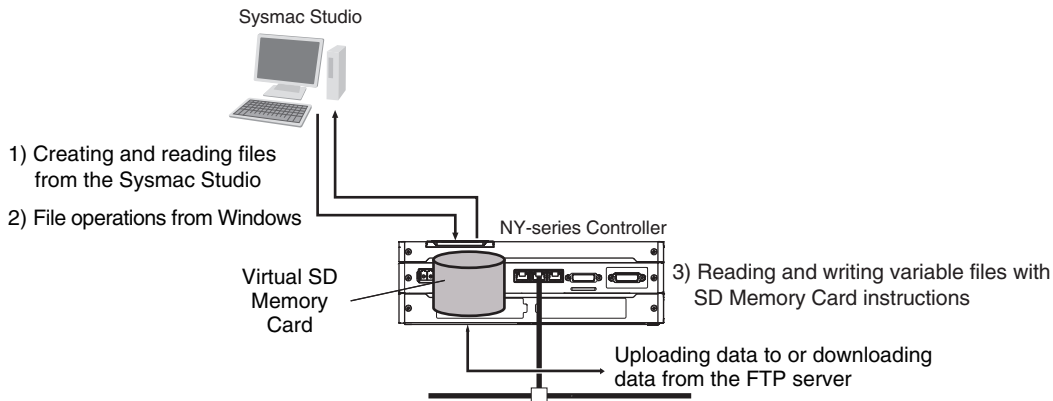
- Ends the FTP sessions.

10-6 Using SD Memory Card Operations

The built-in EtherNet/IP port can be used to upload and download the following data between the Virtual SD Memory Card in the NY-series Controller and the FTP server.

- Variables files (binary format)

The following three methods are used by the Controller to store and read data on the Virtual SD Memory Card.



Precautions for Correct Use

To use a Virtual SD Memory Card, you need the settings for the SD Memory Card operations in the Controller. For information on setting the SD Memory Card operations, refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558).

10-6-1 File Types

File Names

Files are distinguished by assigning file names and extensions. The following characters can be used in file names and extensions: File names are not case sensitive.

A to Z, a to z, 0 to 9, and the following symbols: \$ % ' - _ @ ! ' () ~ = # & + ^ [] { } , . ;

The following characters cannot be used in files names and extensions:

Blanks, multi-bytes characters, and the following symbols: / \ ? * " : < > etc.

The maximum file name length with the extension is 65 characters. The first period (.) in a file name is taken as the delimiter between the file name and extension. Extensions are determined by the file type.

Directory

You can create up to five levels of directories to store files on the Virtual SD Memory Card (count the root directory as one level). A maximum of 65 characters can be used in a directory name.

File Names Handled by Controller

The files described in the following table can be read or written by the Controller.

File type	File names	Extension	Contents	Description
Variables file (binary format)	Refer to <i>10-6-1 File Types</i> .	.bin	Specified variables	This variables file contains the values of specified variables (which include arrays and structures) in binary format (.bin).

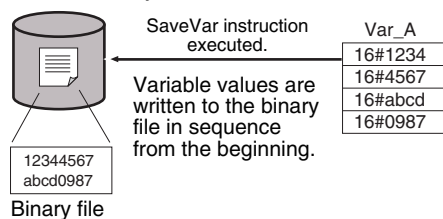
Refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558) for details.

10-6-2 Format of Variable Data

Binary Format

The IOM format is a data format used for binary data specified by the ladder instructions, FileReadVar (Read Variables File) and FileWriteVar (Save Variables File), in the Controller. You can also read and save arrays and structures. Data is created as shown below when the data of variable *Var_A* is placed in an attached file in binary format.

Virtual SD Memory Card



Additional Information

- When you handle a binary file on the NY-series Controller, always specify the binary data type with the **type** command before you read or write the file via FTP. (Refer to *10-5-2 Using the Commands*.)
- For details on how to use ladder diagram instructions to process files, refer to the *NY-series Instructions Reference Manual* (Cat. No. W560).

10-7 Application Example from a Host Computer

The following procedure provides an example of FTP operations from a host computer. In this example, the following assumptions are made.

- The IP address of the built-in EtherNet/IP port is registered in the hosts as host name [nj].
- The FTP login name is "LogIn".
- Manufacturing results is stored in the Virtual SD Memory Card in the NY-series Controller in a file that is named RESULT.BIN.
- A processing instructions data file called PLAN.BIN already exists on the workstation.

In the following procedure, the manufacturing results file (RESULT.BIN) in the Virtual SD Memory Card in the Controller is transferred to a workstation, and then a manufacturing instructions file (PLAN.BIN) on the workstation is transferred to the Virtual SD Memory Card in the Controller. Underlined text is keyed in from the FTP client. The workstation prompt is indicated as \$ and the cursor is indicated as ■.

- 1 Start the FTP application and connect to the built-in EtherNet/IP port.

```
$ ftp nj ←
connected to nj
220 **IPAddress** NY512-1300 FTP server(FTP**version**)ready
Name(nj:root): ■
```

FTP started.

- 2 Enter the login name.

```
Name(nj:root):LogIn ←
331 Password required for LogIn.
Password: ←
230 LogIn logged in.
ftp> ■
```

Enter the login name.

Enter the password.

- 3 Make sure the Virtual SD Memory Card settings are correctly made. The MEMCARD1 directory is displayed if the Virtual SD Memory Card settings are correctly made.

```
ftp> ls ←
200 PORT command successful.
150 opening data connection for ls(**IPAddress**port#**)(0 bytes).
MEMCARD1
226 Transfer complete.
15 bytes received in 0 seconds(**bytes/s)
ftp> ■
```

Make sure the Virtual SD Memory Card settings are correctly made.

- 4 Change to the MEMCARD1 directory.

```
ftp> cd MEMCARD1 ←
250 CWD command successful.
ftp> ■
```

Change the directory.

- 5 Change data type to binary.

```
ftp> type binary ←
200 Type set to I.
ftp> ■
```

Set binary data type.

6 Read the file RESULT.BIN and transfer it to the workstation.

```
ftp> get RESULT.BIN ←
200 PORT command successful.
150 opening data connection for result.bin (**IPAddress**port#**) (**bytes).
226 Transfer complete.
** bytes received in *.** seconds (**bytes/s)
ftp> ■
```

Read file.

7 Write the file PLAN.BIN to the Virtual SD Memory Card.

```
ftp> put PLAN.BIN ←
200 PORT command successful.
150 opening data connection for plan.bin (**IPAddress**port#**) .
226 Transfer complete.
** bytes received in *.** seconds (**bytes/s)
ftp> ■
```

Write file.

8 End the FTP session.

```
ftp> bye ←
221 Goodbye.
$ ■
```

FTP ended.

11

FTP Client

11-1 Using the FTP Client to Transfer Files	11-2
11-1-1 Transferring Files	11-2
11-1-2 Connectable FTP Servers	11-2
11-1-3 File Transfer Options	11-3
11-1-4 Other Functions	11-4
11-2 FTP Client Communications Instructions	11-5
11-2-1 Functions of the FTP Client Communications Instructions	11-5
11-2-2 Restrictions on the FTP Client Communications Instructions	11-7
11-3 FTP Client Application Example	11-9

11-1 Using the FTP Client to Transfer Files

You can use the FTP client to transfer files between the FTP client and an FTP server. You can transfer files in either direction: download data from the FTP server to the FTP client or upload data from the FTP client to the FTP server.

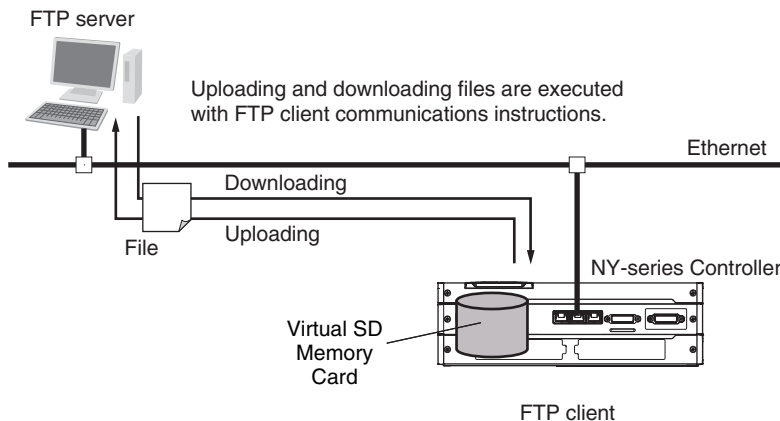
11-1-1 Transferring Files

All file transfers that use the FTP client are executed with FTP client communications instructions in the user program. The file transfer settings are all made with the parameters of the FTP client communications instructions. No settings are required from the Sysmac Studio.

The FTP client communications instructions and their functions are given in the following table. You can execute up to three FTP client communications instructions at the same time.

Instruction	Function
FTPGetFileList	Gets a file list from the FTP server.
FTPGetFile	Downloads one or more files from the FTP server.
FTPPutFile	Uploads one or more files to the FTP server.
FTPRemoveFile	Deletes one or more files on the FTP server.
FTPRemoveDir	Deletes a directory on the FTP server.

Downloaded files are stored on the Virtual SD Memory Card. When uploading files, files that are stored on the Virtual SD Memory Card are uploaded to the FTP server. Therefore, to upload or download files, a Virtual SD Memory Card must be set in the NY-series Controller.



Precautions for Correct Use

To use a Virtual SD Memory Card, you need the settings for the SD Memory Card operations in the Controller. For information on setting the SD Memory Card operations, refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558).

11-1-2 Connectable FTP Servers

An NY-series Controller can connect to the following FTP servers. Refer to the relative manuals for information on setting and using the FTP servers.

- Built-in EtherNet/IP port on NY-series Controller

- FTP server in a built-in EtherNet/IP port on NJ/NX-series Controller
- CJ-series EtherNet/IP Unit with unit version 2.0 or later
- CJ-series CJ2 CPU Unit with Built-in EtherNet/IP
- CJ-series CJ1M CPU Unit with Ethernet Functions
- CJ-series Ethernet Unit
- Windows7: Windows Server 2008 R2 (Internet Information Services (IIS) 7.5)
- Windows8: Windows Server 2012 (IIS8.0)
- Linux

11-1-3 File Transfer Options

You can use the following options for file transfers. All options are specified in the parameters of the FTP client communications instructions.

- File transfer mode
- Open mode for data connection
- Deleting files after transfer
- Overwriting

The following sections describe each of these options.

● File Transfer Mode

There are two file transfer modes, ASCII Mode and Binary Mode, which differ in how line feeds in the text data are processed. The following table describes the differences.

Transfer mode	Handling of line feeds in text data
ASCII Mode	Line feeds are converted to the line feed code of the destination system, e.g., Unix or Windows.
Binary Mode	Line feeds are transferred without conversion.

● Open Mode for Data Connection

Opening a TCP connection between the FTP server and FTP client is necessary to transfer files. TCP connections include control connections to control communications and data connections to transfer data. There are also an Active Mode and a Passive Mode to open a data connection that differ in which party makes the connection request. The following table describes the differences.

Open mode	Request to establish a connection
Active Mode	The FTP server makes the connection request.
Passive Mode	The FTP client makes the connection request.

For example, if the FTP server is not on the Internet and you use Active Mode to open the data connection, the connection request from the FTP server may not be permitted due to security policies. In that case, you must use Passive Mode to open the data connection when you request a connection from the FTP client.

● File Deletion after Transfer

You can specify whether to delete the source files after the file transfer. If the file transfer fails for any reason, the source files are not deleted even if deletion is specified.

- **Overwriting**

You can specify whether to overwrite a file of the same name at the file transfer destination. If you specify not overwriting files and a file of the same name as the transferred file exists at the transfer destination, the source file is not transferred.

11-1-4 Other Functions

You can also use the following two functions for file transfers.

- Retrying connection processing with the FTP server
- Using wildcards to specify the files to transfer

These functions are described in the following sections.

- **Retrying Connection Processing with the FTP Server**

You can specify automatically retrying connection processing up to three times when connection processing to the FTP server fails. You can set the timeout time that is used to determine connection failure, the number of retries, and the retry interval.

- **Using Wildcards to Specify the Files to Transfer**

You can use wildcards to specify the names of the files to transfer. This allows you to transfer more than one file at one time.

11-2 FTP Client Communications Instructions

All file transfers that use the FTP client are executed with FTP client communications instructions. The FTP client communications instructions and their functions are given in the following table.

Instruction	Function
FTPGetFileList	Gets a file list from the FTP server.
FTPGetFile	Downloads one or more files from the FTP server.
FTPPutFile	Uploads one or more files to the FTP server.
FTPRemoveFile	Deletes one or more files on the FTP server.
FTPRemoveDir	Deletes a directory on the FTP server.

For details on the FTP client communications instructions, refer to the *NY-series Instructions Reference Manual* (Cat. No. W560).

11-2-1 Functions of the FTP Client Communications Instructions

This section describes the functions of the FTP client communications instructions.

FTPGetFileList Instruction

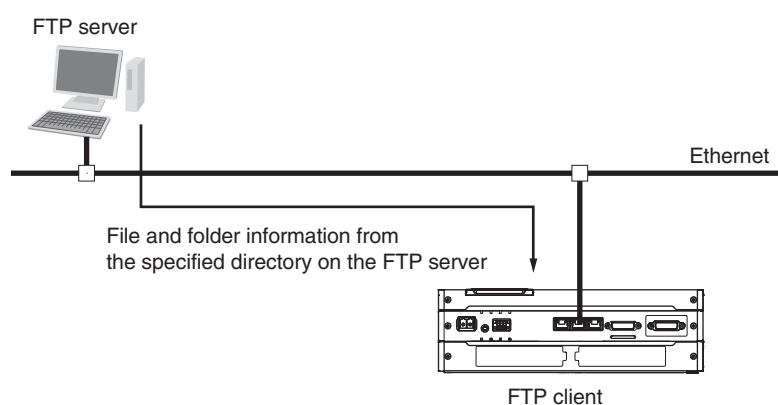
The FTPGetFileList instruction gets a list of the files and folders in a specified directory on the FTP server.

The following information is obtained.

- The number of files and folders in the specified directory
- The names of the files and folders
- The last updated dates of the files and folders
- The file sizes
- The read-only attributes of the files and folders

You can specify the following option.

- Open mode for data connection





Additional Information

The updated dates of files at 12 am and 12 pm are improved in the CPU Unit with unit version 1.14 or later.

FTPGetFile Instruction

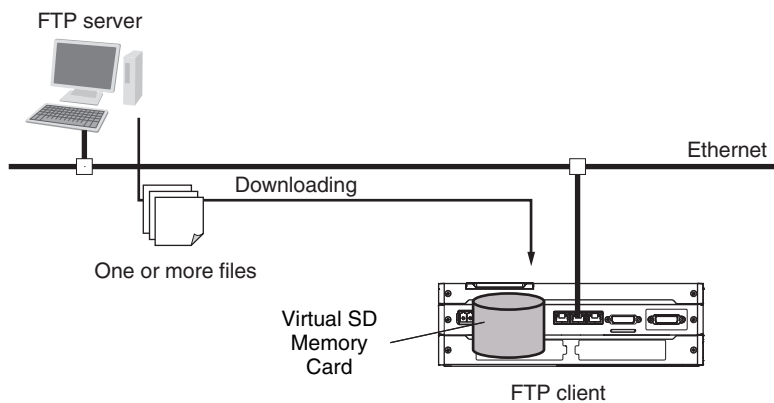
The FTPGetFile instruction downloads the specified file from the specified directory on the FTP server to the specified directory in the Virtual SD Memory Card.

You can use wildcards to specify the file name to allow you to download more than one file at the same time.

If the directory specified for the download does not exist in the Virtual SD Memory Card, the directory is created and the data is downloaded in it.

You can specify the following options.

- Transfer mode
- Open mode for data connection
- Deleting files after transfer
- Overwriting



FTPputFile Instruction

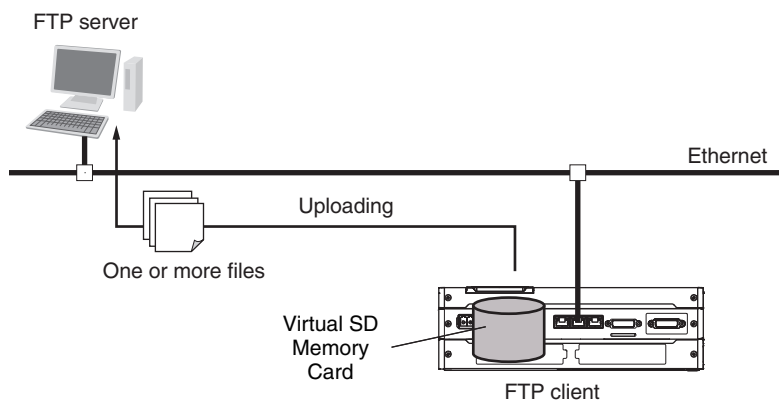
The FTPputFile instruction uploads the specified file from the specified directory in the Virtual SD Memory Card to the specified directory on the FTP server.

You can use wildcards to specify the file name to allow you to upload more than one file at the same time.

If the directory specified for the upload does not exist on the FTP server, the directory is created and the data is uploaded in it.

You can specify the following options.

- Transfer mode
- Open mode for data connection
- Deleting files after transfer
- Overwriting

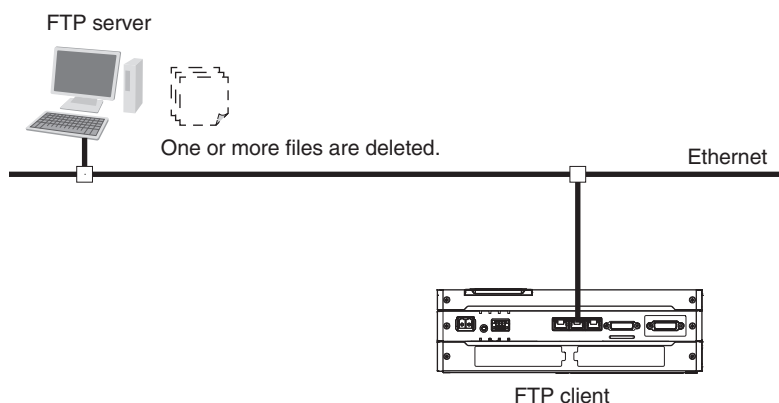


FTPRemoveFile Instruction

The FTPRemoveFile instruction deletes the specified file in the specified directory on the FTP server. You can use wildcards to specify the file name to allow you to delete more than one file at the same time.

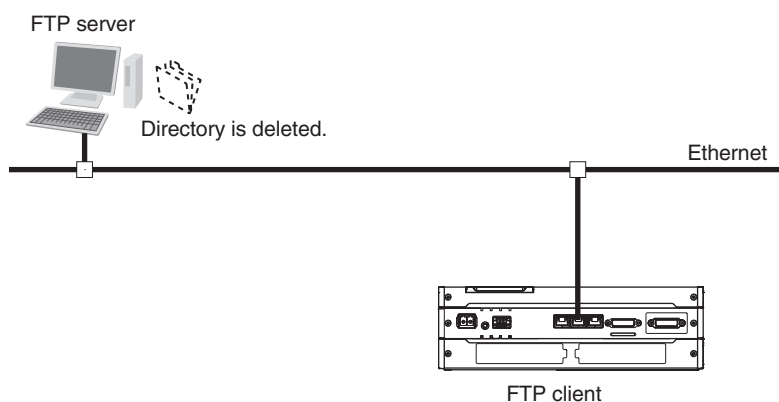
You can specify the following option.

- Open mode for data connection



FTPRemoveDir Instruction

The FTPRemoveDir instruction deletes the specified directory from the FTP server.



11-2-2 Restrictions on the FTP Client Communications Instructions

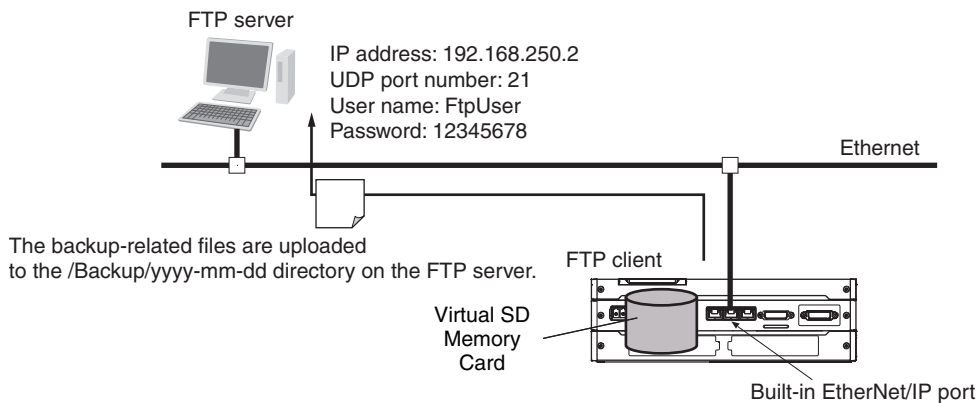
The following restrictions apply to the FTP client communications instructions. Be careful not to exceed these restrictions when you create the user program.

- If you execute more than one FTP client communications instruction to read and write data in a Virtual SD Memory Card at a time, unexpected operation may result, such as reading data from a file to which data is being written. Perform exclusive control of the instructions in the user program.
- If you execute an FTP client communications instruction to read or write data in a Virtual SD Memory Card at the same time as an operation to read or write data in the Virtual SD Memory Card, unexpected operation may result, such as reading data from a file to which data is being written. Perform exclusive control of the instructions in the user program. Refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558) for information on exclusive control of file access in Virtual SD Memory Cards.

11-3 FTP Client Application Example

All FTP client functionality is executed with FTP client communications instructions. This section provides sample programming that uses the FTP client communications instructions.

This programming executes an SD Memory Card backup and then uploads all of the backup-related files to the /Backup/yyyy-mm-dd directory on the FTP server.



The Controller is connected to the FTP server through an EtherNet/IP network. The settings of the parameters to connect to the FTP server are given in the following table.

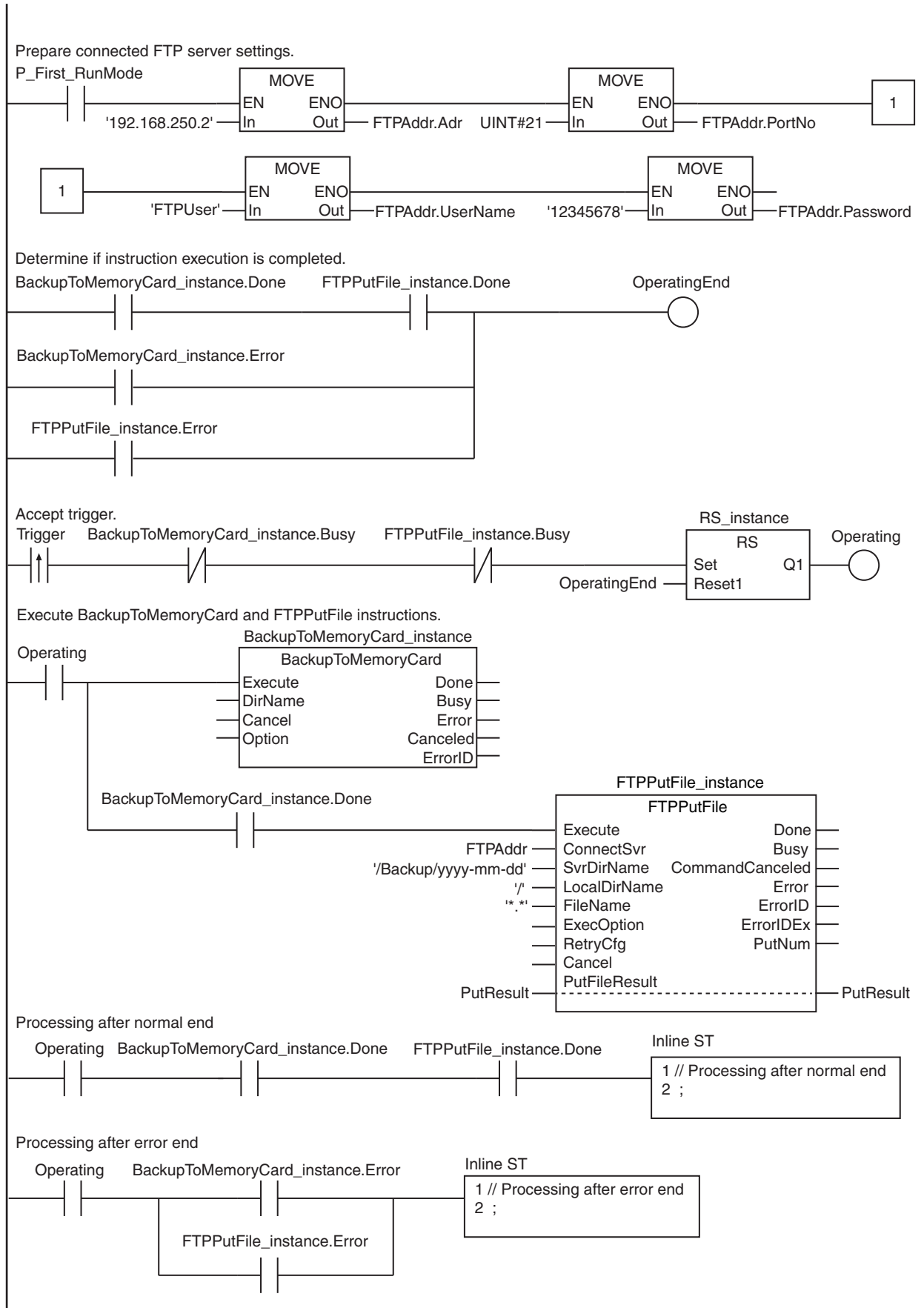
Parameter	Value
IP address	192.168.250.2
UDP port number	21
User name	FtpUser
Password	12345678

The following procedure is used.

- 1** The BackupToMemoryCard instruction is used to save NY-series Controller backup-related files to the root directory on the Virtual SD Memory Card.
- 2** The FTPPutFile instruction is used to upload the backup-related files to the /Backup/yyyy-mm-dd directory on the FTP server.
The wildcard specification *.* is used to specify the names of the files to transfer.
- 3** Normal end processing is executed if all processing ends normally. Processing for an error end is performed if an error occurs.

LD

Internal Variables	Variable	Data type	Initial value	Comment
	FTPPutFile_instance	FTPPutFile		Instance of FTPPutFile instruction
	FTPAddr	_sFTP_CONNECT_SVR	(Adr := ", PortNo := 0, UserName := ", Password := ")	Connected FTP server settings
	PutResult	ARRAY[0..0] OF _sFTP_FILE_RESULT	[(Name := ", TxError := False, RemoveError := False, Reserved := [4(16#0)])]	Uploaded file results
	RS_instance	RS		Instance of RS instruction
	OperatingEnd	BOOL	FALSE	Processing completed
	Trigger	BOOL	FALSE	Execution condition
	Operating	BOOL	FALSE	Processing
	BackupToMemoryCard_instance	BackupToMemoryCard		Instance of BackupToMemoryCard instruction



ST

Internal Variables	Variable	Data type	Initial value	Comment
	R_TRIG_instance	R_TRIG		Instance of R_TRIG instruction
	UP_Q	BOOL	FALSE	Trigger output
	FTPPutFile_instance	FTPPutFile		Instance of FTPPutFile instruction
	DoFTPTrigger	BOOL	FALSE	Execution condition for BackupToMemoryCard and FTPPutFile
	FTPAddr	_sFTP_CONNECT_SVR	(Adr := ", PortNo := 0, UserName := ", Password := ")	Connected FTP server settings
	PutResult	ARRAY[0..0] OF _sFTP_FILE_RESULT	[(Name := ", TxError := False, RemoveError := False, Reserved := [4(16#0)])]	Uploaded file results
	Stage	UINT	0	Instruction execution stage
	Trigger	BOOL	FALSE	Execution condition
	BackupToMemoryCard_instance	BackupToMemoryCard		Instance of BackupToMemoryCard instruction

```

// Prepare connected FTP server settings.
IF P_First_RunMode THEN
  FTPAddr.Adr      := '192.168.250.2'; // Address
  FTPAddr.PortNo  := UINT#21;         // Port number
  FTPAddr.UserName := 'FtpUser';      // User name
  FTPAddr.Password := '12345678';     // Password
END_IF;

// Accept trigger.
R_TRIG_instance(Trigger, UP_Q);
IF ( (UP_Q = TRUE) AND (BackupToMemoryCard_instance.Busy = FALSE) AND
    (FTPPutFile_instance.Busy = FALSE) ) THEN
  DoFTPTrigger := TRUE;
  Stage := INT#1;
  BackupToMemoryCard_instance( // Initialize instance.
    Execute := FALSE);
  FTPPutFile_instance( // Initialize instance.
    Execute      := FALSE,
    ConnectSvr   := FTPAddr,
    SvrDirName   := '/Backup/yyyy-mm-dd',
    LocalDirName := '/',
    FileName     := '*.*',
    PutFileResult := PutResult);
END_IF;

IF (DoFTPTrigger = TRUE) THEN
  CASE Stage OF
    1: // Execute BackupToMemoryCard instruction.
      BackupToMemoryCard_instance(
        Execute := TRUE, // Execution
        IF (BackupToMemoryCard_instance.Done = TRUE) THEN
          Stage := INT#2; // To next stage
        ELSIF (BackupToMemoryCard_instance.Error = TRUE) THEN
          Stage := INT#10; // Error end
        END_IF;
    2: // Execute FTPPutFile instruction.

```

```

FTPputFile_instance(
    Execute           := TRUE,                // Execution
    ConnectSvr       := FTPAddr,            // Connected FTP server
    SvrDirName        := '/Backup/yyyy-mm-dd', // FTP server directory name
    LocalDirName      := '/',                // Local directory name
    FileName          := '.*.*',            // File name
    PutFileResult     := PutResult) ;        // Uploaded file results
IF (FTPputFile_instance.Done = TRUE) THEN
    Stage := INT#0; // Normal end
ELSIF (FTPputFile_instance.Error = TRUE) THEN
    Stage := INT#20; // Error end
END_IF;
0: // Processing after normal end
DoFTPTrigger:=FALSE;
Trigger      :=FALSE;
ELSE // Processing after error end
DoFTPTrigger:=FALSE;
Trigger      :=FALSE;
END_CASE;
END_IF;

```

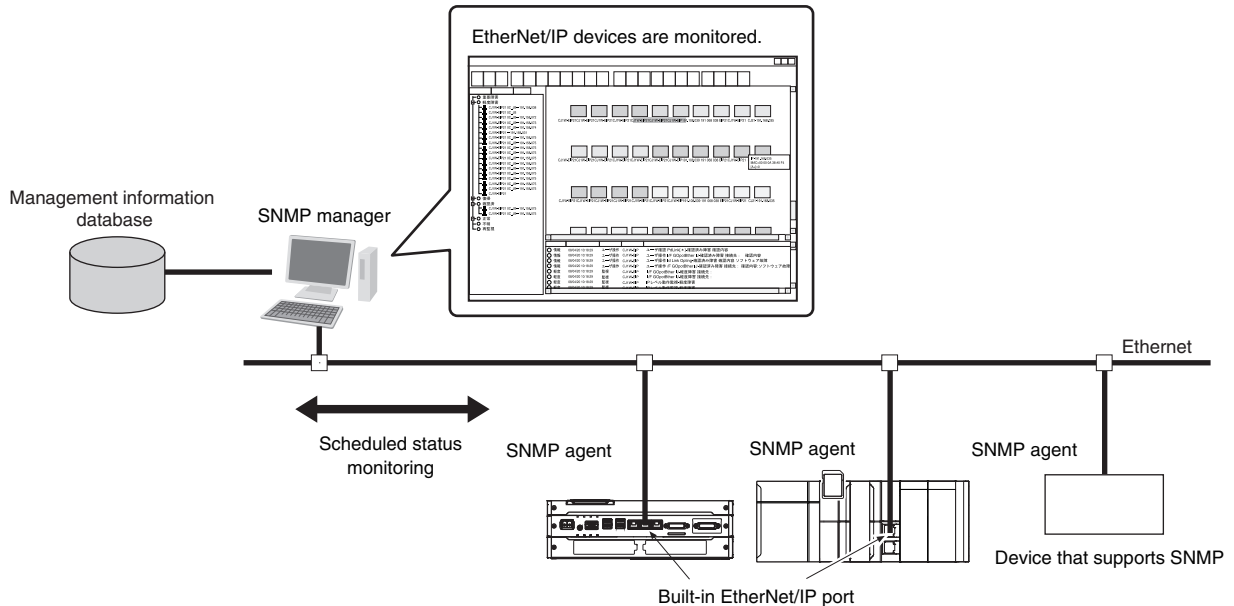

12

SNMP Agent

12-1 SNMP Agent	12-2
12-1-1 Overview	12-2
12-1-2 Specifications	12-3
12-1-3 SNMP Messages	12-3
12-1-4 MIB Specifications	12-4
12-2 Procedure to Use the SNMP Agent	12-21
12-2-1 Procedures	12-21
12-2-2 Settings Required for the SNMP Agent	12-21

12-1 SNMP Agent

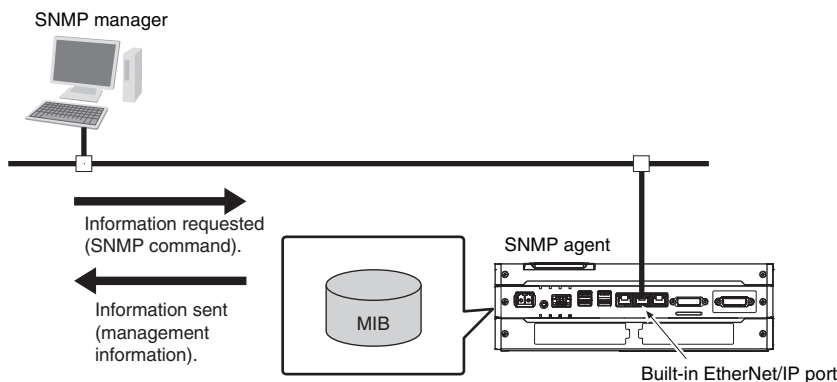
The SNMP (simple network management protocol) is a network management protocol. You can use the SNMP to manage any network that consists of devices that support SNMP. The server that manages the network is called the SNMP manager. The managed network devices are called SNMP agents.



12-1-1 Overview

SNMP Agent

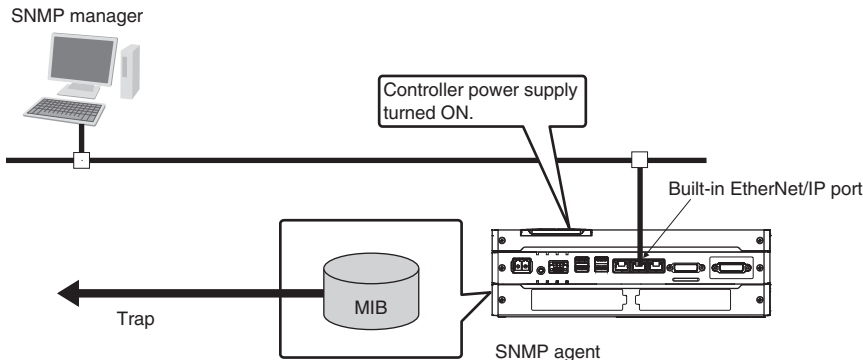
The built-in Ethernet/IP port has its own management information called the MIB (management information base). This information can be provided to the SNMP manager. The SNMP manager is software that gathers and processes information about devices on the SNMP network and provides that information to the network administrator. You can use the SNMP manager to monitor the built-in Ethernet/IP port.



The SNMP manager has a SNMP command to request MIB information. The built-in Ethernet/IP port SNMP agent function supports SNMPv1 (RFC1157) and SNMPv2C (RFC1901). Use the SNMPv1 or SNMPv2C protocol to manage the built-in Ethernet/IP port with the SNMP manager. You can also use both the SNMPv1 and SNMPv2C protocols together at the same time.

SNMP Traps

When a failure or some other specific problem occurs, a status report called a trap is sent. This enables monitoring changes in status even if the SNMP manager does not monitor the built-in EtherNet/IP port periodically. However, traps use UDP. Therefore, you cannot check to see if the SNMP manager receives traps from the EtherNet/IP port. Thus, depending on the network status, some traps may not reach the SNMP manager.

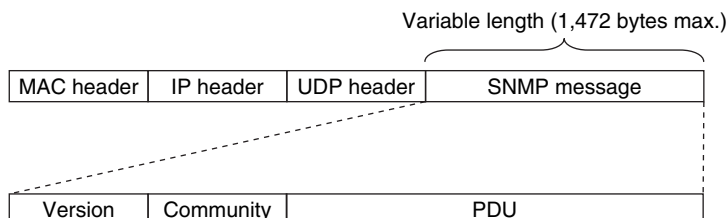


12-1-2 Specifications

Item	Specification
Protocol	SNMP
Agent	SNMPv1, SNMPv2c
MIB	MIB-II
Port No.	SNMP agent: 161 (UDP) SNMP trap: 162 (UDP) These can be changed in the Built-in EtherNet/IP Port Settings from the Sysmac Studio.
Timing of SNMP trap operation	Status reports are sent to the SNMP manager at the following times. <ul style="list-style-type: none"> • When the Controller is turned ON • When links are established • When an SNMP agent fails to be authorized
Supported MIB commands	GetRequest/GetNextRequest

12-1-3 SNMP Messages

The structure of SNMP messages is as follows:



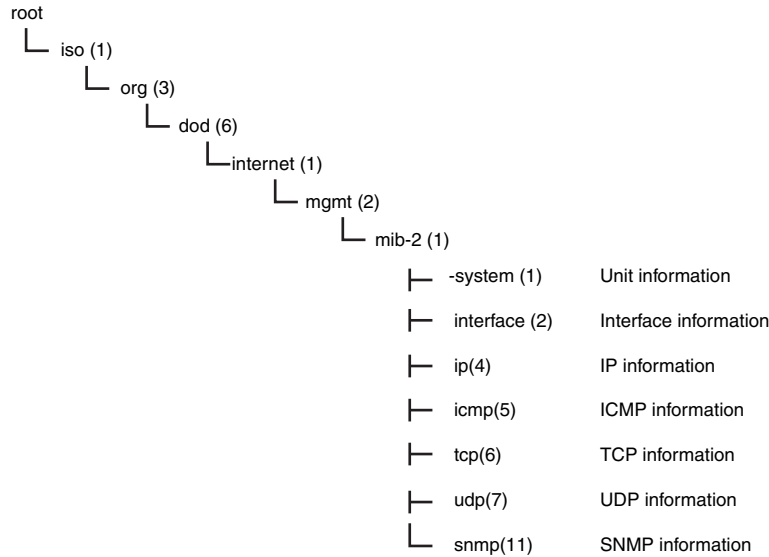
Item	Set value
Version	This value gives the SNMP version. SNMPv1: 0 SNMPv2c: 1
Community	Community name for verification
PDU	This depends on the PDU type.

12-1-4 MIB Specifications

This section describes the specifications of the MIB that is supported by the built-in EtherNet/IP port.

MIB System Diagram

The built-in EtherNet/IP port MIB consists of the following tree structure.



MIB Groups

MIB group		Stored information	
Standard MIB	<i>system</i> group	The MIB for information related to the device.	
	<i>interfaces</i> group	The MIB for information related to the interface.	
	<i>ip</i> group	ip	The MIB for IP information.
		ipAddrTable	The MIB for addressing table information related to IP addresses.
		ipRouteTable	The MIB for information related to IP routing tables.
		ipNetToMediaTable	The MIB for information related to IP address conversion tables.
		ipForward	The MIB for information related to IP forwarding tables.
	<i>icmp</i> group	The MIB for ICMP information.	
	<i>tcp</i> group	tcp	The MIB for TCP information.
	<i>udp</i> group	udp	The MIB for UDP information.
<i>snmp</i> group	snmp	The MIB for SNMP information.	

Detailed Descriptions of MIB Objects

● System Group

Subtree name	Standard [(<i>identifier</i>) <i>attribute</i>]	Support	Implementation specifications
sysDescr	(1) RO Device information (including hardware, OS, software names, and versions) ASCII characters only.	Supported.	“OMRON Corporation” + CPU Unit model + CPU Unit version <ul style="list-style-type: none"> • CPU Unit model (example): NY512-1300 • CPU Unit version (example): Version 1.12
sysObjectID	(2) RO Vendor OID. Tells where this device information was assigned in the private MIB.	Supported.	1.3.6.1.4.1.16838.1.1025.6
sysUpTime	(3) RO The time elapsed since the system was started (unit: 1/100 s).	Supported.	According to the standard.
sysContact	(4) RW How to contact the administrator and information on the administrator.	Supported.	Set by the user.
sysName	(5) RW The name for management. Sets the full domain name of the device.	Supported.	CPU Unit name
sysLocation	(6) RW The physical location of the device.	Supported.	Set by the user.
sysServices	(7) RO The value of the provided service.	Supported.	64

● Interfaces Group

Subtree name	Standard [(<i>identifier</i>) <i>attribute</i>]	Support	Implementation specifications
ifNumber	(1) RO The number of network interfaces.	Supported.	3
ifTable	(2) NA Interface entity table	---	
ifEntry	(1) NA Row data for interface information The index is <i>ifIndex</i> .	---	
ifIndex	(1) RO A number used to identify the interface.	Supported.	1 to 3
ifDescr	(2) RO Information related to the interface (includes manufacturer name, product name, and hardware interface version).	Supported.	1. Loop Back 2. 10/100/1000M Gigabit Ethernet Port 3. Internal Network Port
ifType	(3) RO The type of interface classified according to the physical/link layer protocol directly under the network layer of the protocol stack.	Supported.	ethernet-csmacd(6)
ifMtu	(4) RO MTU value The maximum size (in octets) of datagrams that can be sent and received through this interface.	Supported.	1,500
ifSpeed	(5) RO Estimated bandwidth If a stable, accurate value cannot be obtained for the bandwidth, a nominal value is set instead.	Supported.	10000000/ 100000000/ 1000000000
ifPhysAddress	(6) RO MAC address The physical address under the network layer of the interface.	Supported.	The MAC address of the EtherNet/IP port.
ifAdminStatus	(7) RW The preferred status of the interface. You cannot send normal packets in the testing state. up(1) down(2) testing(3)	Supported.	According to the standard.
ifOperStatus	(8) RO The current status of the interface. You cannot send normal packets in the testing state. up(1) down(2) testing(3)	Supported.	According to the standard.

Subtree name	Standard [(<i>identifier</i>) <i>attribute</i>]	Support	Implementation specifications
ifLastChange	(9) RO The <i>sysUpTime</i> (in 0.01seconds) at the last change in <i>ifOperStatus</i> for this interface.	Supported.	According to the standard.
ifInOctets	(10) RO The number of octets received through this interface. This includes framing characters.	Supported.	According to the standard.
ifInUcastPkts	(11) RO The number of unicast packets reported to a higher level protocol.	Supported.	According to the standard.
ifInNUcastPkts	(12) RO The number of non-unicast packets (broadcast or multicast packets) reported to a higher level protocol.	Supported.	According to the standard.
ifInDiscards	(13) RO The number of packets that had no errors but could not be passed to a higher level protocol (i.e., the number of packets received but discarded due to a buffer overflow).	Supported.	According to the standard.
ifInErrors	(14) RO The number of packets discarded because they contained errors.	Supported.	According to the standard.
ifInUnknown Protos	(15) RO The number of packets received, but discarded because they were of an illegal or unsupported protocol. For example, Ethernet packets did not have IP set for the field that identifies their higher level protocol.	Supported.	According to the standard.
ifOutOctets	(16) RO The number of octets of packets sent through this interface. This includes framing characters.	Supported.	According to the standard.
ifOutUcast Pkts	(17) RO The number of unicast packets sent by higher level protocols. This includes discarded packets and unsent packets.	Supported.	According to the standard.
ifOutNUcast Pkts	(18) RO The number of non-unicast packets sent by higher level protocols. This includes discarded packets and unsent packets.	Supported.	According to the standard.
ifOutDiscards	(19) RO The number of packets that had no errors but were discarded in the sending process (due to a send buffer overflow, etc.).	Supported.	According to the standard.
ifOutErrors	(20) RO The number of packets that could not be sent because of an error.	Supported.	According to the standard.
ifOutQLen	(21) RO The size of the send packet queue (i.e., the number of packets).	Supported.	Always 0.

Subtree name		Standard [(<i>identifier</i>) <i>attribute</i>]	Support	Implementation specifications
	ifSpecific	(22) RO The object ID that represents a reference to the media-specific MIB for the interface. For example, for Ethernet, set the object ID of the MIB that defines Ethernet. If there is no information, set { 0.0 }.	Supported.	0.0

● Ip Group: Ip

Subtree name	Standard [(<i>identifier</i>) attribute]	Support	Implementation specifications
ipForwarding	(1) RW Indicates if the device operates as a gateway. IP gateways can transfer datagrams, but IP hosts can perform only source routing. Some nodes take only one of these values. Therefore, if you attempt to change this object from the SNMP Manager, a <i>badValue</i> error is returned. Forwarding (1) Not-forwarding (2)	Supported.	<ul style="list-style-type: none"> Forwarding (1) Not-forwarding (2) Depends on the settings.
IpDefaultTTL	(2) RW The default value set for the IP header TTL if no TTL value was given by the transport layer protocol.	Supported.	64
IpInReceives	(3) RO The number of all IP datagrams that reached the interface, including errors.	Supported.	According to the standard.
IpInHdrErrors	(4) RO The number of received datagrams that were discarded because of an IP header error (checksum error, version number error, format error, TTL error, IP option error, etc.).	Supported.	According to the standard.
IpInAddrErrors	(5) RO The number of packets that were discarded because the destination address in the IP header was not valid.	Supported.	According to the standard.
ipForwDatagrams	(6) RO The number of IP datagrams that were transferred to their final destination. If this node does not operate as an IP gateway, this is the number of datagrams that were successfully transferred through source routing.	Supported.	According to the standard.
ipInUnknownProtos	(7) RO The number of IP datagrams that were received but discarded because they were of an unsupported or unrecognized protocol.	Supported.	According to the standard.
ipInDiscards	(8) RO The number of IP datagrams that could have continued to be processed without any problems, but were discarded (for example, because of insufficient buffer space).	Supported.	According to the standard.
ipInDelivers	(9) RO The number of datagrams delivered to an IP user protocol (any higher level protocol, including ICMP).	Supported.	According to the standard.
ipOutRequests	(10) RO The number of times a send request was made for an IP datagram by a local IP user protocol (any higher level protocol, including ICMP). This counter does not include <i>ipForwDatagrams</i> .	Supported.	According to the standard.
ipOutDiscards	(11) RO The number of IP datagrams that could have been sent without any problems, but were discarded (for example, because of insufficient buffer space).	Supported.	According to the standard.

Subtree name	Standard [(<i>identifier</i>) <i>attribute</i>]	Support	Implementation specifications
ipOutNoRoutes	(12) RO The number of IP datagrams that were discarded because there was no transmission path. This counter includes datagrams that attempted to be sent through <i>ipForwDatagrams</i> , but were discarded because they were set with no-route. This value indicates the number of datagrams that could not be transferred because the default gateway was down.	Supported.	According to the standard.
ipReasmTimeout	(13) RO The maximum number of seconds to wait to receive all IP datagrams for reassembly if a fragmented IP datagram is received.	Supported.	60 s
ipReasmReqds	(14) RO The number of IP datagrams received that require reassembly. There is a flag in the IP header that indicates if the datagram is fragmented. You can use that flag to identify fragments.	Supported.	According to the standard.
ipReasmOKs	(15) RO The number of IP datagrams received that were successfully reassembled.	Supported.	According to the standard.
ipReasmFails	(16) RO The number of IP datagrams received that were not successfully reassembled.	Supported.	According to the standard.
ipFragOKs	(17) RO The number of IP datagrams that were successfully fragmented.	Supported.	According to the standard.
ipFragFails	(18) RO The number of IP datagrams that were not successfully fragmented. (For example, because the Don't Fragment flag was set for the IP datagram.)	Supported.	According to the standard.
ipFragCreates	(19) RO The number of IP datagrams created as a result of fragmentation.	Supported.	According to the standard.
ipAddrTable	(20) NA An address information table for IP addresses.	---	---
ipAddrEntry	(1) NA Row data of address information for IP addresses. The index is <i>ipAdEntAddr</i> .	---	---
ipAdEntAddr	(1) RO The IP address.	Supported.	According to the standard.
ipAdEntIfIndex	(2) RO The index value of the interface that this entry applies to. This is the same value as <i>ifIndex</i> .	Supported.	According to the standard.
ipAdEntNetMask	(3) RO The subnet mask for the IP address of this entry.	Supported.	According to the standard.
ipAdEntB-castAddr	(4) RO The value of the least significant bit of the address when an IP broadcast is sent. An address represented by all 1 bits is used for broadcasting as an Internet standard. In that case, this value is always 1.	Supported.	According to the standard.

Subtree name		Standard [(<i>identifier</i>) <i>attribute</i>]	Support	Implementation specifications
	ipAdEntReasmMaxSize	(5) RO The maximum IP packet size that can be reassembled from IP fragmented input IP datagrams received through the interface.	Supported.	According to the standard.
ipRouteTable		(21) NA The IP routing table for this entity.	---	---
ipRouteEntry		(1) NA Route information for a specific destination. The index is <i>ipRouteDest</i> .	---	---
	ipRouteDest	(1) RW The destination IP address for this route. A value of 0.0.0.0 for this entry indicates the default route.	Supported.	According to the standard.
	ipRouteIfIndex	(2) RW The ID number of the interface required to send to the next destination host in this route. This ID number is the same number as <i>ifIndex</i> , which is used to identify the interface.	Supported.	According to the standard.
	ipRouteMetric1	(3) RW The primary routing metric for this route. This value is determined based on the protocol specified in <i>ipRouteProto</i> . Set to -1 if you do not want to use this metric (this is also the same for <i>ipRouteMetric 2</i> through 4).	Supported.	According to the standard.
	ipRouteMetric2	(4) RW The alternative routing metric for this route.	Supported.	According to the standard.
	ipRouteMetric3	(5) RW The alternative routing metric for this route.	Supported.	According to the standard.
	ipRouteMetric4	(6) RW The alternative routing metric for this route.	Supported.	According to the standard.
	ipRouteNextHop	(7) RW The IP address of the next hop in this route (for routes connected by a broadcast or media, this is the agent address or address of that interface).	Supported.	According to the standard.
	ipRouteType	(8) RW The type of route. Other (1): Not any of the following types. Invalid (2): An invalid route. Direct (3): A direct connection. Indirect (4): An indirect connection (not connected to LOCAL).	Supported.	According to the standard.

Subtree name	Standard [(<i>identifier</i>) <i>attribute</i>]	Support	Implementation specifications
ipRouteProto	<p>(9) RO</p> <p>This is the routing mechanism used to determine routes. Some values correspond to gateway routing protocols, but be aware that the host may not support those protocols.</p> <p>Other (1): Other than the following items.</p> <p>Local (2): A route set on the local machine.</p> <p>Netmgmt (3): A route set by network management.</p> <p>Icmp (4): A route set by an ICMP redirect or some other ICMP function.</p> <p>Egp (5): EGP</p> <p>The following are gateway protocols:</p> <p>Ggp (6): GGP</p> <p>Hello (7): HELLO</p> <p>Rip (8): RIP</p> <p>is-is (9)</p> <p>es-is (10)</p> <p>ciscoIgrp (11)</p> <p>bbnSpfIgrp (12)</p> <p>ospf (13): OSPF</p> <p>bgp (14)</p>	Supported.	According to the standard.
ipRouteAge	<p>(10) RW</p> <p>The elapsed time since this route was updated (in seconds).</p>	Supported.	Always 0.
ipRouteMask	<p>(11) RW</p> <p>The subnet mask value in relation to <i>ipRouteDest</i>. On systems that do not support a custom subnet mask value, this value is based on the address class of the <i>ipRouteDest</i> field. If <i>ipRouteDest</i> is 0.0.0.0, this value is also 0.0.0.0.</p>	Supported.	According to the standard.
ipRouteMetric5	<p>(12) RW</p> <p>The alternative routing metric.</p>	Supported.	According to the standard.
ipRouteInfo	<p>(13) RO</p> <p>The MIB object ID for the routing protocol used by this route. If not defined, set to {0.0}.</p>	Supported.	0.0

Subtree name	Standard [(<i>identifier</i>) attribute]	Support	Implementation specifications
ipNetToMediaTable	(22) NA The IP address conversion table used to map IP addresses to physical addresses.	---	---
ipNetToMediaEntry	(1) NA Row data for the conversion table. The indices are <i>ipNetToMediaIfIndex</i> and <i>ipNetToMediaNetAddress</i> .	---	---
ipNetToMediaIfIndex	(1) RW The interface ID number for this entry. The value of <i>ifIndex</i> is used for this value.	Supported.	According to the standard.
ipNetToMediaPhysAddress	(2) RW The media-dependent physical address.	Supported.	According to the standard.
ipNetToMediaNetAddress	(3) RW The IP address that corresponds to the media-dependent physical address.	Supported.	According to the standard.
ipNetToMedia-Type	(4) RW The address conversion method. Other (1): A method other than the following items. Invalid (2): An invalid value. Dynamic (3): Dynamic conversion. Static (4): Static conversion.	Supported.	According to the standard.
ipRoutingDiscards	(23) RO The number of routing entries that were valid but discarded. For example, if there was not enough buffer space because of other routing entries.	Supported.	According to the standard.

● Ip Group: Icmp

Name	Standard [(<i>identifier</i>) attribute]	Support	Implementation specifications
icmpInMsgs	(1) RO The total number of received ICMP messages. This includes messages counted by <i>icmpInErrors</i> .	Supported.	According to the standard.
icmpInErrors	(2) RO The number of received ICMP message errors. (Checksum errors, frame length errors, etc.)	Supported.	According to the standard.
icmpInDestUnreachs	(3) RO The number of <i>Destination Unreachable</i> messages received.	Supported.	According to the standard.
icmpInTimeExcds	(4) RO The number of <i>Time Exceed</i> messages received.	Supported.	According to the standard.
icmpInParmProbs	(5) RO The number of <i>Parameter Problem</i> messages received.	Supported.	According to the standard.
icmpInSrcQuenchs	(6) RO The number of <i>Source Quench</i> messages received.	Supported.	According to the standard.
icmpInRedirects	(7) RO The number of <i>Redirect</i> messages received.	Supported.	According to the standard.
icmpInEchos	(8) RO The number of <i>Echo (request)</i> messages received.	Supported.	According to the standard.
icmpInEchoReps	(9) RO The number of <i>Echo Reply</i> messages received.	Supported.	According to the standard.
icmpInTimestamps	(10) RO The number of <i>Timestamp</i> messages received.	Supported.	According to the standard.
icmpInTimestampReps	(11) RO The number of <i>Timestamp Reply</i> messages received.	Supported.	According to the standard.
icmpInAddrMasks	(12) RO The number of <i>Address Mask Request</i> messages received.	Supported.	According to the standard.
icmpInAddrMaskReps	(13) RO The number of <i>Address Mask Reply</i> messages received.	Supported.	According to the standard.
icmpOutMsgs	(14) RO The total number of ICMP messages sent. This includes messages counted by <i>icmpOutErrors</i> .	Supported.	According to the standard.
icmpOutErrors	(15) RO The number of ICMP messages that could not be sent because of an error.	Supported.	According to the standard.
icmpOutDestUnreachs	(16) RO The number of <i>Destination Unreachable</i> messages sent.	Supported.	According to the standard.
icmpOutTimeExcds	(17) RO The number of <i>Time Exceed</i> messages sent.	Supported.	According to the standard.
icmpOutParmProbs	(18) RO The number of <i>Parameter Problem</i> messages sent.	Supported.	According to the standard.
icmpOutSrcQuenchs	(19) RO The number of <i>Source Quench</i> messages sent.	Supported.	According to the standard.

Name	Standard [(<i>identifier</i>) <i>attribute</i>]	Support	Implementation specifications
icmpOutRedirects	(20) RO The number of <i>Redirect</i> messages sent.	Supported.	According to the standard.
icmpOutEchos	(21) RO The number of <i>Echo (request)</i> messages sent.	Supported.	According to the standard.
icmpOutEchoReps	(22) RO The number of <i>Echo Reply</i> messages sent.	Supported.	According to the standard.
icmpOutTimestamps	(23) RO The number of <i>Timestamp</i> messages sent.	Supported.	According to the standard.
icmpOutTimestampReps	(24) RO The number of <i>Timestamp Reply</i> messages sent.	Supported.	According to the standard.
icmpOutAddrMasks	(25) RO The number of <i>Address Mask Request</i> messages sent.	Supported.	According to the standard.
icmpOutAddrMaskReps	(26) RO The number of <i>Address Mask Reply</i> messages sent.	Supported.	According to the standard.

● Ip Group: Tcp

Name	Standard [(<i>identifier</i>) attribute]	Support	Implementation specifications
tcpRtoAlgorithm	(1) RO The algorithm used to determine the timeout value for resending. Other (1): Other than the following items. Constant (2): A constant RTO value. Rsre (3): The algorithm specified by the MIL-STD-1778 standard. Vanj (4): The Van Jacobson algorithm.	Supported.	According to the standard.
tcpRtoMin	(2) RO The minimum resend timeout value (in 0.01 s). This value depends on the algorithm used to determine the resend timeout value.	Supported.	According to the standard.
tcpRtoMax	(3) RO The maximum resend timeout value (in 0.01 s). This value depends on the algorithm used to determine the resend timeout value.	Supported.	According to the standard.
tcpMaxConn	(4) RO The total number of supported TCP connections. If the maximum number of connections is dynamic, this value is -1.	Supported.	According to the standard.
tcpActiveOpens	(5) RO The number of times the TCP connection changed from the CLOSE state directly to the SYN-SENT state. (Active connection establishment.)	Supported.	According to the standard.
tcpPassiveOpens	(6) RO The number of times the TCP connection changed from the LISTEN state directly to the SYN-RCVD state. (Passive connection establishment.)	Supported.	According to the standard.
tcpAttemptFails	(7) RO The total number of times the TCP connection changed from the SYN-SENT or SYN-RCVD state directly to the CLOSE state and from the SYN-RCVD state directly to the LISTEN state.	Supported.	According to the standard.
tcpEstabResets	(8) RO The number of times the TCP connection changed from the ESTABLISHED or the CLOSE-WAIT state directly to the CLOSE state.	Supported.	According to the standard.
tcpCurrEstab	(9) RO The total number of TCP connections currently in the ESTABLISHED or the CLOSE-WAIT state.	Supported.	According to the standard.
tcpInSegs	(10) RO The total number of received segments. This includes the number of error segments.	Supported.	According to the standard.
tcpOutSegs	(11) RO The total number of sent segments. This includes the number of segments for the current connection, but does not include the number of segments for resent data only.	Supported.	According to the standard.
tcpRetransSegs	(12) RO The total number of resent segments.	Supported.	According to the standard.

Name	Standard [(<i>identifier</i>) attribute]	Support	Implementation specifications
tcpConnTable	(13) NA The information table specific to the TCP connection.	---	According to the standard.
tcpConnEntry	(1) NA Entry information related to a specific TCP connection. This value is deleted if the connection changes to the CLOSE state. The indices are <i>tcpConnLocalAddress</i> , <i>tcpConnLocalPort</i> , <i>tcpConnRemAddress</i> , and <i>tcpConnRemPort</i> .	---	According to the standard.
tcpConnState	(1) RW The status of the TCP connection. closed(1) listen(2) synSent(3) synReceived(4) established(5) finWait1(6) finWait2(7) closeWait(8) lastAck(9) closing(10) timeWait(11)	Supported.	According to the standard.
tcpConnLocalAddress	(2) RO The local IP address of this TCP connection. A value of 0.0.0.0 is used for connections in the LISTEN state that accept connections from any IP interface related to the node.	Supported.	According to the standard.
tcpConnLocalPort	(3) RO The local port number for this TCP connection.	Supported.	According to the standard.
tcpConnRemAddress	(4) RO The remote IP address for this TCP connection.	Supported.	According to the standard.
tcpConnRemPort	(5) RO The remote port number for this TCP connection.	Supported.	According to the standard.
tcpInErrs	(14) RO The total number of error segments received (TCP checksum errors, etc.).	Supported.	According to the standard.
tcpOutRsts	(15) RO The number of segments sent with the RST flag (the number of times the TCP connection was reset).	Supported.	According to the standard.

● Ip Group: Udp

Name	Standard [(<i>identifier</i>) attribute]	Support	Implementation specifications
udpInDatagrams	(1) RO The total number of UDP datagrams (i.e., the number of packets) sent to the UDP user.	Supported.	According to the standard.
udpNoPorts	(2) RO The number of UDP datagrams that were received but did not start an application at the destination port.	Supported.	According to the standard.
udpInErrors	(3) RO The number of UDP datagrams that were not sent to a higher level protocol for a reason other than <i>udpNoPorts</i> .	Supported.	According to the standard.
udpOutDatagrams	(4) RO The total number of sent UDP datagrams.	Supported.	According to the standard.
udpTable	(5) NA The information table for the UDP listener.	---	According to the standard.
udpEntry	(1) NA An entry related to a specific UDP listener. The indices are <i>udpLocalAddress</i> and <i>udpLocalPort</i> .	---	According to the standard.
udpLocalAddress	(1) RO The local IP address of this UDP listener. A value of 0.0.0.0 is used for UDP listeners that accept datagrams from any IP interface related to the node.	Supported.	According to the standard.
udpLocalPort	(2) RO The local port number for this UDP listener.	Supported.	According to the standard.

● Ip Group: Snmp

Name	Standard [(<i>identifier</i>) attribute]	Support	Implementation specifications
snmplnPkts	(1) RO The total number of SNMP messages received.	Supported.	According to the standard.
snmpOutPkts	(2) RO The total number of SNMP messages sent.	Supported.	According to the standard.
snmplnBadVersions	(3) RO The total number of messages received of an unsupported version.	Supported.	According to the standard.
snmplnBadCommunityNames	(4) RO The total number of messages received from an unregistered community.	Supported.	According to the standard.
snmplnBadCommunityUses	(5) RO The total number of messages received that specify an operation that is not allowed by that community.	Supported.	According to the standard.
snmplnASNParseErrs	(6) RO The total number of messages received that resulted in an ASN.1 error or BER error during decoding.	Supported.	According to the standard.
snmplnTooBig	(8) RO The total number of PDUs received with an error status of <i>tooBig</i> .	Supported.	According to the standard.

Name		Standard [(identifier) attribute]
snmpInNoSuchNames	(9) RO The total number of PDUs received with an error status of <i>noSuchName</i> .	Supported. According to the standard.
snmpInBadValues	(10) RO The total number of PDUs received with an error status of <i>badValue</i> .	Supported. According to the standard.
snmpInReadOnlys	(11) RO The total number of PDUs received with an error status of <i>readOnly</i> .	Supported. According to the standard.
snmpInGenErrs	(12) RO The total number of PDUs received with an error status of <i>genErr</i> .	Supported. According to the standard.
snmpInTotalReqVars	(13) RO The total number of MIB objects read normally after receiving <i>GetRequest</i> or <i>GetNextRequest</i> .	Supported. According to the standard.
snmpInTotalSetVars	(14) RO The total number of MIB objects updated normally after receiving <i>SetRequest</i> .	Supported. According to the standard.
snmpInGetRequests	(15) RO The total number of <i>GetRequest</i> PDUs received.	Supported. According to the standard.
snmpInGetNexts	(16) RO The total number of <i>GetNextRequest</i> PDUs received.	Supported. According to the standard.
snmpInSetRequests	(17) RO The total number of <i>SetRequest</i> PDUs received.	Supported. According to the standard.
snmpInGetResponses	(18) RO The total number of <i>GetResponse</i> PDUs received.	Supported. According to the standard.
snmpInTraps	(19) RO The total number of trap PDUs received.	Supported. According to the standard.
snmpOutTooBig	(20) RO The total number of PDUs sent with an error status of <i>tooBig</i> .	Supported. According to the standard.
snmpOutNoSuchNames	(21) RO The total number of PDUs sent with an error status of <i>noSuchName</i> .	Supported. According to the standard.
snmpOutBadValues	(22) RO The total number of PDUs sent with an error status of <i>badValue</i> .	Supported. According to the standard.
snmpOutGenErrs	(24) RO The total number of PDUs sent with an error status of <i>genErr</i> .	Supported. According to the standard.
snmpOutGetRequests	(25) RO The total number of <i>GetRequest</i> PDUs sent.	Supported. According to the standard.
snmpOutGetNexts	(26) RO The total number of <i>GetNextRequest</i> PDUs sent.	Supported. According to the standard.
snmpOutSetRequests	(27) RO The total number of <i>SetRequest</i> PDUs sent.	Supported. According to the standard.
snmpOutGetResponses	(28) RO The total number of <i>GetResponse</i> PDUs sent.	Supported. According to the standard.
snmpOutTraps	(29) RO The total number of trap PDUs sent.	Supported. According to the standard.

Name			Standard [(identifier) attribute]
snmpEnableAuthenTraps	(30) RW Determines if the agent generates verification failed traps. Enabled (1) Disabled (2)	Supported.	According to the standard.

12-2 Procedure to Use the SNMP Agent

12-2-1 Procedures

- 1** Make the basic settings.
Refer to *1-5 EtherNet/IP Communications Procedures* for the flow of basic operations.
- 2** Select **Built-in EtherNet/IP Port Settings** from the **Controller Setup** on the Sysmac Studio, and then set the following on the SNMP Settings Display or SNMP Trap Settings Display.
 - SNMP Service
 - Recognition 1
 - Recognition 2
- 3** Select **Transfer to Controller** from the Controller Menu and click the **Yes** Button. The built-in EtherNet/IP port settings are transferred to the Controller.

12-2-2 Settings Required for the SNMP Agent

The following Built-in EtherNet/IP Port Settings are made from the Sysmac Studio to use the SNMP agent.

Tab page	Setting	Setting conditions	Reference
SNMP Settings	SNMP service	Required.	page 5-8
	Port No.	Specified by user. Note Required to change from the default value of 161.	
	Contact, location	Specified by user.	
	Send a recognition trap	Specified by user. Select this check box to send a recognition trap if there is access from an SNMP manager that is not specified (Access other than Recognition 1 and 2).	
	Recognition 1 and Recognition 2	Specified by user. Make these settings to permit access by only certain SNMP managers.	page 5-9
	IP address		
	Host name		
	Community name		

Tab page	Setting	Setting conditions	Reference
SNMP Trap Settings	SNMP trap	Required.	page 5-10
	Port No.	Specified by user. Note Required to change from the default value of 162.	
	Trap 1 and trap 2		page 5-11
	IP address	Required.	
	Host name	Set an IP address or a host name as the SNMP trap destination.	
	Community name	Specified by user.	
	Version	Required. Set the version of the SNMP manager.	



Additional Information

Make the settings in the SNMP Settings Dialog Box and SNMP Trap Dialog Box if the SNMP agent is used.

Refer to *5-4 SNMP Settings Display* for information on the SNMP Settings Dialog Box. Refer to *5-5 SNMP Trap Settings Display* for information on the SNMP Trap Dialog Box.

13

Communications Performance and Communications Load

13-1 Communications System	13-2
13-1-1 Tag Data Link Communications Method	13-2
13-1-2 Calculating the Number of Connections	13-4
13-1-3 Packet Interval (RPI) Accuracy	13-5
13-2 Adjusting the Communications Load	13-6
13-2-1 Checking Bandwidth Usage for Tag Data Links	13-7
13-2-2 Tag Data Link Bandwidth Usage and RPI	13-8
13-2-3 Adjusting Device Bandwidth Usage	13-9
13-2-4 Changing the RPI	13-10
13-2-5 RPI Setting Examples	13-16
13-3 I/O Response Time in Tag Data Links	13-21
13-3-1 Timing of Data Transmissions	13-21
13-3-2 Built-in EtherNet/IP Port Data Processing Time	13-22
13-3-3 Relationship between Task Periods and Packet Intervals (RPIs)	13-24
13-3-4 Maximum Tag Data Link I/O Response Time	13-25
13-4 Message Service Transmission Delay	13-27

13-1 Communications System

13-1-1 Tag Data Link Communications Method

Requested Packet Interval (RPI) Settings

In tag data links for the built-in EtherNet/IP port, the data transmission period is set for each connection as the RPI.

The target device will send data (i.e., output tags) once each RPI, regardless of the number of nodes.

Also, the heartbeat frame is sent from the originator to the target for each connection. The target uses the heartbeat to check to see if errors have occurred in the connection with the originator. The data transmission period of the heartbeat frame depends on the RPI settings.

Heartbeat Frame Transmission Period

- If packet interval < 100 ms, the heartbeat frame transmission period is 100 ms.
- If packet interval \geq 100 ms, the heartbeat frame transmission period is the same as the RPI.

Example:

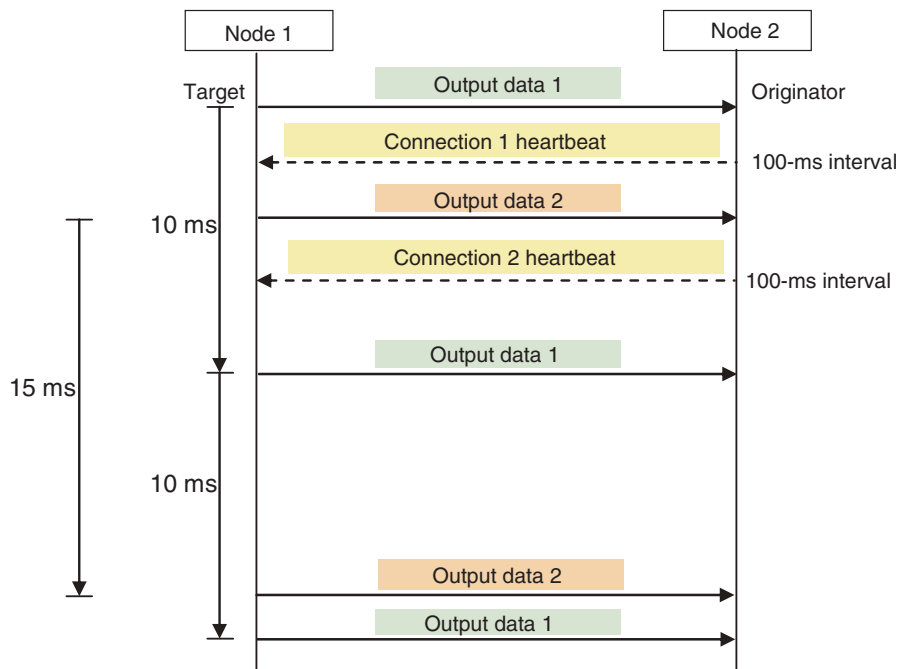
In this example, 2 tag data link connections are set for node 2 (the originator) and node 1 (the target).

The RPI for output data 1 is set to 10 ms.

The RPI for output data 2 is set to 15 ms.

In this case, output data 1 is sent from node 1 to node 2 every 10 ms, and output data 2 is sent from node 1 to node 2 every 15 ms, as shown in the following diagram.

Also, data is sent from node 2 (the originator) to node 1 (the target) with a heartbeat of 100 ms for connection 1 and a heartbeat of 100 ms for connection 2.



Requested Packet Interval (RPI) and Bandwidth Usage (PPS)

The number of packets transferred each second is called the used bandwidth or PPS (packets per second).

The PPS is calculated from the RPI and heartbeat as follows for each connection:

PPS used in a connection (pps) = $(1,000 \div \text{RPI (ms)}) + (1,000 \div \text{Heartbeat transmission period (ms)})$

Use the following equation to calculate the total number of packets transferred by each built-in Ethernet/IP port (Unit) in 1 second.

Built-in Ethernet/IP port's total PPS = Total PPS of originator connections + Total PPS of target connections*

* Connections set as target connections must be added, too.

The maximum number of packets that the built-in Ethernet/IP port for each series can transfer in 1 second (called the allowed Unit bandwidth) is as follows, so set the connection below this maximum value.

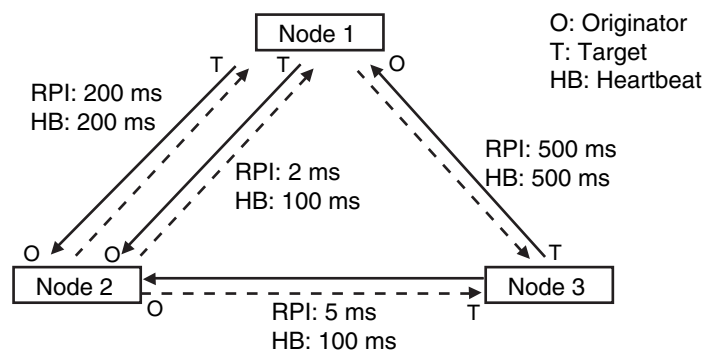
- NY-series Controller: 20,000 pps

Example:

Node 1 has both originator and target connections, with send RPI of 200 ms and 2 ms, and receive RPI of 500 ms.

Node 2 has originator connections only, with receive RPIs of 200 ms, 2 ms, and 5 ms.

Node 3 has target connections only, with send RPIs of 5 ms and 1 ms.



Each node's total PPS is calculated as follows:

- Total PPS of node 1's Unit
 $= 1000 / 200 \text{ ms} + 1000 / 2 \text{ ms} + 1000 / 500 \text{ ms}$ (for data)
 $+ 1000 / 200 \text{ ms} + 1000 / 100 \text{ ms} + 1000 / 500 \text{ ms}$ (for heartbeat)
 $= 524 \text{ pps}$
- Total PPS of node 2's Unit
 $= 1000 / 200 \text{ ms} + 1000 / 2 \text{ ms} + 1000 / 5 \text{ ms}$ (for data)
 $+ 1000 / 200 \text{ ms} + 1000 / 100 \text{ ms} + 1000 / 100 \text{ ms}$ (for heartbeat)
 $= 730 \text{ pps}$
- Total PPS of node 3's Unit
 $= 1000 / 5 \text{ ms} + 1000 / 500 \text{ ms}$ (for data)
 $+ 1000 / 100 \text{ ms} + 1000 / 500 \text{ ms}$ (for heartbeat)
 $= 214 \text{ pps}$

All of the Units are within the allowed Unit bandwidth, so they can transfer data.

13-1-2 Calculating the Number of Connections

The maximum number of connections for the built-in EtherNet/IP port for each series is as follows:

- NY-series Controller: 128

The number of connections must be set to the above numbers or less combining both connections that the Unit opens as the originator and connections that are opened from an originator with the Unit as the target.

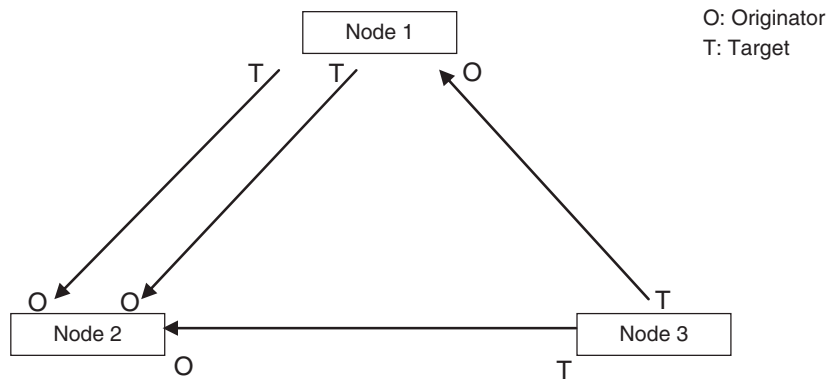
Example:

Node 1 opens two connections as the target with node 2 and one connection as the originator with node 3. Therefore, the total is three connections.

Node 2 opens two connections as the originator with node 1 and one connection as the originator with node 2. Therefore, the total is two connections.

Node 3 opens one connection as the target with node 1 and one connection as the target with node 2. Therefore, the total is two connections.

In either case, the connections can be opened because the numbers of connections for all nodes are within the maximum number of connections for the built-in EtherNet/IP port.

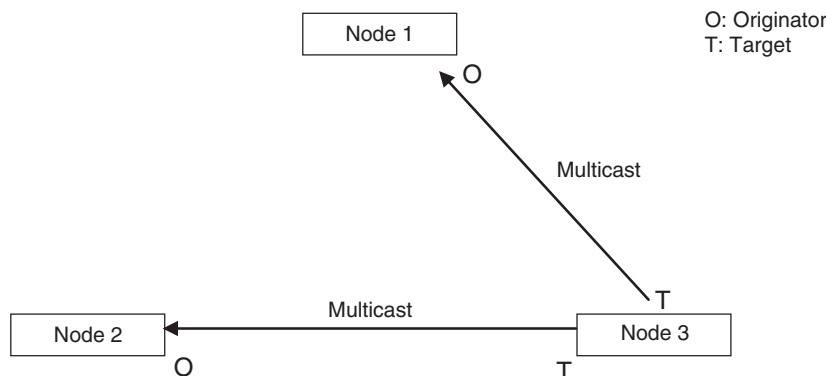


Also, if multicast is set, one packet will be sent, but the number of connections will be consumed.

Example:

Node 3 sends one multicast packet to node 1 and node 2. Node 3 opens one connection as the target with node 1 and one connection as the target with node 2.

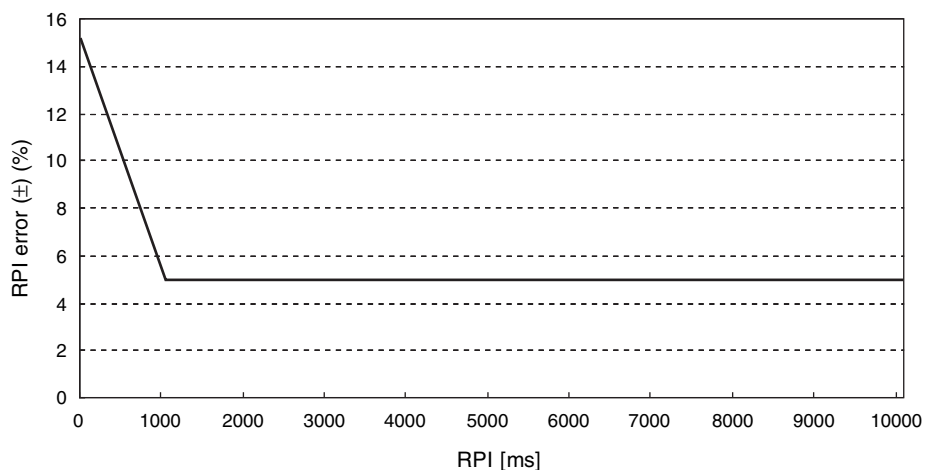
Caution is required because the number of connections consumed is the same as for unicast connections even when multicast connections are set.



13-1-3 Packet Interval (RPI) Accuracy

The send processing delay occurs in the built-in EtherNet/IP port when data packets are sent once each packet interval (RPI). This delay varies with the RPI error shown in the following graph, so the send processing delay time is the maximum value for each RPI.

Packet interval (RPI)	RPI error (\pm) (%)
1 ms to 1,000 ms	$15 - (\text{RPI [ms]}/100)$
1,000 to 10,000 ms	5% of the RPI



13-2 Adjusting the Communications Load

In an Ethernet network using an Ethernet switch, the network bandwidth is not shared by all of the nodes; independent transmission paths are established between individual nodes through the Ethernet switch.

A dedicated communications buffer is established in the Ethernet switch for communications between the nodes and full-duplex communications (simultaneous transmission and reception) are performed asynchronously with other transmission paths. The communications load in other transmission paths does not affect communications, so packet collisions do not occur and stable, high-speed communications can be performed.

The Ethernet switch functions shown in the following table determine the performance of tag data links.

Item	Description
Buffer capacity	This is the amount of data that can be buffered when packets accumulate at the Ethernet switch.
Multicast filtering	This function transfers multicast packets to specific nodes only.
QoS function	This function performs priority control on packet transfers.

The following table shows the setting ranges of the tag data link settings that can be made for a built-in EtherNet/IP port.

Item	Description	NY-series Controller
Network bandwidth	Physical Ethernet baud rate	1,000 Mbps
Allowed tag data link communications bandwidth	Maximum number of tag data link packets that can be processed in 1 second (pps: packets per second)	20,000 pps max.
Connection resources	Number of connections that can be established	128 max.
Packet interval (RPI: Requested Packet Interval)	Refresh period for tag data	1 to 10,000 ms in 1-ms increments

When the tag data link settings exceed the capabilities of the Ethernet switch being used, increase the packet interval (RPI) value. Particularly when using an Ethernet switch that does not support multicast filtering, the settings must be made considering that multicast packets will be sent even to nodes without connection settings.



Additional Information

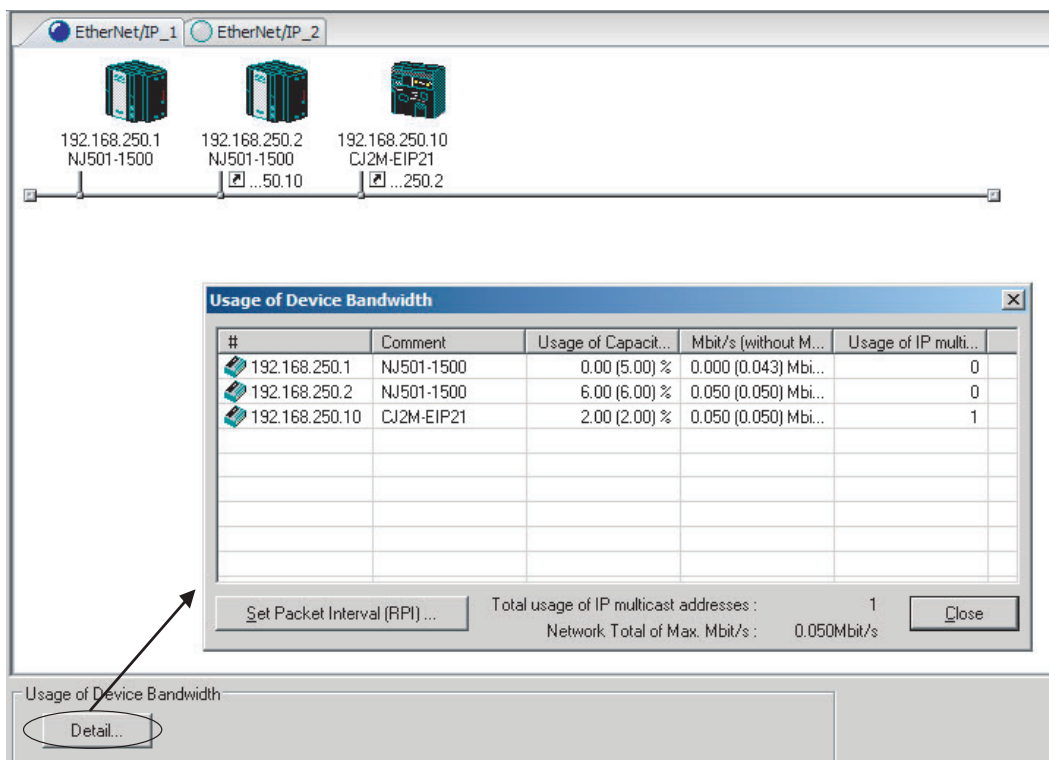
If the Network Configurator is used to set the connection type in the connection settings to a multicast connection, multicast packets will be used. If the connection type is set to a point-to-point connection, multicast packets are not used.

In addition, if the required tag data link performance cannot be achieved with the Ethernet switch's capabilities, re-evaluate the overall network configuration and correct it by taking steps such as selecting a different Ethernet switch or splitting the network.

The following sections show how to check the device bandwidth being used by the tag data links in the designed network, and how to set the appropriate values.

13-2-1 Checking Bandwidth Usage for Tag Data Links

The Network Configurator can display the bandwidth actually used for tag data links at each built-in EtherNet/IP port, based on the connections set in the network configuration. The device bandwidth used by tag data links can be checked by clicking the **Detail** Button in the Usage of Device Bandwidth Area at the bottom of the Network Configuration Window.



Item	Description
#	The IP address of the device.
Comment	A description of the device. The comment is displayed below the device icon. The model number of the device is displayed by default.
Usage of Capacity (without Multicast Filter)	The percentage of the allowable communications bandwidth used for tag data links for the device is displayed. Bandwidth used ÷ Allowable tag data link bandwidth The values outside parentheses are for when multicast filtering is used. The values inside parentheses are for when multicast filtering is not used.
Mbit/s (without Multicast Filter)	The bandwidth used for communications by the device of the network bandwidth is shown. The values outside parentheses are for when multicast filtering is used. The values inside parentheses are for when multicast filtering is not used.
Usage of IP Multicast Addresses	The number of multicast IP addresses actually used for communications by the device is shown.
Total usage of IP multicast addresses	The number of multicast IP addresses used in the entire network is shown. This value is used to estimate the number of multicast filters for switching.
Network Total of Max. Mbit/s	The total network bandwidth used for tag data link communications in the entire network is shown. Tag data links will not operate normally if the network bandwidth that can be set is exceeded.

● Checking the Usage of Capacity and Network Bandwidth for Tag Data Links

The percentage of the allowable communications bandwidth for tag data links for each built-in Ethernet/IP port is displayed as the *Usage of Capacity* and the bandwidth used for tag data link communications in the entire network is displayed as the *Mbit/s*. The usage of capacity and used network bandwidth that are displayed in parentheses are for an Ethernet switch that does not use multicast filtering. In this case, multicast packets will be sent to even the nodes without connection settings, so the displayed values will include these packets as well. These values can be adjusted according to instructions in *13-2-4 Changing the RPI*.

● Checking the Total Number of Multicast IP Addresses in the Network

When using an Ethernet switch that provides multicast filtering, there must be enough multicast filters for the network being used. The number of multicast IP address used in the entire network that is displayed by the Network Configurator is based on connection settings. Make sure that the number of multicast IP addresses used in the entire network does not exceed the number of multicast filters supported by the Ethernet switch. If necessary, change to an Ethernet switch with enough multicast filters, or adjust the usage of capacity and network bandwidth for tag data links (*Mbit/s*) values given for an Ethernet switch without multicast filtering (i.e., the values in parentheses). These values can be adjusted according to instructions in *13-2-4 Changing the RPI*.

● Checking the Total Maximum Network Bandwidth

The Network Configurator displays the total maximum bandwidth that can be used for the entire network. This value indicates the maximum bandwidth that can be used on the transmission paths when Ethernet switches are cascaded. If the value exceeds the bandwidth of a cascade connection in the actual network (e.g., 1,000 Mbps), the maximum bandwidth for part of the communications path may be exceeded, depending on how the network is wired. This may prevent the tag data links from operating correctly. If this occurs, either calculate the bandwidth usage for each communications path and be sure that the maximum bandwidth is not exceeded for any cascade connection, or adjust the bandwidth for all cascade connections so that the total maximum network bandwidth is not exceeded. Adjust the bandwidth according to instructions in *13-2-4 Changing the RPI*.

13-2-2 Tag Data Link Bandwidth Usage and RPI

The usage of capacity without multicast filtering can be adjusted against the tag data link's allowable bandwidth by using the packet interval (RPI) setting. If the RPI is made shorter, the usage of capacity will increase. If the RPI is made longer, the usage of capacity will decrease.

The RPI can be set in any one of the following ways.

- Setting the same interval for all connections
- Setting a particular device's connection
- Setting a particular connection

When the same RPI is set for all connections, the usage of capacity will basically increase proportionally as the RPI is made shorter.

Example: If the RPI is set to 50 ms for all connections and the usage of capacity is 40%, the usage of capacity may increase to 80% when the RPI is reduced to 25 ms for all connections.



Precautions for Correct Use

Performing message communications or other network operations from the Network Configurator (such as monitoring or other operations that place a load on the network) or from the user application when the tag data link bandwidth usage of capacity is between 80% and 100% can temporarily create an excessive load on the network and result in timeouts. If timeouts occur, increase one or all of the RPI settings and reduce the usage of capacity.

13-2-3 Adjusting Device Bandwidth Usage

This paragraph provides the method to adjust the device bandwidth usage for tag data links.



Precautions for Correct Use

Ethernet switch must be corresponding to the maximum values of the network bandwidth, which can be set to Controller. The maximum value of the network bandwidth for NY-series Controller is 1,000 Mbit/s.

● Ethernet Switches without Multicast Filtering

- Is the network bandwidth usage without multicast filtering under the network bandwidth that can be set for each node?
If any node exceeds its maximum network bandwidth that can be set, change the connections settings, such as the RPI.
- Is the usage of capacity without multicast filtering under 100% for each node?
If any node exceeds 100%, change the connections settings, such as the RPI.
- Is the total network bandwidth usage under the network bandwidth that can be set?
If the total bandwidth usage exceeds the network bandwidth that can be set, the bandwidth of part of the transmission path (e.g., an Ethernet switch or media converter) may be exceeded as the result of how the network was wired (e.g., cascade connections of Ethernet switches), causing a tag data link to operate abnormally. Check the bandwidth of the transmission path for all cascade connections. If the bandwidth is exceeded, rewire the network or increase the bandwidth between Ethernet switches (e.g., to 1 Gbps). If these countermeasures are not possible, change the connection settings, e.g., the RPI settings, and adjust the bandwidth for all cascade connections until the total network bandwidth is not exceeded.

● Ethernet Switches with Multicast Filtering

- Is the network bandwidth usage under the network bandwidth that can be set for each node?
If any node exceeds its maximum network bandwidth that can be set, change the connections settings, such as the RPI.
- Is the usage of capacity under 100% for each node?
If any node exceeds 100%, change the connections settings, such as the RPI.
- Is the total network bandwidth usage under the network bandwidth that can be set?
If the total bandwidth usage exceeds the network bandwidth that can be set, the bandwidth of part of the transmission path (e.g., an Ethernet switch or media converter) may be exceeded as the result of how the network was wired (e.g., cascade connections of Ethernet switches), causing a tag data link to operate abnormally. Check the bandwidth of the transmission path for all cascade connections. If the bandwidth is exceeded, rewire the network or increase the bandwidth between Ethernet switches (e.g., to 1 Gbps). If these countermeasures are not possible, change the connection settings, e.g., the RPI settings, and adjust the bandwidth for all cascade connections until the total network bandwidth is not exceeded.
- Is the network bandwidth usage without multicast filtering under the network bandwidth that can be set for each node or is the usage of capacity without multicast filtering under 100% for each node?
If any node exceeds the maximum network bandwidth that can be set or the usage of 100%, check whether the multicast filtering on the Ethernet switch is functioning correctly. If the number of multicast filters on the Ethernet switch is less than the total usage of IP multicast addresses, bandwidth overloads may occur in some paths and prevent tag data links from operating correctly depending on the network connection (e.g., cascade connections of Ethernet switches). Calculate the number of multicast filters required by each Ethernet switch on the network and make sure that the number does not exceed the number of Ethernet switch multicast filters. If the number of Ethernet switch multicast filters is not sufficient, use switches with enough multicast filters or revise connection settings, such as the RPI settings.

- 3** The usage of capacity without multicast filtering can be adjusted against the tag data link's allowable bandwidth by changing the associated devices' packet interval (RPI) settings.

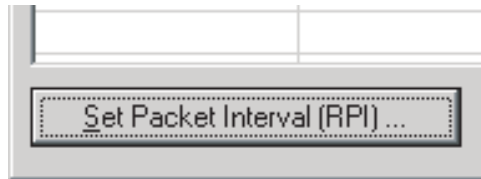
The RPI settings can be changed with the following three methods.

Method 1:

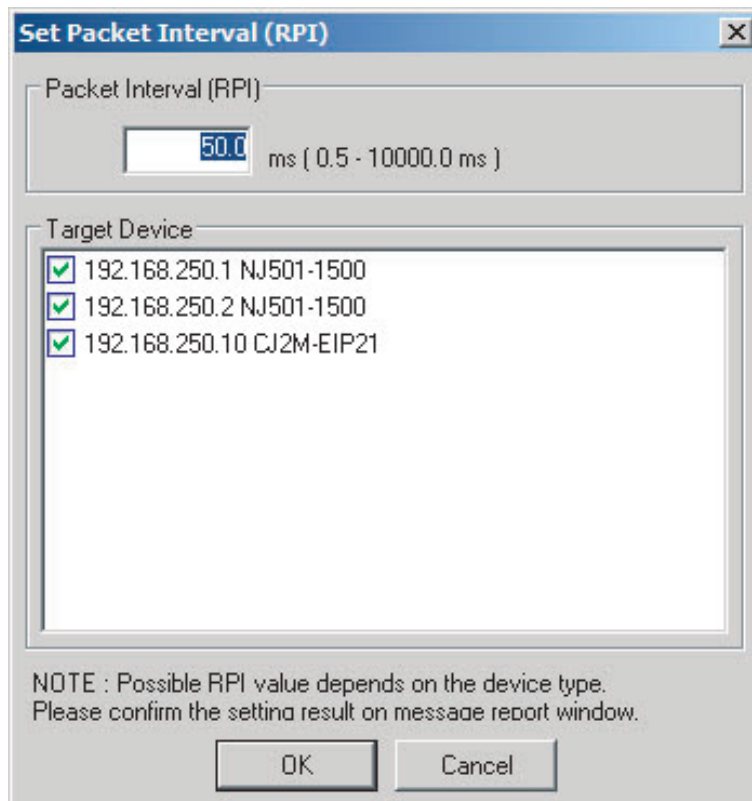
Change All Connections to the Same RPI

The usage of capacity without a multicast filter can be adjusted for all devices by changing the packet intervals (RPI) settings for all of the device's connections to the same RPI at the same time.

- (1) **Click the Set Packet Interval (RPI) Button at the bottom of the Usage of Device Bandwidth Dialog Box.**



- (2) **The Set Packet Interval (RPI) Dialog Box will be displayed. Input a new RPI value, and click the OK Button.**

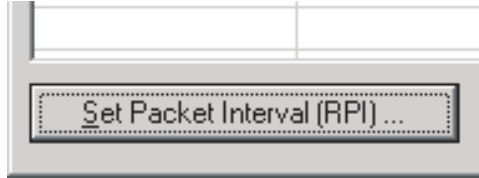


Method 2:

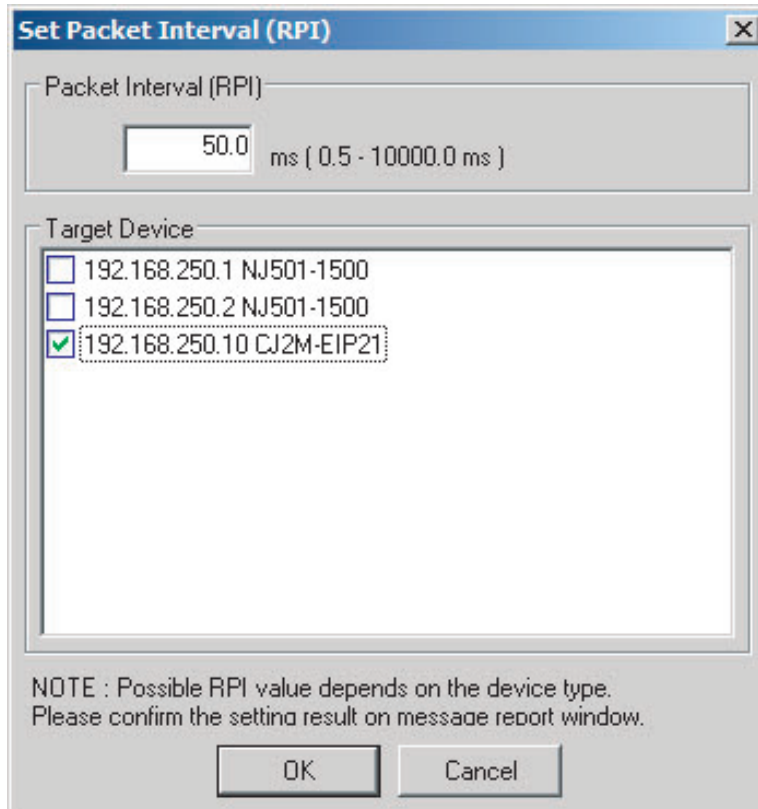
Change a Particular Device's Packet Interval (RPI) Setting:

The usage of capacity without multicast filtering can be adjusted for only a particular device against the tag data link's allowable bandwidth by changing the packet intervals (RPI) settings for all of the device's connections together. In this case, the usage of capacity will also change for the target devices of the connection for which the packet interval is changed.

- (1) Click the **Set Packet Interval (RPI) Button** at the bottom of the **Usage of Device Bandwidth Dialog Box**.



- (2) The **Set Packet Interval (RPI) Dialog Box** will be displayed. In the *Target Device Area*, **deselect the target devices that are not being adjusted by removing the check marks.**



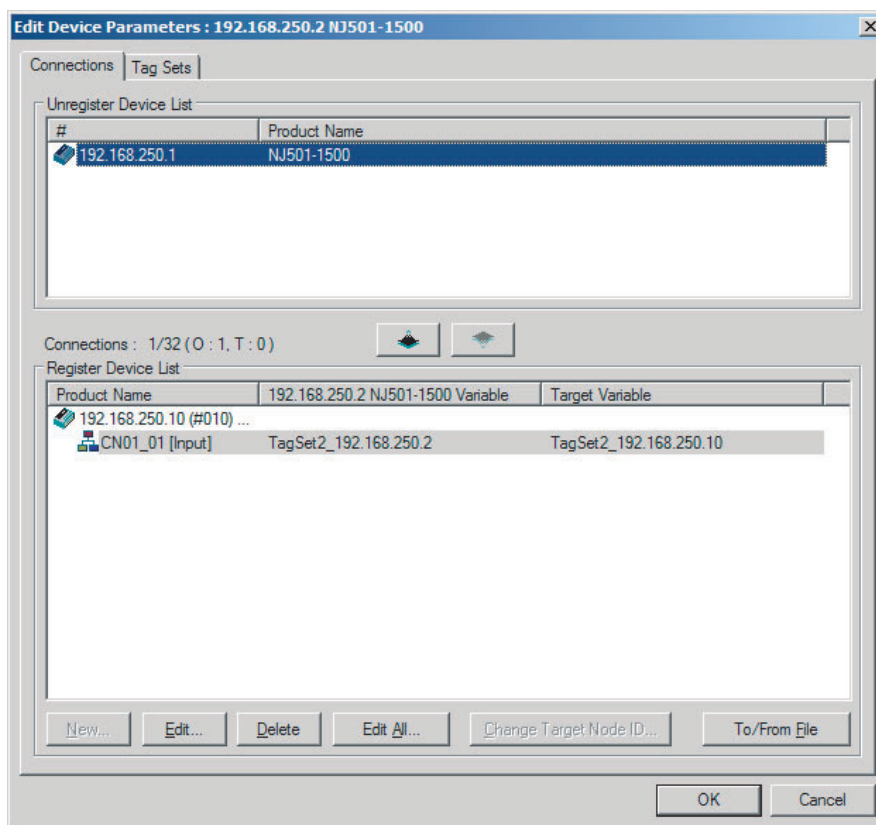
- (3) **Input a new RPI value, and click the OK Button.**

Method 3:

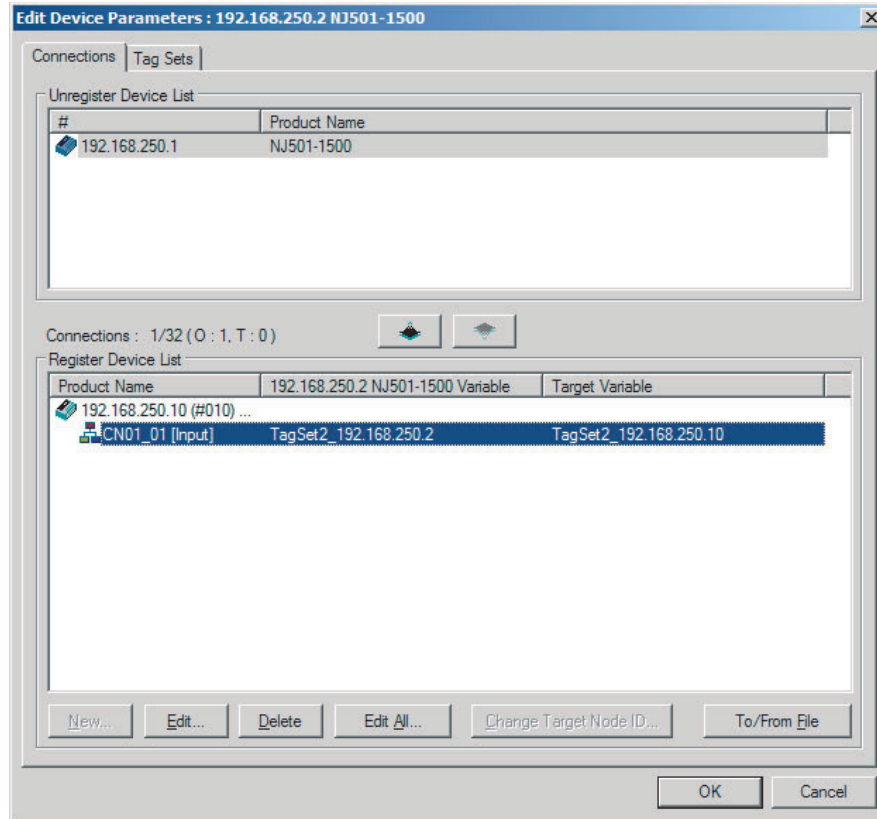
Changing a Particular Connection's Packet Interval (RPI) Setting:

The usage of capacity without multicast filtering can be adjusted against the tag data link's allowable bandwidth by individually changing the packet interval (RPI) for a particular connection. In this case, the usage of capacity will also change for target device of the connection for which the packet interval is changed.

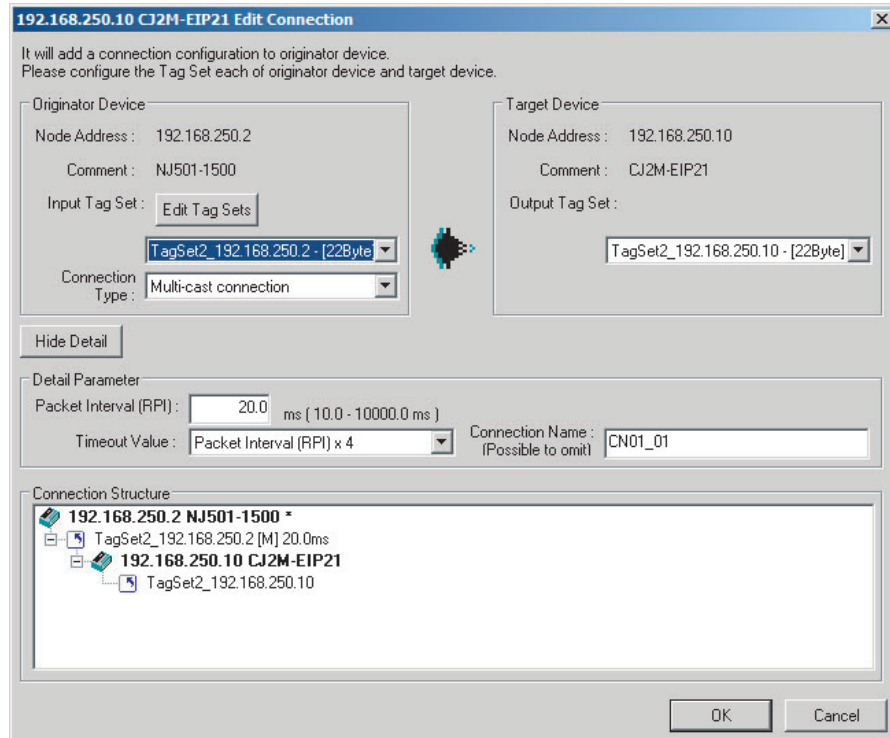
- (1) Click the Close Button at the bottom of the Usage of Device Bandwidth Dialog Box.
- (2) Double-click the device that is set as the originator of the desired connection. The Edit Device Parameters Dialog Box will be displayed.



- (3) In the Register Device List, select the connection for which you want to change the RPI, and click the Edit Button.



- (4) The device's Edit Connection Dialog Box will be displayed. Input a new packet interval (RPI) value, and click the OK Button.



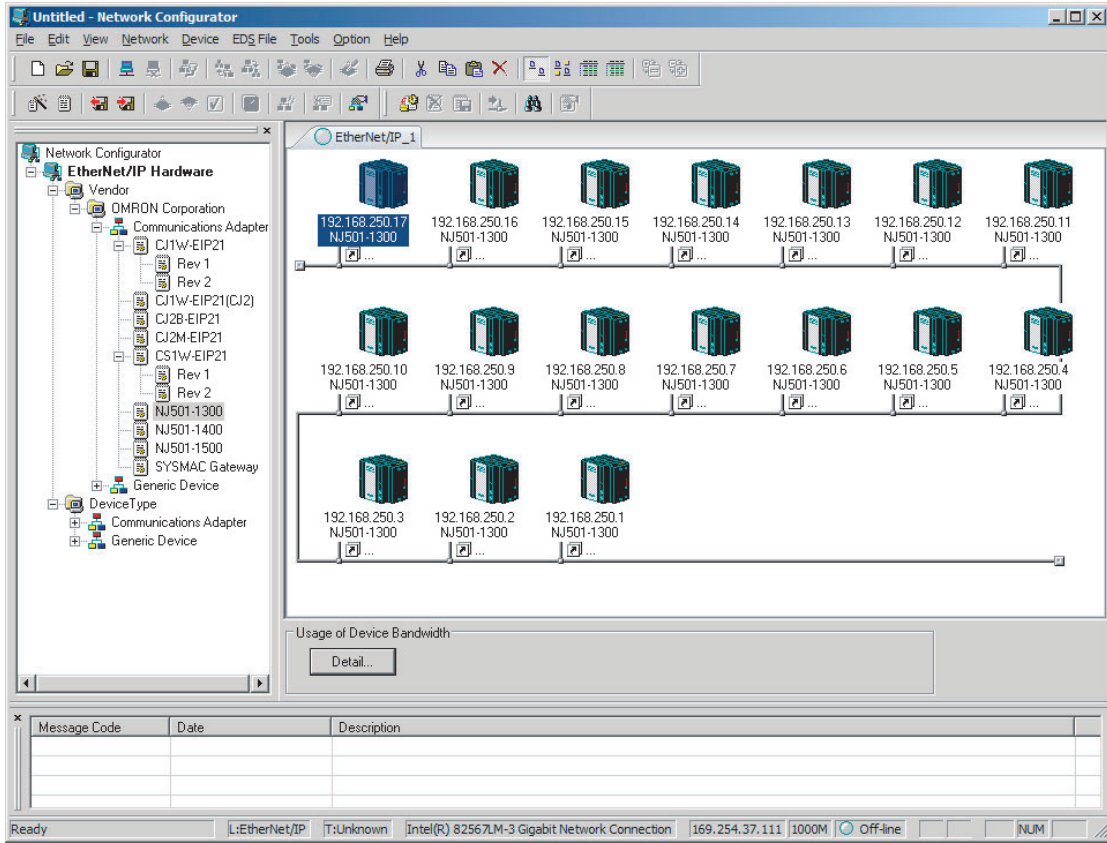
- 4** If the usage of capacity cannot be adjusted to the desired level when the setting described above has been performed, reconsider the network configuration considering the following points. Refer to *13-2-3 Adjusting Device Bandwidth Usage*.
 - Reduce the number of nodes and number of connections.
 - Split the network.
- 5** Check the bandwidth usage again.

If you have changed the connection settings, click the **Detail** Button in the Usage of Device Bandwidth Area at the bottom of the Network Configuration Window and check bandwidth usage according to the instructions in *13-2-1 Checking Bandwidth Usage for Tag Data Links*. It is particularly important to check the usage of capacity when an individual connection's RPI setting was changed without using the **Set Packet Interval (RPI)** Button.
- 6** Run user tests to verify that there are no problems with the new settings.

13-2-5 RPI Setting Examples

The following examples explain how to calculate the packet intervals (RPIs) in the following network configuration.

Conditions

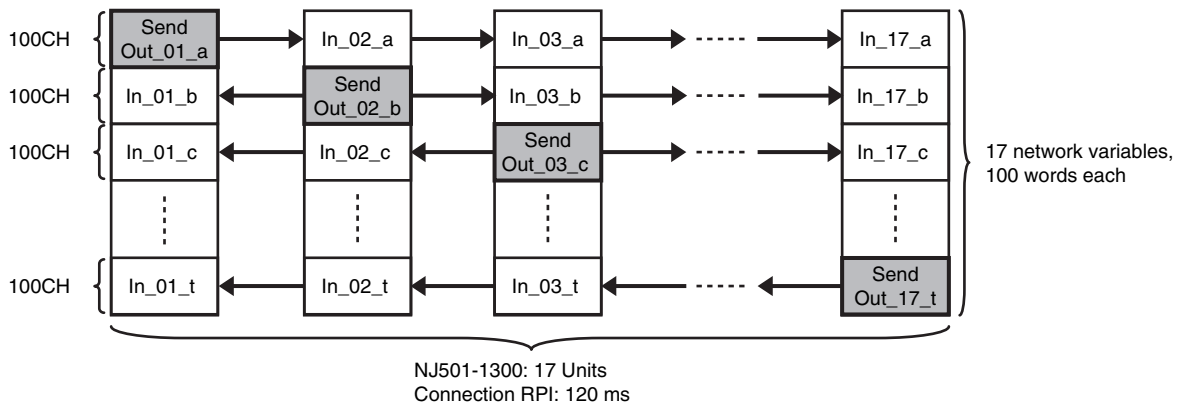


- Connections:

Example: Seventeen NJ501-1300 Units are connected to the network.

Each device has one 100-word tag for transmission and sixteen 100-word tags for reception, so that the Units exchange data mutually. By default, the packet intervals (RPIs) are set to 120 ms for all of the connections. The devices' IP addresses range from 192.168.250.1 to 192.168.250.17.

IP address: 192.168.250.1 192.168.250.2 192.168.250.3 192.168.250.17
 NJ501-1300 NJ501-1300 NJ501-1300 NJ501-1300



Checking the Device Bandwidth Usage

When the **Detail** Button is clicked in the Usage of Device Bandwidth Area, it is apparent that the percentage of the allowed tag data link bandwidth being used by each device's tag data link (Usage of Capacity) is 40.83%, as shown in the following dialog box.

#	Comment	Usage of Capacit...	Mbit/s (without Multic...	Usage of IP multi...
192.168.250.17	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.16	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.15	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.14	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.13	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.12	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.11	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.10	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.9	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.8	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.7	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.6	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.5	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.4	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.3	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.2	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1
192.168.250.1	NJ501-1300	40.83 (40.83) %	0.510 (0.510) Mbit/s	1

Set Packet Interval (RPI) ... Total usage of IP multicast addresses : 17 Network Total of Max. Mbit/s : 1.886Mbit/s Close

Changing Settings

Method 1: Same Packet Interval Setting for All Connections

The percentage of the allowed tag data link bandwidth being used (Usage of Capacity) was 40.83% with the RPI set to 120 ms for all of the connections, so the RPI will be set to 40 ms, with a target of 80% or less of the allowable bandwidth.

Click the **Set Packet Interval (RPI)** Button at the bottom of the Usage of Device Bandwidth Dialog Box to display the Set Packet Interval (RPI) Dialog Box. Input 40 ms as the new RPI value, then click the **OK** Button.

Set Packet Interval (RPI)

Packet Interval (RPI): 40 ms (0.5 - 10000.0 ms)

Target Device

<input checked="" type="checkbox"/>	192.168.250.17 NJ501-1300	<input checked="" type="checkbox"/>	192.168.250.8 NJ501-1300
<input checked="" type="checkbox"/>	192.168.250.16 NJ501-1300	<input checked="" type="checkbox"/>	192.168.250.7 NJ501-1300
<input checked="" type="checkbox"/>	192.168.250.15 NJ501-1300	<input checked="" type="checkbox"/>	192.168.250.6 NJ501-1300
<input checked="" type="checkbox"/>	192.168.250.14 NJ501-1300	<input checked="" type="checkbox"/>	192.168.250.5 NJ501-1300
<input checked="" type="checkbox"/>	192.168.250.13 NJ501-1300	<input checked="" type="checkbox"/>	192.168.250.4 NJ501-1300
<input checked="" type="checkbox"/>	192.168.250.12 NJ501-1300	<input checked="" type="checkbox"/>	192.168.250.3 NJ501-1300
<input checked="" type="checkbox"/>	192.168.250.11 NJ501-1300	<input checked="" type="checkbox"/>	192.168.250.2 NJ501-1300
<input checked="" type="checkbox"/>	192.168.250.10 NJ501-1300	<input checked="" type="checkbox"/>	192.168.250.1 NJ501-1300
<input checked="" type="checkbox"/>	192.168.250.9 NJ501-1300		

NOTE : Possible RPI value depends on the device type. Please confirm the setting result on message report window.

OK Cancel

If the packet interval for all connections has been set to the same setting, the dialog box will show that the usage of capacity for the tag data link's allowable communications bandwidth is 74.50% and the fastest set value is 40 ms.

#	Comment	Usage of Capacit...	Mbit/s (without Multic...	Usage of IP multi...
192.168.250.17	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.16	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.15	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.14	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.13	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.12	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.11	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.10	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.9	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.8	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.7	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.6	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.5	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.4	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.3	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.2	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1
192.168.250.1	NJ501-1300	74.50 (74.50) %	1.199 (1.199) Mbit/s	1

Set Packet Interval (RPI) ... Total usage of IP multicast addresses : 17
 Network Total of Max. Mbit/s : 2.851Mbit/s

Method 2: Changing the Packet Interval (RPI) of Only Specific Devices

In this example, we want faster tag data links for devices 192.168.250.1 and 192.168.250.10 only. To do this, click the **Set Packet Interval (RPI)** Button at the bottom of the Usage of Device Bandwidth Dialog Box. The Set Packet Interval (RPI) Dialog Box is displayed.

In the Target Device Area, clear the selections of all devices other than 192.168.250.1 and 192.168.250.10. Input 30 ms as the new RPI value, then click the **OK** Button.

Set Packet Interval (RPI)

Packet Interval (RPI)
 ms (0.5 - 10000.0 ms)

Target Device

- 192.168.250.17 NJ501-1300
- 192.168.250.16 NJ501-1300
- 192.168.250.15 NJ501-1300
- 192.168.250.14 NJ501-1300
- 192.168.250.13 NJ501-1300
- 192.168.250.12 NJ501-1300
- 192.168.250.11 NJ501-1300
- 192.168.250.10 NJ501-1300
- 192.168.250.9 NJ501-1300
- 192.168.250.8 NJ501-1300
- 192.168.250.7 NJ501-1300
- 192.168.250.6 NJ501-1300
- 192.168.250.5 NJ501-1300
- 192.168.250.4 NJ501-1300
- 192.168.250.3 NJ501-1300
- 192.168.250.2 NJ501-1300
- 192.168.250.1 NJ501-1300

NOTE : Possible RPI value depends on the device type.
 Please confirm the setting result on message report window.

OK Cancel

The percentage of the allowed tag data link bandwidth being used (Usage of Capacity) increases to 87.00% for devices 192.168.250.1 and 192.168.250.10, which indicates that the RPI is set to a higher speed for these devices' connections.

The Usage of Capacity values also indicate that the Usage of Capacity has increased (from 40.83% to 44.67%) for all of the other devices, which connect with devices 192.168.250.1 and 192.168.250.10.

#	Comment	Usage of Capacit...	Mbit/s (without Multic...	Usage of IP multi...
192.168.250.17	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.16	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.15	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.14	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.13	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.12	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.11	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.10	NJ501-1300	87.00 (100.33) %	1.528 (1.835) Mbit/s	2
192.168.250.9	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.8	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.7	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.6	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.5	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.4	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.3	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.2	NJ501-1300	44.50 (97.83) %	0.589 (1.818) Mbit/s	2
192.168.250.1	NJ501-1300	87.00 (100.33) %	1.528 (1.835) Mbit/s	2

Set Packet Interval (RPI) ... Total usage of IP multicast addresses : 34 Network Total of Max. Mbit/s : 3.228Mbit/s Close

In this case, if there is no multicast filter, the value becomes 100.33%. If there is no multicast filter for an Ethernet switch, communications errors may occur depending on the communications load of the built-in EtherNet/ IP Unit port.

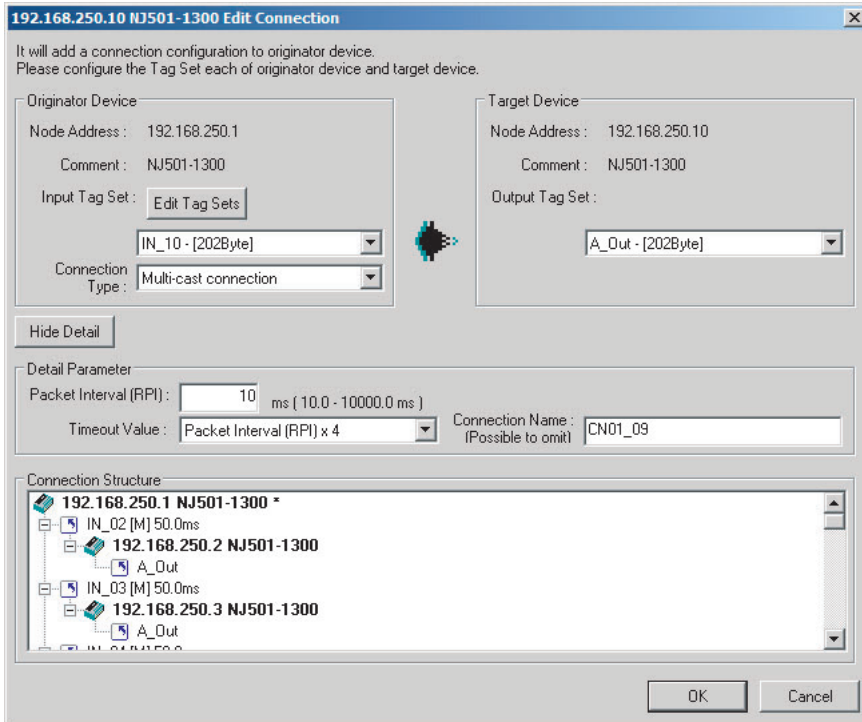
Method 3: Changing the Packet Intervals (RPIs) of Only Specific Connections

In this example, we want a faster tag data links for just a particular connection of device 192.168.250.1. Double-click device 192.168.250.1 in the Network Configuration Window.

Connections : 32/32 (O : 16, T : 16)

Product Name	192.168.250.1 NJ501-1300 Variable	Target Variable
192.168.250.2 (#002) NJ501-1300		
192.168.250.3 (#003) NJ501-1300	IN_02	A_Out
192.168.250.4 (#004) NJ501-1300	IN_03	A_Out
192.168.250.5 (#005) NJ501-1300	IN_04	A_Out
192.168.250.6 (#006) NJ501-1300	IN_05	A_Out
192.168.250.7 (#007) NJ501-1300	IN_06	A_Out

Information about the connection with device 192.168.250.10 is registered in the Register Device List. Double-click this connection to edit the settings.



In the Edit Connection Dialog Box, input 10 ms as the new RPI value, and click the **OK** Button. The tag data link bandwidth used by device 192.168.250.1 (Usage of Capacity) increases to 50.17%, which indicates that a RPI is set to a higher speed for this device.

#	Comment	Usage of Capacit...	Mbit/s (without M...	Usage of IP multi...
192.168.250.17	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.16	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.15	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.14	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.13	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.12	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.11	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.10	NJ501-1300	51.00 (51.00) %	0.741 (0.741) Mbi...	2
192.168.250.9	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.8	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.7	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.6	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.5	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.4	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.3	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.2	NJ501-1300	40.83 (50.83) %	0.510 (0.740) Mbi...	1
192.168.250.1	NJ501-1300	50.17 (51.00) %	0.722 (0.741) Mbi...	1

Set Packet Interval (RPI) ... Total usage of IP multicast addresses : 18 Network Total of Max. Mbit/s : 2.117Mbit/s Close

In this case, the tag data link bandwidth that is used by device 192.168.250.10 (Usage of Capacity) also increases (from 40.83% to 51.00%).

13-3 I/O Response Time in Tag Data Links

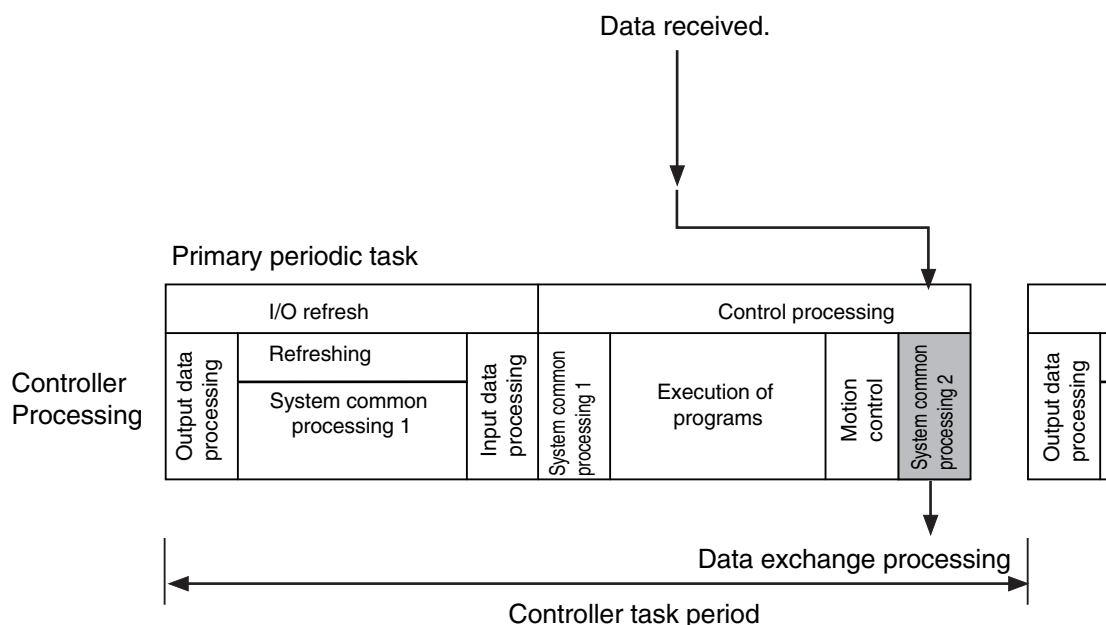


Additional Information

This section describes the response time of the NY-series Controller built-in Ethernet/IP port. The data processing times for the EtherNet/IP Unit, the built-in EtherNet/IP port on the NX- and NJ-series Controller, the built-in EtherNet/IP port on the CJ2H-CPU6□-EIP CPU Unit and the built-in EtherNet/IP port on the CJ2M-CPU3□ CPU Unit are different. For details, refer to 7-4 *Tag Data Links with Other Models*.

13-3-1 Timing of Data Transmissions

The following diagram shows the timing of tag data link transfers between the built-in EtherNet/IP port and the Controller. Data is transferred when system common processing 2 is performed for the task set as the refreshing task.



You can set either of the following types of tasks as the refreshing task.

- Primary periodic task
The primary periodic task has the highest execution priority. It executes processes with high speed and high precision.
- Periodic tasks
Periodic tasks are executed during the time between executions of the primary periodic task.

The task during which to perform tag data link processing is specified for each tag. Set the refreshing task on the Sysmac Studio for each variable you want to set as a tag. Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for details on setting refreshing tasks.

13-3-2 Built-in EtherNet/IP Port Data Processing Time

This section describes the data processing time required to transfer data between the built-in EtherNet/IP port and the Controller.

Data Processing Time Overview

The time required for data processing consists of the following three elements.

1. Variable Access Time

First, calculate the time required to transfer tag data (or the time required to access variables). This calculation is performed for each task. Therefore, if the same refreshing task is set for multiple tag sets, calculate the total time required for all tags in the tag sets.

Use the following equation to calculate the variable access time.

$$\text{Variable access time } [\mu\text{s}] = \text{Total size of variables } [\text{bytes}] \times a + \text{Number of variables} \times b + \text{Number of accesses} \times c + d$$

Number of accesses: Number of tag sets

a to d: Constant values as given below

Controller model	Constant value [μs]			
	a	b	c	d
NY5□2-□□□□	0.0006	0.100	1.40	6.6

2. Number of Data Transfers

Tag data is transferred as a part of task processing.

If the time required to process the data transfer is greater than the "variable access time" *2, the data cannot be sent entirely in one task period and is sent separately instead.

$$\text{Number of data transfers} = \frac{\text{"Time required to send the data entirely" } *1}{\text{"Variable access time" } *2 \text{ set for the task}}$$

*1 This is the variable access time as calculated in step 1 above.

*2 The "variable access time" refers to the maximum processing time for accessing variables. Set the time for each task on the Task Setup Display, which is displayed by selecting **Configurations and Setup - Task Setup** in the Sysmac Studio.



Precautions for Correct Use

The maximum number of tag data link words that can be transferred through the built-in EtherNet/IP port is 92,416 words. If the number of tag data link words exceeds the number of words that can be exchanged with the Controller at one time, the data is divided and transferred in multiple data exchanges.

3. Actual Time Required for Data Transfer

You can use the task period of the refreshing task and the number of data transfers as calculated in (2) above to calculate the actual time required to transfer the data.

$$\text{Task period} \times \text{Number of data transfers}$$

Data Processing Time Calculation Example

Here we provide an example of how to perform the tag data link calculations described earlier for the following tag data transfers.

- Model Numbers of Controllers for Tag Data Links
NY5□2-□□□□
- Refreshing Task
Primary periodic task
Task period: 500 μ s (variable access time: 2%)
- Setting Tag Sets

Tag set	Refreshing task	Number of variables	Total size of variables
Tag set A	Primary periodic task	8	600 bytes
Tag set B	Primary periodic task	4	200 bytes
Tag set C	Primary periodic task	10	1,000 bytes

- 1 Calculate the variable access time as shown below.

$$[(600 + 200 + 1,000) \text{ bytes} \times 0.0006 \mu\text{s}] + [(8 + 4 + 10) \text{ variables} \times 0.100 \mu\text{s}] + 3 \times 1.40 \mu\text{s} + 6.60 \mu\text{s} = 11.28 \mu\text{s}$$
- 2 Calculate the number of data transfers.
 Time required for data transfer: "Variable access time" in step 1 = 11.28 μ s
 Variable access time set for the task: 500 μ s \times 0.02 = 10 μ s
 Number of data transfers: 11.28 μ s \div 10 μ s = 1.128 times
 Thus, approximately two data transfers are required.
- 3 Calculate the actual time required for the data transfer.

$$500 \mu\text{s} \times 2 \text{ times} = 1,000 \mu\text{s}$$

13-3-3 Relationship between Task Periods and Packet Intervals (RPIs)

Effect of Tag Data Links on Task Periods

The tag data is transferred during task processing. Therefore, the tag data transfer process is added to the task processing for tasks set as a tag's refreshing task. This requires you to make adjustments to the variable access time and task period in the Task Setup so that these processes are completed within a single task period.

- 1 Calculate the time required for the data transfer and set the result as the "variable access time".

For the formula for calculating the time required for the data transfer, refer to *Data Processing Time Calculation Example*.

* If the same refreshing task is set for multiple tag sets, calculate the total time required for all tags in tag sets.

- 2 Set the variable access time in the Task Setup to a value equal to or greater than the value calculated in step 1 above.

Adjust the task period time after adding in the time calculated in step 1. Use the Sysmac Studio to set the variable access time and task period settings. For details, refer to the *NY-series Industrial Panel PC / Industrial Box PC Software User's Manual* (Cat. No. W558).

Adjusting Packet Intervals (RPIs) According to the Task Periods

Tag data is transferred based on the actual time required for the transfer (task period × number of data transfers), regardless of the packet interval (RPI) setting. Therefore, set the packet interval (RPI) according to the following guideline.

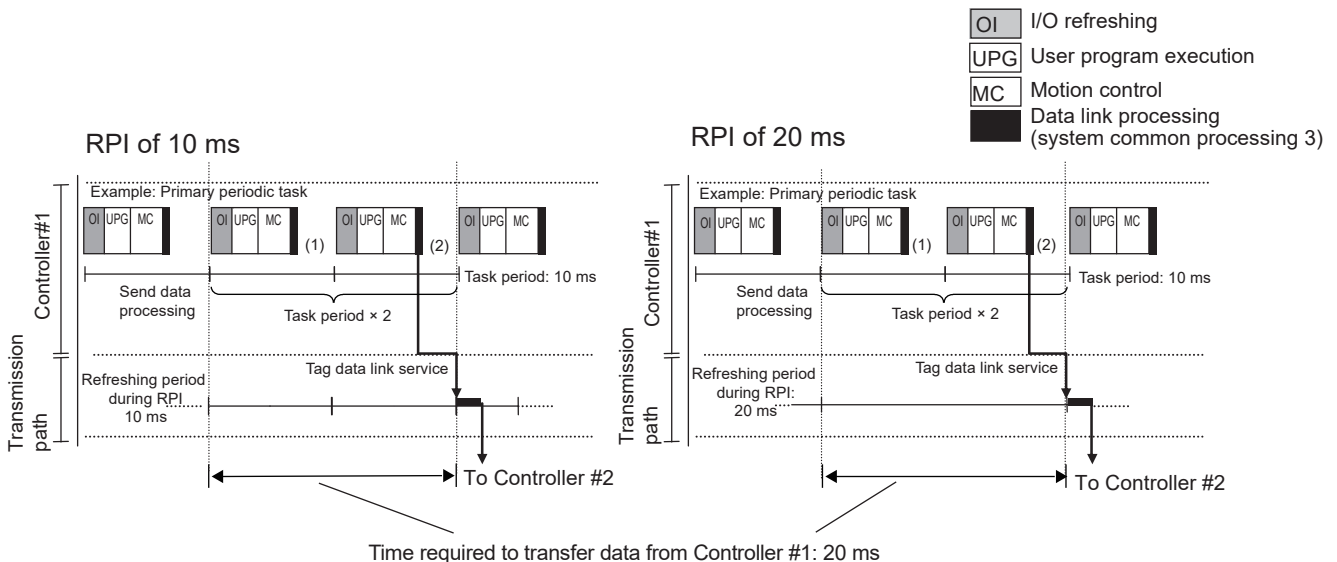
Actual time required for data transfer (Task period × Number of data transfers) < RPI

For details on the actual time required to transfer data, refer to *13-3-2 Built-in EtherNet/IP Port Data Processing Time*.

Example: Relationship between the RPI Setting and the Time Required for Data Transfer

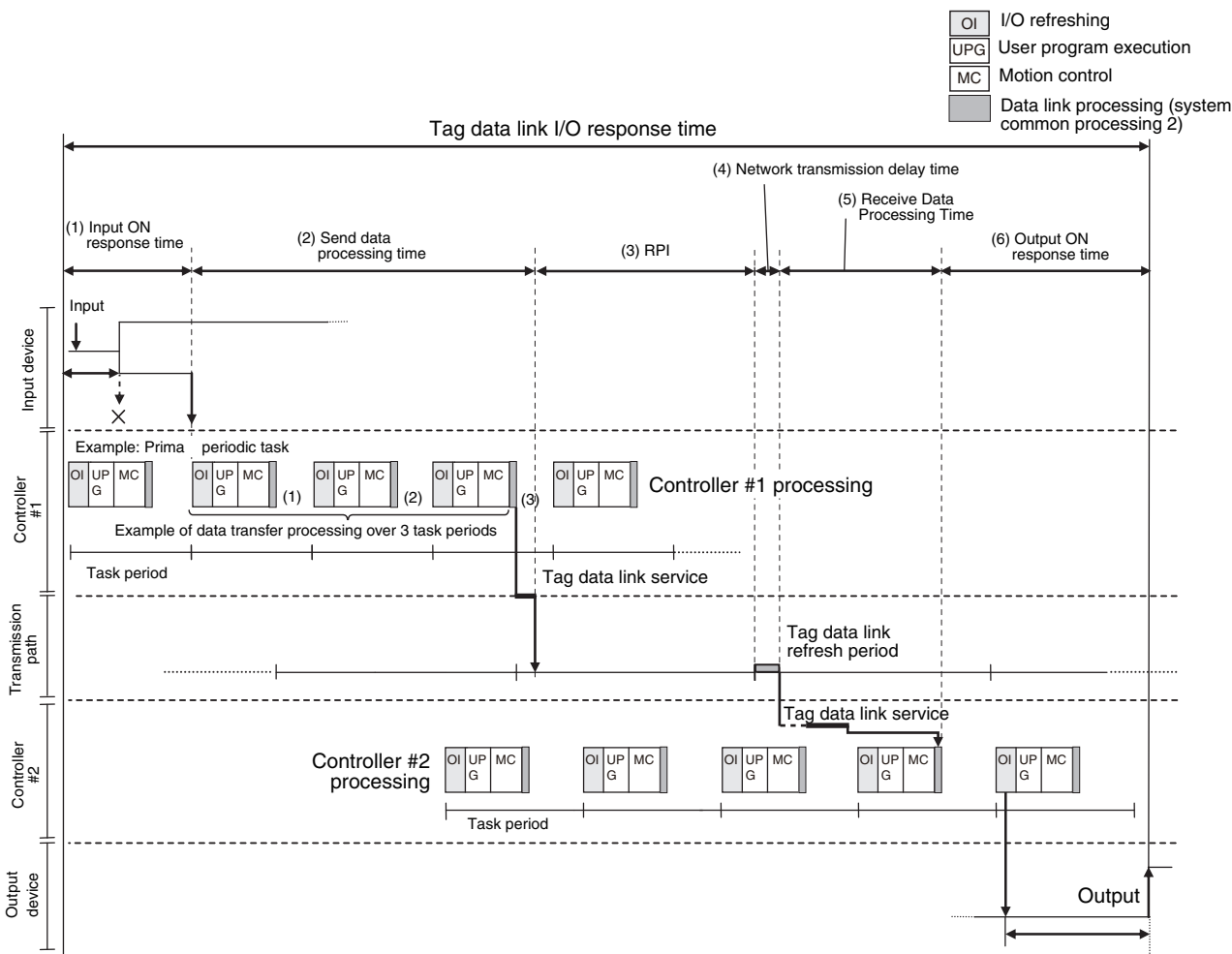
- Task period: 10 ms
- Number of data transfers: 2
- Actual time required for data transfer: 10 ms × 2 times = 20 ms

Regardless of the RPI value, the time required for the data transfer is 20 ms.



13-3-4 Maximum Tag Data Link I/O Response Time

You can find the maximum I/O response time from the total of (1) to (6) in the following figure.



(1) Input ON Response Time

This is the delay time for the external input device from when the input occurs until the switch actually changes to ON and the time until the input data is stored in the memory area of the Controller. Refer to the input delay of each device for the input switch delay time. Also, one task period is required until the data is stored in the memory area of the Controller. Therefore, the input ON response time is obtained as shown below.

$$\text{Input ON response time} = \text{Input device delay time} + \text{Controller task period}$$

(2) Send Data Processing Time

This is the time until the variables in the Controller are transferred to the built-in EtherNet/IP port. Data is transferred during task processing. Therefore, the time required for send data processing is the same as the task period. If the data that is transferred is larger than the amount of data that can be sent during a single task (as set in the variable access time for the task), the data is transferred over multiple task periods. Therefore, add (task period × the number of transfers) to the time required. For details on how to determine the time required to send data, refer to 13-3-2 Built-in EtherNet/IP Port Data Processing Time.

(3) Packet Interval (RPI)

This is the communications refresh period set for each connection using the Network Configurator.

(4) Network Transmission Delay Time

The transmission delay on an Ethernet line is 50 μ s or less. This delay time can be ignored.

(5) Receive Data Processing Time

This is the time required to transfer data received on the built-in EtherNet/IP port to a variable in the Controller. Data is received during task processing. Therefore, the time required for receive data processing is the same as the task period. If the data that is transferred is larger than the amount of data that can be received during a single task (as set in the variable access time for the task), the data is transferred over multiple task periods. Therefore, add (task period \times the number of transfers) to the time required. For details on how to determine the time required to receive data, refer to *13-3-2 Built-in EtherNet/IP Port Data Processing Time*. Data is transferred once in each task period. Therefore, if data transfer has ended in the task period in which data is received, the start of transmission for received data will be delayed by one Controller task period.



Additional Information

The total amount of data transferred increases if there are connections with multiple nodes, and the data that is transferred may exceed the amount that can be processed in a single transfer. In this case, the number of data transfers increases.

(6) Output ON Response Time

This is the delay time for the external output device from when the Controller specified turning ON the output until the output is actually turned ON.

Output ON response time = Output device delay time + Controller task period

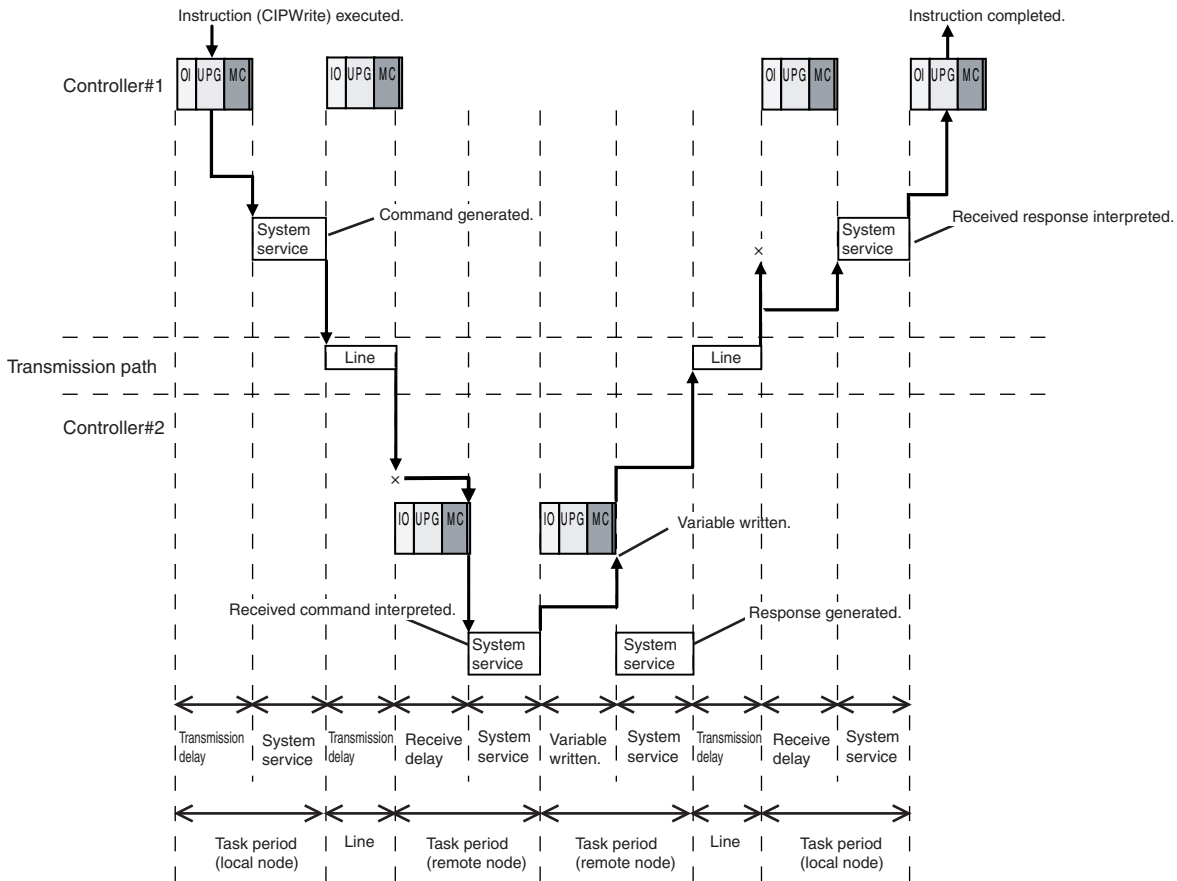


Additional Information

The I/O response time may be longer due to noise, or other events.

13-4 Message Service Transmission Delay

This section describes the delay time that occurs in CIP communications instruction (CIPWrite) service processing.



$$\begin{aligned}
 &\text{Maximum transmission delay time} = \underbrace{\text{Send delay} + \text{System service execution time} + \text{Transmission delay} + \text{Receive delay} + \text{System service execution time}}_{\text{(Local node task period)}} \\
 &+ \underbrace{\text{Variable write time} + \text{System service execution time} + \text{Transmission delay} + \text{Receive delay} + \text{System service execution time}}_{\text{(Remote node task period)}} \\
 &\qquad\qquad\qquad \underbrace{\hspace{10em}}_{\text{(Local node task period)}}
 \end{aligned}$$

Processes that cause a delay time are processed in the task periods at each node as shown in the above diagram. Line-based delays are as follows:

● **Transmission Delay**

The transmission delay on an Ethernet line is 50 μs or less. This delay time can be ignored.



Additional Information

- Depending on the actual operating environment, the transmission time may be longer than the one calculated with the equations given here. The following factors can cause longer transmission times: other traffic on the network, window sizes of network nodes, other traffic at the built-in EtherNet/IP port itself (e.g., simultaneous tag data link communications), and the system configuration.
 - CIP communications processing is executed as a system service. If a timeout occurs for a CIP communications instruction, reconsider the execution time for system services.
-

14

Checking Communications Status of Network and Troubleshooting

This section describes how to use the Network Configurator to check the communications status on the EtherNet/IP network, and how to identify and correct errors based on the connection status of tag data links.

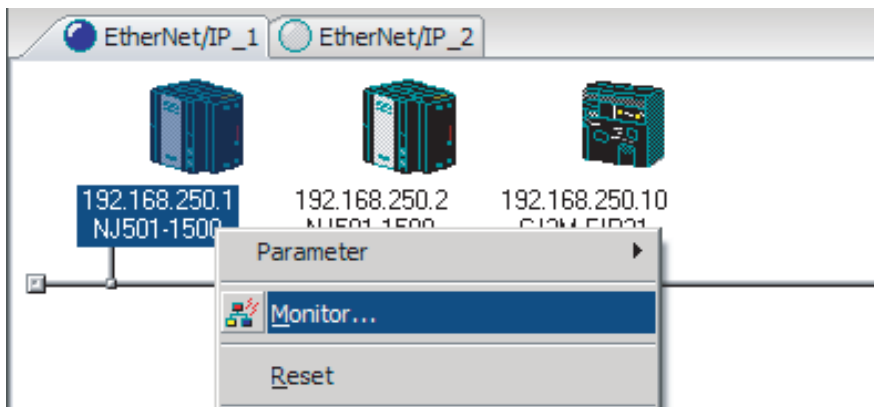
14-1 Device Monitoring on the Network Configurator	14-2
14-1-1 Starting the Device Monitoring	14-2
14-1-2 Status Displays of the Device Monitoring	14-2
14-2 Connection Status Codes and Troubleshooting	14-9

14-1 Device Monitoring on the Network Configurator

This section describes how to start the device monitoring and status to display on the Network Configurator.

14-1-1 Starting the Device Monitoring

Connect the Network Configurator online, select the device to be checked, right-click to display the pop-up menu, and select **Monitor**.



The Monitor Device Dialog Box will be displayed.



Additional Information

If a communications error occurs during monitoring, the dialog box will continue to show the last information that was collected. To start monitoring again, close the Monitor Device Dialog Box, and then open the dialog box again.

14-1-2 Status Displays of the Device Monitoring

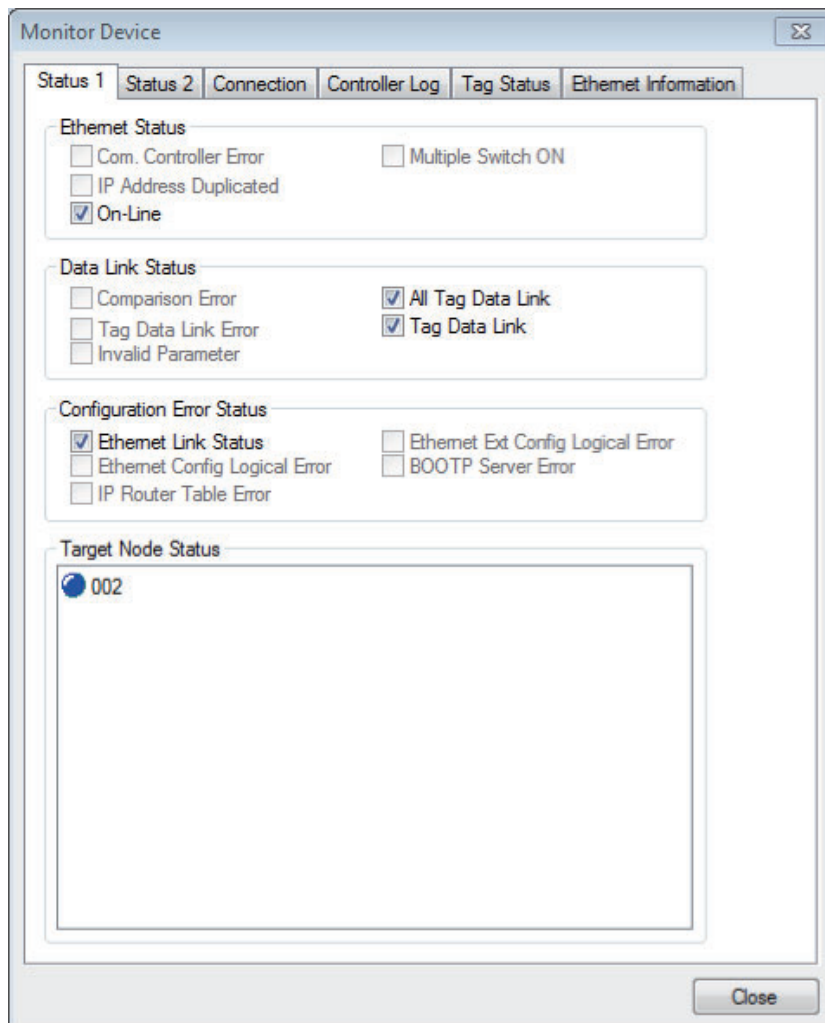
● Status 1 Tab Page

The following check boxes are displayed for the status. If a check box is selected, the status is TRUE.

Classification	Item	Corresponding system-defined variable
Ethernet Status	Com. Controller Error	An error occurred in the communications controller.
	IP Address Duplicated	The same IP address is assigned to more than one node.
	On-Line	Indicates that the Unit is online. (The EtherNet/IP Unit can perform communications processing.)
	Multiple Switch ON	Indicates that more than one data link start/stop switch changed to TRUE at the same time.

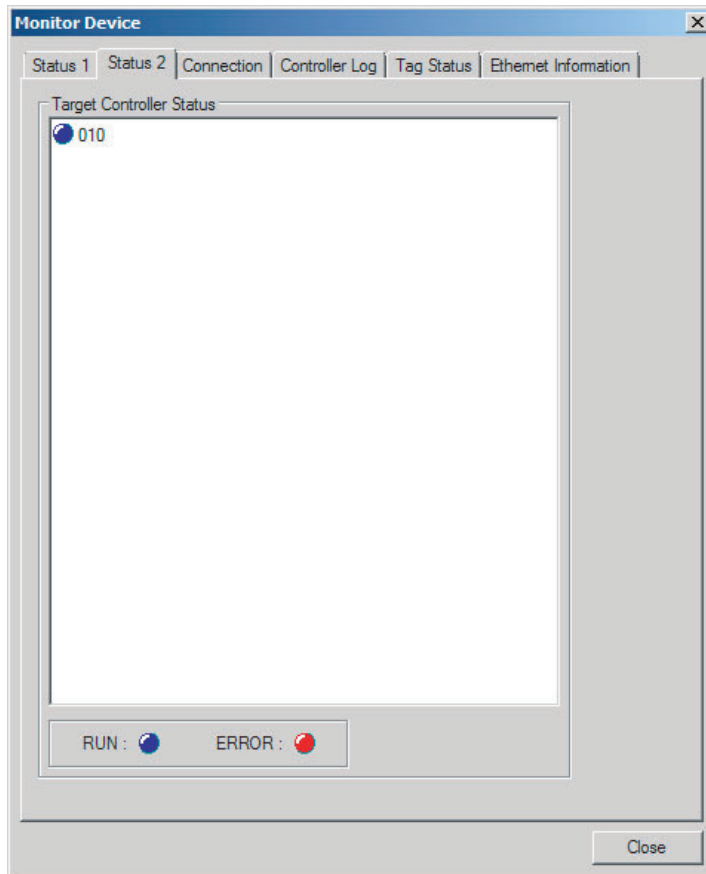
Classification	Item	Corresponding system-defined variable
Data Link Status	Comparison Error	The remote node information in the tag data link parameters was different from the actual node information. Main causes: •The specified target does not exist. •The variable name does not match. •The connection size is different. •Connection resources are not sufficient.
	Tag Data Link Error	There were two or more errors in a connection as an originator.
	Invalid Parameter	An error was found in the validation check of the parameters for tag data links that are saved in non-volatile memory.
	All Tag Data Links	Tag data links are communicating in all connections as the originator.
	Tag Data Link	Tag data links are communicating in one or more connections as the originator.
Configuration Error Status	Ethernet Link Status	TRUE when a link is established with the Ethernet switch.
	Ethernet Basic Settings Logic Error	TRUE when the following settings are incorrect: • TCP/IP settings (IP address, subnet mask, or link settings)
	IP Router Table Error	TRUE when there is a mistake in the IP router table information.
	Ethernet Ext Config Logical Error	Always FALSE.
	BOOTP Server Error	TRUE when one of the following errors occurs when using the BOOTP server. • The IP address received from the BOOTP server is incorrect. • A communications timeout occurred with the server.

Information about the target node that acts as the originator is displayed. If all tag data link connections to the node are established and normal, this information is displayed in blue. However, if any connection is broken it is displayed in red.



● Status 2 Tab Page

This tab page displays information on nodes with tag data link originator settings based on `_EIP_TargetPLCModeSta` (Target PLC Operating Mode) and `_EIP_TargetPLCErr` (Target PLC Error Information) in the `_EIP_EstbTargetSta` (Normal Target Node Information) system-defined variable. This information is in blue if the connection is normal, or red if an error occurred.

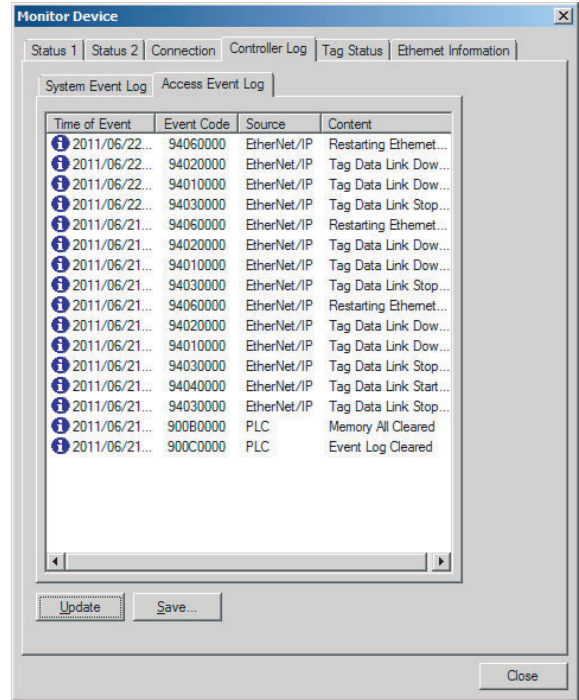
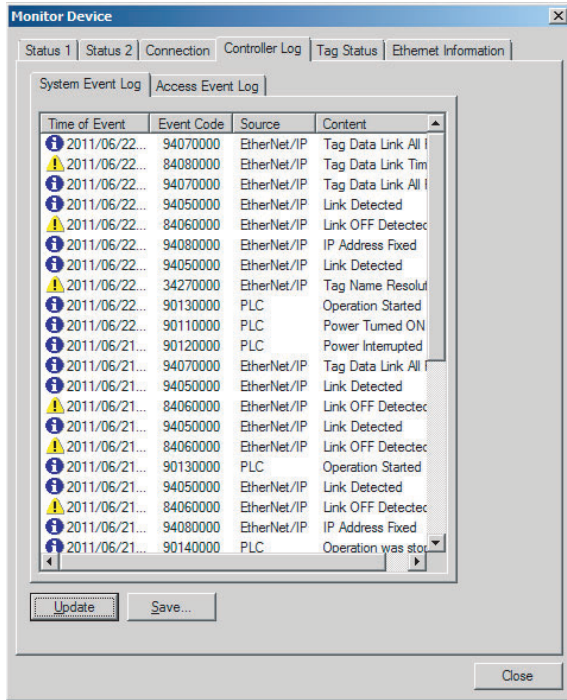


Additional Information

The target Controller status can be used when the Controller status is selected for all the target sets for both originator and target connections. If it is not selected, it is grayed out on the display.

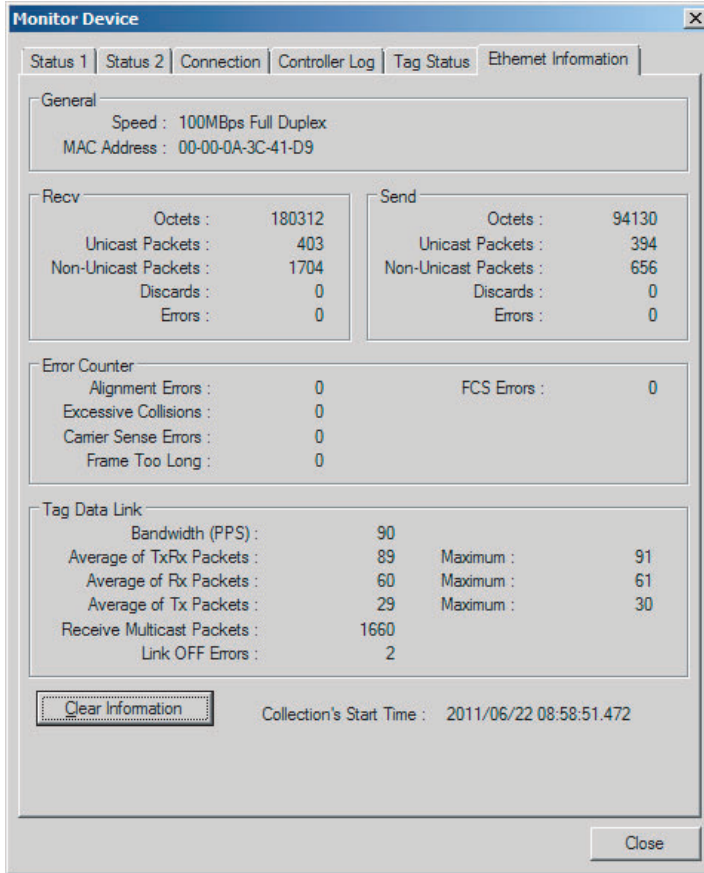
● **Controller Log Tab Page**

This tab page displays the Controller event log that is stored in the Controller. The error history shows errors that have occurred. It can be saved in a file in the computer. Refer to the operation manual of the Controller for details on error information.



● Ethernet Information Tab Page

This tab page displays the communications status at the communications driver level of the built-in Ethernet/IP port. The error counter information can be used to confirm whether communications problems have occurred. The tag data link information can be used to confirm characteristics such as the bandwidth usage (pps).



14-2 Connection Status Codes and Troubleshooting

This section explains how to identify and correct errors based on the tag data link's connection status. The connection status can be read using the Connection Tab Page of the Network Configurator's Monitor Device Window. Refer to *14-1 Device Monitoring on the Network Configurator* for details.



Additional Information

The connection status has the same meaning as the Connection Manager's General and Additional error response codes, as defined in the CIP specifications.

The following table shows the likely causes of the errors causes for each configuration and connection status (code).

	Originator	Target
Configuration 1	CS1W-EIP21, CJ1W-EIP21, CJ2H-CPU□□-EIP, CJ2M-CPU3□, NJ501-□□□□, NJ301-□□□□, NJ101-□□□□, and NX701-□□□□, NY5□2-□□□□	CS1W-EIP21, CJ1W-EIP21, CJ2H-CPU□□-EIP, CJ2M-CPU3□, NJ501-□□□□, NJ301-□□□□, NJ101-□□□□, and NX701-□□□□, NY5□2-□□□□
Configuration 2	CS1W-EIP21, CJ1W-EIP21, CJ2H-CPU□□-EIP, CJ2M-CPU3□, NJ501-□□□□, NJ301-□□□□, NJ101-□□□□, and NX701-□□□□, NY5□2-□□□□	Products from other manufacturers
Configuration 3	Products from other manufacturers	CS1W-EIP21, CJ1W-EIP21, CJ2H-CPU□□-EIP, CJ2M-CPU3□, NJ501-□□□□, NJ301-□□□□, NJ101-□□□□, and NX701-□□□□, NY5□2-□□□□

Connection status		Source of error	Handling		
General Status (hex)	Additional Status (hex)		Configuration 1	Configuration 2	Configuration 3
00	0000	Normal status code: The connection has been opened and the tag data link is communicating normally.	---	---	---
01	0100	Error code returned from target: Attempted to open multiple connections for the same connection.	This error does not occur.	Depends on the target's specifications. (This error should not occur. If it does, contact the target device's manufacturer.)	Depends on the originator's specifications. (This error should not occur. If it does, contact the originator device's manufacturer.)
01	0103	Error code returned from target: Attempted to open a connection with an unsupported transport class.	This error does not occur.	Confirm that the target supports Class 1.	Confirm that the originator supports Class 1.
01	0106	Duplicate consumers: Attempted to open multiple connections for single-consumer data.	If the tag data link is stopped or started, this error may occur according to the timing, but the system will recover automatically.	Depends on the target's specifications. (Contact the target device's manufacturer.)	If the tag data link is stopped or started, this error may occur according to the timing, but the system will recover automatically.
01	0107	Error code returned from target: Attempted to close a connection, but that connection was already closed.	This error does not occur.	This error does not occur.	This is not an error because the connection is already closed.

Connection status		Source of error	Handling		
General Status (hex)	Additional Status (hex)		Configuration 1	Configuration 2	Configuration 3
01	0108	Error code returned from target: Attempted to open a connection with an unsupported connection type.	This error does not occur.	Check which connection types can be used by the target. (Contact the manufacturer.) Only multicast and point-to-point connections can be set.	Check which connection types can be used by the originator. (An error will occur if a connection other than a multicast or point-to-point connection is set.)
01	0109	Error code returned from target: The connection size settings are different in the originator and target.	Check the connection sizes set in the originator and target.		
01	0110	Error code returned from target: The target was unable to open the connection, because of its operating status, such as downloading settings.	Check whether the tag data link is stopped at the target. (Restart the tag data link communications with the software switch.)	Depends on the target's specifications. (Contact the target device's manufacturer.)	Check whether the tag data link is stopped at the target. (Restart the tag data link communications with the software switch.)
01	0111	Error code returned from target: The RPI was set to a value that exceeds the specifications.	This error does not occur.	Check the target's RPI setting specifications.	Set the originator's RPI setting to 10 seconds or less.
01	0113	Error code generated by originator or returned from target: Attempted to open more connections than allowed by the specifications (32).	Check the connection settings (number of connections) at the originator and target.	Check the connection settings (number of connections) at the originator and target. Check the connection specifications for devices from other manufacturers.	Check the connection settings (number of connections) at the originator and target. Check the connection specifications for devices from other manufacturers.
01	0114	Error code returned from target: The Vendor ID and Product Code did not match when opening connection.	This error does not occur.	Depends on the target's specifications. (Contact the target device's manufacturer.) Check that the target device's EDS file is correct.	Check the originator's connection settings.
01	0115	Error code returned from target: The Product Type did not match when opening connection.	This error does not occur.	Depends on the target's specifications. (Contact the target device's manufacturer.) Check that the target device's EDS file is correct.	Check the originator's connection settings.
01	0116	Error code returned from target: The Major/Minor Revisions did not match when opening connection.	Check the major and minor revisions set for the target device and connection. If necessary, obtain the most recent EDS file and set it again.	Depends on the target's specifications. (Contact the target device's manufacturer.) Check that the target device's EDS file is correct.	Check the originator's connection settings.
01	0117	Error code returned from target: The tag set specified in the connection's target variables does not exist.	Check whether the originator and target tag sets and tags are set correctly.	Depends on the target's specifications. (Contact the target device's manufacturer.)	Check the originator's connection settings. Check whether the target tag sets and tags are set correctly.

Connection status		Source of error	Handling		
General Status (hex)	Additional Status (hex)		Configuration 1	Configuration 2	Configuration 3
01	011A	Error code generated by originator: Connection could not be established because the buffer was full due to high traffic.	Unexpected network traffic may have been received. Use the Network Configurator Device Monitor or the Ethernet Tab Page to check the bandwidth usage, and correct the load. If there are places where broadcast storms occur, such as loop connections in the network connection format, then correct them.	Unexpected network traffic may have been received. Use the Network Configurator Device Monitor or the Ethernet Tab Page to check the bandwidth usage, and correct the load. If there are places where broadcast storms occur, such as loop connections in the network connection format, then correct them.	Depends on the target's specifications. (Contact the target device's manufacturer.)
01	011B	Error code returned from target: The RPI was set to a value that is below the specifications.	This error does not occur.	Depends on the target's specifications. (Contact the target device's manufacturer.)	Set the originator's RPI setting to 1 ms or greater.
01	0203	Error code generated by originator: The connection timed out.	Tag data link communications from the target timed out. Check the power supply and cable wiring of the devices in the communications path, including the target and switches. If performance has dropped due to heavy traffic, change the performance settings. For example, increase the timeout time or RPI setting.		
01	0204	Error code generated by originator: The connection open process timed out.	There was no response from the target. Check the power supply and cable wiring of the devices in the communications path, including the target and switches.		
01	0205	Error code returned from target: There was a parameter error in the frame used to open the connection.	This error does not occur.	Depends on the target's specifications. (Contact the target device's manufacturer.)	Depends on the originator's specifications. (Contact the originator device's manufacturer.)
01	0302	Error code generated by originator or returned from target: The tag data link's allowable bandwidth (pps) was exceeded.	Check the connection settings (number of connections and RPI) at the originator and target.	Check the target's connection settings (number of connections and RPI). Check the connection settings (number of connections and RPI) at the originator and target.	Check the connection settings (number of connections and RPI) at the originator and target.
01	0311	Error code returned from target: There was a parameter error in the frame used to open the connection.	This error does not occur.	Depends on the target's specifications. (Contact the target device's manufacturer.)	Depends on the originator's specifications. (Contact the originator device's manufacturer.)
01	0312	Error code returned from target: There was a parameter error in the frame used to open the connection.	This error does not occur.	Depends on the target's specifications. (Contact the target device's manufacturer.)	Depends on the originator's specifications. (Contact the originator device's manufacturer.)
01	0315	Error code returned from target: There was a parameter error in the frame used to open the connection.	This error does not occur.	Depends on the target's specifications. (Contact the target device's manufacturer.)	Depends on the originator's specifications. (Contact the originator device's manufacturer.)
01	0316	Error code returned from target: There was a parameter error in the frame used to close the connection.	This error does not occur.	Depends on the target's specifications. (Contact the target device's manufacturer.)	Depends on the originator's specifications. (Contact the originator device's manufacturer.)
01	031C	Error code generated by originator: Some other error occurred.	This error does not occur.	The originator generates this code when an unsupported response code is returned from the target in reply to an open request.	Depends on the originator's specifications. (Contact the originator device's manufacturer.)

Connection status		Source of error	Handling		
General Status (hex)	Additional Status (hex)		Configuration 1	Configuration 2	Configuration 3
08	---	Error code returned from target: There is no Forward Open or Large Forward Open service in the target device.	This error does not occur.	Depends on the target's specifications. (Contact the target device's manufacturer.)	Depends on the originator's specifications. (Contact the originator device's manufacturer.)
D0	0001	Error code generated by originator: The connection operation is stopped.	The connection was stopped because the Tag Data Link Stop Bit was turned ON, or the settings data is being downloaded. Either turn ON the Tag Data Link Start Switch, or wait until the settings data has been downloaded. This code includes fatal Controller errors and Unit failure. To handle these errors, refer to the <i>NY-series Troubleshooting Manual</i> (Cat. No. W564).	The meaning of this error code is defined by each vendor, so it depends on the target's specifications. (Contact the target device's manufacturer.)	Depends on the originator's specifications. (Contact the originator device's manufacturer.)
D0	0002	Error code generated by originator: The connection is being opened (opening processing in progress).	Wait until the opening processing is completed.	The meaning of this error code is defined by each vendor, so it depends on the target's specifications. (Contact the target device's manufacturer.)	Depends on the originator's specifications. (Contact the originator device's manufacturer.)
OMRON error code					
01	0810	Error code returned from target: New data could not be obtained from the Controller when opening connection. (The Unit will automatically recover, and attempt to open the connection again.)	This error may occur if the Controller's task period was long when opening the connection or some problem in the Controller caused the Controller to stop. If the task period was too long, operation recovers automatically. If the Controller has stopped, identify the error from the error information in the Controller.	The meaning of this error code is defined by each vendor, so it depends on the target's specifications. (Contact the target device's manufacturer.)	The meaning of this error code is defined by each vendor, so it depends on the originator's specifications. (Contact the originator device's manufacturer.)
01	0811	Error code generated by originator: New data could not be obtained from the Controller when opening connection. (The Unit will automatically recover, and attempt to open the connection again.)	This error may occur if the Controller's task period was long when opening the connection. If the task period was too long, operation recovers automatically.	The meaning of this error code is defined by each vendor, so it depends on the target's specifications. (Contact the target device's manufacturer.)	The meaning of this error code is defined by each vendor, so it depends on the originator's specifications. (Contact the originator device's manufacturer.)



Appendices

A-1	Functional Comparison of the EtherNet/IP Port with Other Series	A-2
A-2	Use the Sysmac Studio to Set the Tag Data Links (EtherNet/IP Connections)	A-3
A-2-1	Overview of the Tag Data Links (EtherNet/IP Connections) Settings with the Sysmac Studio	A-3
A-2-2	Procedure to Make the EtherNet/IP Connection Settings with the Sysmac Studio	A-4
A-2-3	EtherNet/IP Connection Settings	A-5
A-2-4	Making the EtherNet/IP Connection Settings with the Sysmac Studio	A-10
A-2-5	Checking Communications Status with the Sysmac Studio and Troubleshooting	A-33
A-2-6	Troubleshooting	A-37
A-3	EDS File Management	A-43
A-3-1	Installing EDS Files	A-44
A-3-2	Creating EDS Files	A-44
A-3-3	Deleting EDS Files	A-45
A-3-4	Saving EDS Files	A-45
A-3-5	Searching EDS Files	A-46
A-3-6	Displaying EDS File Properties	A-47
A-3-7	Creating EDS Index Files	A-47
A-4	Precautions for Using the Network Configurator on Windows XP, Windows Vista, or Windows 7 or Higher	A-48
A-4-1	Changing Windows Firewall Settings	A-48
A-5	Variable Memory Allocation Methods	A-51
A-5-1	Variable Memory Allocation Rules	A-51
A-5-2	Important Case Examples	A-59
A-6	Precautions When Accessing External Outputs in Controllers	A-63
A-7	TCP State Transitions	A-64

A-1 Functional Comparison of the EtherNet/IP Port with Other Series

OK: Supported., ---: Not supported.

Item	Built-in EtherNet/IP port on NY-series Controller	Built-in EtherNet/IP port on NX-series Controller	Built-in EtherNet/IP port on NJ-series Controller	CJ-series EtherNet Unit	EtherNet/IP Unit (built-in port on CJ2 CPU Unit)		
					Unit version 1.0	Unit version 2.0	Unit version 2.1
Tag data link communications service	OK	OK	OK	---	OK	OK	OK
CIP message communications service	OK	OK	OK	---	OK	OK	OK
IP routing	OK	OK	---	---	---	---	---
Socket service	OK	OK	OK	OK	---	---	---
FTP server	OK	OK	OK	OK	---	OK	OK
FTP client	OK	OK	OK	---	---	---	---
Mail send/receive	---	---	---	OK	---	---	---
Web functions	---	---	---	OK	---	---	---
Automatic adjustment of PLC/Controller's internal clock	---	OK	OK	OK	---	OK	OK
Error history	OK*1	OK*1	OK*1	OK	OK	OK	OK
Response to PING command	OK	OK	OK	OK	OK	OK	OK
SNMP/SNMP trap	OK	OK	OK	---	---	OK	OK
CIDR function for IP addresses	OK	OK	OK	---	---	OK	OK
Online connection via EtherNet/IP using CX-One	---	---	---	OK	---	OK	OK
Online connection via EtherNet/IP using Network Configurator	OK	OK	OK	---	OK	OK	OK
Connection settings using the Sysmac Studio	OK	OK	OK	---	---	---	OK

*1 This is equivalent to the event log in the built-in EtherNet/IP.

A-2 Use the Sysmac Studio to Set the Tag Data Links (EtherNet/IP Connections)

A-2-1 Overview of the Tag Data Links (EtherNet/IP Connections) Settings with the Sysmac Studio

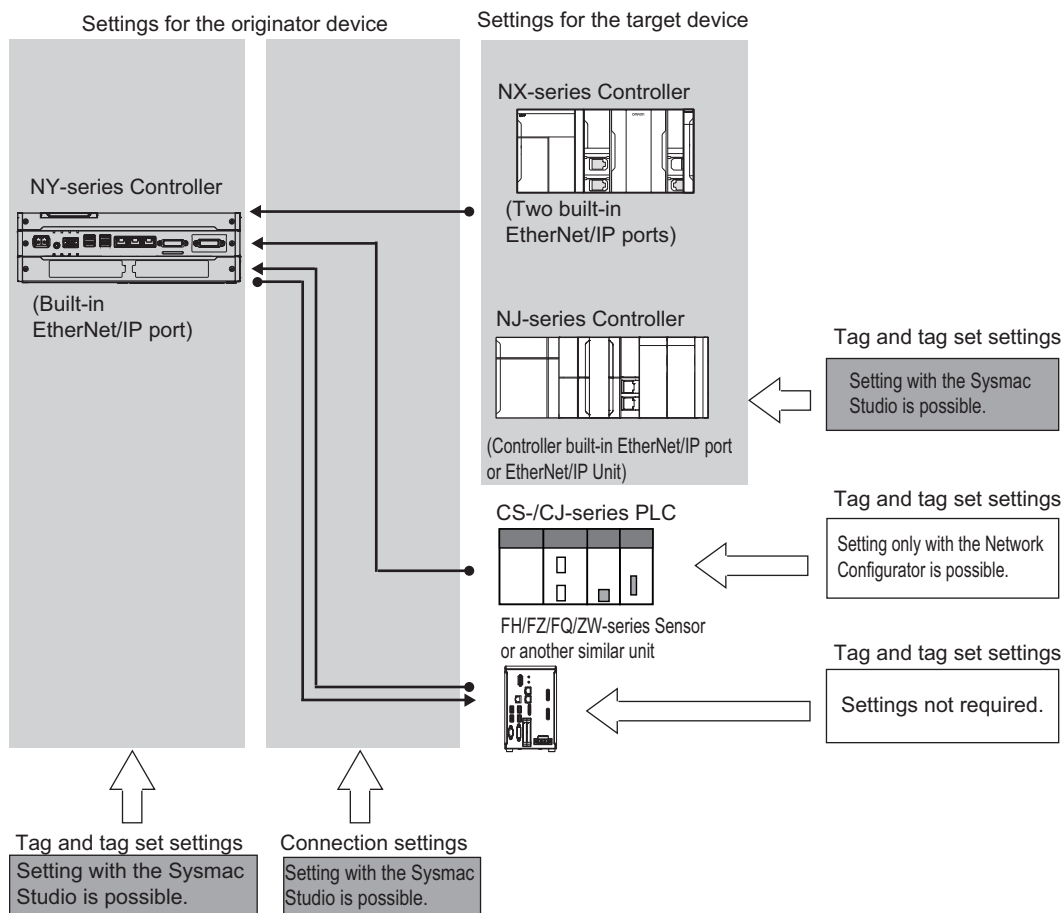
You can use the Sysmac Studio to set the settings required for creating tag data links (EtherNet/IP connections)*¹ between NY-series Controllers.

*1 The tag data links and EtherNet/IP connections enable cyclic tag data exchanges on an EtherNet/IP network between Controllers or between Controllers and other devices. Here, "EtherNet/IP connection" refers to both the tag data links and the EtherNet/IP connections.

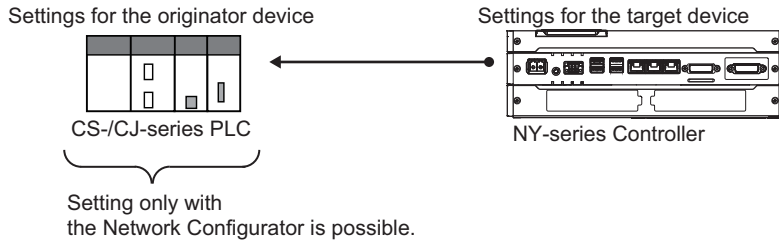
Acceptable System Configuration Conditions for Setting the EtherNet/IP Connection Settings on the Sysmac Studio

If an NY-series Controller operates as the originator device, you can use the Sysmac Studio to set the originator device settings for the EtherNet/IP connections.

Similarly, if an NY-series Controller operates as the target device, you can use the Sysmac Studio to set the tags and tag sets of the target device.



Use the Network Configurator if a CS-/CJ-series PLC operates as the originator device.



A-2-2 Procedure to Make the EtherNet/IP Connection Settings with the Sysmac Studio

1	Registering devices		<ul style="list-style-type: none"> • Main Window
<p>Register devices with which the EtherNet/IP connections are established to the project.</p>			
2	Creating network variables(*1)	... Refer to <i>Registering the Network Variable for the Originator Device</i> on page A-11.	<p>Setup Window</p> <p>Global Variable Table on the Sysmac Studio</p>
<p>Register the network variables that are set in step 2 as tags and tag sets.</p>			
3	Registering tags and tag sets	... Refer to <i>Registering the Tag and Tag Set</i> on page A-13.	<p>EtherNet/IP Connection Settings (Tag Set Display)</p>
<p>Specify devices (i.e., target devices and originator devices) and tag sets to communicate with using the EtherNet/IP connections.</p>			
4	Setting Connections	... Refer to <i>Setting Connections for the Originator Device</i> on page A-16.	<p>EtherNet/IP Connection Settings (Connections Display)</p>
<p>Specify devices (i.e., target devices and originator devices) and tag sets to communicate with using the EtherNet/IP connections.</p>			
5	Going online from the Sysmac Studio	... Refer to <i>Transferring the Connection Settings Data</i> on page A-28.	<ul style="list-style-type: none"> • Main Window
<p>Specify devices (i.e., target devices and originator devices) and tag sets to communicate with using the EtherNet/IP connections.</p>			
6	Downloading EtherNet/IP connection settings	... Refer to <i>Transferring the Connection Settings Data</i> on page A-28.	<ul style="list-style-type: none"> • Synchronization Window / Transfer to Controller Dialog Box • EtherNet/IP Connection Settings
<p>Note Connections automatically start after the download.</p>			
7	Checking operation Stopping and starting connections	... Refer to <i>A-2-5 Checking Communications Status with the Sysmac Studio and Troubleshooting</i> on page A-33.	<p>EtherNet/IP Connection Monitor Tab Page</p>
<p>Stopping and starting connections</p>			

*1 Variables with its Network Publish attribute set to **Output** or **Input** in the Global Variable Table are called network variables.

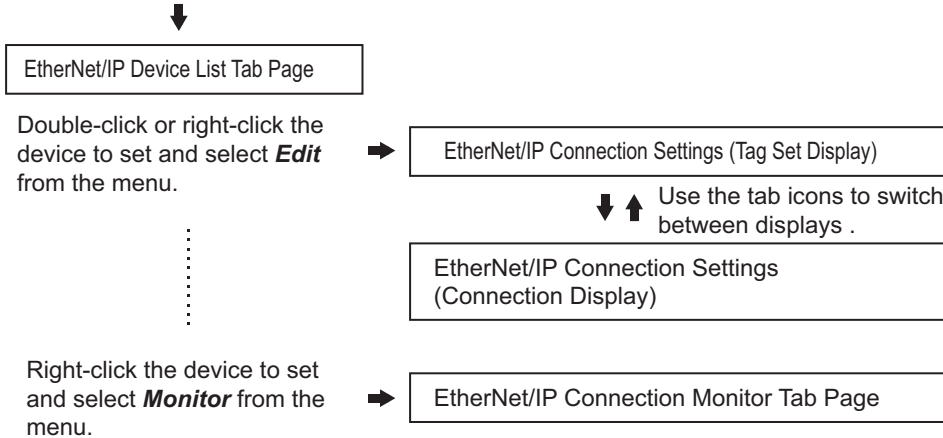
A-2-3 EtherNet/IP Connection Settings

This section describes the screen configuration for EtherNet/IP connection settings.

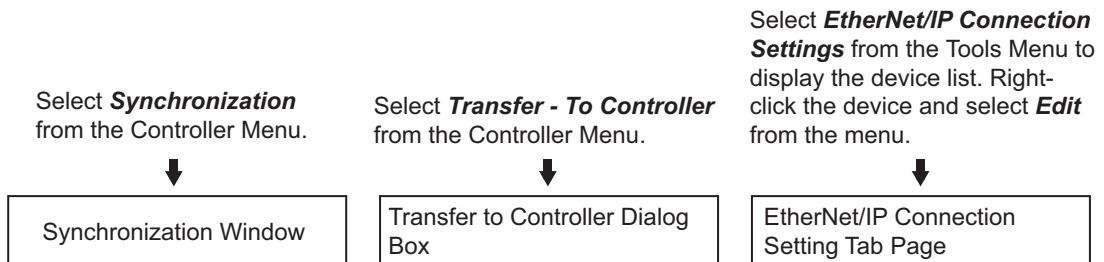
● Screen Transitions in the EtherNet/IP Connection Settings

- Connection Settings

Select **EtherNet/IP Connection Settings** from the Tools Menu.



- Transferring connection settings to the Controller from the computer



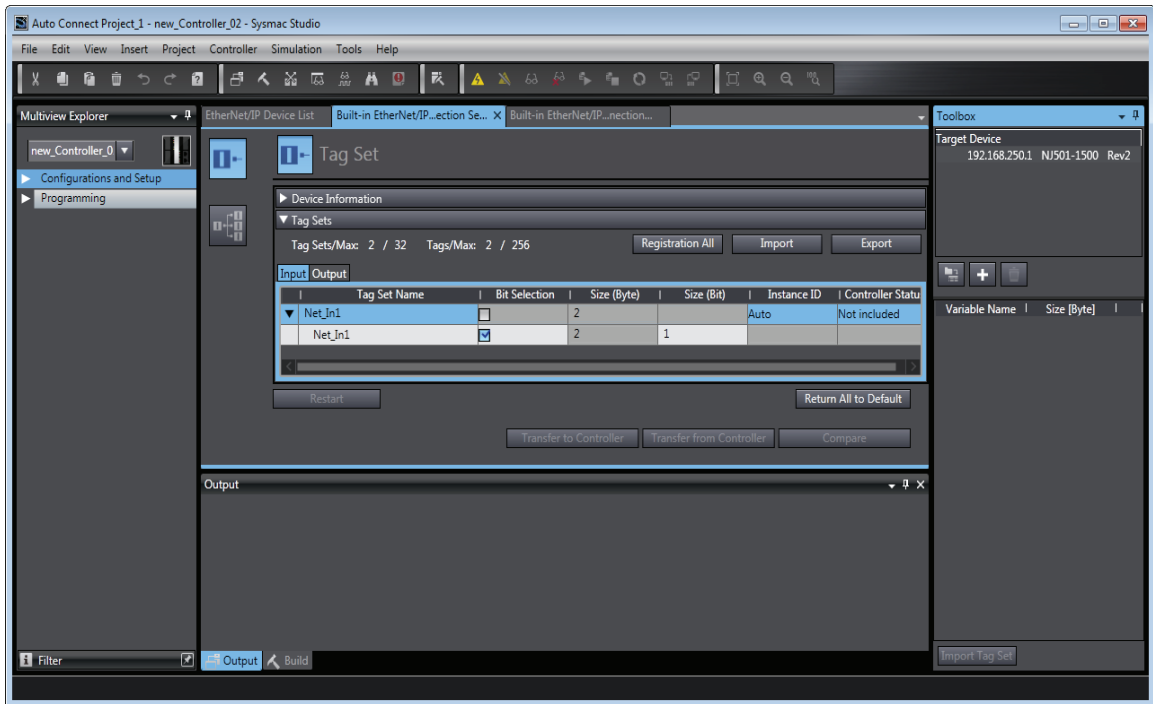
Precautions for Correct Use

To transfer only the connection settings, execute Transfer from the EtherNet/IP Connection Setting Tab Page.

The connection settings are not transferred from the Synchronization Window, the Transfer to Controller Dialog Box, and the Transfer from Controller Dialog Box, even if you clear the *Do not transfer the connection setting* Check Box, as long as the data in the computer and in the Controller is the same.

- **EtherNet/IP Device List Tab Page (Refer to *Registering the Tag and Tag Set* on page A-13.)**

The list indicates the devices to which EtherNet/IP connections can be set.



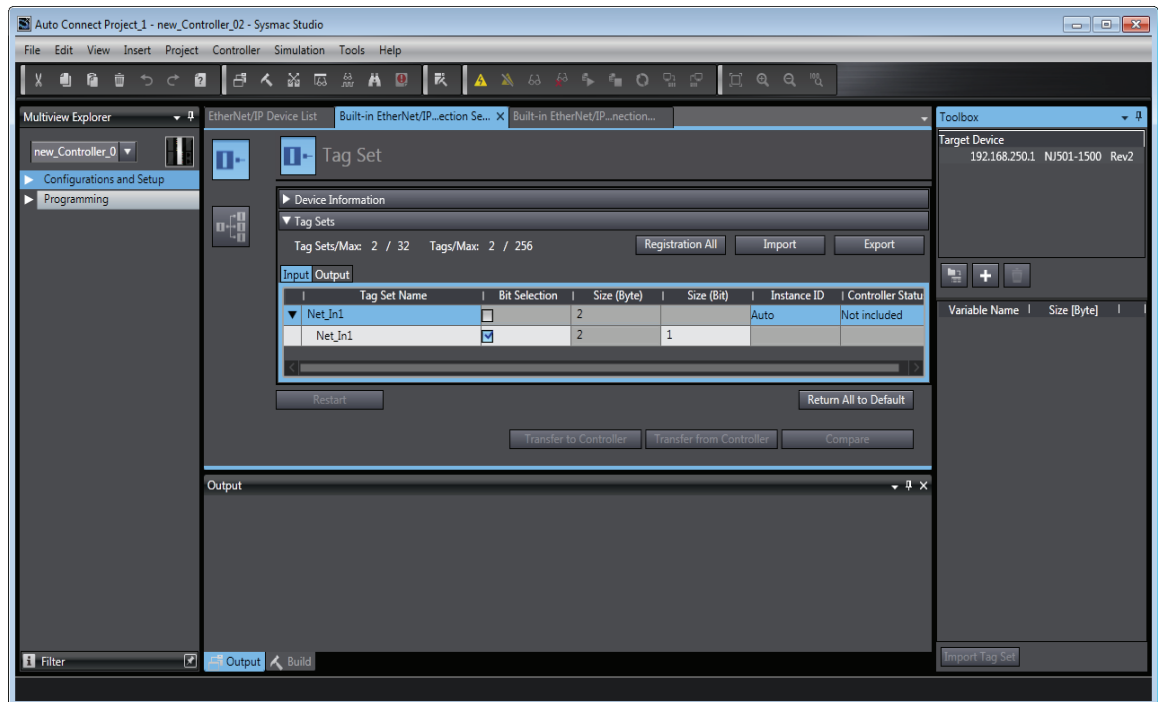
● **EtherNet/IP Connection Settings (Tag Set Display) (Refer to *Registering the Tag and Tag Set* on page A-13.)**

Register tag sets required to create connections. Each tag set represents the data that is sent and received through a connection. You can register up to eight tags in one tag set.

The name and size of the tag must be the same as those of the network variable*¹.

Set whether to include the Controller status information in tags for the tag sets. You can also set the data output operation at a fatal error occurrence for output tags.

*1 Variables with its Network Publish attribute set to **Output** or **Input** in the Global Variable Table are called network variables.



● **EtherNet/IP Connection Settings (Connection Display) (Refer to *Setting Connections for the Originator Device* on page A-16.)**

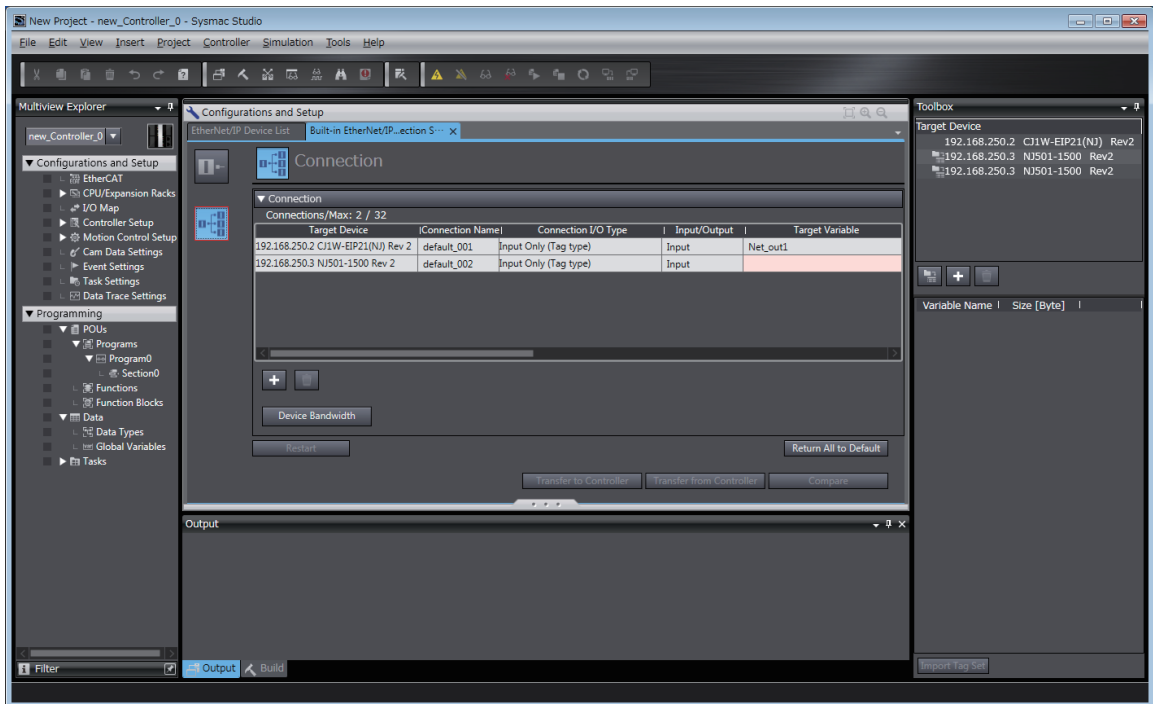
Specify the target devices and set their connections.

For each connection, set the following information: Connection Name, Connection I/O Type, I/O, target device tag set (target variable), originator device tag set (originator variable), Packet Interval (RPI), and Timeout Value.



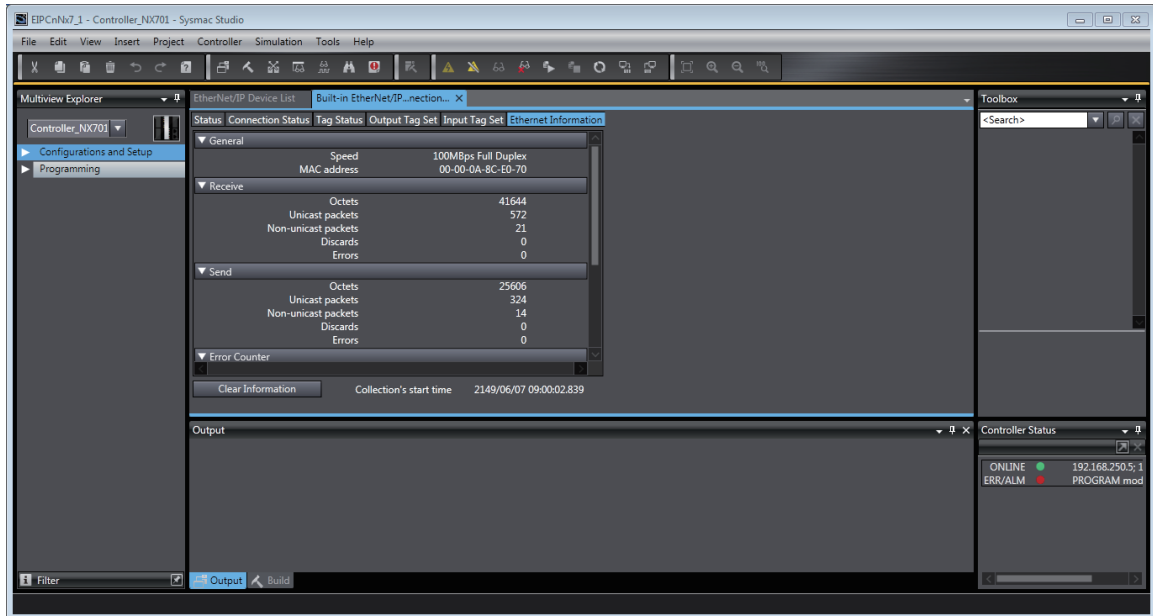
Precautions for Correct Use

If you changed the IP address, model or revision of the target device after making the connection settings, change the connection settings entirely.



- **EtherNet/IP Connection Monitor Tab Page (Refer to A-2-5 Checking Communications Status with the Sysmac Studio and Troubleshooting on page A-33.)**

You can check the EtherNet/IP connection setting status offline and communications status online. When online, you can start and stop connections.



A-2 Use the Sysmac Studio to Set the Tag Data Links (EtherNet/IP Connections)

A

A-2-3 EtherNet/IP Connection Settings

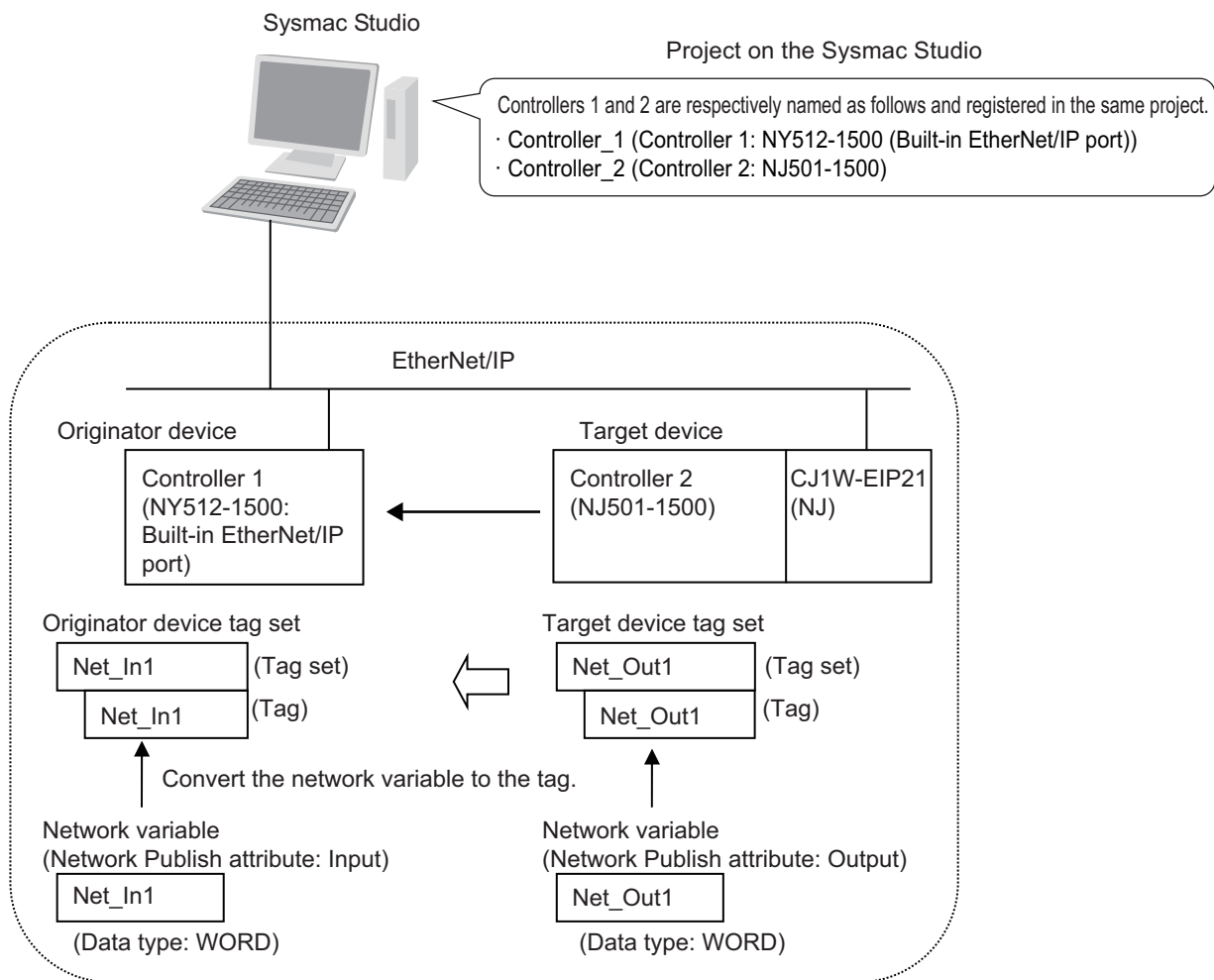
A-2-4 Making the EtherNet/IP Connection Settings with the Sysmac Studio

This section describes the procedure to make the EtherNet/IP connection settings with the Sysmac Studio.

Here, we take the following system configuration as an example to describe how to set the EtherNet/IP connection settings.

Example: System that connects the built-in EtherNet/IP port on Controller 1 and the built-in EtherNet/IP port on Controller 2 via Ethernet

- Set the settings so that values in the network variable *Net_Out1* allocated for Controller 2 are sent to the network variable *Net_In1* allocated for Controller 1 at the set RPI of 50 ms cycle.
- This example assumes the programs for both Controllers 1 and 2 are registered in the same project.

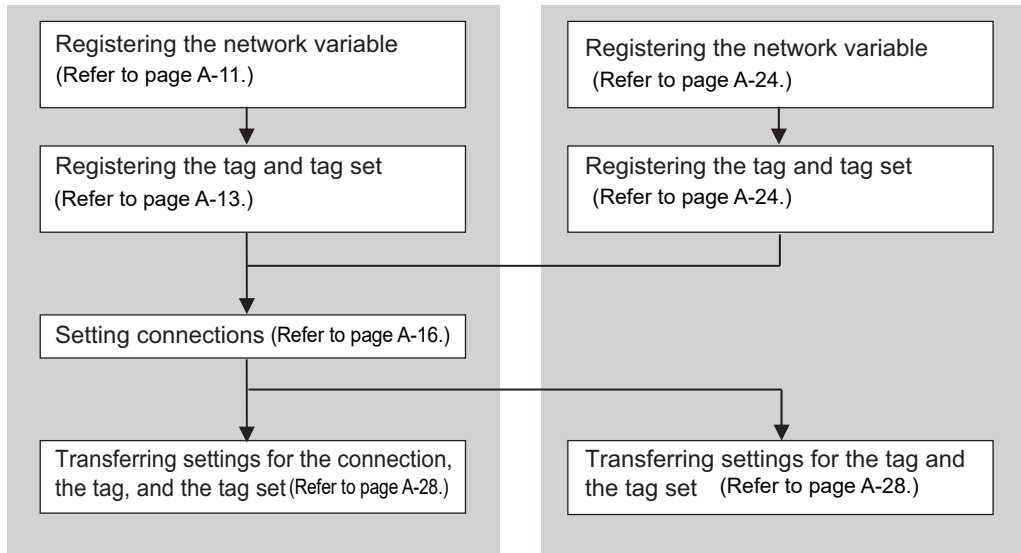


Follow the flow below to set the settings to Controllers 1 and 2 for which to establish EtherNet/IP connections.

The required settings for the originator device and the target device are shown below.

Settings for the originator device (Controller 1)

Settings for the target device (Controller 2)



Registering the Network Variable for the Originator Device

Register the network variable that is sent and received using the EtherNet/IP connections.

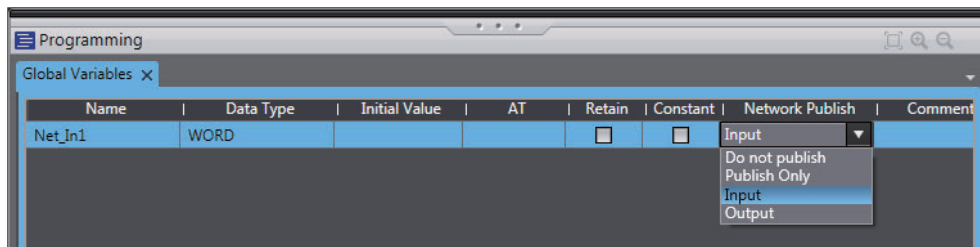
Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for the operations for registering variables.

- 1 Assign the network variable to the tag used for the EtherNet/IP connection for Controller 1 (originator device).

This network variable receives data from Controller 2 (target device).

Set the Network Publish attribute to **Input** or **Output** in the Global Variable Table for the variable so that the variable serves as a network variable, i.e., the variable can be used for the EtherNet/IP connections.

In this example, set the network variable for Controller 1 as shown below.



- Variable name: Net_In1
- Data type: WORD
- Network Publish attribute: Input

● **Network Variables Used for the EtherNet/IP Connections**

- Network variable name

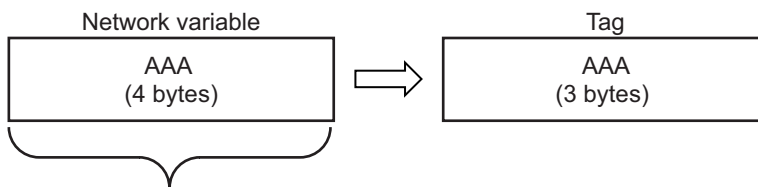
You cannot specify an I/O memory address for a tag name in the EtherNet/IP connection settings. Thus, do not specify an I/O memory address for the network variable name that is to be assigned to a tag.

The following text strings are recognized as the I/O memory address names.

- (1) **Variable names that contain only single-byte numerals from 0000 to 6143**
- (2) **Variable names with the following single-byte letters (uppercase or lowercase) followed by single-byte numerals**
 - H (H000 to H511)
 - W (W000 to W511)
 - D (D00000 to D32767)
 - E0_ to E18_ (E0_00000 to E0_32767, to E18_00000 to E18_32767)

- Size of variables

To use an EtherNet/IP Unit as an EtherNet/IP device, set an even number of bytes for the size of the network variable used for the EtherNet/IP connections regardless of an odd number of bytes for the tag size.



The Controller memory is consumed in units of two bytes. To assign tags of odd numbers of bytes to network variables, specify even byte numbers (i.e., sizes of the tags + 1) to the network variables.

- Data concurrency

To maintain concurrency in the values of network variables that are assigned to tags, you must set refreshing tasks.

Refer to *7-1-7 Concurrency of Tag Data Link Data* for details.

Registering the Tag and Tag Set

Register the required tag and tag set for the EtherNet/IP connections.

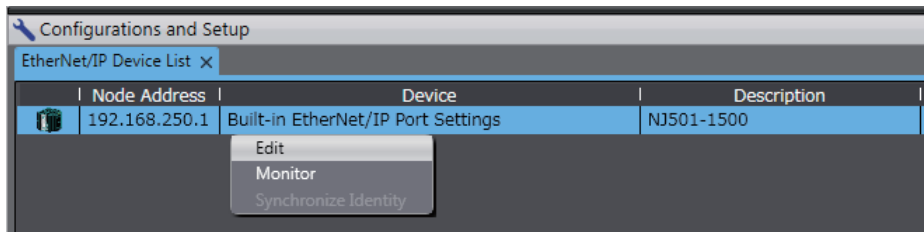
You can register tags and tag sets in the EtherNet/IP Connection Setting Tab Page.

Precautions for Correct Use

Make the following settings to refresh all of the tag data in the same tag set at the same time.

- Use the Sysmac Studio, in advance, to specify the same refreshing task for all of the variables that are assigned to tags in the tag set.

- 1 Select **EtherNet/IP Connection Settings** from the Tools Menu.
The EtherNet/IP Device List Tab Page is displayed.
- 2 In this example, right click *Built-in EtherNet/IP Port Settings* for the originator device and select **Edit** from the menu to open the EtherNet/IP Connection Setting Tab Page.



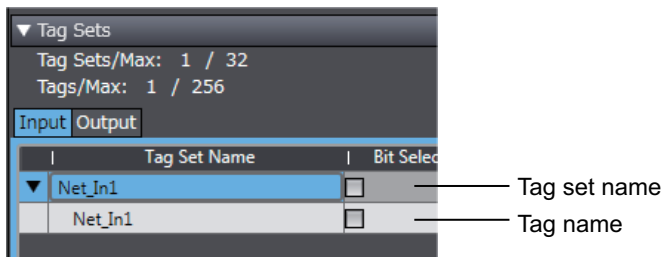
- 3 Click the  (Show Tag Set Display) icon in the EtherNet/IP Connection Setting Tab Page.

- 4 Click the **Input** tab to switch to the Input Tab Page. Register the tag set and the tag.

Use one of the following methods to register the tag set and the tag.

- Independent registration : Manually registers network variables in the Controller as tags.
- Batch registration : Registers all network variables in the Controller as tags at the same time.

- 5 Register tags and tag sets independently.
 - (1) Right-click anywhere in the Input Tab Page of the EtherNet/IP Connection Setting Tab Page and select **Create New Tag Set** from the menu.
 - (2) Enter tag set name *Net_In1* directly into the list in the Input Tab Page.
 - (3) Right-click anywhere in the Input Tab Page and select **Create New Tag** from the menu.
 - (4) Enter tag name *Net_In1*.





Precautions for Correct Use

Any name can be specified for the tag set if the name matches one of the registered network variable names in the Controller.

As you enter characters (or immediately after you press the Ctrl + Space Keys), the Sysmac Studio Entry Assistance provides a list of variable names registered in the Controller. Select the variable name from the list.



Additional Information

You can register up to 8 tags in a tag set.

Set as shown below to register multiple tags.

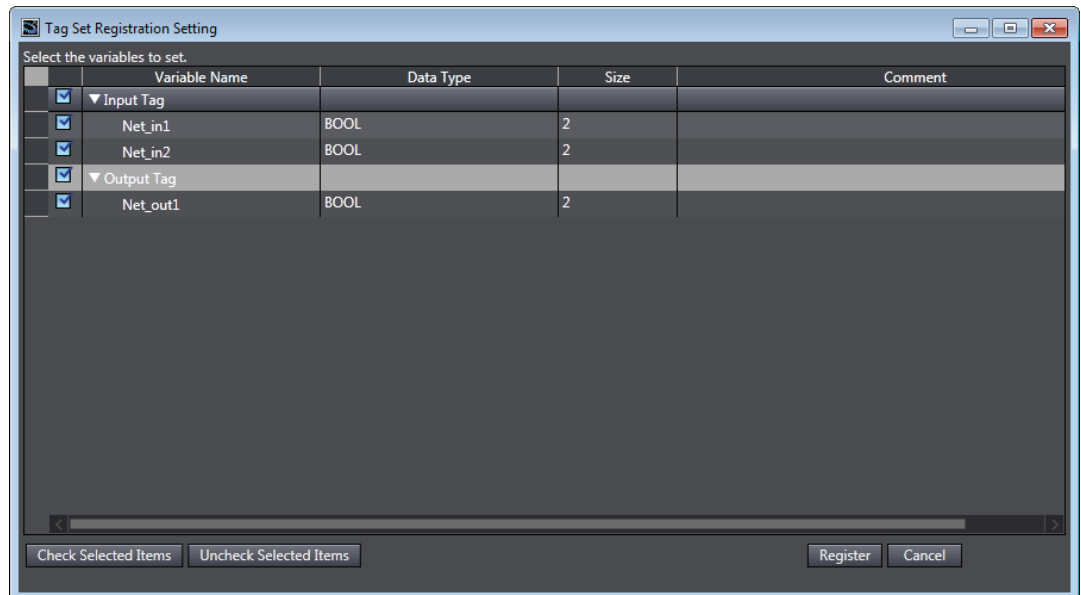
Example:

	Tag set name	
▼	Network_Input_Value (Tag set name)
	Net_In1 (Tag name)
	Net_In2 (Tag name)

6 Register all tags and tag sets at the same time.

- (1) **Right-click anywhere on the Input Tab Page of the EtherNet/IP Connection Settings Tab Page and select *Register All Tag Sets* or click the Registration All Button to display the Tag Set Registration Setting Dialog Box.**

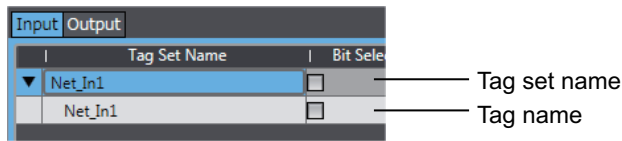
This dialog box lists the variables that are registered in the Global Variable Table and also have the Network Publish attribute set to *Input* or *Output*.



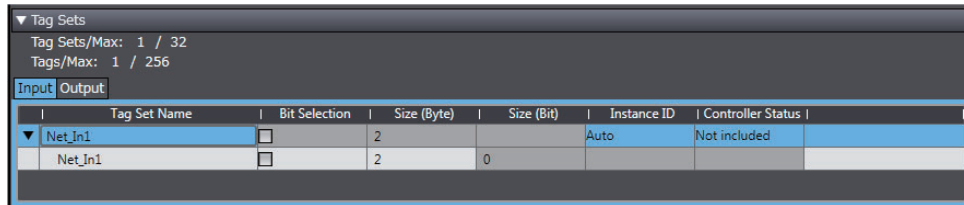
- (2) **Select the variable to register as a tag, and then click the Register Button.**

(3) The automatically registered tag is added to the list in the EtherNet/IP Connection Setting Tab Page.

With automatic registration, the tag is registered under a tag set having the same name as the tag, i.e., a single tag is registered in a single tag set.



7 Set the following settings for the registered tag and tag set.



• Setting for Tag Sets

Name	Item
Tag Set Name	Enter the tag set name. You can change the names as required.
Size (Byte)	Gives the total size of the tag in bytes.
Instance ID	Gives the instance ID. <ul style="list-style-type: none"> • Auto • IN_{min}...IN_{max} {min} represents the minimum number of Produced Assembly identification numbers recorded in the EDS files for the relevant devices. {max} represents the maximum number of Produced Assembly identification numbers recorded in the EDS files for the relevant devices.
Controller Status	Specify whether to include the Controller status in the tag set.

• Setting for Tags

Name	Item
Tag Name	Enter the tag name. Specify the tag name that matches one of the registered network variable names in the Controller.
Bit Selection	Specify whether to set the tag data size in bits. Selected: Set the size in bits. Not selected: Set the size in bytes.
Size (Byte)	Gives the size of the tag in bytes.
Size (Bit)	Gives the size of the tag in bits.
Output at Fatal Error	Specify whether to clear the output data or continue to send it when a major fault level Controller error occurs in the Controller. <ul style="list-style-type: none"> • Retained • Cleared

Setting Connections for the Originator Device

After the tag set registration, set the connection settings for transferring data using the EtherNet/IP connections.

Make the connection settings in the originator device (i.e., Controller 1 in this example) only.

Register the tag and tag set for Controller 2 (Target device) before setting the connection settings as described in this example.

Refer to *Registering the Tag and Tag Set for the Target Device* on page A-24 for the operations for registering tags and tag sets.

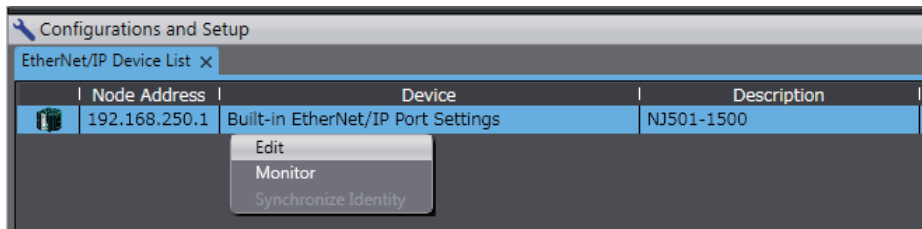



Precautions for Correct Use

If you change the IP address, model, or revision of the target device after making the connection settings, you must also change the target device settings that are included in the connection settings. For information on how to change the target device settings in the connection settings, refer to *Changing the Target Device Settings After Making Connection Settings*.

- 1 Select **EtherNet/IP Connection Settings** from the Tools Menu to display the EtherNet/IP Device List Tab Page.
- 2 Right click **Built-in EtherNet/IP Port Settings** for Controller 1 (originator device in this example) and select **Edit** from the menu.

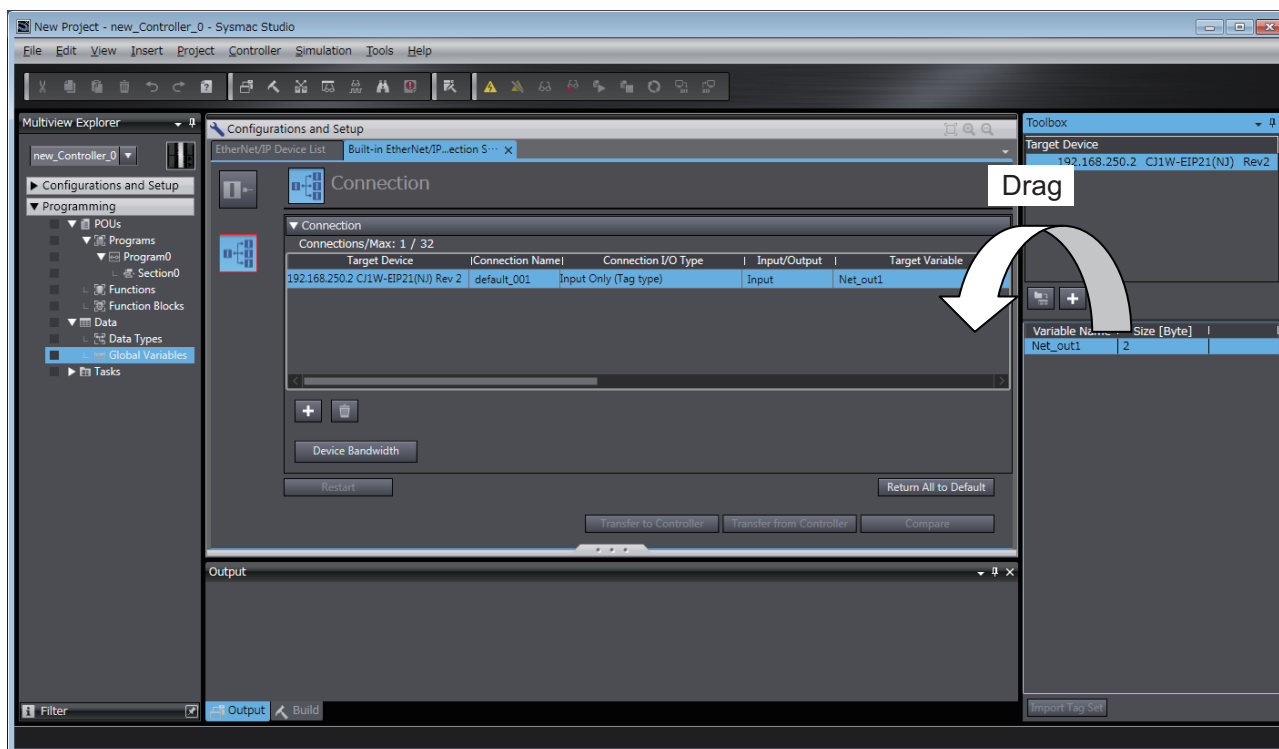
The EtherNet/IP Connection Setting Tab Page is displayed.



- 3 Click the  (Show Connection Display) icon in the EtherNet/IP Connection Setting Tab Page.
- 4 Select **CJ1W-EIP21(NJ)** from **Target Device** in the Toolbox on the right of the tab page.
This operation displays the target device tag set *Net_Out1* that is set for Controller 2 in the **Variable Name** column.

5 Drag the target device tag set **Net_Out1** in the **Variable Name** column of the Toolbox to the connection list.

As you enter characters (or immediately after you press the Ctrl + Space Keys), a list of target device variables that can be set for the connection is provided. Select the value from the list.



6 Specify **Originator Variable** and its **Size [Byte]** for the tag set **Net_Out1** added in step 5. Here, specify **Net_In1** for **Originator Variable** and 2 for its **Size [Byte]**.

Change the other settings as required.

You can set the following items in the connection settings.

Name	Setting Methods
Target Device	Select the target device.
Connection Name	Any name can be given to the connection (32 single-byte characters max.).
Connection I/O Type	Input Only (Tag type) is selected if EtherNet/IP connections are used with a CS1W-EIP21, CJ1W-EIP21, CJ2B-EIP21, CJ2M-EIP21, CJ1W-EIP21(CJ2), CJ1W-EIP21(NJ), NX701, NJ501-□□□□, NJ301-□□□□, NJ101, or NY5□2. When you create EtherNet/IP connections for other target devices, select the connection I/O type specified in that device's EDS file. Use the Input Only (ID type) setting when another company's node is the originator and does not support connection settings with a Tag type setting.
Input/Output	The connection's input/output is automatically displayed based on the selected connection. Input Only: Just Input is displayed.
Target Variable	Select the target node's tag set to assign it. <ul style="list-style-type: none"> Input is specified for Input/Output: Select the target's output (produce) tag set. Output is specified for Input/Output: Select the target's input (consume) tag set.

Name	Setting Methods
Size [Byte]	The data sizes of the target variables are displayed.
Originator Variable	Select the originator node's tag set to assign it. <ul style="list-style-type: none"> • Input is specified for Input/Output: Select the originator's input (consume) tag set. • Output is specified for Input/Output: Select the originator's output (produce) tag set.
Size [Byte]	Enter the data sizes of the originator variables.
Connection Type	Select whether the data is sent in multi-cast or unicast (point-to-point) form. The default setting is multi-cast. <ul style="list-style-type: none"> • Multi-cast connection: Select when the same data is shared by multiple nodes. This setting is usually used. • Point-to-point connection: Select when the same data is not shared by multiple nodes. In a unicast transmission, other nodes are not burdened with an unnecessary load. <p>Note Refer to <i>7-1-4 Overview of Operation</i> for details on using multi-cast and unicast connections, and counting the number of connections.</p>
RPI [ms]	Set the data update cycle (i.e., the packet interval) of each connection between the originator and target. <p>The default setting is 50 ms (i.e., data is updated once every 50 ms).</p>
Timeout Value	Set the time until a connection timeout is detected. <p>The timeout value is set as a multiple of the packet interval (RPI) and can be set to 4, 8, 16, 32, 64, 128, 256, or 512 times the packet interval.</p> <p>The default setting is RPI x 4.</p> <p>The timeout value must be at least 10 ms.</p>

7 The Toolbox displays the target devices if the devices are registered in the same Sysmac Studio project as where the originator devices are registered.

You can use one of the following methods to add unregistered devices in the same Sysmac Studio project as where the originator devices are registered to the Target Device List.

- Importing devices that are registered in another project
 You can import NJ/NX/NY-series Controllers registered in another project data and add them to the Device List.
- Registering devices using user-specified settings
 You can manually add target devices to the device list.




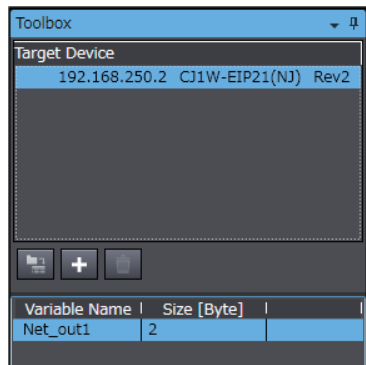
Additional Information

You can add target devices to the Device List by installing EDS files that include connection information for the devices in the Sysmac Studio and register the devices to the project.

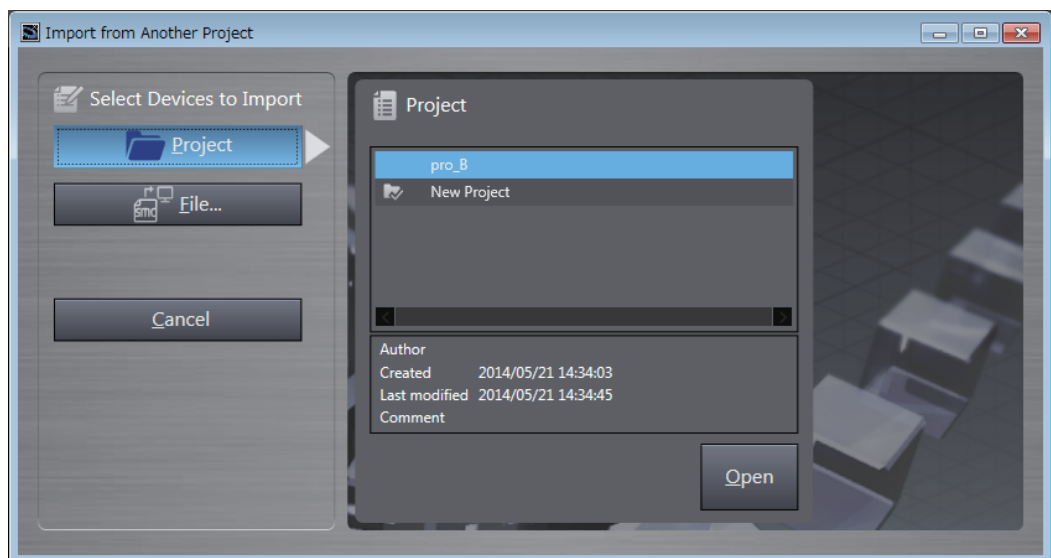
Refer to *Adding EDS Files* on page A-21 for details.

8 Import devices that are registered in another project.

- (1) Click the  (Import a device from another project) Button in the Toolbox on the right of the EtherNet/IP Connection Setting Tab Page.

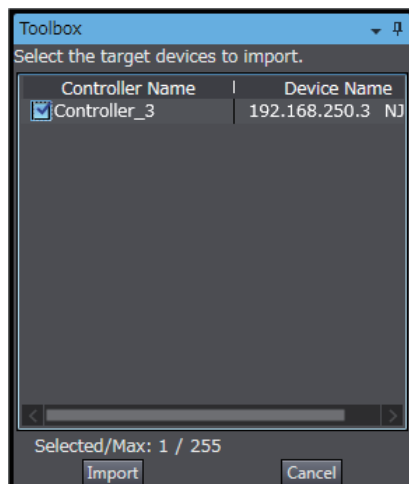


- (2) The Import from Another Project Dialog Box is displayed. Click the Project Button, select a project to import and click the Open Button.



- (3) The list of EtherNet/IP devices registered in the selected project will be displayed. Select the target devices to import, and click the Import Button.

Note Only the project for which the EtherNet/IP connection settings are set will be displayed. The imported EtherNet/IP devices are added to the Target Device List in the Toolbox.

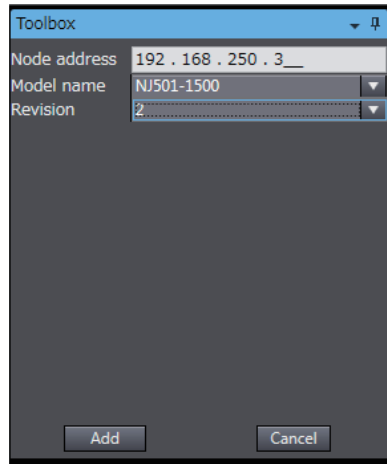


9 Register devices as required.

- (1) **Click the + Button under the Target Device List in the Toolbox.**

The Add Target Device Pane is displayed.

- (2) **Enter relevant items for the target devices to add.**



Menu	Description
Node address	Enter the target device IP address.
Model name	Select the target device model.
Revision	Select the revision of the target device.

- (3) **Here, set the following items for Controller 3 and click the Add Button.**

The target device is added to the Target Device List in the Toolbox.

Node address: 192.168.250.3

Model name: NJ501-1500

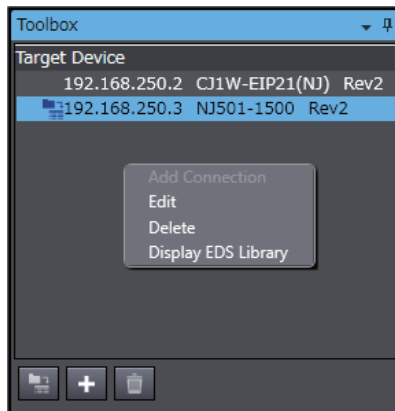
Revision: 2

- (4) **You can click the Import Tag Set Button to import the tag sets that are set in the Network Configurator to the target devices.**

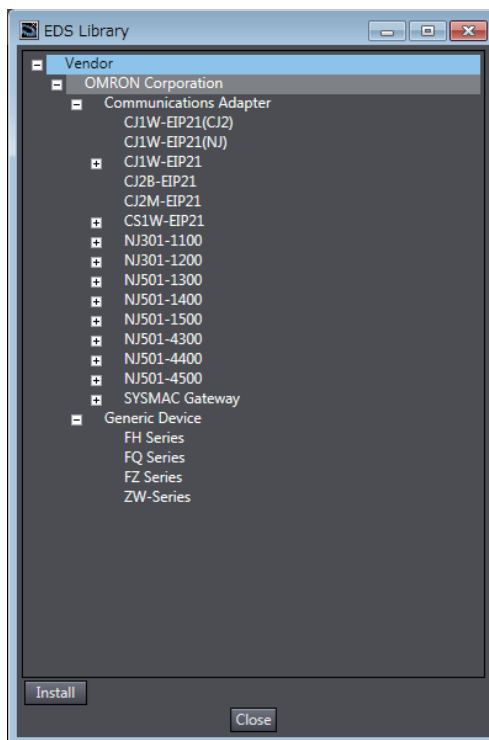
Select **Export to File** from the **To/From File** Button in the Tag Sets Tab Page of the Edit Device Parameters Dialog Box to generate CSV files to import.

● Adding EDS Files

- 1 Right-click anywhere in the Target Device List in the Toolbox of the EtherNet/IP Connection Setting Tab Page and select **Display EDS Library** from the menu.



- 2 The EDS Library Dialog Box is displayed. Click the Install Button.



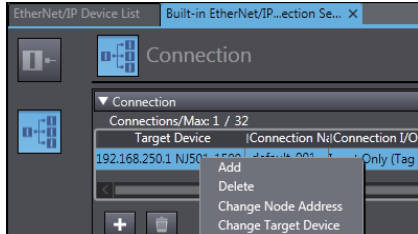
- 3 Select the EDS file to add, and then click the **Open** Button.
The EDS file is added.
- 4 The EtherNet/IP device with the EDS file installed is added to the EDS Library.
Devices listed in the EDS Library are used as a candidate device list when adding devices to the Target Device List in the Toolbox of the EtherNet/IP Connection Setting Tab Page.

● **Changing the Target Device Settings After Making Connection Settings**

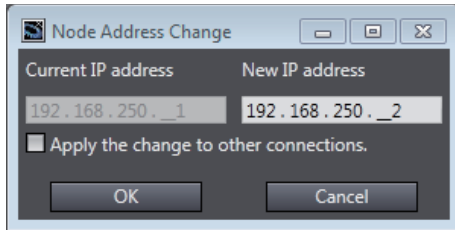
If you change the IP address, model, or revision of the target device after making the connection settings, you must also change the target device settings that are included in the connection settings. You can change the target device settings entirely.

Changing the IP Addresses for All Target Devices

- 1 Right-click one of the connection lines and select **Change Node Address** from the menu.



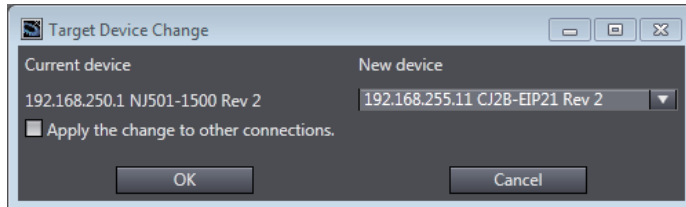
- 2 The Node Address Change Dialog Box is displayed. Enter a new IP address in *New IP address*.



- 3 To apply the same change to other connections, select the *Apply the change to other connections* Check Box.
- 4 Click the **OK** Button.

Changing All Target Device Information including Model Names and Revisions

- 1 Right-click one of the connection lines and select **Change Node Address** from the menu.
- 2 The Target Device Change Dialog Box is displayed. Select a target device from *New device*.



Precautions for Correct Use

- Changeable target devices are limited to ones that have "OMRON" in the Vendor ID and is an EDS device of the Communications Adapter in the Device Type.
- To display a device in the list of selectable new target devices, the device must be registered as the target device in the Toolbox.

- 3 To apply the same change to other connections, select the *Apply the change to other connections* Check Box.
- 4 Click the **OK** Button.

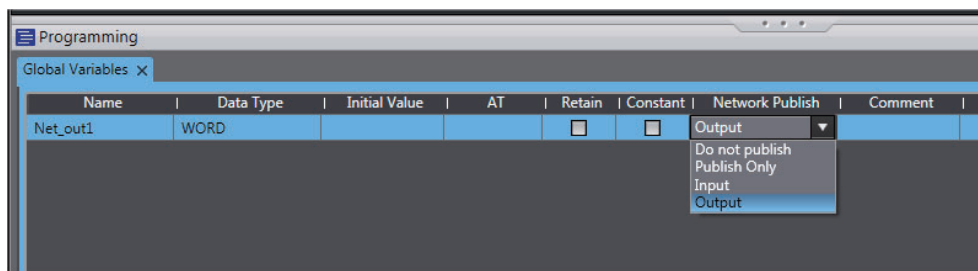
Registering the Network Variable for the Target Device

- 1 Assign the network variable to the tag used for the EtherNet/IP connection for Controller 2 (target device).

This network variable stores data to send to Controller 1 (originator device).

Set the Network Publish attribute to **Input** or **Output** in the Global Variable Table for the variable so that the variable serves as a network variable, i.e., the variable can be used for the EtherNet/IP connections.

In this example, set the network variable for Controller 1 as shown below.



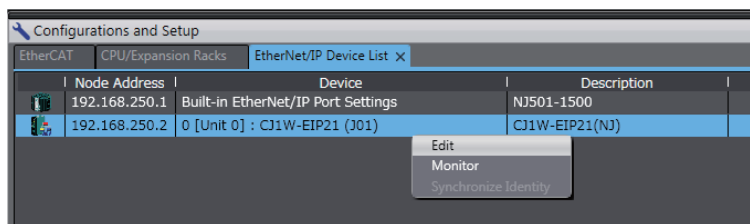
- Name: Net_Out1
- Data type: WORD
- Network Publish attribute: Output


Registering the Tag and Tag Set for the Target Device

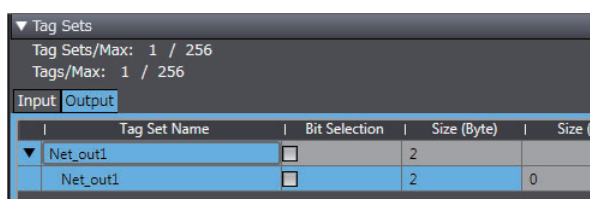
Set the tag and tag set for the target device.

- 1 Select **EtherNet/IP Connection Settings** from the Tools Menu. The EtherNet/IP Device List Tab Page is displayed.
- 2 Right-click CJ1W-EIP21, the EtherNet/IP Unit connected to the Controller 2 (originator device in this example), and select **Edit** from the menu.

The EtherNet/IP Connection Setting Tab Page is displayed.



- 3 Click the  (Show Tag Set Display) icon in the EtherNet/IP Connection Setting Tab Page.
- 4 Click the **Output** tab to switch to the Output Tab Page. Register the following tag and tag set. The tag and tag set can be registered in the same way as for the target device. (Refer to *Registering the Tag and Tag Set* on page A-13.)



Checking the Device Bandwidth Usage

The bandwidth usage for the device can be displayed from the EtherNet/IP Connection Setting Tab Page.

This value is for when multicast filtering is used.



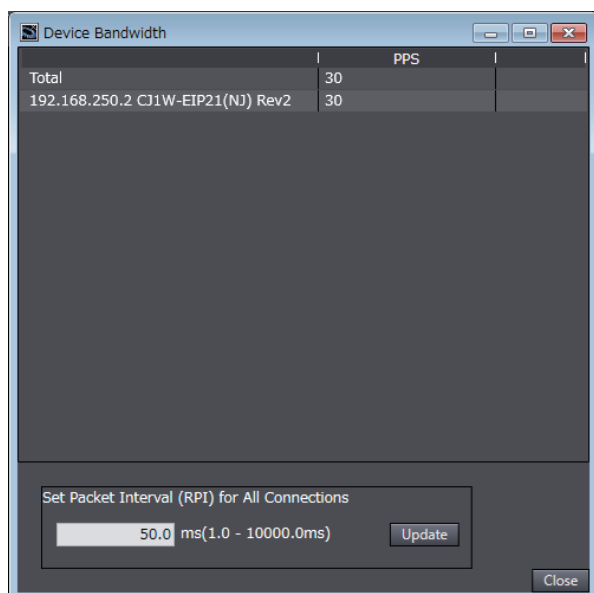
Precautions for Correct Use

In the Device Bandwidth Dialog Box, you can only check the bandwidth being used for the EtherNet/IP connections from one originator device to its target devices.

The actual bandwidth used for the EtherNet/IP network must be calculated by taking into account of all bandwidths used on the EtherNet/IP network (i.e., bandwidths used for connections for the other devices in the EtherNet/IP network than the one given on the dialog box must be included into the calculation).

● Procedure

Click the **Device Bandwidth** Button in the EtherNet/IP Connection Setting Tab Page for the target device.



Menu	Description
PPS	Gives the bandwidth used for each target device and total bandwidth used for all target devices.
Set Packet Interval (RPI) for All Connections	Changes all Packet Interval (RPI) values for all target devices.



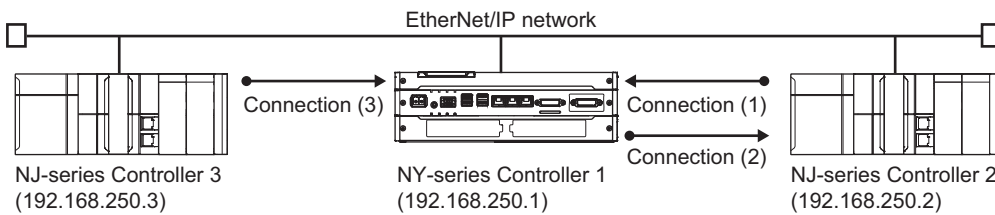
Additional Information

You can specify a value in **Set Packet Interval (RPI) for All Connections** and click the **Update** Button to change packet interval (RPI) values set in the connection settings for all target devices to the specified value.

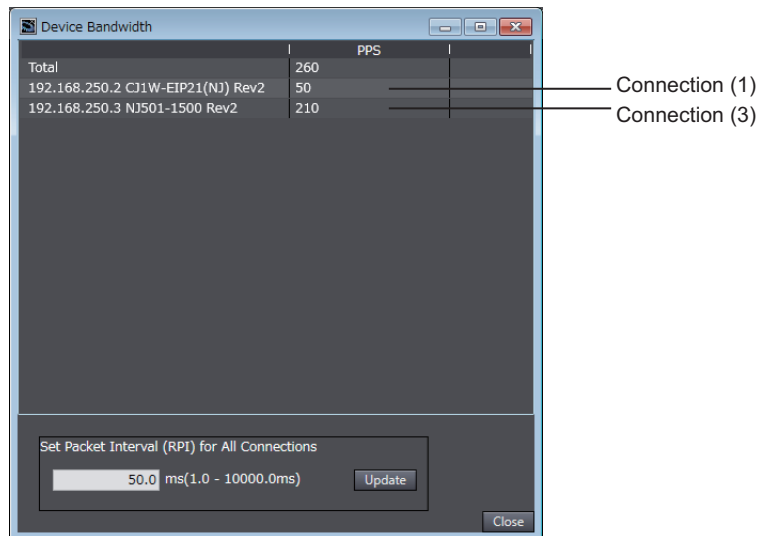
● **Calculation Example for Bandwidth Used (PPS) for Each Device by the EtherNet/IP Connections**

Establishing following three EtherNet/IP connections between Controllers (1) to (3) in the EtherNet/IP network

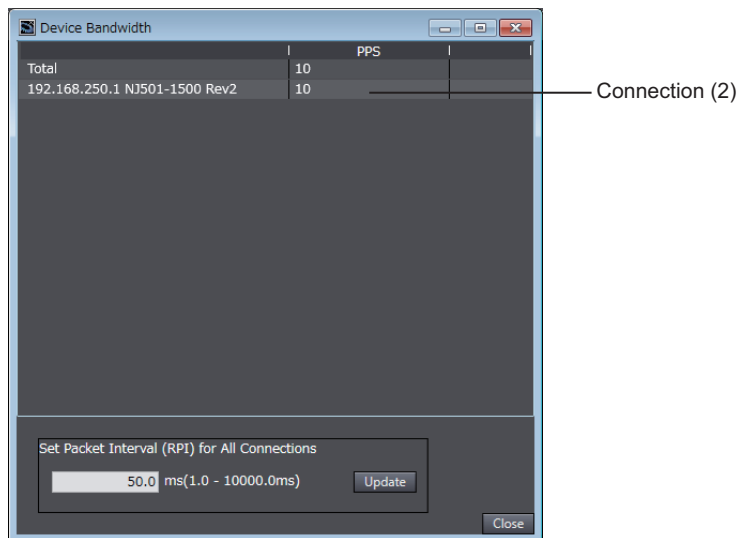
Connection type	Relevant devices in the EtherNet/IP connections	Device bandwidth usage (PPS)
Connection (1)	NJ-series Controller 2 (target device) to NY-series Controller 1 (originator device)	50 pps
Connection (2)	NY-series Controller 1 (target device) to NJ-series Controller 2 (originator device)	10 pps
Connection (3)	NJ-series Controller 3 (target device) to NY-series Controller 1 (originator device)	210 pps



Bandwidth used (PPS) for each EtherNet/IP device is as given below.



EtherNet/IP connection settings for Controller 1



EtherNet/IP connection settings for Controller 2

In this example, the PPS for Connection (1) is 50 pps, the PPS for Connection (2) is 10 pps, and the PPS for Connection (3) is 210 pps. Therefore, bandwidth used (PPS) for each EtherNet/IP device is as given below.

192.168.250.1: 270 pps = 50 pps (for Connection (1)) + 10 pps (for Connection (2)) + 210 pps (for Connection (3))

192.168.250.2: 60 pps = 50 pps (for Connection (1)) + 10 pps (for Connection (2))

192.168.250.3: 210 pps = 210 pps (for Connection (3))

● **Adjusting Method**

If the calculation result value exceeds the values in the specifications of the devices used in the EtherNet/IP connections, re-evaluate the overall network configuration and correct it by taking steps such as selecting a different Ethernet switch or splitting the network.

If the RPI is made longer, the PPS for the EtherNet/IP connections will decrease.

You can change the RPI value in the connection settings for all target devices by specifying a value in Set Packet Interval (RPI) for All Connections in this dialog box.

Refer to *13-2-2 Tag Data Link Bandwidth Usage and RPI* on page 13-8 for the relationship between the PPS for the device and the RPI.

Transferring the Connection Settings Data




Precautions for Correct Use

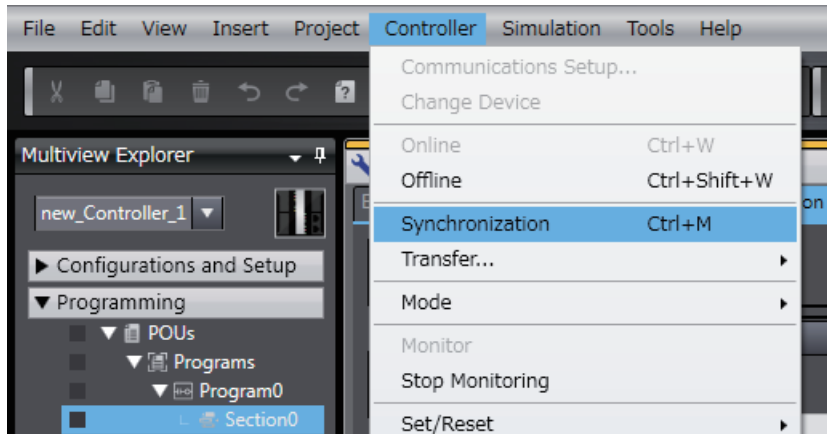
- If the node addresses (IP addresses) are not set correctly, you may connect to the wrong Controller and set incorrect device parameters. Download data only after you confirm that you are connected to the correct Controller.
 - If incorrect connection settings are set, it may cause equipment to operate unpredictably. Even when the correct connection settings are set, make sure that there will be no effect on equipment before you transfer the data.
 - A connection error will result if the network variables that are used in the tag settings are not set in the Controller. Before downloading the connection settings, check to confirm that the network variables used in the tag settings are set in the Controller.
 - If a communications error occurs, the output status depends on the specifications of the device being used. When a communications error occurs for a device that is used along with output devices, check the operating specifications and implement safety countermeasures.
 - The built-in EtherNet/IP port is automatically restarted after the parameters are downloaded. This restart is required to enable the tag set and connection information. Before you download the parameters, check to confirm that problems will not occur with the equipment when the port is restarted.
 - Do not disconnect the Ethernet cable during the parameter download.
 - The EtherNet/IP connections between relevant nodes is stopped during a download. Before you download data in RUN mode, make sure that it will not affect the controlled system.
Also implement interlocks on data processing in ladder programming that uses EtherNet/IP connections when the connections are stopped or a connection error occurs.
 - In the EtherNet/IP network, if the device bandwidth usage (PPS) exceeds the unit's allowable bandwidth (PPS), the EtherNet/IP connection operations may not agree with the settings.
If you increase the RPI value in such a case, there are cases when the problem can be resolved (i.e., the operations agree the settings).
-

● Synchronizing/Transferring a Whole Project

You can synchronize and transfer the EtherNet/IP connection settings along with the program data. You can also transfer all the EtherNet/IP connection settings along with the program data.

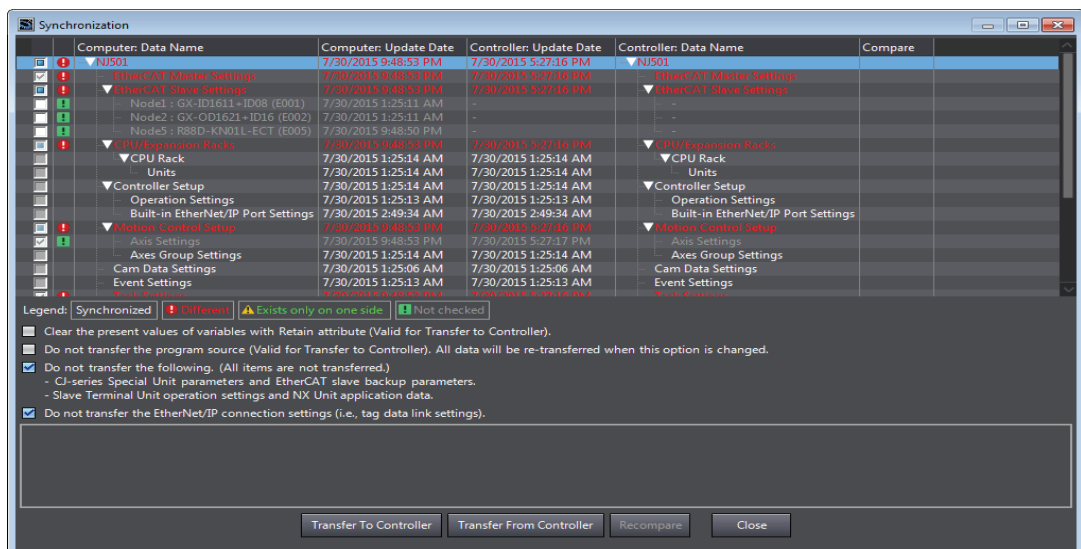
- Synchronizing and transferring the data

- 1 Establish an online connection between the computer and the Controller and then select **Synchronization** from the Controller Menu. (Or, click the  Button on the Toolbar.)



The Synchronization Window is displayed, and comparison of the user program and parameter settings between the Sysmac Studio and the Controller is started.


- 2 The following Uploading and Downloading Data Window is displayed after the automatic comparison.



- 3 Clear the **Do not transfer the EtherNet/IP connection settings (i.e., tag data link settings)** Check Box and then click the **Transfer To Controller** Button.

Then the EtherNet/IP connection settings are transferred along with the not-synchronized data. If no EtherNet/IP connection settings are set in the Sysmac Studio, no data will be sent.

- Transferring all data

1 Establish an online connection between the computer and the Controller and then select **Transfer - To Controller** from the Controller Menu. (Or, click the  Button on the Toolbar.)

2 The **Transfer to Controller** Dialog Box is displayed.

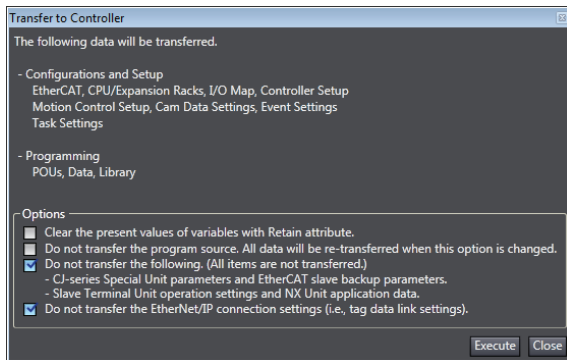
Clear the selection of the *Do not transfer the EtherNet/IP connection settings (i.e., tag data link settings)* Check Box then click the **Execute** Button.



Precautions for Correct Use

To transfer only the connection settings, execute Transfer from the EtherNet/IP Connection Setting Tab Page.

The connection settings are not transferred from the Synchronization Window, the Transfer to Controller Dialog Box, and the Transfer from Controller Dialog Box, even if you clear the *Do not transfer the connection setting* Check Box, as long as the data in the computer and in the Controller is the same.



● **Transferring Only the EtherNet/IP Connection Settings**

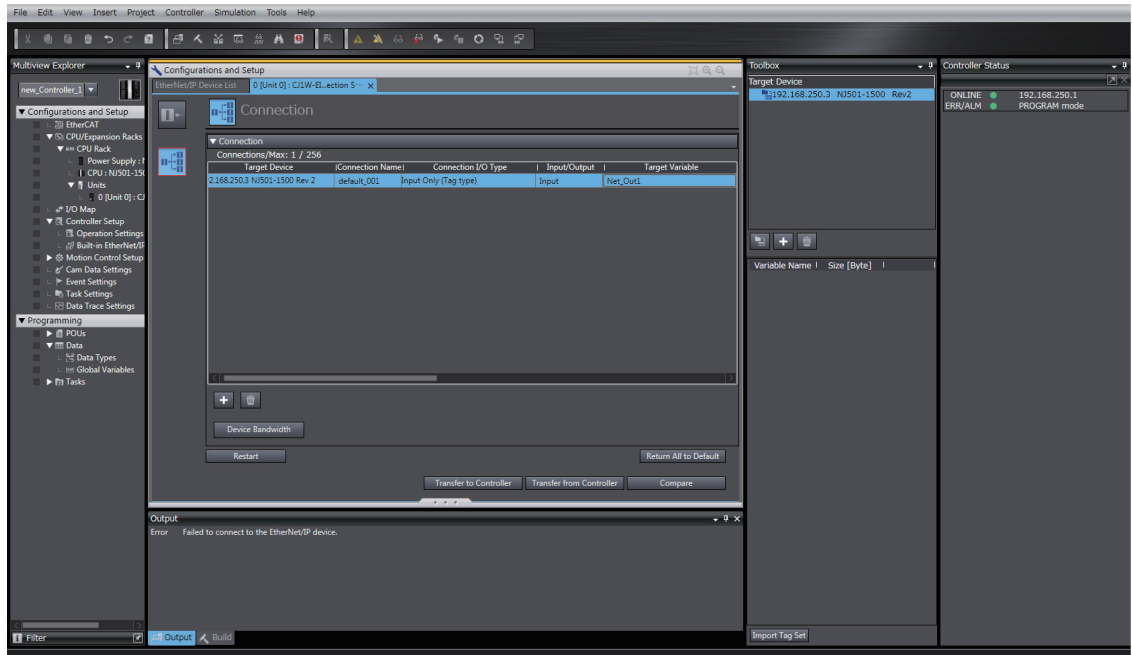
You can transfer tag sets and connections to the EtherNet/IP devices.

1 Establish an online connection with the Controller.

2 Click the **Transfer to Controller** or **Transfer from Controller** Button in the EtherNet/IP Connection Setting Tab Page.

The tag settings and connection settings set at that time are transferred to the Controller connected online.

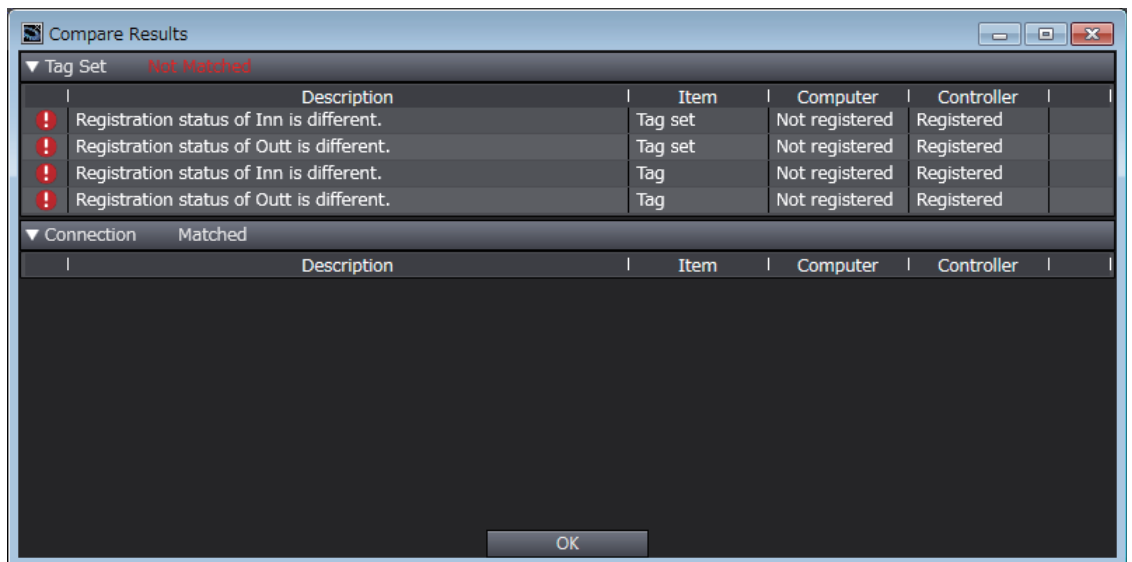
- 3 If the Controller connected online is in RUN mode, the dialog box to confirm whether to switch to PROGRAM mode before transferring the settings is displayed.



● Comparison

The differences in the tag set and connection settings between the project and the EtherNet/IP devices can be displayed.

- 1 Click the **Compare** Button in the EtherNet/IP Connection Setting Tab Page.



Starting and Stopping EtherNet/IP Connections

● Automatically Starting EtherNet/IP Connections

The EtherNet/IP device is automatically restarted and EtherNet/IP connections are automatically started immediately after the connection settings are downloaded from the Sysmac Studio.



Precautions for Correct Use

Connections are adversely cut off if any of the following errors occurs in the Controller that is the originator while EtherNet/IP connections are active.

- Major fault level Controller error
 - Partial fault level Controller error
-

● Starting and Stopping the EtherNet/IP Connections for the Entire Network

You can start and stop EtherNet/IP connections from the user program or from the Sysmac Studio.



Precautions for Correct Use

Use the same method (i.e., either the user program or the tool software) to both start and stop EtherNet/IP connections.

For example, if you use the *_EIP_TDLINKStopCmd* (Tag Data Link Communications Stop Switch) system-defined variable to stop EtherNet/IP connections, you cannot start them from the Sysmac Studio and the Network Configurator.

A-2-5 Checking Communications Status with the Sysmac Studio and Troubleshooting

You can monitor the communications status of the EtherNet/IP connections after their settings are set. You can also check errors.



Precautions for Correct Use

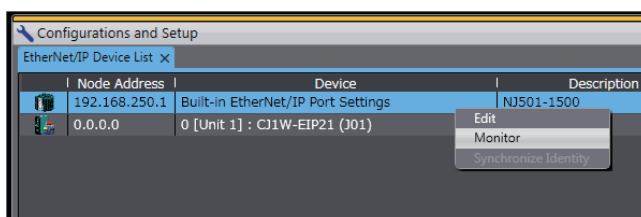
Make sure that the connection settings in both the Sysmac Studio and the Controller are consistent before using the monitor functions. You can use the *Comparison* on page A-31 to see if they are the same.

Checking Communications Status with the Sysmac Studio

You can check the communications status on the EtherNet/IP connections in the EtherNet/IP Connection Monitor Tab Page.

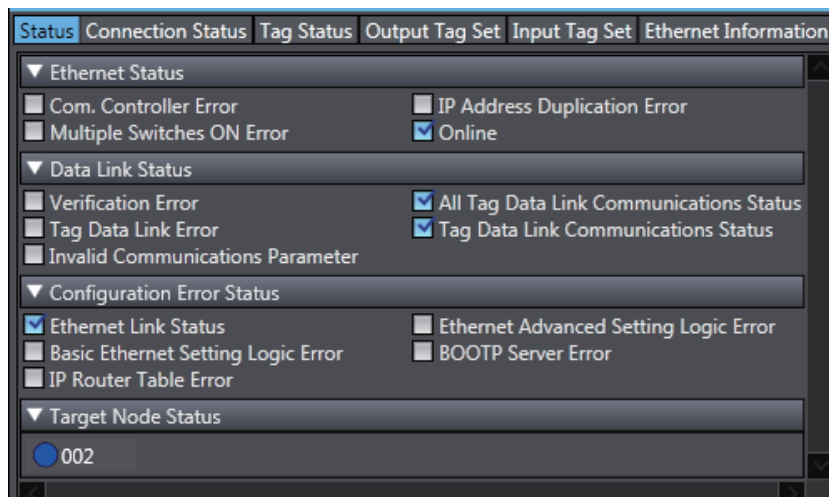
- 1** Select **EtherNet/IP Connection Settings** from the Tools Menu to display the EtherNet/IP Device List Tab Page.
- 2** Right-click the Controller you want to check the communications status and select **Monitor** from the menu.

The pane to monitor the EtherNet/IP connection is displayed. This pane has six tabs for each communications status.

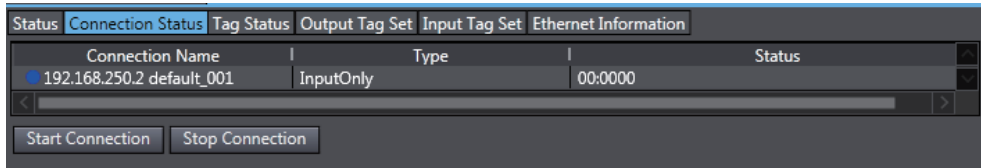


- 3** Select one of the six tabs for which you want to confirm the communications status.
 - Status Tab Page

This tab page gives the TRUE/FALSE status of the system-defined variables that monitors the tag data link errors and communication status. If any of the variables are TRUE, the checkbox in front of the variable will be selected. Refer to *14-1-1 Starting the Device Monitoring* for details on each status item.

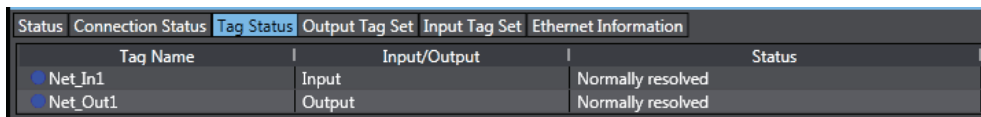


- Connection Status Tab Page
Current status of each connection is given.



Name	Description
Connection Name	Gives the current status of each connection with the following text colors. Blue: Normal Red: There is at least one connection that has not been established. Gray: There are no connections or the connection operation is stopped.
Type	Gives the connection type.
Status	Gives the current status on each connection with codes. <ul style="list-style-type: none"> • Normal operation: 00:0000 • Abnormal operation: Gives an error code. This information can be used to identify the cause of EtherNet/IP connection errors. Refer to <i>14-2 Connection Status Codes and Troubleshooting</i> on page 14-9 for details on the connection status.

- Tag Status Tab Page
This tab page gives if the tag settings for each tag for EtherNet/IP connections are set so that data can be exchanged with target devices.



Name	Description
Tag Name	The current status of each tag is indicated by its color. Red: Tag name resolution error Blue: Tag name resolution normal Gray: Not yet transferred (no information in device).
Input/Output	Gives the type of the tag.
Status	The following status is displayed depending on the status that is set. <ul style="list-style-type: none"> • Normally resolved: Normal data exchange is possible. • Different sizes: Different sizes are set for the network variables and the tag settings. A connection will not be established for a tag for which this error occurs. • No tag: A network variable is not set in the variable table in the Controller for the specified tag setting. A connection will not be established for a tag for which this error occurs. • Attribute error: The following two factors cause this error. <ol style="list-style-type: none"> 1. Writing is not possible for Constant attributes. 2. The I/O direction that is set in the tag data link settings does not agree with the I/O direction of the variable in the Controller. There is an error in the setting of a Network Publish attribute for a Controller variable. A connection will not be established for a tag for which this error occurs.

- Output Tag Set and Input Tag Set Tab Pages

You can monitor the status of each input/output tag set that is used for the EtherNet/IP connections.

Note The tag set status monitor is not available for a built-in EtherNet/IP port on NJ-series Controller version 1.08 or earlier.

Click ▼ of each tag to display its detailed information.

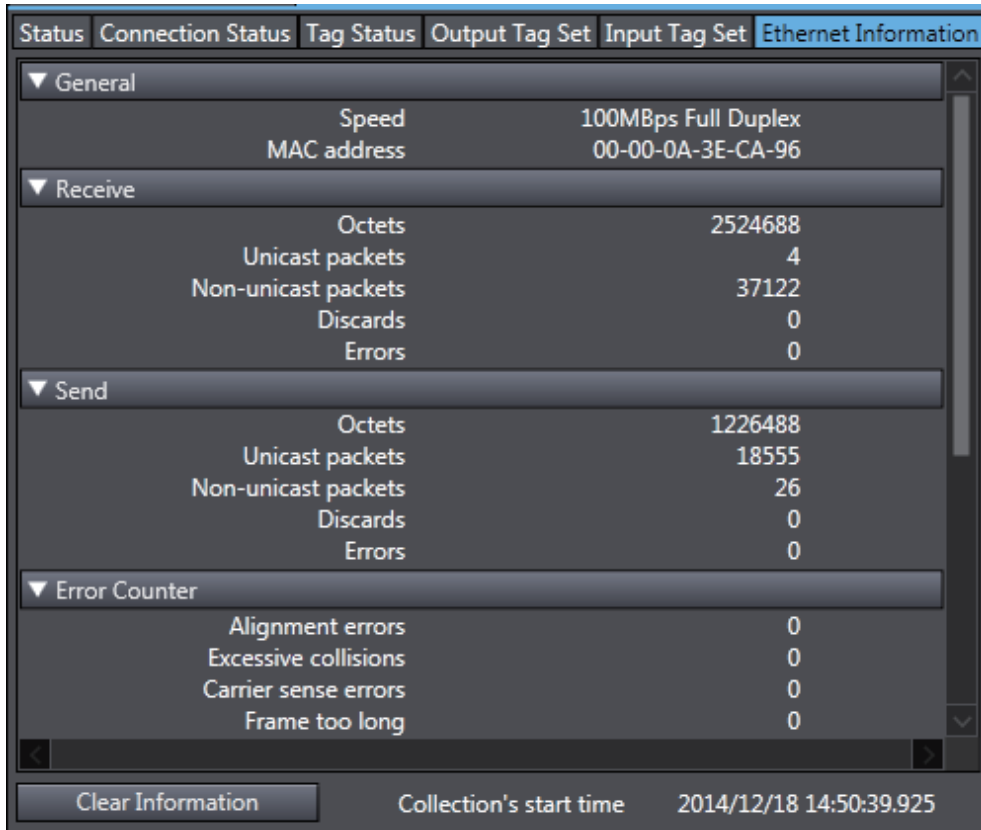
Status	Connection Status	Tag Status	Output Tag Set	Input Tag Set	Ethernet Information
		Tag Set Name			Monitor Value
		▼ TagSetin001			Normal operation
		Tag set size			2
		Connected time			1790973 ms
		Unconnected time			0 ms
		Destination IP address			192.168.250.2
		▼ Target list			
		▼ Target name			
		Remote IP address			192.168.250.2
		O->T RPI (packet interval)			100.0 ms
		T->O Heartbeat transmission cycle [ms]			50.0 ms
		O->T Timeout			400.0 ms
		T->O Timeout			200.0 ms
		O->T API (actual packet interval)			100.0 ms
		T->O Actual heartbeat transmission cycle [ms]			50.0 ms
		O->T Connection ID			0x5E860081
		T->O Connection ID			0x5E8600A1

Name	Description
Tag Set Name	Gives the connection status. If there is a connection error, "Not connected or error" is given.
Tag set size	Gives the size of the tag set in bytes.
Connected time	Gives the total connection duration in milliseconds.
Unconnected time	Gives the total disconnection duration in milliseconds.
Number of connections (in the Output Tag Set Tab Page)	Gives the number of connections.
Number of connected originators (in the Output Tag Set Tab Page)	Gives the number of the connected originator devices.
Originator list (in the Output Tag Set Tab Page), Target list (in the Input Tag Set Tab Page)	Gives the detailed information list of the connected originators.
Originator name (in the Output Tag Set Tab Page), Produced tag name (in the Input Tag Set Tab Page)	Gives no information.
IP address (in the Output Tag Set Tab Page), Remote IP address (in the Input Tag Set Tab Page)	Gives the IP addresses allocated for the originators.
Connected time (in the Output Tag Set Tab Page)	Gives the total duration of connection with the originator in milliseconds.
Unconnected time (in the Output Tag Set Tab Page)	Gives the total duration of disconnection with the originator in milliseconds.

Name	Description
Destination IP address (in the Output Tag Set Tab Page)	Gives the destination IP addresses. If the multi-cast connections are used, its own multi-cast address is displayed.
O->T RPI (packet interval)	Gives the RPI of connection from the originator to the target in milliseconds.
T->O Heartbeat transmission cycle (ms)	Gives the heartbeat transmission period of the connections from the target to the originator in milliseconds.
O->T Timeout	Gives the timeout time for the connections from the originator to the target in milliseconds.
T->O Timeout	Gives the timeout time for the connections from the target to the originator in milliseconds.
O -> T API (actual packet interval)	Gives the API of connection from the originator to the target in milliseconds.
T->O Actual heartbeat transmission cycle (ms)	Gives the actual heartbeat transmission period of the connections from the target to the originator in milliseconds.
O->T Connection ID	Gives the connection identification for the connections from the originator to the target in hexadecimal.
T->O Connection ID	Gives the connection identification for the connections from the target to the originator in hexadecimal.

- Ethernet Information Tab Page

This tab page displays the communications status at the communications driver level of the built-in EtherNet/IP port. The error counter information can be used to confirm whether communications problems have occurred. Under the Tag Data Link, you can confirm characteristics such as the bandwidth usage (PPS).



Display example for an NY-series Controller

A-2-6 Troubleshooting

In the case that there is a setting error or a communications error in the EtherNet/IP networks, the Sysmac Studio displays the error in the Troubleshooting Dialog Box.

Refer to *NYseries Troubleshooting Manual* (Cat. No.W564) for the confirmation methods for errors and information on errors.

Troubleshooting When Transferring and Monitoring the EtherNet/IP Connection Settings Fail

The first time you establish an online connection between the Controller and the computer with Windows Firewall enabled, the dialog box to confirm the connection may be displayed. If that occurs, make the following selection in the dialog box.

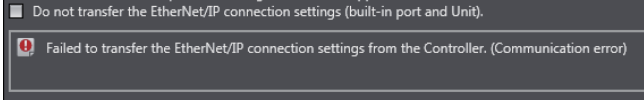
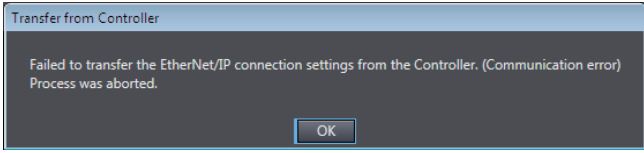
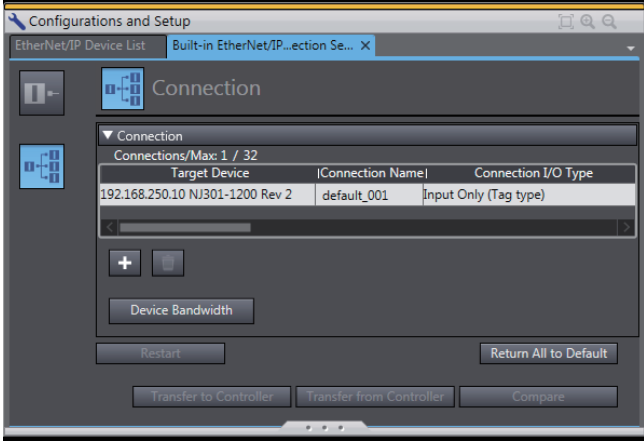
- Unblock (on Windows XP/Vista)
- Allow access (on Windows 7 higher)

If you make other selections than above, there are cases when transferring and monitoring the EtherNet/IP connection settings cannot properly be performed even if the online connection is successfully established.

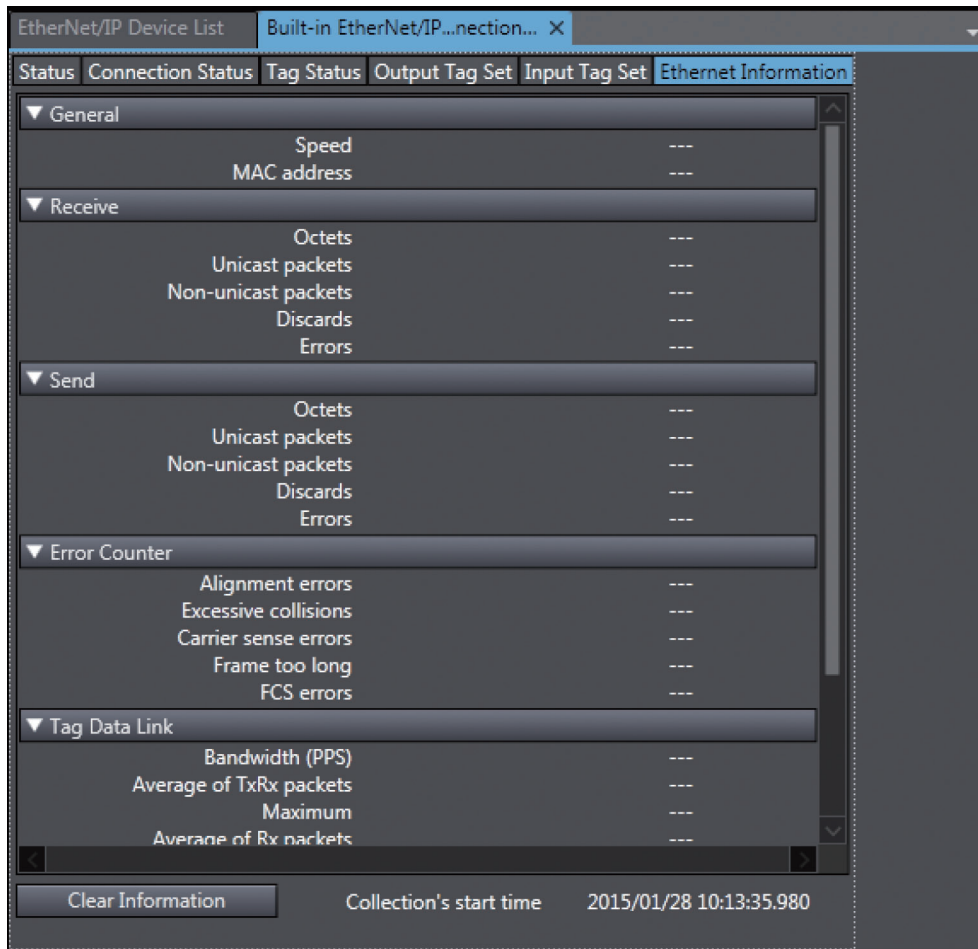
If the above problem occurs, take the following corrective method 1 or 2.

● **Problems**

- The connection setting data cannot be transferred

Data Transmission Screen	Problem
Synchronization Window	<p>The Sysmac Studio displays the following error message and the data will not be transferred.</p> 
Transfer to Controller Dialog Box	<p>The Sysmac Studio displays the following error dialog box and the data will not be transferred.</p> 
EtherNet/IP Connection Setting Tab Page	<p>The Transfer to Controller and Transfer from Controller Buttons are grayed out and the data cannot be transferred/compared.</p> 

- Monitoring the settings cannot be performed
Monitor data items in the EtherNet/IP Connection Monitor Tab Page remain "---".



● **Method 1: Disabling Windows Firewall Settings**



Precautions for Correct Use

The main function of the firewall is to prevent unwanted access from external sources (e.g., the Internet).

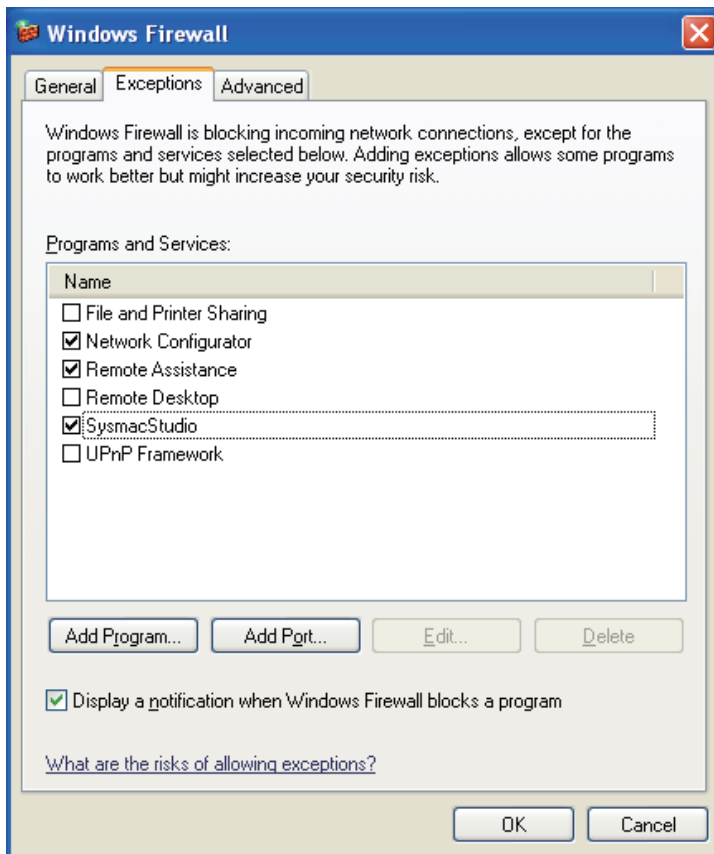
The changes that are made with the following procedures are to allow the Sysmac Studio and the NY-series Controller to connect. If your computer is on an inhouse network, make sure that security will not be jeopardized before you change the settings.

- Windows XP

1 Open the **Control Panel** from the **Windows Start Menu** and then select **Windows Firewall** icon.

The **Windows Firewall** Dialog Box is displayed.

2 Click on the **Exceptions** tab and select **Sysmac Studio** in the **Programs and Services** list.



- Windows Vista, Windows 7, or later version

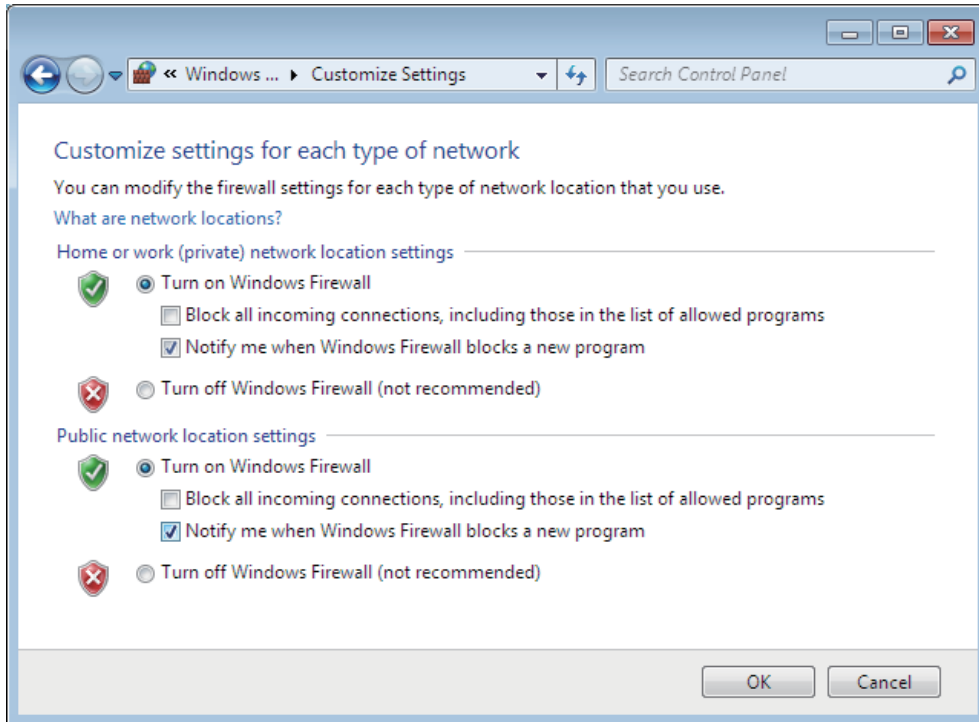
1 Open the **Control Panel** from the **Windows Start Menu** and then select **Windows Firewall** icon.

The **Windows Firewall** Dialog Box is displayed.

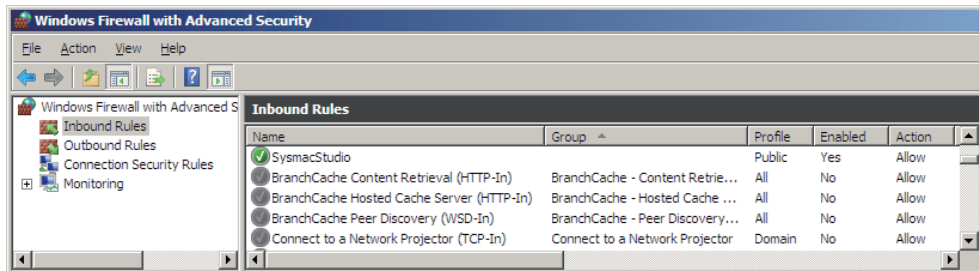
2 Select Turn Windows Firewall on or off.

The **Customize Settings** Dialog box is displayed.

- 3 Clear the **Block all incoming connections, including those in the list of allowed programs** Check Box and click the OK Button.



- 4 Select Advanced settings in the Windows Firewall Dialog Box.
The **Windows Firewall with Advanced Security** Dialog Box is displayed.
- 5 Click **Inbound Rules** in the left pane and then double click **SysmacStudio** in the **Inbound Rules** list.
The SysmacStudio Properties Dialog Box is displayed.



- 6 In the **General** Tab Page of the dialog box, set the following settings.
Select Enabled under the General section.
Select Allow the connection under the Action section.

- **Method 2: Cycle the power supply to the Controller**

Cycle the power supply to the NY-series Controller and transfer/monitor the EtherNet/IP connections settings again.

Note You may need to cycle the power supply when reflecting the changes in the IP address of the built-in EtherNet/IP port or executing Transfer to Controller.

A-3 EDS File Management

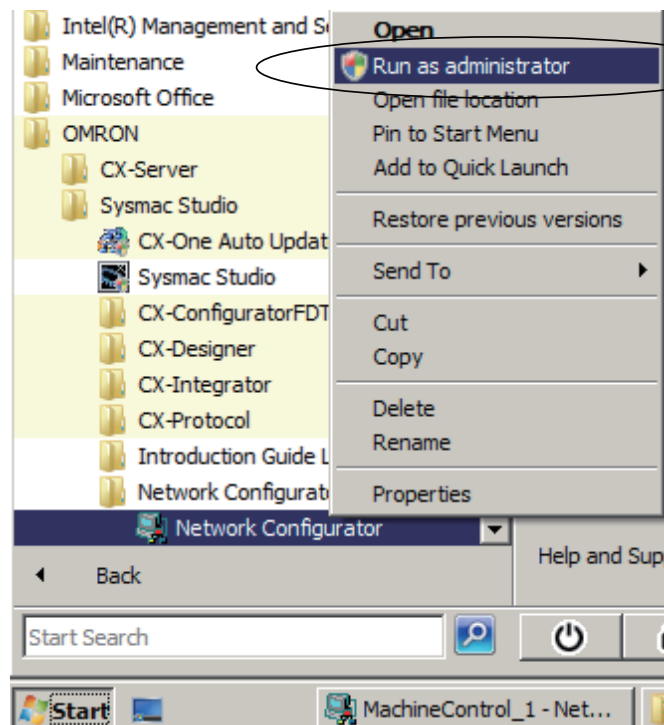
This section describes the EDS file management functions used in the Network Configurator.



Precautions for Correct Use

For Windows Vista or Windows 7, we recommend that you start the Network Configurator as the administrator. Otherwise, the following condition will result due to user management for Windows security functions. The results of the following operations are not applied for logins with other user accounts and must be repeated: installing, creating, and deleting EDS files, and creating EDS index files. You can run the Network Configurator as the administrator with the following procedure.

1. Select the **Network Configurator** from the **Start Menu**, and then right-click.
2. Select **Run as administrator** from the pop-up menu that is displayed.



A-3-1 Installing EDS Files

EDS File - Install

The Network Configurator can support new devices if the proper EDS files are installed.

To install the EDS file, use the following procedure.

- 1** Select **EDS File - Install**.
The Install EDS File Dialog Box is displayed.
- 2** Select the EDS file to install and click the **Open** Button. Next, select the icon file (*.ico). The EDS file is added to the Hardware List as a new device. If the hardware already exists, the new Hardware List will overwrite the previous one. If the hardware versions are different, a hardware device is added to the Hardware List for each version.

A-3-2 Creating EDS Files

EDS File - Create

The EDS files are required by the Network Configurator to create a network configuration. To create an EDS file, use the following procedure.

- 1** Select **EDS File - Create**.
- 2** Set the device information. You can obtain the device information from the device on the network if the network is online.
- 3** The device is added to the Hardware List as a new device, just like when you install an EDS file.



Additional Information

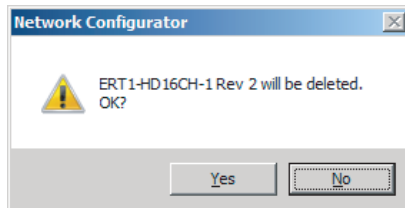
You cannot set device parameters with the Network Configurator's EDS file creation function. Obtain a proper EDS file from the manufacturer of the device to make device parameter settings for the device.

A-3-3 Deleting EDS Files

EDS File - Delete

To delete an EDS file, use the following procedure.

- 1 Select the device from the Hardware List.
- 2 Select **EDS File - Delete**.
The following confirmation dialog box is displayed.



- 3 Click the **Yes** Button.
The selected device is deleted from the Hardware List together with the EDS file.

A-3-4 Saving EDS Files

EDS File - Save

To save the EDS file, use the following procedure.

- 1 Select the target hardware device in the Hardware List, and then select **EDS File - Save**.
A Save EDS File Dialog Box is displayed.
- 2 Input the folder and file names and click the **Save** Button.
The EDS file is saved.

A-3-5 Searching EDS Files

EDS File - Find

To search the devices (EDS files) displayed in the Hardware List, use the following procedure.

- 1** Select *EDS file - Find*.
The following dialog box is displayed.



- 2** Input the character string to search for and click the **Find Next** Button.
- 3** When a matching device is found, the cursor moves to that position.
- 4** To quit the search operation, click the **Cancel** Button.



Additional Information

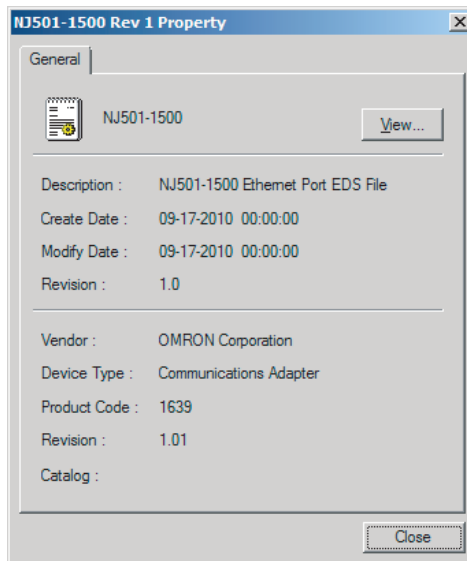
- The device is found only if it is located below the present cursor position in the Hardware List.
- To search all the devices, select *Hardware* in the Hardware List before you perform the search procedure.

A-3-6 Displaying EDS File Properties

EDS File - Property

To display the properties of the EDS file, use the following procedure.

- 1** Select the desired hardware (device) from the Hardware List.
- 2** Select **EDS File - Property**.
The following dialog box is displayed.



The time and date when the EDS file was created is displayed, along with the device information.

A-3-7 Creating EDS Index Files

EDS File - Create EDS Index File

To manually add an EDS file or if a device is not displayed correctly in the hardware list, use the following procedure to recreate the EDS index file. (This applies to Network Configurator version 3.30 or higher.)

- 1** Select **EDS File - Create EDS Index File**.
- 2** Restart the Network Configurator.

A-4 Precautions for Using the Network Configurator on Windows XP, Windows Vista, or Windows 7 or Higher

Better firewall security for Windows XP (SP2 or higher), Windows Vista, and Windows 7 higher has increased the restrictions for data communications. Therefore, you must perform the corresponding procedure given below to change the settings of the Windows firewall before you use the following operations to perform communications with the Network Configurator connected to an NY-series Controller.

- If you select *Option - Select Interface - Ethernet I/F*
- If you select *Option - Select Interface - NJ/NX/NY Series Ethernet Direct I/F*



Precautions for Correct Use

The main function of the firewall is to prevent unwanted access from external sources (e.g., the Internet). The changes that are made with the following procedures are to allow the Network Configurator and the NY-series Controller to connect. If your computer is on an inhouse network, make sure that security will not be jeopardized before you change the settings.

A-4-1 Changing Windows Firewall Settings

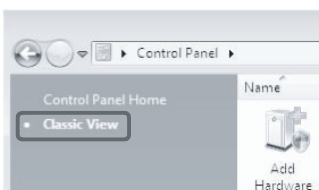
Windows XP

- 1 When you attempt to connect to the NY-series Controller from the Network Configurator, the Windows Security Warning Dialog Box is displayed.
- 2 Click the **Unblock** Button.
An EtherNet/IP connection will be approved for the Network Configurator, and you will be able to connect the Network Configurator in the future.

Windows Vista/Windows 7 or Higher

Use the following procedure to change the settings. Always perform steps 1 to 6 if you cannot go online. The User Account Control Dialog Box may be displayed during this procedure. If it appears, click the **Continue** Button and continue with the procedure.

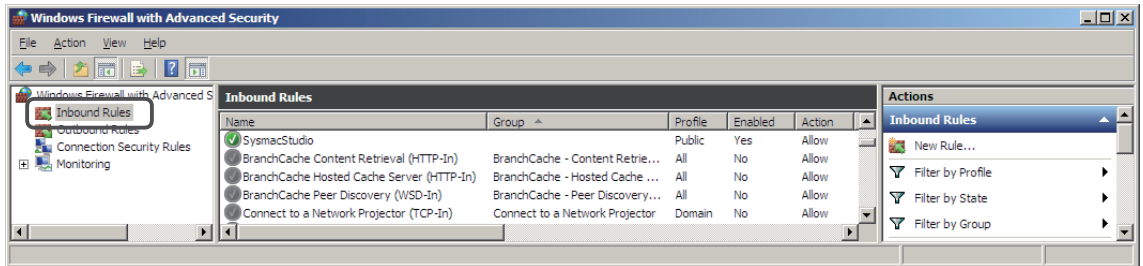
- 1 Select *Control Panel* from the Windows Start Menu and change the display to Classic View.



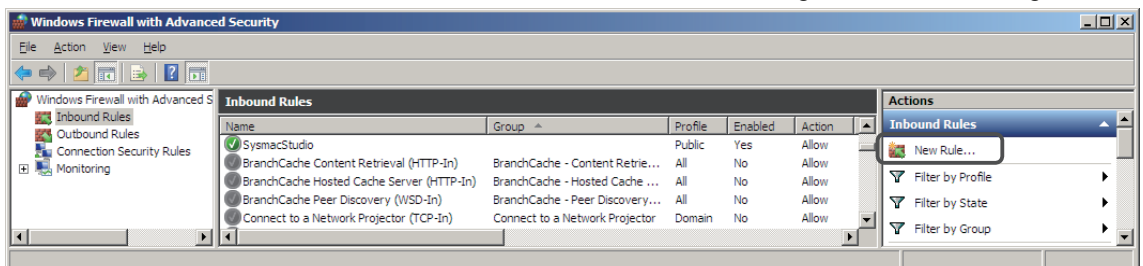
- Open the Administrative Tools and select *Windows Firewall with Advanced Security* from the dialog box that is displayed.



- Select *Inbound Rules* under *Windows Firewall with Advanced Security on Local Computer* on the left side of the Windows Firewall with Advanced Security Dialog Box.



- Select *New Rule* under *Inbound Rules* in the Actions Area on the right side of the dialog box.

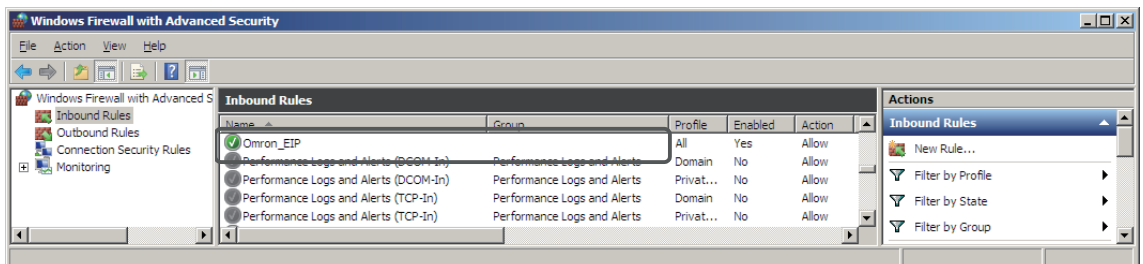


- Make the following settings for each step in the New Inbound Rule Wizard Dialog Box, and click the **Next** Button to move between steps.

Rule Type	Select <i>Custom</i> .
Program	Select <i>All Programs</i> .
Protocol and support	Select <i>ICMPv4</i> as the protocol type. <div style="border: 1px solid gray; padding: 5px; width: fit-content;"> Protocol type: <input type="text" value="ICMPv4"/> Protocol number: <input type="text" value="1"/> </div>
Scope	Select <i>Any IP address</i> for everything.
Action	Select <i>Allow the connection</i> .
Profile	Select <i>Domain, Private, and Public</i> .
Name	Enter any name, e.g., <i>Omron_EIP</i> .

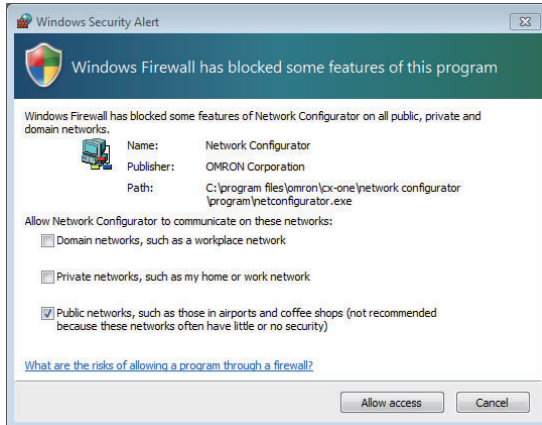
- Click the **Finish** Button. The rule that you defined will be registered in the Inbound Rules (e.g., *Omron_EIP*).

Close the *Windows Firewall with Advanced Security* Dialog Box.



- When you attempt to connect to the NY-series Controller from the Network Configurator, the Windows Security Warning Dialog Box is displayed.

8 Click the **Allow access** Button.



(Windows 7)

An EtherNet/IP connection will be approved for the Network Configurator, and you will be able to connect the Network Configurator in the future.

A-5 Variable Memory Allocation Methods

You must be aware of the way in which memory is allocated to variables to align the memory locations of the members of structure or union variables with variables in other devices. Adjustments are necessary mainly when structure variables are used in the following type of communications with other devices.

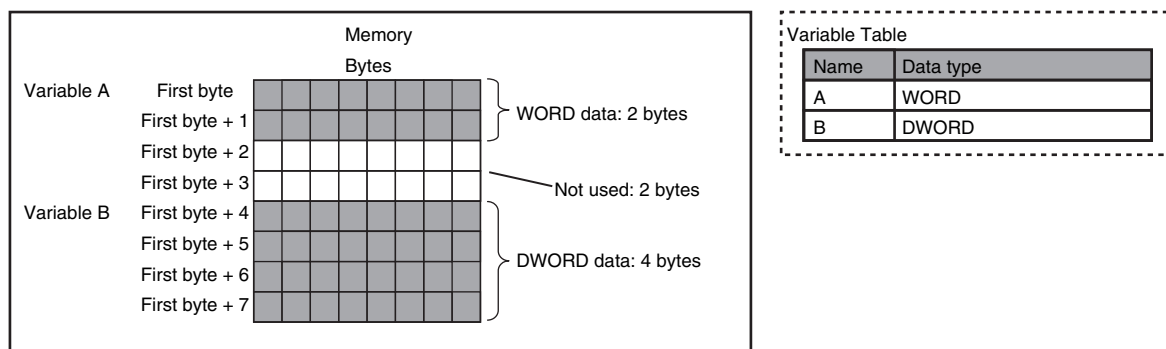
- When using EtherNet/IP tag data links or CIP messages to access variables between NY-series Controllers and other Controllers
- When using structure variables to exchange data with devices other than Controllers, such as ID Tags

A-5-1 Variable Memory Allocation Rules

The amount of memory and the memory locations that are allocated for a variable depend on the data type of the variable. The amount of memory and the memory locations that are allocated for array elements, structure members, and union members depend on the data types, but also on the declarations that are made for the arrays, structures, and unions.

Data Type Alignment and Memory Allocation Amounts

The data size is determined for each data type. The data size is the minimum amount of memory that is required to store the value or values of that data type. On the other hand, memory for variables is automatically structured by the Controller for the most efficient access. Therefore, the total amount of memory that is required for variables is not necessarily the total of the data sizes of the variables. For example, if WORD and DWORD variables are declared, the total of the data sizes is six bytes, but eight bytes are allocated in memory, as shown in the following figure.



This information for determining the location of a variable in memory is called the alignment. The alignment is determined for each data type. The amount of memory and the memory locations for the variables are given below.

Item	Specification
Amount of memory that is allocated	An integral multiple of the alignment. However, the minimum amount of memory is the data size.
Locations in memory	At an integral multiple of the alignment starting from the start of the variable in memory.

The alignments and the amounts of memory that are allocated for the basic data types and enumerations are given below.

Data type	Alignment [bytes]	Amount of memory that is allocated [bytes]
BOOL	2	2
BYTE, USINT, or SINT	1	1
WORD, UINT, or INT	2	2
DWORD, UDINT, or DINT	4	4
LWORD, ULINT, or LINT	8	8
REAL	4	4
LREAL	8	8
TIME, DATE, TIME_OF_DAY, or DATE_AND_TIME	8	8
STRING[N+1] ^{*1}	1	N+1
Enumerations	4	4

*1 N is the maximum number of characters handled. For example, if a maximum of 10 single-byte characters are handled, the NULL character is added, so memory for 11 characters must be reserved.

The elements of arrays and the members of structures and unions are located in memory for the most efficient access. The alignments and the amounts of memory that are allocated for arrays, structures, and unions are determined by the variable declarations, as described below.

Data type	Alignment	Amount of memory that is allocated
Array	Same as alignment of the data type of the elements	(Amount of memory that is allocated for the data type of the elements) × Number of elements [*]
Structure	The largest alignment of all of the members	The integral multiple of the alignment that is larger than the total amount of memory that is allocated when the members are arranged in order at integral multiples of the alignment of the data types of the members
Union	The largest alignment of all of the members	The largest amount of memory that is allocated for any of the members

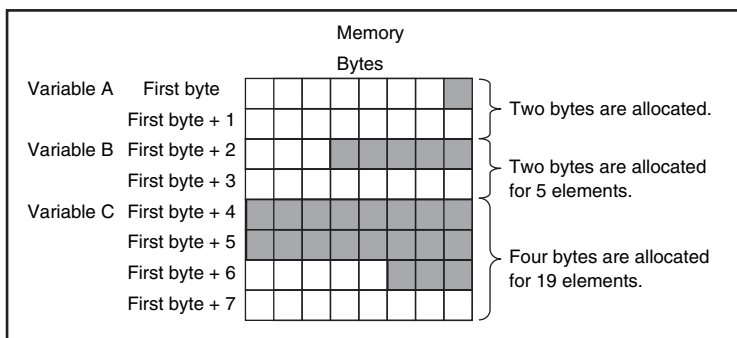
* BOOL arrays are an exception. Refer to *Precautions for Correct Use*, below, for the amount of memory that is allocated for BOOL arrays.



Precautions for Correct Use

Amount of Memory That Is Allocated for BOOL Arrays

Two bytes are allocated in memory for individual BOOL variables, BOOL structure members, and BOOL union variables. However, for a BOOL array, two bytes of memory are not allocated for each element. One bit is allocated in order for each element. For the entire array, a multiple of two bytes of memory is allocated (including unused bits).



Name	Data type
A	BOOL
B	ARRAY[1..5]OF BOOL
C	ARRAY[0..18]OF BOOL

Therefore, the following formula gives the amount of memory that is allocated for a BOOL array. For 1 to 16 elements, 2 bytes are allocated. For 17 to 32 elements, 4 bytes are allocated.

$$\text{Amount of memory} = 2 \left\lceil \frac{\text{Number of elements} - 1}{16} \right\rceil + 2$$

Truncate the decimal portion of the result of the calculation in brackets.

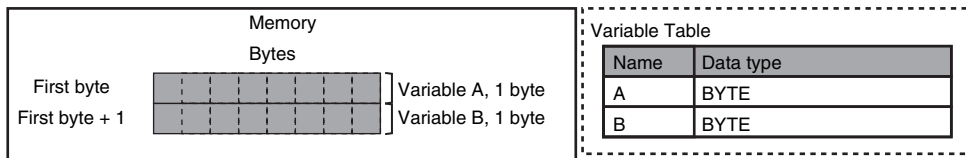
Specific examples of the rules for memory allocation for variables of each data type are given below.

Basic Data Types

● Variables with One-Byte Alignments (e.g., BYTE)

One byte of memory is allocated for the one-byte alignment.

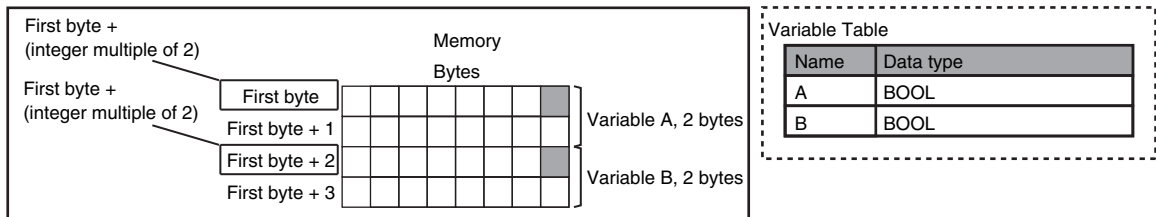
Example: Two consecutive BYTE variables



● Variables with Two-byte Alignments (e.g., BOOL and WORD)

Two bytes of memory are allocated for the two-byte alignment.

Example: Two consecutive BOOL variables

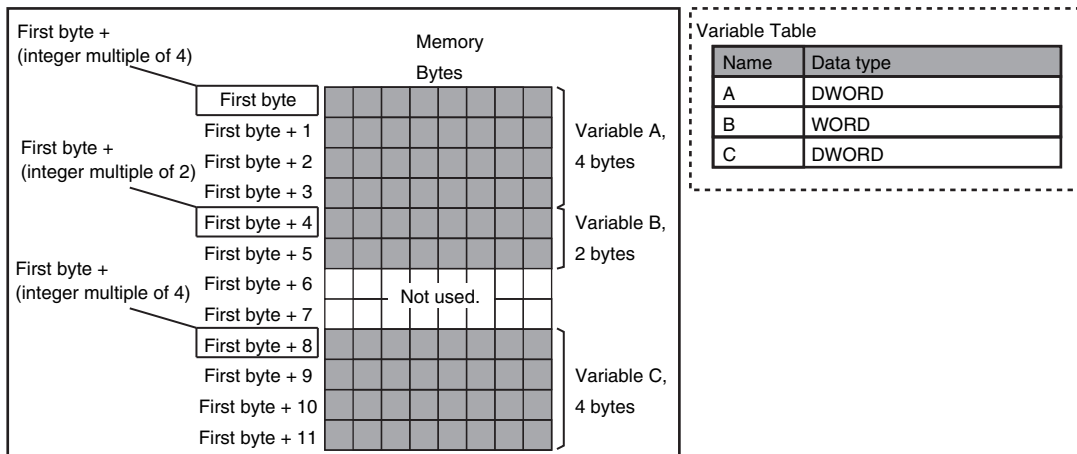


● Variables with Four-byte Alignments (e.g., DWORD)

Four bytes of memory are allocated for the four-byte alignment.

The location of the first byte of data in memory is an integer multiple of four bytes. Therefore, if a variable with a two-byte alignment, such as WORD data, is inserted, two bytes of unused memory will remain.

Example: Consecutive variables in the following order: DWORD, WORD, and DWORD

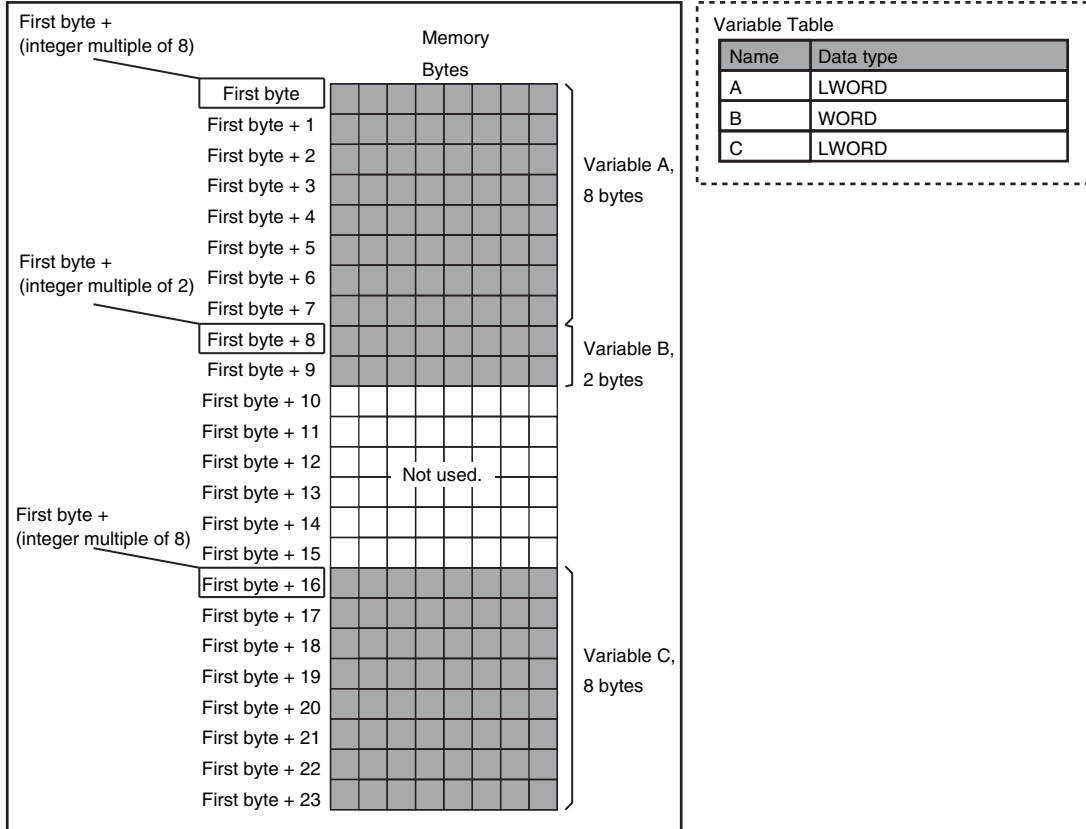


● **Variables with Eight-byte Alignments (e.g., LWORD)**

Eight bytes of memory are allocated for the eight-byte alignment.

The location of the first byte of data in memory is an integer multiple of eight bytes. Therefore, if a variable with a two-byte alignment, such as WORD data, is inserted, six bytes of unused memory will remain. If a variable with a four-byte alignment, such as DWORD data, is inserted, four bytes of unused memory will remain.

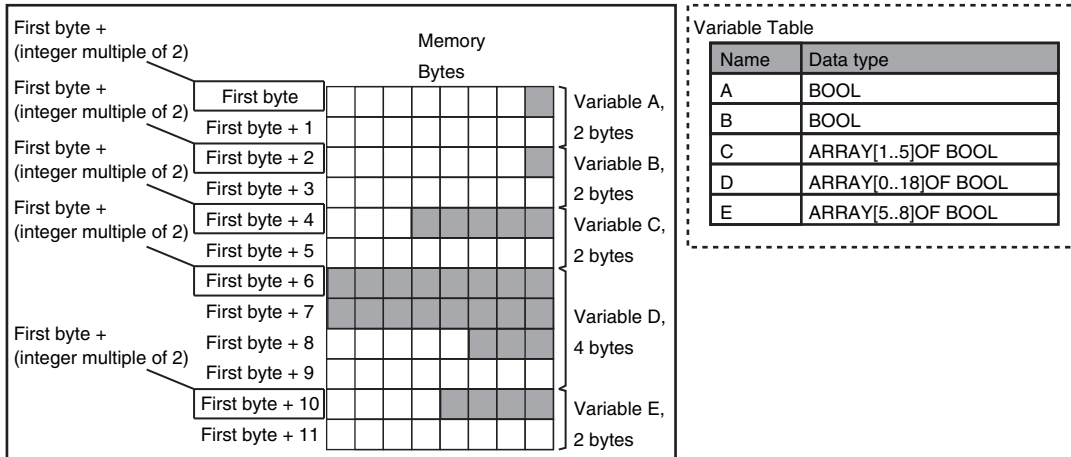
Example: Consecutive variables in the following order: LWORD, WORD, and LWORD



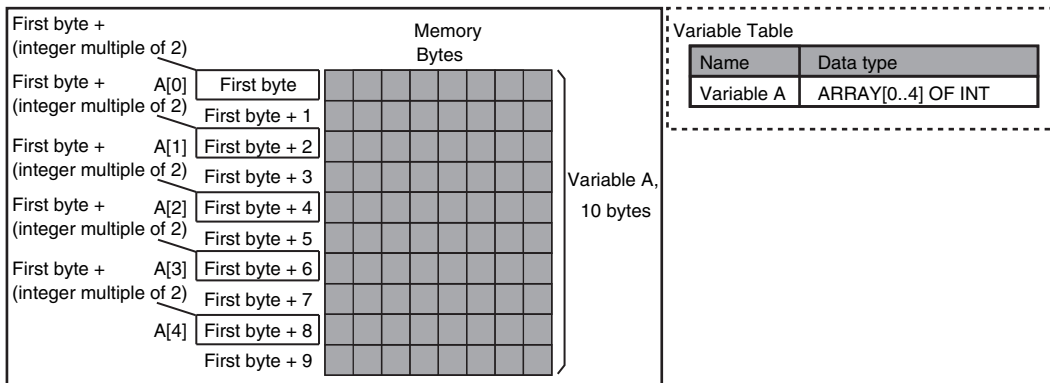
Arrays

A continuous section of memory is allocated for the elements of the array based on the data size of the data type of the array variable. The alignment of an array is the same as alignment of the data type of the elements.

Example: Continuous variables in the following order: two BOOL variable, one BOOL array with five elements, one BOOL array with 19 elements, and one BOOL array with four elements



Example: INT array with five elements

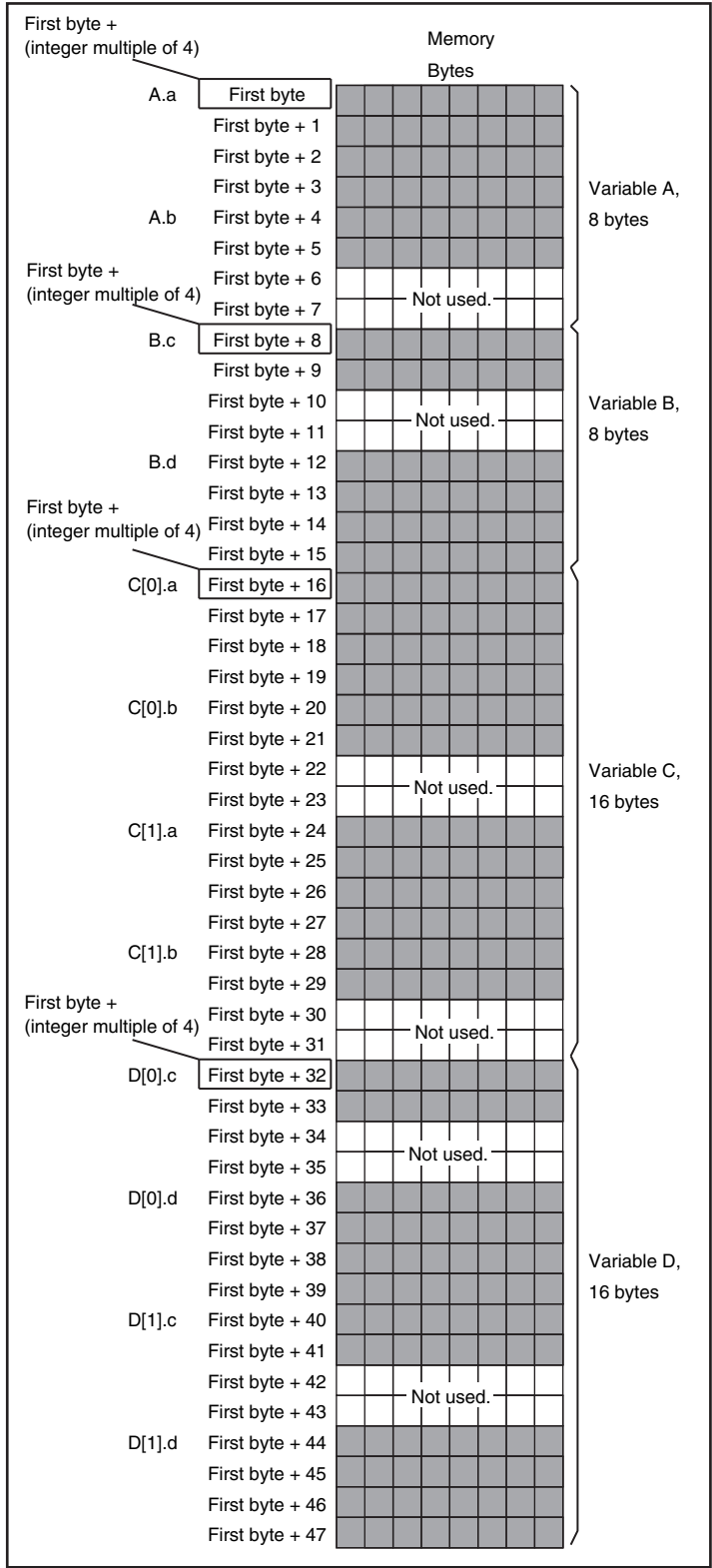


Structures

For a structure variable, the members are located in memory in the order that they are declared. Each member is located at an integer multiple of the alignment of the data type of the member. Therefore, there can be unused memory between members or at the end of members. The alignment of a structure is the largest alignment of all of the members. The amount of memory that is allocated is the integral multiple of the alignment that is larger than the total amount of memory that is allocated when the members are arranged in order at integral multiples of the alignment of the data types of the members.

Example: The alignments and the amounts of memory that are allocated for the four variable declarations given in the following figure are given in the following table.

Variable	Alignment [bytes]	Amount of memory that is allocated [bytes]
A	4	8
B	4	8
C	4	16
D	4	16



Data Type Definitions

Name	Data type
Structure <i>STR_A</i>	STRUCT
a	DINT
b	INT

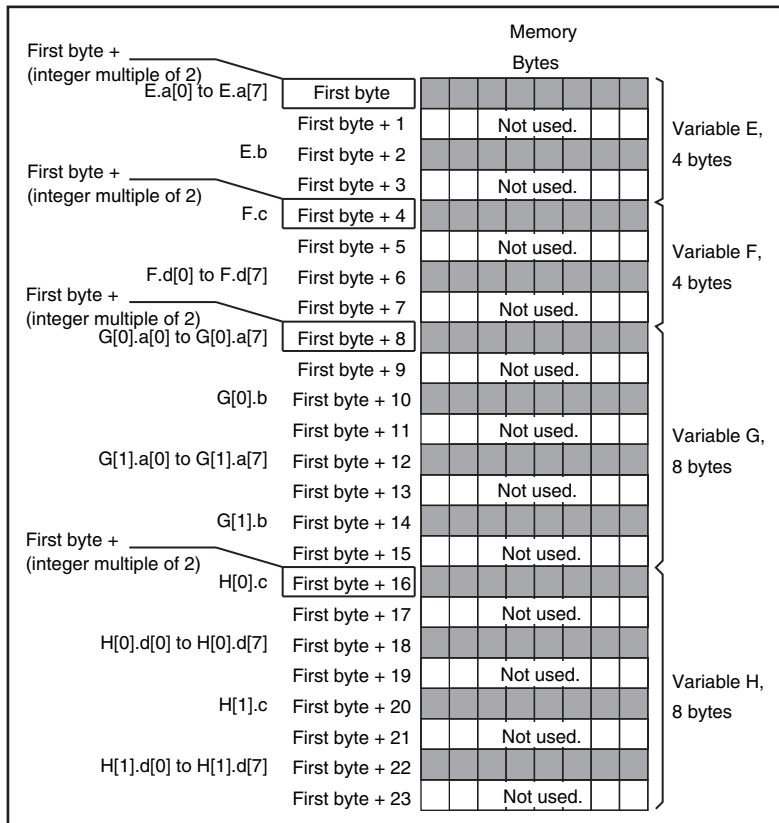
Name	Data type
Structure <i>STR_B</i>	STRUCT
c	INT
d	DINT

Variable Table

Name	Data type
Variable A	Structure <i>STR_A</i>
Variable B	Structure <i>STR_B</i>
Variable C	ARRAY[0..1] OF <i>STR_A</i>
Variable D	ARRAY[0..1] OF <i>STR_B</i>

Example: The alignments and the amounts of memory that are allocated for the four variable declarations given in the following figure are given in the following table.

Variable	Alignment [bytes]	Amount of memory that is allocated [bytes]
E	2	4
F	2	4
G	2	8
H	2	8



Data Type Definitions

Name	Data type
Structure <i>STR_C</i>	STRUCT
a	ARRAY[0..7] OF BOOL
b	BYTE

Name	Data type
Structure <i>STR_D</i>	STRUCT
c	BYTE
d	ARRAY[0..7] OF BOOL

Variable Table

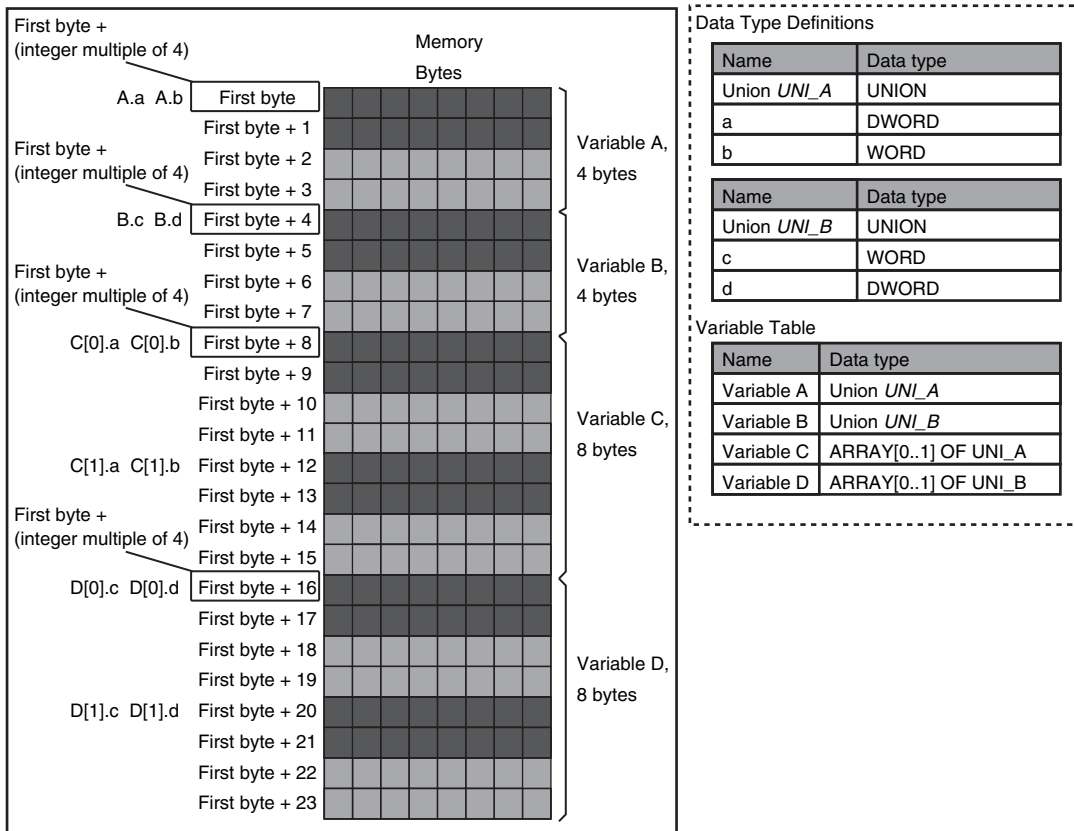
Name	Data type
Variable E	Structure <i>STR_C</i>
Variable F	Structure <i>STR_D</i>
Variable G	ARRAY[0..1] OF STR_C
Variable H	ARRAY[0..1] OF STR_D

Unions

For a union variable, the members overlap in the same memory locations. The alignment of a union is largest alignment of all of the members. The amount of memory that is allocated is the largest amount of memory that is allocated for any of the members.

Example: The alignments and the amounts of memory that are allocated for the four variable declarations given in the following figure are given in the following table.

Variable	Alignment [bytes]	Amount of memory that is allocated [bytes]
A	4	4
B	4	4
C	4	8
D	4	8



A-5-2 Important Case Examples

When you exchange structure variable data between an NY-series Controller and a remote device, you must align the memory configuration of the structure variable members with those of the remote device. This section describes what to do in either the NY-series Controller or in the remote device.



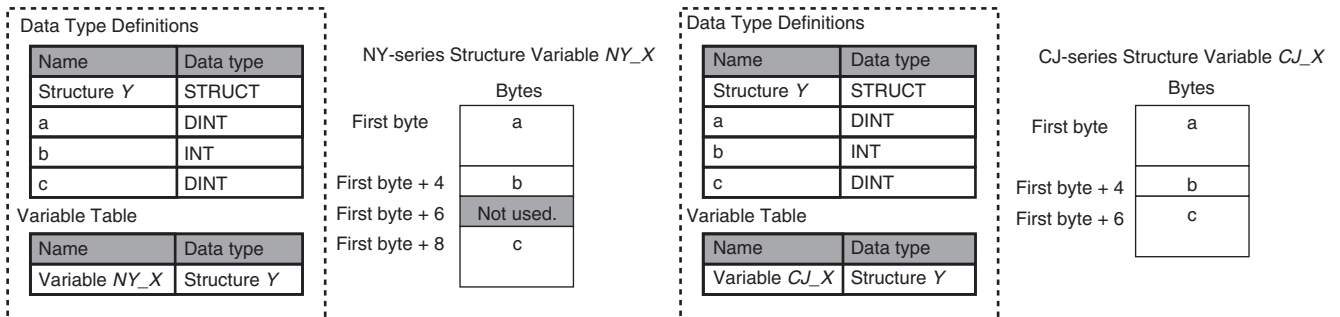
Additional Information

This is not necessary when you exchange data between NY-series Controllers.

Aligning the Memory Configuration with a Remote Device

There are two methods that you can use to align the memory configuration with a remote device. For example, the differences in the memory configuration for structure variables between an NY-series Controller and a CJ-series CPU Unit are shown below.

This section describes how to align the memory configuration for these Units.



● Method 1: Changing the Memory Configuration of the Structure Variable in the NY-series Controller

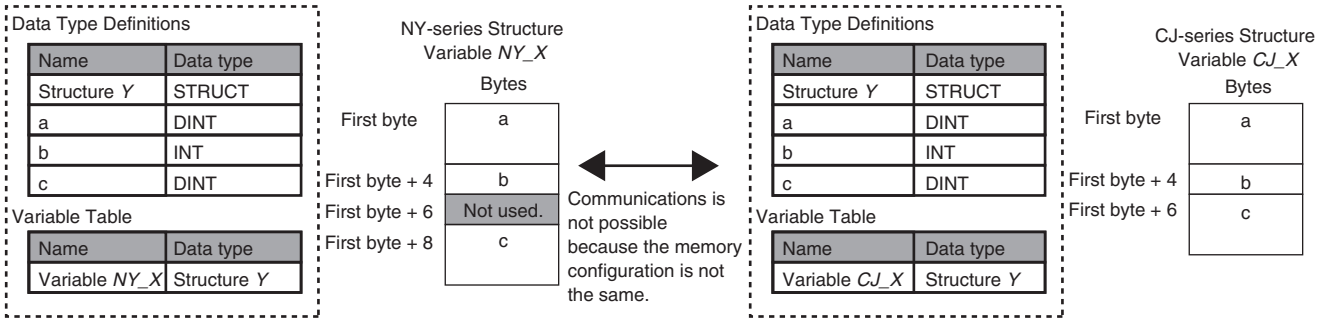
With an NY-series Controller, you can specify member offsets to change the memory configuration of the members of a structure variable. You can change the memory configuration of the members of a structure variable in the NY-series Controller so that it is the same as the memory configuration in a remote device that the Controller will communicate with. Specify the member offsets for a structure variable when you register the structure data type.

To communicate with a CJ-series CPU Unit, you can set the offset type to *CJ* to automatically use the CJ-series memory structure. You can set the offset type to *User* to freely set your own offsets.

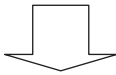
If you change the memory configuration of a structure variable by setting offsets, you must make the same changes for the same structure variable in other NY-series Controllers on the network.

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No W504-E1-03 or higher) for the procedure to change the memory configuration of a structure variable.

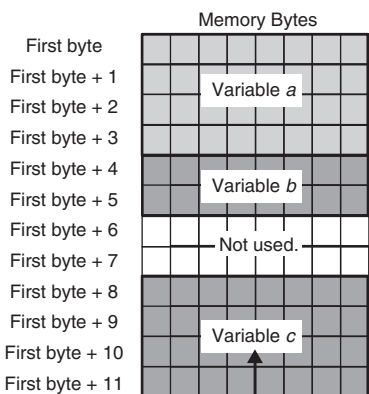
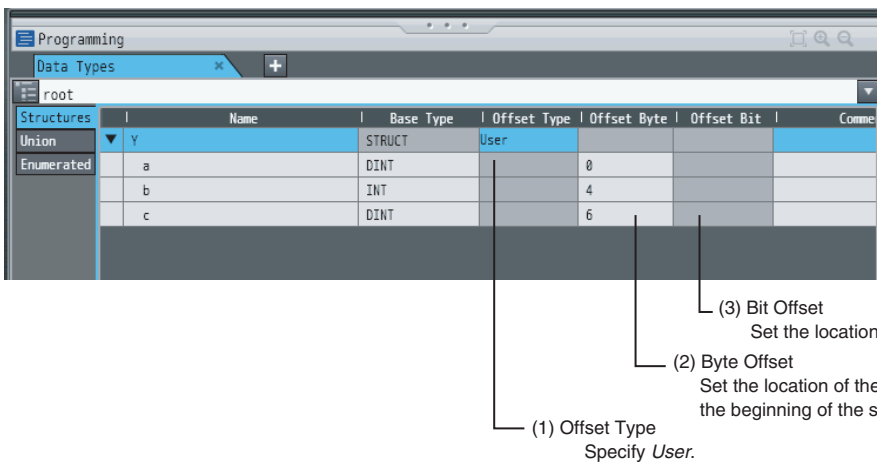
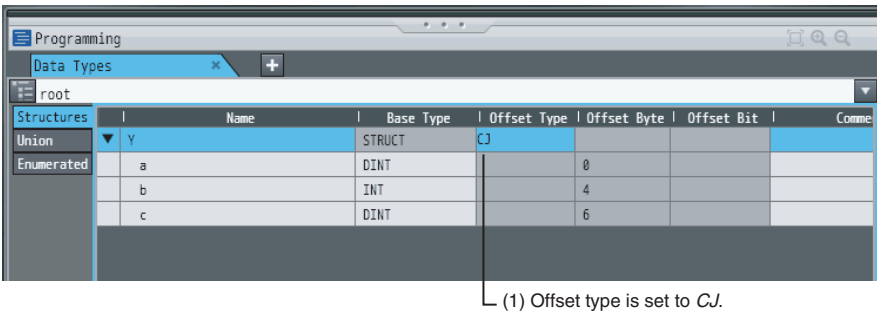
Example: The following example shows how the memory configuration of the structure variable members in the NY-series Controller is changed to match the memory configuration of the structure variable members in the CJ-series CPU Unit.



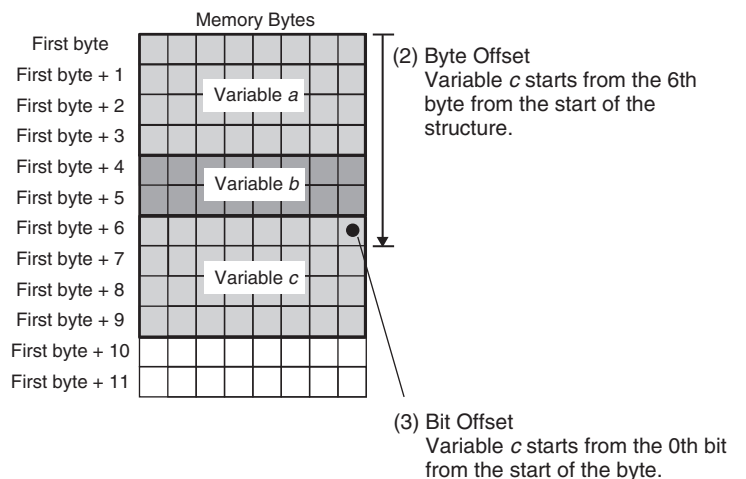
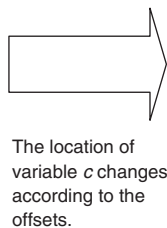
To align the memory configurations in the NY-series Controller CJ-series CPU Units, offsets are set in the Sysmac Studio.



Here, the following offsets are set for member c of data type Y of the structure variable NY_X.



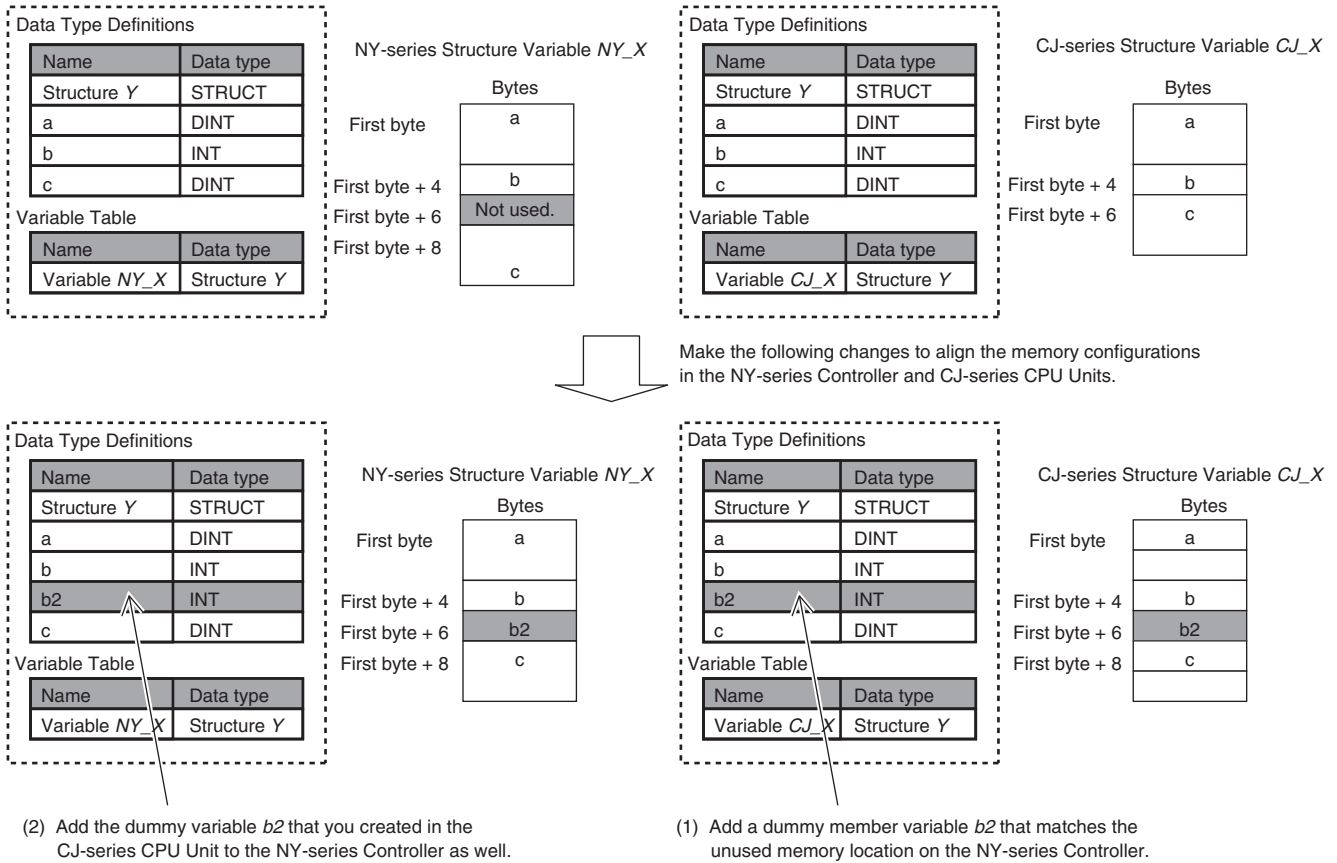
Set a byte offset of 6 and a bit offset of 0 (no offset) for variable c.



● Method 2: Changing the Memory Configuration of the Structure Variable in the Remote Device

You can insert a member into the structure variable of the remote device to change it to match the memory configuration of the structure variable in the NY-series Controller. Both the memory configuration and the data types must be the same between the two structure variables. You therefore need to create the same members in both the remote device and the NY-series Controller.

Example: The following example shows how the memory configuration of the structure variable in the CJ-series CPU Unit is changed to match the memory configuration of the structure variable in the NY-series Controller.



A-6 Precautions When Accessing External Outputs in Controllers

Observe the following precautions when you access variables that are assigned to external outputs in an NY-series Controller.

- **Precaution on Writing from External Devices, Variables* That Are Assigned to External Outputs**

Any value that is written to a variable* that is assigned to an external output in an NY-series Controller through a tag data link or communications instruction will be overwritten by the execution results of the user program.

The value that is written from the tag data link or communications instruction will therefore not be output to the external device.

The following types of variable are assigned to the external outputs.

- The devices variables (or global variables) that are assigned to an I/O port of an EtherCAT output-slave.

A-7 TCP State Transitions

TCP protocol operates in 11 states for connections.

You can check the TCP state of the socket service with the TCP connection status that is output by the SktGetTCPStatus (Read TCP Socket Status) instruction.

The TCP states and their meanings are given in the following table.

TCP state	Meaning
CLOSED	The connection was closed.
LISTEN	The server is waiting for a connection request (SYN) with a passive open.
SYN SENT	The client sent a connection request (SYN) for an active open and is waiting for acknowledgement (SYN + ACK).
SYN RECEIVED	The server sent an acknowledgement (SYN + ACK) to a connection request (SYN) and is waiting for acknowledgement (ACK).
ESTABLISHED	A connection was established.
CLOSE WAIT	The server sent acknowledgement (ACK) to a connection close request (FIN) and is waiting for the server application to be ready to close.
FIN WAIT-1	The client sent a connection close request (FIN) and is waiting for acknowledgement (ACK).
CLOSING	The client and server simultaneously received a connection close request (FIN) and are waiting for acknowledgement (ACK).
LAST-ACK	The server sent a connection close request (FIN) and is waiting for acknowledgement (ACK).
FIN WAIT-2	The client is waiting for a connection close request (FIN).
TIME WAIT	The client received acknowledgement (ACK) to a connection close request (FIN) and is waiting for it to be received and processed by the server.

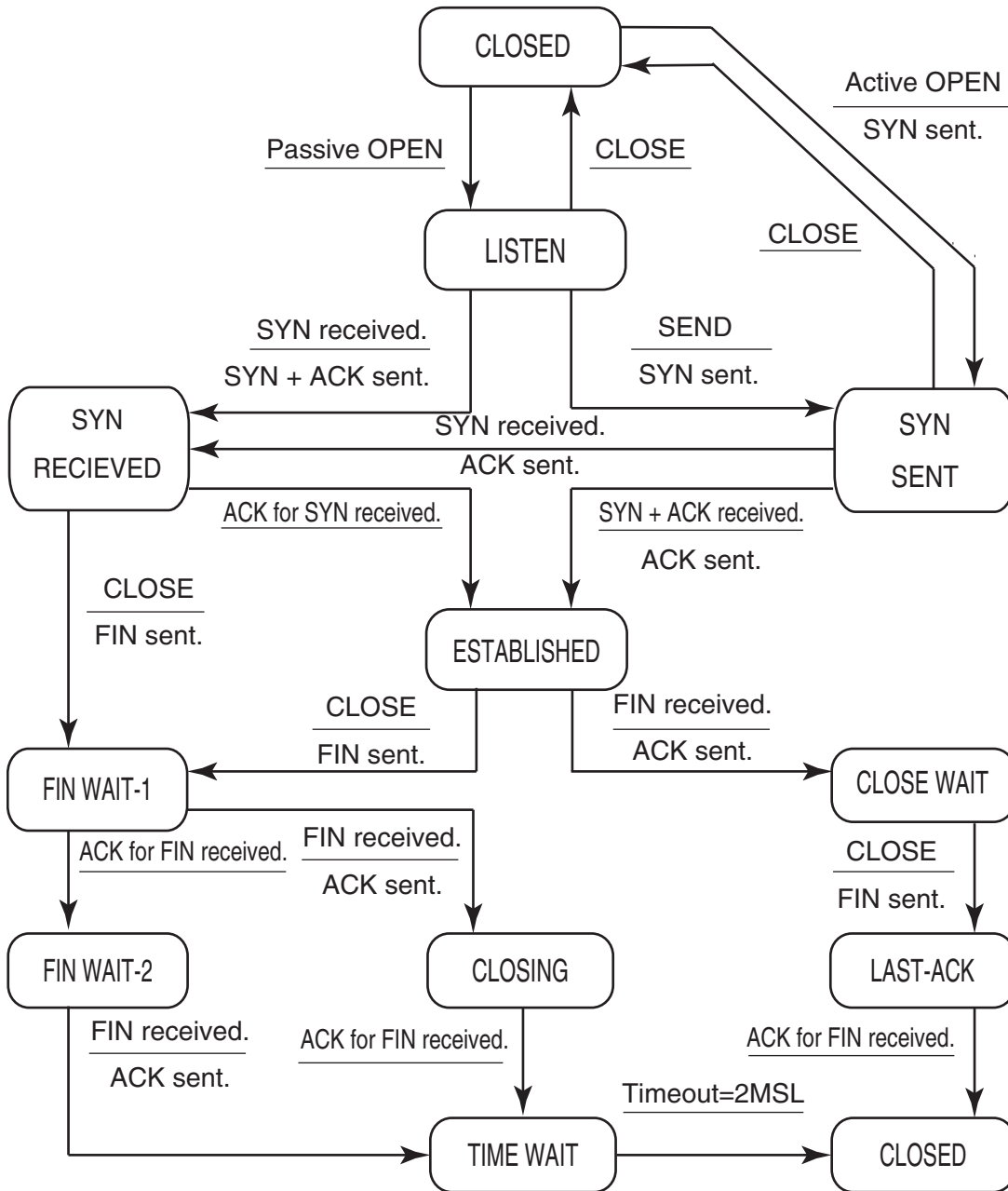
The TCP state changes as requests and acknowledgements are received from the remote node, and as TCP socket connection and close instructions are executed in the user program.

When the state changes, connection requests (SYN), close requests (FIN), and acknowledgements (ACK) to those requests are sent to and received from the remote node.

The following figure shows TCP state transitions.

The TCP states are given in the boxes in the figure. Between the states, the text on top is the condition for a change in state and the text on the bottom is the action that is performed when the state changes. (If no action is given, then none is performed.)

Example: When SYN and ACK are received in SYN SENT state, ACK is sent and the state changes to ESTABLISHED.





Index



Index

A

- Accept TCP Socket instruction 9-13
- addresses 5-8
- All Tag Data Link Communications Status 3-10, 3-21
- array variables
 - preparing array variables to input
 - and output service data and response data 8-14
- Auto Connection Configuration 7-39
- automatically setting connections 7-38

B

- bandwidth usage
 - requested packet intervals 13-3
 - tag data links 13-7
- Basic Ethernet Setting Error 3-5, 3-16
- binary format 10-17
- BOOTP client 1-17
- BOOTP Server Error 3-6, 3-18
- broadcasting 9-10
- built-in EtherCAT port 1-10
- Built-in EtherNet/IP Error 3-3, 3-13
- built-in EtherNet/IP port 1-10
 - settings 5-1
 - socket services
 - overview 9-11
 - specifications 1-7

C

- CIDR 4-4
- CIP Communications 1-12
- CIP Communications Error 3-4, 3-15
- CIP communications instructions 8-5
 - using 8-6
- CIP message communications 8-4
- CIP message communications service
 - overview 8-3
 - specifications 8-3
- Clear TCP/UDP Socket Receive Buffer instruction 9-13
- Close CIP Class 3 Connection instruction 8-5
- Close TCP/UDP Socket instruction 9-13
- Communications Controller Error 3-5, 3-16
- communications load
 - adjusting 13-6
- Communications Port Error 3-3, 3-14
- Communications Port1 Error 3-4, 3-14
- community names 5-9, 5-11
- Connect TCP Socket instruction 9-13
- Connection I/O Type 7-35, 7-36
- Connection Name 7-35, 7-36
- connection settings
 - automatically setting connections 7-38

- batch editing 7-35
- editing individual connections 7-34
- Register Device List 7-32
- Connection Tab Page 14-5
- Connection Type 7-35, 7-36
- connections
 - checking 7-69
- Controller Event Log Tab Page 14-6
- Controller Object 8-52
- Controller status 7-9
- Create UDP Socket instruction 9-13
- cyclic communications 1-12

D

- data areas 7-3
- data processing time
 - calculation example 13-23
 - overview 13-22
- data transmissions
 - timing 13-21
- default gateway 5-4
- destination IP addresses 5-4
- destination mask IP addresses 5-4
- device bandwidth usage
 - adjusting 13-9
- Device Connection Structure Tree 7-41
- device parameters
 - clearing 7-64
 - verifying 7-60
- device status
 - displaying 7-72
- devices
 - changing 7-71
 - registering 7-19
- DNS 5-3
- DNS Server Connection Error 3-8, 3-20
- DNS Setting Error 3-6, 3-18
- domain names 5-3

E

- EDS files
 - management A-43
- _EIP1_BootpErr 3-6, 3-18
- _EIP1_EtnCfgErr 3-5, 3-16
- _EIP1_EtnOnlineSta 3-10, 3-21
- _EIP1_IPAdrCfgErr 3-6, 3-17
- _EIP1_IPAdrDupErr 3-6, 3-18
- _EIP1_LanHwErr 3-5, 3-16
- _EIP1_MacAdrErr 3-5, 3-16
- _EIP1_PortErr 3-4, 3-14
- _EIP_BootpErr 3-6, 3-18
- _EIP_CipErr 3-4, 3-15
- _EIP_DNSCfgErr 3-6, 3-18

_EIP_DNSSrvErr	3-8
_EIP_ErrSta	3-3, 3-13
_EIP_EstbTargetSta	3-10, 3-22, 14-4
_EIP_EtnCfgErr	3-5, 3-16
_EIP_EtnOnlineSta	3-9, 3-21
_EIP_IdentityErr	3-7, 3-19
_EIPIn1_EtnOnlineSta	3-10
_EIPIn1_IPAdrCfgErr	3-6, 3-17
_EIPIn1_IPAdrDupErr	3-18
_EIPIn1_PortErr	3-4, 3-15
_EIP_IPAdrCfgErr	3-5, 3-17
_EIP_IPAdrDupErr	3-6, 3-17
_EIP_IPRTblErr	3-7, 3-18
_EIP_LanHwErr	3-5, 3-16
_EIP_MacAdrErr	3-5, 3-15
_EIP_MultiSwONErr	3-7, 3-20
_EIP_NTPResult	3-11
_EIP_NTPResult.ExecNormal	3-12, 3-23
_EIP_NTPResult.ExecTime	3-12, 3-23
_EIP_NTPSrvErr	3-8, 3-20
_EIP_PortErr	3-3, 3-14
_EIP_RegTargetSta	3-10, 3-22
_EIP_TagAdrErr	3-7, 3-20
_EIP_TargetNodeErr	3-11, 3-23
_EIP_TargetPLCErr	3-11, 3-22, 7-9, 14-4
_EIP_TargetPLCModeSta	3-11, 3-22, 7-9, 14-4
_EIP_TcpAppCfgErr	3-8
_EIP_TcpAppErr	3-5, 3-15
_EIP_TDLinkAllRunSta	3-10, 3-21
_EIP_TDLinkCfgErr	3-7, 3-19
_EIP_TDLinkErr	3-7, 3-19
_EIP_TDLinkOpnErr	3-7, 3-19
_EIP_TDLinkRunSta	3-10, 3-21
_EIP_TDLinkStartCmd	3-12, 3-23
_EIP_TDLinkStopCmd	3-12, 3-23
_EIP_TopAppCfgErr	3-20
error status	3-9
Ethernet addresses	1-10
Ethernet connectors	2-10
Ethernet Information Tab Page	14-8
Ethernet Link Object	8-47
Ethernet switches	1-5, 2-2
functions	2-3
selection precautions	2-4
types	2-3
F	
fragmenting	
send data	9-5
FTP commands	
append	10-13
bye	10-15
cd	10-13
close	10-15
delete	10-15
dir	10-12
get	10-14
ls	10-11
mdelete	10-15
mdir	10-12
mget	10-14
mkdir	10-12
mls	10-11
mput	10-14
open	10-11
put	10-14
pwd	10-13
quit	10-15
rename	10-12
rmdir	10-13
type	10-13
user	10-11
FTP Display	5-7
FTP server	1-17, 5-7
application example	10-8
application example from host computer	10-18
application procedure	10-7
commands	10-10
overview	10-2
specifications	10-2
functions	
functional comparison with other series	A-2
G	
gateway addresses	5-4
general status	8-30
general status codes	8-28
global addresses	4-11
global broadcasting	9-10
H	
host names	5-3, 5-9, 5-11
specifying	1-19
I	
Identity Error	3-7, 3-19
Identity Object	8-40
indicators	1-10, 1-11
input ON response time	13-25
instructions	
CIPClose	8-5
CIPOpen	8-5
CIPOpenWithDataSize	8-5
CIPRead	8-5
CIPSend	8-5
CIPUCMMRead	8-5
CIPUCMMSend	8-5
CIPUCMMWrite	8-5
CIPWrite	8-5
SkClearBuf	9-13
SkClose	9-13
SkGetTCPStatus	9-13
SkSetOption	9-13
SkTCPAccept	9-13

SkdTCPConnect	9-13
SkdTCPRecv	9-13
SkdTCPSend	9-13
SkdUDPCreate	9-13
SkdUDPRcv	9-13
SkdUDPSend	9-13
Internal Port1 Error	3-4, 3-15
Internal Port1 IP Address Duplication Error	3-6, 3-18
Internal Port1 IP Address Setting Error	3-6, 3-17
Internal Port1 Online	3-10, 3-21
IOI	8-10
IP address configuration	4-2
IP Address Duplication Error	3-6, 3-17
IP Address Setting Error	3-5, 3-17
IP addresses	5-2, 5-3, 5-9, 5-11
allocation	4-3
checking current addresses	4-10
setting	4-6
IP Communications	1-12
IP Route Table Error	3-7, 3-18

K

Keep Alive Monitoring Time setting	5-3
Keep Alive setting	5-3

L

Linger option	5-3
LINK settings	5-6
LINK/ACT	1-11
LINK/ACT indicator	1-11
local broadcasting	9-10
location	
SNMP	5-8
login names	5-7

M

MAC Address Error	3-5, 3-15
message communications	1-13
message service	
transmission delay	13-27
MIB	
groups	12-4
system design	12-4
MIB objects	
detailed descriptions	12-5
multi-cast communications	7-8
multicast filtering	2-3
Multiple Switches ON Error	3-7, 3-20

N

NET RUN	1-11
network configuration	
verifying	7-59
network configuration file	
reading	7-68

saving	7-66
Network Configurator	1-6
connections via Ethernet to NJ-series CPU Unit ...	7-51
Ethernet connections	7-48
network transmission delay time	13-26
network variables	7-7
importing to Network Configurator	7-29
Normal Target Node Information	3-10, 3-22, 14-4
NTP Last Operation Time	3-12, 3-23
NTP Operation Information	3-11
NTP Operation Result	3-12, 3-23
NTP Server Connection Error	3-8, 3-20
number of connections	
calculating	13-4

O

Online	3-9, 3-21
Originator Variable	7-36
output ON response time	13-26
output variables	
operation and timing	8-27, 9-17

P

Packet Interval (RPI)	7-35
passwords	5-7
PING command	6-2
port numbers	5-7, 5-8, 5-10
Port1 Basic Ethernet Setting Error	3-5, 3-16
Port1 BOOTP Server Error	3-6, 3-18
Port1 Communications Controller Error	3-5, 3-16
Port1 IP Address Duplication Error	3-6, 3-18
Port1 IP Address Setting Error	3-6, 3-17
Port1 MAC Address Error	3-5, 3-16
Port1 Online	3-10, 3-21
PPS	13-3
precautions	
socket services	9-30
priority DNS server	5-3
private addresses	4-11
programming	
CIP message communications	8-15
ladder programming for tag data links	7-73
socket services	9-19, 9-24

R

Read TCP Socket Status instruction	9-13
Read Variable Class 3 Explicit instruction	8-5
Read Variable UCMM Explicit instruction	8-5
receive data processing time	13-26
Recognition 1 settings	5-9
Recognition 2 settings	5-9
recognition method	5-9
recognition trap	
sending	5-8
Registered Target Node Information	3-10, 3-22
registering devices	7-19

request path 8-10
 requested packet intervals 7-8, 13-25
 accuracy 13-5
 adjusting according to task periods 13-24
 bandwidth usage 13-3
 changing 13-10
 settings 13-2
 response codes 8-28
 route path 8-7
 RPI setting 7-36

S

SD Memory Cards
 file types 10-16
 format of variable data 10-17
 secondary DNS server 5-3
 send data processing time 13-25
 Send Explicit Message UCMM instruction 8-5
 Set TCP Socket Option instruction 9-13
 settings 5-1
 SNMP 5-8
 SNMP agent 1-20, 12-2
 application procedure 12-21
 required settings 12-21
 SNMP messages 12-3
 SNMP Settings Display 5-8
 SNMP specifications 12-3
 SNMP Trap Settings Display 5-10, 5-12
 SNMP traps 1-20, 5-10, 5-12, 12-3
 socket service 1-19
 socket service communications
 data reception processing 9-7
 fragmenting of send data 9-5
 socket service instructions 9-13
 socket services
 application procedure 9-14
 built-in EtherNet/IP port
 overview 9-11
 port numbers 9-2
 precautions 9-30
 required settings 9-12
 sockets 9-2
 specifying method 5-11
 Status 1 Tab Page 14-2
 Status 2 Tab Page 14-4
 structure variables
 providing to input request paths 8-11
 subnet masks 4-3, 5-2
 Sysmac Studio 1-6
 system-defined variables 3-2
 EtherNet/IP communications errors 3-3, 3-13
 EtherNet/IP communications status 3-9
 EtherNet/IP communications switches 3-12

T

tag data link
 functions 7-6

 specifications 7-6
 Tag Data Link Communications Error 3-7, 3-19
 Tag Data Link Communications Start Switch 3-12, 3-23
 Tag Data Link Communications Status 3-10, 3-21
 Tag Data Link Communications Stop Switch 3-12, 3-23
 Tag Data Link Connection Failed 3-7, 3-19
 tag data link parameters
 downloading 7-53
 setting and downloading 7-7
 uploading all 7-56
 uploading from individual devices 7-58
 Tag Data Link Setting Error 3-7, 3-19
 tag data links 1-12
 automatically starting 7-63
 bandwidth usage and requested packet intervals
 13-8
 communications method 13-2
 data concurrency 7-12
 effect on task periods 13-24
 introduction 7-2
 maximum response time 13-25
 models other than NJ-series CPU Units 7-78
 setting 7-16
 starting and stopping 7-9
 starting and stopping for individual devices 7-63
 verifying 7-59
 Tag Name Resolution Error 3-7, 3-20
 tag sets 7-3
 creating 7-21
 Tag Status Tab Page 14-7
 tags 7-3
 Target Device 7-36
 Target Node Error Information 3-11, 3-23
 Target PLC Error Information 3-11, 3-22, 7-9, 14-4
 Target PLC Operating Mode 3-11, 3-22, 7-9, 14-4
 Target Variable 7-36
 task periods
 relationship with requested packet intervals 13-24
 TCP Application Communications Error 3-5, 3-15
 TCP Application Setting Error 3-8, 3-20
 TCP communications 9-3
 procedure 9-4
 TCP Socket Receive instruction 9-13
 TCP Socket Send instruction 9-13
 TCP/IP Display 5-2
 TCP/IP Interface Object 8-44
 Timeout Value 7-35, 7-36
 Trap 1 settings 5-11
 twisted-pair cable 1-5
 installation environment precautions 2-9
 installation precautions 2-6

U

UDP communications 9-3
 UDP Socket Receive instruction 9-13
 UDP Socket Send instruction 9-13
 unicast communications 7-8
 USB port 1-10

V

variable memory allocation
 rulesA-51
versions 5-11

W

Windows firewall settings
 changingA-48
Write Variable Class 3 Explicit instruction 8-5
Write Variable UCMM Explicit instruction 8-5

OMRON Corporation Industrial Automation Company
Kyoto, JAPAN

Contact: www.ia.omron.com

Regional Headquarters

OMRON EUROPE B.V.

Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31)2356-81-300/Fax: (31)2356-81-388

OMRON ELECTRONICS LLC

2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.

No. 438A Alexandra Road # 05-05/08 (Lobby 2),
Alexandra Technopark,
Singapore 119967
Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON (CHINA) CO., LTD.

Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor:

© OMRON Corporation 2016-2019 All Rights Reserved.
In the interest of product improvement,
specifications are subject to change without notice.

Cat. No. W563-E1-05

0719