OMRON

Machine Automation Controller

NJ-series

CPU Unit Built-in EtherNet/IP Port

User's Manual

NJ501-1300
NJ501-1400
NJ501-1500

CPU Unit

SYSMAC
always in control

W506-E1-01

© **OMRON, 2011**

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.

Introduction

Thank you for purchasing an NJ-series CPU Unit.
This manual contains information that is necessary to use the NJ-series CPU Unit. Please read this manual and make sure you understand the functionality and performance of the NJ-series CPU Unit 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 B3503.

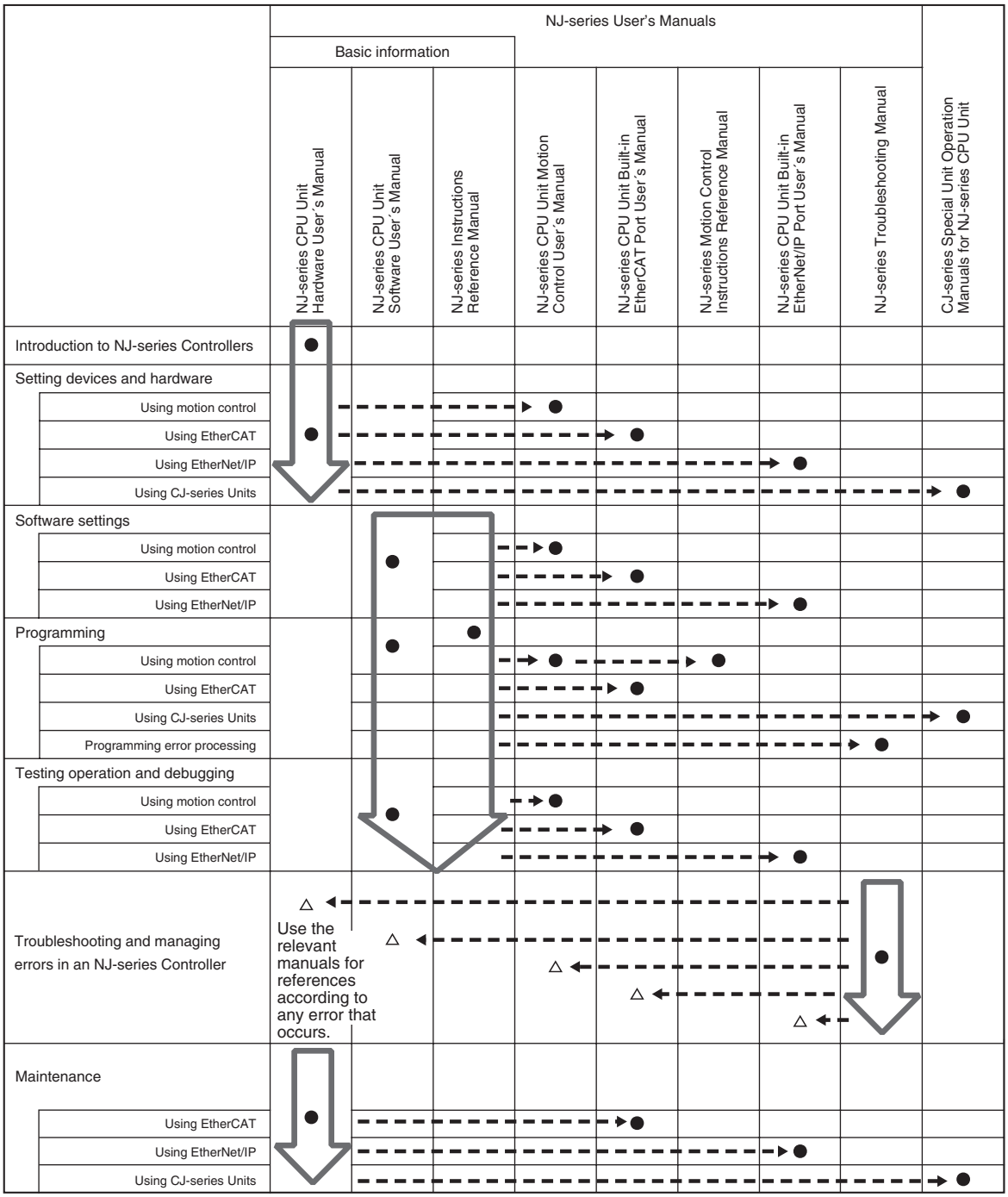
Applicable Products

This manual covers the following products.

- NJ-series CPU Units
  - NJ501-1300
  - NJ501-1400
  - NJ501-1500

Relevant Manuals

There are three manuals that provide basic information on the NJ-series CPU Units: the *NJ-series CPU Unit Hardware User's Manual*, the *NJ-series CPU Unit Software User's Manual* (this manual), and the *NJ-series Instructions Reference Manual*.
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.
Other manuals are necessary for specific system configurations and applications.
Read all of the manuals that are relevant to your system configuration and application to make the most of the NJ-series CPU Unit.



Manual Configuration

NJ-series CPU Unit Hardware User's Manual (Cat. No. W500)

Section	Description
Section 1 Introduction	This section provides an introduction to the NJ-series Controllers and their features, and gives the NJ-series Controller specifications.
Section 2 System Configuration	This section describes the system configuration used for NJ-series Controllers.
Section 3 Configuration Units	This section describes the parts and functions of the configuration devices in the NJ-series Controller configuration, including the CPU Unit and Configuration Units.
Section 4 Installation and Wiring	This section describes where and how to install the CPU Unit and Configuration Units and how to wire them.
Section 5 Troubleshooting	This section describes the event codes, error confirmation methods, and corrections for errors that can occur.
Section 6 Inspection and Maintenance	This section describes the contents of periodic inspections, the service life of the Battery and Power Supply Units, and replacement methods for the Battery and Power Supply Units.
Appendices	The appendices provide the specifications of the Basic I/O Units, Unit dimensions, load short-circuit protection detection, line disconnection detection, and measures for EMC Directives.

NJ-series CPU Unit Software User's Manual (Cat. No. W501)

Section	Description
Section 1 Introduction	This section provides an introduction to the NJ-series Controllers and their features, and gives the NJ-series Controller specifications.
Section 2 CPU Unit Operation	This section describes the variables and control systems of the CPU Unit and CPU Unit status.
Section 3 I/O Ports, Slave Configuration, and Unit Configuration	This section describes how to use I/O ports, how to create the slave configuration and unit configuration and how to assign functions.
Section 4 Controller Setup	This section describes the initial settings of the function modules.
Section 5 Designing Tasks	This section describes the task system and types of tasks.
Section 6 Programming	This section describes programming, including the programming languages and the variables and instructions that are used in programming.
Section 7 Simulation, Transferring Projects to the Physical CPU Unit, and Operation	This section describes simulation of Controller operation and how to use the results of simulation.
Section 8 CPU Unit Status	This section describes CPU Unit status.
Section 9 CPU Unit Functions	This section describes the functionality provided by the CPU Unit.
Section 10 Communications Setup	This section describes how to go online with the CPU Unit and how to connect to other devices.
Section 11 Example of Actual Application Procedures	This section describes the procedures that are used to actually operate an NJ-series Controller.
Section 12 Troubleshooting	This section describes the event codes, error confirmation methods, and corrections for errors that can occur.
Appendices	The appendices provide the CPU Unit specifications, task execution times, system-defined variable lists, data attribute lists, CJ-series Unit memory information, CJ-series Unit memory allocation methods, and data type conversion information.

NJ-series CPU Unit Built-in EtherNet/IP Port User's Manual (Cat. No. W506) (This Manual)

Section	Description
Section 1 Introduction	This section provides an overview and the specifications of the built-in EtherNet/IP port on an NJ-series Controller. It introduces EtherNet/IP communications and describes the system configuration and operating procedures.
Section 2 Installing Ethernet Networks	This section describes the Ethernet network configuration devices, network installation, and cable connection methods.
Section 3 System-defined Variables Related to the Built-in EtherNet/IP Port	This section describes the system-defined variables that are related to the built-in EtherNet/IP port.
Section 4 Determining IP Addresses	This section describes how to set IP addresses for built-in EtherNet/IP ports.
Section 5 Sysmac Studio Settings for the Built-in EtherNet/IP Port	This section describes the settings that are required for EtherNet/IP communications.
Section 6 Testing Communications	This section describes how to perform communications test with EtherNet/IP nodes to confirm that the built-in EtherNet/IP port is set correctly.
Section 7 Tag Data Link Functions	This section introduces tag data link communications and describes the settings that are required to use tag data links.
Section 8 Message Communications	This section describes how to use CIP message communications for devices on the EtherNet/IP network, e.g., to read and write data.
Section 9 Socket Service	This section describes how to use socket communications to send and receive data with TCP/UDP.
Section 10 FTP Server	This section describes how to use the FTP server to download and upload files in the SD Memory Card to and from FTP clients.
Section 11 Automatic Clock Adjustment	This section describes how to automatically get clock information from an NTP server to update the clock information in the CPU Unit.
Section 12 SNMP Agent	This section describes how to use the SNMP to manage the built-in EtherNet/IP port as an SNMP agent.
Section 13 Communications Performance and Communications Load	This section describes tag data links communications, adjustment of the communications load, and communications time.
Section 14 Troubleshooting	This section describes how to use event codes and network status to confirm errors and corrections for them.
Appendices	The appendices provide a functional comparison of EtherNet/IP between NJ-series CPU Units and other series, and describe EDS file management, Windows firewall settings for connections from computers, and details on memory used for CJ-series Units.

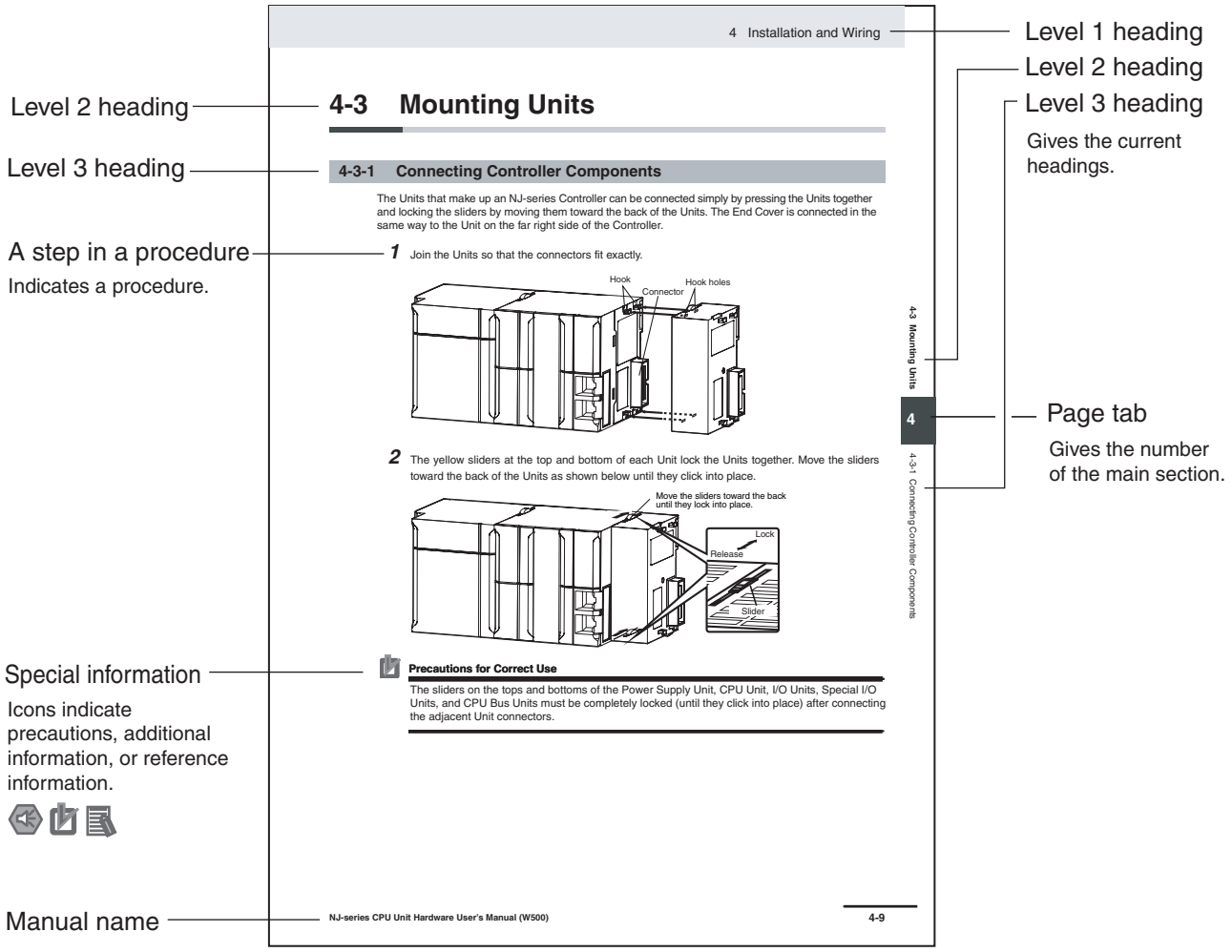
Sysmac Studio Version 1 Operation Manual (Cat. No. W504)

Section	Description
Section 1 Introduction	This section provides an overview and lists the specifications of the Sysmac Studio and describes its features and components.
Section 2 Installation and Uninstallation	This section describes how to install and uninstall the Sysmac Studio.
Section 3 System Design	This section describes the basic concepts for designing an NJ-series System with the Sysmac Studio and the basic operating procedures.
Section 4 Programming	This section describes how to create programs with the Sysmac Studio.
Section 5 Online Connections to a Controller	This section describes how to go online with a Controller.
Section 6 Debugging	This section describes how to debug the programs online on the Controller or debug it offline with the Simulator.
Section 7 Other Functions	This section describes Sysmac Studio functions other than system design functions.
Section 8 Reusing Programming	This section describes how to reuse the programs that you create with the Sysmac Studio.
Section 9 Support Software Provided with the Sysmac Studio	This section describes the Support Software that is provided with the Sysmac Studio.
Section 10 Troubleshooting	This section describes the error messages that are displayed when you check a program on the Sysmac Studio and how to correct those errors.
Appendices	The appendices describe the following: Driver Installation for Direct USB Cable Connection Specifying One of Multiple Ethernet Interface Cards Online Help Simulation Instructions

Manual Structure

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.

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

2 Installing Ethernet Networks

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

4 Determining IP Addresses

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

6 Testing Communications

7 Tag Data Link Functions

8 Message Communications

9 Socket Service

10 FTP Server

11 Automatic Clock Adjustment

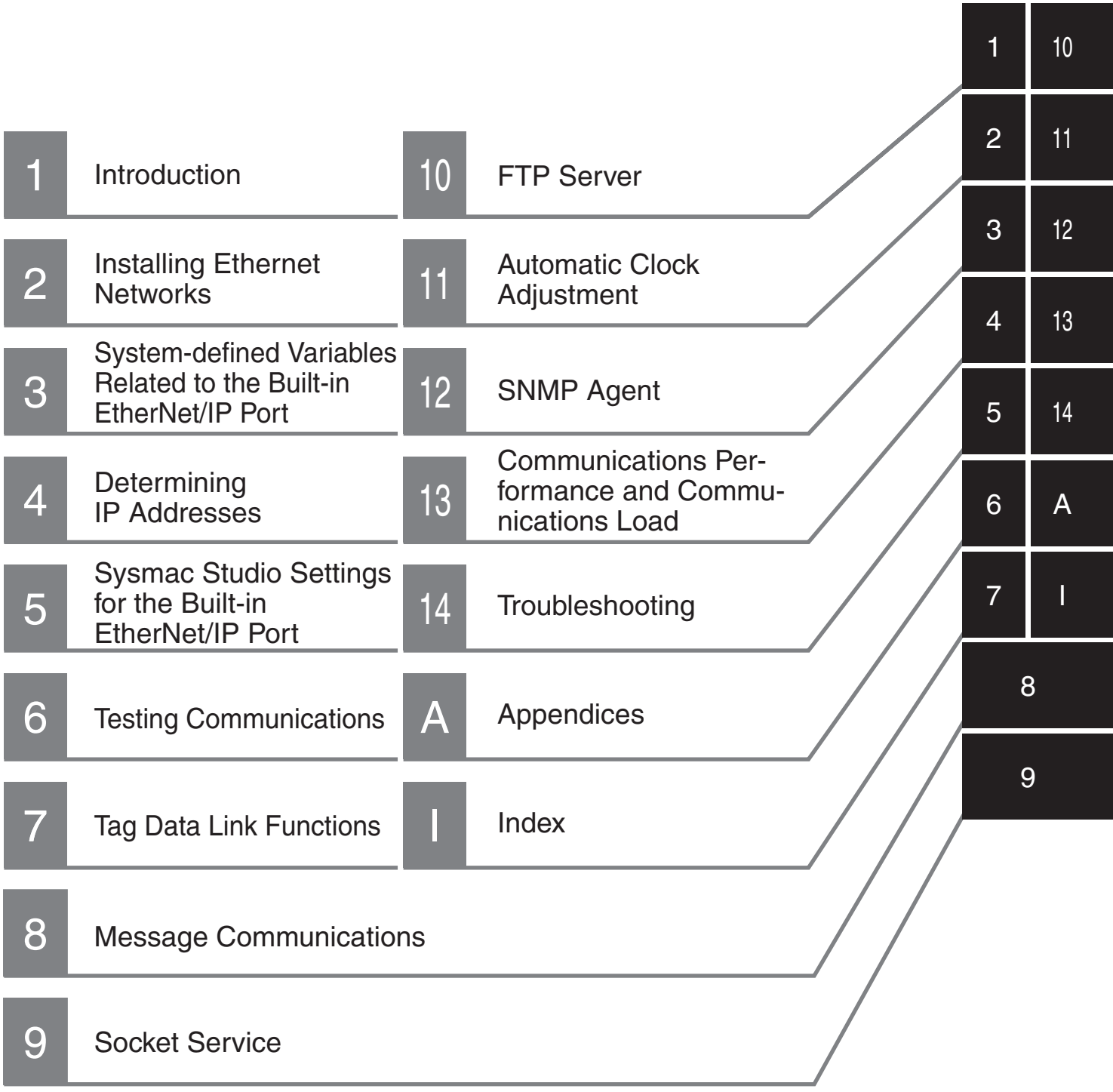
12 SNMP Agent

13 Communications Performance and Communications Load

14 Troubleshooting

A Appendices

I Index



CONTENTS

Introduction 1
Relevant Manuals 2
Manual Configuration 3
Manual Structure 6
Sections in this Manual 9
Read and Understand this Manual 17
Safety Precautions 21
Precautions for Safe Use 23
Precautions for Correct Use 24
Regulations and Standards 25
Unit Versions 27
Related Manuals 30
Revision History 32

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 NJ-series CPU Units 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 1-9
1-4 Introduction to Communications Services 1-11
1-4-1 CIP (Common Industrial Protocol) Communications Services 1-11
1-4-2 BOOTP Client 1-12
1-4-3 FTP Server 1-13
1-4-4 Automatic Clock Adjustment 1-13
1-4-5 Socket Service 1-14
1-4-6 Specifying Host Names 1-14
1-4-7 SNMP Agent 1-15
1-5 EtherNet/IP Communications Procedures 1-16
1-5-1 Outline 1-16

Section 2 Installing Ethernet Networks

2-1 Selecting the Network Devices 2-2
2-1-1 Recommended Network Devices 2-2
2-1-2 Network Devices Manufactured by OMRON 2-3
2-1-3 Ethernet Switch Types 2-3
2-1-4 Ethernet Switch Functions 2-3
2-1-5 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 2-6

2-3 Connecting to the Network 2-8
2-3-1 Ethernet Connectors 2-8
2-3-2 Connecting the Cable 2-8

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 3-10

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 4-3

4-2 Built-in EtherNet/IP Port IP Address Settings 4-4
4-2-1 Determining IP Addresses 4-4
4-2-2 Setting IP Addresses 4-5
4-2-3 Online Connection 4-7
4-2-4 Checking the Current IP Address 4-9

4-3 Private and Global Addresses 4-10
4-3-1 Private and Global Addresses 4-10
4-3-2 Using a Private Address for the Built-in EtherNet/IP Port 4-11
4-3-3 Using a Global Address for the Built-in EtherNet/IP Port 4-12

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

5-1 TCP/IP Display 5-2

5-2 Link Settings Display 5-4

5-3 FTP Display 5-5

5-4 NTP Settings Display 5-6

5-5 SNMP Settings Display 5-7

5-6 SNMP Trap Settings Display 5-9

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 7-5
7-1-4 Overview of Operation 7-6
7-1-5 Starting and Stopping Tag Data Links 7-8
7-1-6 Controller Status 7-8
7-1-7 Concurrency of Tag Data Link Data 7-10
7-2 Setting Tag Data Links 7-14
7-2-1 Starting the Network Configurator 7-14
7-2-2 Tag Data Link Setting Procedure 7-16
7-2-3 Registering Devices 7-17
7-2-4 Creating Tags and Tag Sets 7-18
7-2-5 Connection Settings 7-28
7-2-6 Connecting the Network Configurator to the Network 7-38
7-2-7 Downloading Tag Data Link Parameters 7-43
7-2-8 Uploading Tag Data Link Parameters 7-46
7-2-9 Verifying the Tag Data Links 7-49
7-2-10 Starting and Stopping Tag Data Links 7-52
7-2-11 Clearing the Device Parameters 7-54
7-2-12 Saving the Network Configuration File 7-55
7-2-13 Reading a Network Configuration File 7-57
7-2-14 Checking Connections 7-58
7-2-15 Changing Devices 7-60
7-2-16 Displaying Device Status 7-61
7-3 Ladder Programming for Tag Data Links 7-62
7-3-1 Ladder Programming for Tag Data Links 7-62
7-3-2 Status Flags Related to Tag Data Links 7-66
7-4 Tag Data Links with Models Other than NJ-Series CPU Units 7-67

Section 8 Message Communications

8-1 Overview of the CIP Message Communications Service 8-2
8-1-1 Overview of the CIP Message Communications Service 8-2
8-1-2 Message Communications Service Specifications 8-2
8-2 Using CIP Message Communications 8-3
8-2-1 Overview 8-3
8-2-2 CIP Communications Instructions 8-3
8-2-3 Using CIP Communications Instructions 8-4
8-2-4 Route Path 8-6
8-2-5 Preparing Derivative Data Types to Use CIP Communications Instructions 8-9
8-2-6 Sample Programming for CIP Connectionless (UCMM) Message Communications 8-12
8-2-7 Sample Programming for CIP Connection (Class 3) Message Communications 8-18
8-2-8 Operation Timing 8-25
8-2-9 Response Codes 8-26
8-3 CIP Objects Sent to the Built-in EtherNet/IP Port 8-30
8-3-1 CIP Objects Sent to the Built-in EtherNet/IP Port 8-30
8-3-2 Identity Object (Class ID: 01 Hex) 8-30
8-3-3 TCP/IP Interface Object (Class ID: F5 hex) 8-32

8-3-4 Ethernet Link Object (Class ID: F6 Hex) 8-34
8-3-5 Controller Object (Class ID: C4 Hex) 8-38

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 9-4
9-2-3 Data Reception Processing 9-6
9-2-4 Broadcasting 9-8
9-3 Overview of Built-in EtherNet/IP Port Socket Services 9-9
9-3-1 Overview 9-9
9-3-2 Procedure 9-9
9-4 Settings Required for the Socket Services 9-10
9-4-1 Settings Required for the Socket Services 9-10
9-4-2 Making the Settings with the Sysmac Studio 9-11
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 9-16
9-6-4 UDP Sample Programming 9-18
9-6-5 TCP Sample Programming 9-23
9-7 Precautions in Using Socket Services 9-29
9-7-1 Precautions for UDP and TCP Socket Services 9-29
9-7-2 Precautions for UDP Socket Services 9-29
9-7-3 Precautions for TCP Socket Services 9-29

Section 10 FTP Server

10-1 Overview and Specifications 10-2
10-1-1 Overview 10-2
10-1-2 Specifications 10-2
10-2 FTP Server Function Details 10-3
10-2-1 Supported Files 10-3
10-2-2 Connecting to the FTP Server 10-3
10-3 Using the FTP Server Function 10-5
10-3-1 Procedure 10-5
10-3-2 List of Settings Required for the FTP Server Function 10-5
10-3-3 FTP Settings Display 10-6
10-4 FTP Server Application Example 10-7
10-5 Using FTP Commands 10-9
10-5-1 Table of Commands 10-9
10-5-2 Using the Commands 10-10
10-6 Using SD Memory Card Operations 10-16
10-6-1 SD Memory Card Types 10-16
10-6-2 File Types 10-17
10-6-3 Initializing SD Memory Cards 10-17
10-6-4 Format of Variable Data 10-17
10-7 Application Example from a Host Computer 10-19

Section 11 Automatic Clock Adjustment

11-1 Automatic Clock Adjustment 11-2
11-1-1 Overview 11-2
11-1-2 Specifications 11-3
11-2 Procedure to Use the Automatic Clock Adjustment Function 11-4
11-2-1 Procedure 11-4
11-2-2 Settings Required for Automatic Clock Adjustment 11-4
11-2-3 NTP Display 11-5
11-3 Troubleshooting Automatic Clock Adjustment 11-7

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 12-20
12-2-1 Procedures 12-20
12-2-2 Settings Required for the SNMP Agent 12-20
12-2-3 SNMP Settings Display 12-22
12-2-4 SNMP Trap Settings Display 12-24
12-3 Troubleshooting SNMP Agent 12-26
12-3-1 SNMP Agent Errors 12-26
12-3-2 Event Log Error Codes for the SNMP Agent 12-26

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 13-21
13-3-3 Relationship between Task Periods and Packet Intervals (RPIs) 13-23
13-3-4 Maximum Tag Data Link I/O Response Time 13-25
13-4 Message Service Transmission Delay 13-27

Section 14 Troubleshooting

14-1 Overview of Errors 14-2
14-1-1 How to Check for Errors 14-2
14-1-2 Errors Related to the EtherNet/IP Function Module 14-6

14-2 Troubleshooting 14-9
14-2-1 Error Table 14-9
14-2-2 Error Descriptions 14-14
14-2-3 Troubleshooting 14-37

14-3 Checking Status with the Network Configurator 14-38
14-3-1 The Network Configurator's Device Monitor Function 14-38
14-3-2 Connection Status Codes and Troubleshooting 14-45

Appendices

A-1 Functional Comparison of EtherNet/IP Ports on NJ-series CPU Units and Other Series.A-2

A-2 EDS File Management A-3
A-2-1 Installing EDS Files A-4
A-2-2 Creating EDS Files A-5
A-2-3 Deleting EDS Files A-5
A-2-4 Saving EDS Files A-6
A-2-5 Searching EDS Files A-6
A-2-6 Displaying EDS File Properties A-6
A-2-7 Creating EDS Index Files A-7

A-3 Precautions for Using the Network Configurator on Windows XP, Windows Vista, or Windows 7 ..... A-8
A-3-1 Changing Windows Firewall Settings A-8

A-4 Variable Memory Allocation Methods A-11
A-4-1 Variable Memory Allocation Rules A-11
A-4-2 Important Case Examples A-14

A-5 Precautions When Accessing External Outputs from NJ-series CPU Units A-15

Index

Read and Understand this Manual

Please read and understand this manual before using the product. Please consult your OMRON representative if you have any questions or comments.

Warranty and Limitations of Liability

WARRANTY

OMRON's exclusive warranty is that the products are free from defects in materials and workmanship for a period of one year (or other period if specified) from date of sale by OMRON.

OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, REGARDING NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR PARTICULAR PURPOSE OF THE PRODUCTS. ANY BUYER OR USER ACKNOWLEDGES THAT THE BUYER OR USER ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE. OMRON DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED.

LIMITATIONS OF LIABILITY

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

In no event shall the responsibility of OMRON for any act exceed the individual price of the product on which liability is asserted.

IN NO EVENT SHALL OMRON BE RESPONSIBLE FOR WARRANTY, REPAIR, OR OTHER CLAIMS 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 OR REPAIR.

Application Considerations

SUITABILITY FOR USE

OMRON shall not be responsible for conformity with any standards, codes, or regulations that apply to the combination of products in the customer's application or use of the products.

At the customer's request, OMRON will provide applicable third party certification documents identifying ratings and limitations of use that apply to the products. This information by itself is not sufficient for a complete determination of the suitability of the products in combination with the end product, machine, system, or other application or use.

The following are some examples of applications for which particular attention must be given. This is not intended to be an exhaustive list of all possible uses of the products, nor is it intended to imply that the uses listed may be suitable for the products:

- Outdoor use, uses involving potential chemical contamination or electrical interference, or conditions or uses not described in this manual.
- Nuclear energy control systems, combustion systems, railroad systems, aviation systems, medical equipment, amusement machines, vehicles, safety equipment, and installations subject to separate industry or government regulations.
- Systems, machines, and equipment that could present a risk to life or property.

Please know and observe all prohibitions of use applicable to the products.

NEVER USE THE PRODUCTS FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCTS ARE PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

PROGRAMMABLE PRODUCTS

OMRON shall not be responsible for the user's programming of a programmable product, or any consequence thereof.

Disclaimers

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 model numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the products may be changed without any notice. When in doubt, special model numbers may be assigned to fix or establish key specifications for your application on your request. Please consult with your OMRON representative at any time to confirm actual specifications of purchased products.

DIMENSIONS AND WEIGHTS

Dimensions and weights are nominal and are not to be used for manufacturing purposes, even when tolerances are shown.

PERFORMANCE DATA

Performance data given in this manual 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 users must correlate it to actual application requirements. Actual performance is subject to the OMRON Warranty and Limitations of Liability.

ERRORS AND OMISSIONS

The information in this manual has been carefully checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical, or proofreading errors, or omissions.

Safety Precautions

Definition of Precautionary Information

The following notation is used in this manual to provide precautions required to ensure safe usage of the built-in EtherNet/IP port on an NJ-series CPU Unit. The safety precautions that are provided are extremely important to safety. Always read and heed the information provided in all safety precautions.
The following notation is used.
Refer to the following manuals for safety precautions for the built-in EtherNet/IP port.
Installation precautions are also provided for the NJ-series CPU Unit and NJ-series Controller system.

- NJ-series CPU Unit Hardware User's Manual (W500)
- NJ-series CPU Unit Software User's Manual (W501)

WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury. Additionally, there may be severe property damage.
Caution	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury, or property damage.

Precautions for Safe Use
Indicates precautions on what to do and what not to do to ensure safe usage of the product.

Precautions for Correct Use
Indicates precautions on what to do and what not to do to ensure proper operation and performance.

Symbols

The circle and slash symbol indicates operations that you must not do.
The specific operation is shown in the circle and explained in text.
This example indicates prohibiting disassembly.

The triangle symbol indicates precautions (including warnings).
The specific operation is shown in the triangle and explained in text.
This example indicates a precaution for electric shock.

The triangle symbol indicates precautions (including warnings).
The specific operation is shown in the triangle and explained in text.
This example indicates a general precaution.

The filled circle symbol indicates operations that you must do.
The specific operation is shown in the circle and explained in text.
This example shows a general precaution for something that you must do.

Precautions for Safe Use

Refer to the following manuals for precautions for the safe use of the built-in EtherNet/IP port.
Installation precautions are also provided for the NJ-series CPU Unit and NJ-series Controller system.

- NJ-series CPU Unit Hardware User's Manual (W500)
- NJ-series CPU Unit Software User's Manual (W501)

Precautions for Correct Use

Refer to the following manuals for precautions for the correct use of the built-in EtherNet/IP port. Installation precautions are also provided for the NJ-series CPU Unit and NJ-series Controller system.

- NJ-series CPU Unit Hardware User's Manual (W500)
- NJ-series CPU Unit Software User's Manual (W501)

Regulations and Standards

Conformance to EC Directives

Applicable Directives

- EMC Directives
- Low Voltage Directive

Concepts

● EMC Directive

OMRON devices that comply with EC Directives also conform to the related EMC standards so that they can be more easily built into other devices or the overall machine. The actual products have been checked for conformity to EMC standards.*
Whether the products conform to the standards in the system used by the customer, however, must be checked by the customer. EMC-related performance of the OMRON devices that comply with EC Directives will vary depending on the configuration, wiring, and other conditions of the equipment or control panel on which the OMRON devices are installed. The customer must, therefore, perform the final check to confirm that devices and the overall machine conform to EMC standards.

* Applicable EMC (Electromagnetic Compatibility) standards are as follows:
EMS (Electromagnetic Susceptibility): EN 61131-2 and EN 61000-6-2
EMI (Electromagnetic Interference): EN 61131-2 and EN 61000-6-4 (Radiated emission: 10-m regulations)

● Low Voltage Directive

Always ensure that devices operating at voltages of 50 to 1,000 VAC and 75 to 1,500 VDC meet the required safety standards. The applicable directive is EN 61131-2.

● Conformance to EC Directives

The NJ-series Controllers comply with EC Directives. To ensure that the machine or device in which the NJ-series Controller is used complies with EC Directives, the Controller must be installed as follows:

- The NJ-series Controller must be installed within a control panel.
- You must use reinforced insulation or double insulation for the DC power supplies connected to DC Power Supply Units and I/O Units.
- NJ-series Controllers that comply with EC Directives also conform to the Common Emission Standard (EN 61000-6-4). Radiated emission characteristics (10-m regulations) may vary depending on the configuration of the control panel used, other devices connected to the control panel, wiring, and other conditions.
  You must therefore confirm that the overall machine or equipment complies with EC Directives.

Conformance to Shipbuilding Standards

The NJ-series Controllers comply with the following shipbuilding standards. Applicability to the shipbuilding standards is based on certain usage conditions. It may not be possible to use the product in some locations. Contact your OMRON representative before attempting to use a Controller on a ship.

Usage Conditions for NK and LR Shipbuilding Standards

- The NJ-series Controller must be installed within a control panel.
- Gaps in the door to the control panel must be completely filled or covered with gaskets or other material.
- The following noise filter must be connected to the power supply line.

Noise Filter

Manufacturer	Model
Cosel Co., Ltd.	TAH-06-683

Trademarks

- Sysmac and SYSMAC are trademarks or registered trademarks of OMRON Corporation in Japan and other countries for OMRON factory automation products.
- Windows, Windows 98, Windows XP, Windows Vista, and Windows 7 are registered trademarks of Microsoft Corporation in the USA and other countries.
- EtherCAT® is a registered trademark of Beckhoff Automation GmbH for their patented technology.
- The SD logo is a trademark of SD-3C, LLC.

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

Software Licenses and Copyrights

This product incorporates certain third party software. The license and copyright information associated with this software is available at http://www.fa.omron.co.jp/nj_info_e/.

Unit Versions

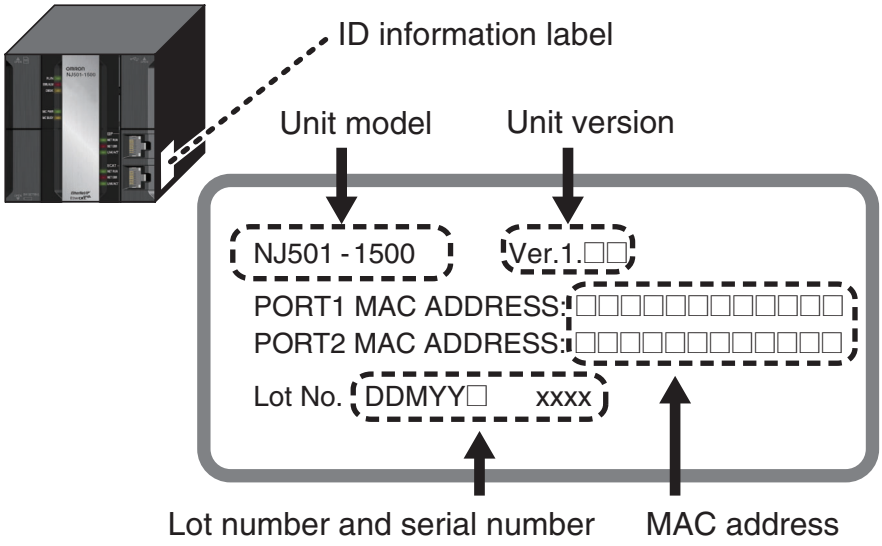
Unit Versions

A “unit version” has been introduced to manage CPU Units in the NJ Series according to differences in functionality accompanying Unit upgrades.

Notation of Unit Versions on Products

The unit version is given on the ID information label of the products for which unit versions are managed, as shown below.

Example for NJ-series NJ501-□□□□ CPU Unit:



The following information is provided on the ID information label.

Item	Description
Unit model	Gives the model of the Unit.
Unit version	Gives the unit version of the Unit.
Lot number and serial number	Gives the lot number and serial number of the Unit. DDMYY: Lot number, □: For use by OMRON, xxxx: Serial number “M” gives the month (1 to 9: January to September, X: October, Y: November, Z: December)
MAC address	Gives the MAC address of the built-in port on the Unit.

Confirming Unit Versions with Sysmac Studio

You can use the Unit Production Information on the Sysmac Studio to check the unit version of the CPU Unit, CJ-series Special I/O Units, CJ-series CPU Bus Units, and EtherCAT slaves. The unit versions of CJ-series Basic I/O Units cannot be checked from the Sysmac Studio.

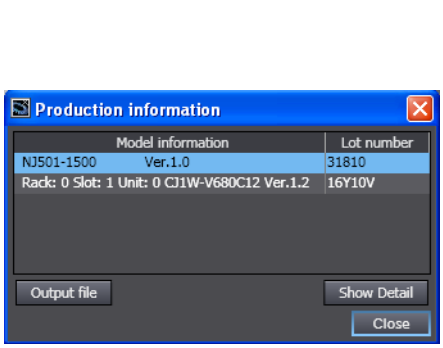
● CPU Unit and CJ-series Units

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

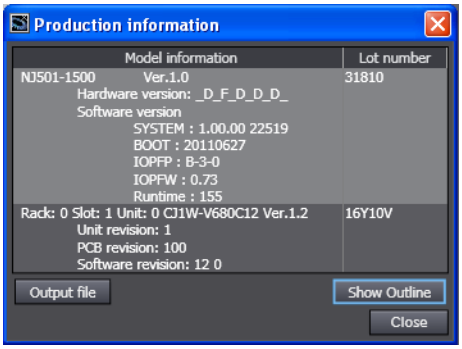
The Unit Editor is displayed for the Controller Configurations and Setup layer.

2 Right-click any open space in the Unit Editor and select ***Production Information***.

The Production Information Dialog Box is displayed.



Simple Display



Detailed Display

In this example, “Ver.1.0” is displayed next to the unit model.

The following items are displayed.

CPU Unit	CJ-series Units
Unit model	Unit model
Unit version	Unit version
Lot number	Lot number
	Rack number, slot number, and unit number

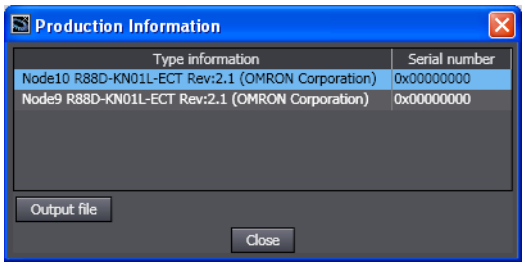
● EtherCAT Slaves

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 Configuration Tab Page is displayed for the Controller Configurations and Setup layer.

2 Right-click the master in the EtherCAT Configurations Editing Pane and select **Display Production Information**.

The Production Information Dialog Box is displayed.



The following items are displayed.
Node address
Type information*
Serial number

* If the model number cannot be determined (such as when there is no ESI file), the vendor ID, product code, and revision number are displayed.

Unit Version Notation

In this manual, unit versions are specified as shown in the following table.

Product nameplate	Notation in this manual	Remarks
“Ver.1.0” or later to the right of the lot number	Unit version 1.0 or later	Unless unit versions are specified, the information in this manual applies to all unit versions.

Related Manuals

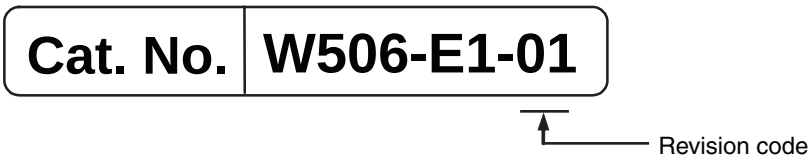
The following manuals are related to the NJ-series Controllers. Use these manuals for reference.

Manual name	Cat. No.	Model numbers	Application	Description
NJ-series CPU Unit Hardware User's Manual	W500	NJ501-□□□□	Learning the basic specifications of the NJ-series CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided.	An introduction to the entire NJ-series system is provided along with the following information on a Controller built with an NJ501 CPU Unit. • Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection Use this manual together with the *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).
NJ-series CPU Unit Software User's Manual	W501	NJ501-□□□□	Learning how to program and set up an NJ-series CPU Unit. Mainly software information is provided.	The following information is provided on a Controller built with an NJ501 CPU Unit. • CPU Unit operation • CPU Unit features • Initial settings • Programming based on IEC 61131-3 language specifications Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500).
NJ-series CPU Unit Motion Control User's Manual	W507	NJ501-□□□□	Learning about motion control settings and programming concepts.	The settings and operation of the CPU Unit and programming concepts for motion control are described. Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) and *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).
NJ-series Instructions Reference Manual	W502	NJ501-□□□□	Learning about the specifications of the instruction set that is provided by OMRON.	The instructions in the instruction set (IEC 61131-3 specifications) are described. When programming, use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) and *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).
NJ-series Motion Control Instructions Reference Manual	W508	NJ501-□□□□	Learning about the specifications of the motion control instructions that are provided by OMRON.	The motion control instructions are described. When programming, use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500), NJ-series CPU Unit Software User's Manual (Cat. No. W501) and *NJ-series CPU Unit Motion Control User's Manual* (Cat. No. W507).
CJ-series Special Unit Manuals for NJ-series CPU Unit	W490 W498 W499 W491 Z317 W492 W494 W497	CJ1W-□□□□	Learning how to use CJ-series Units with an NJ-series CPU Unit.	The methods and precautions for using CJ-series Units with an NJ501 CPU Unit are described, including access methods and programming interfaces. Manuals are available for the following Units. Analog I/O Units, Insulated-type Analog I/O Units, Temperature Control Units, ID Sensor Units, High-speed Counter Units, Serial Communications Units, and DeviceNet Units. Use these manuals together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) and *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).

Manual name	Cat. No.	Model numbers	Application	Description
NJ-series CPU Unit Built-in EtherCAT Port User's Manual	W505	NJ501-□□□□	Using the built-in EtherCAT port on an NJ-series CPU Unit.	Information on the built-in EtherCAT port is provided. This manual provides an introduction and provides information on the configuration, features, and setup. Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) and *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).
NJ-series CPU Unit Built-in EtherNet/IP Port User's Manual	W506	NJ501-□□□□	Using the built-in EtherNet/IP port on an NJ-series CPU Unit.	Information on the built-in EtherNet/IP port is provided. Information is provided on the basic setup, tag data links, and other features. Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) and *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).
NJ-series Troubleshooting Manual	W503	NJ501-□□□□	Learning about the errors that may be detected in an NJ-series Controller.	Concepts on managing errors that may be detected in an NJ-series Controller and information on individual errors are described. Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) and *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).
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.
CX-Integrator CS/CJ/CP/NSJ-series Network Configuration Tool Operation Manual	W464		Learning how to configure networks (data links, routing tables, Communications Unit settings, etc.).	Describes operating procedures for the CX-Integrator.
CX-Designer User's Manual	V099		Learning to create screen data for NS-series Programmable Terminals.	Describes operating procedures for the CX-Designer.
CX-Protocol Operation Manual	W344		Creating data transfer protocols for general-purpose devices connected to CJ-series Serial Communications Units.	Describes operating procedures for the CX-Protocol.

Revision History

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



Revision code	Date	Revised content
01	July 2011	Original production

1

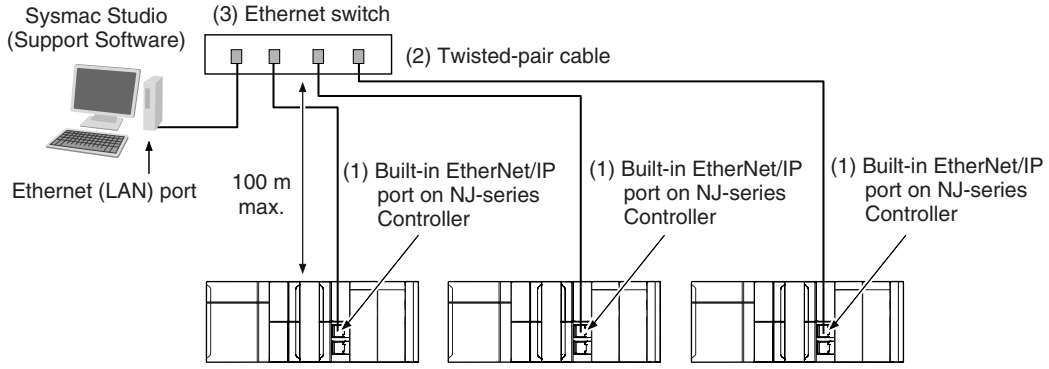
Introduction

1-1 Introduction 1-2
1-1-1 EtherNet/IP Features 1-2
1-1-2 Features of Built-in EtherNet/IP Port on NJ-series CPU Units 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 1-9
1-4 Introduction to Communications Services 1-11
1-4-1 CIP (Common Industrial Protocol) Communications Services 1-11
1-4-2 BOOTP Client 1-12
1-4-3 FTP Server 1-13
1-4-4 Automatic Clock Adjustment 1-13
1-4-5 Socket Service 1-14
1-4-6 Specifying Host Names 1-14
1-4-7 SNMP Agent 1-15
1-5 EtherNet/IP Communications Procedures 1-16
1-5-1 Outline 1-16

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.



EtherNet/IP System Configuration Example

● 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. Data of up to 9,600 words can be exchanged at high speed between Controllers and 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 NJ-series CPU Units

● 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 for up to 9,600 words of data.

● 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 NJ-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.

● Automatic Controller Clock Adjustment

The clocks built into Controllers connected to Ethernet can be automatically adjusted to the time of the clock in the NTP server. If all of the clocks in the system are automatically adjusted to the same time, time stamps can be used to analyze production histories.

* A separate NTP server is necessary to automatically adjust the Controller clocks.

● 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 NTP servers, 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

Additional Information

CIP (Common Industrial Protocol)

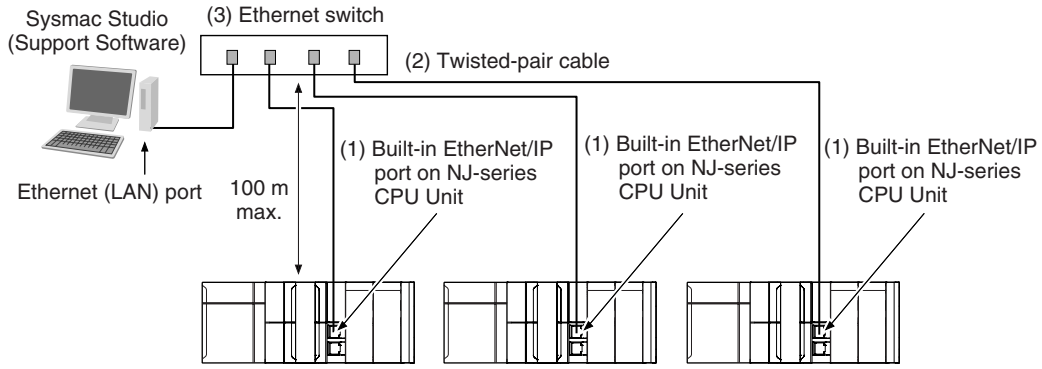
CIP is a shared industrial protocol for the OSI application layer. The CIP is used in networks such as EtherNet/IP, CompoNet, and DeviceNet. Data can be routed easily between networks that are based on the CIP. You can therefore easily configure a transparent network from the field device level to the 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 NJ501-□□□□ NJ-series CPU Unit (built-in EtherNet/IP port) OMRON PLCs CJ2H-CPU□□-EIP or CJ2M-CPU3□ CJ2 CPU Unit (built-in EtherNet/IP port) CJ-series CJ1W-EIP21 EtherNet/IP Unit* CS-series CS1W-EIP21 EtherNet/IP Unit	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 a 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 *2-1-1 Recommended Network Devices*.

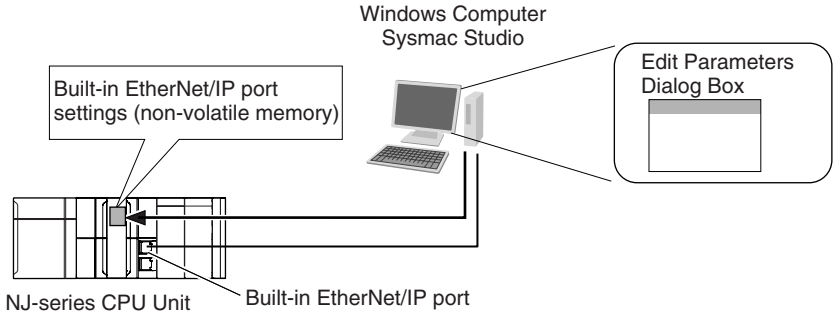
* The CJ1W-EIP21 cannot be mounted to an NJ-series CPU Unit.

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 CPU Unit. 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

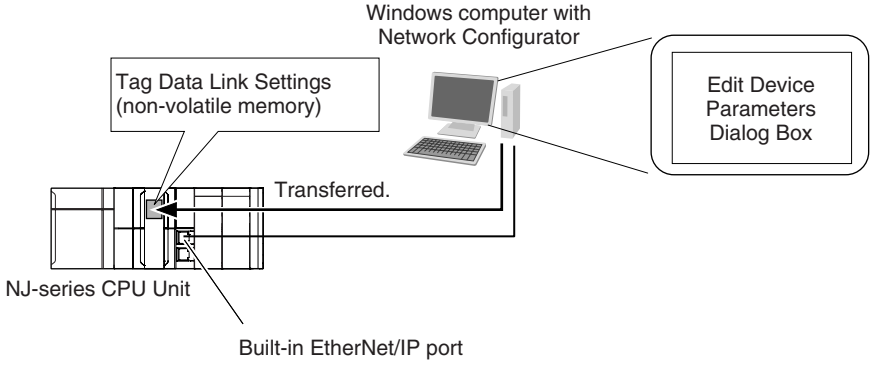
Use the Network Configurator to set the tag data links for the built-in EtherNet/IP port. (The Network Configurator is included in Sysmac Studio.) The main functions of the Network Configurator are given below.

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.*

1-3 Built-in EtherNet/IP Port

1-3-1 Specifications

Item		Specifications
Communications protocol		TCP/IP or UDP/IP
Supported services		Sysmac Studio connection, tag data link, CIP message communications, socket services, FTP server, automatic clock adjustment (NTP client), SNMP agent, DNS client, and BOOTP client
Physical layer		100Base-TX or 10Base-T (100Base-TX is recommended.) *1
Transmission specifications	**Media access method**	CSMA/CD
	Modulation	Baseband
	Transmission paths	Star form
	Baud rate	100 Mbps (100Base-TX)
	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 a Ethernet switch is used.
CIP service: Tag data links (cyclic communications)	**Number of connections**	32
	Packet interval (refresh cycle)	10 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	1,000 pps *2 **Note** This includes the heartbeat.
	Number of registrable tags	256
	Tag types	Network variables CIO, Work, Holding, DM, or EM Area
	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	19,200 bytes
	Maximum data size per connection	600 bytes *3 **Note** Data concurrency is maintained within each connection. Refer to *7-1-7 Concurrency of Tag Data Link Data* for methods to maintain concurrency.
	Number of registrable tag sets	32 (1 connection = 1 tag set)
	Maximum size of 1 tag set	300 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

Item		Specifications
CIP message service: Explicit messages *5	**Class 3 (connected)**	Number of connections: 32 (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.
SNMP	**Agents**	SNMPv1 or SNMPv2c
	MIB	MIB-II
EtherNet/IP conformance test		Conforms to A8
Ethernet interface		10Base-T or 100Base-TX Auto negotiation or fixed settings

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

*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-series Units support a large forward open, but before connecting to nodes of other companies, confirm that those devices also support it.

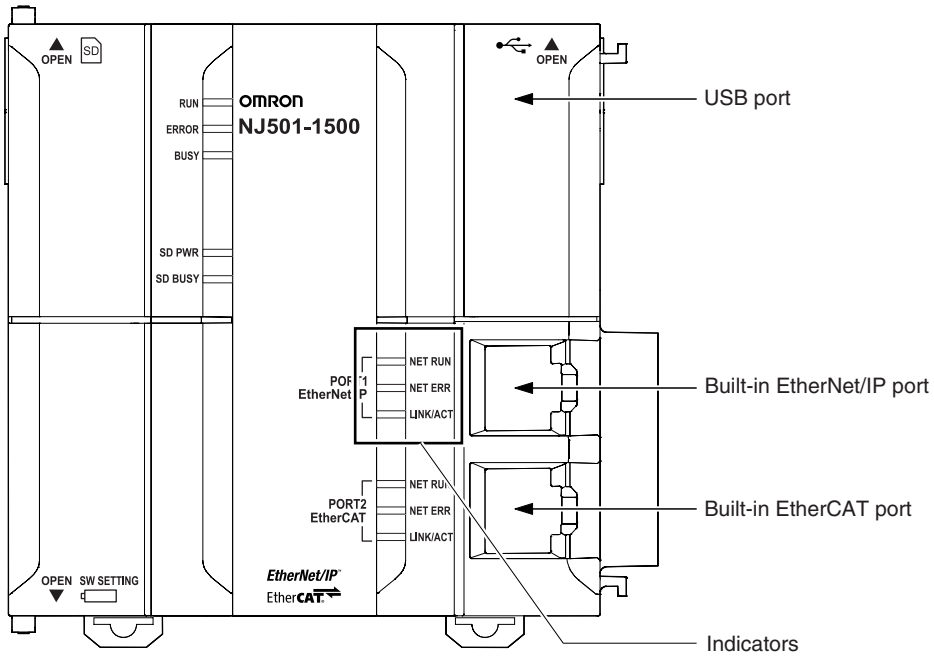
*4 If the parameters of the built-in EtherNet/IP port are changed, the port is restarted. When other nodes are in communications with the affected node, the communications will temporarily time out and automatically recover later.

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

*6 The built-in EtherNet/IP port uses the TCP/UDP port numbers shown in the following table.

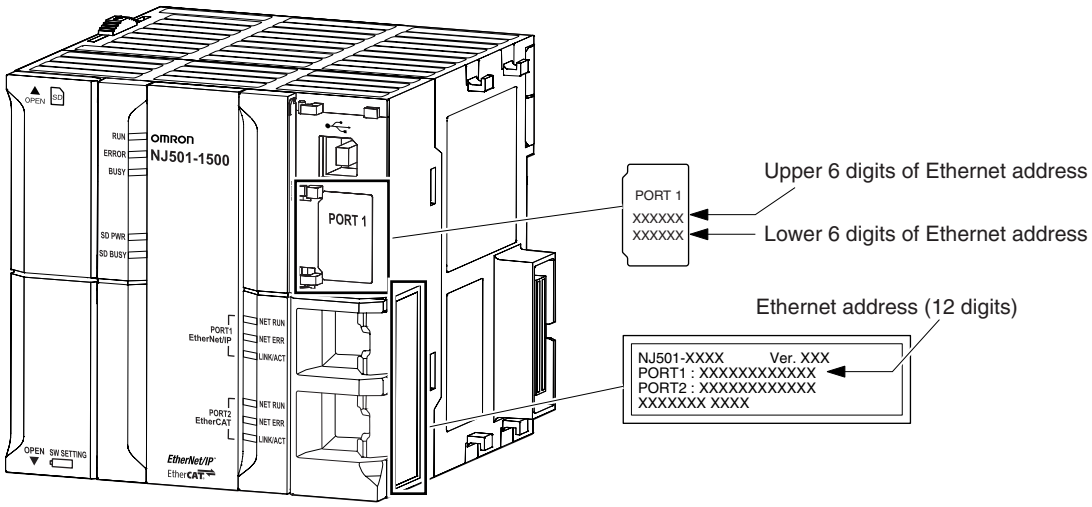
Service	Protocol	Port number	Remarks
EIP data links	UDP	2222	Fixed values
Used by system	UDP	2223, 2224	
	TCP	9610	
CIP messages	TCP	44818	
FTP client data transfer port	TCP	20	
DNS client	TCP/UDP	53	
BOOTP client	UDP	68	
HTTP server	TCP	80	
Used by system, other	TCP/UDP	9600	You can change the port number in the Unit Settings on the Sysmac Studio.
FTP client control port	TCP	21	
NTP client	UDP	123	
SNMP agent	UDP	161	
SNMP trap	UDP	162	

1-3-2 Part Names and Functions

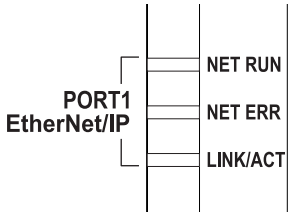


● Ethernet Address Notation

A specific Ethernet address is allocated to each device connected to the Ethernet network. The built-in EtherNet/IP port's address is listed in 12-digit hexadecimal in the two places shown below on the CPU Unit.



Indicators (LEDs)



● NET RUN, NET ERR, and LINK/ACT

- NET RUN indicator: This shows the status of the CIP connection (tag data links, Class 3 messages).
- NET ERR indicator: This shows the network communications error status. Refer to *Section 14 Troubleshooting* and Error Processing for details.
- LINK/ACT indicator: This shows the Ethernet communications status.

Indicator	Color	Status	Operating status
NET RUN	---	Not lit	Ethernet communications are not possible. • The power supply is OFF or the Controller is being reset. • A MAC address error or communications Controller error occurred.
	Green	Flashing	Ethernet communications are in progress. • Tag data link connection establishment in progress (originator operation) • IP address acquisition with BOOTP in progress.
		Lit	Normal If only the target is set for the tag data link, this indicator lights regardless of whether the connection from the originator is established. It remains lit even if the data links are stopped.
NET ERR	---	Not lit	There are no Ethernet communications errors. • The power supply is OFF or was reset.
	Red	Flashing	A user-recoverable error occurred. • An error occurred in TCP/IP communications or CIP communications. • FTP Server Setting Error, NTP Server Setting Error, etc. • Tag Data Link Setting Error, Tag Data Link Verification Error, etc.
		Lit	A user-non-recoverable error occurred. • A MAC address error or communications Controller error occurred.
LINK/ACT	---	Not lit	Link not established. • 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.

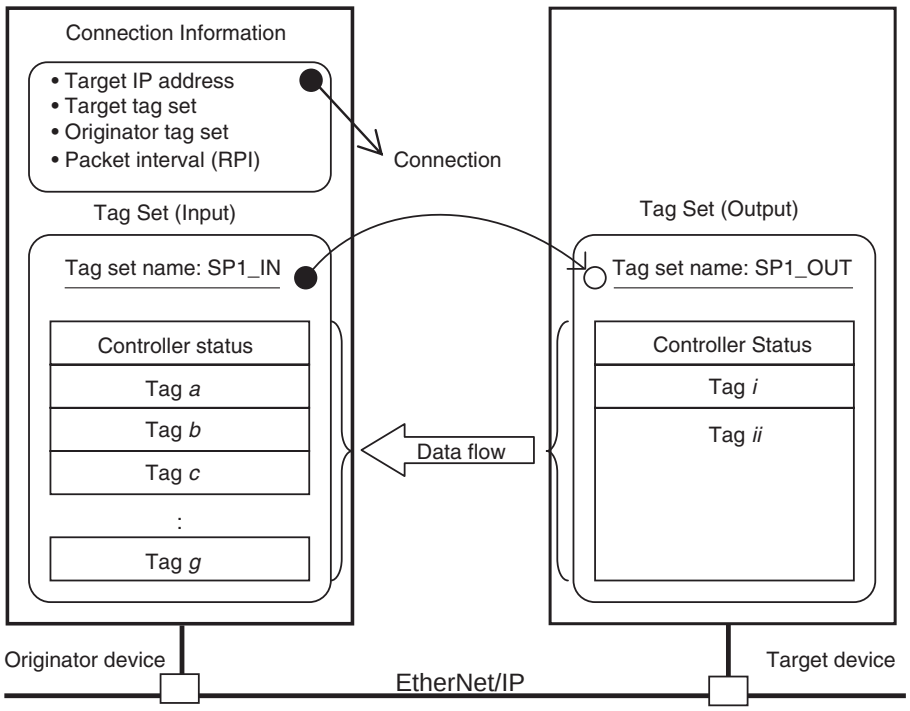
1-4 Introduction to Communications Services

1-4-1 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. You can register up to 32 connections. The following table gives the built-in EtherNet/IP port tag and tag set specifications.

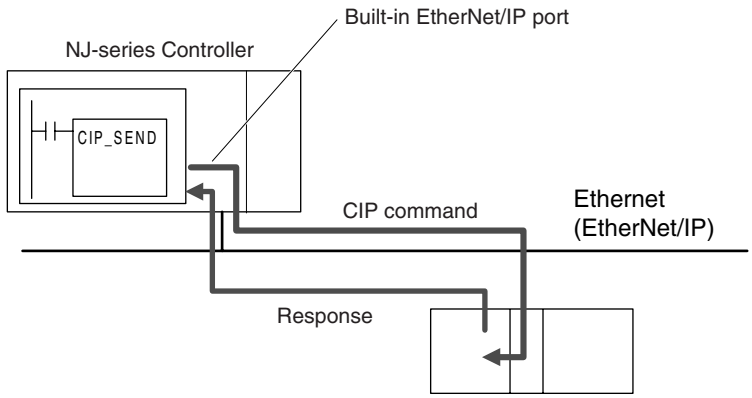
Tags	Tag sets
Total size of all tags ≤ 9,600 words	Maximum size of 1 tag set ≤ 300 words (The maximum size is 299 words when the tag set includes the Controller status.)
Maximum size of 1 tag ≤ 300 words (The maximum size is 299 words when the tag set includes the Controller status.)	Number of tags per tag set ≤ 8 (7 tags/tag set when the tag set includes the Controller status) **Note** Input and output variables cannot be combined.
Number of registrable tags ≤ 256	Number of registrable tag sets ≤ 32



* 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.

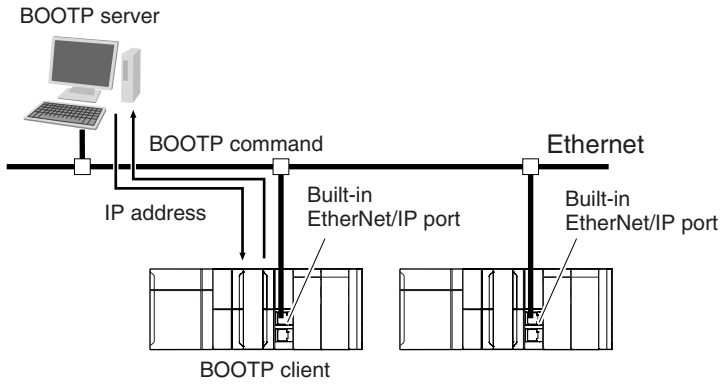
Message Communications (Class 3 UCMM: Unconnected Message Service)

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 Send instruction from the user program in the NJ-series CPU Unit (without a connection).



1-4-2 BOOTP Client

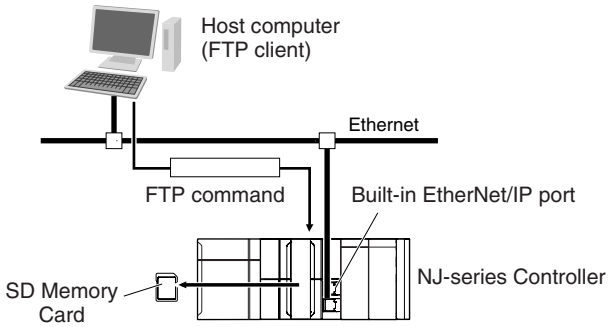
You set the built-in EtherNet/IP port in the BOOTP settings to use the BOOTP client to obtain settings, such as the built-in EtherNet/IP port IP address.

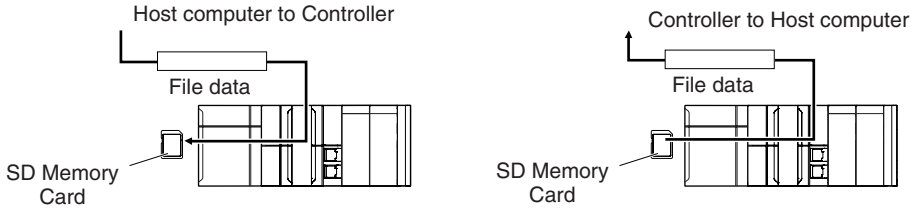


The built-in EtherNet/IP port IP address is obtained from the BOOTP server when the power is turned ON.

1-4-3 FTP Server

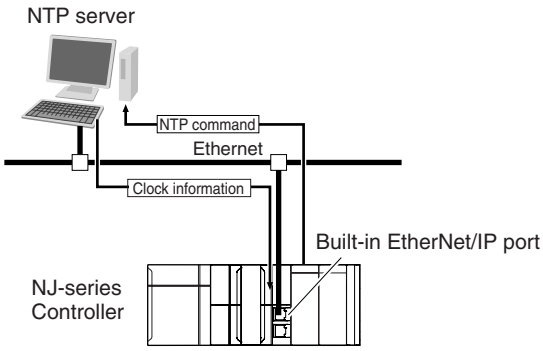
An FTP server is built into the built-in EtherNet/IP port so that files can be read from and written to the SD Memory Card in the CPU Unit of the Controller from computers at other Ethernet nodes. This makes it possible to exchange data files between a host computer and the Controller with the host computer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





1-4-4 Automatic Clock Adjustment

With the built-in EtherNet/IP port, clock information is read from the NTP server at the specified time or at a specified interval after the power supply to the CPU Unit is turned ON. The internal clock time in the CPU Unit is updated with the read time.

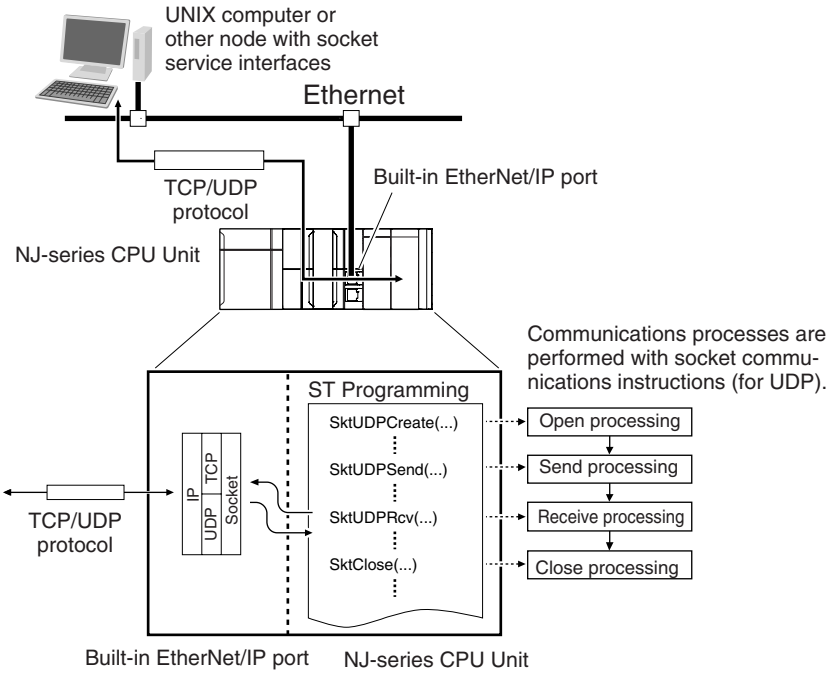


Precautions for Correct Use

An NTP server is required to use automatic clock adjustment.

1-4-5 Socket Service

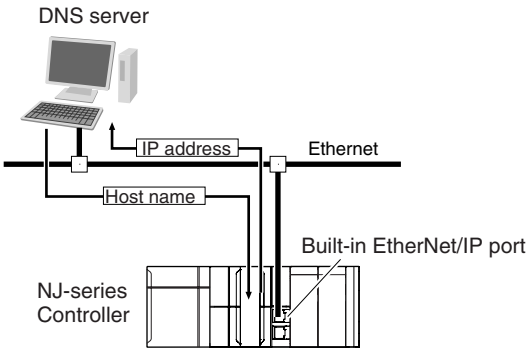
You can send data to and receive data from any node on Ethernet with the UDP or TCP protocol. To send/receive data with a socket service, you execute multiple socket communications instructions in sequence in an ST program to execute the required communications processes. After a connection with the 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.



1-4-6 Specifying Host Names

You can directly specify IP addresses, but you can also use the host names instead of the IP addresses for NTP servers, 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

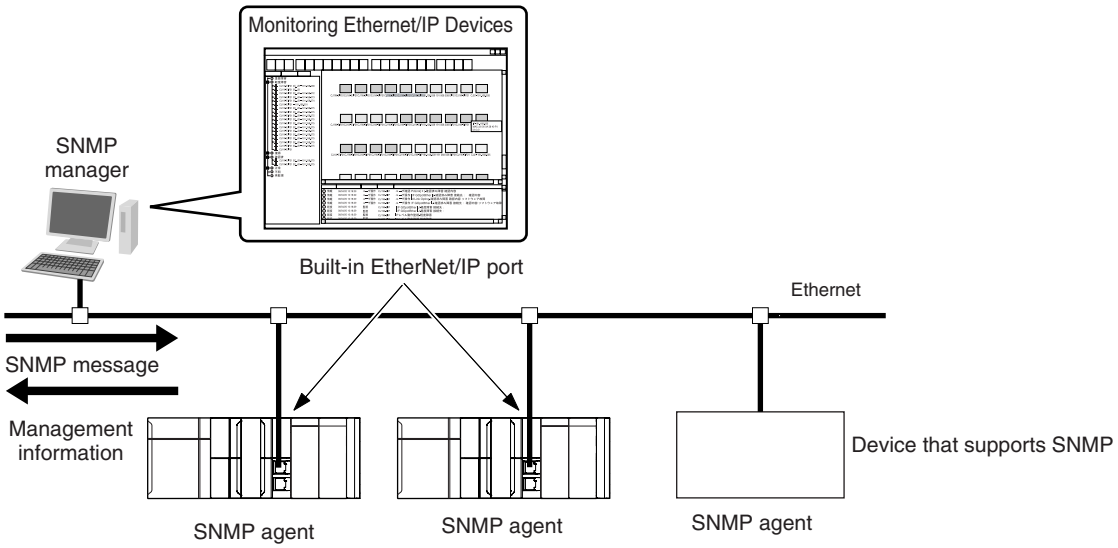


Precautions for Correct Use

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

1-4-7 SNMP Agent

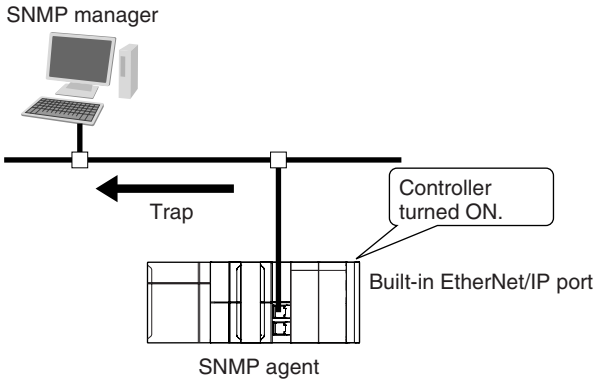
The SNMP agent passes internal status information from the built-in EtherNet/IP port to network management software that uses an SNMP manager.



SNMP 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

1-5-1 Outline

● 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:

- Default: 192.168.250.1 (subnet mask = 255.255.255.0)
- Set any IP address.
- Obtain from BOOTP server.

↓

3 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.

↓

4 Turn the power ON, and connect the Sysmac Studio online.

↓

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

Section 6 Testing Communications

● Using Tag Data Links

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

Section 7 Tag Data Link Functions

- Create the network configuration.
- Set the connections.

↓

2 Turn the power ON, and connect the Network Configurator online.

↓

3 Download the tag data link setting.

↓

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

↓

5 Check operation.

1-3-2 Part Names and Functions

Section 14 Troubleshooting

- 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.

● Using the Message Communications Service

- CIP Communications Instructions

1 Execute CIP communications instructions in the user program.

Section 8 Message Communications

↓

2 Check operation.

1-3-2 Part Names and Functions

Section 14 Troubleshooting

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

● 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 the NJ-series CPU Unit from an FTP client application.

Input the FTP login name and password to log onto the built-in EtherNet/IP port.

● Using the Automatic Clock Adjustment

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

Section 11 Automatic Clock Adjustment

- Set the NTP settings (enabling NTP and execution conditions).

↓

2 Execute automatic clock adjustment.

- Execute automatic adjustment at specified times or specified intervals.
- Use the Sysmac Studio to check the NTP Last Operation Time and NTP Operation Result system-defined variables.

● 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 Network Devices Manufactured by OMRON 2-3
2-1-3 Ethernet Switch Types 2-3
2-1-4 Ethernet Switch Functions 2-3
2-1-5 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 2-6
2-3 Connecting to the Network 2-8
2-3-1 Ethernet Connectors 2-8
2-3-2 Connecting the Cable 2-8

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

Product		Manufacturer	Model	Contact phone number
Sizes and conductor pairs: AWG 24 × 4 pairs	Cables	Tonichi Kyosan Cable, Ltd.	NETSTAR-C5E SAB 0.5 × 4P	Kanetsu Planning Department
		Kuramo Electric Co., Ltd.	KETH-SB	Kuramo Electric Co., Ltd.
		SWCC Showa Cable Systems Co. Ltd.	FAE-5004	SWCC Showa Cable Systems Co. Ltd.
	RJ45 Connectors	Panduit Corporation	MPS588	Panduit Corporation, Japan Branch, Osaka Sales Office
Sizes and conductor pairs: AWG 22 × 2 pairs	Cables	Kuramo Electric Co., Ltd.	KETH-PSB-OMR*	Kuramo Electric Co., Ltd.
	RJ45 Assembly Connectors	OMRON	XS6G-T421-1*	OMRON Corporation, Customer Support Center
Sizes and conductor pairs: 0.5 mm × 4 pairs	Cables	Fujikura Ltd.	F-LINK-E 0.5mm × 4P	Kanetsu Planning Department
	RJ45 Connectors	Panduit Corporation	MPS588	Panduit Corporation, Japan Branch, Osaka Sales Office

* We recommend using combinations of the above cables and connectors.

● Boots

Product	Model	Contact phone number
TSUKO	MK Boots (IV) LB	TSUKO

Precautions for Correct Use

- Always use a Ethernet switch for tag data links in the network.
- If a repeater hub is used for EtherNet/IP tag data links (cyclic communications), the network's communications load will increase, data collisions will occur frequently, and stable communications will be impossible.

2-1-2 Network Devices Manufactured by OMRON

The following network devices are manufactured by OMRON for EtherNet/IP networks.

Name	Model number	Specifications	OK: Supported, ---: Not supported.	
		Description	Number of ports	Failure detection output
Ethernet switch	W4S1-03B	Packet priority control (QoS): EtherNet/IP control data priority Failure detection: Broadcast storm, LSI error detection, 10/100Base-TX, Auto negotiation	3	---
	W4S1-05B		5	---
	W4S1-05C		5	OK

2-1-3 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.

2-1-4 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.

2-1-5 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 a 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 the following precautions when you select a Ethernet switch. 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.

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 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.

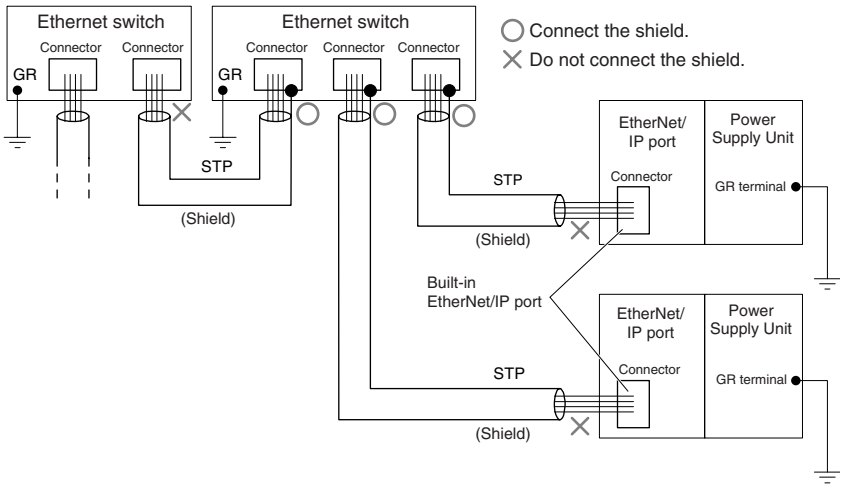
2-2-2 Recommended Network Devices

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

Precautions on Laying Twisted-pair Cable

- Noise immunity may be reduced by ground loops, which can occur due to improper shield connections and grounding. Ground the shield at one location, as shown in the following diagram.
- Do not connect the shield to the built-in EtherNet/IP port's connector.
- If a cable connects two Ethernet switches, connect the shield at only one end.



- 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.

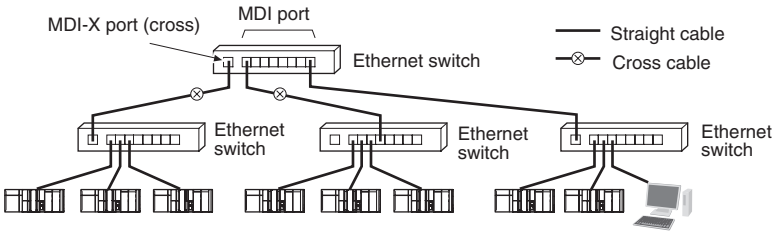
Ethernet Switch Installation Environment Precautions

- 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 a 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

- Connect two Ethernet switches to each other as follows:
  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)	
			Full	Half	Full	Half
Auto-Nego		Best	---	OK	---	OK
10 Mbps (fixed)	Full	---	OK	---	---	---
	Half	OK	---	OK	---	---
100 Mbps (fixed)	Full	---	---	---	Best	---
	Half	OK	---	---	---	OK

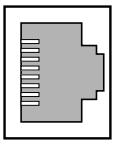
Best = Recommended; OK = Allowed; --- = Not allowed.

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)

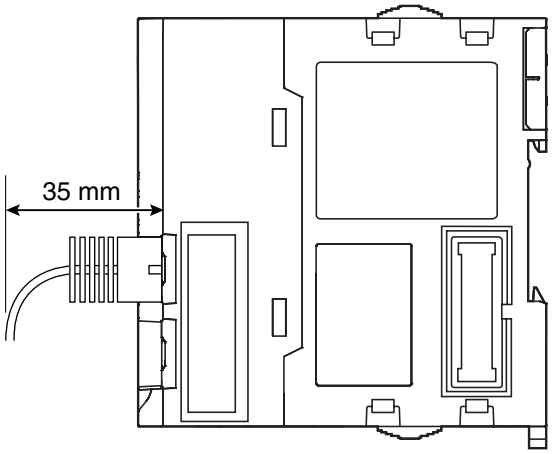


Connector pin	Signal name	Abbr.	Signal direction
1	Transmission data +	TD+	Output
2	Transmission data –	TD–	Output
3	Reception data +	RD+	Input
4	Not used.	---	---
5	Not used.	---	---
6	Reception data –	RD–	Input
7	Not used.	---	---
8	Not used.	---	---
Hood	Frame ground	FG	---

2-3-2 Connecting the Cable

Precautions for Correct Use

- Turn OFF the power supply to the Controller before you connect or disconnect twisted-pair cable.
- Allow enough space for the bending radius of the twisted-pair cable as shown below.



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

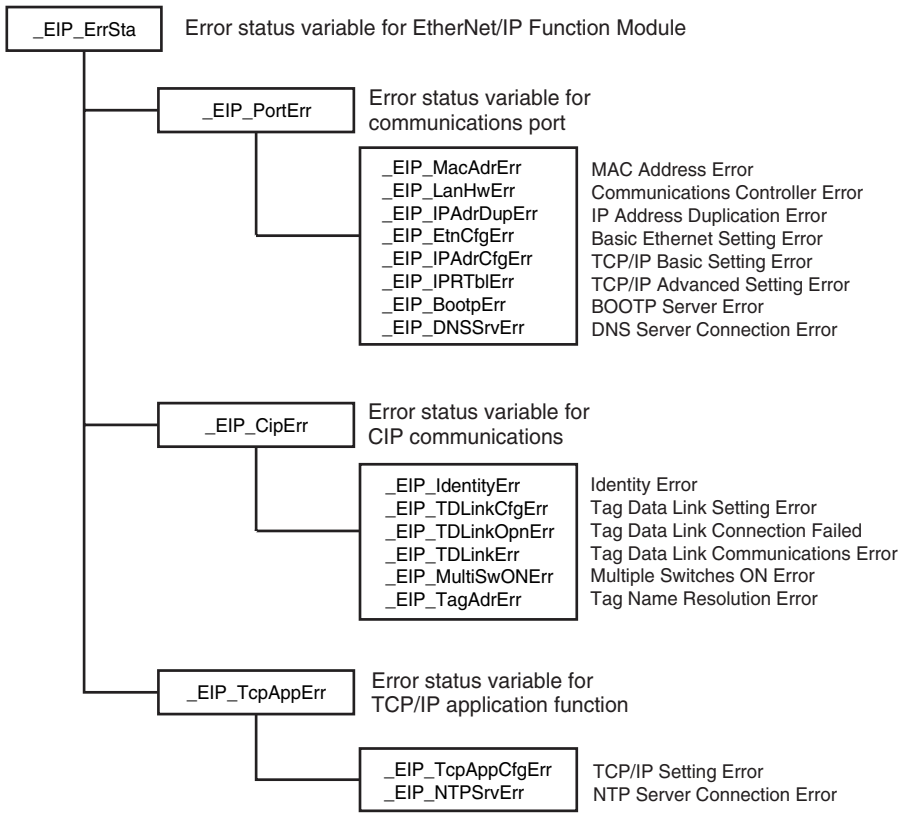
3-1 System-defined Variables Related to the Built-in EtherNet/IP Port 3-2
3-2 System-defined Variables 3-3
3-3 Specifications for Individual System-defined Variables 3-10

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, Sysmac Studio setting errors, Network Configurator setting errors, TCP/IP application errors (e.g., FTP or NTP), 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.



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	This is the error status variable for the built-in EtherNet/IP port. It represents the following error flags. • *_EIP_PortErr* (Communications Port Error) • *_EIP_CipErr* (CIP Communications Error) • *_EIP_TcpAppErr* (TCP Application Communications Error) **Note** Refer to *Meanings of Error Status Bits* given later for the meaning of individual bits.	WORD	16#0000 to 16#00F0	page 3-10
_EIP_PortErr	Communications Port Error	This is the error status variable for the communications port. It represents the following error flags. • *_EIP_MacAdrErr* (MAC Address Error) • *_EIP_LanHwErr* (Communications Controller Error) • *_EIP_EtnCfgErr* (Basic Ethernet Setting Error) • *_EIP_IPAdrCfgErr* (TCP/IP Basic Setting Error) • *_EIP_IPAdrDupErr* (IP Address Duplication Error) • *_EIP_BootpErr* (BOOTP Server Error) • *_EIP_IPRTblErr* (TCP/IP Advanced Setting Error) **Note** If a Link OFF or Built-in EtherNet/IP Processing Error occurs, it is recorded in the event log and then the corresponding bit turns ON. Refer to *Meanings of Error Status Bits* given later for the meaning of individual bits.	WORD	16#0000 to 16#00F0	page 3-11

Variable name	Meaning	Function	Data type	Range of values	Reference
_EIP_CipErr	CIP Communications Error	This is the error status variable for CIP communications. It represents the following error flags. • *_EIP_IdentityErr* (Identity Error) • *_EIP_TDLinkCfgErr* (Tag Data Link Setting Error) • *_EIP_TDLinkOpnErr* (Tag Data Link Connection Failed) • *_EIP_TDLinkErr* (Tag Data Link Communications Error) • *_EIP_TagAdrErr* (Tag Name Resolution Error) • *_EIP_MultiSwONErr* (Multiple Switches ON Error) **Note** If a Tag Resolution Error occurs, it is recorded in the event log and this variable changes to TRUE. Refer to *Meanings of Error Status Bits* given later for the meaning of individual bits.	WORD	16#0000 to 16#00F0	page 3-11
_EIP_TcpAppErr	TCP Application Communications Error	This is the error status variable for TCP application communications. It represents the following error flags. • *_EIP_TopAppCfgErr* (TCP/IP Setting Error) • *_EIP_NTPSrvErr* (NTP Server Connection Error) • *_EIP_DNSSrvErr* (DNS Server Connection Error) **Note** Refer to *Meanings of Error Status Bits* given later for the meaning of individual bits.	WORD	16#0000 to 16#00F0	page 3-11
_EIP_MacAdrErr	MAC Address Error	Indicates that an error occurred when the MAC address was read at startup. TRUE: Error FALSE: Normal	BOOL	TRUE or FALSE	page 3-12
_EIP_LanHwErr	Communications Controller Error	TRUE: The communications controller failed. FALSE: Normal	BOOL	TRUE or FALSE	page 3-12
_EIP_EtnCfgErr	Basic Ethernet Setting Error	TRUE: The Ethernet communications speed setting (Speed/Duplex) is incorrect. Or, a read operation failed. FALSE: Normal	BOOL	TRUE or FALSE	page 3-12
_EIP_IPAdrCfgErr	TCP/IP Basic Setting Error	TRUE: • There is an illegal IP address setting. • A read operation failed. • The IP address obtained from the BOOTP server is inconsistent. • The DNS settings are not correct. FALSE: Normal	BOOL	TRUE or FALSE	page 3-12
_EIP_IPAdrDupErr	IP Address Duplication Error	TRUE: The same IP address is assigned to more than one node. FALSE: Other than the above.	BOOL	TRUE or FALSE	page 3-12
_EIP_BootpErr	BOOTP Server Error	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 3-13

Variable name	Meaning	Function	Data type	Range of values	Reference
_EIP_IPRTblErr	TCP/IP Advanced Setting Error	TRUE: There is an error in one of the following settings. Or, a read operation failed. • IP router table settings • Hosts settings FALSE: Normal	BOOL	TRUE or FALSE	page 3-13
_EIP_IdentityErr	Identity Error	TRUE: The identity information (which you cannot overwrite) is not correct. Or, a read operation failed. FALSE: Normal	BOOL	TRUE or FALSE	page 3-13
_EIP_TDLinkCfgErr	Tag Data Link Setting Error	TRUE: The tag data link settings are incorrect. Or, a read operation failed. FALSE: Normal	BOOL	TRUE or FALSE	page 3-13
_EIP_TDLinkOpnErr	Tag Data Link Connection Failed	TRUE: The connection was not established because the remote node information in the tag data link parameters was different from the actual node information. **Note** This variable does not change to TRUE if there is no remote node when the power is turned ON. FALSE: Other than the above.	BOOL	TRUE or FALSE	page 3-13
_EIP_TDLinkErr	Tag Data Link Communications Error	TRUE: A timeout occurred in a tag data link connection. FALSE: Other than the above.	BOOL	TRUE or FALSE	page 3-14
_EIP_TagAdrErr	Tag Name Resolution Error	TRUE: Tag resolution failed (i.e., the address could not be identified from the tag name). The following causes are possible. • The size of the network-published variable does not agree with the tag setting. • 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 CPU Unit. • There is no network-published variable in the CPU Unit that corresponds to the tag setting. FALSE: Other than the above.	BOOL	TRUE or FALSE	page 3-14
_EIP_MultiSwONErr	Multiple Switches ON Error	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 3-14
_EIP_TcpAppCfgErr	TCP/IP Setting Error	TRUE: At least one of the set values for a TCP/IP application (FTP, NTP, SNMP) is incorrect. Or, a read operation failed. FALSE: Normal	BOOL	TRUE or FALSE	page 3-14
_EIP_NTPSrvErr	NTP Server Connection Error	TRUE: The NTP client failed to connect to the server (timeout). FALSE: NTP is not set or the connection was successful.	BOOL	TRUE or FALSE	page 3-14
_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 3-15

● 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	Master-detected error: This bit indicates whether the master detected a Controller error in the Unit/slave for the error status of the Controller error. TRUE: The master detected a Controller error. FALSE: The master has not detected a Controller error. (Valid for *_CJB_U_ErrSta* and *_EC_SlvErrTbl*.)
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 *_CJB_ErrSta, _MC_ErrSta,* and *_EC_ErrSta*.)
8 to 13	Not used.
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	Not used.

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), *_CJB_ErrSta* (I/O Bus Error Status), *_CJB_MstrErrSta* (I/O Bus Master Error Status), and *_CJB_UnitErrSta* (I/O Bus Unit 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
_EIP_EtnOnlineSta	Online	TRUE: The built-in EtherNet/IP port's communications can be used. (The link is ON and IP address is defined. Also, there are no errors.) FALSE: The built-in EtherNet/IP port's communications is disabled due to an error in initial processing or restart processing.	BOOL	TRUE or FALSE	page 3-15
_EIP_TDLinkRunSta	Tag Data Link Communications Status	TRUE: At least one connection is in normal operation. FALSE: Other than the above.	BOOL	TRUE or FALSE	page 3-15
_EIP_TDLinkAllRunSta	All Tag Data Link Communications Status	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 3-15
_EIP_RegTargetSta [255]	Registered Target Node Information	This variable gives a list of nodes for which built-in EtherNet/IP connections are registered. This variable is valid only when the built-in EtherNet/IP port is the originator. *Array[x]* is TRUE: The connection to the node with a target node ID of x is registered. *Array[x]* 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 3-15
_EIP_EstbTargetSta [255]	Normal Target Node Information	This variable gives a list of nodes that have normally established built-in EtherNet/IP connections. *Array[x]* is TRUE: The connection to the node with a target node ID of x was established normally. *Array[x]* 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 3-16
_EIP_TargetPLCModeSta [255]	Target PLC Operating Mode	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 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. *Array[x]* is TRUE: This is the operating state of the target Controller with a node address of x. *Array[x]* is FALSE: Other than the above.	ARRAY [0..255] OF BOOL	TRUE or FALSE	page 3-16

Variable name	Meaning	Function	Data type	Range of values	Reference
_EIP_TargetPLCErr [255]	Target PLC Error Information	This variable shows the error status (logical OR of fatal and non-fatal errors) of the target node Controllers that are connected 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. *Array[x]* is TRUE: A fatal or non-fatal error occurred in the target Controller with a target node ID of x. *Array[x]* is FALSE: Other than the above.	ARRAY [0..255] OF BOOL	TRUE or FALSE	page 3-16
_EIP_TargetNodeErr [255]	Target Node Error Information	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 array elements are valid only when the Registered Target Node Information is TRUE. *Array[x]* 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. *Array[x]* 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.).	ARRAY [0..255] OF BOOL	TRUE or FALSE	page 3-16
_EIP_NTPResult	NTP Operation Information	Use the GetNTPStatus instruction to read the NTP operation information from the user program. Direct access is not possible.	_sNTP_RESULT		page 3-17
.ExecTime	NTP Last Operation Time	Gives the last time that NTP processing ended normally. The time that was obtained from the NTP server is stored when the time is obtained normally. The time is not stored if it is not obtained from the NTP server normally. **Note** Do not use this variable in the user program. There may be a delay in updating it. Use this variable only to access status through communications from an external device.	DATE_AND_TIME	Depends on data type.	page 3-17
.ExecNormal	NTP Operation Result	TRUE: Indicates an NTP normal end. FALSE: Indicates that NTP operation ended in an error or has not been executed even once. **Note** Do not use this variable in the user program. There may be a delay in updating it. Use this variable only to access status through communications from an external device.	BOOL	TRUE or FALSE	page 3-17

Additional Information

Relationships between the Target Node Information Tables

Registered Target Node Information (_EIP_RegTargetSta)	Registered Target Node Information Valid only when *_EIP_RegTargetSta* is TRUE	Normal Target Node Information Valid only when *_EIP_EstbTargetSta* is TRUE	Registered Target Node Information Valid only when *_EIP_RegTargetSta* is TRUE	Description
	Normal Target Node Information (_EIP_EstbTargetSta)	Target PLC Error Information (_EIP_EIP_TargetPLCErr)	Target Node Error Information (_EIP_TargetNodeErr)	
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
_EIP_TDLinkStartCmd	Tag Data Link Communications Start Switch	This is the start switch for data links.	BOOL	TRUE or FALSE	page 3-17
_EIP_TDLinkStopCmd	Tag Data Link Communications Stop Switch	This is the stop switch for data links.	BOOL	TRUE or FALSE	page 3-17

You can start and stop tag data links.

Additional Information

There is a system-defined variable for communications that is related to FINS/TCP* communications.

Variable name	Meaning	Description	Data type	Operation/refresh timing	Range of values	Default	R/W access	Related instructions
FINSTCP-ConnSta	FINS/TCP Connection Status	This is the FINS/TCP connection status.	WORD	Anytime	16#0000 to 16#FFFF	0	R	None

* You can use FINS message communications with NJ-series Controllers. However, not all memory areas in the NJ-series CPU Unit can be accessed. If you require this functionality, e.g., to connect to existing systems, consult with your OMRON representative.

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	This is the error status variable for the built-in EtherNet/IP port. It represents the following error flags. • *_EIP_PortErr* (Communications Port Error) • *_EIP_CipErr* (CIP Communications Error) • *_EIP_TcpAppErr* (TCP Application Communications Error) **Note** Refer to *Meanings of Error Status Bits* given later for the meaning of individual bits.				
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. • GetEIPError		

Variable name	_EIP_PortErr				
Meaning	Communications Port Error			Global/local	Global
Function	This is the error status variable for the communications port. It represents the following error flags. • *_EIP_MacAdrErr* (MAC Address Error) • *_EIP_LanHwErr* (Communications Controller Error) • *_EIP_EtnCfgErr* (Basic Ethernet Setting Error) • *_EIP_IPAdrCfgErr* (TCP/IP Basic Setting Error) • *_EIP_IPAdrDupErr* (IP Address Duplication Error) • *_EIP_BootpErr* (BOOTP Server Error) • *_EIP_IPRTblErr* (TCP/IP Advanced Setting Error) **Note** If a link OFF or Built-in EtherNet/IP Processing Error occurs, it is recorded in the event log and then corresponding bit turns ON. Refer to *Meanings of Error Status Bits* given later for the meaning of individual bits.				
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. • GetEIPError		

Variable name	_EIP_CipErr				
Meaning	CIP Communications Error			Global/local	Global
Function	This is the error status variable for CIP communications. It represents the following error flags. • *_EIP_IdentityErr* (Identity Error) • *_EIP_TDLinkCfgErr* (Tag Data Link Setting Error) • *_EIP_TDLinkOpnErr* (Tag Data Link Connection Failed) • *_EIP_TDLinkErr* (Tag Data Link Communications Error) • *_EIP_TagAdrErr* (Tag Name Resolution Error) • *_EIP_MultiSwOnErr* (Multiple Switches ON Error) **Note** If a Tag Resolution Error occurs, it is recorded in the event log and this variable changes to TRUE. Refer to *Meanings of Error Status Bits* given later for the meaning of individual bits.				
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. • 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 following error flags. • *_EIP_TcpAppCfgErr* (TCP/IP Setting Error) • *_EIP_NTPSrvErr* (NTP Server Connection Error) • *_EIP_DNSSrvErr* (DNS Server Connection Error) **Note** Refer to *Meanings of Error Status Bits* given later for the meaning of individual bits.				
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. • GetEIPError		

Variable name	_EIP_MacAdrErr				
Meaning	MAC Address Error			Global/local	Global
Function	Indicates that an error occurred when the MAC address was read 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	TRUE: The communications controller 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_EtnCfgErr				
Meaning	Basic Ethernet Setting Error			Global/local	Global
Function	TRUE: The Ethernet communications speed setting (Speed/Duplex) is incorrect. Or, a read operation 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	TCP/IP Basic Setting Error			Global/local	Global
Function	TRUE: • There is an illegal IP address setting. • A read operation failed. • The IP address obtained from the BOOTP server is inconsistent. • The DNS settings are not correct. 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	TRUE: The same IP address is assigned to more than one 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	_EIP_BootpErr				
Meaning	BOOTP Server Error			Global/local	Global
Function	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	TCP/IP Advanced Setting Error			Global/local	Global
Function	TRUE: There is an error in one of the following settings. Or, a read operation failed. • IP router table settings • Hosts settings 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_IdentityErr				
Meaning	Identity Error			Global/local	Global
Function	TRUE: The identity information (which you cannot overwrite) is not correct. Or, a read operation 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_TDLinkCfgErr				
Meaning	Tag Data Link Setting Error			Global/local	Global
Function	TRUE: The tag data link settings are incorrect. Or, a read operation 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_TDLinkOpnErr				
Meaning	Tag Data Link Connection Failed			Global/local	Global
Function	TRUE: The connection was not established because the remote node information in the tag data link parameters was different from the actual node information. **Note** This variable does not change to TRUE if there is no remote node when the power is turned ON. 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_TDLinkErr				
Meaning	Tag Data Link Communications Error			Global/local	Global
Function	TRUE: A timeout occurred in a tag data link connection. 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_TagAdrErr				
Meaning	Tag Name Resolution Error			Global/local	Global
Function	TRUE: Tag resolution failed (i.e., the address could not be identified from the tag name). The following causes are possible. • The size of the network-published variable does not agree with the tag setting. • 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 CPU Unit. • There is no network-published variable in the CPU Unit that corresponds to the tag setting. 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_MultiSwONErr				
Meaning	Multiple Switches ON Error			Global/local	Global
Function	TRUE: More than one data link start/stop switch changed to TRUE at the same time. 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_TcpAppCfgErr				
Meaning	TCP/IP Setting Error			Global/local	Global
Function	TRUE: At least one of the set values for a TCP/IP application (FTP, NTP, SNMP) is incorrect. Or, a read operation 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_NTPSrvErr				
Meaning	NTP Server Connection Error			Global/local	Global
Function	TRUE: The NTP client failed to connect to the server (timeout). FALSE: NTP is not set or the connection was successful.				
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	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.				
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	TRUE: The built-in EtherNet/IP port's communications can be used. (The link is ON and IP address is defined. Also, there are no errors). FALSE: The built-in EtherNet/IP port's communications is disabled due to an error in initial processing or restart processing.				
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	TRUE: At least one connection is in 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	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	---		

Variable name	_EIP_RegTargetSta [255]				
Meaning	Registered Target Node Information			Global/local	Global
Function	This variable gives a list of nodes for which built-in EtherNet/IP connections are registered. This variable is valid only when the built-in EtherNet/IP port is the originator. *Array[x]* is TRUE: The connection to the node with a target node ID of x is registered. *Array[x]* is FALSE: The connection to the node with a target node ID of x is not registered.				
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_EstbTargetSta [255]				
Meaning	Normal Target Node Information			Global/local	Global
Function	This variable gives a list of nodes that have normally established EtherNet/IP connections. *Array[x]* is TRUE: The connection to the node with a target node ID of x was established normally. *Array[x]* is FALSE: The connection to the node with a target node ID of x was not established, or an error occurred.				
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_TargetPLCModeSta [255]				
Meaning	Target PLC Operating Mode			Global/local	Global
Function	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 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. *Array[x]* is TRUE: This is the operating state of the target Controller with a node address of x. *Array[x]* is FALSE: Other than the above.				
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_TargetPLCErr [255]				
Meaning	Target PLC Error Information			Global/local	Global
Function	This variable shows the error status (logical OR of fatal and non-fatal errors) of the target node Controllers that are connected 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. *Array[x]* is TRUE: A fatal or non-fatal error occurred in the target Controller with a target node ID of x. *Array[x]* is FALSE: Other than the above.				
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				
Meaning	Target Node Error Information			Global/local	Global
Function	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 array elements are valid only when the Registered Target Node Information is TRUE. *Array[x]* 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. *Array[x]* 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).				
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			Members	.ExecTime
Meaning	NTP Last Operation Time			Global/local	Global
Function	Gives the last time that NTP processing ended normally. The time that was obtained from the NTP server is stored when the time is obtained normally. The time is not stored if it is not obtained from the NTP server normally. **Note** Do not use this variable in the user program. There may be a delay in updating it. Use this variable only to access status through communications from an external device.				
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			Members	.ExecNormal
Meaning	NTP Operation Result			Global/local	Global
Function	This variable shows if the NTP operation ended normally. TRUE: Indicates an NTP normal end. FALSE: Indicates that NTP operation ended in an error or has not been executed even once. **Note** Do not use this variable in the user program. There may be a delay in updating it. Use this variable only to access status through communications from an external device.				
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	This is the start switch for data links.				
Data type	BOOL			Range of values	TRUE or FALSE
R/W access	RW	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	This is the stop switch for data links.				
Data type	BOOL			Range of values	TRUE or FALSE
R/W access	RW	Retained	Not retained.	Network Publish	Published.
Usage in user program	Possible.	Related instructions	---		

4

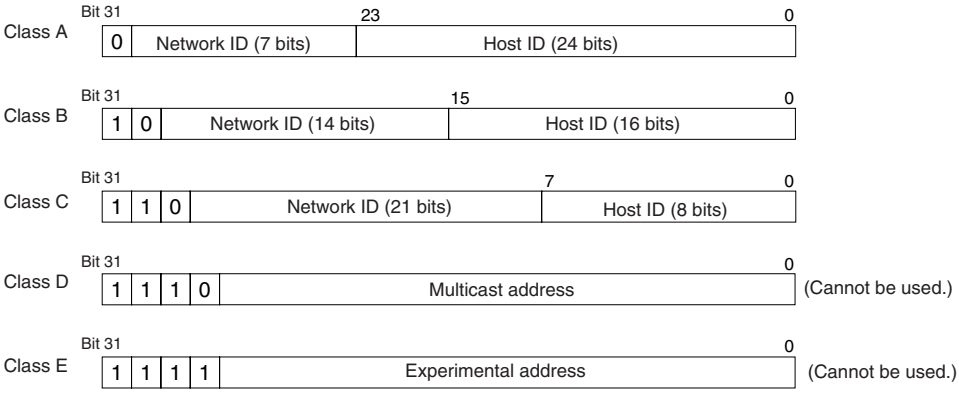
Determining IP Addresses

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 4-3
4-2 Built-in EtherNet/IP Port IP Address Settings 4-4
4-2-1 Determining IP Addresses 4-4
4-2-2 Setting IP Addresses 4-5
4-2-3 Online Connection 4-7
4-2-4 Checking the Current IP Address 4-9
4-3 Private and Global Addresses 4-10
4-3-1 Private and Global Addresses 4-10
4-3-2 Using a Private Address for the Built-in EtherNet/IP Port 4-11
4-3-3 Using a Global Address for the Built-in EtherNet/IP Port 4-12

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.

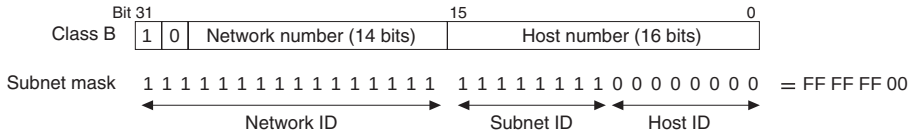
4-1-2 Allocating IP Addresses

You must assign IP addresses nodes so that each IP address is assigned only once in the network or between several networks.

4-1-3 Subnet Masks

Operation and management of a network can become very difficult if too many nodes are connected on a single network. In such a case it can be helpful to configure the system so that a single network is divided up into several subnetworks. Internally the network can be treated as a number of subnetworks, but from the outside it acts as a single network and uses only a single network ID. To establish subnetworks, the host ID in the IP address is divided into a subnet ID and a host ID by using a setting called the subnet mask. The subnet mask indicates which part of the host ID is to be used as the subnet ID. All bits in the subnet mask that correspond to the bits in the IP address used either as the network ID or subnet ID are set to "1," and the remaining bits, which correspond to the bits in the IP address actually used for the host ID, are set to "0."

The following example shows the subnet mask for an 8-bit subnet ID used in class-B IP addresses.



Set the same subnet mask for all of the nodes on that subnetwork. The built-in EtherNet/IP port supports CIDR (Classless Inter-Domain Routing). The subnet mask can be set to 192.0.0.0 to 255.255.255.252. If subnetworks are not used, set the following subnet mask values for IP address classes A to C.

Class	Subnet mask
Class A	255.0.0.0
Class B	255.255.0.0
Class C	255.255.255.0

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 Built-in EtherNet/IP Port IP Address Settings

4-2-1 Determining IP Addresses

Use one of the following methods to set the IP address of the built-in EtherNet/IP port.

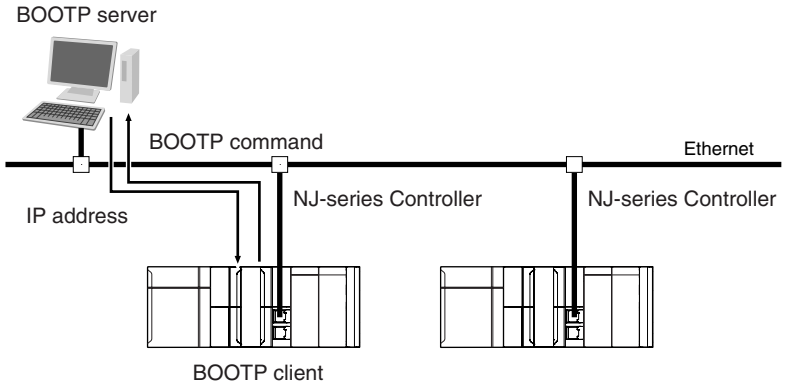
Setting a User-specified IP Address

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.

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.

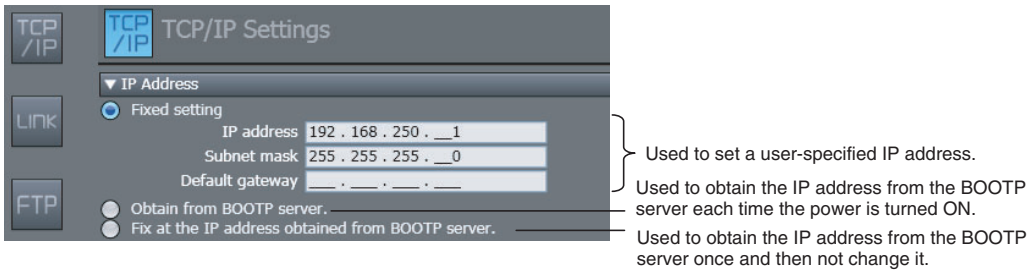


4-2-2 Setting IP Addresses

Use the Sysmac Studio to set the IP address of the built-in EtherNet/IP port.

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.



2 Connect the Sysmac Studio to the NJ-series CPU Unit via a USB connection or the Ethernet network.

3 Connect the Sysmac Studio online to the NJ-series CPU Unit. 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 NJ-series CPU Unit.

- 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 CPU Unit as follows:

Setting a User-specified IP Address

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

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 CPU Unit. Each time the power supply is turned ON, the IP address from the BOOTP server is automatically saved in the CPU Unit.

Additional Information

If you cannot obtain the IP address from the BOOTP server or the obtained IP address is not correct, select the *Fixed setting* Option in the IP Address Area and manually set the IP address, subnet mask, and default gateway. Requests to the BOOTP server to obtain the IP address will continue if there is a failure to connect to the BOOTP server.

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.

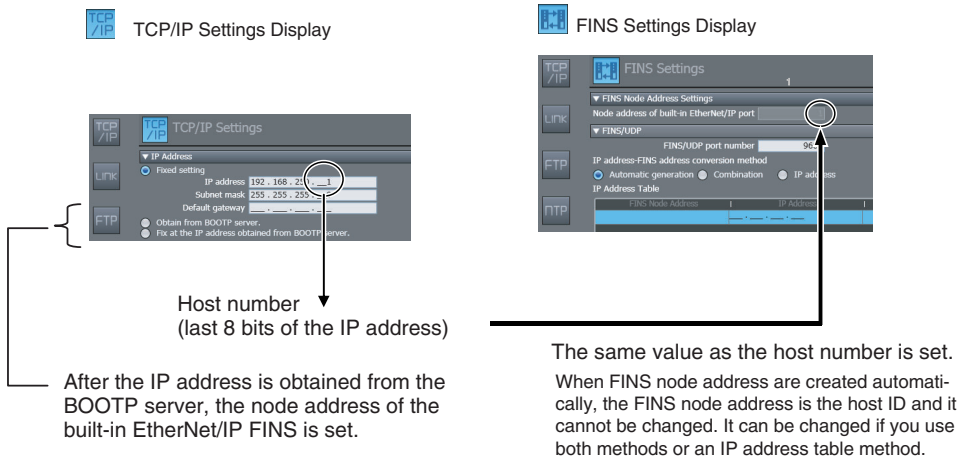
Additional Information

- 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 NJ-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.
- 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.

Additional Information

When the local IP address of the built-in EtherNet/IP is set, the FINS node address is automatically set as follows:

Example: Pairing IP Addresses and FINS Node Addresses with Automatic Creation Method



The FINS node address is required for FINS communications (e.g., to connect to the CX-Integrator and other Support Software).

Note You can use FINS message communications with NJ-series Controllers. However, not all memory areas in the NJ-series CPU Unit can be accessed. If you require this functionality, e.g., to connect to existing systems, consult with your OMRON representative.

4-2-3 Online Connection

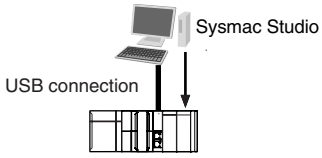
Connect the Sysmac Studio online to the NJ-series CPU Unit.

Types of Connections between the CPU Unit and Computer That Runs the Sysmac Studio

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

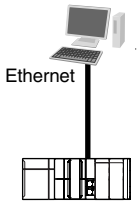
USB Connection

USB Direct Connection



Ethernet Connection

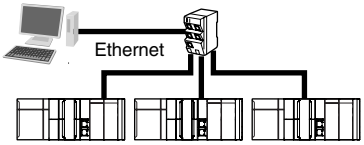
1:1 Ethernet Direct Connection through AutoIP



Note 1 A Ethernet switch is not necessarily required.

2 You can use a straight or cross Ethernet cable to connect.

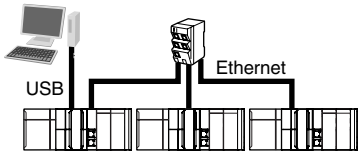
1:N Ethernet Switch Connection



Note A Ethernet switch is required to connect. Refer to *2-1-5 Precautions for Ethernet Switch Selection* for details.

Connection from USB Across Ethernet

1:N USB Remote Connection



Note A Ethernet switch is required to connect. Refer to *2-1-5 Precautions for Ethernet Switch Selection* for details.

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.

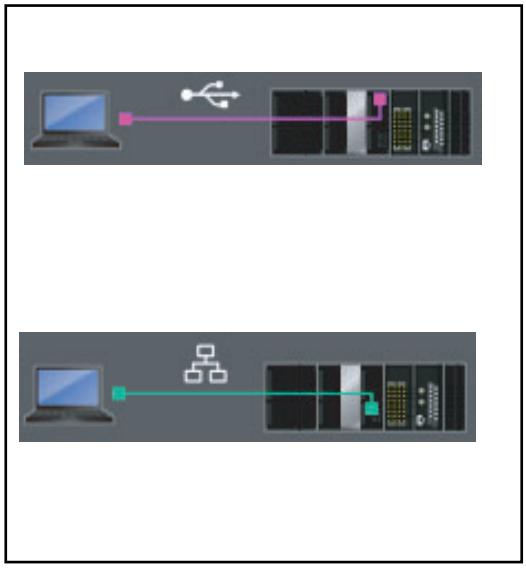
Online Connection Procedure

Connect the CPU Unit and the computer that runs the Sysmac Studio via USB or Ethernet, and then perform the following procedure.

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

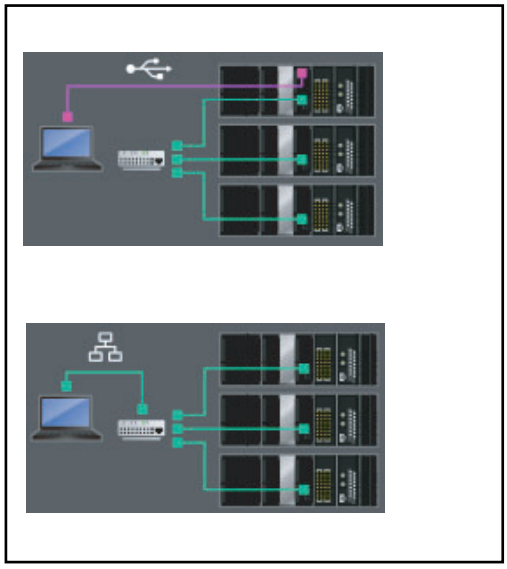
1:1 Connection

Direct Connection



1:N Connection

EtherNet/IP Connection



Additional Information

- If there is an error in the set IP address, the NJ-series CPU Unit behaves as follows:
  - The NET RUN indicator on the CPU Unit does not light and the NET ERR indicator flashes red.
  - A TCP/IP Basic Settings Error is recorded in the event log.

If you use a USB direct connection or an Ethernet direction connection, it is possible to reset the IP address because the Sysmac Studio can connect to the CPU Unit via the built-in EtherNet/IP port.

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

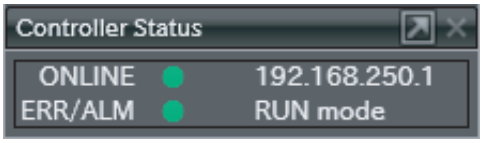
Connecting from a Saved Project

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

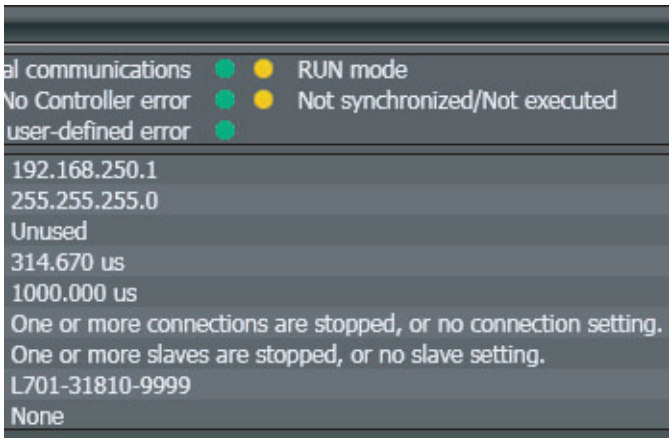
4-2-4 Checking the Current IP Address

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 change the IP address setting from the *Obtain from BOOTP server.* Option to the *Fix at the IP address obtained from BOOTP server* Option and synchronize the Controller Setup from the Sysmac Studio, you can check the IP address that was obtained from the BOOTP server.

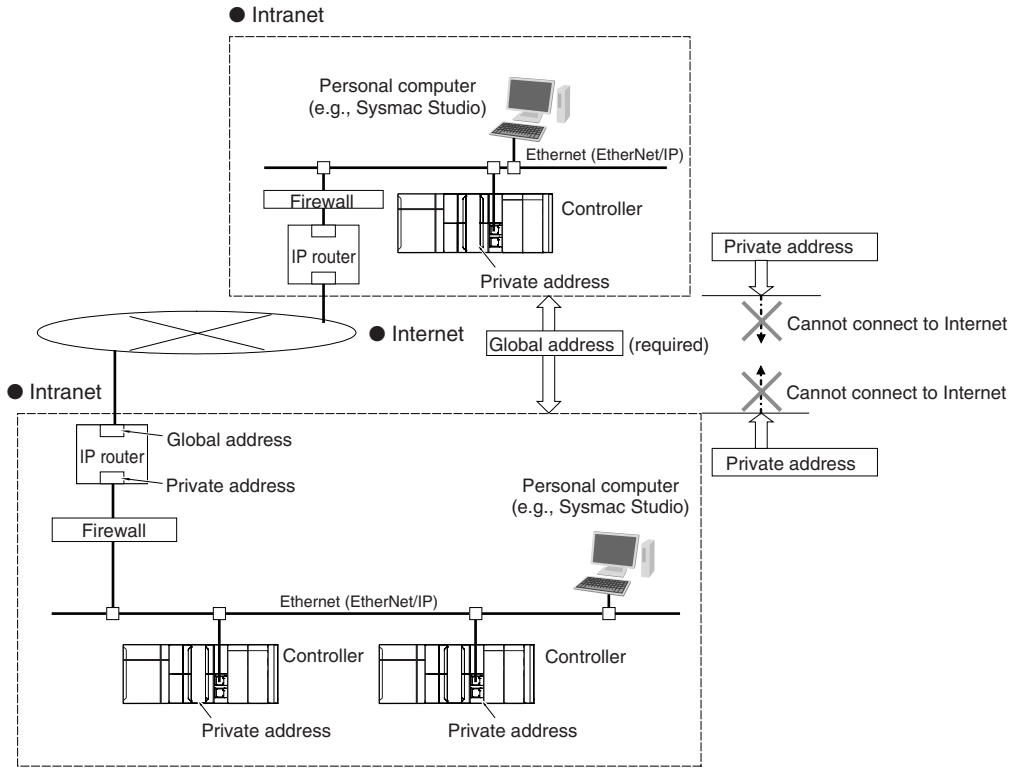
4-3 Private and Global Addresses

4-3-1 Private and Global Addresses

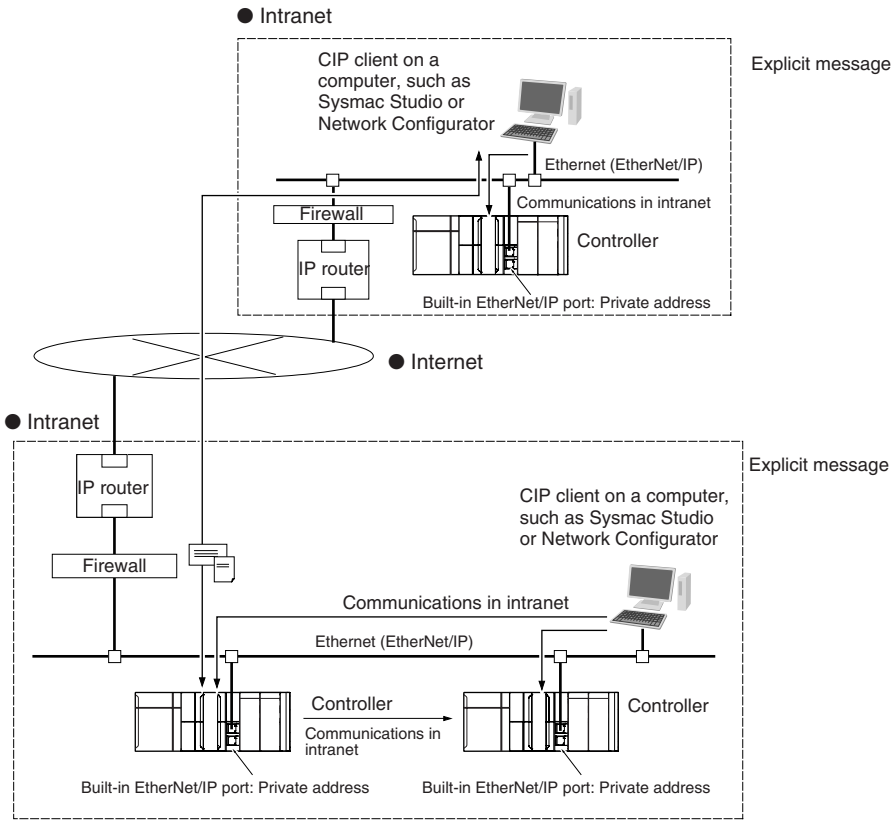
There are two kinds of IP addresses, private and global.

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.

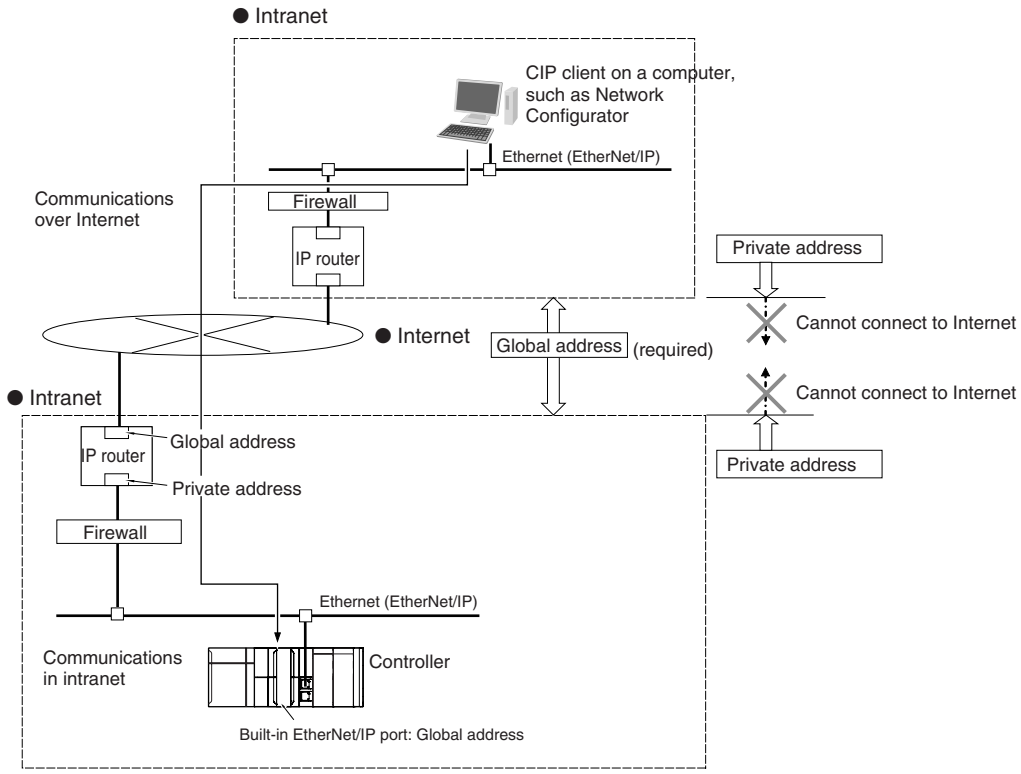
- The explicit message communications service can be executed on the intranet between built-in EtherNet/IP ports with private addresses only.
- A device such as a personal computer (CIP applications including the Network Configurator) cannot connect online and communicate over the Internet with a built-in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

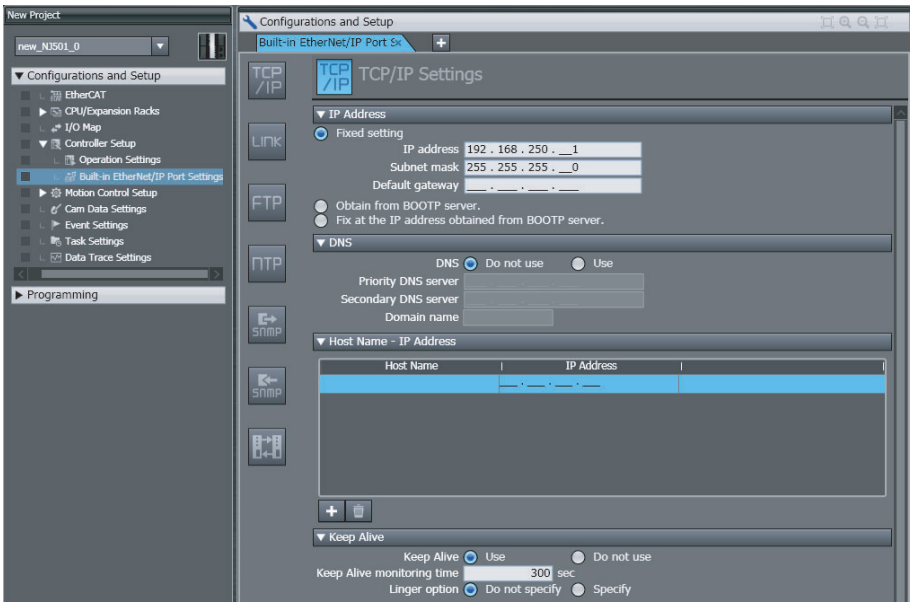
To set a global IP address for a built-in EtherNet/IP port involves network security considerations. It is recommended that the user contract with a communications company for a dedicated line, rather than for a general line such as a broadband line. Also, be sure to consult with a network specialist 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 Display 5-2
5-2 Link Settings Display 5-4
5-3 FTP Display 5-5
5-4 NTP Settings Display 5-6
5-5 SNMP Settings Display 5-7
5-6 SNMP Trap Settings Display 5-9

5-1 TCP/IP Display



● Setting the IP Addresses

Setting	Description	Default
IP Address	Select the setting method for the IP address for the built-in EtherNet/IP port. • Fixed setting • Obtain from BOOTP server. • Fix at the IP address obtained from BOOTP server.	Fixed setting
IP address*	Set the IP address for the built-in EtherNet/IP port.	192.168.250.1
Subnet mask*	Set the subnet mask for the built-in EtherNet/IP port.	255.255.255.0
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.	None

* These settings are required if you select *Fixed setting*.

● DNS

Setting	Description	Default
DNS	Set whether to specify the host name in the DNS server in CIP communications and socket instructions to specify the remote communications node instead of the IP address. A DNS server is required to use DNS. Use/Do not use.	Do not use.
Priority DNS server*	Set the IP address of the DNS server. You can set priority and secondary IP addresses.	None
Secondary DNS server		None
Domain name*	Sets the domain name of the domain to which the built-in EtherNet/IP port belongs. The built-in EtherNet/IP port does not use a domain name in actual communications. (Single-byte alphanumeric characters, dots, and hyphens: 48 characters max.)	None

* These settings are required when DNS is used.

● Hosts Settings

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

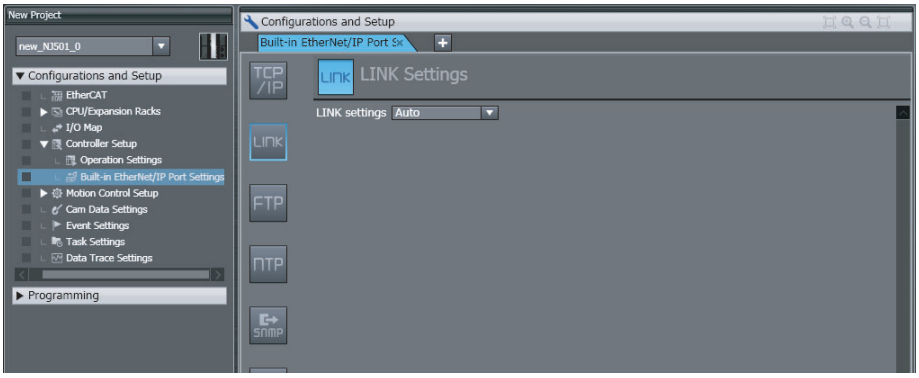
● Keep Alive

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 FINS/TCP) for each connection number. If *Use* is selected here and the remote node does not respond during the set 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. • Use. • Do not use.	Use.
Keep Alive Monitoring Time	The connection is disconnected if the keep-alive function is enabled and the remote node does not respond during the monitoring time set in the Keep Alive Monitoring Setting. Setting range: 1 to 65,535 (seconds)	300
Linger option	Set whether to specify the Linger Option for connections to FINS/TCP or socket services. If the Linger Option is specified, the port number is immediately opened even before the port number is released after the socket closes (approx. 1 minute). • Specify. • Do not specify.	Do not specify.

● IP Router Table

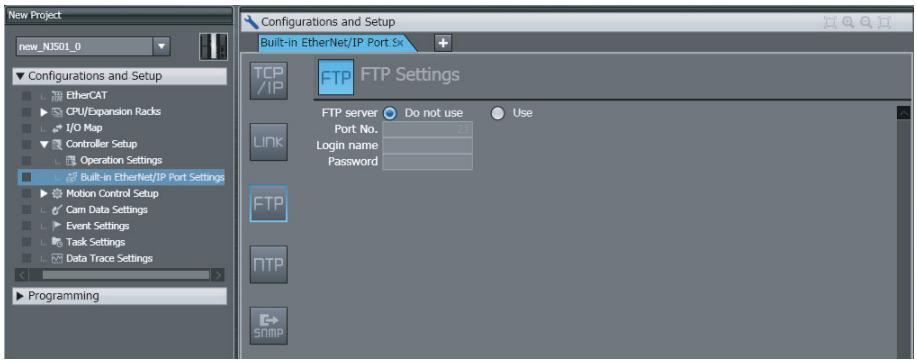
Parameter	Description	Default
Destination IP address	Set these settings when the built-in EtherNet/IP port communicates with nodes on other IP network segments via an IP router. You can set up to eight combinations of IP addresses and gateway addresses. Specify 0 for the host portions of the IP addresses.	None
Destination mask IP address		
Gateway address		None

5-2 Link Settings Display



Setting	Description	Default
LINK settings	Set the baud rate for the built-in EtherNet/IP port. • Auto • 10 Mbps Half Duplex • 10 Mbps Full Duplex • 100 Mbps Half Duplex • 100 Mbps Full Duplex	Auto

5-3 FTP Display

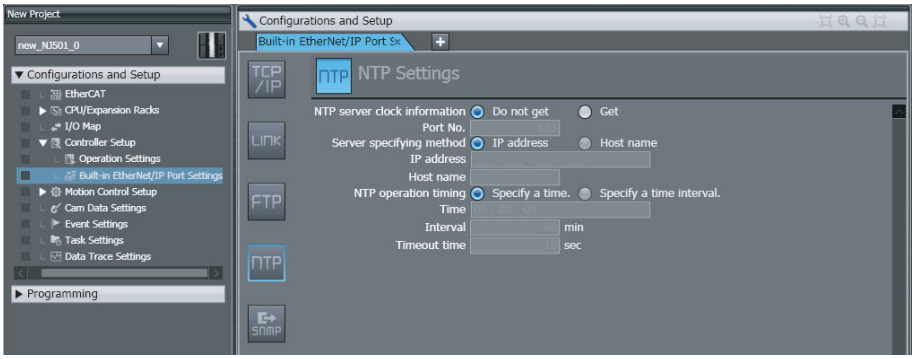


Setting	Description	Default
FTP server	Specify whether to use the FTP server. FTP connections from external devices will not be possible if the *Do not use* Option is selected.	Do not use.
Port No.*1	Set the FTP port number of the built-in EtherNet/IP port. It is normally not necessary to change this 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.

5-4 NTP Settings Display

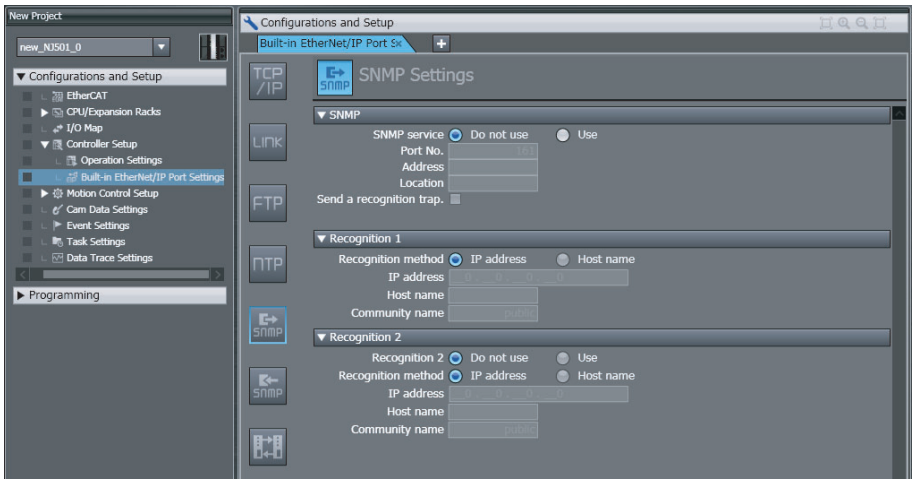


Setting	Description	Default
NTP server clock information	Set whether to obtain clock information from the NTP server to update the clock in the CPU Unit.	Do not get.
Port No.*1*2	Set the port number to use to connect to the NTP server to obtain clock information. It is normally not necessary to change this setting.	123
Server specifying method*1	Set the method to use to specify the NTP server to obtain clock information. • IP address • Host name	IP address
IP address	Set the IP address of the NTP server. (Set this setting if the server specifying method is set to the *IP address* Option.)	None
Host name	Set the host name of the NTP server (i.e., the domain name of the host). (Set this setting if server specifying method is set to the *Host name* 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
NTP operation timing*1	Set the time at which the NTP server is accessed to synchronize the clocks. • Specify a clock time. • Specify a time interval.	Specify a time.
Time [hours:minutes:seconds]	The NTP server is accessed at the specified time. (Setting range: 00:00:00 to 23:59:59) (Set this setting if NTP operation timing is set to the *Specify a clock time* Option.)	00:00:00
Interval [minutes]	The NTP server is accessed when the specified period of time has passed. (Setting range: 1 to 1,440 minutes) (Set this setting if NTP operation timing is set to the *Specify a time interval* Option.)	60 min
Timeout time (seconds)*1	Set the timeout detection time. (Setting range: 1 to 255 seconds) If the remote host does not respond, retry processing is performed four times within the time interval that is set here. If a time interval is specified for NTP operation, timing for the next execution of the NTP operation starts when the fourth retry processing times out.	10 seconds

*1 This setting is required to get the clock information from the NTP server.

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

5-5 SNMP Settings Display



● SNMP

Setting		Description	Default
SNMP		Specify whether to use the SNMP monitor service. If not using the SNMP monitor service is specified, an SNMP manager cannot connect from an external device.	Do not use.
	Port No.	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 *Send a recognition trap* 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 *Send a recognition trap*, specify the SNMP trap settings on the SNMP Trap Tab Page.	Do not use.

● Recognition 1*

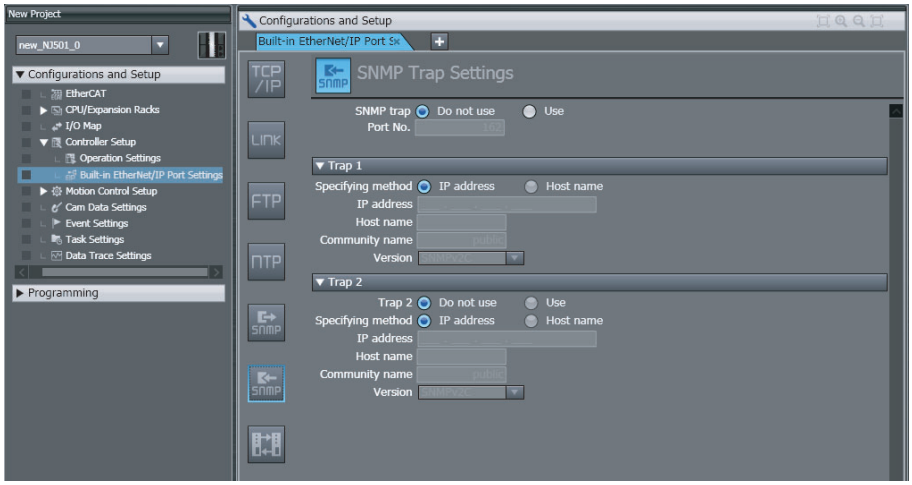
Setting	Description	Default
Recognition method	Set the method to use to specify SNMP managers for which access is permitted. • IP address • Host name Make these settings to permit access by only certain SNMP managers. Access is not allowed unless an IP address or host name is set.	IP address
IP address	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 *IP address* 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 *Host name* 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. • Use. • Do not use.	Do not use.
	Recognition method	Set the method to use to specify SNMP managers for which access is permitted. • 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 *IP address* 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 *Host name* 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

* These settings are required if the SNMP service is enabled.

5-6 SNMP Trap Settings Display



● SNMP Trap

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

● Trap 1*

Setting	Description	Default
Specifying method	Set the specifying method for the SNMP manager destination for SNMP traps. • 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 *IP address* 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 *Host name* 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. • SNMP version 1: SNMPv1 • SNMP version 2C: SNMPv2C	SNMPv1

● Trap 2*

Setting		Description	Default
Trap 2		Specify whether to use the trap 2 settings. • Use. • Do not use.	Do not use.
	Specifying method	Set the specifying method for the SNMP manager destination for SNMP traps. • 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 *IP address* 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 *Host name* 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. • SNMP version 1: SNMPv1 • SNMP version 2C: SNMPv2C	SNMPv1

* These settings are required if the SNMP trap is set is enabled.

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

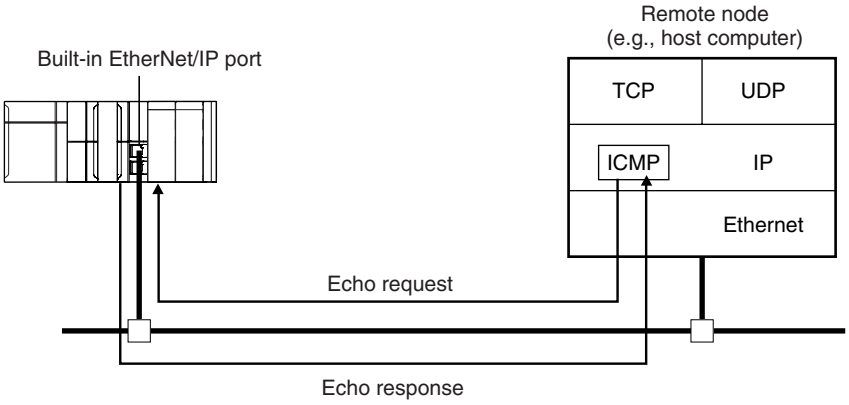
6-1-3 Host Computer Operation 6-3

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

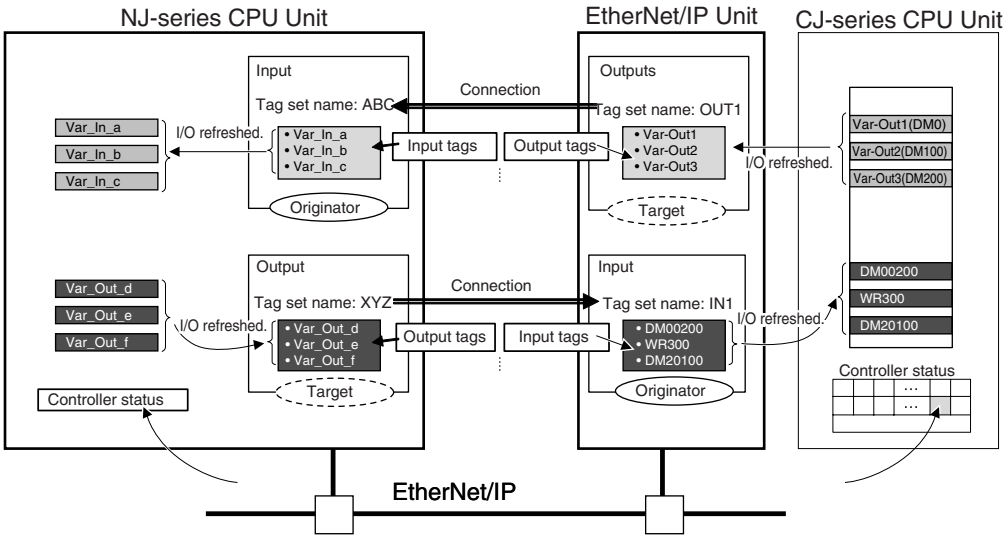
7-1 Introduction to Tag Data Links 7-2
7-1-1 Tag Data Links 7-2
7-1-2 Data Link Data Areas 7-3
7-1-3 Tag Data Link Functions and Specifications 7-5
7-1-4 Overview of Operation 7-6
7-1-5 Starting and Stopping Tag Data Links 7-8
7-1-6 Controller Status 7-8
7-1-7 Concurrency of Tag Data Link Data 7-10
7-2 Setting Tag Data Links 7-14
7-2-1 Starting the Network Configurator 7-14
7-2-2 Tag Data Link Setting Procedure 7-16
7-2-3 Registering Devices 7-17
7-2-4 Creating Tags and Tag Sets 7-18
7-2-5 Connection Settings 7-28
7-2-6 Connecting the Network Configurator to the Network 7-38
7-2-7 Downloading Tag Data Link Parameters 7-43
7-2-8 Uploading Tag Data Link Parameters 7-46
7-2-9 Verifying the Tag Data Links 7-49
7-2-10 Starting and Stopping Tag Data Links 7-52
7-2-11 Clearing the Device Parameters 7-54
7-2-12 Saving the Network Configuration File 7-55
7-2-13 Reading a Network Configuration File 7-57
7-2-14 Checking Connections 7-58
7-2-15 Changing Devices 7-60
7-2-16 Displaying Device Status 7-61
7-3 Ladder Programming for Tag Data Links 7-62
7-3-1 Ladder Programming for Tag Data Links 7-62
7-3-2 Status Flags Related to Tag Data Links 7-66
7-4 Tag Data Links with Models Other than NJ-Series CPU Units 7-67

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. (You can also assign I/O memory addresses 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.

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-2 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.

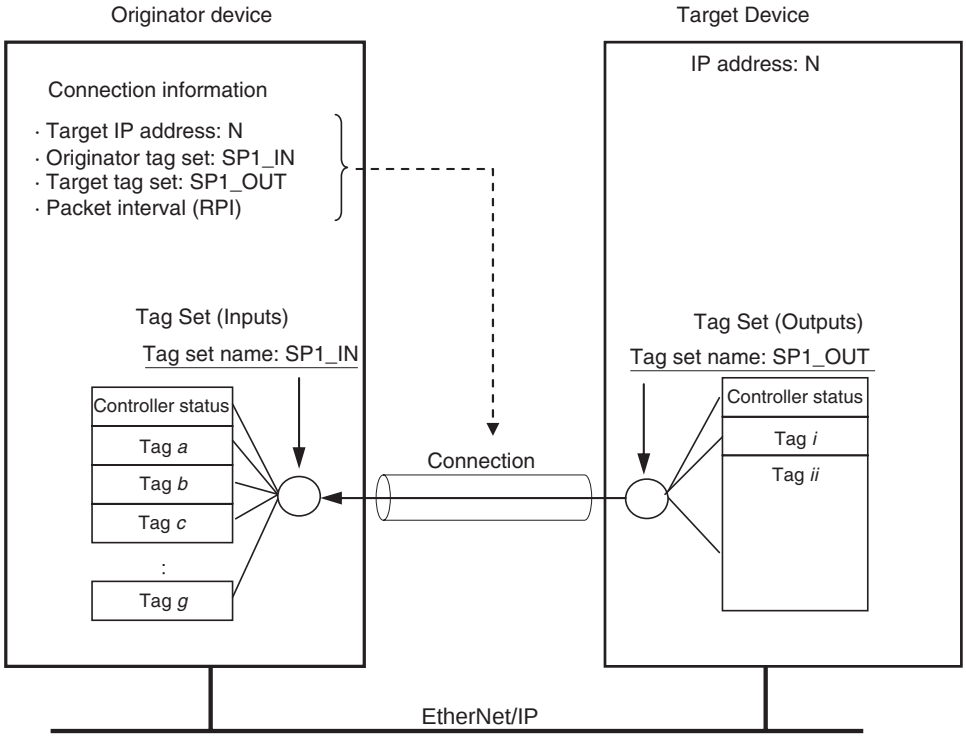
Tag Sets

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.

● 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.



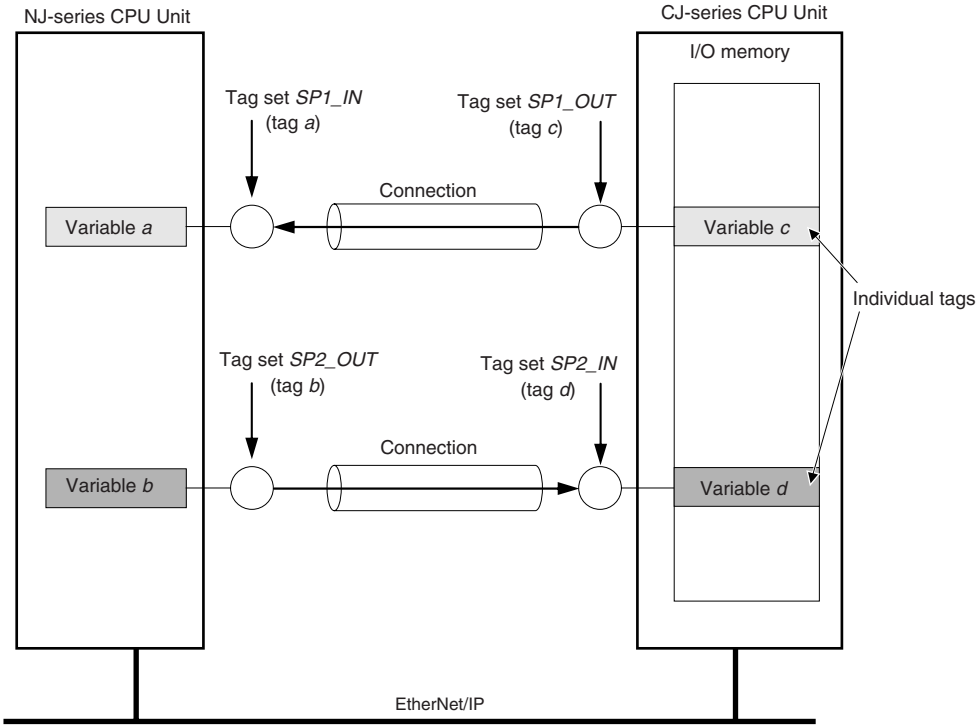
There are both input (consume) and output (produce) tag sets. Each tag set can contain only input tags or only output tags. The same input tag cannot be included in more than one input tag set.

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.

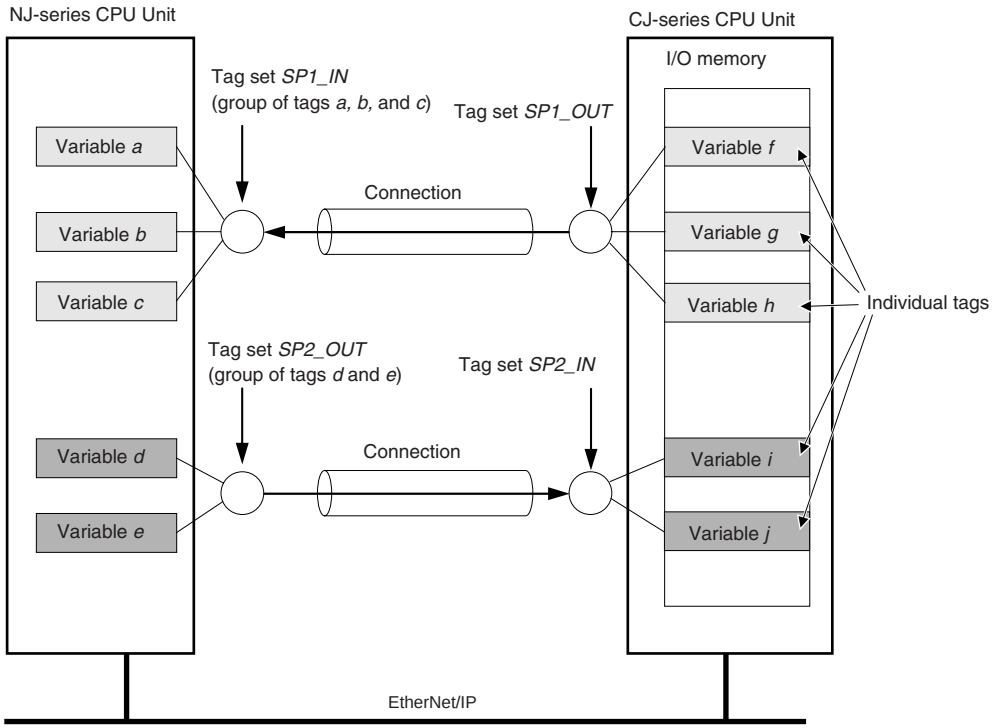
Tag Sets with Only One Tag Each

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



Tag Sets with Multiple Tags Each

As shown below, tags can be grouped. You can place up to eight tags (with a total of up to 300 words) 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 NJ-series CPU Unit are given below.

Item		Specification
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 Units are restarted to start the tag data links.
Tags*1	Supported variable types	You can specify the following network variables as tags. *2, *3 • Global variables
	Maximum number of words per tag	300 words (600 bytes)
	Maximum number of tags	256*4
Tag sets	Maximum number of tags per tag set	8 (7 when Controller status is included)
	Maximum number of words per tag set	300 words (600 bytes)
	Maximum number of tag sets	32
Connections		Maximum number of connections per Unit: 32
Connection type		Each connection can be set for 1-to-1 (unicast) or 1-to-N (multi-cast) communications.
Packet interval (RPI)		10 to 10,000 ms in 1-ms increments The packet interval can be set separately for each connection.

*1 To specify a specific I/O memory address for a tag, create a variable, use 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.

*2 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.

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

Type		Example	Specification	Remarks
Variables with basic data types		**aaa**	Supported	
Array variables	**Arrays**	**bbb**	Supported	
	Elements	**bbb[2]**	Supported	
Structure variables	**Structures**	**ddd**	Supported	
	Member	**ddd.xxx**	Supported	

*4 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.

* If a variable that uses an AT specification is used as a tag, you do not need to set a refreshing task. It is refreshed in the primary periodic task.

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_ (Example: EA_100)

Setting and Downloading Tag Data Link Parameters (Network Configurator)

The tag data link parameters (e.g., connection information) that are described below are created with the Network Configurator, 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.

1 Creating the Configuration Information

You can register the EtherNet/IP ports and EtherNet/IP Units to create the connections that define the tag data links. Refer to the *7-2-3 Registering Devices* for details.

2 Setting Tags

Create CPU Unit variables for input (consume) tags and output (produce) tags. You can create up to 256 tags for a built-in EtherNet/IP port on an NJ-series CPU Unit. (There is a maximum data size of 600 bytes (300 words) for each tag.) 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 CPU Unit.

3 Setting Tag Sets

You can create output tag sets and input tag sets and assign tags to them. (You can create a total of up to eight I/O tag sets.) You can create up to 32 tag sets for a built-in EtherNet/IP port on an NJ-series CPU Unit. (The maximum data size of 1 tag set is 600 bytes (300 words).) You can specify the Controller status in a tag set to indicate the CPU Unit's operating status (operating information and error information).

4 Setting Connections

The target device output tag set and the originator device input tag set are associated as connections. You can open a total of up to 32 connections for the built-in EtherNet/IP port.

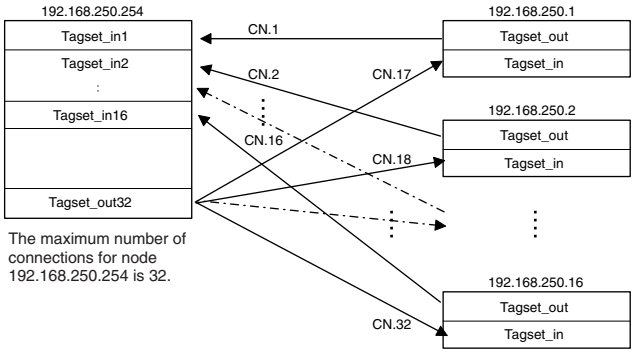
Additional Information

Counting Connections

The number of connections is the total of the number of input tag sets that receive data and the number of nodes that send data for output tag sets. (Refer to the following figure.) One connection is consumed for each connection setting whether the connection is a multi-cast connection or a unicast (point-to-point) connection.

Example of Calculating the Number of Connections

Example for built-in EtherNet/IP port with IP address of 192.168.250.254 in bidirectional connection with 16 nodes



Setting the Requested Packet Interval (RPI)

The RPI is the data I/O refresh cycle on the Ethernet line when performing tag data links. The RPI can be set separately for each connection. You can set the built-in EtherNet/IP port to between 10 and 10,000 ms (in 1-ms increments). The default setting is 50 ms. With EtherNet/IP, data is exchanged on the communications line at the packet interval that is set for each connection, regardless of the number of nodes.

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. If multi-cast connections are used, however, use a Ethernet switch that has multi-cast filtering, unless the tag set is received by all nodes in the network. If a 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. 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

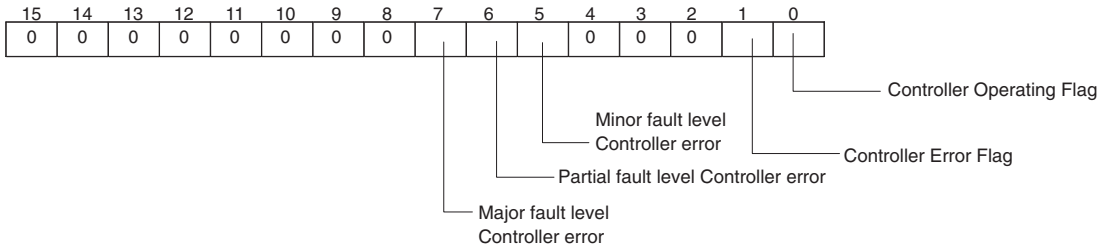
The performance of communications devices is limited to some extent by the limitations of each product's specifications. Consequently, there are limits to the packet interval (RPI) settings. Refer to *13-2 Adjusting the Communications Load* and set an appropriate packet interval (RPI).

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 NJ-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 *7-2-10 Starting and Stopping Tag Data Links* for details.

7-1-6 Controller Status

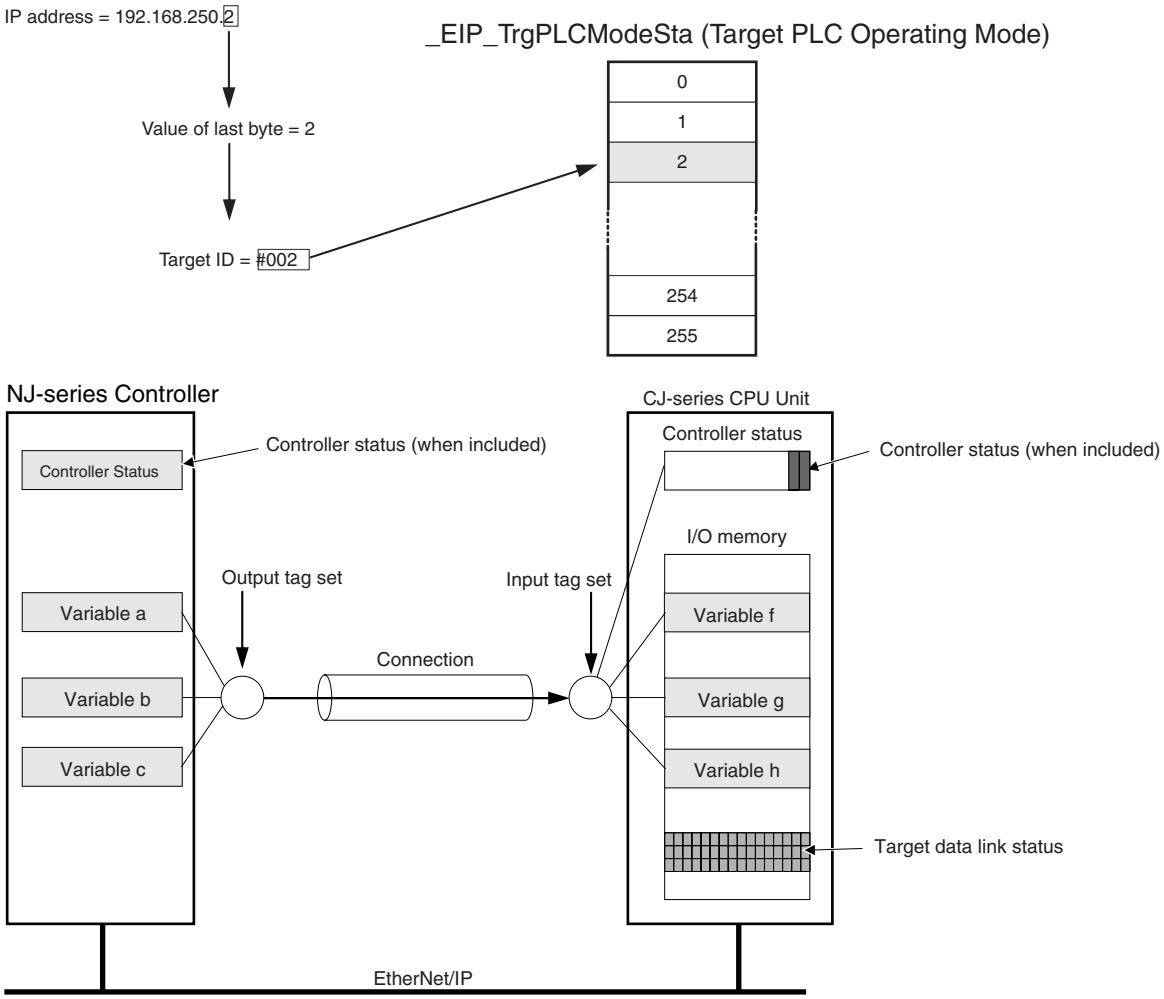
You can include the Controller status as a member of a tag set in the data sent and received. The Controller status is a set of flags that indicate the operating status of the CPU Unit (operating information, error information, Controller error level). If the Controller status is specified as an output (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.)



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.

- _EIP_TargetPLCModeSta (Target PLC Operating Mode)
- _EIP_TargetPLCErr (Target PLC Error Information)

Example: Sending 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	_EIP_TargetPLCModeSta (Target PLC Operating Mode)	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	_EIP _TargetPLCErr (Target PLC Error Information)	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 *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).

● 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 three conditions.

(1) The tags must be assigned to the same tag set (connection).

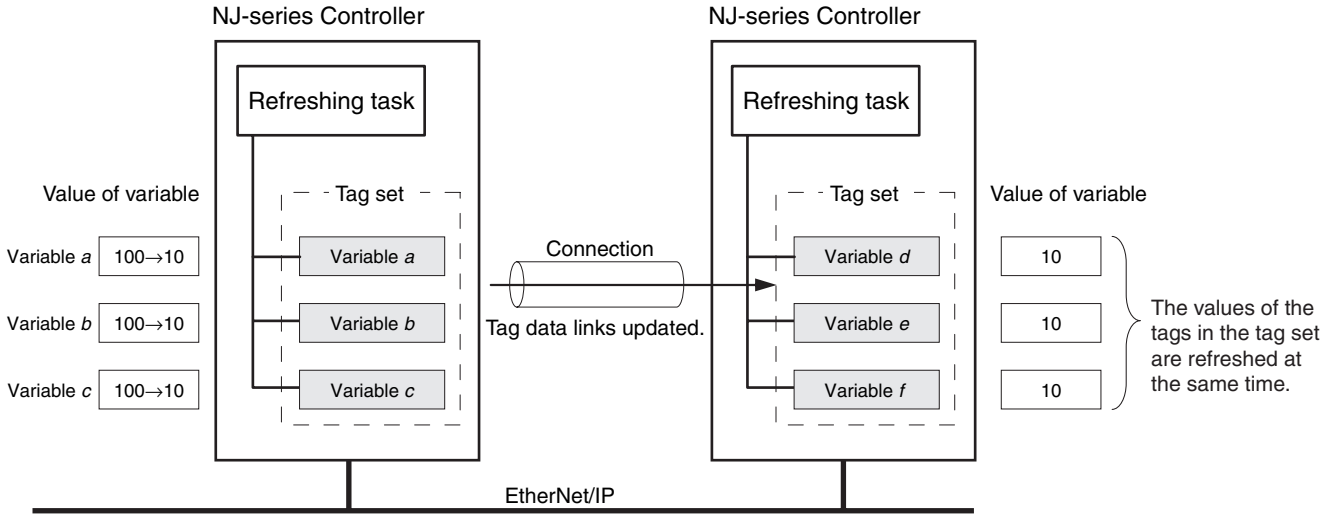
(2) A refreshing task must be set for the network variables assigned to the tags, and the refreshing task must be the same for all tags in the tag set.*

(3) A tag that uses AT specification must not be included in the same tag set.

(4) 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.

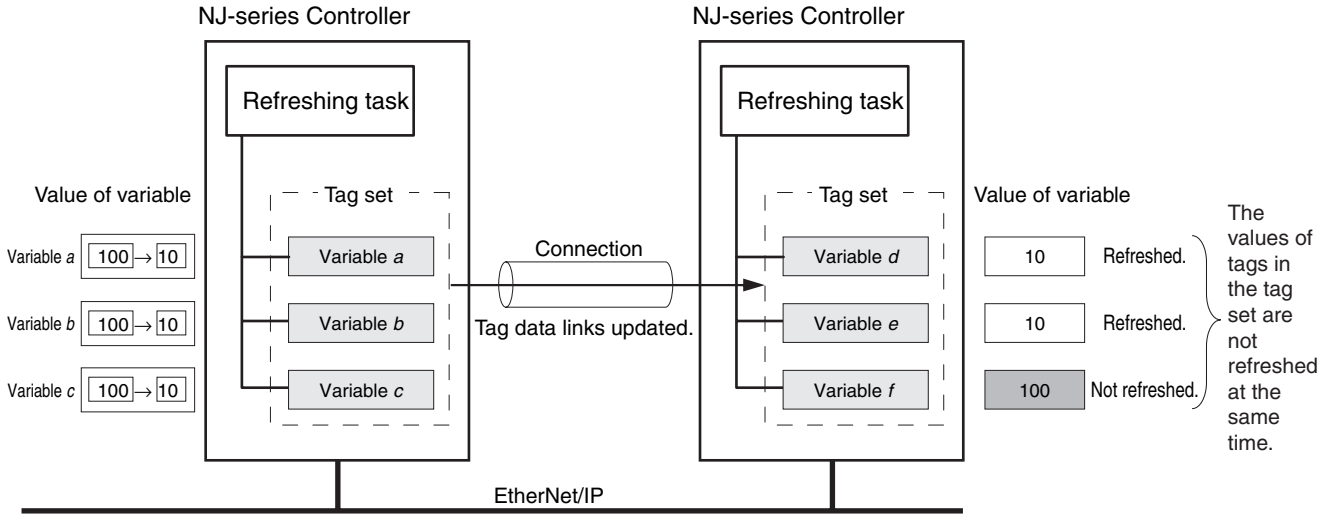
- Setting Refreshing Tasks for Tags (Network Variables)
  Concurrency of the tags in the tag set is maintained.



Additional Information

You do not need to set a refreshing task for variables (tags) that use an AT specification because they are updated in the primary periodic task.

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

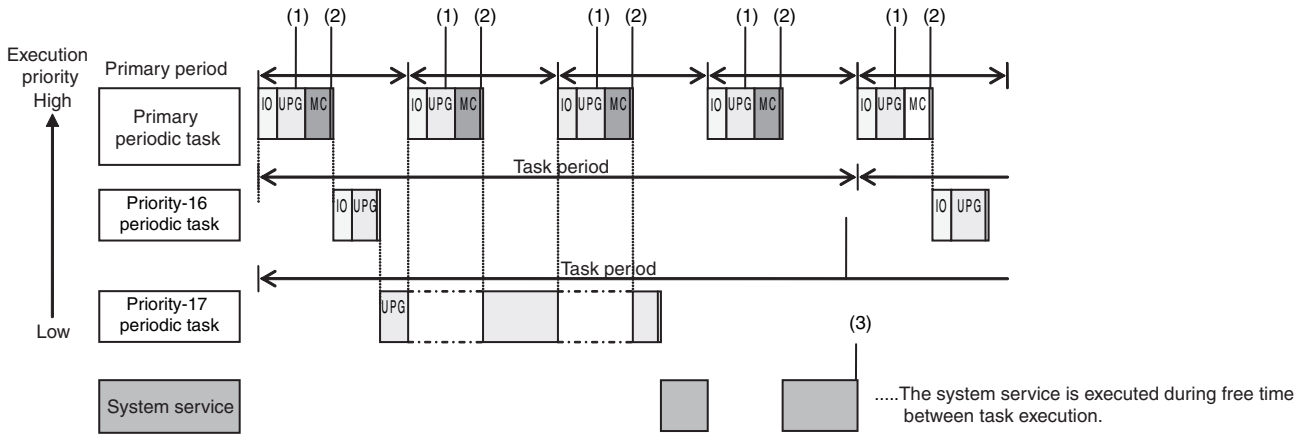


● The timing of updating network variables that are assigned to tags is synchronized with the execution period of the program that accesses the network variables.

Set the refreshing task for the network variables assigned to the tags to the task that contains the program that accesses those network variables.

- Difference between the Operation of Tags with a Refreshing Task and Tags without a Refreshing Task
  When you set a refreshing task for tags (network variables) that is the same as the task that contains the program that accesses them, those tags are refreshed at the same time as the execution of the program. Refreshing of tags (network variables) that have no refreshing task is handled by the system service with the lowest execution priority, and therefore it is not synchronized with the execution of the program.

(1) Execution timing of the program
(2) Refresh timing of network variables (tags) with the primary periodic task* set as the refreshing task*
(3) Refresh timing of network variables (tags) that do not have the primary periodic task set as the refreshing task



*: Refreshed during system common processing 2 in the task processing.

Additional Information

If a program needs to access a network variable with an AT specification, set the program in the primary periodic task so that it matches the refresh timing of the network variable that uses an AT specification.

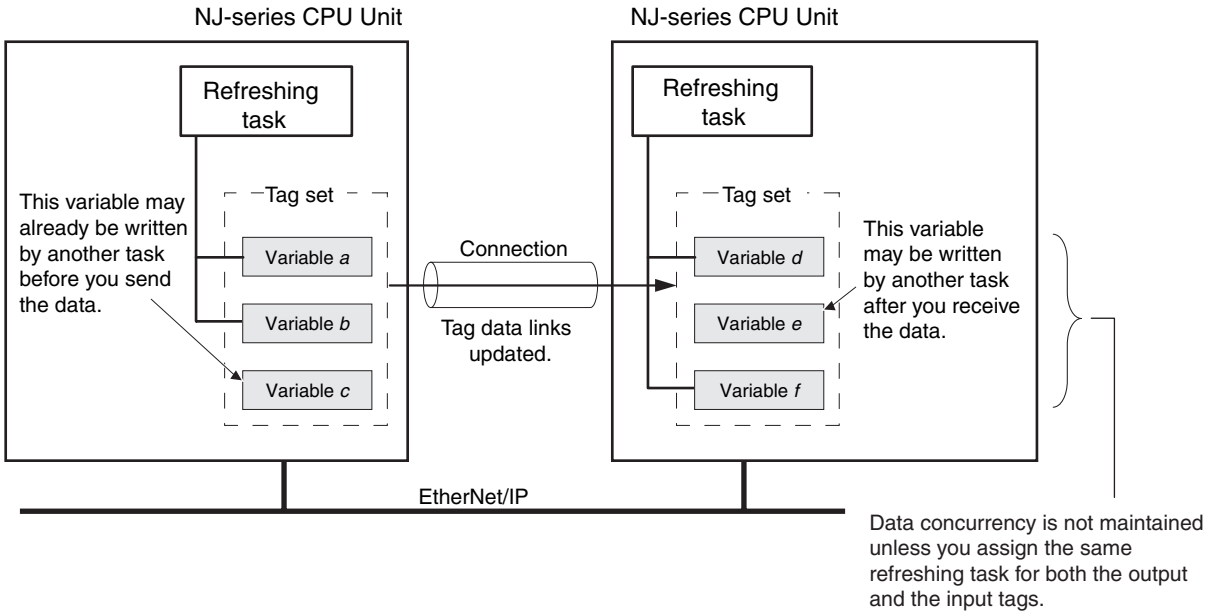
Additional Information

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 CPU Unit and the input CPU Unit.



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 NJ-series CPU Unit Software User's Manual (Cat. No. W501).

7-2 Setting Tag Data Links

7-2-1 Starting the Network Configurator

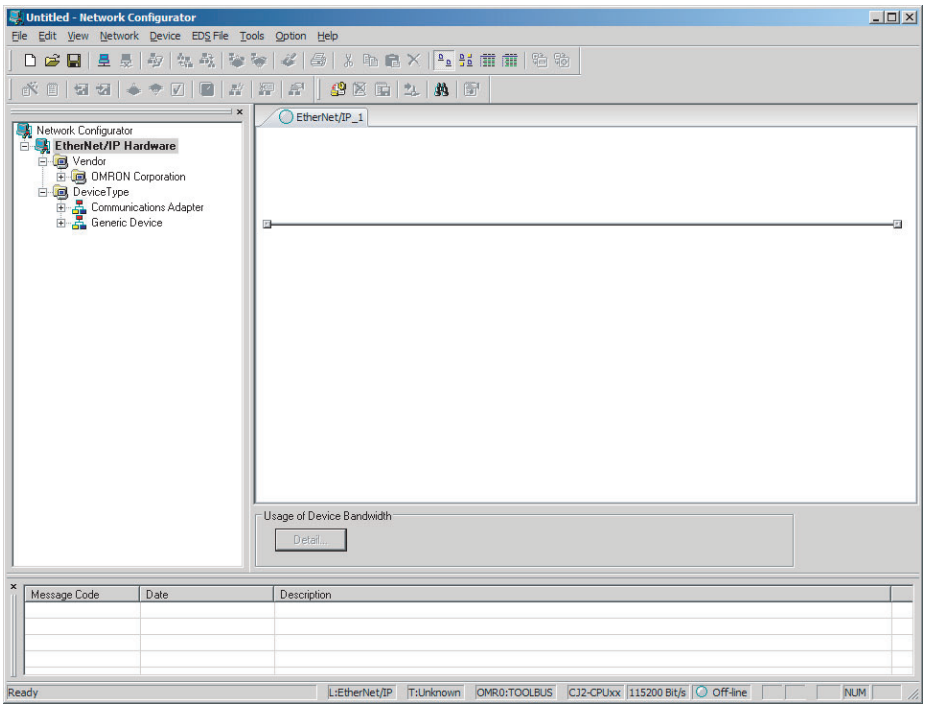
Procedure

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

- **Using the Windows Start Menu**

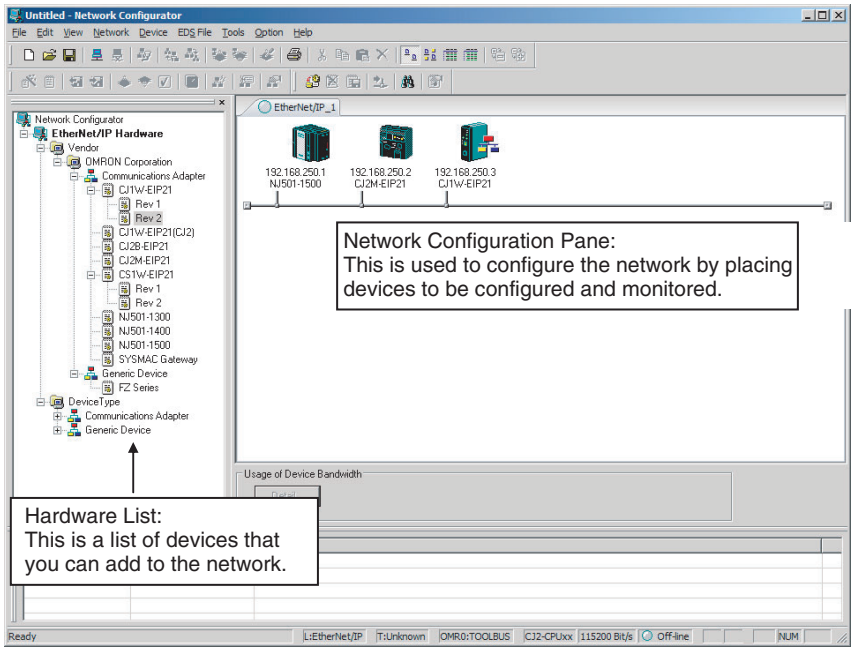
  To start the Network configurator, select ***OMRON*** – ***Sysmac Studio*** – ***Network Configurator for EtherNetIP*** – ***Network Configurator*** from the Windows Start Menu.

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

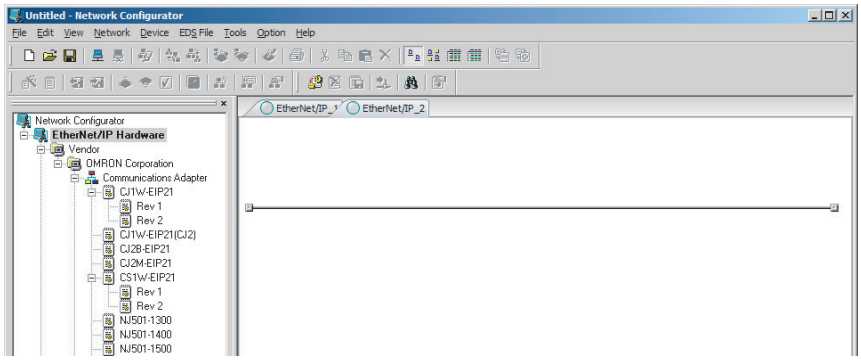


Main Window

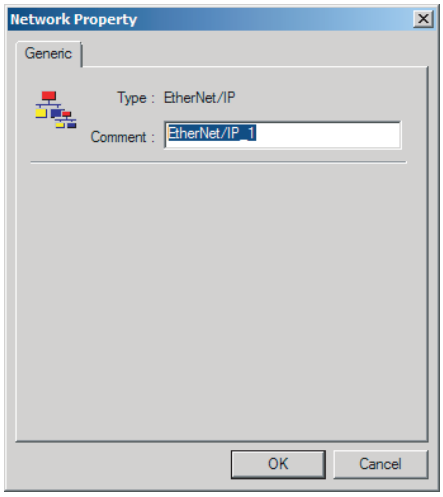
The Main Window consists of a Hardware List and a Network Configuration Pane, as shown in the following diagram.



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 *7-2-8 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 *7-2-8 Uploading Tag Data Link Parameters*.)

↓

3 Download the tag data link parameters. (Refer to *7-2-7 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 *14-1 Overview of Errors*) and the Network Configurator monitor functions. (Refer to *14-3 Checking Status with the Network Configurator*.)

↓

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

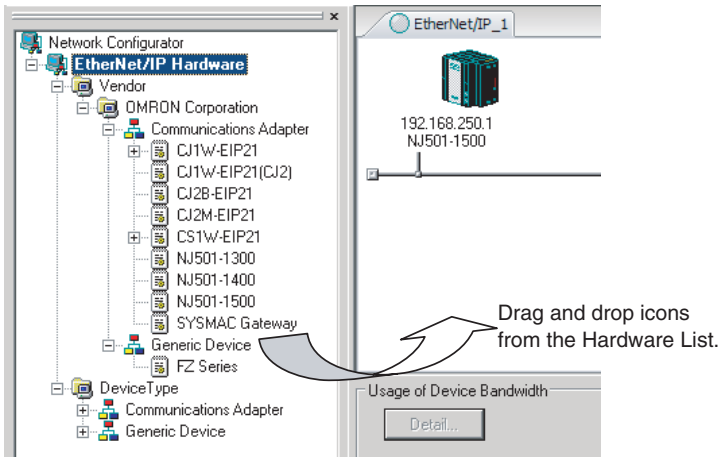
Note Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for the procedure.

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.)

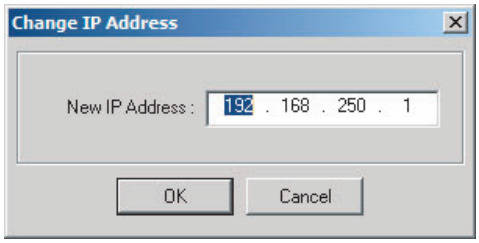
You can also select a device in the Hardware List and press the **Enter** Key to register it. The icon of the device is displayed in the Network Configuration Pane, as shown in the following diagram.



● Hardware List

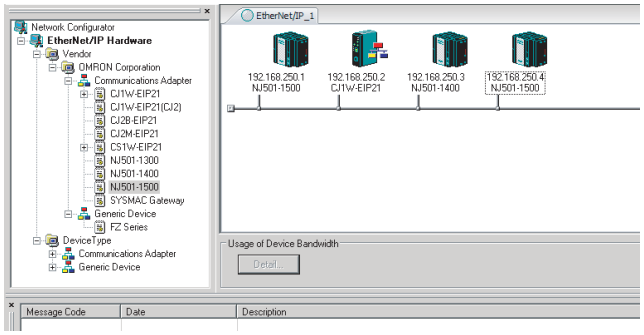
Name in Hardware List	CIP revision	Unit version
NJ501-1□00	Rev. 1.01	Ver. 1.0

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 with the Network Configurator's device parameter editing function.

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

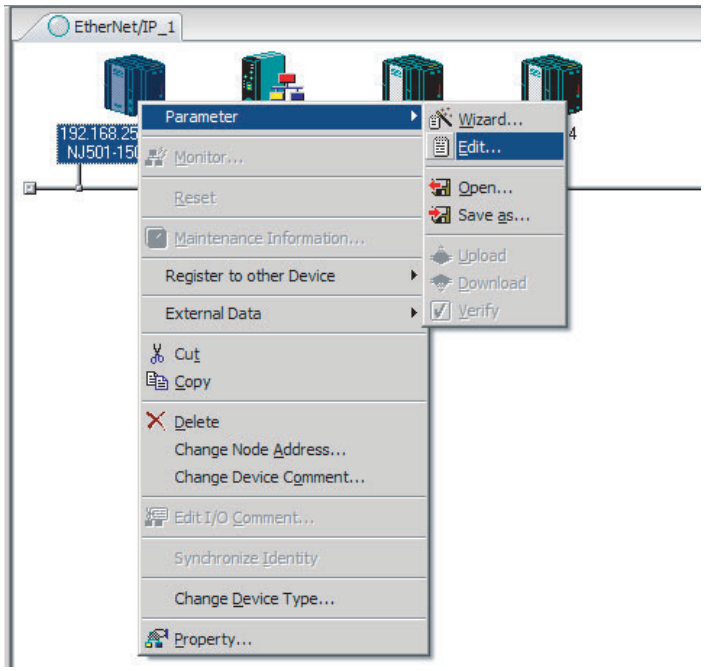
(1) Creating Tags and Tag Sets with the Network Configurator's Device Parameter Editing Function

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

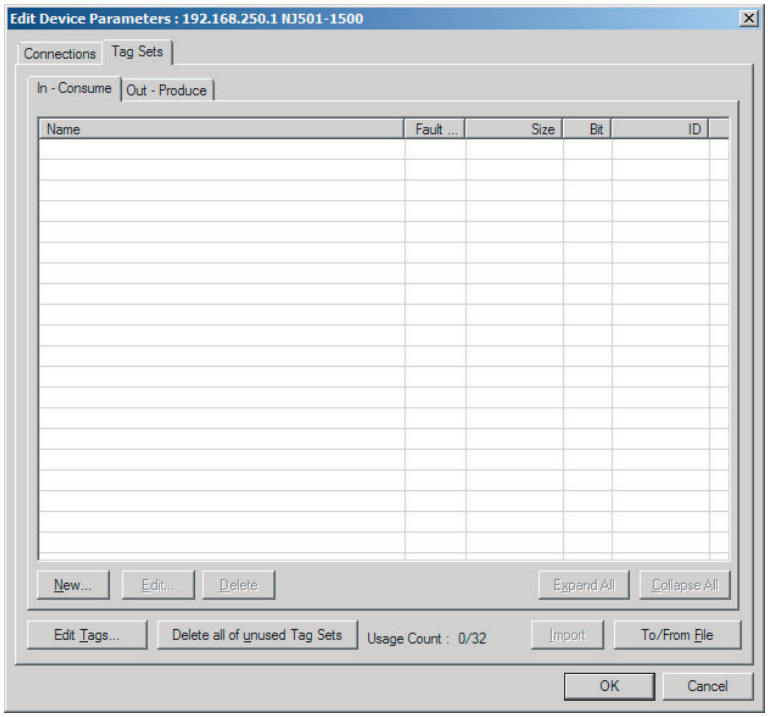
(1) Creating Tags and Tag Sets with the Network Configurator's Device Parameter Editing Function

● Creating a Tag Set

1 Double-click the icon of the device for which to create a tag set to display the Edit Device Parameters Dialog Box. Right-click the icon to display the pop-up menu, and select ***Parameter – Edit***.



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).

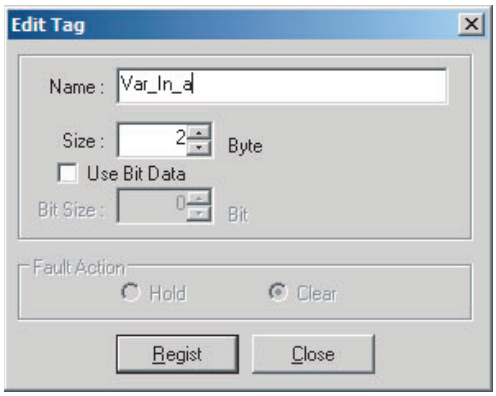


● Creating and Adding Tags

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: Input_signal)

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 256 bytes.
- Specify array and structure 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.)

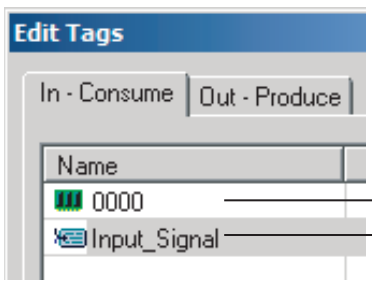
Precautions for Correct Use

If you enter I/O memory addresses for tag names, as shown below, the tags are directly assigned to the I/O memory addresses in the CPU Unit.

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)

Make sure that the tags are assigned correctly. You can check the memory address or variable to which a tag is assigned with the icons in the Edit Tags Dialog Box.



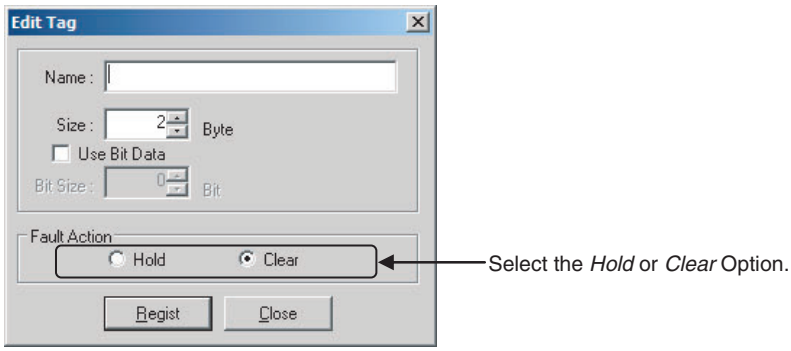
6 Input the size of the tag in bytes in the *Size* Field. Input the tag size so that it is the same as the data type size of the variable. Select the *Use Bit Data* Check Box and change the bit size to 1 to use BOOL variables.

7 Click the **Regist** 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 CPU Unit.

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.

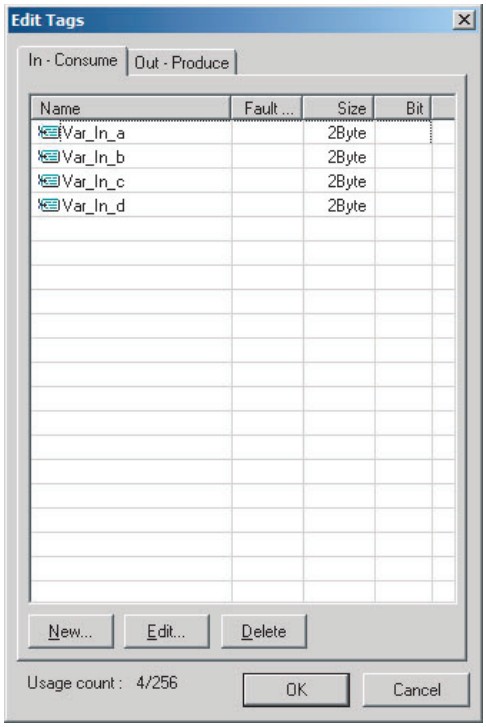


Precautions for Correct Use

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.

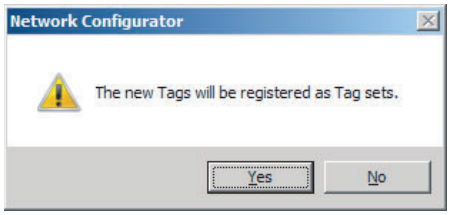


Precautions for Correct Use

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 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.

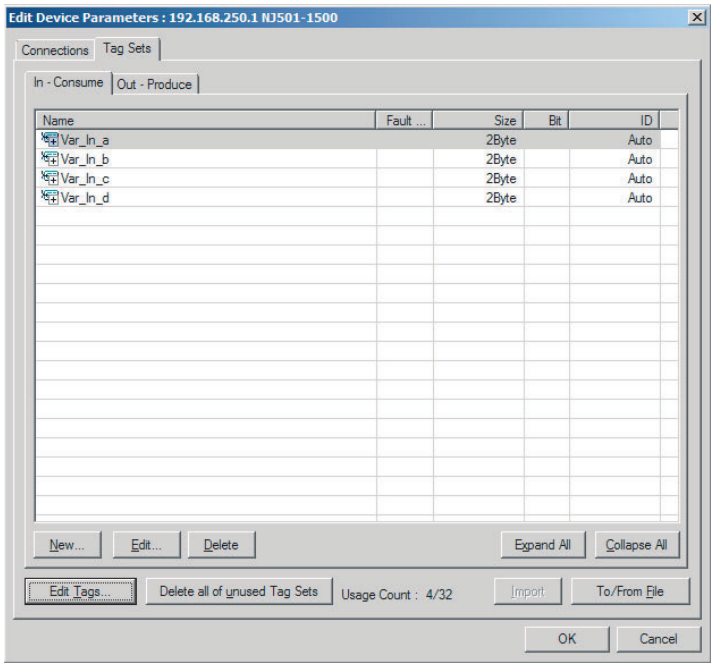
10 At this point, a confirmation dialog box is displayed to check whether the registered tag names are used as the tag set names. A tag set can contain up to eight tags, but tag sets are registered with one tag per tag set if the tag names are registered as tag set names. In this case, click the **Yes**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



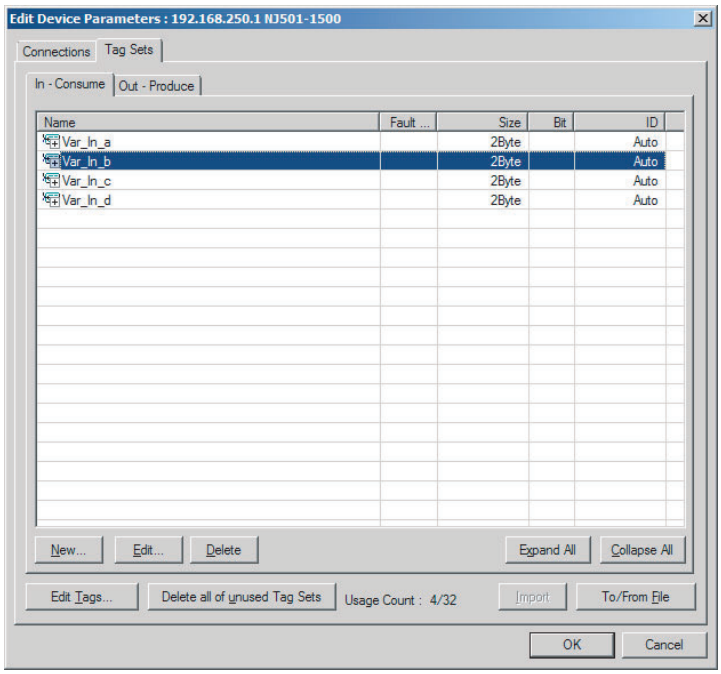
If the **No** Button is clicked, you can add more tags to the tag set. Refer to step 18 for details on how to register new tags first and add more tags to the tag set later.

Changing and Registering Tag Sets

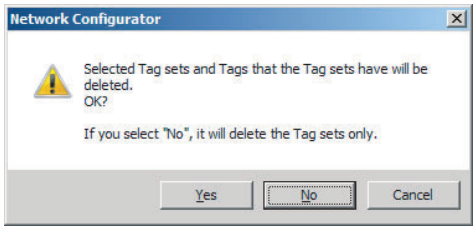
11 The following dialog box is displayed when the tags in the Edit Tags Dialog Box are registered directly as tag sets.



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.)

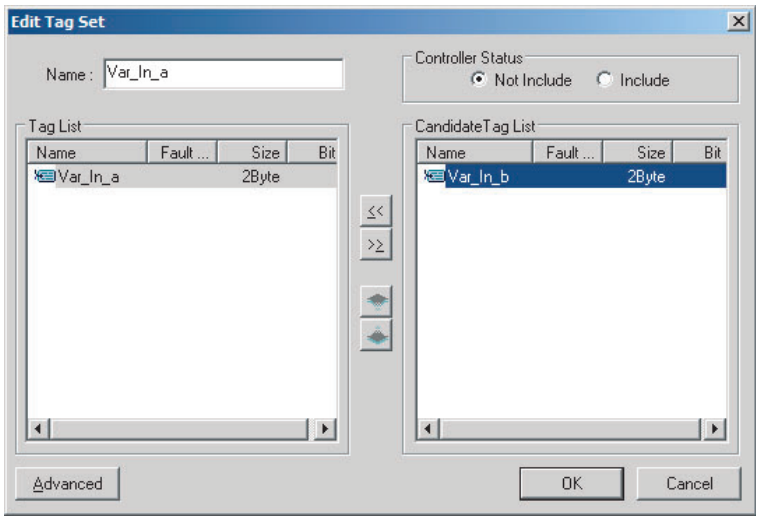


A confirmation message is displayed.



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.



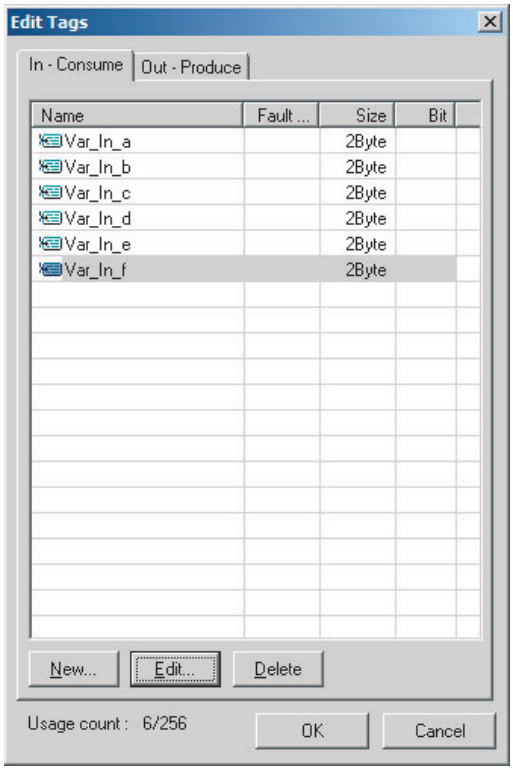
The Tag List on the left side of the dialog box shows the tags that are already registered, and the Candidate Tag List on the right side of the dialog box shows the other tags that are not registered yet. To add a tag, select it in the Candidate Tag List and click the << Button.

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.

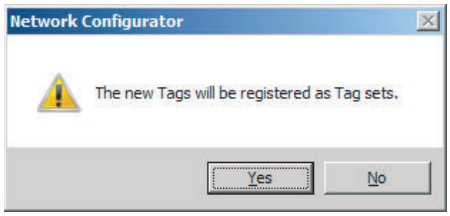
16 Click the **OK** Button in the Edit Device Parameters Dialog Box.

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.

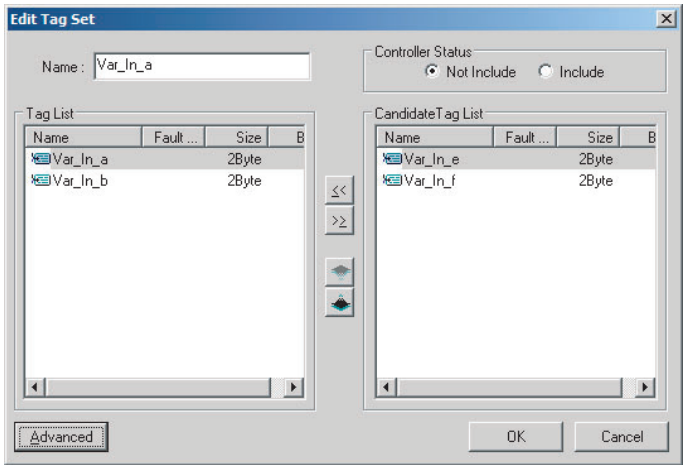


18 After you register all of the required tags, click the **OK** Button at the bottom of the Edit Tags Dialog Box.

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.

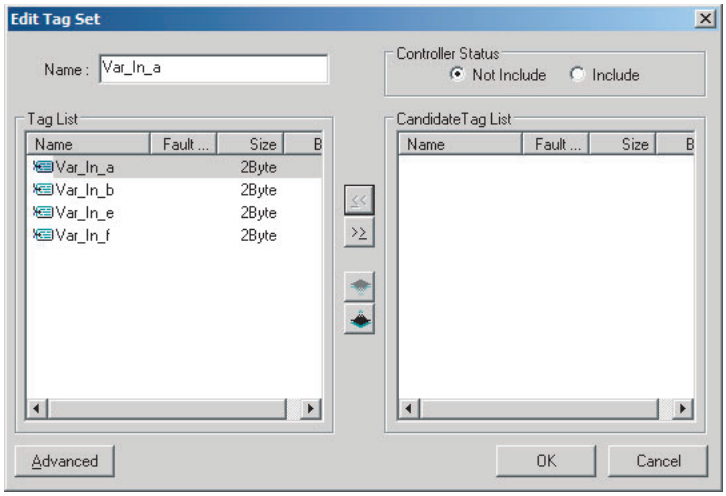


20 To register the newly added tags in a tag set, either double-click the desired tag set, or select the tag set and click the **Edit** Button.



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.

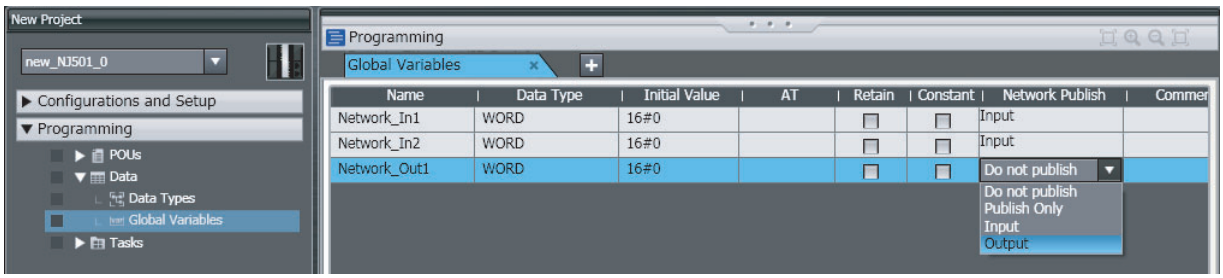
23 Click the **OK** Button in the Edit Device Parameters Dialog Box.

(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)

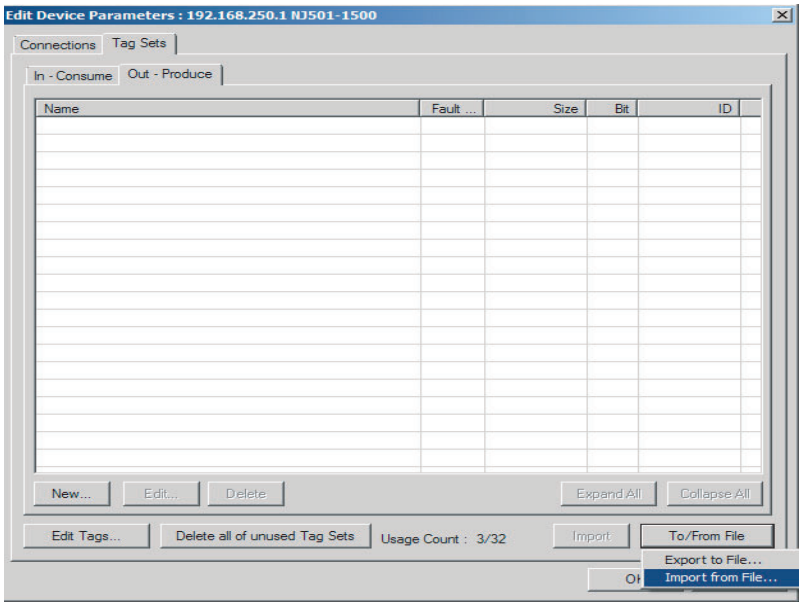
2) Variable names with the following single-byte letters (uppercase or lowercase) followed by single-byte numerals

- 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.

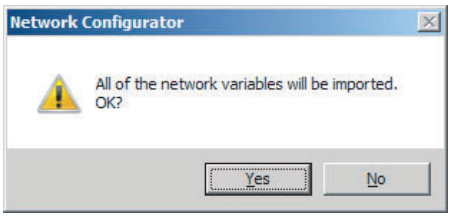
Right-click the icon to display the pop-up menu, and select ***Device*** – ***Parameter*** – ***Edit***.

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.

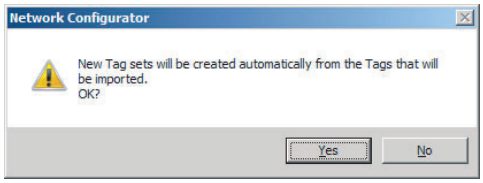


A confirmation dialog box is displayed that asks you how you want to import the variables as shown below.

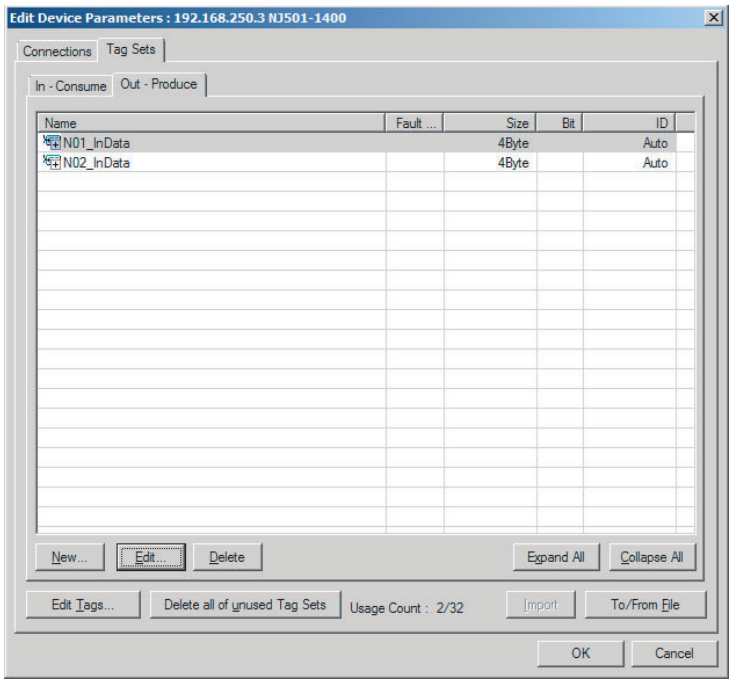
- To import all variables with a Network Publish attribute, click the **Yes** Button. To import only some of these variables, click the **No** Button.



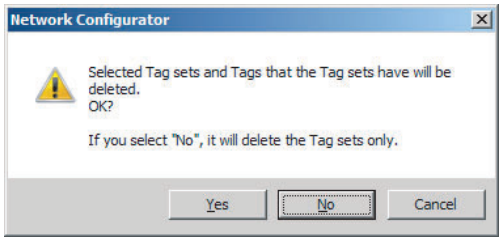
After you import the variables to the tags, click the **Yes** Button to automatically create tag sets, or click the **No** Button to set up tag sets manually.



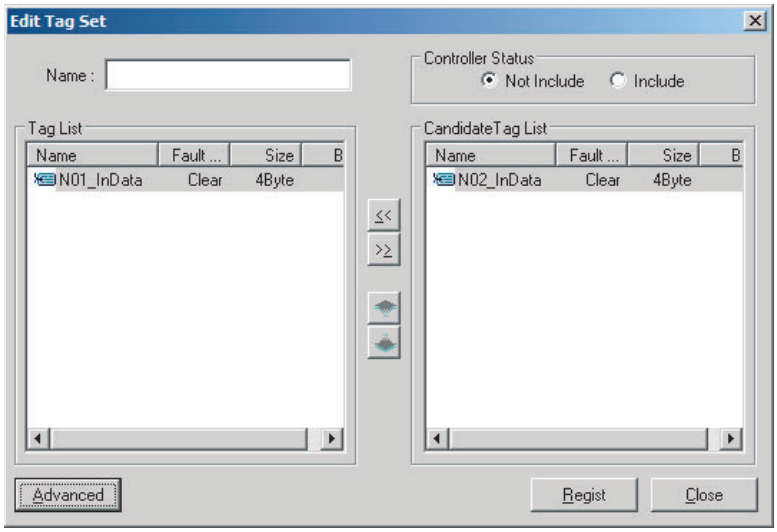
The variables will be imported as shown below on the Tag Sets Tab Page. Each variable will be imported into a different tag set and the device parameters will be automatically edited. (The variable name will be used for the tag set name.)



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 **Regist** 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
- Whether the tag sets are input or output tag sets
- The length of the packet intervals (RPI)

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

Precautions for Correct Use

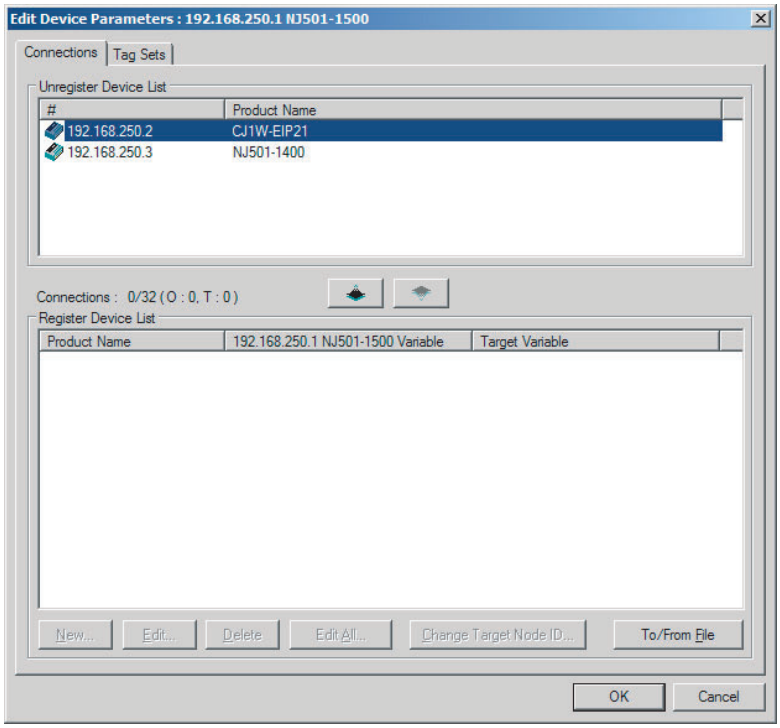
Make the connections settings after you create tag sets for all of the devices involved in tag data links.

Connection Settings (Connections Tab Page)

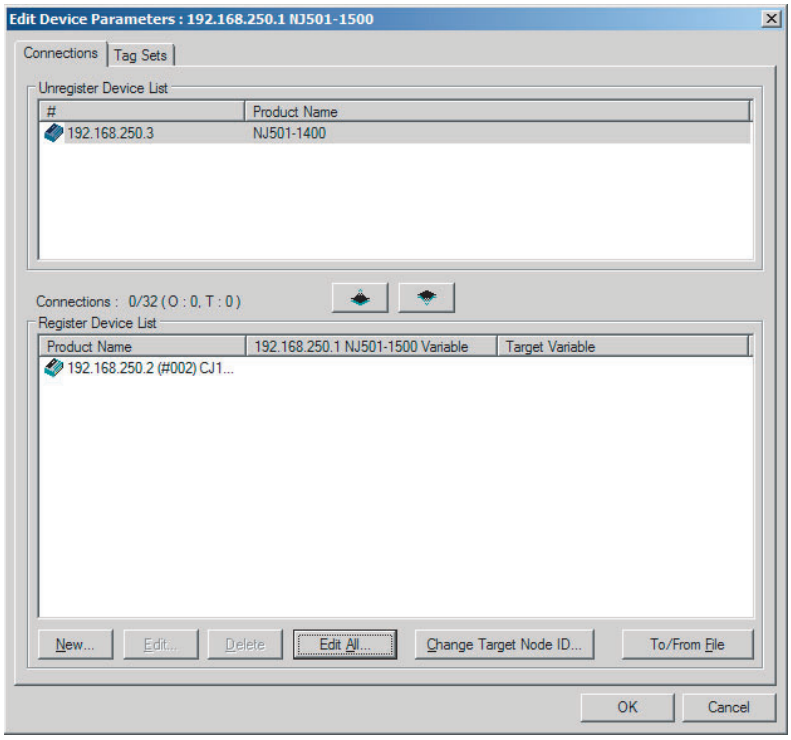
● Registering Devices in the Register Device List

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***.

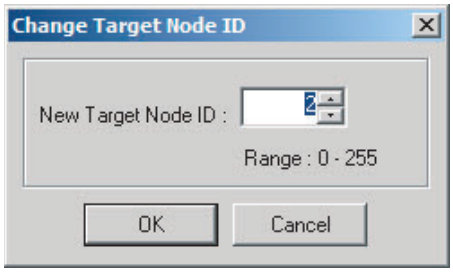
2 Click the **Connections** Tab in the Edit Device Parameters Dialog Box. All of the devices registered in the network (except the local node) are displayed.



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 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.



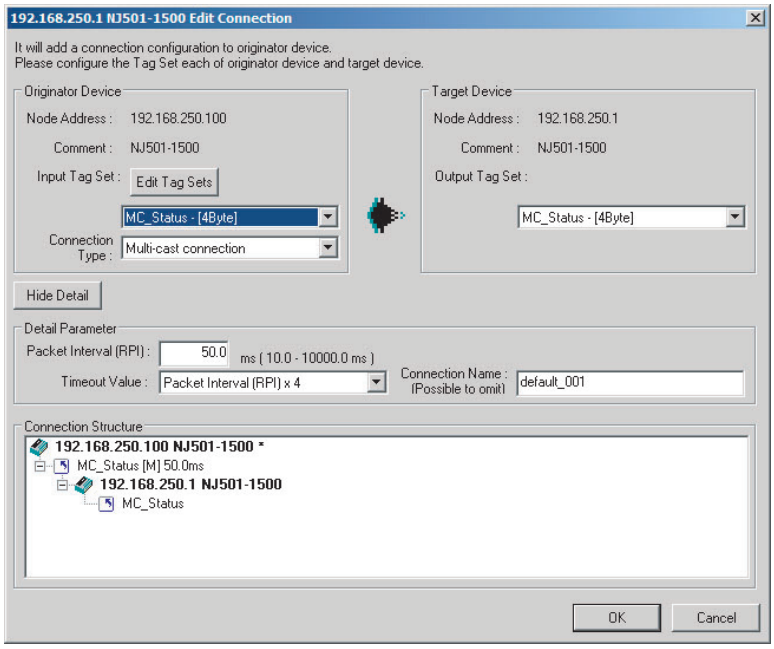
● Editing Settings for Individual Connections

You can edit each connection separately.

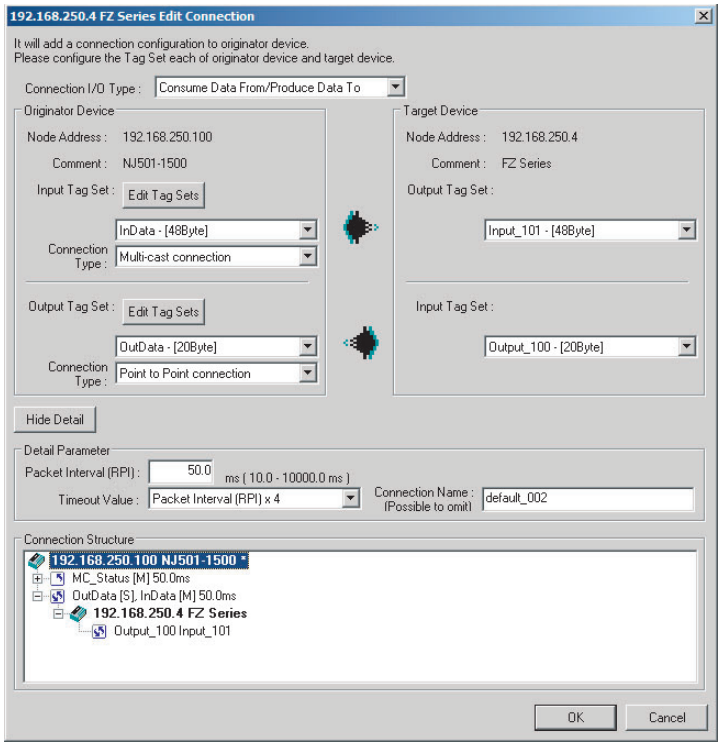
Note Refer to the following section for information on how to perform batch editing in a table format.

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:

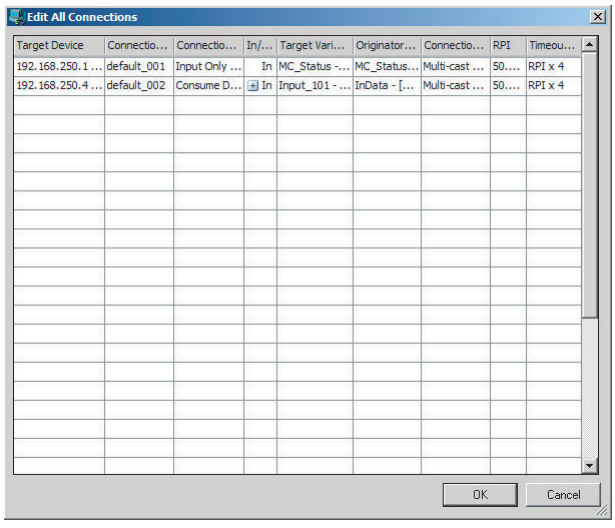
Setting	Description
Connection I/O Type	Select Input Only (tag type) to use tag data links with a CS1W-EIP21, CJ1W-EIP21, CJ2B-EIP21, CJ2M-EIP21, CJ1W-EIP21(CJ2), or NJ501-1□00. When you create tag data links for other devices, select the connection I/O type specified in that device's EDS file. Use the 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. • 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. **Note** Refer to *7-1-4 Overview of Operation* for details on using multi-cast and unicast connections, and counting the number of connections.
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. You can set an interval to between 10 and 10,000 ms in 1-ms increments (NJ-series Controllers: between 10 and 10,000 ms in 1-ms increments). The default setting is 50 ms (i.e., data is updated once every 50 ms).
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.



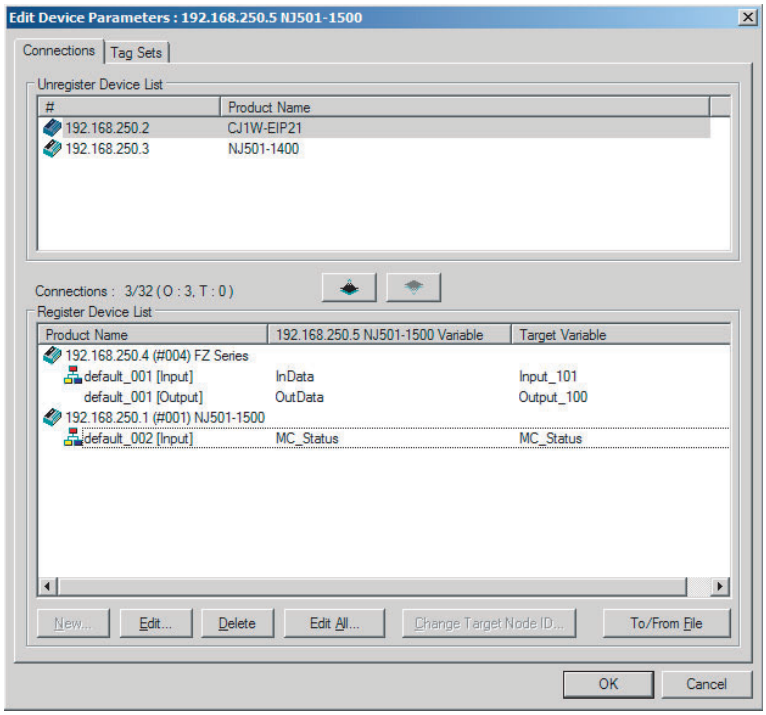
The settings are as follows:

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 CS1W-EIP21, CJ1W-EIP21, CJ2B-EIP21, CJ2M-EIP21, CJ1W-EIP21(CJ2), or NJ501-1□00. 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. • Input Only: Just *In* is displayed.
Target Variable	Select the target node's tag set to assign it. • 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. • 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. • 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. **Note** Refer to *7-1-4 Overview of Operation* for details on using multi-cast and unicast connections, and counting the number of connections.
RPI	Set the packet interval (RPI) of each connection between the originator and target. You can set an RPI between 10 and 10,000 ms in 1-ms increments. The default setting is 50 ms (i.e., data is updated once every 50 ms).
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).

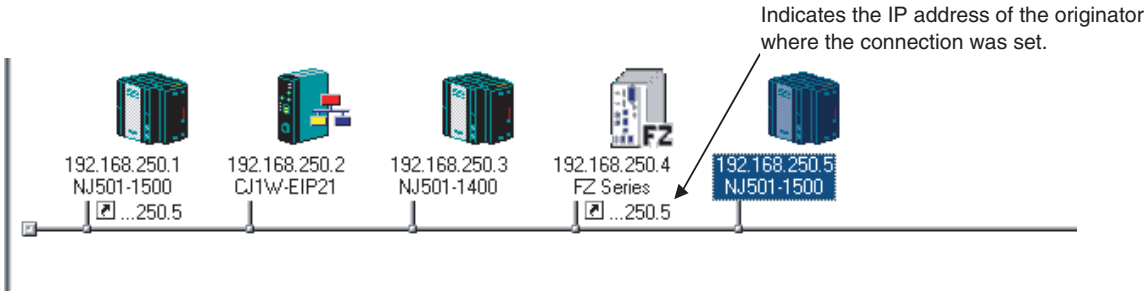
2 After you make all of the settings, click the **OK** Button.

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 *7-2-14 Checking Connections*.)

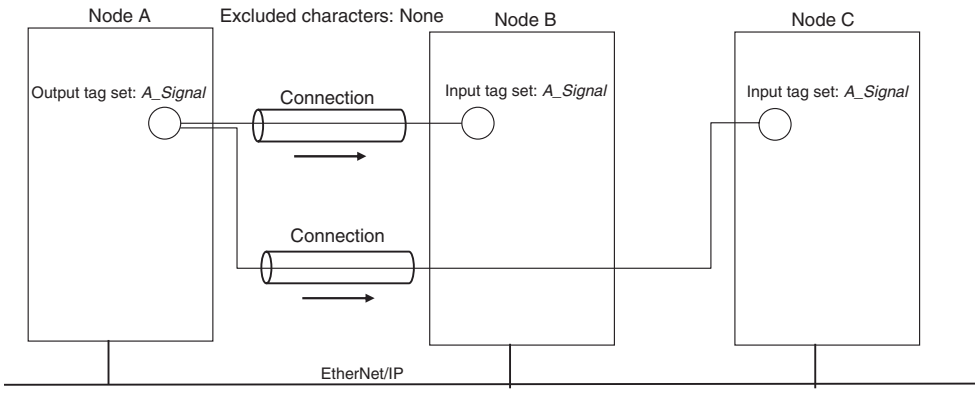
Automatically Setting Connections (*Network* – *Auto Connection*)

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.

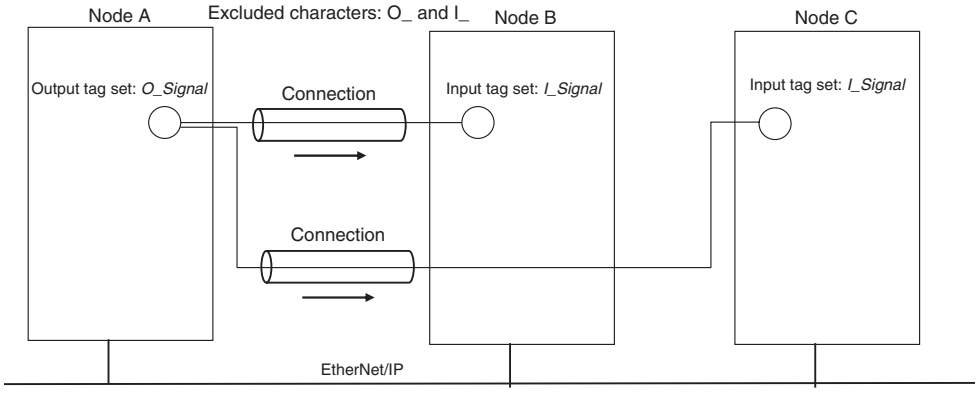
Example 1: Automatic Connections with the Same Tag Set Names

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.



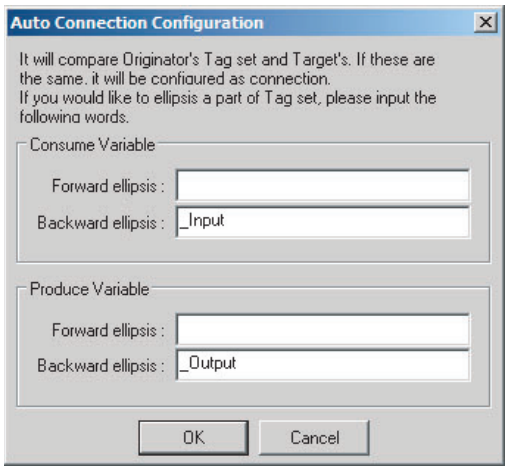
Example 2: Automatic Connections with the Ellipses

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.



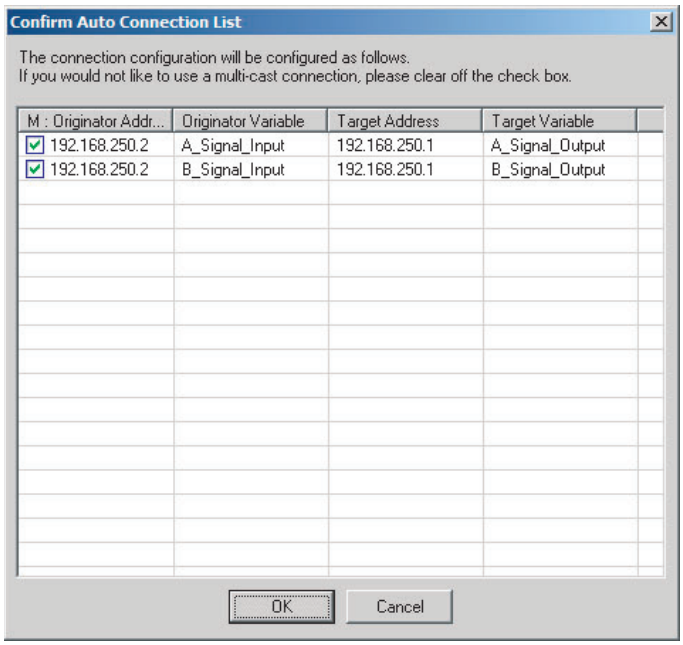
1 Set the same tag set names for the output and input tag sets for the connection. The tag set names can also include forward and backward ellipses.

2 Select ***Auto Connection Configuration*** from the Network Menu. The connections will be set automatically. A dialog box will appear to set forward and backward ellipses for both output and input tag sets as soon as automatic connection setting processing has begun.



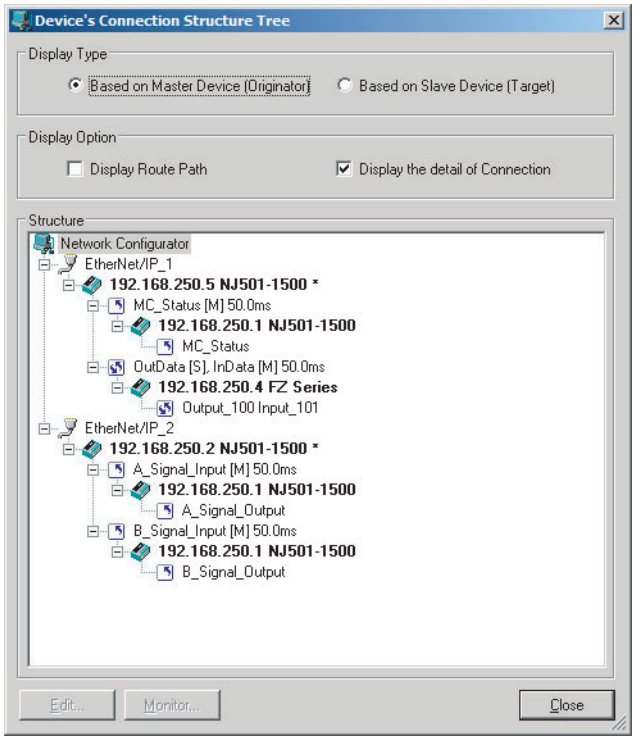
Input the ellipses and click the **OK** Button. Processing for automatic setting is started.

3 If there are tag sets that meet the conditions for automatic connection setting, they are displayed.



Click the **OK** Button. Processing for automatic setting is started.

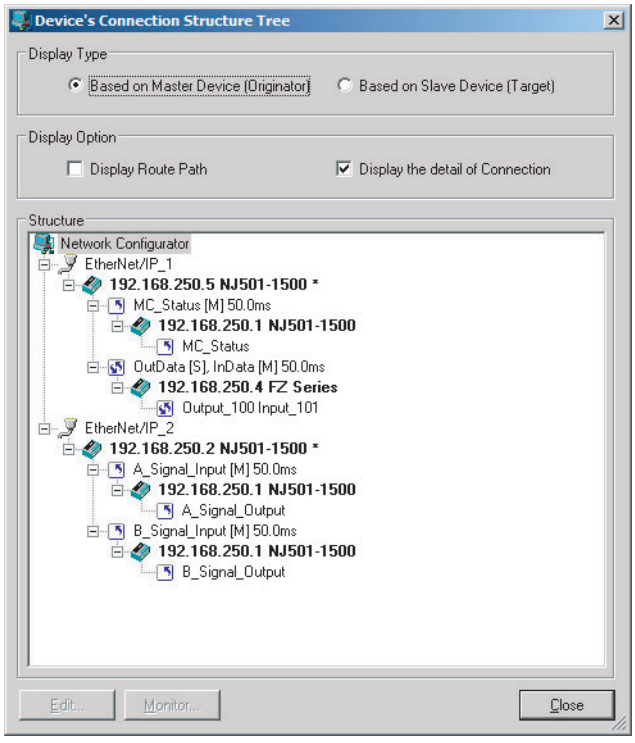
4 A device connection structure tree is displayed when processing is completed.



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 Connecting the Network Configurator to the Network

This section describes how to Connect the Network Configurator online.

Connecting through Ethernet

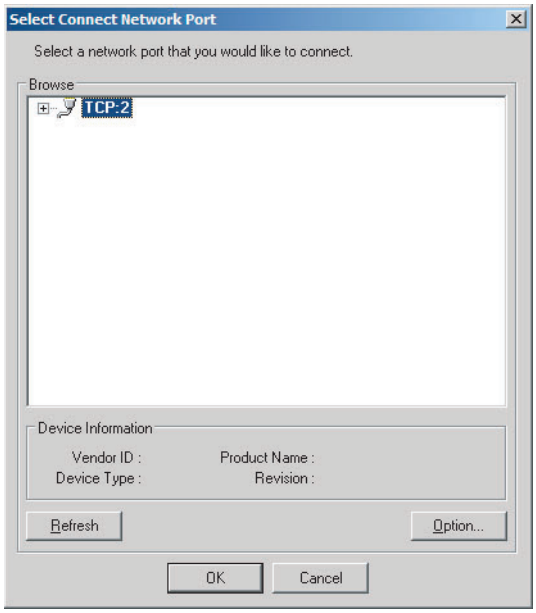
Precautions for Correct Use

The first time you connect via Ethernet with Windows XP (SP2 or higher), Windows Vista, or Windows 7, you much change the Windows firewall settings. For the procedure, refer to *A-3 Precautions for Using the Network Configurator on Windows XP, Windows Vista, or Windows 7*.

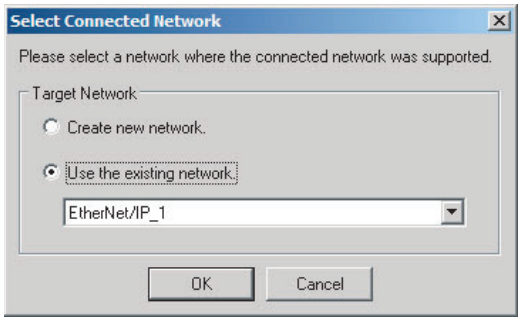
Connect to the EtherNet/IP Unit's Ethernet port via the Ethernet network.

1 Select ***Option*** – ***Select Interface*** – ***Ethernet I/F***.

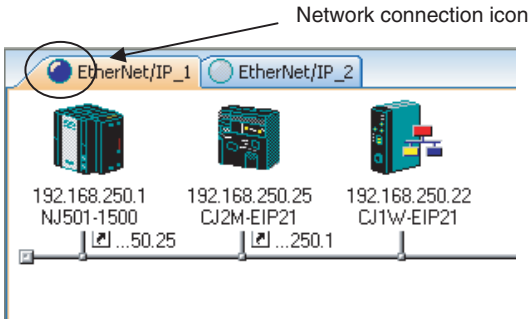
2 Select ***Network*** – ***Connect***. If there are multiple Ethernet interfaces on the computer, the Select Connect Network Port Dialog Box is displayed. Select the interface to connect, and press the **OK** Button.
The following dialog box is displayed.



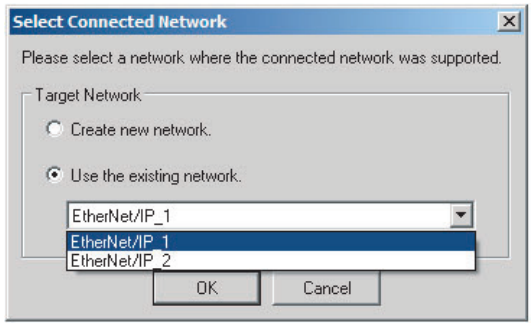
3 Click the **OK** Button.
Select the network to connect.



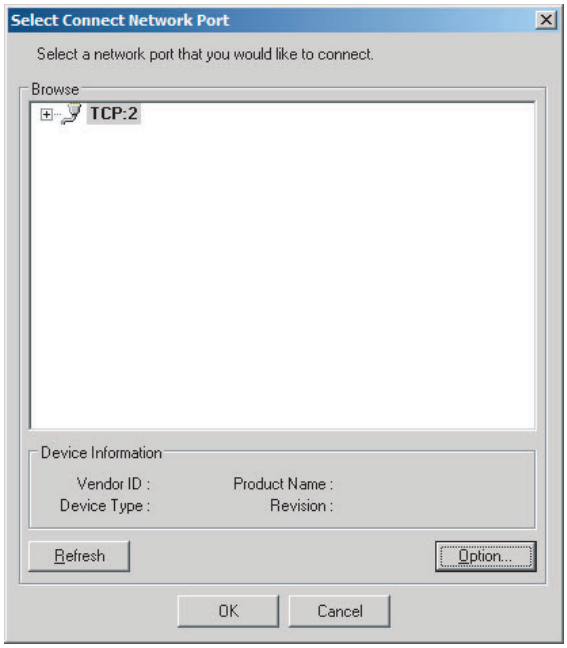
The Network Configurator will connect to the EtherNet/IP network. If the Network Configurator goes online normally, “Online” is displayed in the status bar at the bottom of the window. The network connection icon is displayed in blue on the Network Tab Page in which the Network Configurator is connected.



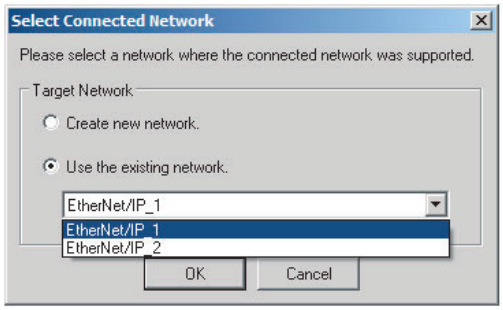
Select ***Network*** – ***Change Connect Network*** to switch the connected network.



1 The following dialog box is displayed.



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



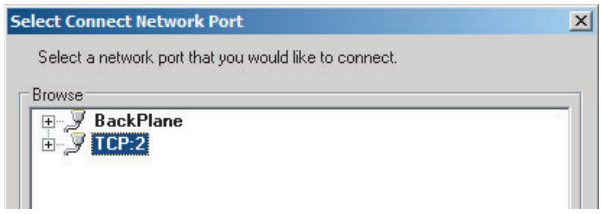
Connections through NJ-series CPU Unit's USB Port

Use the following procedure to connect to the built-in EtherNet/IP port via the USB port on the CPU Unit.

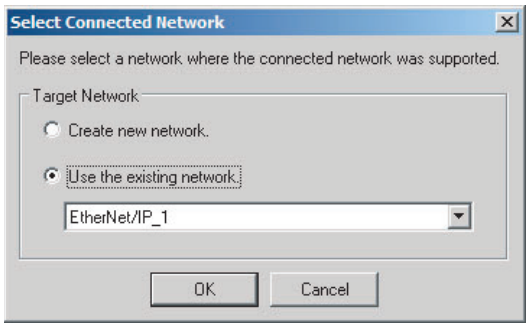
1 Select the communications interface.
Select ***Option*** – ***Select Interface*** – ***NJ Series USB Port***.

2 Select ***Network*** – ***Connect***. The following dialog box is displayed.

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

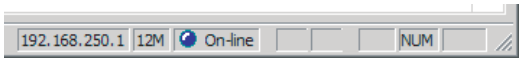


4 Select the network to connect.



5 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.

6 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.



Connections via Ethernet to NJ-series CPU Unit

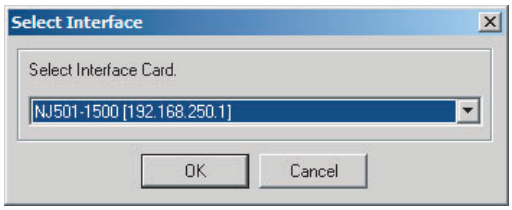
Use the following procedure to directly connect to the built-in EtherNet/IP port via Ethernet. You can connect to the built-in EtherNet/IP port even if the IP address has not been set on the computer.

1 Select the communications interface.

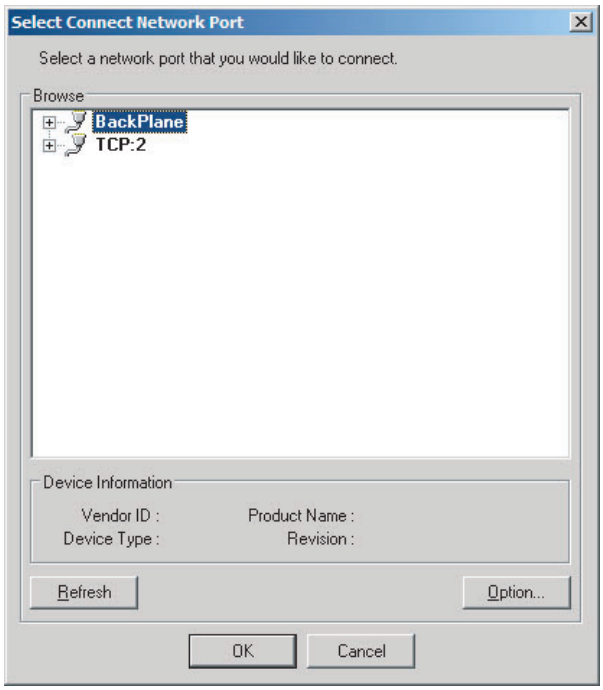
Select ***Option*** – ***Select Interface*** – ***NJ Series Ethernet Direct I/F***.

2 Select ***Network*** – ***Connect***.

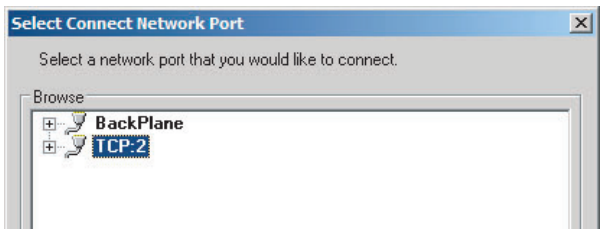
The Setup Interface Dialog Box is displayed if there are several Controllers that you can connect to. Select NJ501-1□00 (IP number)



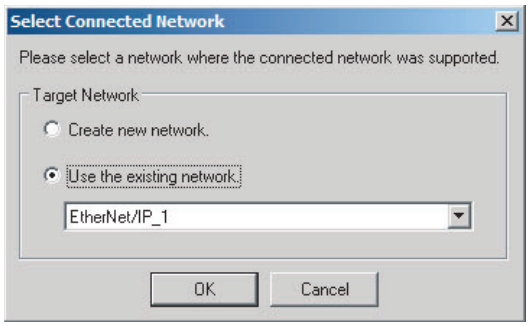
3 Select ***Network*** – ***Connect***. The following dialog box is displayed.



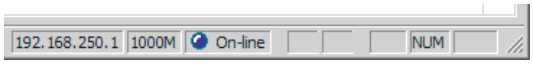
4 Select *TCP:2* and then click the **OK** Button.



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.



7-2-7 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 *7-2-6 Connecting the Network Configurator to the Network* for information on how to Connect the Network Configurator online.

Precautions for Correct Use

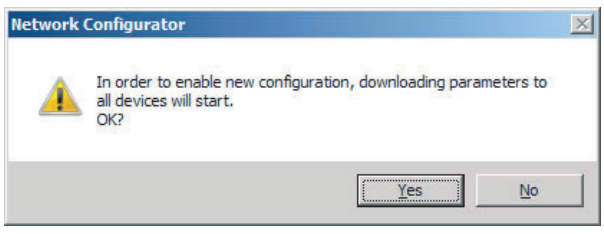
- If the node addresses (IP addresses) are not set correctly, you may connect to the wrong Controller and set incorrect device parameters. Download data only after you confirm that you are connected to the correct Controller.
- If incorrect tag data link parameters are set, it may cause equipment to 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 CPU Unit. Before downloading the tag data link parameters, check to confirm that the network variables are set in the CPU Unit. Check whether the network variable, tag, and connection settings are correct. On the Connection and Tag Status Tab Pages described in *14-3-1 The Network Configurator's Device Monitor Function*.
- 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 or reset or turn OFF the power to the EtherNet/IP Unit 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.
- For EtherNet/IP Units with revision 1, you can download tag data link parameters only when the CPU Unit is in PROGRAM mode.
- Even for Units with revision 2 or later, all CPU Units must be in PROGRAM mode to download the parameters if any Units with revision 1 are included in the network.

1 Connect the Network Configurator online.

2 There are two ways to download the parameters.

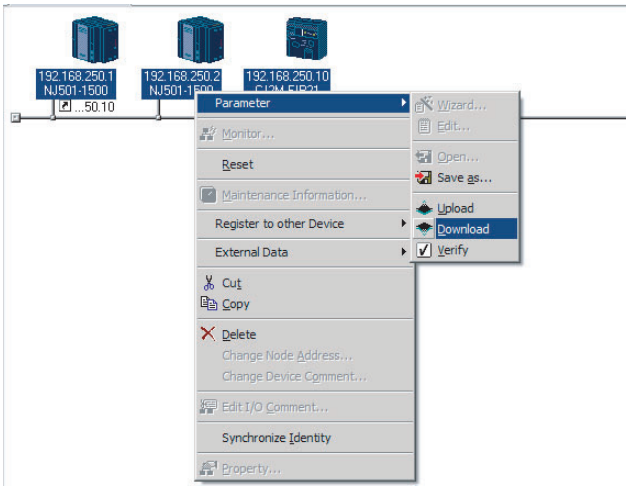
Downloading to All Devices in the Network

Select ***Network*** – ***Download***.
The following dialog box is displayed.

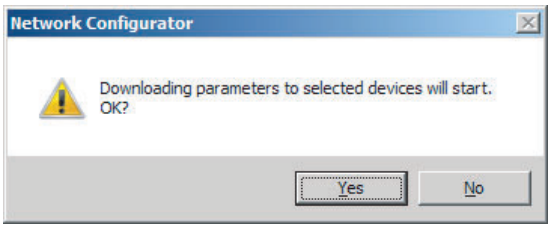


Downloading Individually to Particular Devices

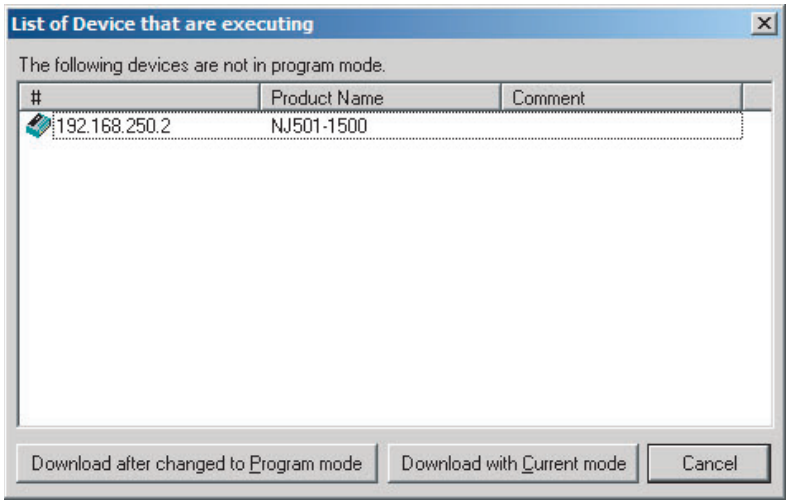
Select the icon of the EtherNet/IP Unit to which you want to download. To select multiple nodes, hold down the **Shift** Key while you click the icons. (In the following example, 2 nodes are selected: 192.168.250.1 and 192.168.250.2.) Right-click the icon to display the pop-up menu, and select ***Parameter***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



The following dialog box is displayed.



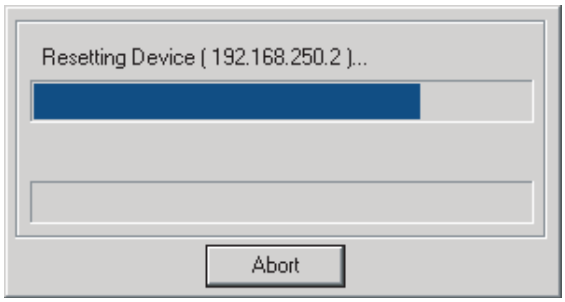
3 Click the **Yes** Button to download the tag data link parameters to the EtherNet/ IP Unit. The following dialog box is displayed if any of the CPU Units is not in PROGRAM mode.



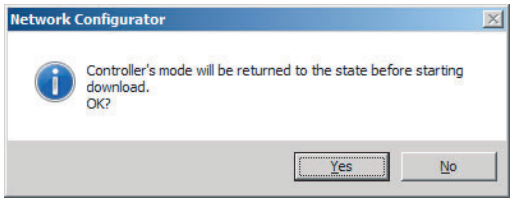
If the **Download after changed to Program mode** Button is clicked, all CPU Units are changed to PROGRAM mode and the parameters are downloaded. Confirm safety for all controlled equipment before you change the CPU Units to PROGRAM mode. You can restore the operating modes after the parameters are downloaded.

You can click the **Download with Current mode** Button to download the parameters even when one or more CPU Units 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).

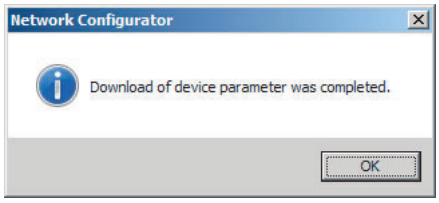
During the download, the following progress monitor is displayed to show the progress of the download.



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



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



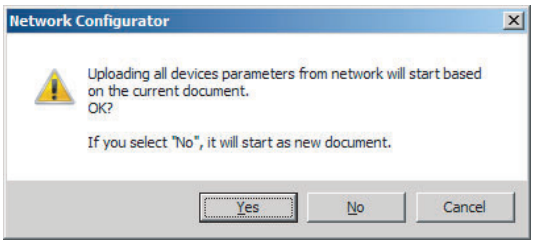
7-2-8 Uploading Tag Data Link Parameters

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 *7-2-6 Connecting the Network Configurator to the Network*.

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.

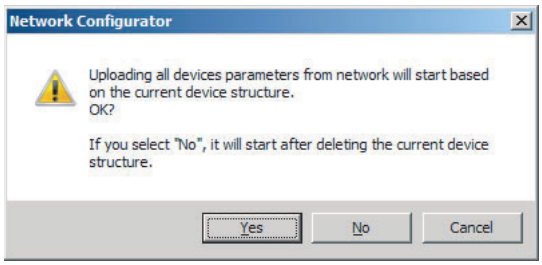
Clicking the No Button:

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.

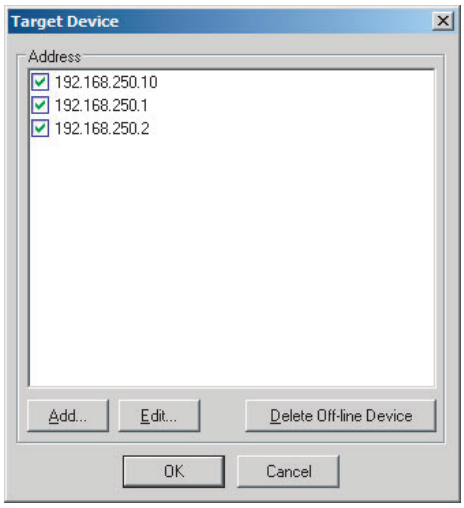


Clicking the Yes Button:

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:

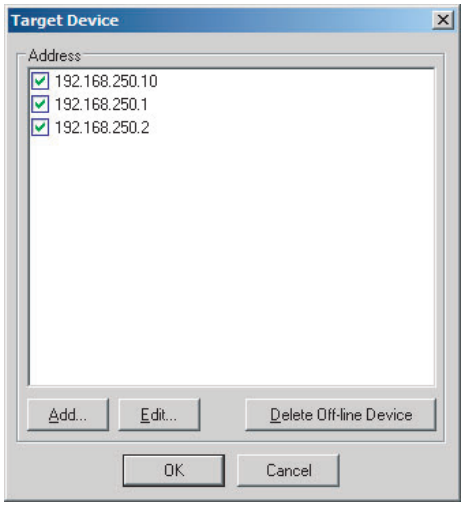
Performing a Batch Upload over the Network
Parameters are uploaded from all devices on the network. The current Network Configuration Information will be lost.
The following dialog box will be displayed. Select the devices for which to upload parameters and click the **OK** Button.



Clicking the Cancel Button:

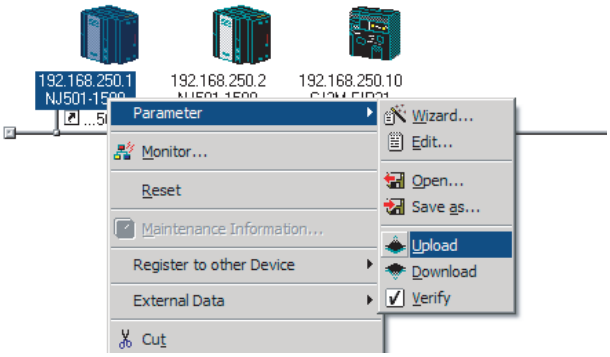
The upload operation is cancelled. The upload is not performed.

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.

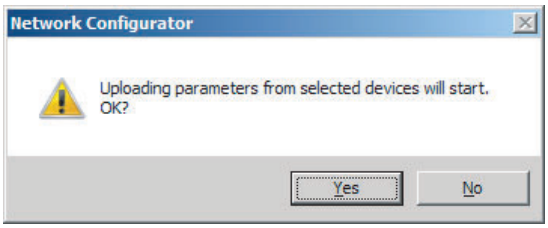


Uploading Individually from Particular Devices

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 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***.

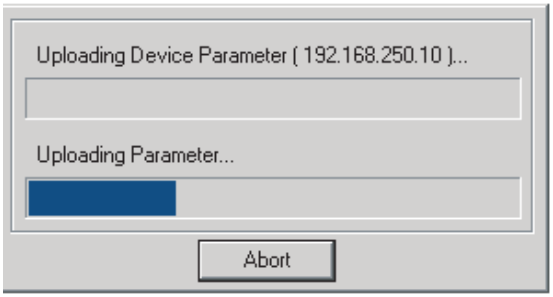


2 The following dialog box is displayed.

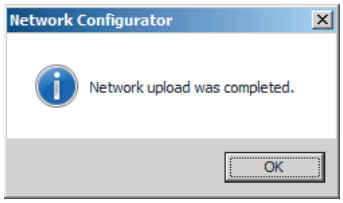


Click the **Yes** Button or the **No** Button.

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.



7-2-9 Verifying the Tag Data Links

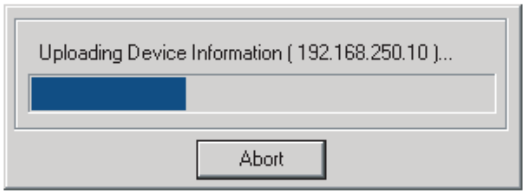
Tag data link parameters (such as the tag set settings and connection settings) can be compared with the parameters of the built-in EtherNet/IP ports in the EtherNet/IP network. The following procedure shows how to compare the parameters. For details on how to connect to the network from the Network Configurator, refer to *7-2-6 Connecting the Network Configurator to the Network*.

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.

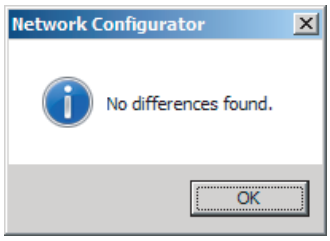
1 Connect the Network Configurator online.

2 Select ***Network*** – ***Verify Structure***.
The following progress monitor is displayed to show the progress as data is read from the network and compared.

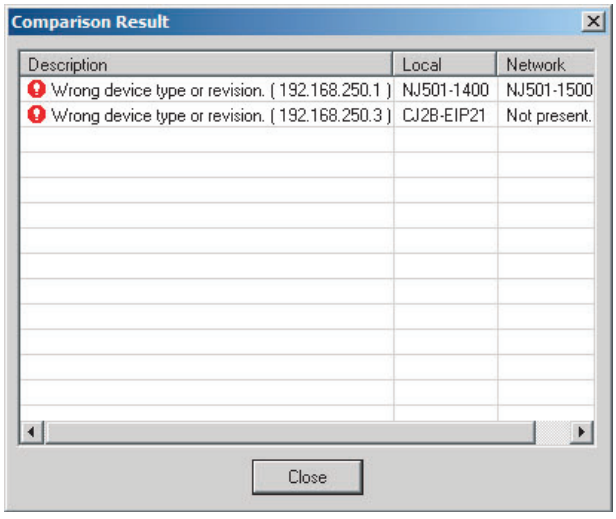


3 The results of the comparison between the network configuration file and data from the network are displayed as shown below.

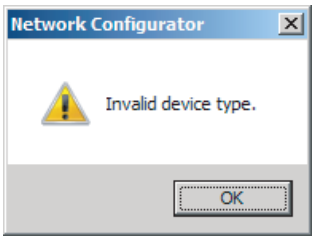
Differences Not Found in the Comparison



Differences Found in the Comparison



Differences Found in the Device Type



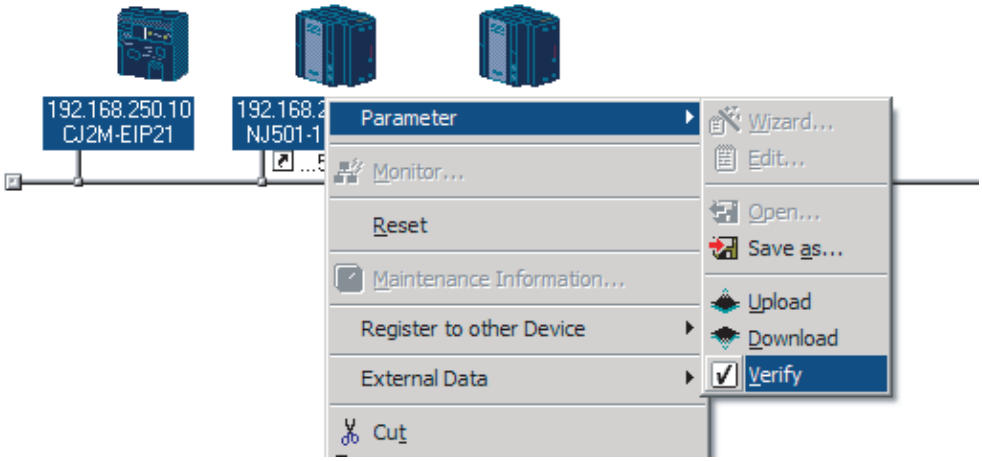
Click the **OK** Button or the **Close** Button.

Verifying the Device Parameters

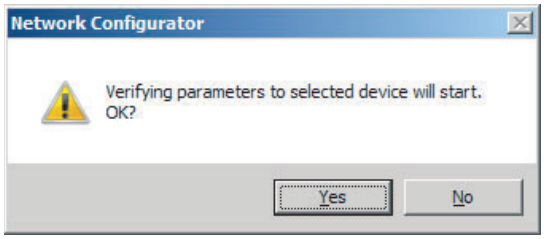
Use the following procedure to compare the device parameters for the devices selected in the Network Configuration Pane with those of the devices connected on the EtherNet/IP network. The IP addresses, device types, and device parameters are compared.

1 Connect the Network Configurator online.

2 Click the icon of the built-in EtherNet/IP port to verify. To select multiple nodes, press and hold the **Shift** 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*** – ***Verify***.



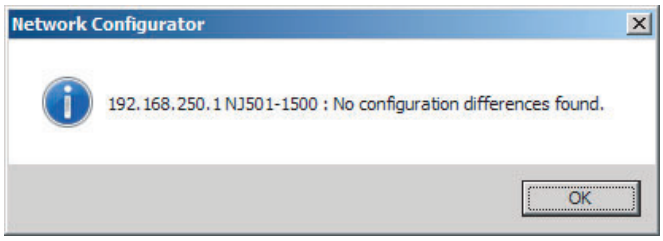
3 The following dialog box is displayed.



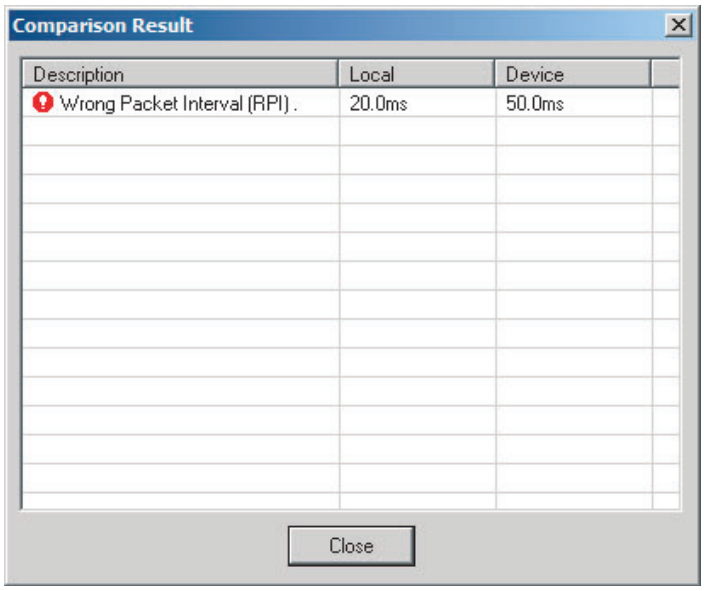
Click the **Yes** Button or the **No** Button.

4 The following dialog box is displayed.

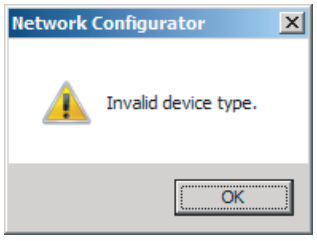
Differences Not Found in the Comparison



Differences Found in the Comparison

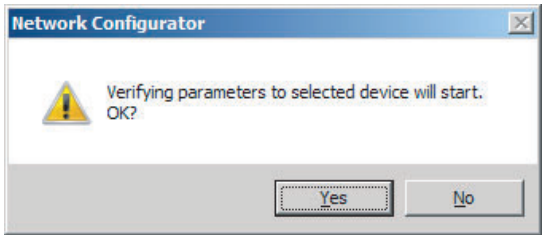


Differences Found in the Device Type



Click the **OK** Button or the **Close** Button.

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.

7-2-10 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.)

Starting and Stopping Tag Data Links for Individual Devices

You can start and stop tag data links for individual devices 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 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 Start Switch)
- _EIP_TDLinkStopCmd (Tag Data Link Stop Switch)

Additional Information

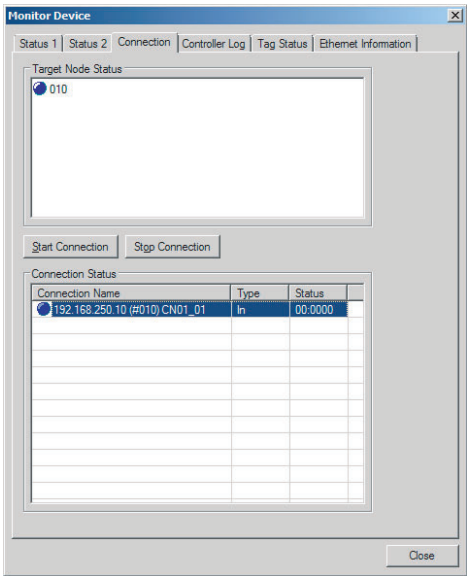
Change the Tag Data Link Start Switch to TRUE, while the Tag Data Link Communications Stop Switch is FALSE. If the Tag Data Link Stop Switch is TRUE, the tag data links do not start even if the Tag Data Link Start Switch is changed to TRUE. Furthermore, if the Tag Data Link Start Switch and the Tag Data 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.

● 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.

● 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.



Start Connection Button:
Starts all connections for which the device is the originator.

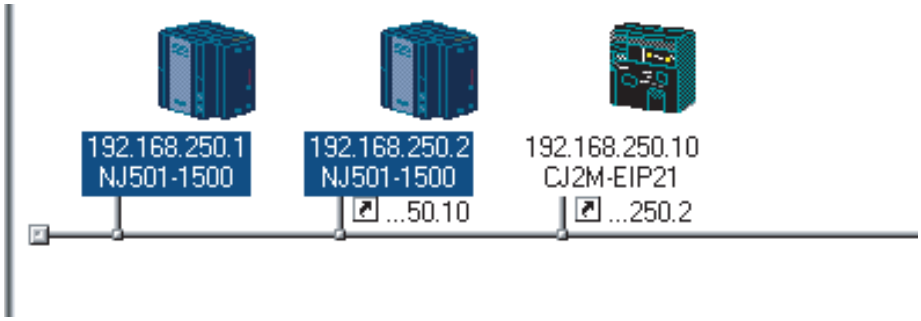
Stop Connection Button:
Stops all connections for which the device is the originator.

7-2-11 Clearing the Device Parameters

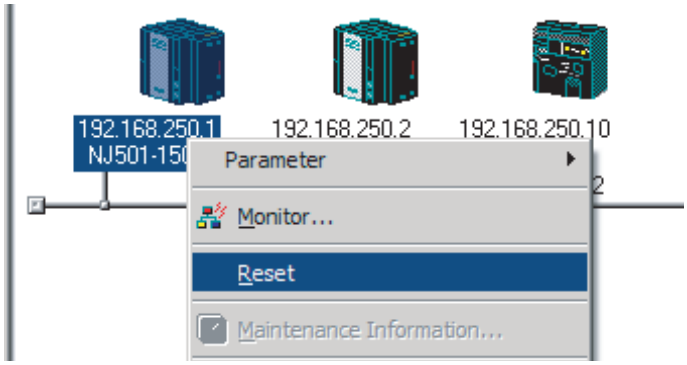
You can clear the tag data link parameters that are saved in the built-in EtherNet/IP port on the EtherNet/IP network to return them to their default settings. The following procedure shows how to clear the tag data link parameters. For details on how to connect to the network from the Network Configurator, refer to *7-2-6 Connecting the Network Configurator to the Network*.

1 Connect the Network Configurator online.

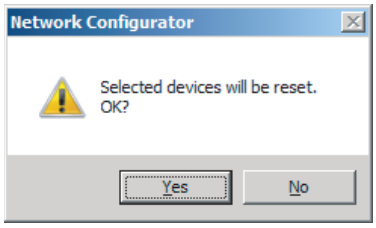
2 Select the icon of the EtherNet/IP Unit from which you want to clear the device parameters. In the following example, 2 nodes are selected: 192.168.250.1 and 192.168.250.2. To select multiple nodes, press and hold the **Shift** Key while you select additional icons.



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

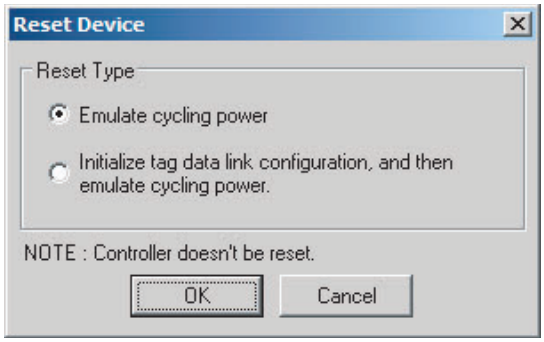


4 The following dialog box is displayed.



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. Only the built-in EtherNet/IP port is restarted.

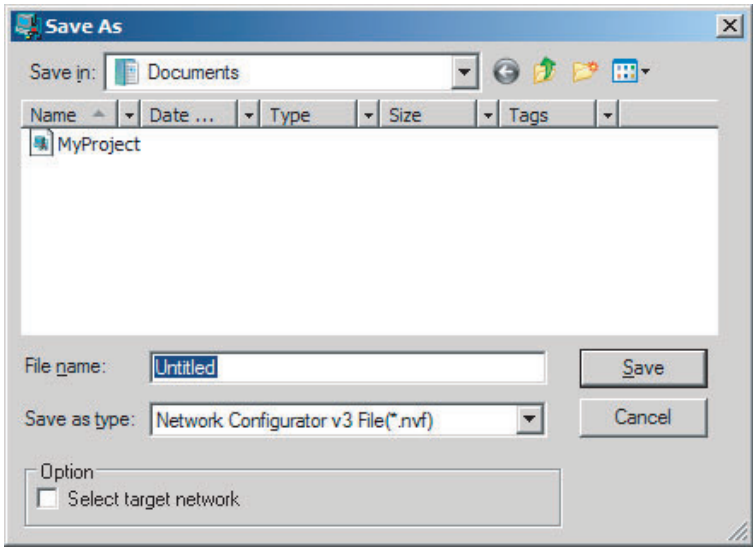
No Button:

The device parameters are not cleared or reset.

7-2-12 Saving the Network Configuration File

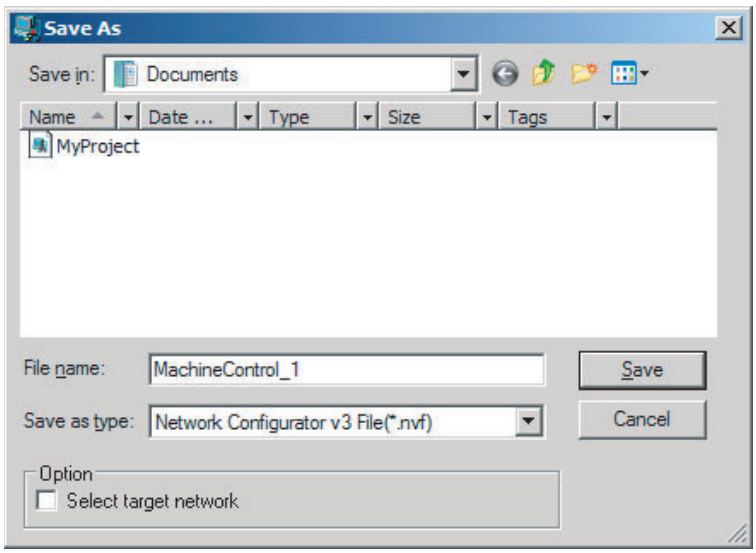
You can save device parameters set in the Network Configurator or device parameters uploaded from the network in a network configuration file.

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



Untitled.nvf is displayed as the default file name.

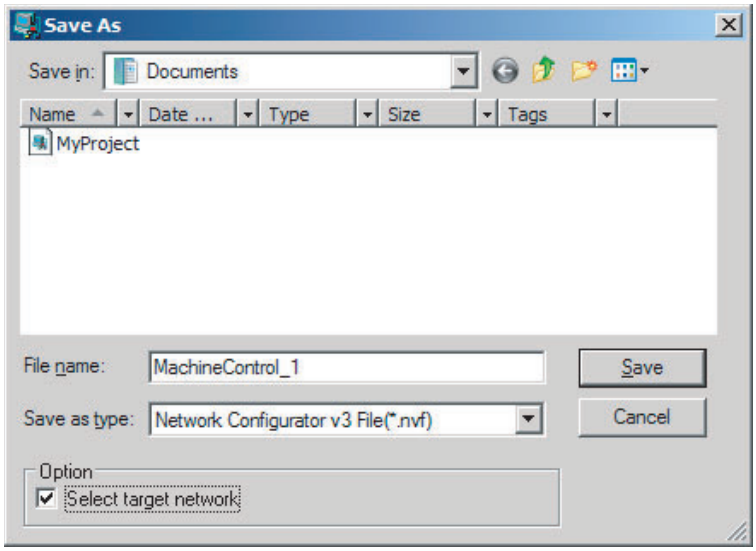
2 Input the file name, and then click the **Save** Button.



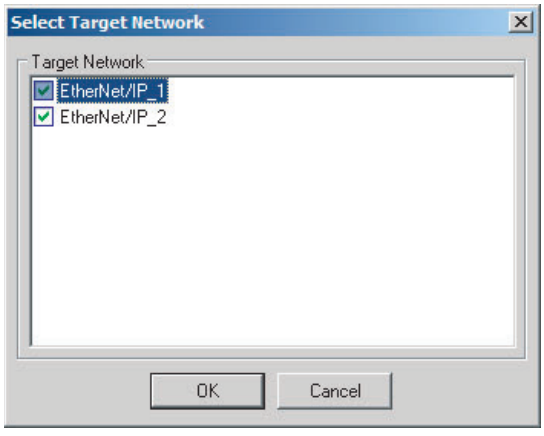
This completes the network configuration file save operation.

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.



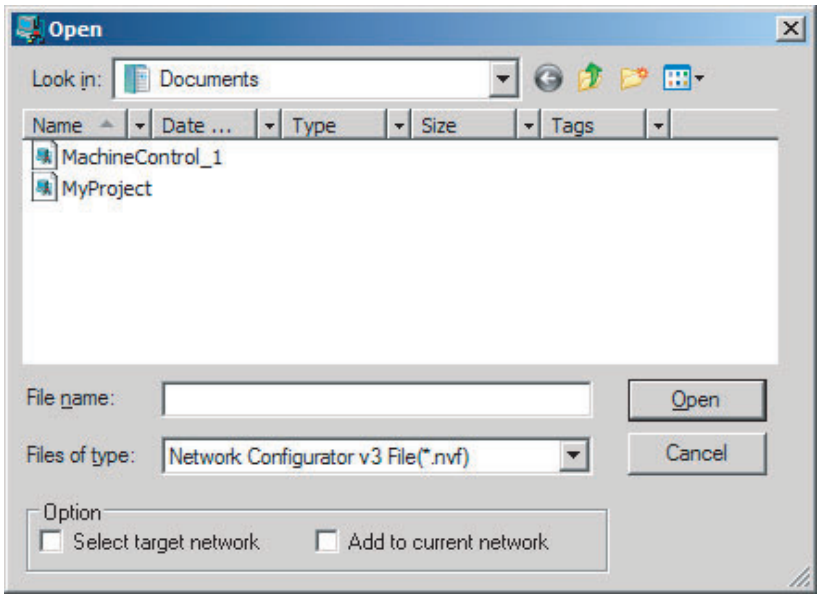
Select the check boxes of the networks to save and click the **OK** Button.



7-2-13 Reading a Network Configuration File

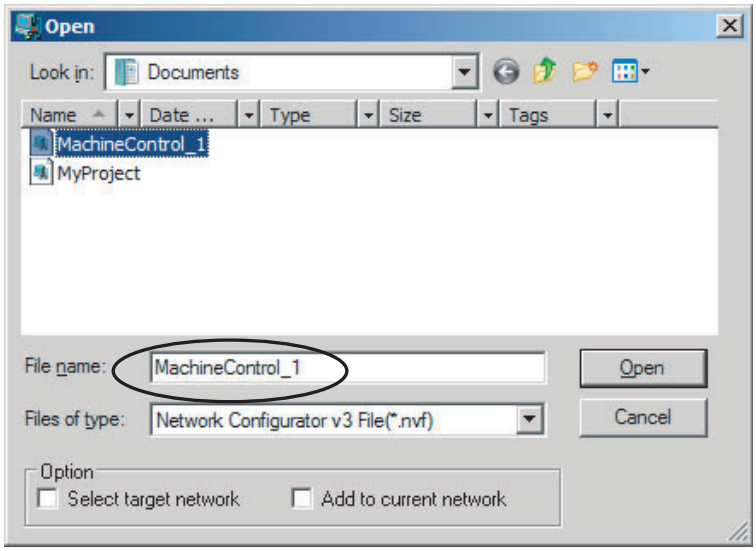
You can read a previously saved network configuration file into the Network Configurator.

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



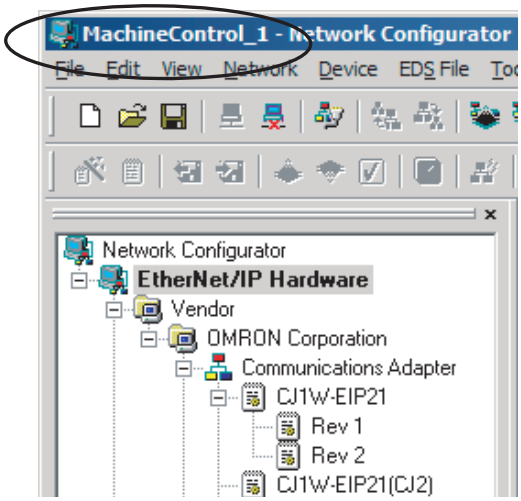
If the network configuration file that you want to read is not displayed, change to another folder.

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.

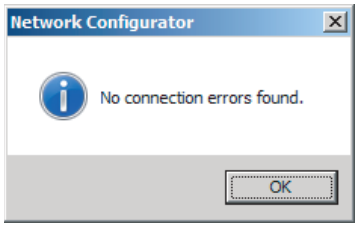
Additional Information

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.

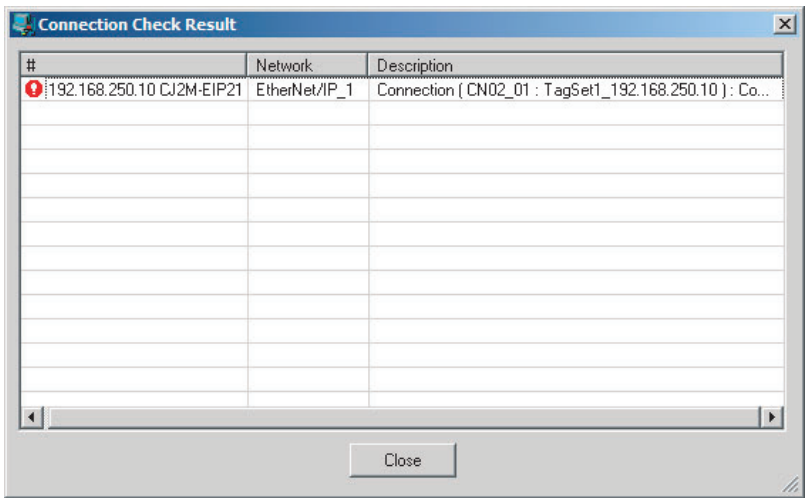
7-2-14 Checking Connections

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.

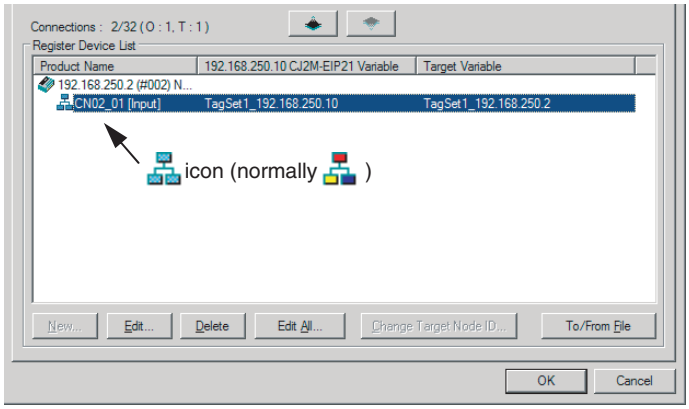
1 Select ***Check Connection*** from the Network Menu.
The following dialog box is displayed if parameters are normal.



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.



7-2-15 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	CS1W-EIP21	CJ1W-EIP21	CS1W-EIP21	CJ2W-EIP21 (CJ2)	CJ2B-EIP21	CJ2M-EIP21	NJ501-1□00
Model before change	**Rev**	**1.01**	**1.01**	**2.01**	**2.01**	**2.01**	**2.01**	**2.01**	**1.01**
CJ1W-EIP21	**1.01**	---	Yes	Yes	Yes	Yes	Yes	*3	*4
CS1W-EIP21	**1.01**	Yes	---	Yes	Yes	Yes	Yes	*3	*4
CJ1W-EIP21	**2.01**	No	No	---	Yes	Yes	Yes	*3	*4
CS1W-EIP21	**2.01**	No	No	Yes	---	Yes	Yes	*3	*4
CJ1W-EIP21 (CJ2)	**2.01**	No	No	*1	*1	---	Yes	*3	*4
CJ2B-EIP21	**2.01**	No	No	*1	*1	Yes	---	*3	*4
CJ2M-EIP21	**2.01**	No	No	*1	*1	Yes	Yes	---	*4
NJ501-1□00	**1.01**	No	No	*5 *9	*5 *9	*7 *9	*7 *9	*8 *9	---

Yes: Can be changed.
No: Cannot be changed.

*1 Cannot be changed if a network variable is specified as a tag.

*2 Cannot be changed if there is a Japanese variable in the tags.

*3 Cannot be changed if the following items exceed the permissible settings of the device after the change: Number of I/O connections, number of tags, number of tag sets, and size of one tag set.

*4 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).

*5 Cannot be changed if a variable that is set in a tag or the tag size is an odd number of bytes.

*6 Cannot be changed if Japanese is specified for the tag or the tag size is an odd number of bytes.

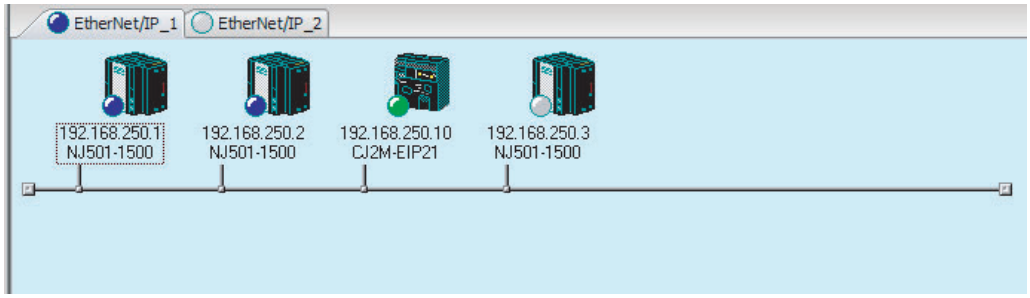
*7 Cannot be changed if a tag or tag set size is an odd number of bytes.

*8 Cannot be changed if tags or tag sets exceed the permissible settings for the refreshing size, or one of the tag sizes is an odd number of bytes.

*9 Cannot be changed if the maximum size of a tag name or tag set name (size after conversion into UTF-8) exceeds 48 bytes.

7-2-16 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 NJ-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*) 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 Target PLC Operating Mode (operating or stopped) (*_EIP_TargetPLCModeSta*) (valid for OMRON Controllers only):**

  (4) The Controller operating mode of the target node is ON.

- **Condition for the Controller error status (fatal or non-fatal error) of the target node (valid for OMRON Controllers only):**

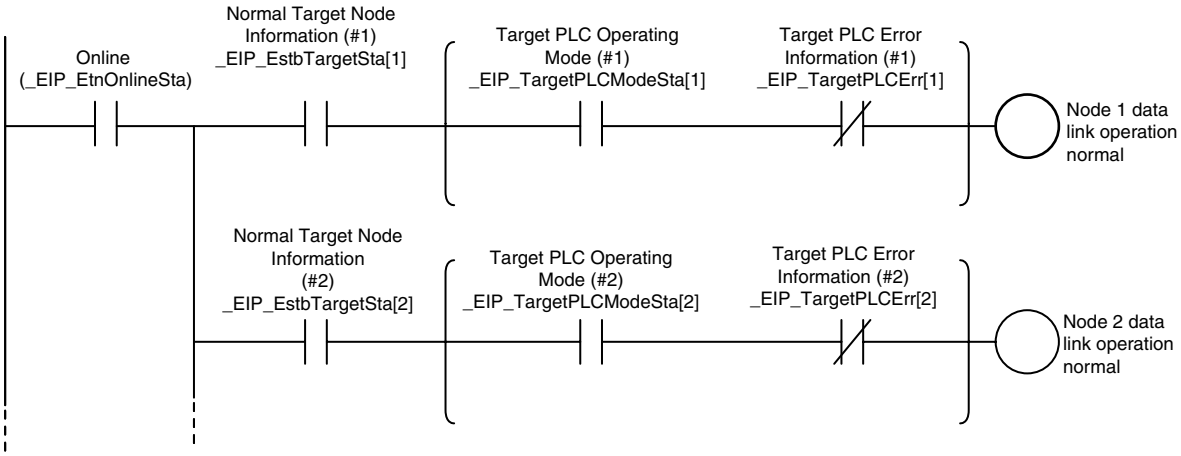
  (5) The target node Controller error information element in the Target PLC Error Information (*_EIP_TargetPLCErr*) is FALSE.

  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.

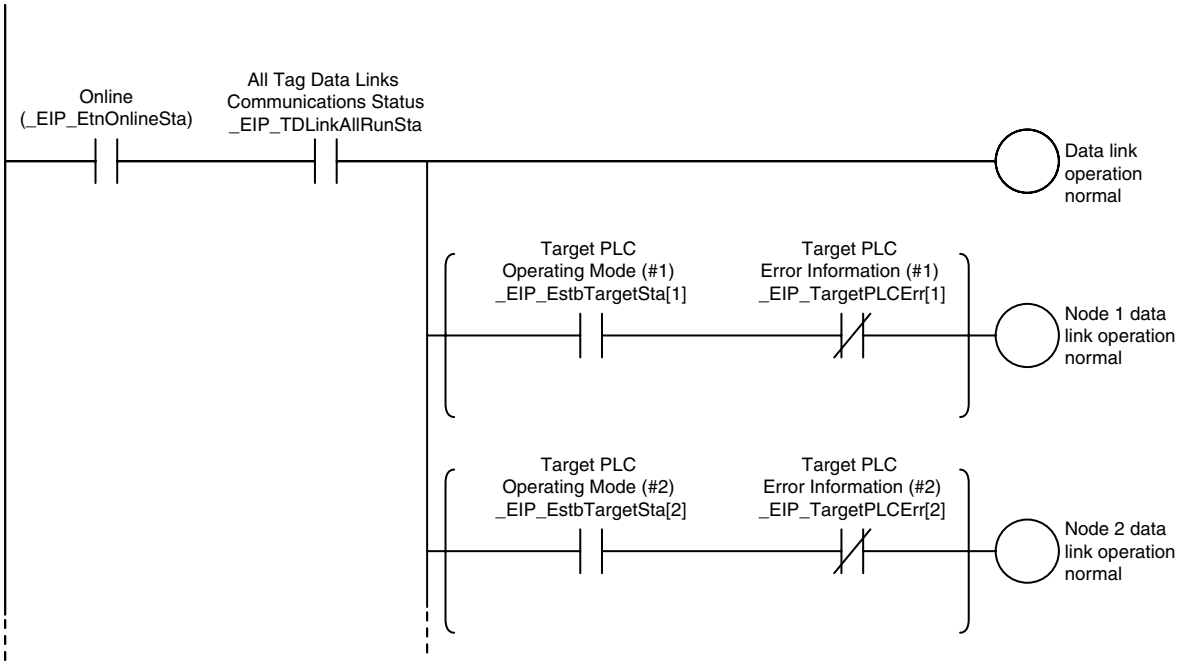
Programming Example to Detect Normal Operation

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

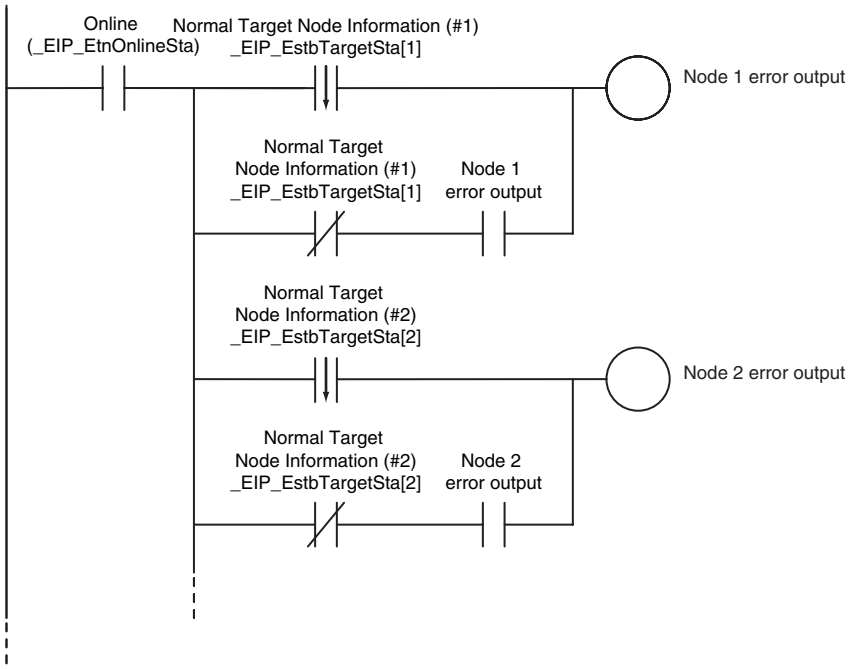


- Programming Example 2 to Detect Normal Operation



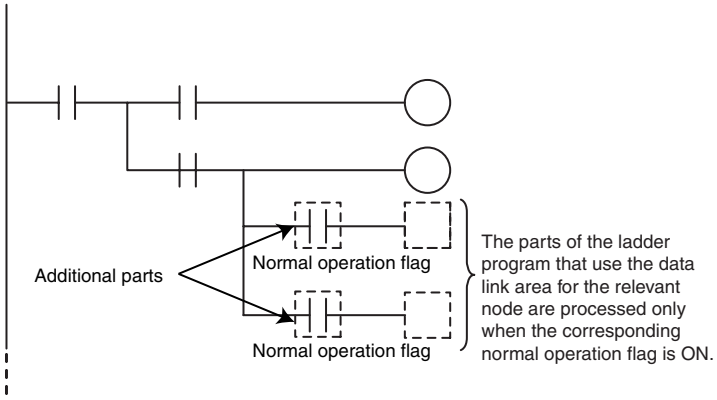
Programming Example to Detect Errors

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.

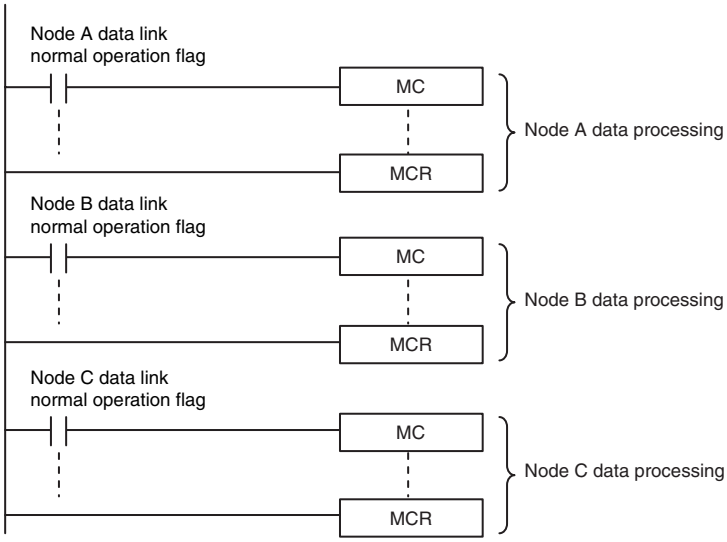


Data Processing Programming Example

- 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 operating information 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. *Array[x]* is TRUE: The target Controller with a node address of x is in operating status. *Array[x]* is FALSE: Other than the above.
_EIP_TargetNodeErr[255] (Target Node Error Information) * Corresponds to the Controller Error Information 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. *Array[x]* 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. *Array[x]* 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 and the Normal Target Node Error Information is TRUE. This is when the Controller 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. *Array[x]* is TRUE: The connection to the node with a node address of x is established normally. *Array[x]* is FALSE: A connection is not established or an error has occurred.

7-4 Tag Data Links with Models Other than NJ-Series CPU Units

The performance of tag data links depends on the CPU Unit and EtherNet/IP Unit model as shown below. When you use tag data links between the built-in EtherNet/IP port on the NJ-series CPU Unit and another CPU Unit or EtherNet/IP Unit, use tag data link settings that match the Unit with the lower communications performance.

● Differences in Tag Data Link Performance Specifications

		NJ-series CPU Unit	CJ2M-CPU3□		CS1W-EIP21, CJ1W-EIP21, or CJ2H-CPU6□-EIP
			Unit version		
			2.0	2.1 or later	
Tags	Total size of all tags	9,600 words	640 words		184,832 words
	Maximum size of 1 tag	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	32		256
Tag sets	Maximum size of 1 tag set	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/tag set when the tag set includes the Controller status) **Note** Input and output variables cannot be combined.			
	Number of registrable tag sets	32	32		256
Connections	Number of connections	32	32		256
	Maximum data size per connection	300 words (Refer to *7-1-7 Concurrency of Tag Data Link Data* 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.)
			(Data concurrency is maintained within each connection.)		
Packet intervals (RPIs)		10 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)		1,000 pps*1	3,000 pps*1		6,000 pps*1

*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 to 1,444 bytes, the system must support a large forward open (an optional CIP specification). The SYSMAC CS/CJ-series Units support a 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 CPU Units 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 CPU Unit are listed in the table below.

Yes: Supported, No: Not supported

CPU Unit		Name in Hardware List of Network Configurator	Network variable name specification	I/O memory address specification
	EtherNet/IP Unit			
NJ-series CPU Unit	---	NJ501-1300, NJ501-1400, or NJ501-1500	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

Message Communications

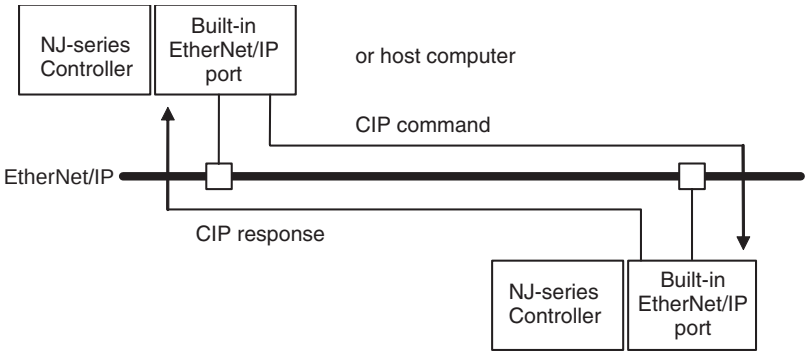
8-1 Overview of the CIP Message Communications Service 8-2
8-1-1 Overview of the CIP Message Communications Service 8-2
8-1-2 Message Communications Service Specifications 8-2
8-2 Using CIP Message Communications 8-3
8-2-1 Overview 8-3
8-2-2 CIP Communications Instructions 8-3
8-2-3 Using CIP Communications Instructions 8-4
8-2-4 Route Path 8-6
8-2-5 Preparing Derivative Data Types to Use CIP Communications Instructions . . 8-9
8-2-6 Sample Programming for CIP Connectionless (UCMM) Message Communications 8-12
8-2-7 Sample Programming for CIP Connection (Class 3) Message Communications 8-18
8-2-8 Operation Timing 8-25
8-2-9 Response Codes 8-26
8-3 CIP Objects Sent to the Built-in EtherNet/IP Port 8-30
8-3-1 CIP Objects Sent to the Built-in EtherNet/IP Port 8-30
8-3-2 Identity Object (Class ID: 01 Hex) 8-30
8-3-3 TCP/IP Interface Object (Class ID: F5 hex) 8-32
8-3-4 Ethernet Link Object (Class ID: F6 Hex) 8-34
8-3-5 Controller Object (Class ID: C4 Hex) 8-38

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 NJ-series CPU Unit 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.



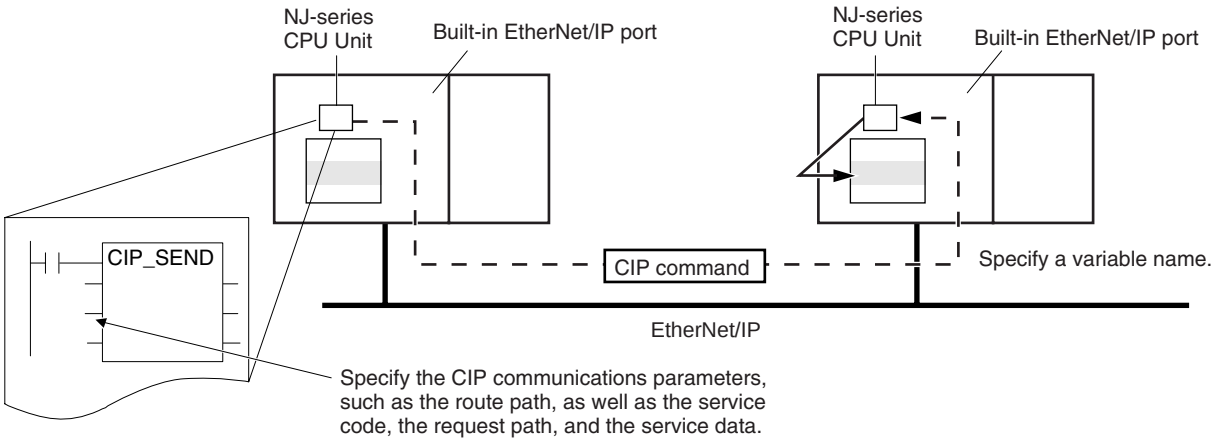
8-1-2 Message Communications Service Specifications

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 CIPUCMM (Read Variable UCMM Explicit) instruction
Data contents	Sending required CIP commands and receiving responses
Communications parameters	Message type, timeout value, and root path specification

8-2 Using CIP Message Communications

8-2-1 Overview

CIP communications instructions can be executed in the user program in the NJ-series CPU Unit to read and write variables in a NJ-series CPU Unit or a CJ2 CPU Unit on the EtherNet/IP network, and send specified CIP commands.



8-2-2 CIP Communications Instructions

The following CIP communications instructions are available. For details on CIP communications instructions, refer to the *NJ-series Instructions Reference Manual* (Cat. No. W502).

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 *8-2-9 Response Codes* and *8-3 CIP Objects Sent to the Built-in EtherNet/IP Port* for information on the service codes and response codes that are used with the NJ-series CPU Units.	

Instruction	Name	Description	Communications method
CIPOpen	Open CIP Class 3 Connection	This instruction opens a CIP class 3 connection with the specified remote node.	CIP class 3 connection message
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 *8-2-9 Response Codes* and *8-3 CIP Objects Sent to the Built-in EtherNet/IP Port* for information on the service codes and response codes that are used with the NJ-series CPU Units.	
CIPClose	Close CIP Class 3 Connection	This instruction closes the CIP class 3 connection that is specified by the handle.	

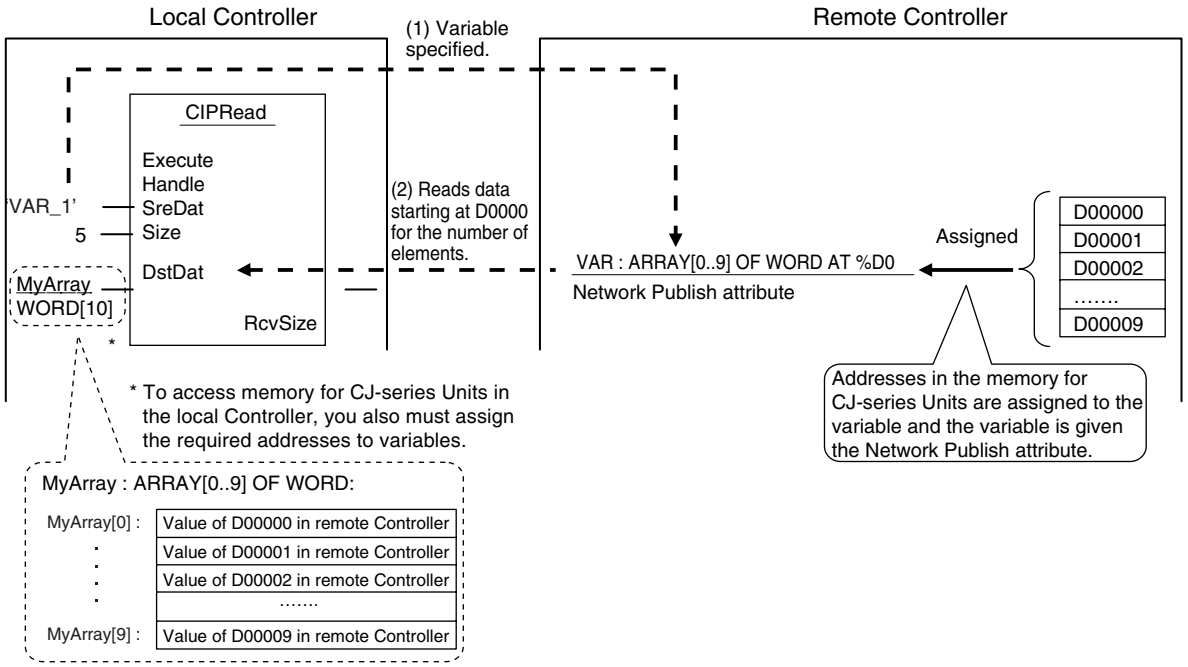
8-2-3 Using CIP Communications Instructions

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*1 (only for CIP class 3 connections)	Execute this process before you use a CIP message. Open processing is continued until a connection is established.	CIPOpen
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 There can be up to 32 handles at the same time from opening connections. 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. To access memory for CJ-series Units, access a variable with an AT specification.



Precautions for Correct Use

You can execute a maximum of 32 CIP communications instructions at the same time. This includes all CIP communications instructions. Use exclusive control in the user program so that no more than 32 socket service instructions are executed at the same time.

8-2-4 Route Path

The route path indicates the path from the local CPU Unit to the remote Controller on the network. Routing is performed for CIP communications instructions based on route paths.

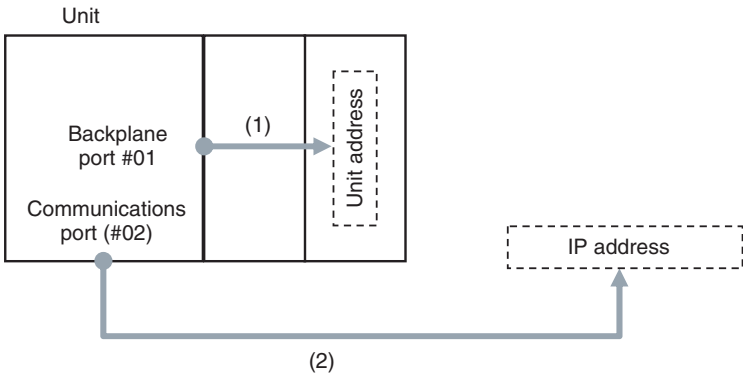
Route Path Notation

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 remote address are determined as shown in the following table according to whether the route type is (1) a Unit on the CPU Rack or (2) a communications port on a Communications Unit.

Route	Network type number (hexadecimal)	Remote address (hexadecimal)
1. Output to Unit on CPU Rack	#01 (backplane port)	Remote Unit address (Refer to *Additional Information* below.)
2. Output from communications port on Communications Unit	#02 (Built-in EtherNet/IP port)	IP address



(1) When Routing the Output to a Unit on the CPU Rack

Output the command to the backplane port as a network with the CPU Rack. Specify the Unit address as the address of the remote Unit.

(2) When Routing the Output to a Communications Port on a Communications Unit

Output the command to an EtherNet/IP 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.

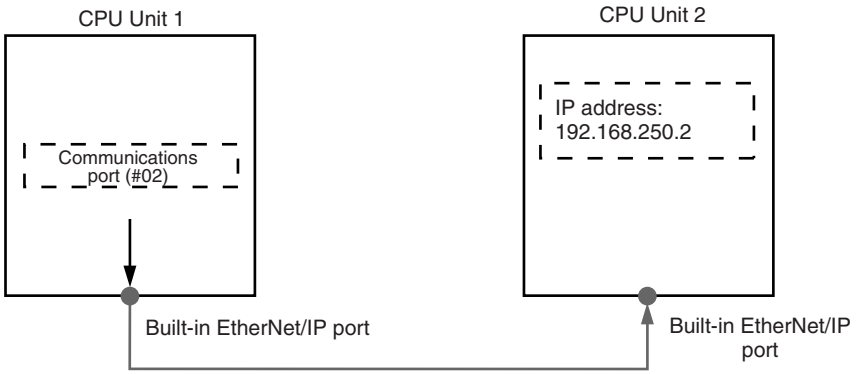
- CPU Unit: 00 hex
- CPU Bus Units (EtherNet/IP Units): Unit number + 10 hex
- Special I/O Unit: Unit number + 20 hex
- Inner Board: E1 hex
- Computer: 01 hex
- The Unit connected to network: FE hex

Route Path Notation Examples

The notation of the route path is different for communications on the built-in EtherNet/IP port and for communication on an EtherNet/IP Unit. This section provides examples of route paths.

● Communicating between Built-in EtherNet/IP Ports

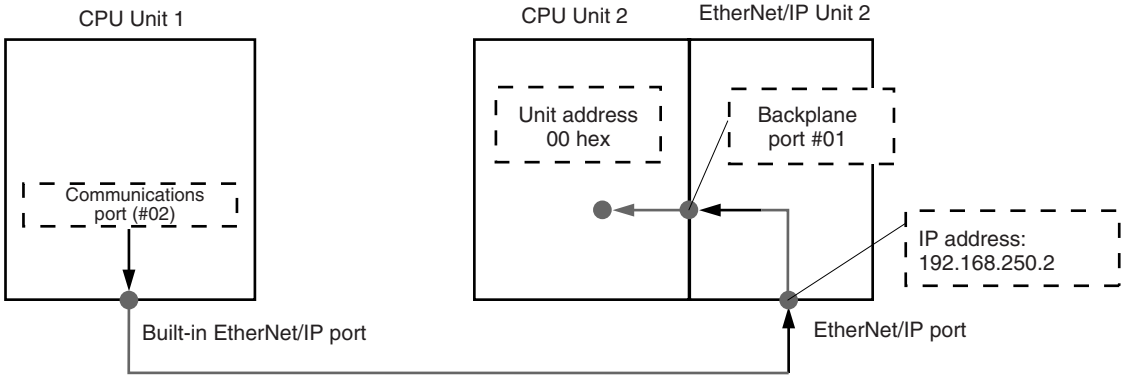
Example: Communicating between Built-in EtherNet/IP Ports on CPU Unit 1 and CPU Unit 2



- Network type number: "#02" (Output the command via EtherNet/IP port.)
- Remote address: Specify the remote IP address.
- Route path: \02\192.168.250.2

● Communicating from a Built-in EtherNet/IP Port to an EtherNet/IP Unit

Example: Communicating from the built-in EtherNet/IP port on CPU Unit 1 to CPU Unit 2 via the EtherNet/IP Unit mounted to CPU Unit 2



(1) CPU Unit 1 to EtherNet/IP Unit 2

- Network type number: "#02" (Output the command via EtherNet/IP port.)
- Remote address: Specify the remote IP address.

(2) EtherNet/IP Unit 2 to CPU Unit 2

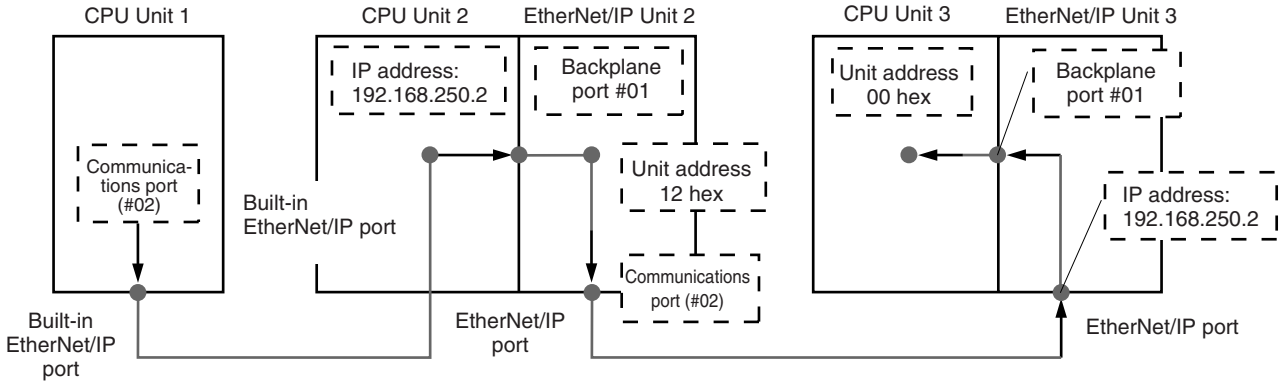
- Network type number: "#01" (Output the command via internal backplane port.)
- Remote address: "#00" (unit address of CPU Unit)

Route path : \02\192.168.250.2\01\#00
(1) (2)

(1) IP address of remote EtherNet/IP Unit
(2) Unit address of remote CPU Unit

● Accessing via a Relay Node

Example: Communicating from CPU Unit 1 to CPU Unit 3 via CPU Unit 2



(1) CPU Unit 1 to CPU Unit 2

- Network type number: "#02" (Output the command via EtherNet/IP port.)
- Remote address: Specify the remote IP address.

(2) CPU Unit 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 CPU Unit 3

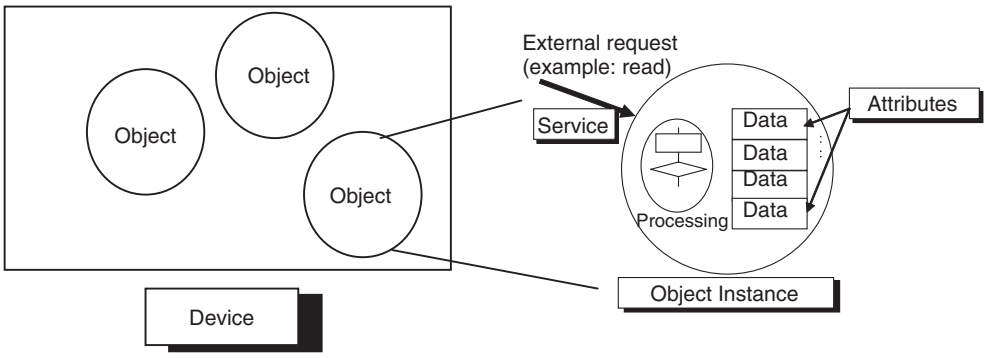
- Network type number: "#01" (Output the command via internal backplane port.)
- Remote address: "#00" (unit address of CPU Unit)

Route path : \02\192.168.250.2\01\#12\02\192.168.250.3\01\#00
(1) (2) (3) (4)

(1) Unit address of Local EtherNet/IP Unit (Unit number: 1+10 hex)
(2) IP address of remote EtherNet/IP Unit
(3) Unit address of remote CPU Unit

Request Path (IOI)

In the CIP world, each device is modeled as a collection of objects. An Object abstractly represents the specific configuration elements of a device.

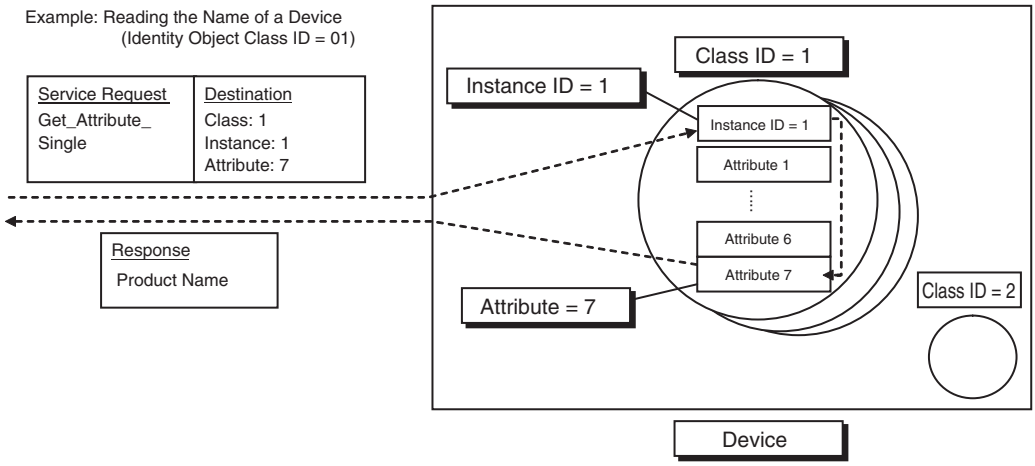


In the CIP Common Specification, “Object,” “Class,” “Instance,” “Attribute” and “Service” are defined as follows: (Source: CIP Common Specification)

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 *8-3 CIP Objects Sent to the Built-in EtherNet/IP Port* for the class ID, instance ID, attribute ID, and service code for each object.

8-2-5 Preparing Derivative Data Types to Use CIP Communications Instructions

To use CIP communications instructions, you must prepare derivative data type variables to input communications parameter settings in advance. Refer to the *NJ-series CPU Unit Software User's Manual* (Cat. No. W501) for the specifications and setting procedures for the variables that are described in this section.

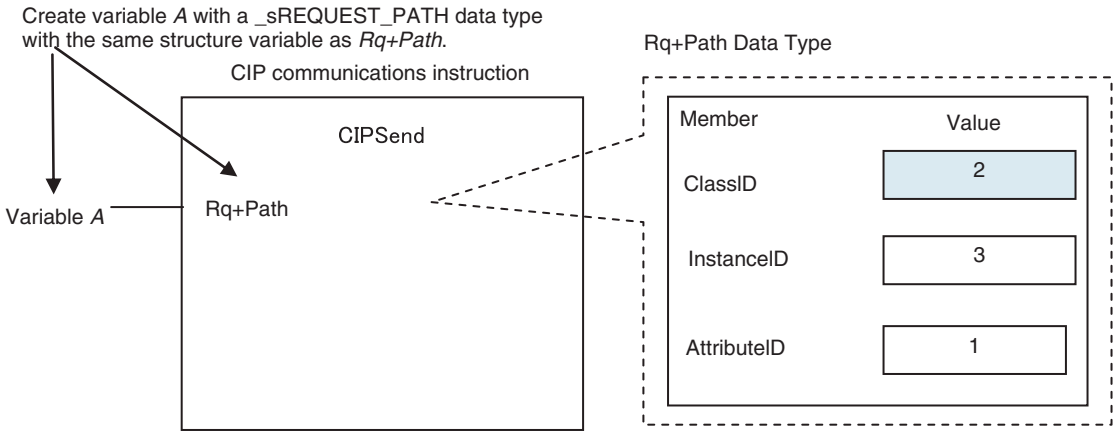
Providing the Structure Variables to Input Request Paths

A CIP communications instruction combines all of the objects in a request path into a single structure variable.

● Creating Structure Variables

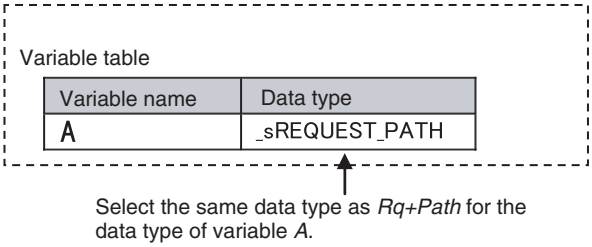
To input a value into the structure variable of a CIP communications instruction, you must create a user-defined variable with the same configuration in advance.

Example: Creating a Variable to Input Data to the CIPSend Instruction Array Variable *Rq+Path*



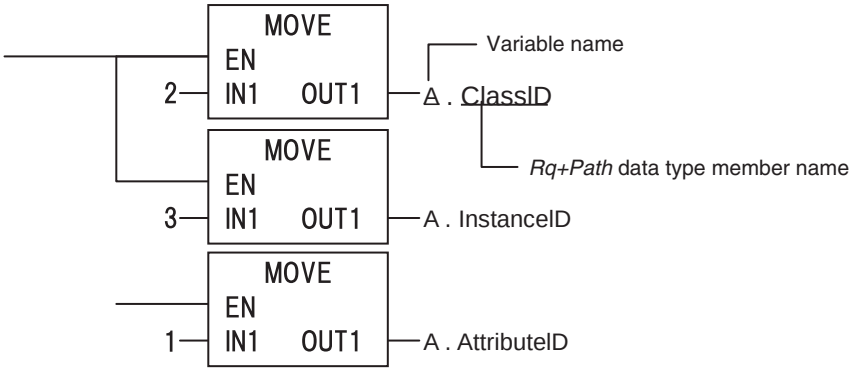
To create a variable in a variable table, select the pre-registered CIP communications instruction data type _sREQUEST_PATH. This is a structure variable with the same configuration as *Rq+Path*.

Example:



● Inputting the Values for Each Structure Variable Member

Input the following values into the communications parameters that were registered as members of the structure variable.



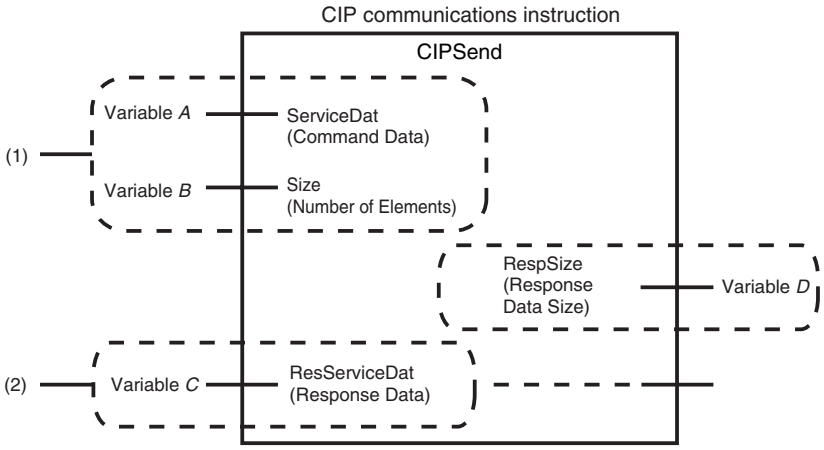
Preparing Array Variables to Input and Output Service Data and Response Data

CIP communications instructions send and receive data that is stored in array variables.

● 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.

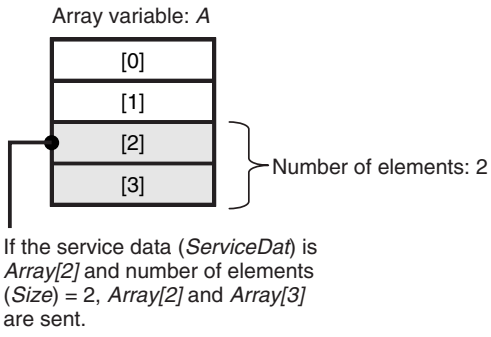
Example: Creating a Variable to Input Data to the CIPSend Instruction Array Variables

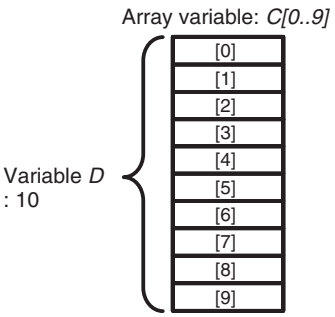


(1) Input the service data to send.

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.



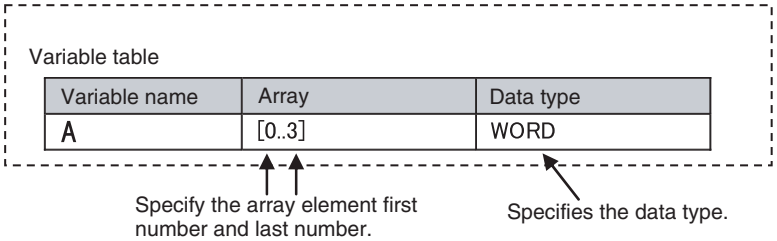


Use the following procedure to create a variable in the variable table.

1 Select the *Array* Check Box.

2 Specify the element first number, the element last number, and the data type.

Example:



● 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)	---

8-2-6 Sample Programming for CIP Connectionless (UCMM) Message Communications

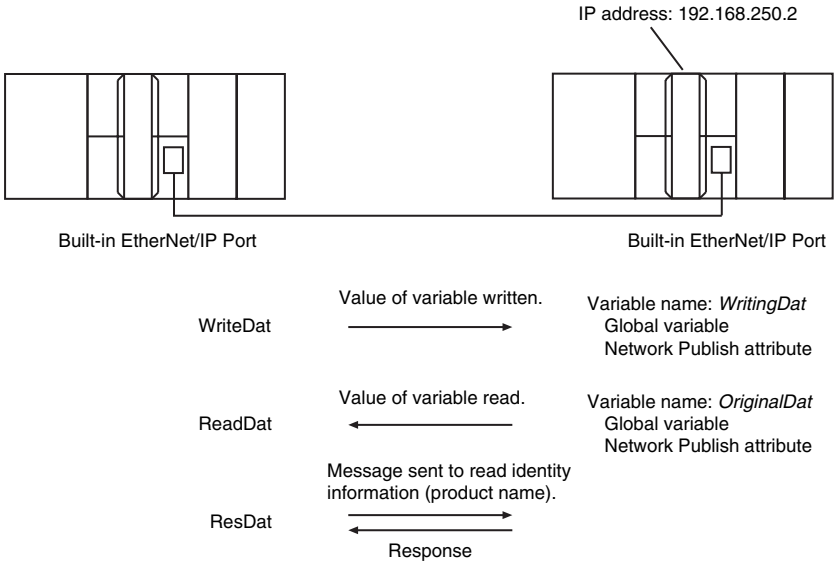
This sample uses CIP UCMM messages to write a variable, read a variable, and send a message. The Controllers are connected to an EtherNet/IP network. The IP address of the remote node is 192.168.250.2. The following procedure is used.

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.

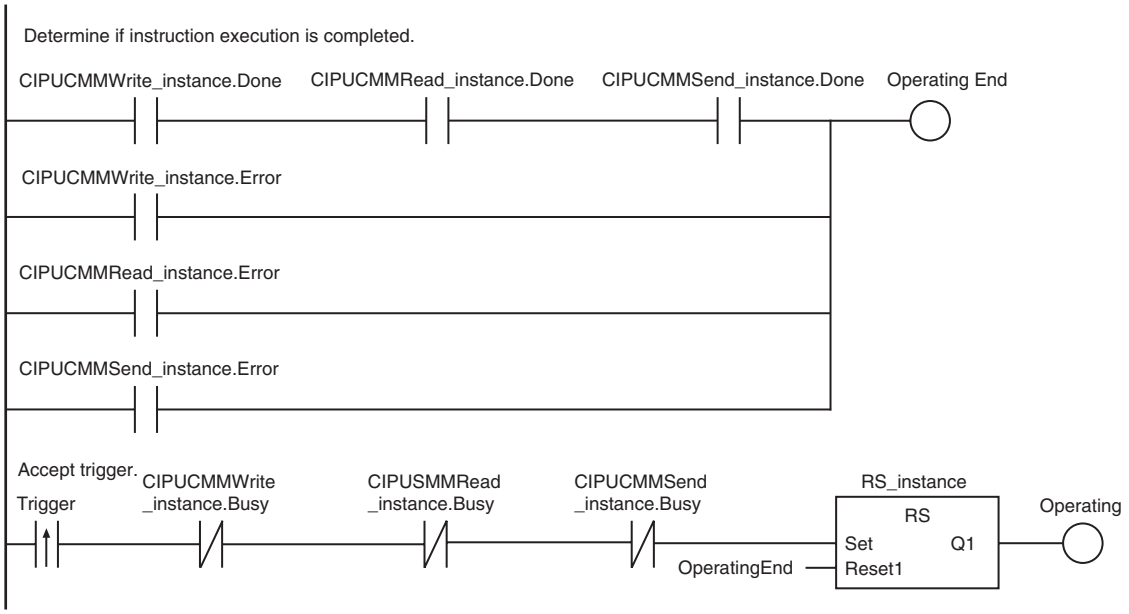
3 The CIPUCMMSend instruction is used to send an explicit message to a remote node. The contents of the message is to read identity information (product name). The class ID, instance ID, attribute ID, and service code are as follows. The response data is stored in the ResDat variable.

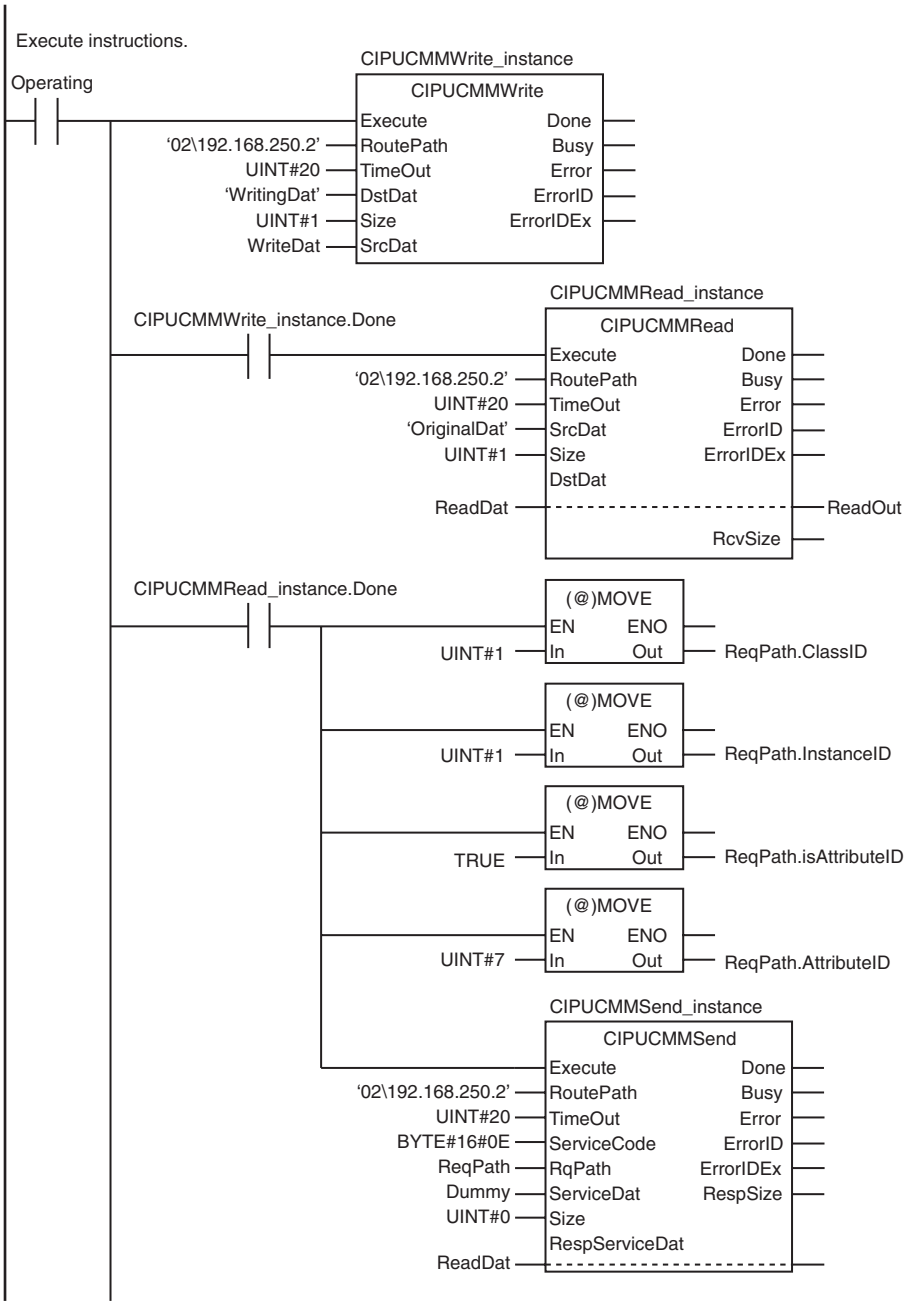
Item	Value
Class ID	1
Instance ID	1
Attribute ID	7
Service code	16#0E

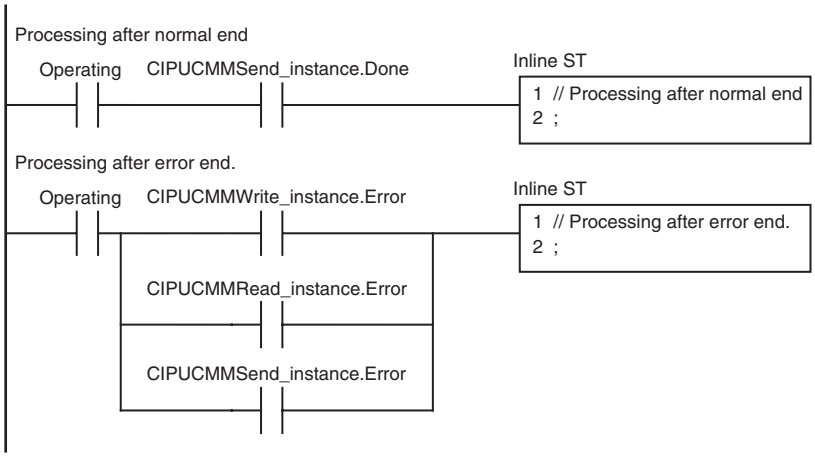


LD

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
ResDat	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		




Execute instructions.
Operating
CIPUCMMWrite_instance
CIPUCMMWrite
Execute
Done
'02\192.168.250.2'
RoutePath
Busy
UINT#20
TimeOut
Error
'WritingDat'
DstDat
ErrorID
UINT#1
Size
ErrorIDEx
WriteDat
SrcDat
CIPUCMMRead_instance
CIPUCMMWrite_instance.Done
CIPUCMMRead
Execute
Done
'02\192.168.250.2'
RoutePath
Busy
UINT#20
TimeOut
Error
'OriginalDat'
SrcDat
ErrorID
UINT#1
Size
ErrorIDEx
DstDat
ReadDat
ReadOut
RcvSize
CIPUCMMRead_instance.Done
(@)MOVE
EN
ENO
UINT#1
In
Out
ReqPath.ClassID
(@)MOVE
EN
ENO
UINT#1
In
Out
ReqPath.InstanceID
(@)MOVE
EN
ENO
TRUE
In
Out
ReqPath.isAttributeID
(@)MOVE
EN
ENO
UINT#7
In
Out
ReqPath.AttributeID
CIPUCMMSend_instance
CIPUCMMSend
Execute
Done
'02\192.168.250.2'
RoutePath
Busy
UINT#20
TimeOut
Error
BYTE#16#0E
ServiceCode
ErrorID
ReqPath
RqPath
ErrorIDEx
Dummy
ServiceDat
RespSize
UINT#0
Size
RespServiceDat
ReadDat


Processing after normal end
Operating
CIPUCMMSend_instance.Done
Inline ST
1 // Processing after normal end
2 ;
Processing after error end.
Operating
CIPUCMMWrite_instance.Error
Inline ST
1 // Processing after error end.
2 ;
CIPUCMMRead_instance.Error
CIPUCMMSend_instance.Error

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	0	Write data
	ReadDat	INT	0	Read data
	ReqPath	_sREQUEST_PATH	(ClassID:=0, InstanceID:=0, isAttributeID:=False, AttributeID:=0)	Request path
	ResDat	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	BOOL	☑	Online

```
// 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  :=ResDat);                  // 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;
```

8-2-7 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 instruction is used to open a class 3 connection. The timeout time is 2 s.

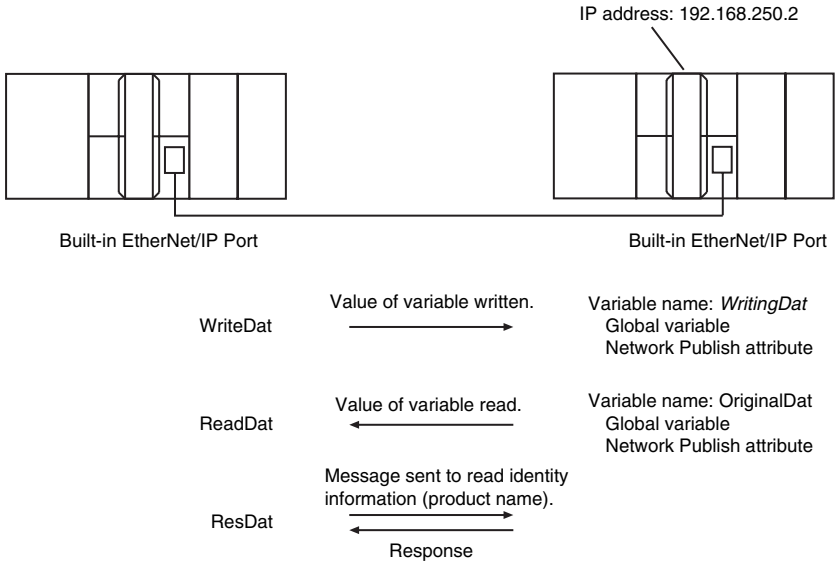
2 The CIPWrite instruction is used to write the value of a variable at a remote node. The variable name at the remote node is *WritingDat* and the contents of the *WriteDat* is written to it. *WritingDat* must be defined as a global variable at the remote node and the Network Publish attribute must be 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 ResDat variable.

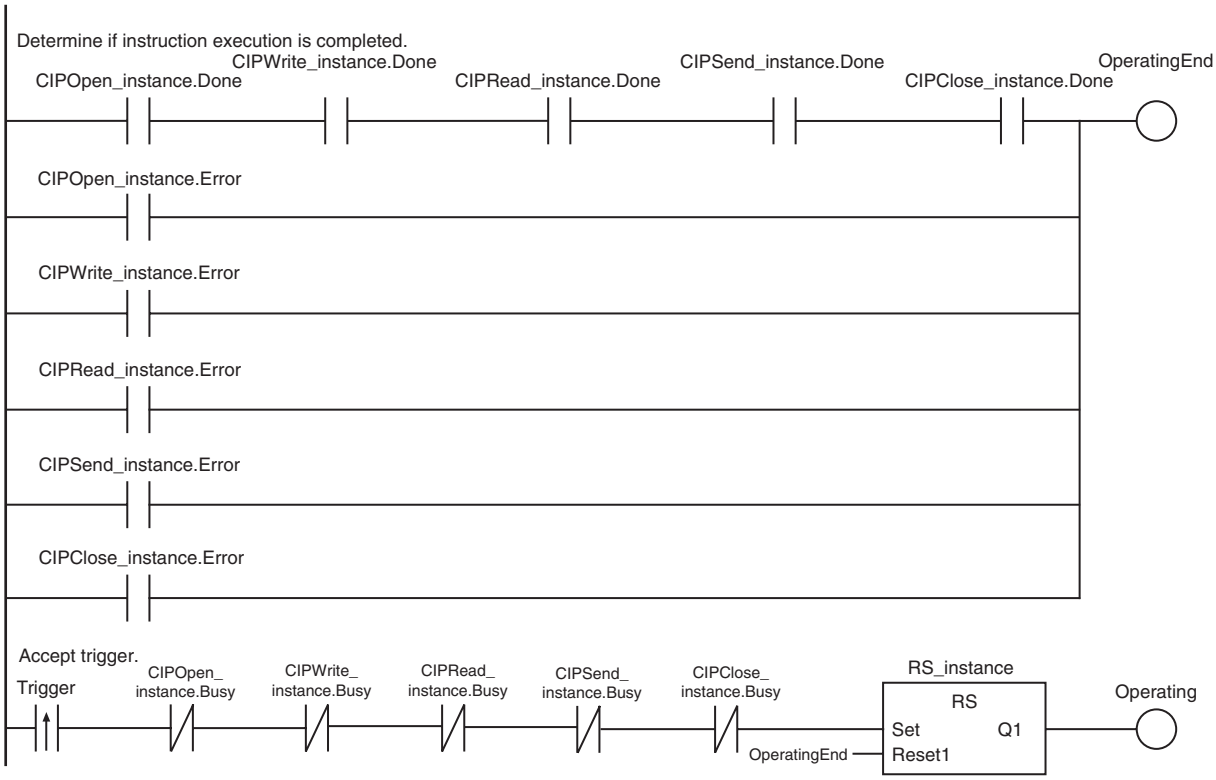
Item	Value
Class ID	1
Instance ID	1
Attribute ID	7
Service code	16#0E

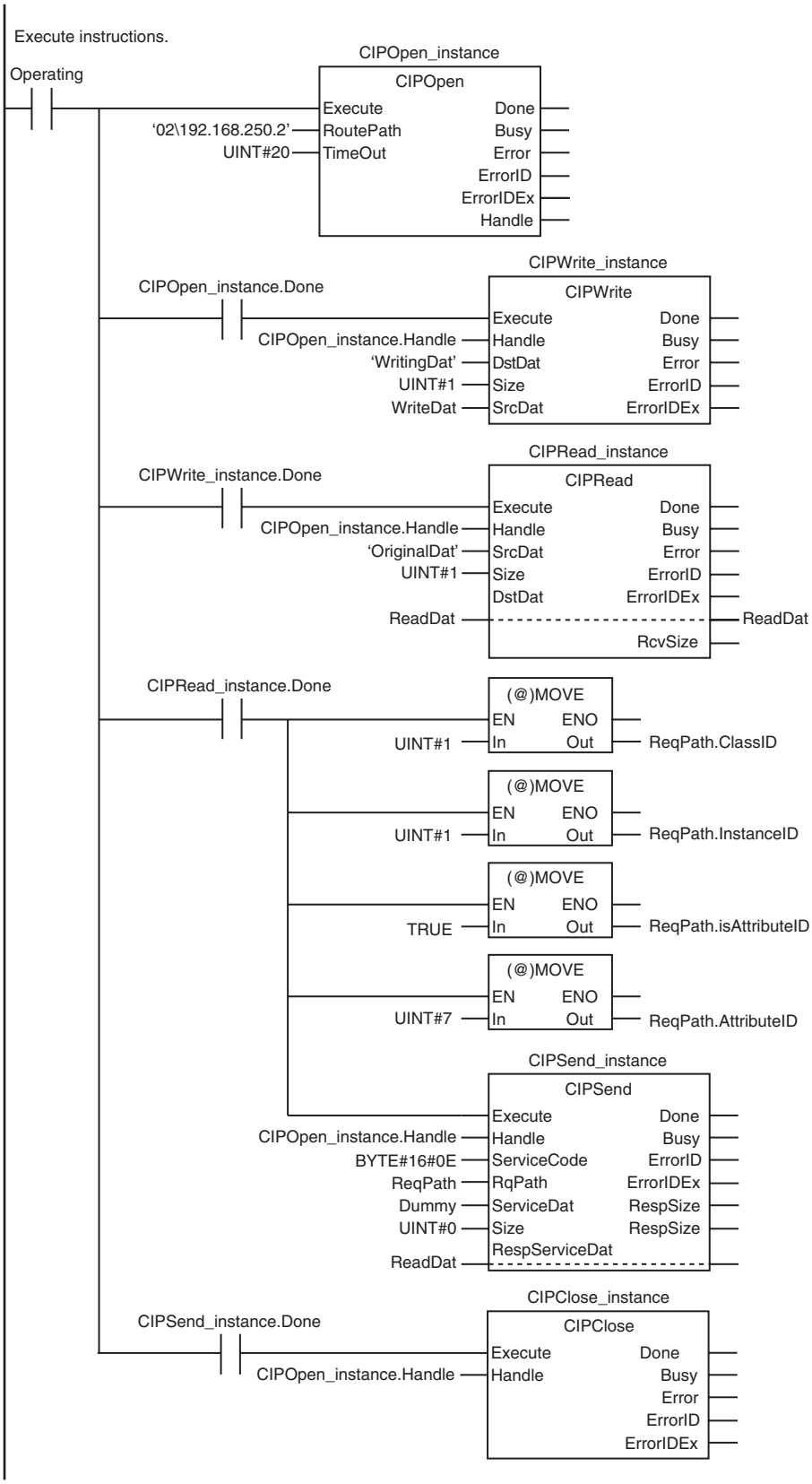
5 The CIPClose instruction is used to close the class 3 connection.

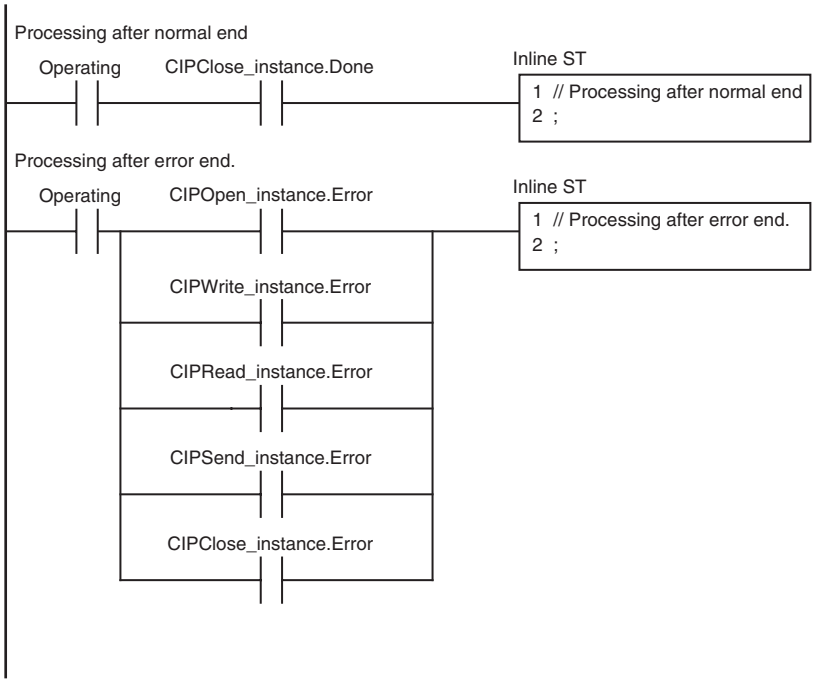


LD

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
ResDat	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		




Execute instructions.
Operating
CIPOpen_instance
CIPOpen
Execute
RoutePath
TimeOut
Done
Busy
Error
ErrorID
ErrorIDEx
Handle
'02\192.168.250.2'
UINT#20
CIPWrite_instance
CIPWrite
CIPOpen_instance.Done
Execute
Handle
DstDat
Size
SrcDat
Done
Busy
Error
ErrorID
ErrorIDEx
CIPOpen_instance.Handle
'WritingDat'
UINT#1
WriteDat
CIPRead_instance
CIPRead
CIPWrite_instance.Done
Execute
Handle
SrcDat
Size
DstDat
ReadDat
Done
Busy
Error
ErrorID
ErrorIDEx
RcvSize
CIPOpen_instance.Handle
'OriginalDat'
UINT#1
ReadDat
CIPRead_instance.Done
(@)MOVE
EN
In
ENO
Out
UINT#1
ReqPath.ClassID
UINT#1
ReqPath.InstanceID
TRUE
ReqPath.isAttributeID
UINT#7
ReqPath.AttributeID
CIPSend_instance
CIPSend
Execute
Handle
ServiceCode
RqPath
ServiceDat
Size
RespServiceDat
Done
Busy
ErrorID
ErrorIDEx
RespSize
RespSize
CIPOpen_instance.Handle
BYTE#16#0E
ReqPath
Dummy
UINT#0
ReadDat
CIPClose_instance
CIPClose
CIPSend_instance.Done
Execute
Handle
Done
Busy
Error
ErrorID
ErrorIDEx
CIPOpen_instance.Handle


Processing after normal end
Operating
CIPClose_instance.Done
Inline ST
1 // Processing after normal end
2 ;
Processing after error end.
Operating
CIPOpen_instance.Error
Inline ST
1 // Processing after error end.
2 ;
CIPWrite_instance.Error
CIPRead_instance.Error
CIPSend_instance.Error
CIPClose_instance.Error

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	0	Write data
	ReadDat	INT	0	Read data
	ReqPath	_sREQUEST_PATH	(ClassID:=0, InstanceID:=0, isAttributeID:=False, AttributeID:=0)	Request path
	ResDat	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	BOOL	☑	Online

```
// 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(                              // Initialize instance.
        Execute        :=FALSE,                    // Dummy
        DstDat         :=ReadDat);                 // Dummy
    CIPSend_instance(
        Execute        :=FALSE,                    // Initialize instance.
        ServiceDat     := Dummy,                   // Dummy
        RespServiceDat    :=RespDat);              // Dummy
    CIPClose_instance(Execute:=FALSE);             // Initialize instance.
END_IF;
```

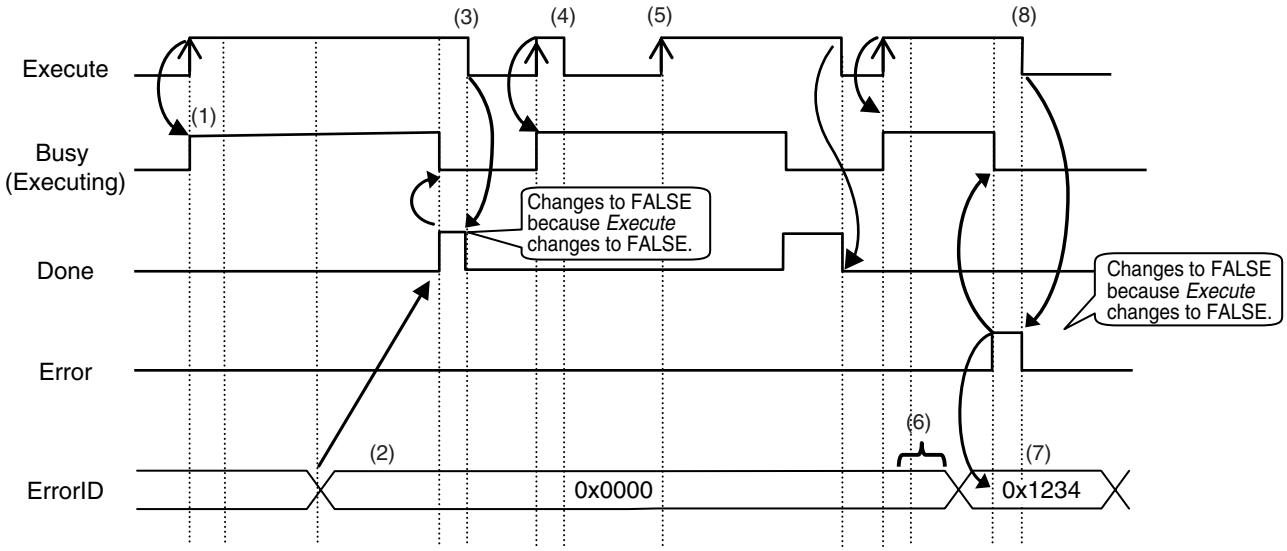
```
IF (DoCIPTrigger=TRUE) THEN
    CASE Stage OF
    1 :                                             // Open CIP class 3 connection.
        CIPOpen_instance(
            Execute     :=TRUE,
            TimeOut     :=UINT#20,                  // Timeout time: 2.0 s
            RoutePath   :='02/192.168.250.2');      // 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:=ResDat);                // 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.
        CIPClose_instance(
            Execute     :=TRUE,
            Handle      :=CIPOpen_instance.Handle);  // Handle

        IF (CIPClose_instance.Done=TRUE) THEN
            Stage       :=INT#0;
        ELSIF (CIPClose_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;
```

8-2-8 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).

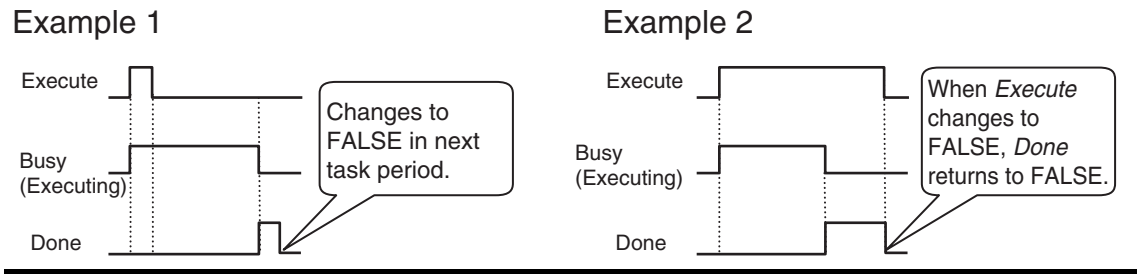
6 If an error occurs, several retries are attempted internally. The error code in the *ErrorID* is not updated during the retries.

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)

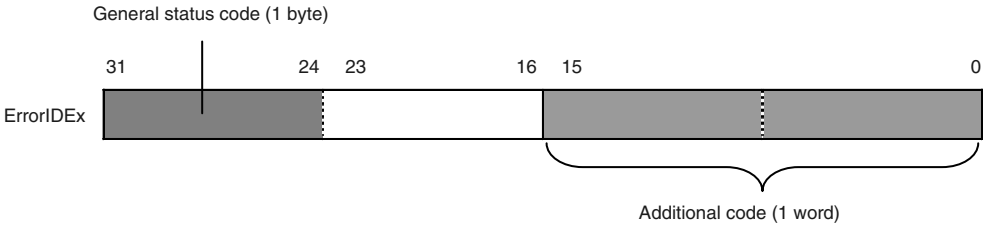


8-2-9 Response Codes

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.
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.

General status code (hex)	Status name	Description of status
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.
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.

General status code (hex)	Status name	Description of status
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 *Service not supported*, *Attribute not supported*, or *Attribute not settable*.
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.
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.

General Status (hex)	Additional Status (hex)	Description
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 Objects Sent to the Built-in EtherNet/IP Port

8-3-1 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	• Reads ID information from the CPU Unit. • Resets the built-in EtherNet/IP port.	page 8-30
TCP/IP interface object	Writes and reads TCP/IP settings.	page 8-32
Ethernet link object	• Reads Ethernet settings. • Reads Ethernet status.	page 8-34
Controller object	• Gets the Controller status. • Changes the operating mode of the Controller.	page 8-38

8-3-2 Identity Object (Class ID: 01 Hex)

This object reads the ID information of the CPU Unit and resets the built-in EtherNet/IP port.

● Class ID

Specify 01 hex.

● Instance ID

You can specify only 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	0001 Hex
02 hex	Device Type	Device type	Read	UINT	0000 Hex
03 hex	Product Code	Product code	Read	UINT	0001 Hex
04 hex	Revision	Device revision	Read	Struct	---
	Max Instance	Major revision	Read	USINT	01 Hex
	Revision	Minor revision	Read	USINT	01 Hex
05 hex	Status	Status of the built-in EtherNet/IP port	Read	WORD	Refer to (1) *Status Details of the Built-in EtherNet/IP Port*, below.

Attribute ID	Parameter name	Description	Attribute	Read data	
				Data type	Value
06 hex	Serial Number	Serial number	Read	UINT	Set value
07 hex	Product Name	Product name	Read	STRING	Set value

(1) 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. b7 b6 b5 b4 0 1 0 1: There is a major fault. 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. • TCP/IP Advanced Setting Error • DNS Server Connection Error • Tag Data Link Setting Error • Tag Data Link Error • Tag Data Link Connection Failed • FTP Server Settings Error • NTP Client Setting Error • SNMP Settings Error • NTP Server Connection Error • Tag Resolution Error
9	Minor Unrecoverable Fault	TRUE when the following error occurs. • Identity Error
10	Major Recoverable Fault	TRUE when any of the following errors occurs. • IP Address Duplication Error • BOOTP Server Error • Basic Ethernet Setting Error • TCP/IP Basic Setting Error
11	Major Unrecoverable Fault	TRUE when any of the following errors occurs. • Communications Controller Error • MAC Address Error
12 to 15	Reserved	Always FALSE.

● 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	Resets the built-in EtherNet/IP port. This parameter is used to reset the built-in EtherNet/IP port when you change the IP address or other parameter settings and want to apply them. Input one of the following values for the *ServiceDat* input variable to the CIPSend instruction to specify the reset method. 00 hex: Restart the built-in EtherNet/IP port. 01 hex: Clear the tag data link settings and restart.	Not supported.	Supported.

● 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	• Specifying a service for a class: 00 hex • Specifying a service for an instance: Always 01 hex	Not required.
0E hex	Get_Attribute_Single			• Reading a class attribute: 01 or 02 hex • Reading an instance attribute: 01 to 07 hex
05 hex	Reset		Always 01 hex	Not required.

8-3-3 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.

● Class ID

Specify F5 hex.

● Instance ID

You can specify only 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/write data	
				Data type	Value
01 hex	Interface Configration Status	Indicates the IP address settings status of the built-in EtherNet/IP port.	Read	DWORD	Bit 0: IP address is not set (also during startup of BOOTP). Bit 1: IP address is set.
02 hex	Configuration Capability	Indicates a Controller Configurations and Setup that can be set to the built-in EtherNet/IP port.	Read	DWORD	Bit 0: BOOTP Client: Always TRUE. 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. Bits 5 to 31: Reserved (always FALSE).
03 hex	Configuration Control	Sets the method used to set the IP address when the EtherNet/IP port starts.	Write	DWORD	Bit 0: Static IP address. Bit 1: Set by BOOTP.
04 hex	Physical Link Object	The path to the link object in the physical layer.	Read	Struct	0000 hex
	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 01 hex
05 hex	Interface Cofiguration	The built-in EtherNet/IP port settings.	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 (reserved).		UDINT	Always 0000 hex.
	Domain Name	The domain name (reserved).		STRING	Always 0000 hex.
06 hex	Host Name	The host name (reserved).	Write	STRING	Always 0000 hex.

● 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.
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.	No supported.	Supported.

● 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	• Specifying a service for a class: 00 hex • Specifying a service for an instance: 01 hex	Not required.
0E hex	Get_Attribute_Single			• Reading a class attribute: 01 or 02 hex • Reading and writing an instance attribute: 01 to 06 hex
10 hex	Set_Attribute_Single			

8-3-4 Ethernet Link Object (Class ID: F6 Hex)

This object is used to set and read Ethernet communications and read Ethernet communications status information.

● Class ID

Specify F6 hex.

● Instance ID

You can specify only 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/write data	
				Data type	Value
01 hex	Interface Speed	Gives the baud rate for the built-in EtherNet/IP port.	Read	UDINT	Reads the current value.
02 hex	Interface Flags	Gives the status of the built-in EtherNet/IP port.	Read	DWORD	Refer to *(1) Interface Flag Details*, below.
03 hex	Physical Address	Gives the MAC address of the built-in EtherNet/IP port.	Read	ARRAY [0...5] OF USINT	Reads the current value of the MAC address.

Attribute ID	Parameter name	Description	Attribute	Read/write data	
				Data type	Value
04 hex	Interface Counters	The path to the link object in the physical layer	Read	Struct	---
	In Octets	The number of octets received through the interface. This includes unnecessary multicast packets and discarded packets counted by *InDiscards*.		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 *InDiscards*.		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 *InDiscards*.		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 *InDiscards*.		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.	Read	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 built-in EtherNet/IP port.	Read	Struct	Reads the current value.
	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.		STRING	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	Counter for the built-in EtherNet/IP port.	Write	Struct	---
	Control Bits	*AutoNego* for Ethernet communications that specifies full duplex.		WORD	Refer to *(2) Control Bit Details*, below.
	Forced Interface Speed	Gives the set value of the Ethernet baud rate.		UINT	Reads the set 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 (communications setup is always set to 10 Mbps). TRUE: Auto-negotiation is enabled (communications setup is automatically set to 100 or 10 Mbps).
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

● 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 *Attribute4* or *Attribute5* to reset the value of the attribute to 0.	Not supported.	Supported.

● 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	Get_Attribute_Single	F6 hex	• Specifying a service for a class: 00 hex • Specifying a service for an instance: Always 01 hex	Not required.
10 hex	Set_Attribute_Single			• Reading a class attribute: 01 or 02 hex • Reading and writing an instance attribute: 01 to 06 hex
4C hex	Get_and_Clear			Specify an attribute to reset its value to 0: 01 to 05 hex

8-3-5 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.

● Class ID

Specify C4 hex.

● Instance ID

You can specify only 01 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 0001 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.	Write	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

● 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.	No supported.
10 hex	Set_Attribute_Single	Writes a value to the specified attribute.	Supported.	No supported.
51 hex	Reset_System_Alarm_All	Resets all errors in the CPU Unit.	Supported.	No supported.

● 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	Get_Attribute_Single	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	Set_Attribute_Single			

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 9-4
9-2-3 Data Reception Processing 9-6
9-2-4 Broadcasting 9-8
9-3 Overview of Built-in EtherNet/IP Port Socket Services 9-9
9-3-1 Overview 9-9
9-3-2 Procedure 9-9
9-4 Settings Required for the Socket Services 9-10
9-4-1 Settings Required for the Socket Services 9-10
9-4-2 Making the Settings with the Sysmac Studio 9-11
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 9-16
9-6-4 UDP Sample Programming 9-18
9-6-5 TCP Sample Programming 9-23
9-7 Precautions in Using Socket Services 9-29
9-7-1 Precautions for UDP and TCP Socket Services 9-29
9-7-2 Precautions for UDP Socket Services 9-29
9-7-3 Precautions for TCP Socket Services 9-29

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

On a UNIX workstation, port numbers 0 to 1023 can be used only by a superuser. In addition, ports 0 to 255 are reserved as well-known ports. Therefore, we recommend that you use port numbers 1024 to 65535 or higher for socket services. 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*
BOOTP client	68	---
HTTP server	---	80
NTP server	123*	---
SNMP Agent	161*	---
SNMP trap	162*	---
FINS/UDP	9600	---
FINS/TCP	---	9600*
Used by system.	2222, 2223, and 2224	9610
CIP messages	---	44818

* These are the default port numbers. You can change the port numbers in the settings.

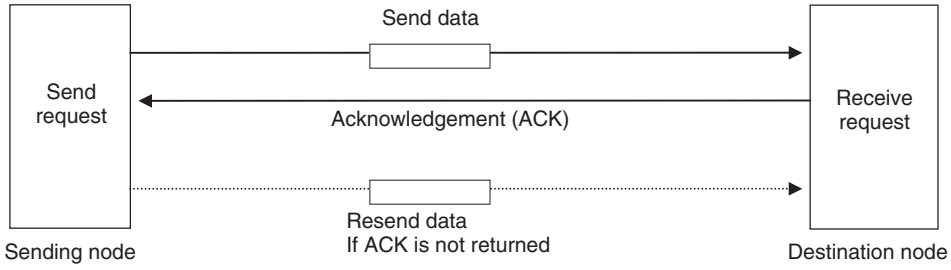
9-2 Basic Knowledge on Protocols

9-2-1 Differences between TCP and UDP

The TCP and UDP functions used on socket services differ as shown below.

TCP Communications

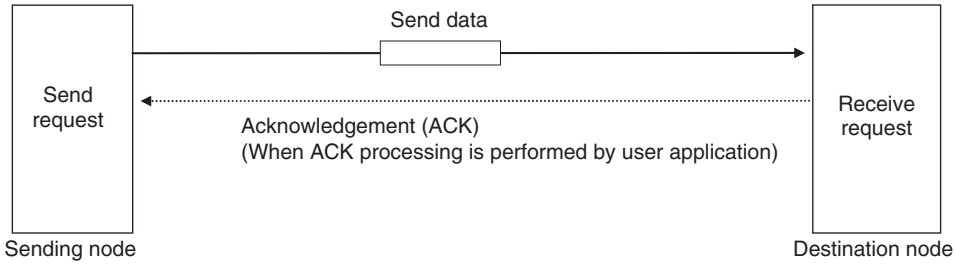
The following operations are performed each time data is sent to ensure that it reaches the destination node. The destination node returns an acknowledgement (ACK) when data is received normally. The sending node sends the next data after ACK is returned. It resends the same data if ACK is not received 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

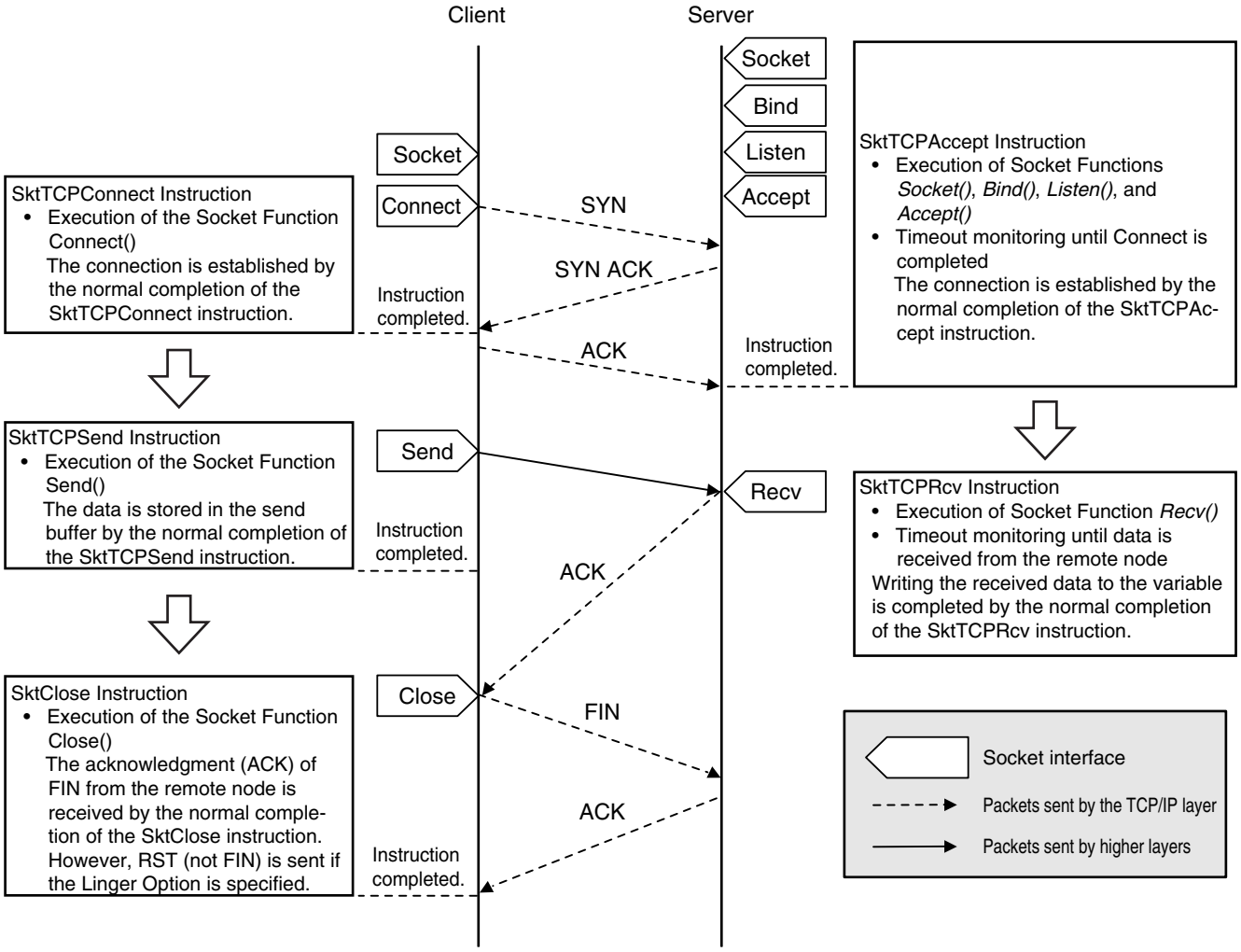
Data is simply sent to the destination node and no acknowledgement or resends are performed like they are for TCP. To increase the reliability of communications, the user application must perform data resend processing.



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.



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 handle. Up to 16 handles can be used at the same time. 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

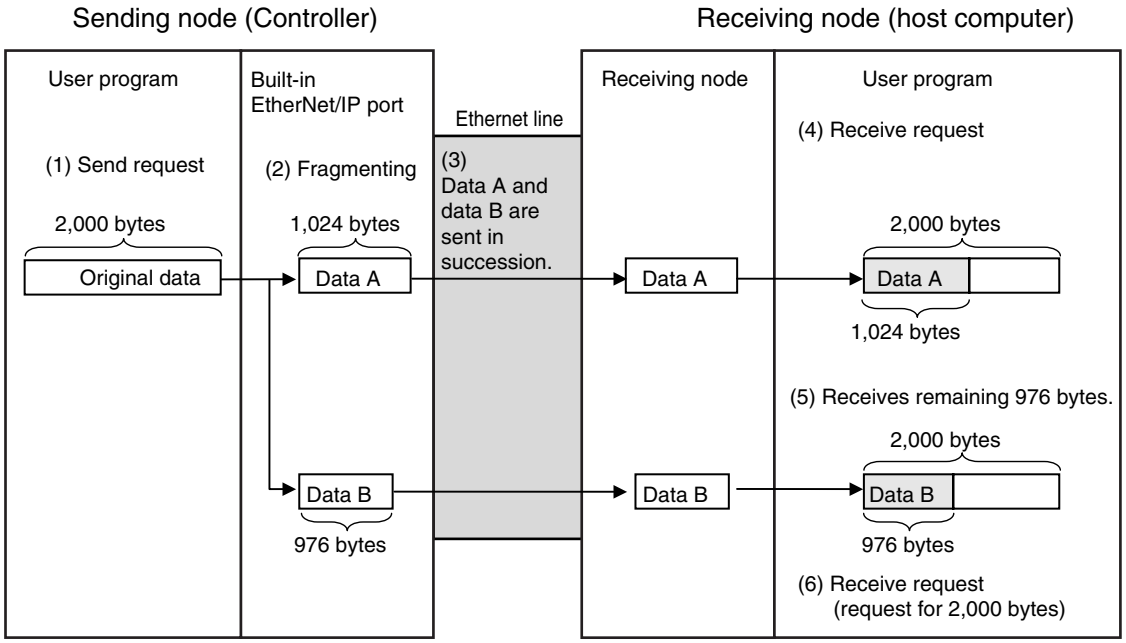
The conditions shown in the following figure occur when data is sent in fragments in TCP communications.

(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

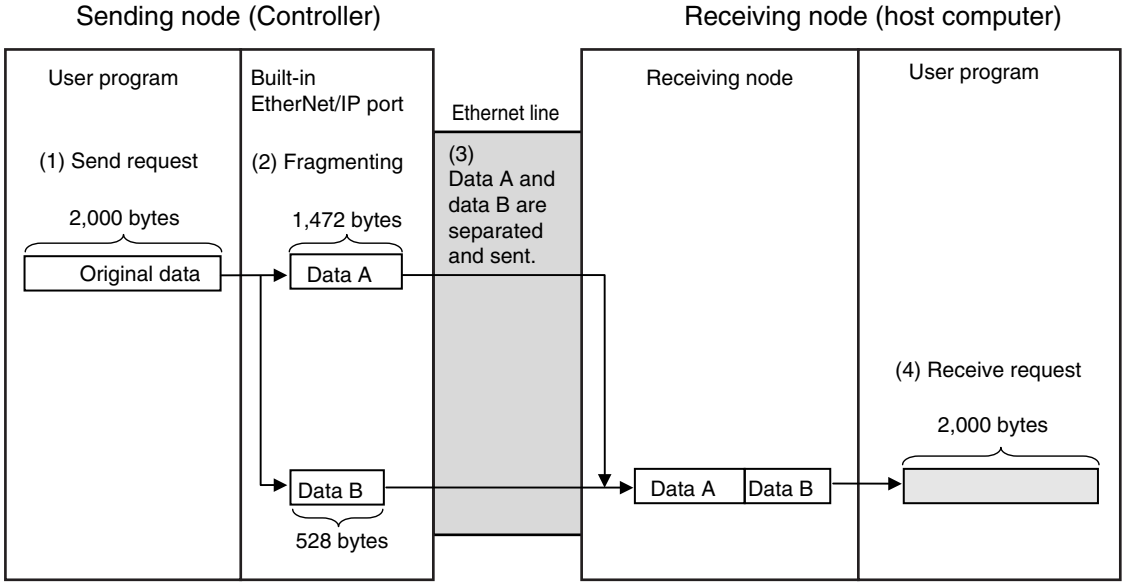
The conditions shown in the following figure occur when data is sent in fragments in UDP communications.

(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.

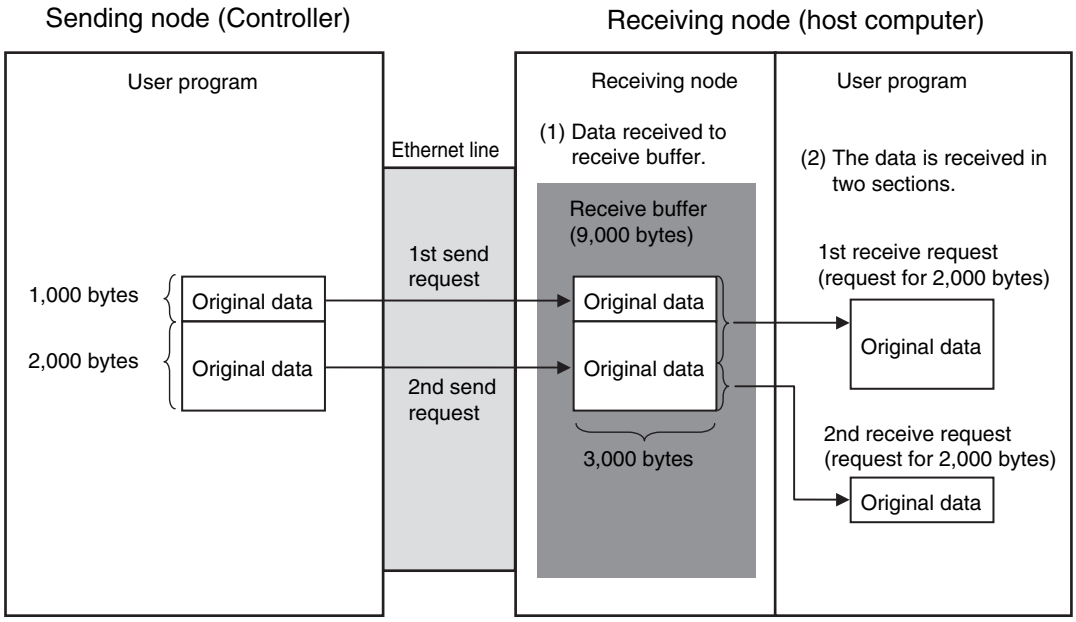
● TCP Receive Processing

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.



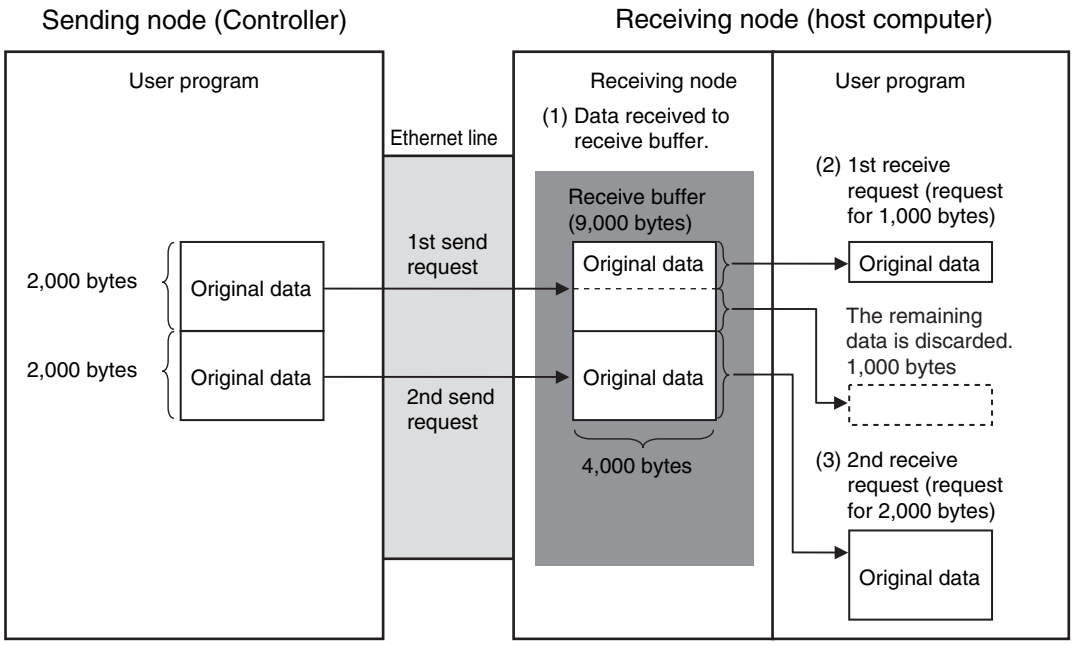
● UDP Receive Processing

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

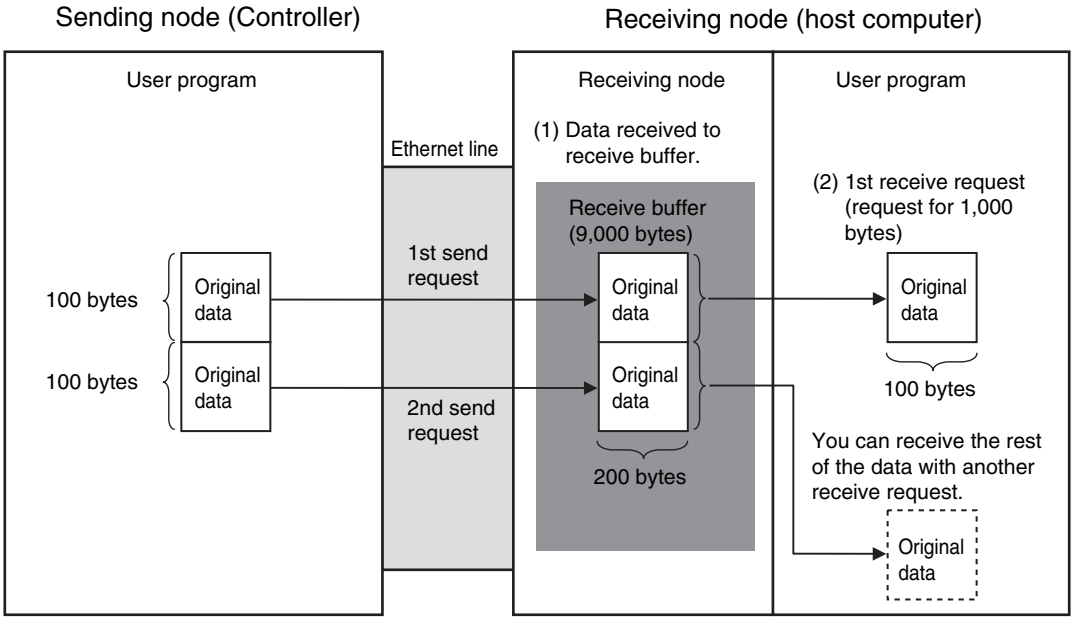
1. The data that was sent from the sending node in two sends is stored in the receive buffer.
2. If a receive request is made for 1,000 bytes for the first send data, the remaining 1,000 bytes is discarded.
3. If the next receive request is for 2,000 bytes, all of the send data for the second send is 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.

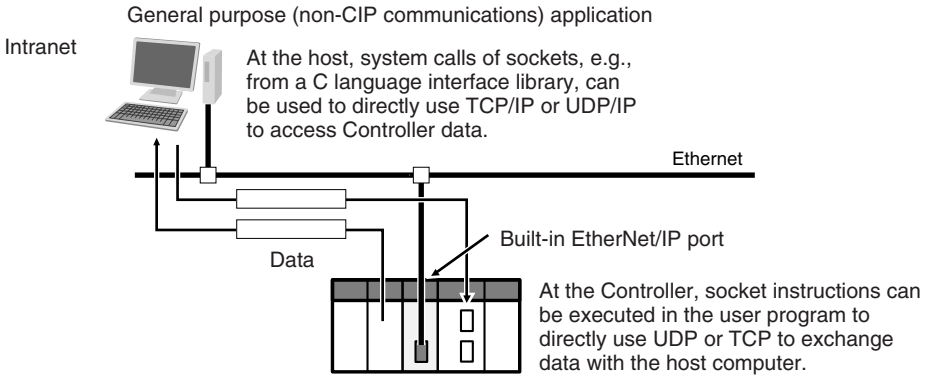
- Local Broadcast
  If no destination IP address is specified, the following IP address is specified automatically.
  Network segment: The network segment of the local IP address is set.
  Host segment: All bits are set to 1.
- Global Broadcast
  Specify this type when the IP address of the local node 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 (ST only) to use the socket services. You can use a total of up to 16 UDP and TCP sockets.

UDP/TCP	Number of sockets
UDP socket services	Total of 16 sockets
TCP socket services	Total of 16 sockets

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

9-4-1 Settings Required for the Socket Services

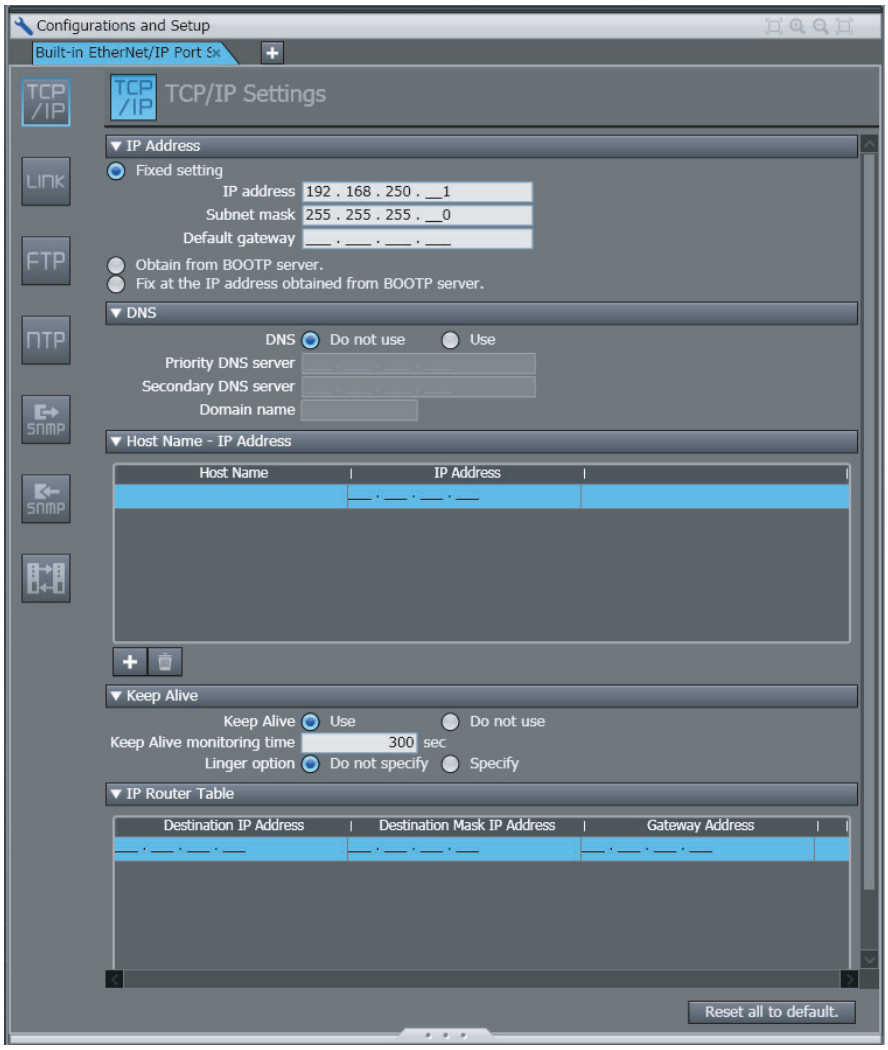
Make the following settings in the Unit Setup to use the socket services.

Sysmac Studio Unit Settings Tab Page	Setting	Setting conditions
Setting	Local IP Address	Required
	Subnet Mask	Required
	IP Router Table	Optional (Required to perform socket communications past an IP router.)
	TCP/IP Keep-Alive	Optional (Change when the default setting of 5 minutes is unacceptable.)
	Linger Option	Optional

9-4-2 Making the Settings with the Sysmac Studio

This section describes the items that are set in the Sysmac Studio to use the socket services.

TCP/IP Display



● IP Address

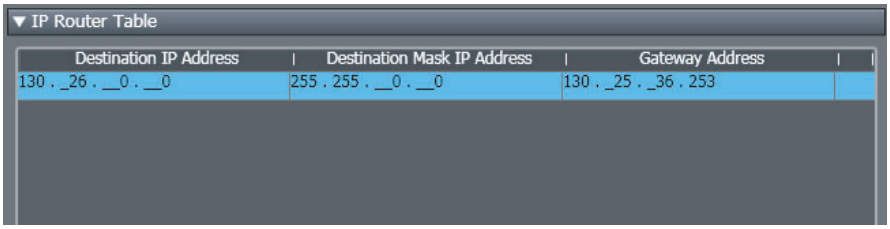
Parameter	Description	Default
IP Address	Set the setting method for the IP address of the built-in EtherNet/IP. • Fixed setting • Obtain from BOOTP server • Fix at the IP address obtained from BOOTP server	Fixed setting
IP address*	Set the IP address for the built-in EtherNet/IP port.	192.168.250.1
Subnet mask*	Set the subnet mask for the built-in EtherNet/IP port.	255.255.255.0
Default gateway	Set the default gateway for the built-in EtherNet/IP port.	

* These settings are required if you select the *Fixed setting* Option.

● Keep Alive

Setting	Description	Default
Keep Alive	Select whether to use the remote node keep alive function of connected servers and clients (such as socket services, FTP server, Sysmac Studio, and FINS/TCP) for each connection number. If *Use* is selected here and the remote node does not respond during the set 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. • Use. • Do not use.	Use.
Keep Alive monitoring time	The connection is disconnected if the keep-alive function is enabled and the remote node does not respond during the monitoring time set in the Keep Alive Monitoring Setting. Setting range: 1 to 65,535 (seconds)	300
Linger option	Set whether to specify the Linger Option for connections to socket services. If the Linger Option is specified, the port number is immediately opened even before the port number is released after the socket closes (approx. 1 minute). • Specify. • Do not specify.	Do not specify.

● IP Router Table Display



Parameter	Description	Default
Destination IP Address	Set these settings when the built-in EtherNet/IP port communicates with nodes on other IP network segments via an IP router. You can set up to eight combinations of IP addresses and gateway addresses. Specify 0 for the host portions of the IP addresses.	None
Destination Mask IP Address		None
Gateway Address		None

9-5 Socket Service Instructions

You can use the following socket service instructions (ST only) for socket services use. Refer to the *NJ-series Instructions Reference Manual* (Cat. No. W502) 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 UDP/TCP Socket instruction
	SktClearBuf	Clear UDP/TCP Socket Receive Buffer 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

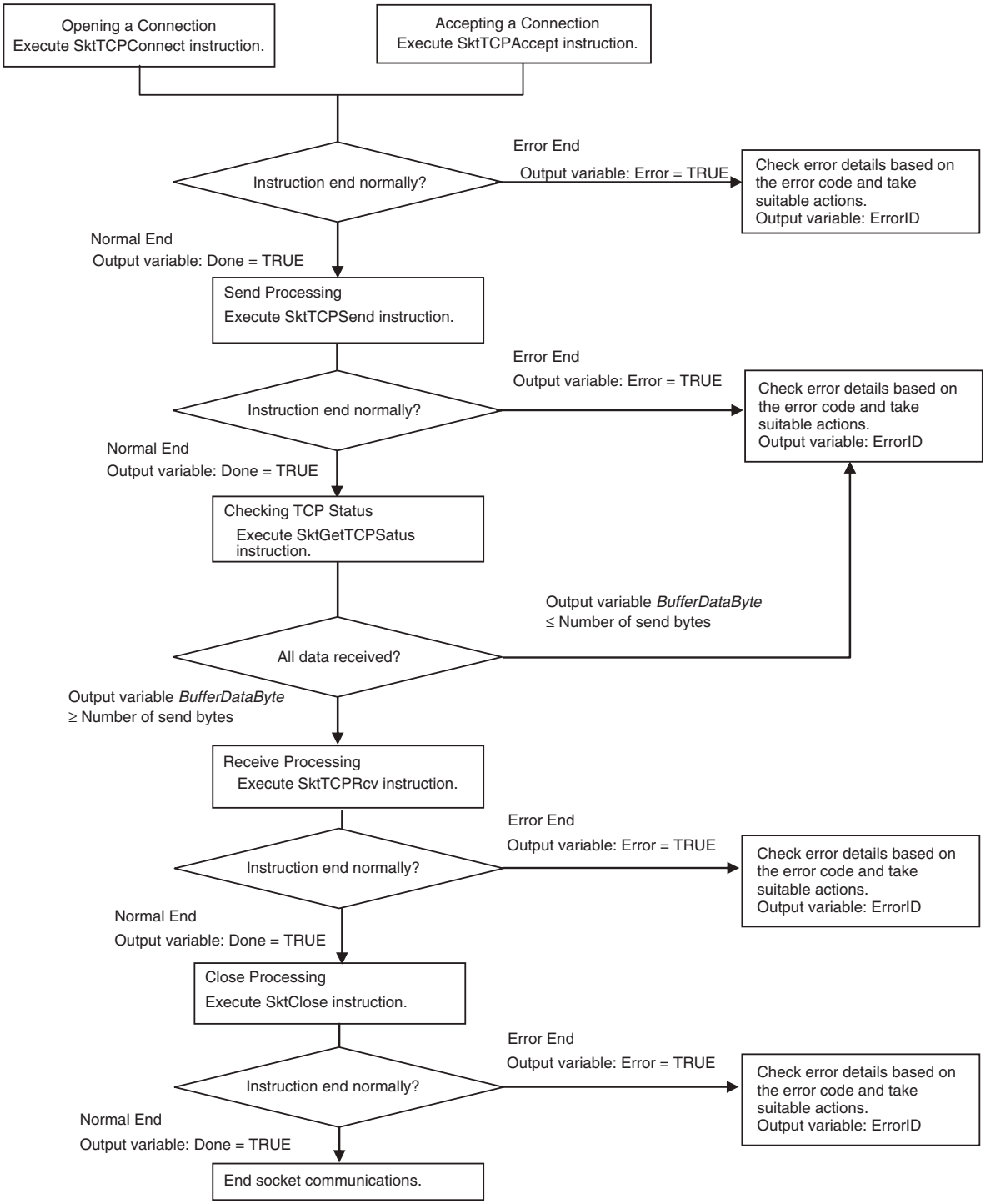
The built-in EtherNet/IP port has 8 sockets for TCP and 8 sockets for UDP. To use these sockets for communications, special ST instructions for sockets are executed to execute the following processes.

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.

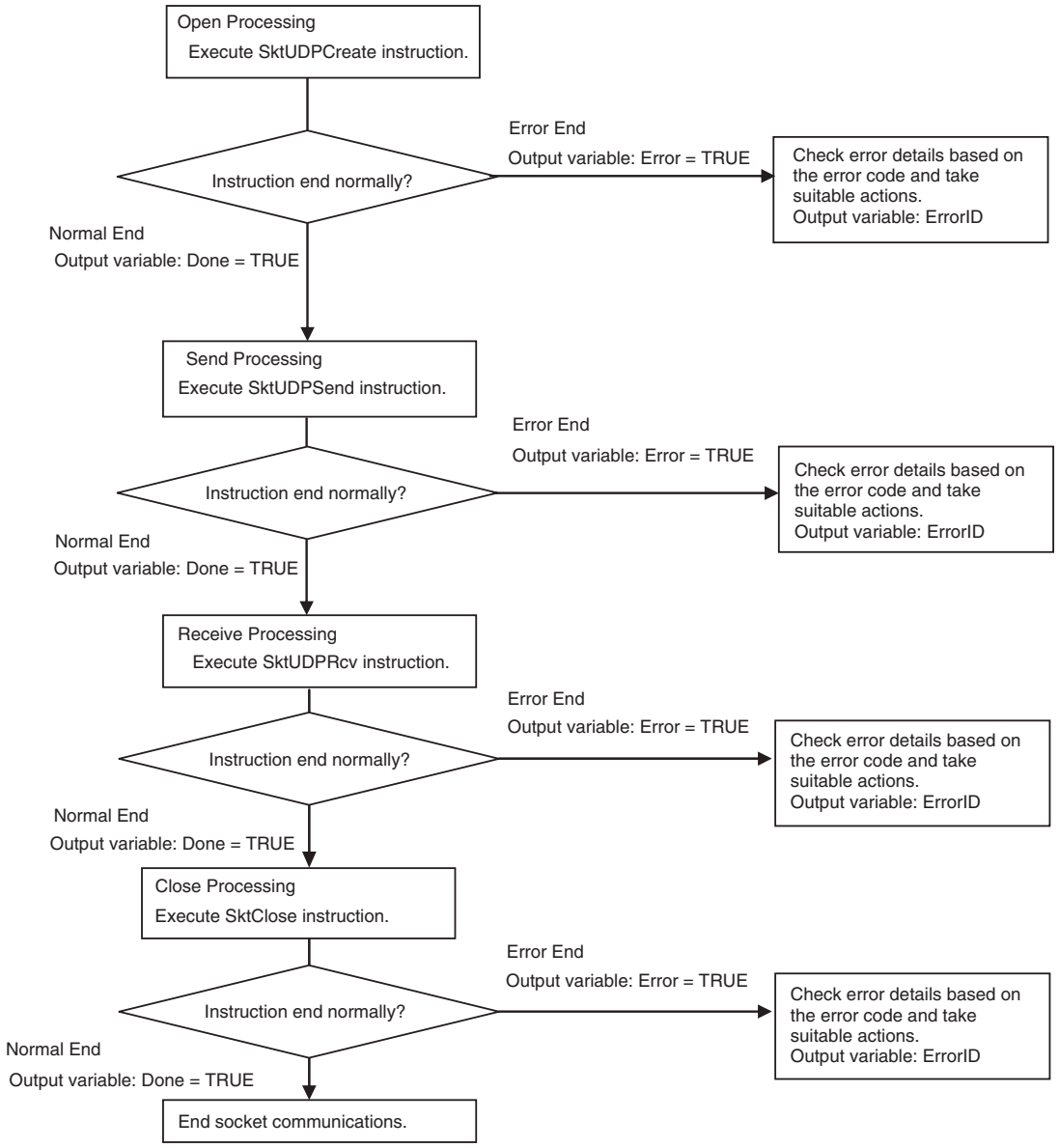
9-6-2 Procedure to Use Socket Services

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



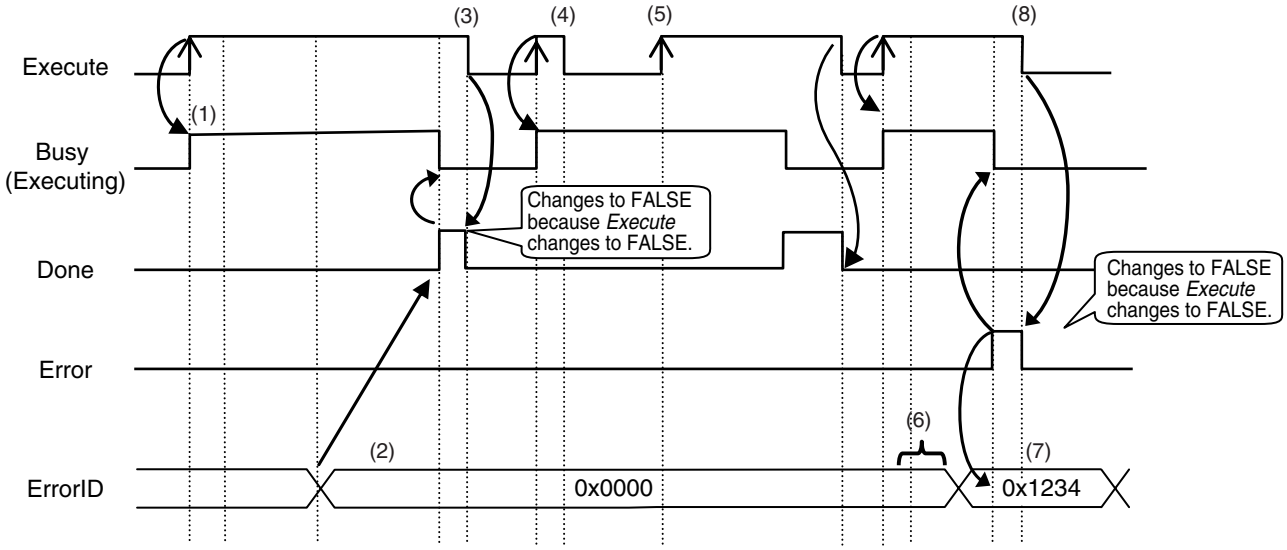
UDP



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.

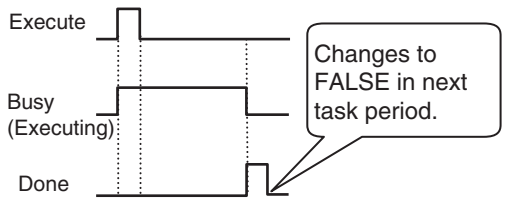
7 When a communications error occurs, *Error* changes to TRUE and the *ErrorID* is stored. Also, *Busy* and *Done* change to FALSE.

8 When *Execute* changes to FALSE, *Error* changes to FALSE.

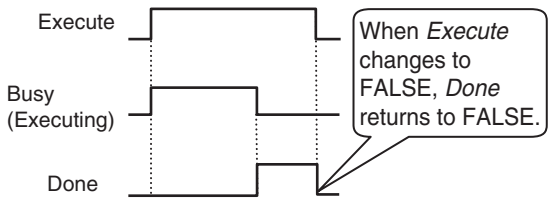
Precautions for Correct Use

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

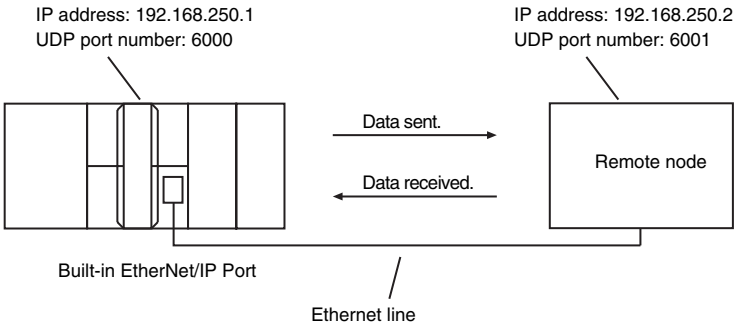


Example 2



9-6-4 UDP Sample Programming

In this sample, the UDP socket service is used for data communications between the NJ-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
	_EIP_EtnOnlineSta	BOOL	☑	Online

```
// 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.IpAd r:='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;
```

```
    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
	SktUDPCreate_instance	SktUDPCreate		
	SktUDPSend_instance	SktUDPSend		
	SktUDPRcv_instance	SktUDPRcv		
	SktClose_instance	SktClose		

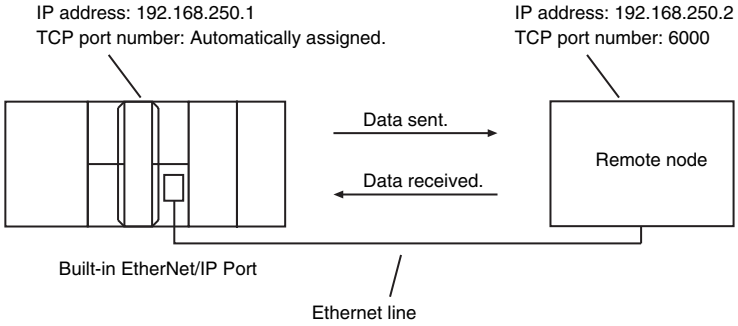
External Variables	Variable	Data type	Constant	Comment
	_EIP_EtnOnlineSta	BOOL	☑	Online

```
// 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#6001,                  // 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 :                         // 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,
            Socke       :=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 NJ-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	BOOL	☑	Online

```
// 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 UDP 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;
```

```
3 :                         // Request reading status.
    SktGetTCPStatus_instance(
        Execute     :=TRUE,
        Socket      :=WkSocket);                    // Socket

    IF (SktGetTCPStatus_instance.Done=TRUE) THEN
        Stage       :=INT#4;                        // Normal end
    ELSIF (SktGetTCPStatus_instance.Error=TRUE) THEN
        Stage       :=INT#30;                       // Error end
    END_IF;

4 :                         // Send request
    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#5;                        // Normal end
    ELSIF (SktTCPSend_instance.Error=TRUE) THEN
        Stage       :=INT#40;                       // Error end
    END_IF;

5 :                         // Request receiving data.
    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#6;                        // Normal end
    ELSIF (SktTCPRcv_instance.Error=TRUE) THEN
        Stage       :=INT#50;                       // Error end
    END_IF;

6 :                         // 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;
```

● 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	BOOL	☑	Online

```
// 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 receiving data.
        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;
```

```
    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 SktTCPConnect 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.

10

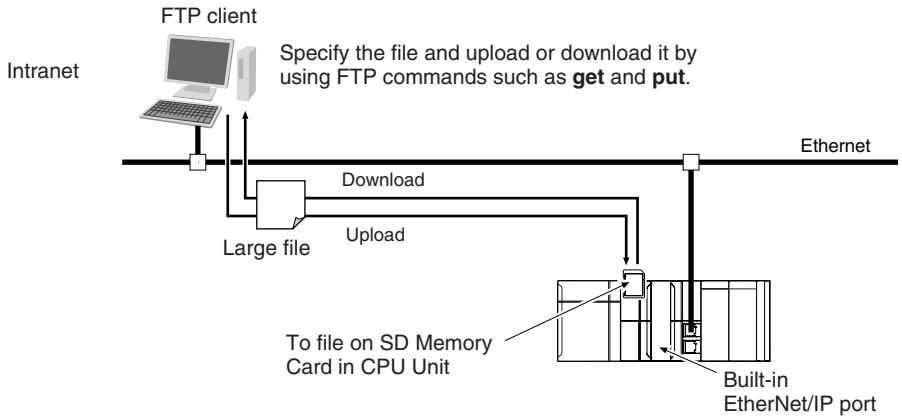
FTP Server

10-1 Overview and Specifications 10-2
10-1-1 Overview 10-2
10-1-2 Specifications 10-2
10-2 FTP Server Function Details 10-3
10-2-1 Supported Files 10-3
10-2-2 Connecting to the FTP Server 10-3
10-3 Using the FTP Server Function 10-5
10-3-1 Procedure 10-5
10-3-2 List of Settings Required for the FTP Server Function 10-5
10-3-3 FTP Settings Display 10-6
10-4 FTP Server Application Example 10-7
10-5 Using FTP Commands 10-9
10-5-1 Table of Commands 10-9
10-5-2 Using the Commands 10-10
10-6 Using SD Memory Card Operations 10-16
10-6-1 SD Memory Card Types 10-16
10-6-2 File Types 10-17
10-6-3 Initializing SD Memory Cards 10-17
10-6-4 Format of Variable Data 10-17
10-7 Application Example from a Host Computer 10-19

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 SD Memory Card.



10-1-2 Specifications

Item	Specification
Executable commands	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 currently specified file data type and appends a local file to the remote host. close: Disconnects the FTP server. bye: Closes the FTP client. quit: Closes the FTP client.
Protection	Login name (up to 12 characters) Password consists of 8 to 32 characters
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 SD Memory Card mounted in the CPU Unit. The directory tree is shown below.

/: *root*

- MEMCARD1: SD
- usr, bin, dev — Used by system

A connection is initially made to the root directory.

Additional Information

- The date of the MEMCARD1 directory displayed for **ls, dir,** or **mdir** 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: CPU Unit model (example: NJ501-1300)
Connected to maximum number of connections (6)	530 FTP server busy, Goodbye.

Setting Restrictions

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 CPU Unit 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 SD Memory Card

<table>
<tr><th colspan="2" rowspan="2">Later access \ First access</th><th colspan="2">Instruction</th><th colspan="2">Communications command</th></tr>
<tr><th>Read</th><th>Write</th><th>Read</th><th>Write</th></tr>
<tr><td rowspan="2">Instruction</td><td>Read</td><td colspan="2">Exclusive control is automatically implemented and the read instruction results in an error (first-come-first-serve).</td><td>(Exclusive control is not required.)</td><td rowspan="3">Implement exclusive control.</td></tr>
<tr><td>Write</td><td colspan="2">Exclusive control is automatically implemented and the write instruction results in an error (first come first serve)</td><td>Implement exclusive control.</td></tr>
<tr><td rowspan="2">Communications command</td><td>Read</td><td>(Exclusive control is not required.)</td><td>Implement exclusive control.</td><td>(Exclusive control is not required.)</td></tr>
<tr><td>Write</td><td colspan="3">Implement exclusive control.</td></tr>
</table>

10-3 Using the FTP Server Function

10-3-1 Procedure

1. Make the basic settings.

   Refer to *1-5-1 Outline* 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 CPU Unit online and transfer the settings to the Controller.
5. Insert the SD Memory Card into the CPU Unit.
6. Connect to the built-in EtherNet/IP port from an FTP client.
7. 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).
8. Look for the target file on the SD Memory Card that you inserted into the Controller in the MEMCARD1 directory.
9. Download the files.
10. Close the connection.

10-3-2 List of Settings Required for the FTP Server Function

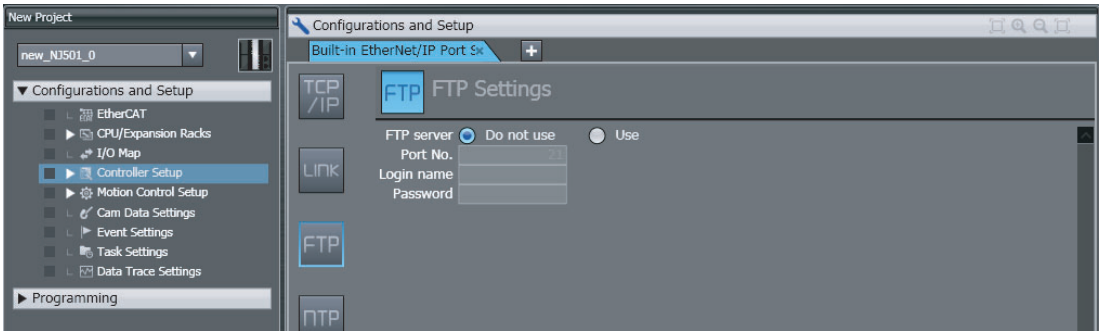
Make the following settings for the unit setup when the FTP server function is used.

Built-in EtherNet/IP Port Settings Tab Page on Sysmac Studio	Setting	Setting conditions	Reference
FTP	FTP server	Required	page 10-6
	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.

10-3-3 FTP Settings Display

The built-in EtherNet/IP port settings that are set when the FTP server is used are shown in the Edit Parameters Pane on the Sysmac Studio.



● Setting

Setting	Description	Default
FTP server	Specify whether to use the FTP server. FTP connections from external devices will not be possible if the *Do not use* Option is selected.	Do not use
Port No.*1	Set the FTP port number of the built-in EtherNet/IP port. It is normally not necessary to change this setting. FTP uses two ports: a port for control and a port for data transfer. • Control Port Set the FTP port number. • Data Transfer Port The data transfer port number is always 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.

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.

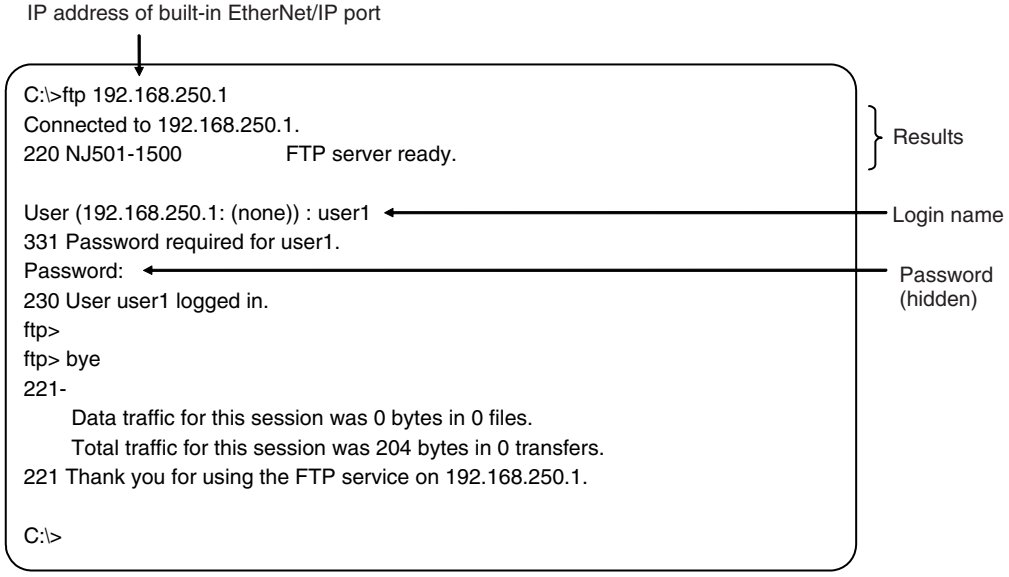
Additional Information

When a general-purpose FTP application is used, you can use a graphical user interface similar to Explorer to transfer and read files.

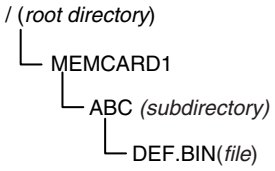
- **Step**

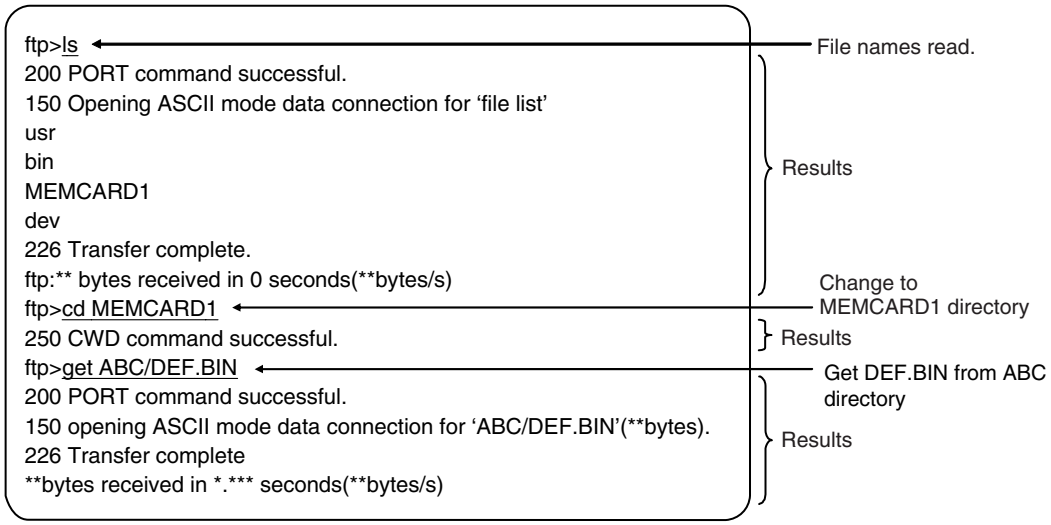
1 Make sure that an SD Memory Card is inserted in the CPU Unit and turn ON the power supply to the Controller.

2 Connect to the FTP server from a computer on the Ethernet by entering the text that is underlined in the following diagram.



3 Enter FTP commands (underlined in the following diagram) to read and write files. The following directory tree is used in this example.





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 SD Memory Card in the CPU Unit.

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 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 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 SD Memory Card).
- Specify the directory or file name on the 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 SD Memory Card).
- It displays the same information as command [ls -l].
- Specify a file on the SD Memory Card for the remote file name.
- If a local file name is specified, the file information is stored in the specified file.

mdir

Format

mdir *remote_file_name local_file_name*

Function

- Displays the names of files, subdirectories, creation dates, and sizes on the remote host (on the SD Memory Card).
- Specify the directory or file name on the 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.

Format

mdir RemoteFiles [...] LocalFile

Remotefiles: Enter a hyphen (-) to use the current working directory.

Localfiles: Enter a hyphen (-) to display a list.

Function Details

- A list of the files and subdirectories in the remote directory is displayed.

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 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 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 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 SD Memory Card).

cd

Format

cd [*directory_name*]

Function

- Changes the remote host work directory to the specified remote directory.
- Files on the 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 an SD Memory Card is not inserted in the CPU Unit or if the SD Memory Card power indicator on the CPU Unit is not lit.

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 CPU Unit 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 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 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 SD Memory Card).
- You can specify the destination file name to specify the name the file is stored under on the SD Memory Card.
- Any existing file with the same name in the remote host (on the 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 SD Memory Card).
- Any existing file with the same name in the remote host (on the 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 SD Memory Card).

mdelete

Format

mdelete *file_name*

Function

- You can include wildcards (*) in the file name to delete multiple remote files from the 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

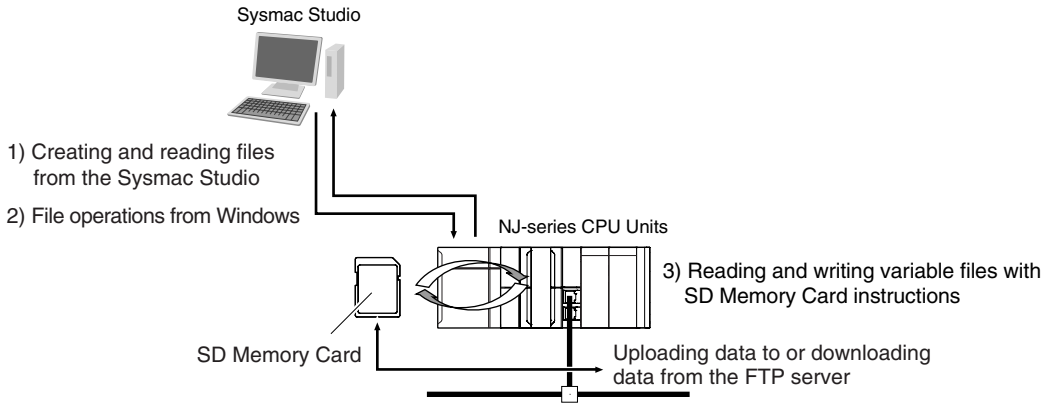
- Ends the FTP sessions.

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 SD Memory Card and the FTP server.

- Variables files (binary format)

The following three methods are used by the CPU Unit to store and read data on the SD Memory Card.



10-6-1 SD Memory Card Types

Use the SD Memory Card that is specified below. Operation may not be dependable if another SD Memory Card is used.

Item	Specification
Model number	HMC-SD291
Capacity	2 GB
Format	FAT16
Number of overwrites	100,000 writes

10-6-2 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 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 CPU Unit

The files described in the following table can be read or written by the CPU Unit.

File type	File names	Extension	Contents	Description
Variables file (binary format)	Refer to *10-6-2 File Types.*	.bin	Specified variables	This variables file contains the values of specified variables (which include arrays and structures) in binary format (.bin).

Refer to the *NJ-series CPU Unit Software User's Manual* (Cat. No. W501) for details.

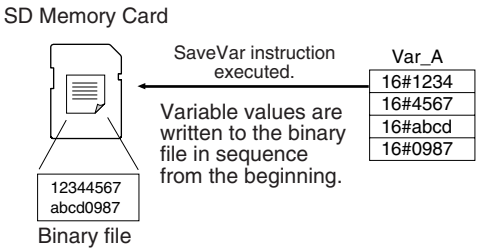
10-6-3 Initializing SD Memory Cards

1. Insert the SD Memory Card into the CPU Unit.
2. Use the Sysmac Studio to initialize the SD Memory Card.

10-6-4 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 CPU Unit. 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.



Additional Information

- When you handle a binary file on the NJ-series CPU Unit, 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 *NJ-series Instructions Reference Manual* (Cat. No. W502).

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 SD Memory Card in the CPU Unit 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 SD Memory Card in the CPU Unit is transferred to a workstation, and then a manufacturing instructions file (PLAN.BIN) on the workstation is transferred to the SD Memory Card in the CPU Unit. 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                                                    <-- FTP started.
connected to nj
220 **IPaddress** NJ501-1300 FTP server(FTP**version**)ready
Name(nj:root): ■
```

2 Enter the login name.

```
Name(nj:root):LogIn                                         <-- Enter the login name.
331 Password required for LogIn.
Password:                                                   <-- Enter the password.
230 LogIn logged in.
ftp> ■
```

3 Make sure the Memory Card is correctly inserted. The MEMCARD1 directory is displayed if there is an SD Memory Card in the CPU Unit.

```
ftp> ls                                                     <-- Make sure the Memory
200 PORT command successful.                                   Card is inserted.
150 opening data connection for ls(**IPaddress**port#**)(0 bytes).
MEMCARD1
226 Transfer complete.
15 bytes received in 0 seconds(**bytes/s)
ftp> ■
```

4 Change to the MEMCARD1 directory.

```
ftp> cd MEMCARD1                                            <-- Change the directory.
250 CWD command successful.
ftp> ■
```

5 Change data type to binary.

```
ftp> type binary                                            <-- Set binary data type.
200 Type set to I.
ftp> ■
```

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 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

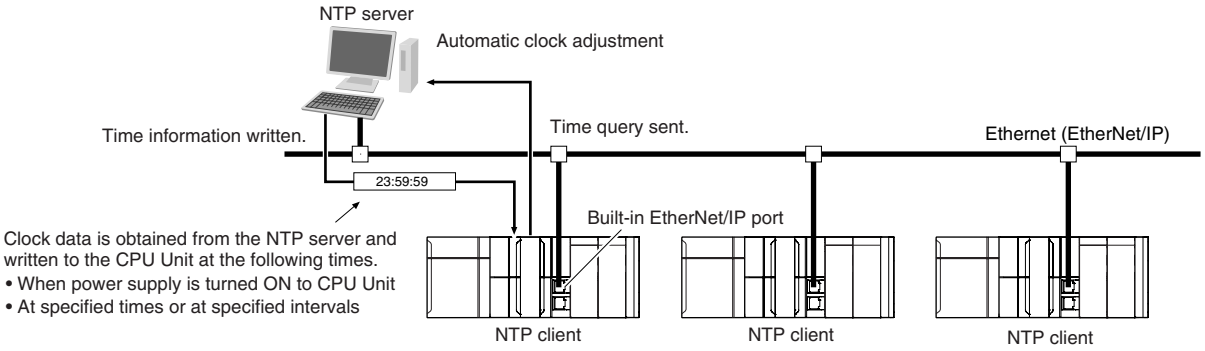
Automatic Clock Adjustment

11-1 Automatic Clock Adjustment 11-2
11-1-1 Overview 11-2
11-1-2 Specifications 11-3
11-2 Procedure to Use the Automatic Clock Adjustment Function 11-4
11-2-1 Procedure 11-4
11-2-2 Settings Required for Automatic Clock Adjustment 11-4
11-2-3 NTP Display 11-5
11-3 Troubleshooting Automatic Clock Adjustment 11-7

11-1 Automatic Clock Adjustment

11-1-1 Overview

With the built-in EtherNet/IP port, clock information is read from the NTP* server at the specified time or at a specified interval after the power supply to the CPU Unit is turned ON. The internal clock time in the CPU Unit is updated with the read time.

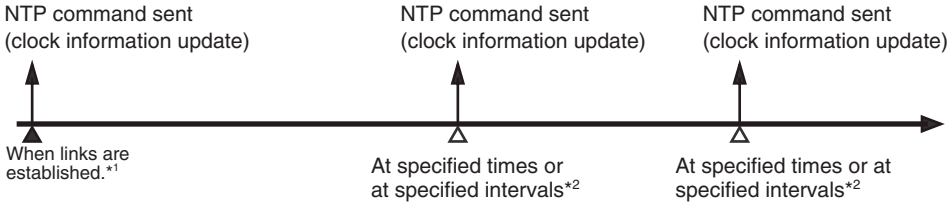


* The NTP (Network Time Protocol) server is used to control the time on the LAN.

11-1-2 Specifications

Item	Specification	
Protocol	NTP	
Port No.	123 (UDP) However, you can change the port number in the Built-in EtherNet/IP Port Settings on the Sysmac Studio.	
Access to NTP server	Writes the clock information from the NTP server to the local CPU Unit.	Obtains the clock information from the NTP server set up on the Network, and applies the information obtained to the local CPU Unit.
NTP Operation Timing	Clock information is automatically updated at the following times if the NTP function is used. • After links are established when the power supply to the CPU Unit is turned ON • At specified times or at specified intervals (according to the option selected for the NTP Operation Timing)	

Clock information is updated at the following times.



*1 The clock information is updated if acquiring the NTP server clock information is enabled on the NTP Tab Page.

*2 According to the option that is selected for the NTP operation timing on the NTP Tab Page.

Additional Information

NTP clock synchronization is normally performed as follows:

- If the clock deviation is within 128 ms: The clock is synchronized every 0.5 ms.
- If the clock deviation exceeds 128 ms: The clock is synchronized immediately.

11-2 Procedure to Use the Automatic Clock Adjustment Function

11-2-1 Procedure

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 NTP Settings Display.
   - NTP server settings (required)
   - NTP operation timing

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 CPU Unit.

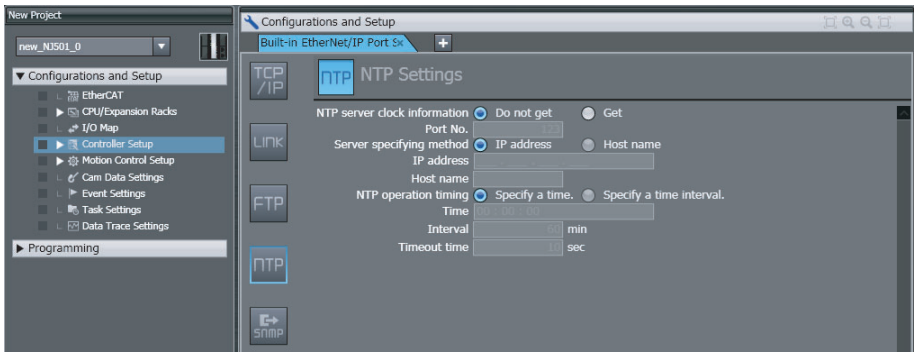
11-2-2 Settings Required for Automatic Clock Adjustment

The following Built-in EtherNet/IP Port Settings are made from the Sysmac Studio to use automatic clock adjustment.

Tab page	Setting		Setting conditions	Reference
NTP	NTP server clock information		Required.	*11-2-3 NTP Display*
	Port No.		Specified by user. **Note** Required to change from the default value of 123.	
	Server specifying method		Required*1	
		IP address	One of these must be set, depending on the *Server specification type* setting.	
		Host name		
	NTP operation timing		Required*1	
		Specify a Time	One of these must be set. (Set according to the option selected for the NTP Operation Timing.)	
		Specify a time interval		
		Timeout time	Specified by user. **Note** Required to change from the default value of 10 seconds.	

11-2-3 NTP Display

The settings to use automatic clock adjustment are displayed in the Built-in EtherNet/IP Port Settings Pane on the Sysmac Studio.



Setting		Description	Default
NTP server clock information		Specify whether to get the clock information from the NTP server and use it to update the clock time in the CPU Unit.	Do not get
Port No.*1*2		Set the port number to use to connect to the NTP server for automatic clock adjustment. It is normally not necessary to change this setting.	123
Server specifying method*1		Set the method to use to specify the NTP server from which to get the clock information. • IP address • Host name	IP address
	IP address	Set the IP address of the NTP server. (Set this setting if an IP address is set for the server specification type.)	None
	Host name	Set the host name of the NTP server (i.e., the domain name of the host). (Set this setting if a host name is set for the specification type.) (You can use up to 200 single-byte alphanumeric characters, dots, and hyphens with up to 63 single-byte characters between dots.)	None

Setting		Description	Default
NTP operation timing*1		Set the time at which the NTP server is accessed to synchronize the clocks. • Specify a clock time • Specify a time interval	Specify a clock time
	Time [hours:minutes:seconds]	The NTP server is accessed at the specified time. (Setting range: 00:00:00 to 23:59:59) (Set this setting if the NTP operation timing is set to the *Specify a clock time* Option.)	00:00:00
	Interval [minutes]	The NTP server is accessed when the specified period of time has passed. (Setting range: 1 to 1,440 minutes) (Set this setting if the NTP operation timing is set to the *Specify a time interval* Option.)	60 minutes
	Timeout time [seconds]*1	Set the timeout detection time. (Setting range: 1 to 255 seconds) When there is no response from the remote host, a request is resent up to 4 times during the interval set here. If a specified interval is specified for the NTP operation, timing to the next operation starts after the 4th request times out.	10 seconds

*1 This setting is required to get the clock information from the NTP server.

*2 You cannot set the following ports because they are used by the system: 25, 68, 110, 2222, 2223, 2224, and 9600.

11-3 Troubleshooting Automatic Clock Adjustment

The following table shows the main causes and solutions of errors that occur for automatic clock adjustment (NTP). When one of the following errors occurs, information such as the time of occurrence, the event level, and the event source are recorded in the event log.

Event code	Event name	Cause	Level	Source details	Correction
34210000 hex	Basic Ethernet Setting Error	• Power was interrupted when a download was in progress for the basic Ethernet settings. • Memory error	Partial fault	Communications port	Cycle the power supply to the Controller or reset the Controller. If normal operation is restored by cycling the power supply to the Controller or resetting the Controller, a memory error has occurred and may occur again. Replace the CPU Unit. If operation is not restored even after you cycle the power supply or reset the Controller, perform either of the following operations and download the settings again. • Perform a Clear Memory All operation. • Identify the cause of error from those listed in the event log details and correct the settings. If the above operations do not correct the problem, replace the CPU Unit.
34220000 hex	TCP/IP Basic Setting Error (Local Port IP Address)	• Power was interrupted when a download was in progress for the basic TCP/IP settings. • Memory error • The IP address acquired from BOOTP server is illegal. • Incorrect DNS server IP address	Partial fault	Communications port	Cycle the power supply to the Controller or reset the Controller. If normal operation is restored by cycling the power supply to the Controller or resetting the Controller, a memory error has occurred and may occur again. Replace the CPU Unit. If operation is not restored even after you cycle the power supply or reset the Controller, perform either of the following operations and download the settings again. • Perform a Clear Memory All operation. • Identify the cause of error from those listed in the event log details and correct the settings. If the above operations do not correct the problem, replace the CPU Unit. Correct the IP address that was provided to this port by the BOOTP server so that it is within the range specified for an NJ-series Controller.

Event code	Event name	Cause	Level	Source details	Correction
34230000 hex	TCP/IP Advanced Setting Error (IP Router Table)	• Power was interrupted when a download was in progress for the advanced TCP/IP settings. • Memory error • There is a mistake in the IP router table settings.	Minor fault	Communications port	Cycle the power supply to the Controller or reset the Controller. If normal operation is restored by cycling the power supply to the Controller or resetting the Controller, a memory error has occurred and may occur again. Replace the CPU Unit. If operation is not restored even after you cycle the power supply or reset the Controller, perform either of the following operations and download the settings again. • Perform a Clear Memory All operation. • Identify the cause of error from those listed in the event log details and correct the settings. If the above operations do not correct the problem, replace the CPU Unit. Correct the IP router table settings.
34250000 hex	NTP Client Setting Error	• Power was interrupted while downloading the NTP Client settings. • Memory error	Minor fault	NTP	Cycle the power supply to the Controller or reset the Controller. If normal operation is restored by cycling the power supply to the Controller or resetting the Controller, a memory error has occurred and may occur again. Replace the CPU Unit. If operation is not restored even after you cycle the power supply or reset the Controller, perform either of the following operations and download the settings again. • Perform a Clear Memory All operation. • Identify the cause of error from those listed in the event log details and correct the settings. If the above operations do not correct the problem, replace the CPU Unit.
84040000 hex	NTP Server Connection Error	• The NTP server settings are incorrect. • The NTP server is down. • A communications path error occurred.	Minor fault	NTP	Take either of the following measures. • Correct the NTP server settings. • Check to make sure that the server at the remote connection is operating normally. • Check the communications path (EtherNet/IP Unit or built-in EtherNet/IP ports, cable connections, Ethernet switches, routers, and servers) and correct the situation that is causing the error.
84030000 hex	DNS Server Connection Error	• Parameter error • The server is down. • An error occurred in the communications path.	Minor fault	Communications port	If there is a mistake with the specifications of the connected server, correct the server specifications and download them again. Check if the server at the remote connection is operating normally and set it to operate normally if it is not. Check the communications path to the server and take corrective measures if there are any problems.

Event code	Event name	Cause	Level	Source details	Correction
940B0000 hex	NTP Client Started	• The NTP client started normally and a request for the NTP server to obtain the time started.	Information	Communications port	---

* For details on other error log information, refer to *Section 14 Troubleshooting*.

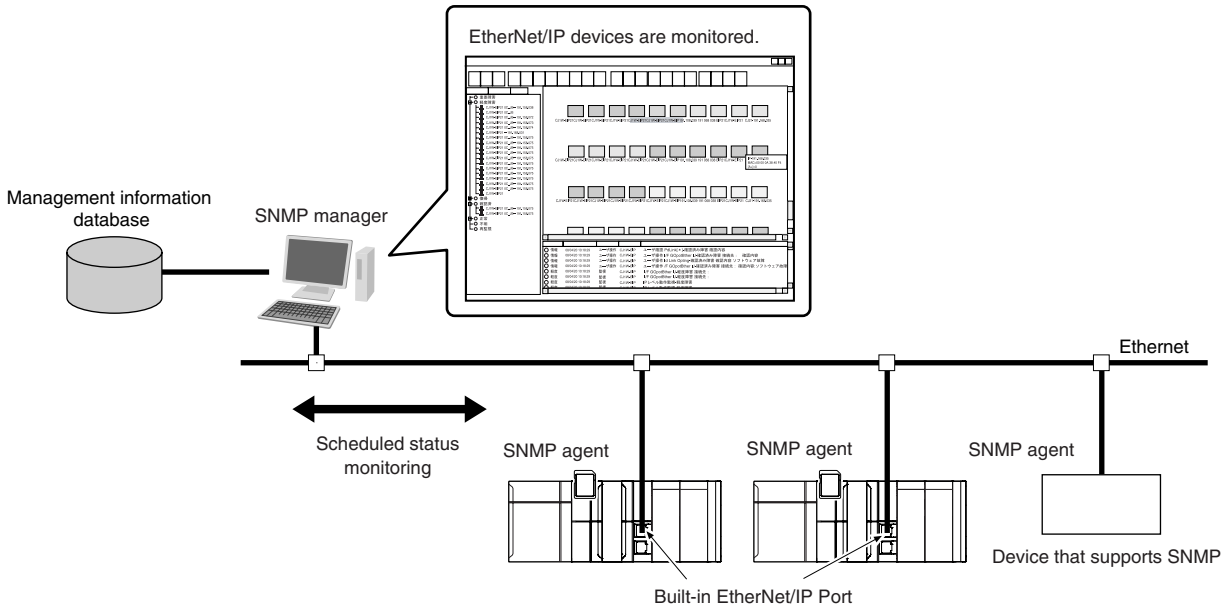
12

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 12-20
12-2-1 Procedures 12-20
12-2-2 Settings Required for the SNMP Agent 12-20
12-2-3 SNMP Settings Display 12-22
12-2-4 SNMP Trap Settings Display 12-24
12-3 Troubleshooting SNMP Agent 12-26
12-3-1 SNMP Agent Errors 12-26
12-3-2 Event Log Error Codes for the SNMP Agent 12-26

12-1 SNMP Agent

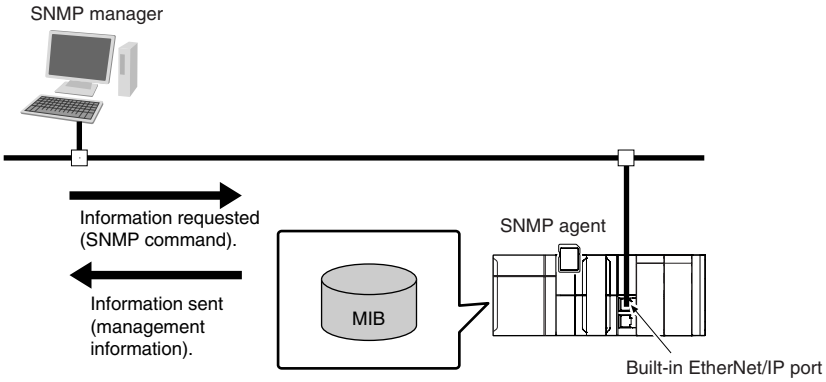
The SNMP (simple network management protocol) is a network management protocol. You can use the SNMP to manage any network that consists of devices that support SNMP. The server that manages the network is called the SNMP manager. The managed network devices are called SNMP agents.



12-1-1 Overview

SNMP Agent

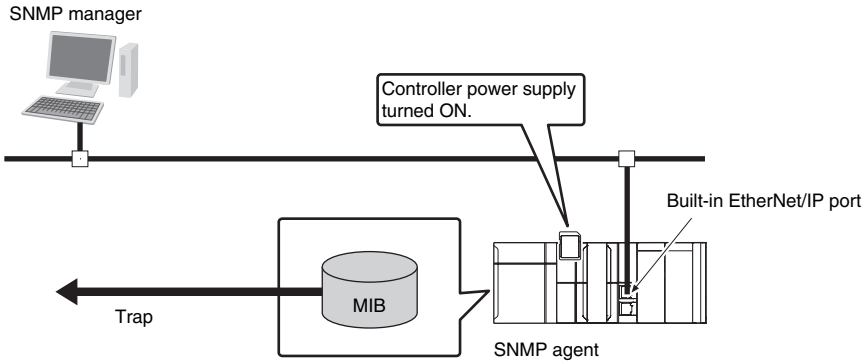
The built-in EtherNet/IP port has its own management information called the MIB (management information base). This information can be provided to the SNMP manager. The SNMP manager is software that gathers and processes information about devices on the SNMP network and provides that information to the 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.

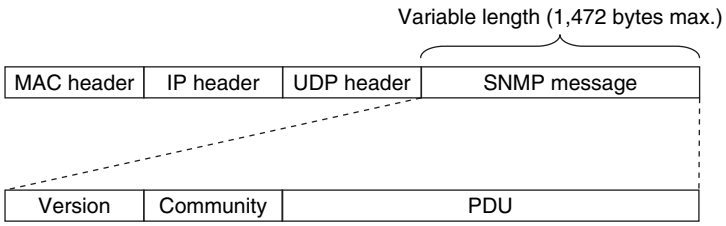


12-1-2 Specifications

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. • 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:



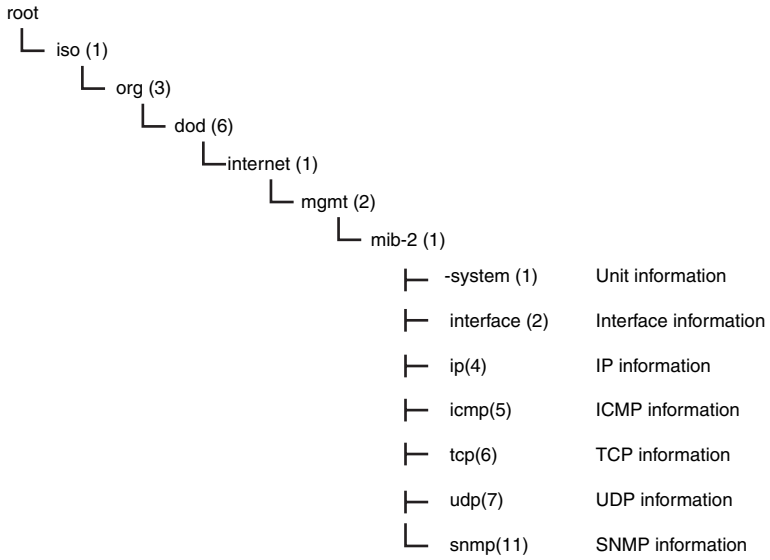
Item	Set value
Version	This value gives the SNMP version. SNMPv1: 0 SNMpv2c: 1
Community	Community name for verification
PDU	This depends on the PDU type.

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	*system* group		The MIB for information related to the device.
	interfaces group		The MIB for information related to the interface.
	ip 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.
	icmp group		The MIB for ICMP information.
	tcp group	tcp	The MIB for TCP information.
	udp group	udp	The MIB for UDP information.
	snmp group	snmp	The MIB for SNMP information.

Detailed Descriptions of MIB Objects

● *System* Group

Subtree name	Standard [(*identifier*) *attribute*]	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 • CPU Unit model (example): NJ501-1300 • CPU Unit version (example): Version 1.0
sysObjectID	(2) RO Vendor OID. Tells where this device information was assigned in the private MIB.	Supported.	1.3.6.1.4.1.16838.1.1025.4
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 [(*identifier*) *attribute*]	Support	Implementation specifications
ifNumber			(1) RO The number of network interfaces.	Supported.	1
	ifTable		(2) NA Interface entity table	---	
		ifEntry	(1) NA Row data for interface information The index is *ifIndex*.	---	
		ifIndex	(1) RO A number used to identify the interface.	Supported.	1
		ifDescr	(2) RO Information related to the interface (includes manufacturer name, product name, and hardware interface version).	Supported.	10/100M Fast Ethernet 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)

Subtree name	Standard [(*identifier*) *attribute*]	Support	Implementation specifications
ifMtu	(4) RO MTU value The maximum size (in octets) of datagrams that can be sent and received through this interface.	Supported.	1500
ifSpeed	(5) RO Estimated bandwidth If a stable, accurate value cannot be obtained for the bandwidth, a nominal value is set instead.	Supported.	10000000
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.
ifLastChange	(9) RO The *sysUpTime* (in 0.01seconds) at the last change in *ifOperStatus* 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.

Subtree name			Standard [(*identifier*) *attribute*]	Support	Implementation specifications
		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.
		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 [(*identifier*) *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 *badValue* error is returned. Forwarding (1) Not-forwarding (2)	Supported.	Not-forwarding (2)
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 *ipForwDatagrams*.	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 [(*identifier*) *attribute*]	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 *ipForwDatagrams*, 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.	---	---

Subtree name			Standard [(*identifier*) *attribute*]	Support	Implementation specifications
	ipAddrEntry		(1) NA Row data of address information for IP addresses. The index is *ipAdEntAddr*.	---	---
		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 *ifIndex*.	Supported.	According to the standard.
		ipAdEntNetMask	(3) RO The subnet mask for the IP address of this entry.	Supported.	According to the standard.
		ipAdEntBcastAddr	(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.
		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 *ipRouteDest*.	---	---
		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 *ifIndex*, which is used to identify the interface.	Supported.	According to the standard.

Subtree name		Standard [(*identifier*) *attribute*]	Support	Implementation specifications
	ipRouteMetric1	(3) RW The primary routing metric for this route. This value is determined based on the protocol specified in *ipRouteProto*. Set to −1 if you do not want to use this metric (this is also the same for *ipRouteMetric* 2 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.
	ipRouteProto	(9) RO 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. Other (1): Other than the following items. Local (2): A route set on the local machine. Netmgmt (3): A route set by network management. Icmp (4): A route set by an ICMP redirect or some other ICMP function. Egp (5): EGP The following are gateway protocols: Ggp (6): GGP Hello (7): HELLO Rip (8): RIP is-is (9) es-is (10) ciscolgrp (11) bbnSpflgp (12) ospf (13): OSPF bgp (14)	Supported.	According to the standard.
	ipRouteAge	(10) RW The elapsed time since this route was updated (in seconds).	Supported.	According to the standard.

Subtree name	Standard [(*identifier*) *attribute*]	Support	Implementation specifications
ipRouteMask	(11) RW The subnet mask value in relation to *ipRouteDest*. On systems that do not support a custom subnet mask value, this value is based on the address class of the *ipRouteDest* field. If *ipRouteDest* is 0.0.0.0, this value is also 0.0.0.0.	Supported.	According to the standard.
ipRouteMetric5	(12) RW The alternative routing metric.	Supported.	According to the standard.
ipRouteInfo	(13) RO The MIB object ID for the routing protocol used by this route. If not defined, set to {0.0}.	Supported.	0.0
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 *ipNetToMediaIfIndex* and *ipNetToMediaNetAddress*.	---	---
ipNetToMediaIfIndex	(1) RW The interface ID number for this entry. The value of *ifIndex* 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.
ipNetToMediaType	(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 [(*identifier*) *attribute*]	Support	Implementation specifications
icmpInMsgs	(1) RO The total number of received ICMP messages. This includes messages counted by *icmpInErrors*.	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 *Destination Unreachable* messages received.	Supported.	According to the standard.
icmpInTimeExcds	(4) RO The number of *Time Exceed* messages received.	Supported.	According to the standard.
icmpInParmProbs	(5) RO The number of *Parameter Problem* messages received.	Supported.	According to the standard.
icmpInSrcQuenchs	(6) RO The number of *Source Quench* messages received.	Supported.	According to the standard.
icmpInRedirects	(7) RO The number of *Redirect* messages received.	Supported.	According to the standard.
icmpInEchos	(8) RO The number of *Echo (request)* messages received.	Supported.	According to the standard.
icmpInEchoReps	(9) RO The number of *Echo Reply* messages received.	Supported.	According to the standard.
icmpInTimestamps	(10) RO The number of *Timestamp* messages received.	Supported.	According to the standard.
icmpInTimestampReps	(11) RO The number of *Timestamp Reply* messages received.	Supported.	According to the standard.
icmpInAddrMasks	(12) RO The number of *Address Mask Request* messages received.	Supported.	According to the standard.
icmpInAddrMaskReps	(13) RO The number of *Address Mask Reply* messages received.	Supported.	According to the standard.
icmpOutMsgs	(14) RO The total number of ICMP messages sent. This includes messages counted by *icmpOutErrors*.	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 *Destination Unreachable* messages sent.	Supported.	According to the standard.
icmpOutTimeExcds	(17) RO The number of *Time Exceed* messages sent.	Supported.	According to the standard.
icmpOutParmProbs	(18) RO The number of *Parameter Problem* messages sent.	Supported.	According to the standard.
icmpOutSrcQuenchs	(19) RO The number of *Source Quench* messages sent.	Supported.	According to the standard.

Name	Standard [(*identifier*) *attribute*]	Support	Implementation specifications
icmpOutRedirects	(20) RO The number of *Redirect* messages sent.	Supported.	According to the standard.
icmpOutEchos	(21) RO The number of *Echo (request)* messages sent.	Supported.	According to the standard.
icmpOutEchoReps	(22) RO The number of *Echo Reply* messages sent.	Supported.	According to the standard.
icmpOutTimestamps	(23) RO The number of *Timestamp* messages sent.	Supported.	According to the standard.
icmpOutTimestampReps	(24) RO The number of *Timestamp Reply* messages sent.	Supported.	According to the standard.
icmpOutAddrMasks	(25) RO The number of *Address Mask Request* messages sent.	Supported.	According to the standard.
icmpOutAddrMaskReps	(26) RO The number of *Address Mask Reply* messages sent.	Supported.	According to the standard.

● Ip Group: Tcp

Name	Standard [(*identifier*) *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 [(*identifier*) *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 *tcpConnLocalAddress*, *tcpConnLocalPort*, *tcpConnRemAddress*, and *tcpConnRemPort*.	---	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 [(*identifier*) *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 *udpNoPorts*.	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 *udpLocalAddress* and *udpLocalPort*.	---	According to the standard.
udpLocal Address	(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		Support	Standard [(*identifier*) *attribute*]
snmpInPkts	(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.
snmpInBadVersions	(3) RO The total number of messages received of an unsupported version.	Supported.	According to the standard.
snmpInBadCommunityNames	(4) RO The total number of messages received from an unregistered community.	Supported.	According to the standard.
snmpInBadCommunityUses	(5) RO The total number of messages received that specify an operation that is not allowed by that community.	Supported.	According to the standard.
snmpInASNParseErrs	(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.
snmpInTooBigs	(8) RO The total number of PDUs received with an error status of *tooBig*.	Supported.	According to the standard.

Name			Standard [(*identifier*) *attribute*]
snmpInNoSuchNames	(9) RO The total number of PDUs received with an error status of *noSuchName*.	Supported.	According to the standard.
snmpInBadValues	(10) RO The total number of PDUs received with an error status of *badValue*.	Supported.	According to the standard.
snmpInReadOnlys	(11) RO The total number of PDUs received with an error status of *readOnly*.	Supported.	According to the standard.
snmpInGenErrs	(12) RO The total number of PDUs received with an error status of *genErr*.	Supported.	According to the standard.
snmpInTotalReqVars	(13) RO The total number of MIB objects read normally after receiving *GetRequest* or *GetNextRequest*.	Supported.	According to the standard.
snmpInTotalSetVars	(14) RO The total number of MIB objects updated normally after receiving *SetRequest*.	Supported.	According to the standard.
snmpInGetRequests	(15) RO The total number of *GetRequest* PDUs received.	Supported.	According to the standard.
snmpInGetNexts	(16) RO The total number of *GetNextRequest* PDUs received.	Supported.	According to the standard.
snmpInSetRequests	(17) RO The total number of *SetRequest* PDUs received.	Supported.	According to the standard.
snmpInGetResponses	(18) RO The total number of *GetResponse* PDUs received.	Supported.	According to the standard.
snmpInTraps	(19) RO The total number of trap PDUs received.	Supported.	According to the standard.
snmpOutTooBigs	(20) RO The total number of PDUs sent with an error status of *tooBig*.	Supported.	According to the standard.
snmpOutNoSuchNames	(21) RO The total number of PDUs sent with an error status of *noSuchName*.	Supported.	According to the standard.
snmpOutBadValues	(22) RO The total number of PDUs sent with an error status of *badValue*.	Supported.	According to the standard.
snmpOutGenErrs	(24) RO The total number of PDUs sent with an error status of *genErr*.	Supported.	According to the standard.
snmpOutGetRequests	(25) RO The total number of *GetRequest* PDUs sent.	Supported.	According to the standard.
snmpOutGetNexts	(26) RO The total number of *GetNextRequest* PDUs sent.	Supported.	According to the standard.
snmpOutSetRequests	(27) RO The total number of *SetRequest* PDUs sent.	Supported.	According to the standard.
snmpOutGetResponses	(28) RO The total number of *GetResponse* 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*]
snmpEnableAuthen Traps	(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 CPU Unit.

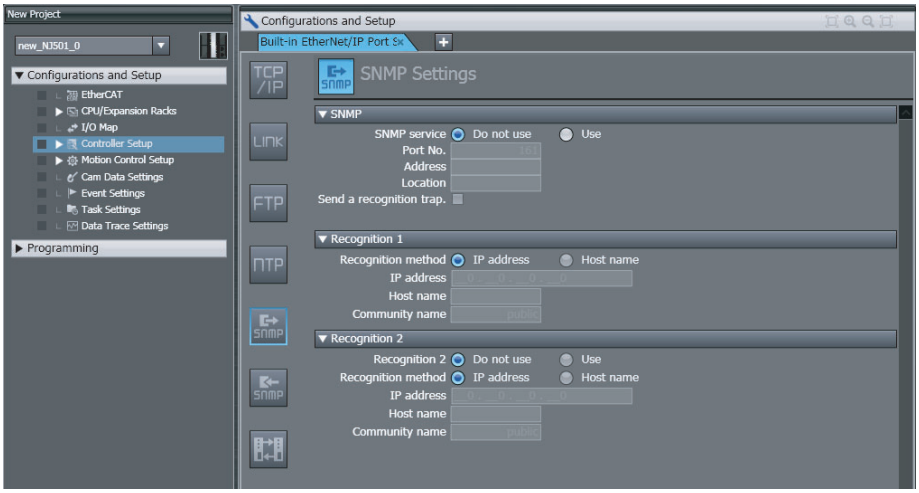
12-2-2 Settings Required for the SNMP Agent

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 12-22
	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 12-23
		IP address		
		Host name		
		Community name		

Tab page	Setting		Setting conditions	Reference
SNMP Trap Settings	SNMP trap		Required.	page 12-24
	Port No.		Specified by user. **Note** Required to change from the default value of 162.	
	Trap 1 and trap 2			page 12-24
		IP address	Required. Set an IP address or a host name as the SNMP trap destination.	
		Host name		
		Community name	Specified by user.	
		Version	Required. Set the version of the SNMP manager.	

12-2-3 SNMP Settings Display



● SNMP

Setting	Description	Default
SNMP	Specify whether to use the SNMP monitor service. If the SNMP monitor service is not enabled, an SNMP manager cannot connect from an external device.	Do not use.
Port No.	Set the port number to use to connect to the SNMP server that is used to connect from an SNMP manager. It is normally not necessary to change this setting.	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 alphanumeric characters for each item.)	None
Location		None
Send a recognition trap	Set whether to send a recognition trap. If you select the *Send a recognition trap* Check Box and there is access from an SNMP manager that is not set in Recognition 1 or Recognition 2, a recognition trap is sent to the SNMP manager. If you enable the send a recognition trap, specify the SNMP trap settings on the SNMP Trap Tab Page.	Do not use.

● Recognition 1*

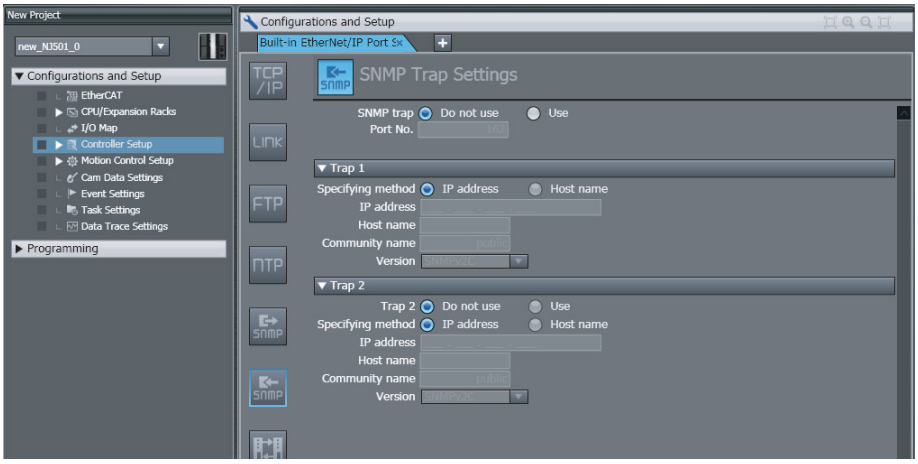
Setting	Description	Default
Recognition method	Set the method to use to specify SNMP managers for which access is permitted. • IP address • Host name Make these settings to permit access by only certain SNMP managers. Access is not allowed unless an IP address or host name is set.	IP address
IP address	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 *IP address* 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 *Host name* 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. • Use. • Do not use.	Do not use.
	Recognition method	Set the method to use to specify SNMP managers for which access is permitted. • 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 recognition method 2 is set to the *IP address* 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 *Host name* 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

* These settings are required if the SNMP service is enabled.

12-2-4 SNMP Trap Settings Display



● SNMP Trap

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

● Trap 1*

Setting	Description	Default
Specifying method	Set the specifying method for the SNMP manager destination for SNMP traps. • 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 *IP address* 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 *Host name* 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 characters.)	public
Version	Set the version of the SNMP manager. • SNMP version 1: SNMPv1 • SNMP version 2C: SNMPv2C	SNMPv1

● Trap 2*

Setting		Description	Default
Trap 2		Specify whether to use the trap 2 settings. • Use. • Do not use.	Do not use.
	Specifying method	Set the specifying method for the SNMP manager destination for SNMP traps. • 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 *IP address* 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 *Host name* 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 characters.)	public
	Version	Set the version of the SNMP manager. • SNMP version 1: SNMPv1 • SNMP version 2C: SNMPv2C	SNMPv1

* These settings are required if the SNMP trap is set is enabled.

12-3 Troubleshooting SNMP Agent

12-3-1 SNMP Agent Errors

The following table shows the main causes and corrections for errors that occur in SNMP agent.

Event log	Cause	Corrective action
SNMP Setting Error	• The power supply was turned OFF during a download operation for the SNMP agent/trap settings. • A memory error occurred.	• Restart the Controller.
	• There is a conflict in the SNMP agent/trap settings*	• Change the setting values to ensure they are within the setting range.

* The server where the error occurred can be checked based on the detail code in the event log.

12-3-2 Event Log Error Codes for the SNMP Agent

When an error occurs in the SNMP agent, the following information is recorded in the event log along with the date and time of the error, the error level, and the event source.

Event code	Event name	Cause	Level	Source details	Correction
34260000 hex	SNMP Setting Error	• The power supply was turned OFF during a download operation for the SNMP agent/trap settings. • A memory error occurred.	Minor fault level Controller error	SNMP	Restart the CPU Unit. If restarting the CPU Unit does not restore operation, perform one of the following operations and then download the settings again. • Perform a Memory All Clear operation. • Identify the cause of the error based on the detailed information in the event log, and correct the settings. If this does not restore the system, replace the CPU Unit.
940C0000 hex	SNMP Agent Started	• The SNMP agent server started normally.	---	SNMP	---

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 13-21
13-3-3 Relationship between Task Periods and Packet Intervals (RPIs) 13-23
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 ≥ 100 ms, the heartbeat frame transmission period is the same as the RPI.

Example:

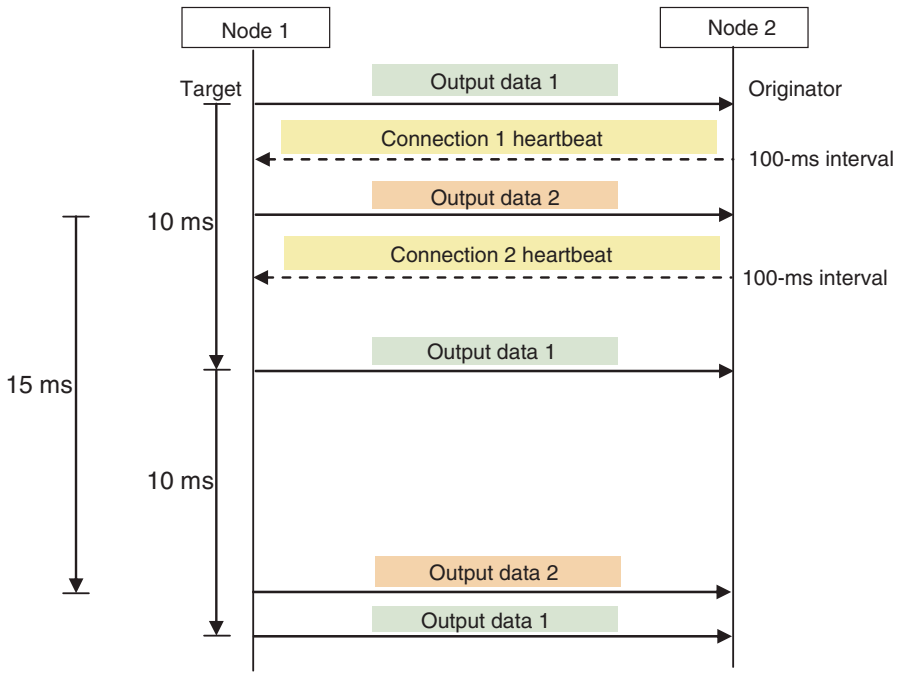
In this example, 2 tag data link connections are set for node 2 (the originator) and node 1 (the target).

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 ÷ RPI (ms)) + (1,000 ÷ 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.

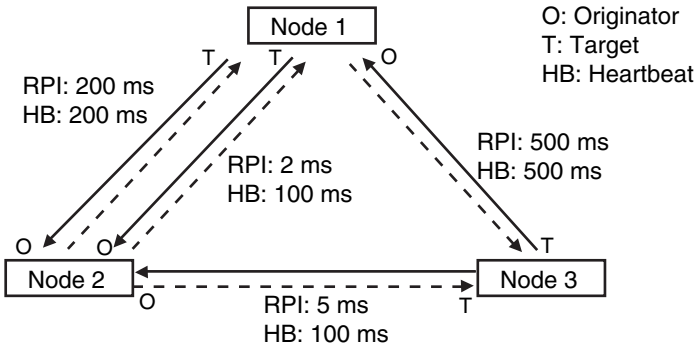
The maximum number of packets that the CPU Unit can transfer in 1 second (called the allowed Unit bandwidth) is 1,000 pps, so set the connection below this maximum value.

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 ms + 1000 / 2 ms + 1000 / 500 ms (for data)
  + 1000 / 200 ms + 1000 / 100 ms + 1000 / 500 ms (for heartbeat)
  = 524 pps
- Total PPS of node 2's Unit
  = 1000 / 200ms + 1000 / 2 ms + 1000 / 5 ms (for data)
  + 1000 / 200ms + 1000 / 100 ms + 1000 / 100 ms (for heartbeat)
  = 730 pps
- Total PPS of node 3's Unit
  = 1000 / 5 ms + 1000 / 500 ms (for data)
  + 1000 / 100 ms + 1000 / 500 ms (for heartbeat)
  = 214pps

All of the Units are within the allowed Unit bandwidth of 1,000 pps, so they can transfer data.

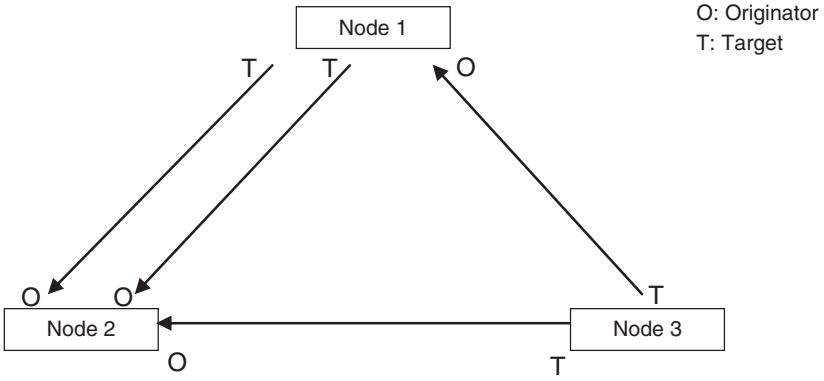
13-1-2 Calculating the Number of Connections

The maximum number of connections for the built-in EtherNet/IP port is 32.

The number of connections must be set to 32 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 maximum number of connections for the built-in EtherNet/IP port is 32 max.

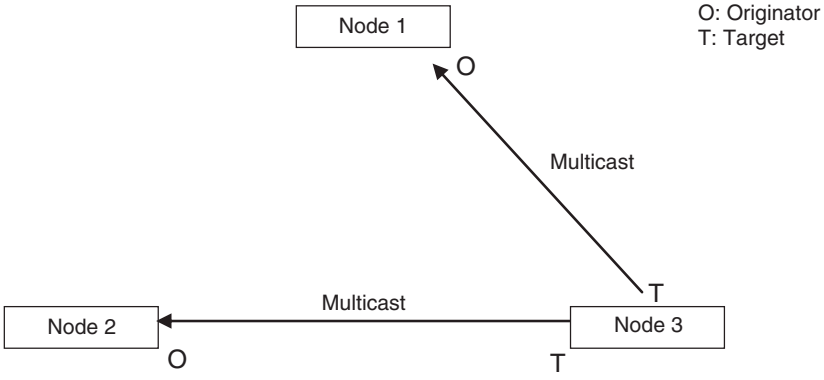


Also, if multicast is set, one packet will be sent, but the number of connections will be consumed.

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.

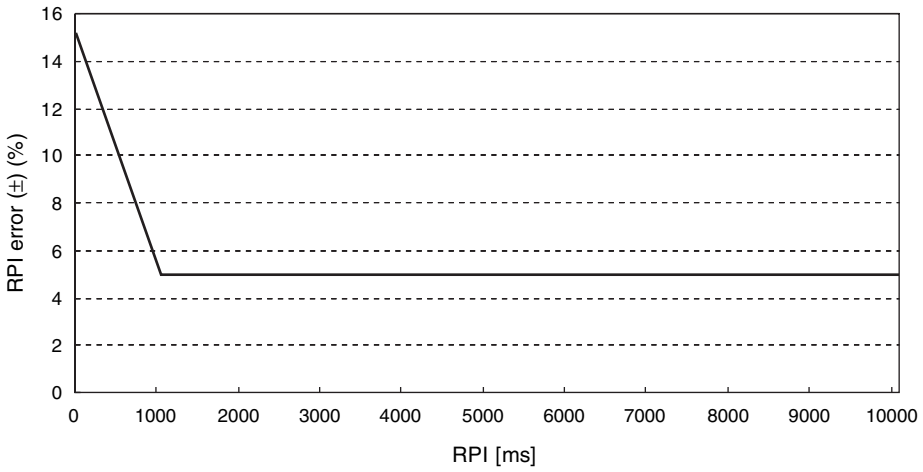
Caution is required because the number of connections consumed is the same as for unicast connections even when multicast connections are set.



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 (±) (%)
10 to 1,000 ms	15 – (RPI [ms]/100)
1,000 to 10,000 ms	5% of the RPI



13-2 Adjusting the Communications Load

In an Ethernet network using a 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	Settings
Network bandwidth	Physical Ethernet baud rate	100 Mbps or 10 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)	1,000 pps max.
Connection resources	Number of connections that can be established	32 max.
Packet interval (RPI: Requested Packet Interval)	Refresh period for tag data	10 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 a 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

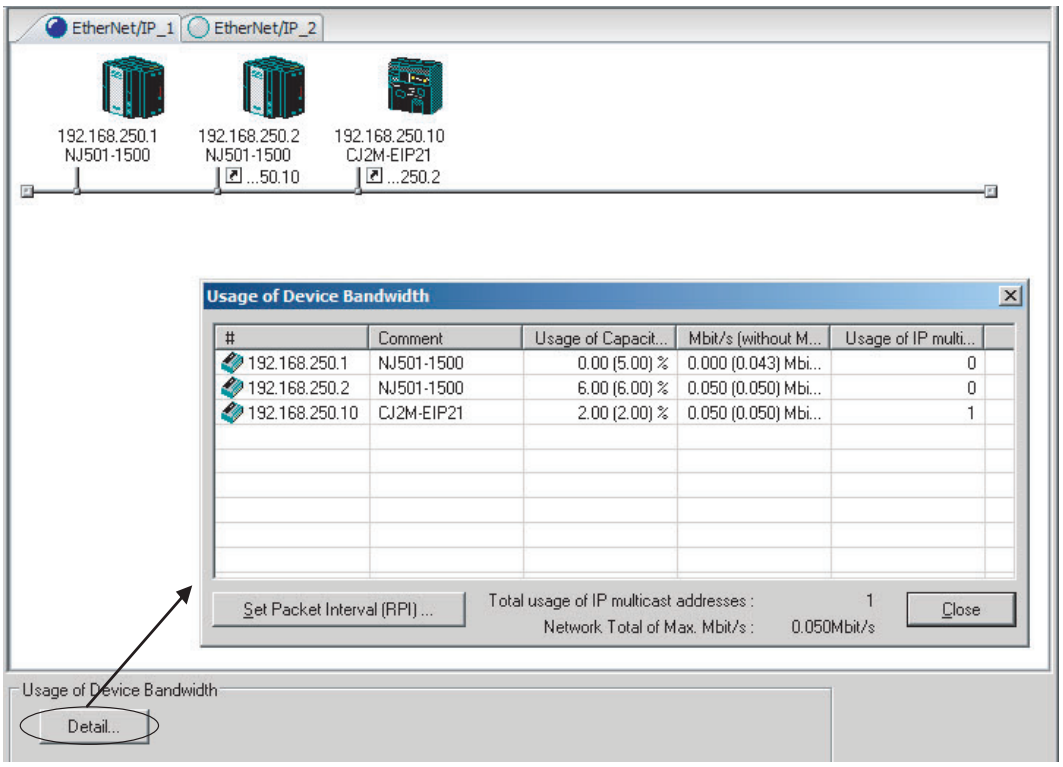
If the Network Configurator is used to set the connection type in the connection settings to a multicast connection, multicast packets will be used. If the connection type is set to a point-to-point connection, multicast packets are not used.

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 100-Mbps 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 100 Mbps is exceeded for the network bandwidth.

● 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 a 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 a 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 a Ethernet switch with enough multicast filters, or adjust the usage of capacity and network bandwidth for tag data links (*Mbit/s*) values given for a 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., 100 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

● Ethernet Switches without Multicast Filtering (100-Mbps Ethernet Switches)

- Is the network bandwidth without multicast filtering usage under 100 Mbps for each node?
  If any node exceeds 100 Mbps, 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 Mbps, change the connections settings, such as the RPI.
- Is the total network bandwidth usage under 100 Mbps?
  If the total bandwidth usage exceeds 100 Mbps, the bandwidth of part of the transmission path (e.g., a 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 (100-Mbit/s Ethernet Switches)

- Is the network bandwidth usage under 100 Mbps for each node?
  If any node exceeds 100 Mbps, 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 100 Mbps?
  If the total bandwidth usage exceeds 100 Mbps, the bandwidth of part of the transmission path (e.g., a 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 100 Mbps for each node or the usage of capacity without multicast filtering under 100% for each node?
  If any node exceeds 100 Mbps or 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.

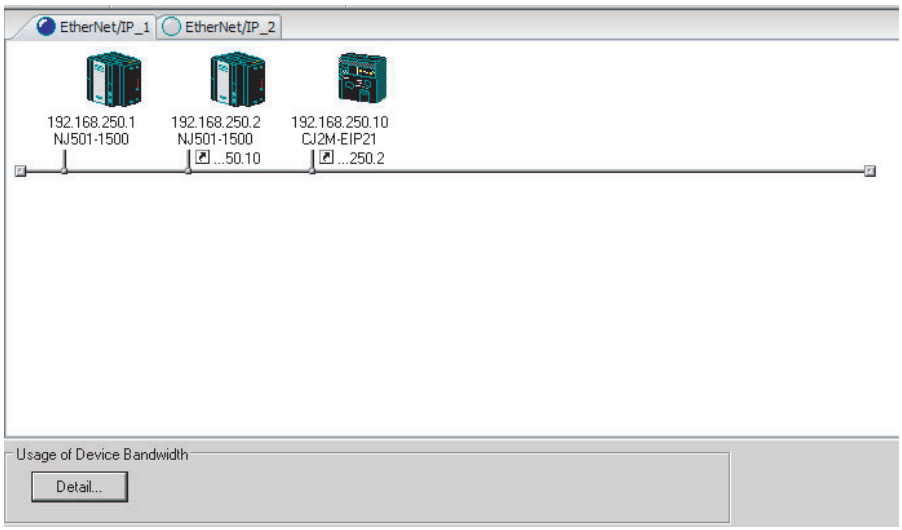
13-2-4 Changing the RPI

You can check the usage of capacity offline without multicast filtering against the tag data link's allowable bandwidth by following the procedures in *13-2-1 Checking Bandwidth Usage for Tag Data Links*. The usage of capacity without multicast filtering can be adjusted against the tag data link's allowable bandwidth by changing the packet interval (RPI).

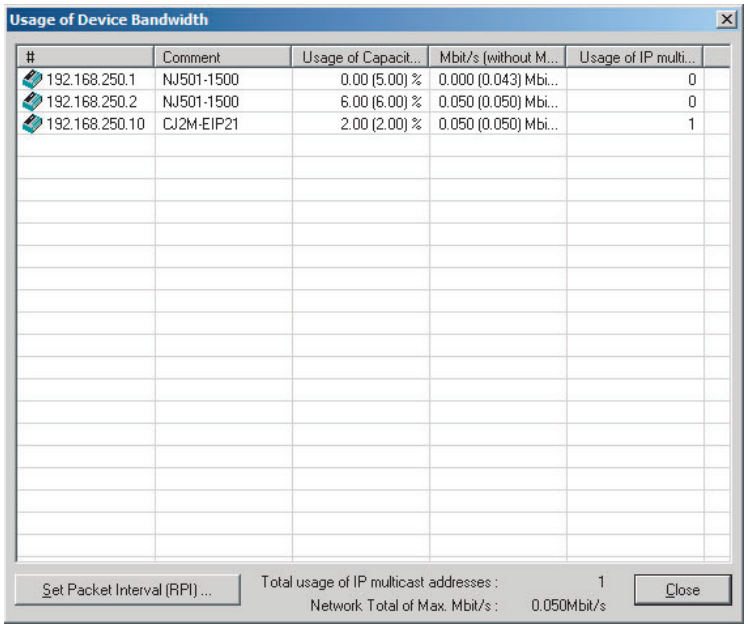
If the required communications performance cannot be achieved by changing the settings, re-evaluate the network starting with the network configuration.

1 Make the required settings in the Network Configurator's Network Configuration Window.

2 Click the **Detail** Button in the Usage of Device Bandwidth Area at the bottom of the Network Configuration Window.



The Usage of Device Bandwidth Dialog Box will be displayed.



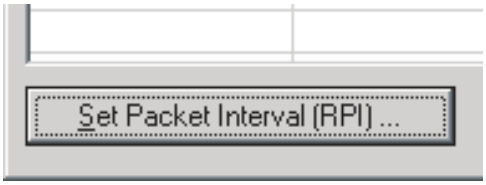
The *Usage of Capacity* (without multicast filter) column will show the percentage of the allowed tag data link bandwidth being used, and the *Mbit/s* (without multicast filter) column will show the network bandwidth being used.

3 The usage of capacity without multicast filtering can be adjusted against the tag data link's allowable bandwith by changing the associated devices' packet interval (RPI) settings.

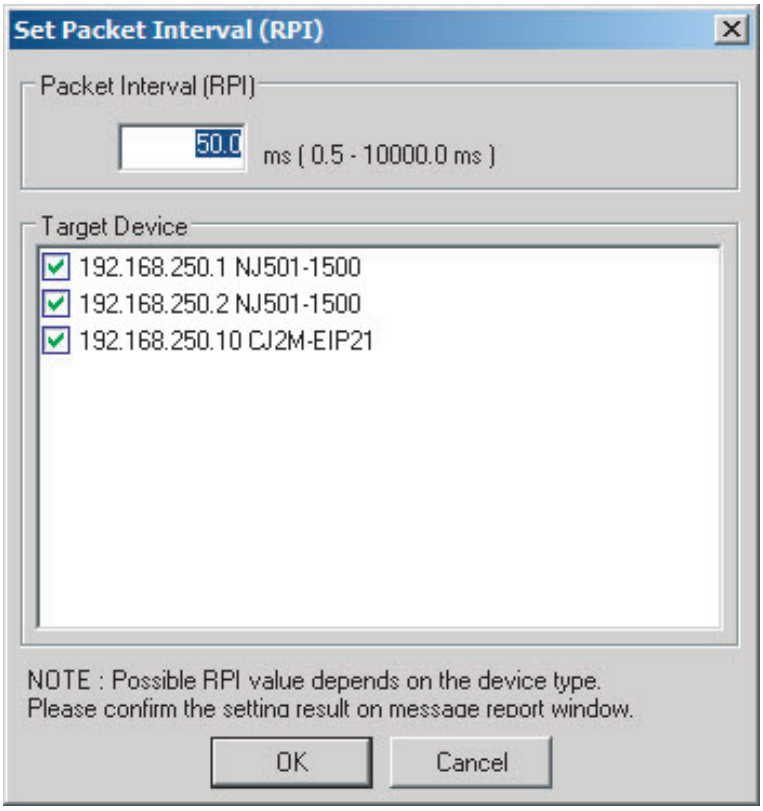
The RPI settings can be changed with the following three methods.

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.

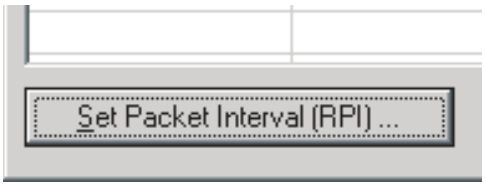


(2) The Set Packet Interval (RPI) Dialog Box will be displayed. Input a new RPI value, and click the **OK** Button.

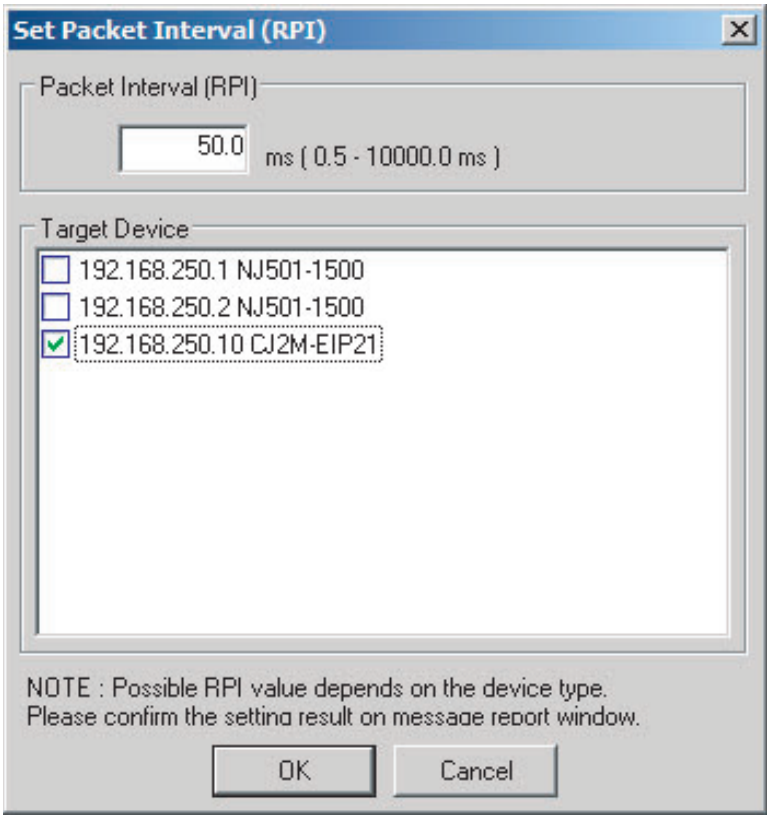


Method 2:
Change a Particular Device's Packet Interval (RPI) Setting:
The usage of capacity without multicast filtering can be adjusted for only a particular device against the tag data link's allowable bandwith by changing the packet intervals (RPI) settings for all of the device's connections together. 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.

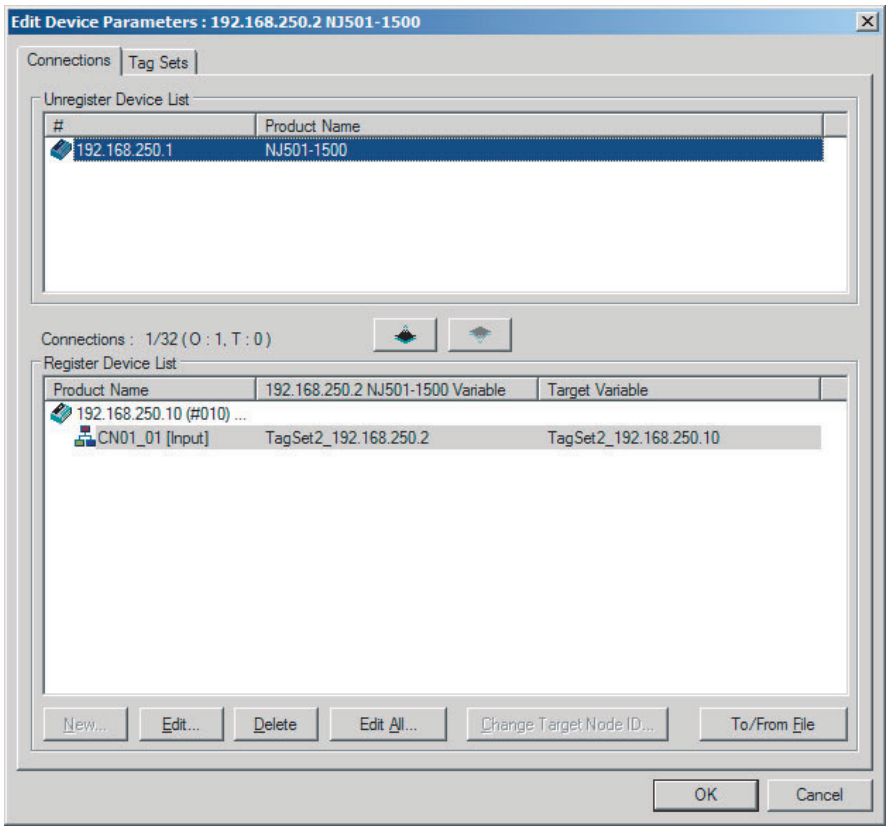


(3) Input a new RPI value, and click the **OK** Button.

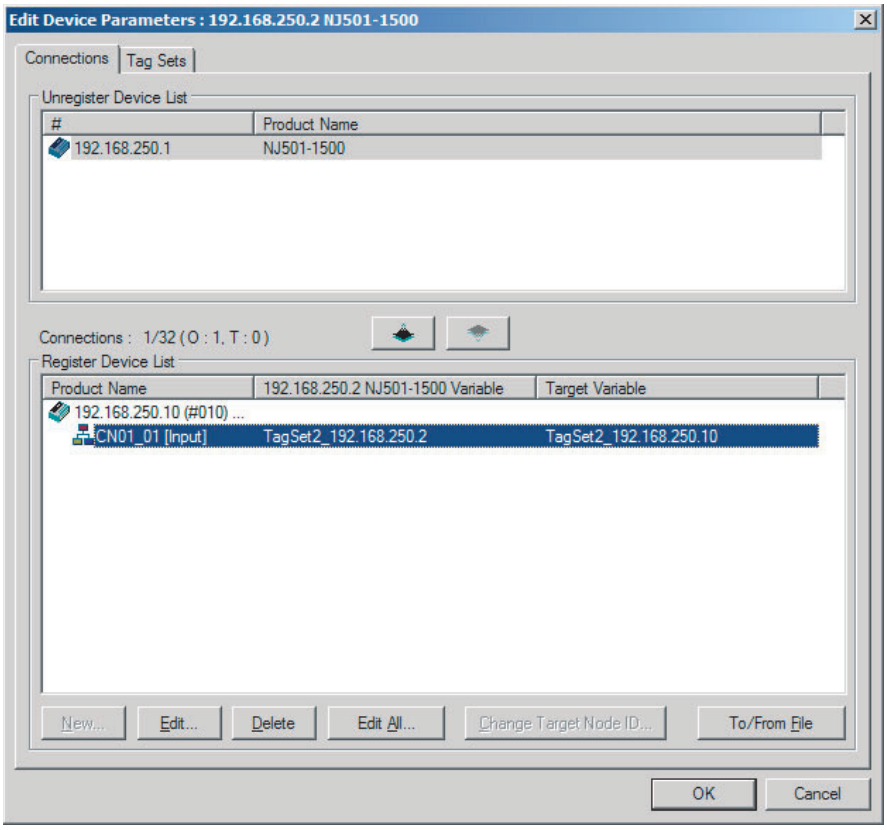
Method 3:
Changing a Particular Connection's Packet Interval (RPI) Setting:
The usage of capacity without multicast filtering can be adjusted against the tag data link's allowable bandwidth by individually changing the packet interval (RPI) for a particular connection. In this case, the usage of capacity 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.

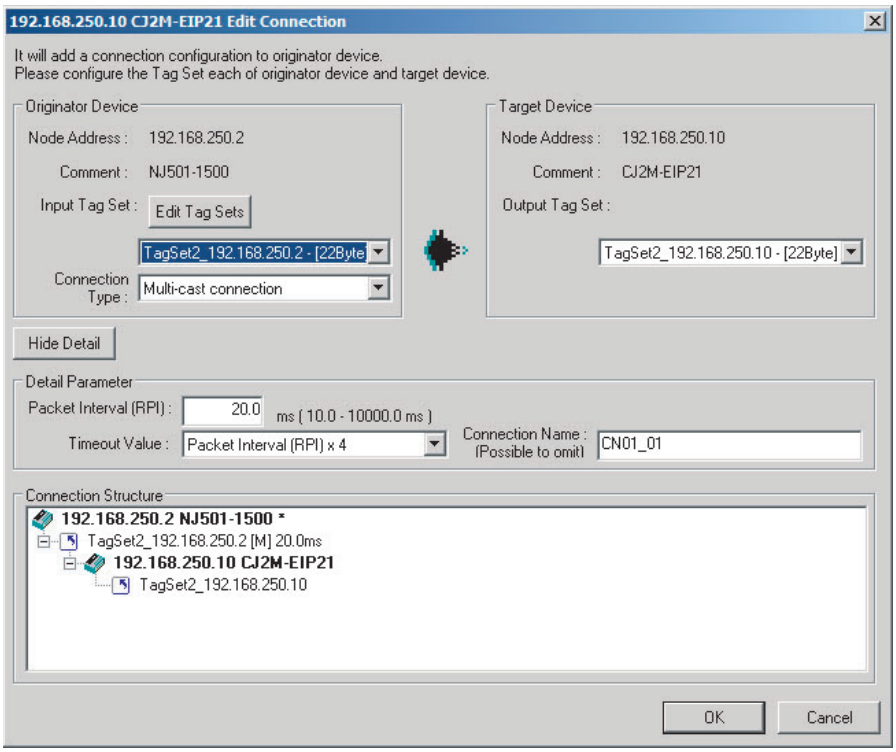
(2) Double-click the device that is set as the originator of the desired connection. The Edit Device Parameters Dialog Box will be displayed.



(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.

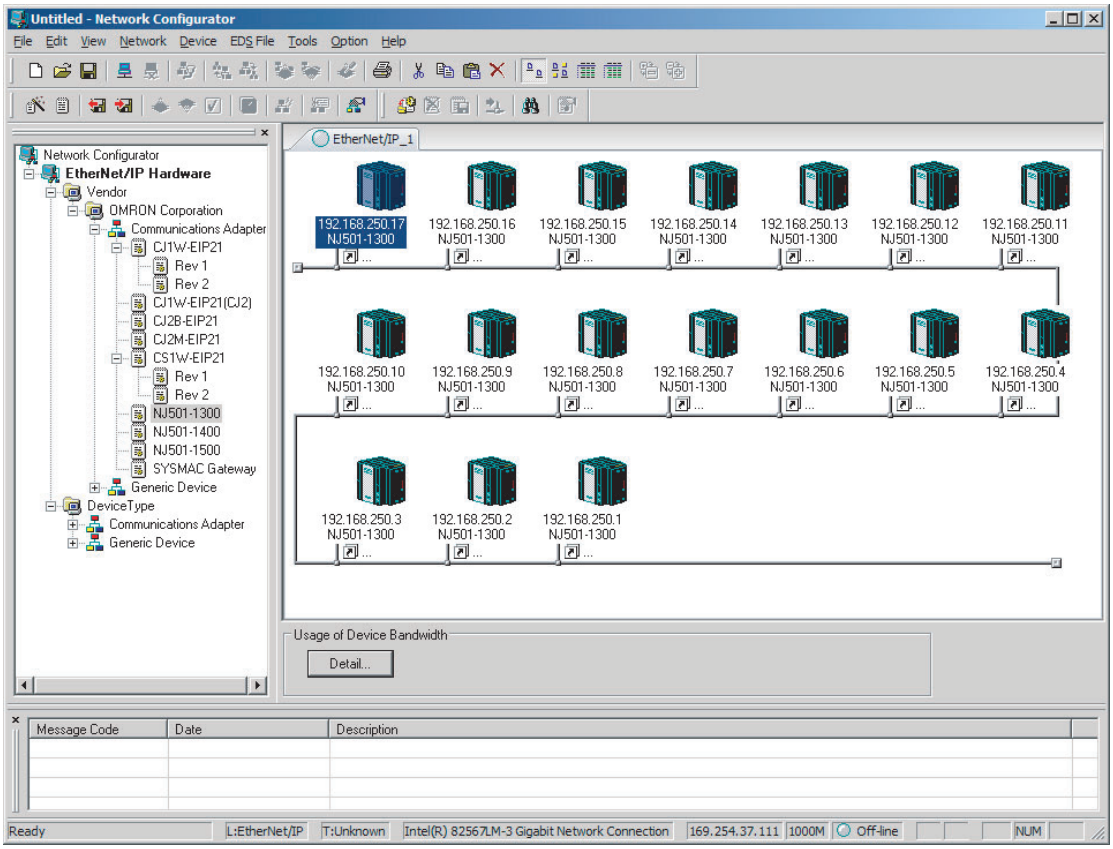
5 Check the bandwidth usage again.
If you have changed the connection settings, click the **Detail** Button in the Usage of Device Bandwidth Area at the bottom of the Network Configuration Window and check bandwidth usage according to the instructions in *13-2-1 Checking Bandwidth Usage for Tag Data 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

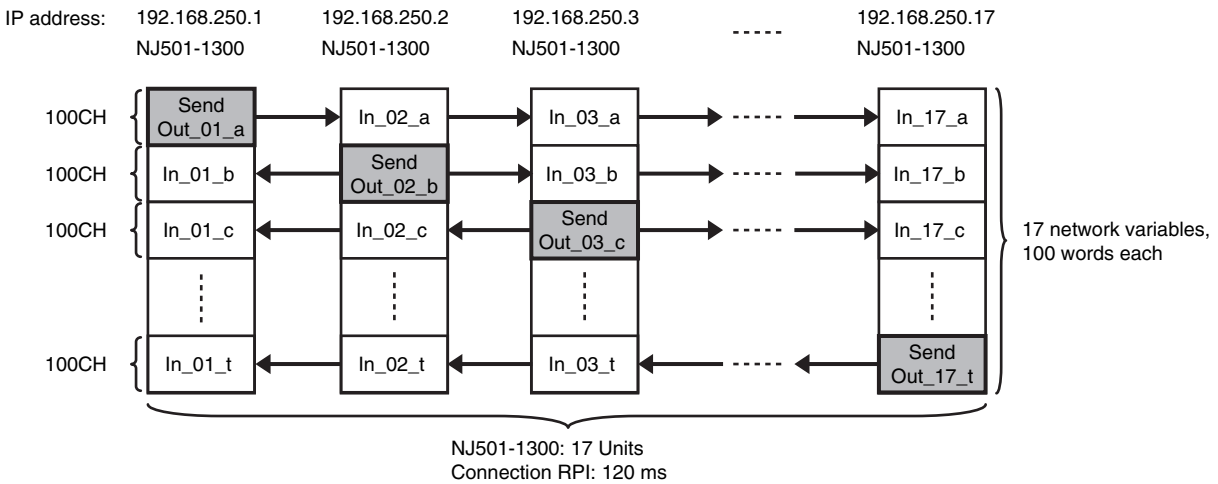
The following examples explain how to calculate the packet intervals (RPIs) in the following network configuration.

Conditions



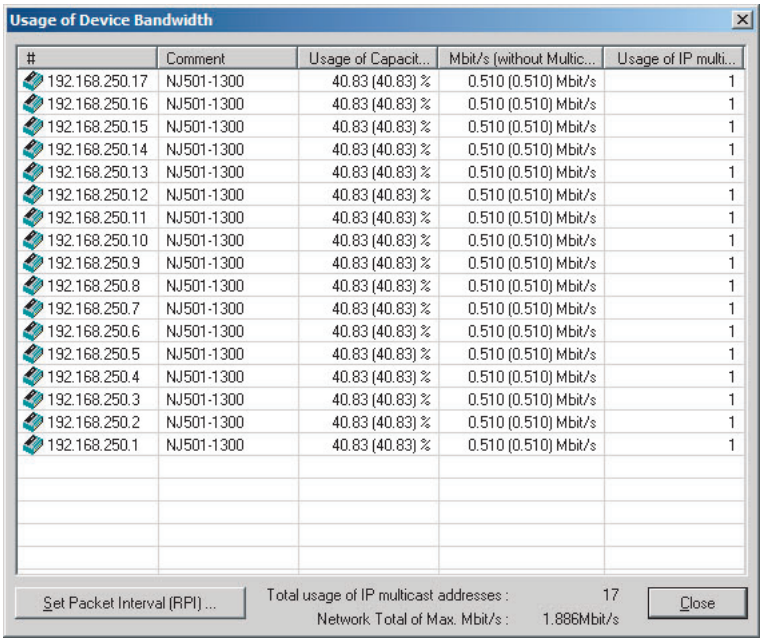
- Connections:

  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.



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.

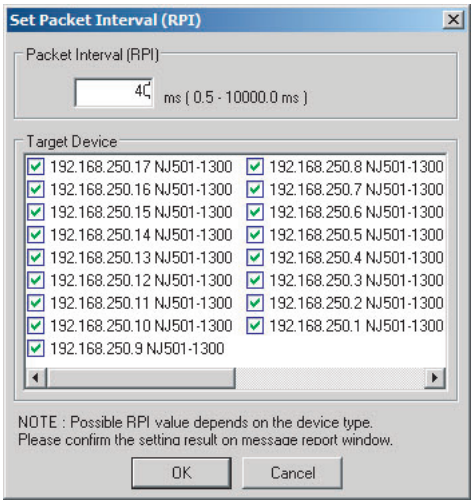


Changing Settings

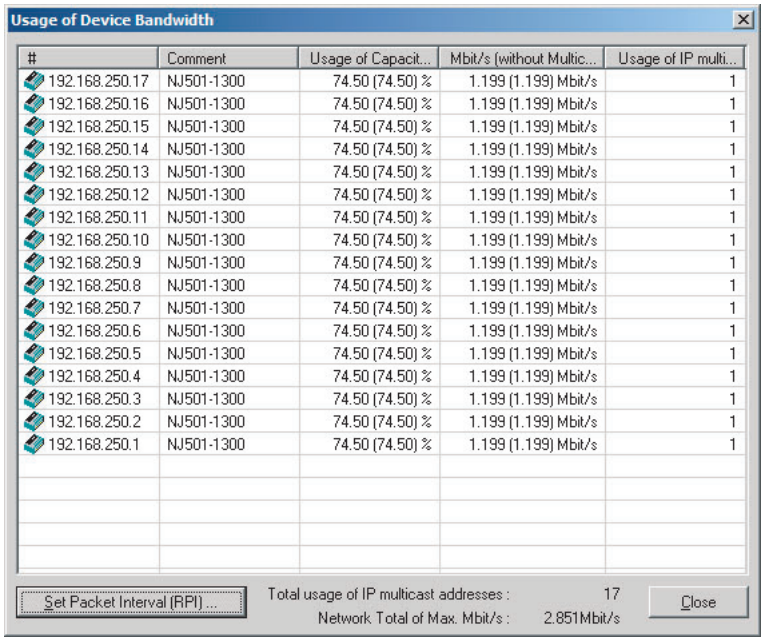
Method 1: Same Packet Interval Setting for All Connections

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.

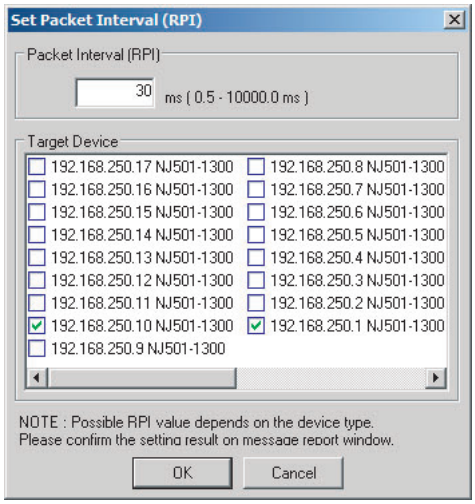


If the packet interval for all connections has been set to the same setting, the dialog box will show that the usage of capacity for the tag data link's allowable communications bandwidth is 74.50% and the fastest set value is 40 ms.

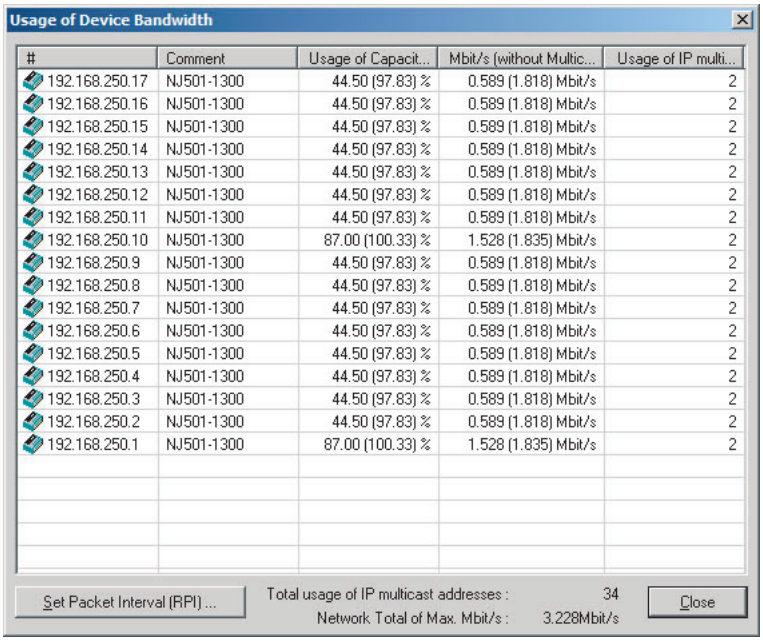


Method 2: Changing the Packet Interval (RPI) of Only Specific Devices

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.



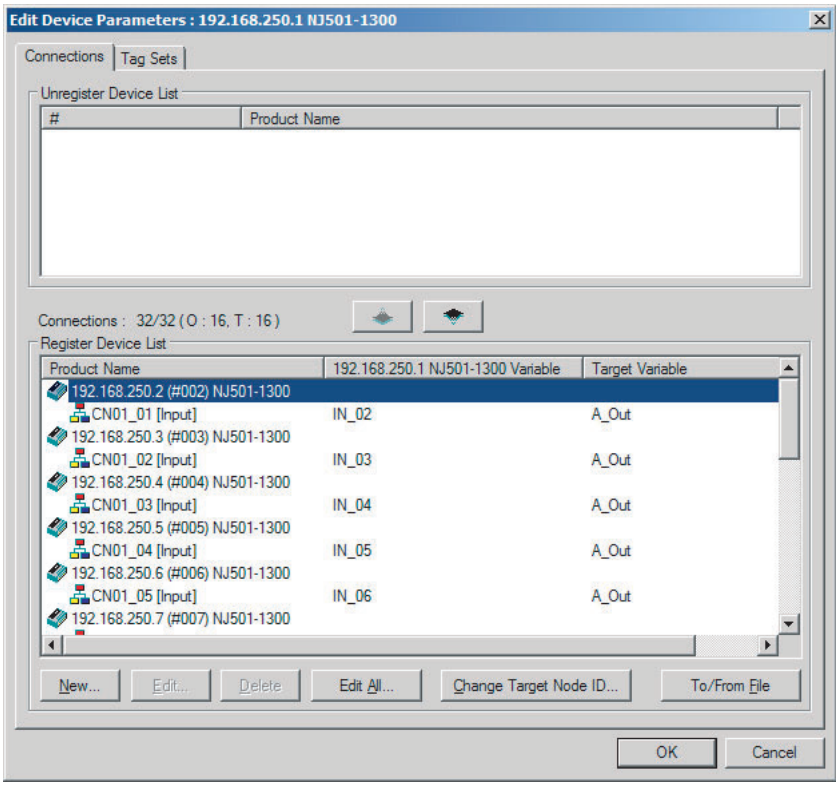
The percentage of the allowed tag data link bandwidth being used (Usage of Capacity) increases to 87.00% for devices 192.168.250.1 and 192.168.250.10, which indicates that the RPI is set to a higher speed for these devices' connections.
The Usage of Capacity values also indicate that the Usage of Capacity 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.



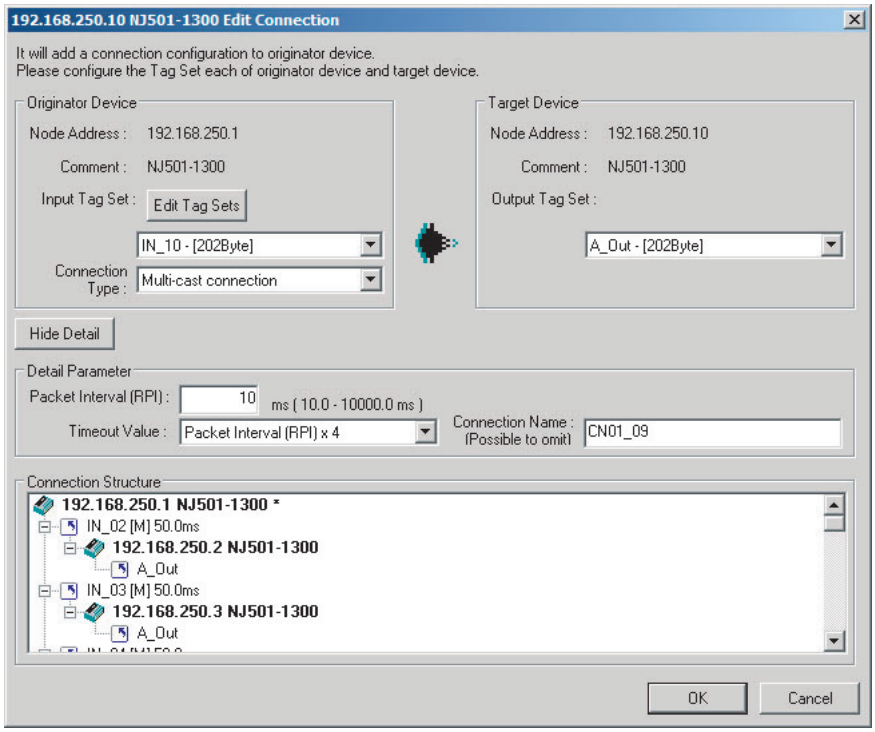
In this case, if there is no multicast filter, the value becomes 100.33%. If there is no multicast filter for a Ethernet switch, communications errors may occur depending on the communications load of the built-in EtherNet/ IP Unit port.

Method 3: Changing the Packet Intervals (RPIs) of Only Specific Connections

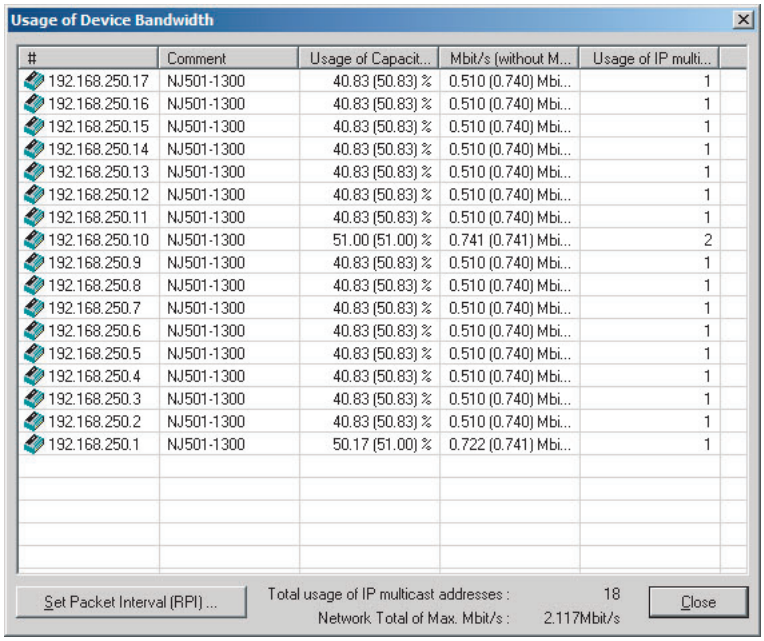
In this example, we want a faster tag data links for just a particular connection of device 192.168.250.1. Double-click device 192.168.250.1 in the Network Configuration Window.



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.



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%).

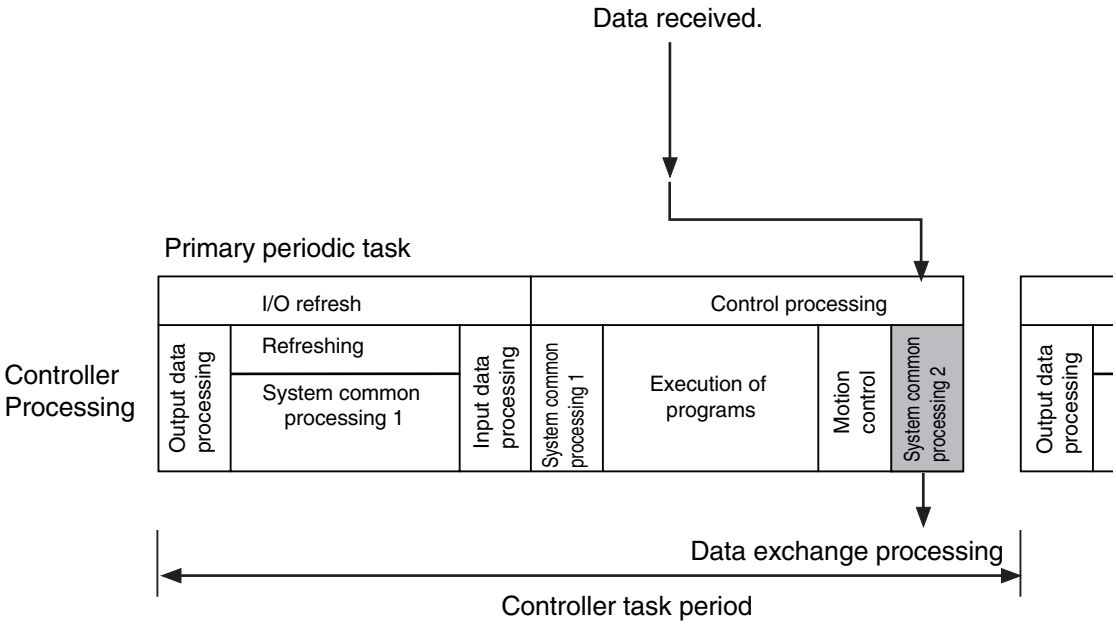
13-3 I/O Response Time in Tag Data Links

Additional Information

This section describes the response time of the NJ-series CPU Unit built-in Ethernet/IP port. The data processing times for the EtherNet/IP Unit, 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 Models Other than NJ-Series CPU Units*.

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 CPU Unit. 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.

You do not need to specify a refreshing task for tags that use an AT specification. Data is transferred for these tags during 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 CPU Unit.

Data Processing Time Overview

The time required for data processing consists of the following three elements.

1. Variable Access Time

First, we calculate the time required to transfer tag data.
This calculation is performed for each task. Therefore, if multiple tag sets are set for the same refreshing task, use the total for all tag values in the tag sets.
(Number of variables × 0.83 μs) + (Total number of bytes in variables × 0.0004 μs) + 7.29 μs

2. Number of Data Transfers

Tag data is transferred as a part of the task processing.
Therefore, if the time required to process the data transfer is greater than the time allowed for data transfer in the task process (i.e., the variable access time), the data cannot be sent entirely during one task period and is sent in fragments instead.

Number of data transfers = Time required for data transfer*1 ÷ Variable access time set for the task*2

*1 This is the time required for data transfer as calculated in (1) above.

*2 This is the variable access time that is set for each task in the Task Setup.

Precautions for Correct Use

The maximum number of words that can be transferred through the built-in EtherNet/IP port is 9,600 words for tag data links. If the number of tag data link words exceeds the number of words that can be exchanged with the CPU Unit 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.

Task period × 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.

- Refreshing task
  Primary periodic task
  Task period: 500 μs (variable access time: 3%)
- 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.

Tag set A: (8 variables × 0.55 μs) + (600 bytes × 0.0007 μs) + 7.1 μs = 11.92 μs
Tag set B: (4 variables × 0.55 μs) + (200 bytes × 0.0007 μs) + 7.1 μs = 9.44 μs
Tag set C: (10 variables × 0.55 μs) + (1,000 bytes × 0.0007 μs) + 7.1 μs = 13.3 μs

2 Calculate the number of data transfers.

Time required for the data transfer: 11.92 μs + 9.44 μs+13.3 μs = 34.66 μs
Variable access time set for the task: 500 μs × 0.03 = 15 μs
Number of data transfers: 34.66 μs ÷ 15 μs = 2.31 times

Thus, approximately three data transfers are required.

3 Calculate the actual time required for the data transfer.

500 μs × 3 times = 1,500 μs

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 The time required for the data transfer is calculated using the following formula and is set as the variable access time*1.

(Number of variables × 0.55 μs) + (Total number of bytes in variables × 0.0007 μs) + 7.1 μs

* If the same refreshing task is set for multiple tag sets, the total of all tag values in the tag sets is used.

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 *NJ-series CPU Unit Software User's Manual* (Cat. No. W501).

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.

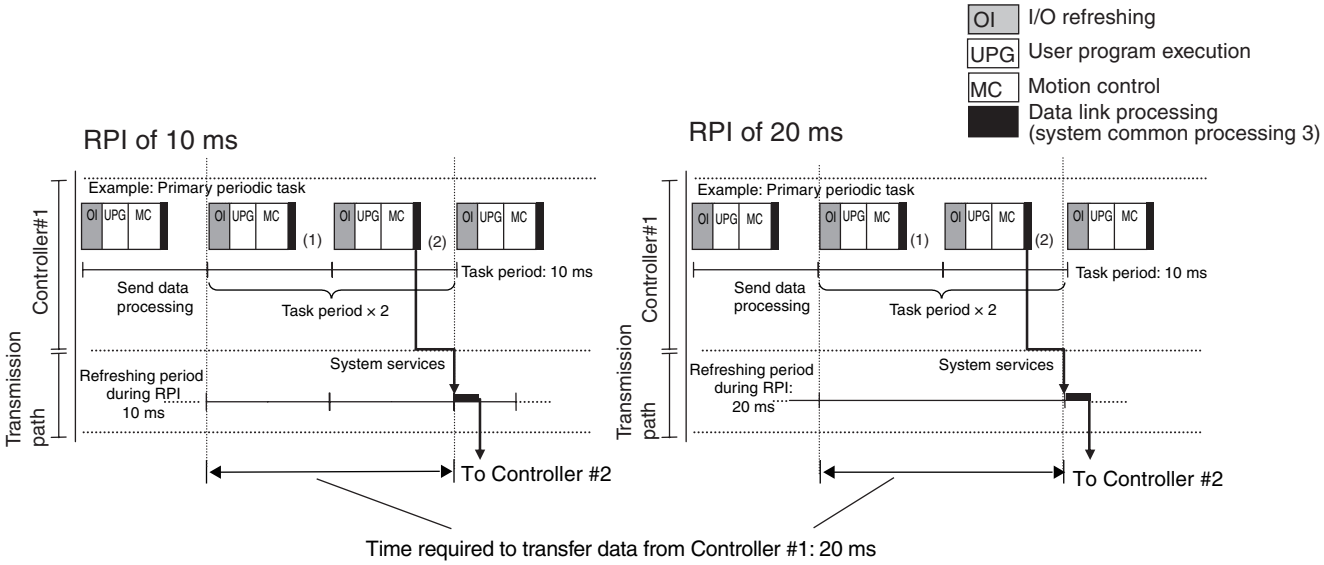
Actual time required for data transfer (Task period × Number of data transfers) < RPI

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

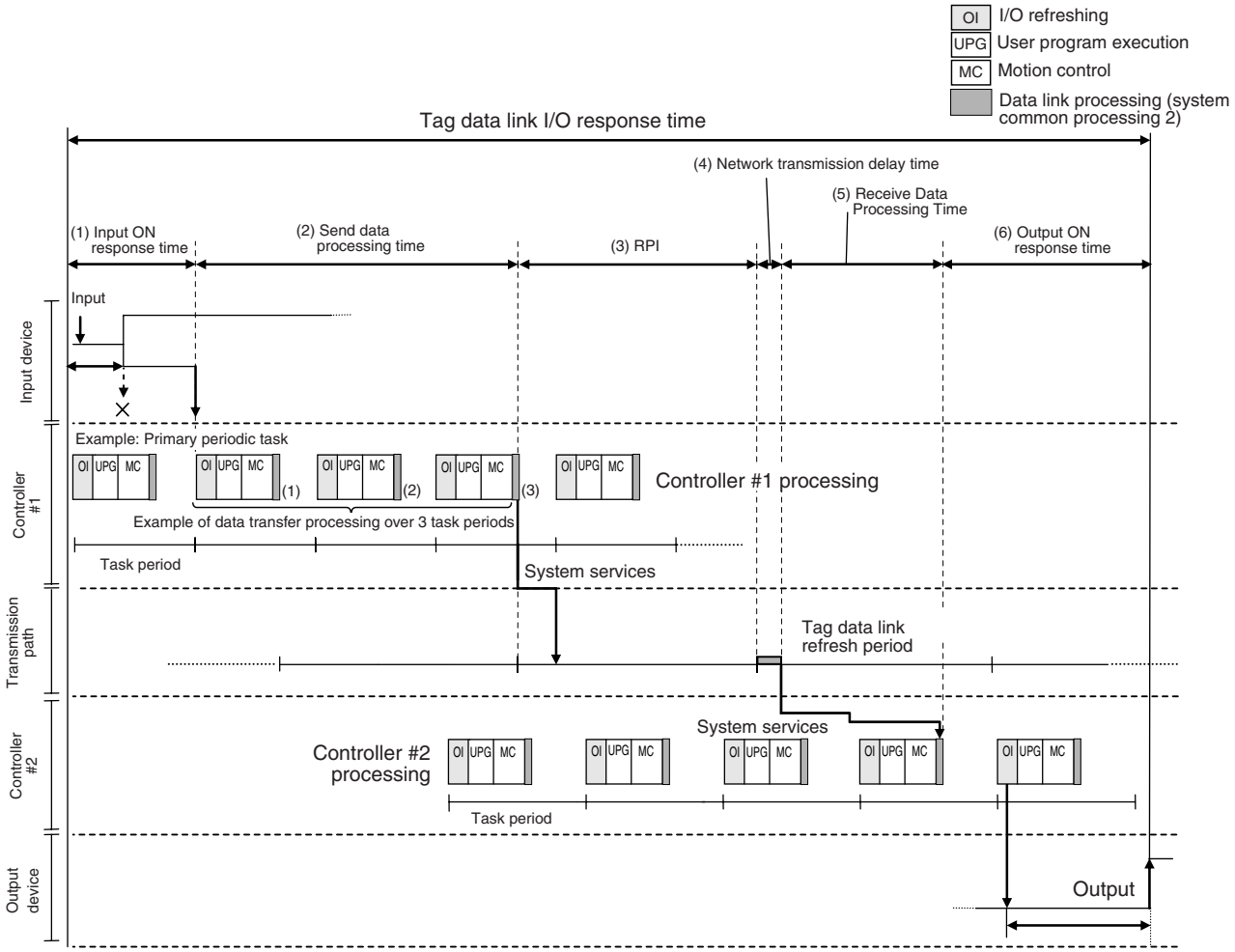
- Task period: 10 ms
- Number of data transfers: 2
- Actual time required for data transfer: 10 ms × 2 times = 20 ms

Regardless of the RPI value, the time required for the data transfer is 20 ms.


OI I/O refreshing
UPG User program execution
MC Motion control
Data link processing (system common processing 3)
RPI of 10 ms
RPI of 20 ms
Example: Primary periodic task
Controller#1
Transmission path
OI UPG MC
(1)
(2)
Task period: 10 ms
Send data processing
Task period × 2
System services
Refreshing period during RPI 10 ms
Refreshing period during RPI: 20 ms
To Controller #2
Time required to transfer data from Controller #1: 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.



Additional Information

Data link processing is executed as a system service. If a tag data link timeout occurs, reconsider the execution time for system services.

(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 CPU Unit. 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 CPU Unit. Therefore, the input ON response time is obtained as shown below.

Input ON response time = Input device delay time + Task period

(2) Send Data Processing Time

This is the time until the variables in the CPU Unit 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 μs 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 CPU Unit. 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 × 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

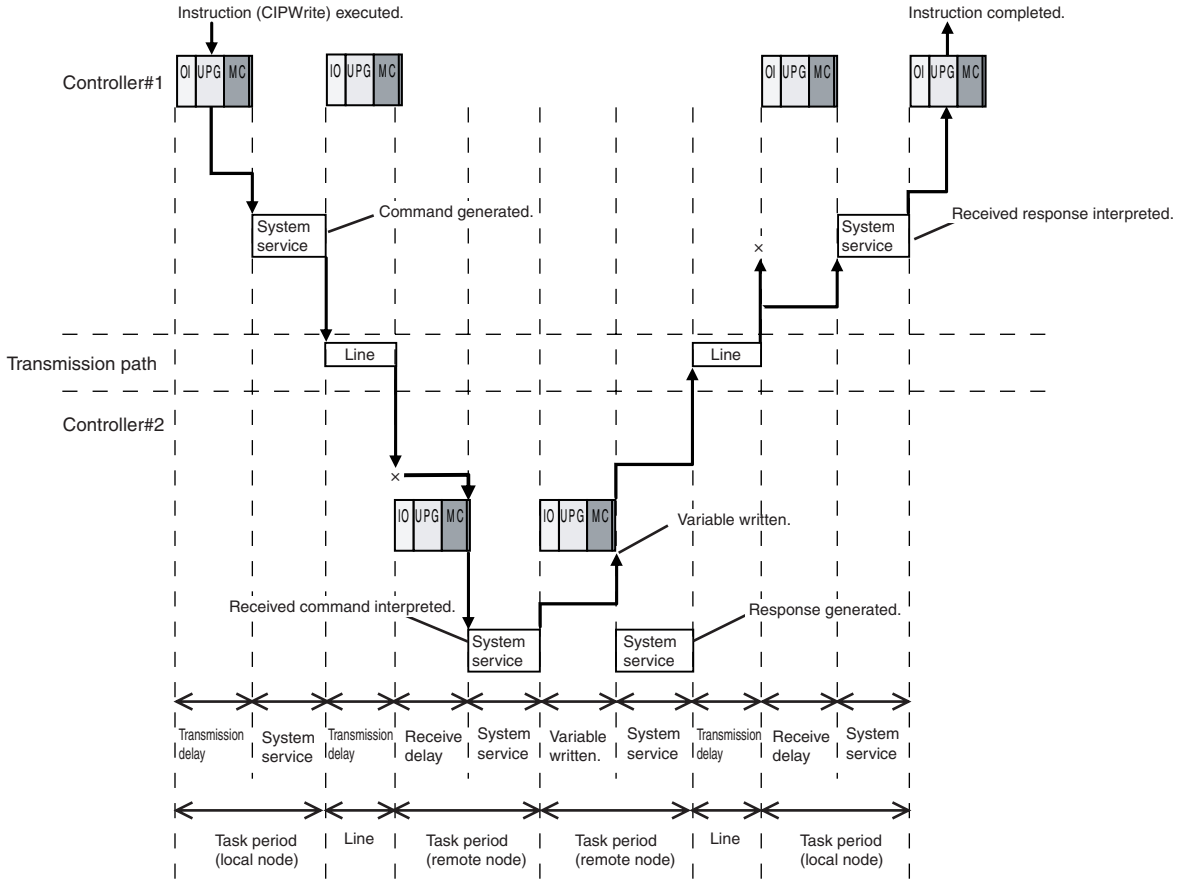
This is the delay time for the external output device from when the Controller specified turning ON the output until the output is actually turned ON.
Output ON response time = Output device delay time + Task period

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.



Maximum transmission delay time = Send delay + System service execution time + Transmission delay + Receive delay + System service execution time

(Local node task period) (Remote node task period)

+ Variable write time + System service execution time + Transmission delay + Receive delay + System service execution time

(Remote node task period) (Local node task period)

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

Troubleshooting

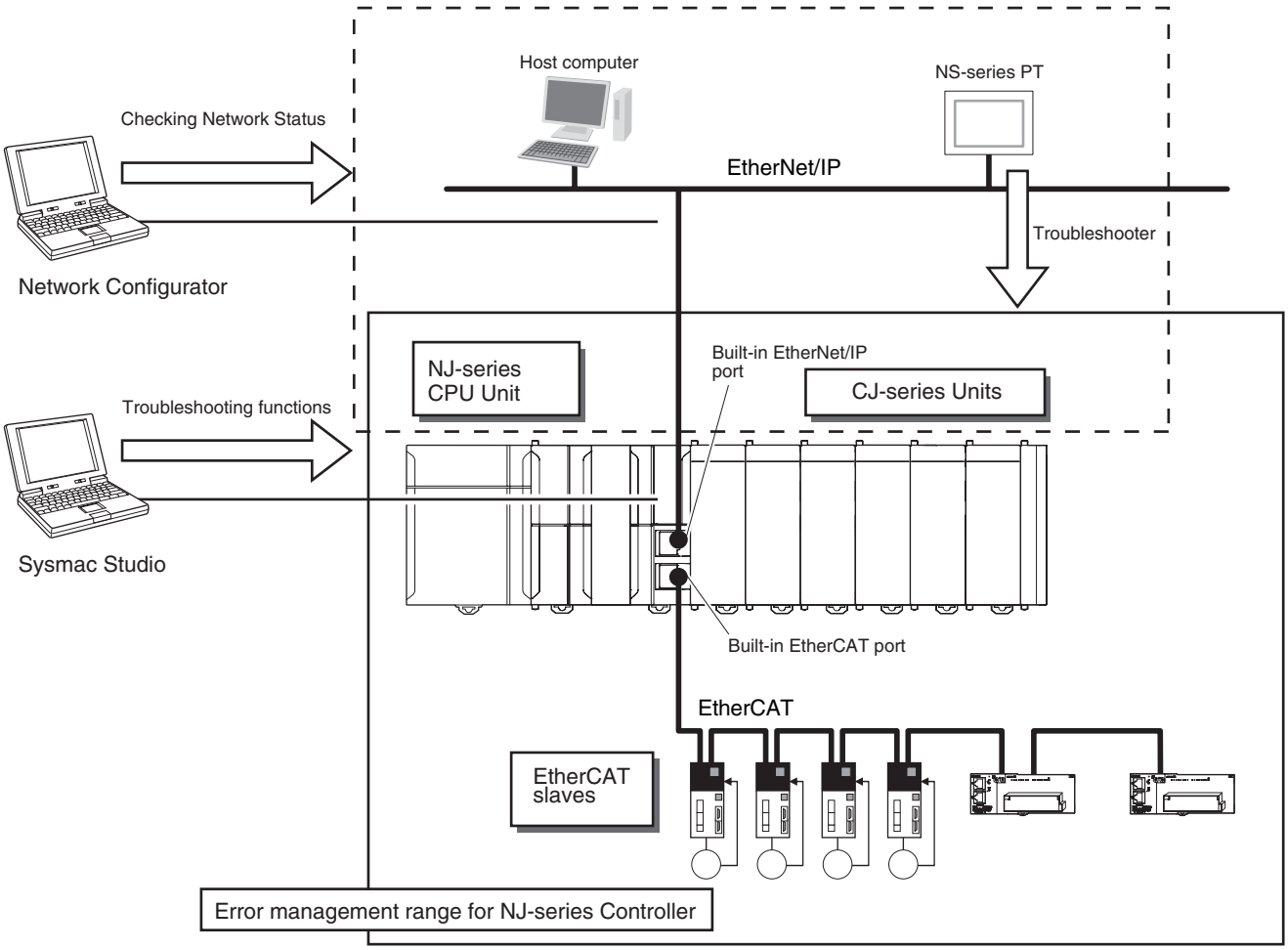
This section describes the items to check when errors occur in the EtherNet/IP Function Module. It includes error diagnosis and countermeasures for error indications, and error diagnosis and countermeasures for operating conditions.

14-1 Overview of Errors 14-2
14-1-1 How to Check for Errors 14-2
14-1-2 Errors Related to the EtherNet/IP Function Module 14-6
14-2 Troubleshooting 14-9
14-2-1 Error Table 14-9
14-2-2 Error Descriptions 14-14
14-2-3 Troubleshooting 14-37
14-3 Checking Status with the Network Configurator 14-38
14-3-1 The Network Configurator's Device Monitor Function 14-38
14-3-2 Connection Status Codes and Troubleshooting 14-45

14-1 Overview of Errors

You manage all of the errors that occur on the NJ-series Controller as events. The same methods are used for all events. This allows you to see what errors have occurred and find corrections for them with the same methods for the entire range of errors that is managed (i.e., CPU Unit, EtherCAT slaves,* and CJ-series Units).

* The EtherCAT slaves must support NJ-series error management.



You can use the troubleshooting functions of the Sysmac Studio or the Troubleshooter on an NS-series PT to quickly check for errors that have occurred and find corrections for them.

You can also use the Network Configurator to check the network status. Refer to *14-3 Checking Status with the Network Configurator* for the procedure to check network status.

This manual describes the errors that originate in the EtherNet/IP Function Module. Refer to the *NJ-series Troubleshooting Manual* (Cat. No. W503) for specific corrections when errors occur and for troubleshooting information on the entire NJ-series Controller.

14-1-1 How to Check for Errors

You can check to see if an error has occurred with the following methods.

Checking method	What you can check
Checking the indicators	You can check the error status of the CPU Unit and EtherNet/IP Function Module.
Checking with the troubleshooting function of Sysmac Studio	You can check for current Controller errors, a log of past Controller errors, error sources, error causes, and corrections.
Checking with the Troubleshooter of an NS-series PT	You can check for current Controller errors, a log of past Controller errors, error sources, error causes, and corrections.
Checking with instructions that read function module error status	You can check the highest-level status and highest-level event code in the current Controller errors.
Checking with system-defined variables	You can check the current Controller error status for each function module.
Checking communications status with the Network Configurator	You can check the communications status (e.g., tag data link connection status) for each device on the EtherNet/IP network.

This section describes the above checking methods.

Checking the Indicators

● CPU Unit Operating Status

You can use the PWR indicator on the Power Supply Unit and the RUN and ERROR indicators on the CPU Unit to determine the event level for an error. The following table shows the relationship between the Controller's indicators and the event level.

Indicators			CPU Unit operating status	Error confirmation with the Sysmac Studio or an NS-series PT
PWR	RUN	ERROR		
Not lit	Not lit	Not lit	Power Supply Error	Not possible: Refer to the *NJ-series Troubleshooting Manual* (Cat. No. W503).
Lit	Not lit	Not lit	CPU Unit Reset*1	
Lit	Flashing	Lit	Incorrect Power Supply Unit Connected	
Lit	Not lit	Lit	CPU Unit Watchdog Timer Error*2	
Lit	Not lit	Lit	Major fault level	Possible: Connect the Sysmac Studio or an NS-series PT and check the cause of and correction for the error in the troubleshooting functions of the Sysmac Studio or the Troubleshooter of the NS-series PT.
Lit	Lit	Flashing	Partial fault level*2	
Lit	Lit	Flashing	Minor fault level	
Lit	Lit	Not lit	Observation	
Lit	Lit	Not lit	Normal operation in RUN mode	---
Lit	Not lit	Not lit	Normal operation in PROGRAM mode*1	---
Lit	Flashing	Not lit	Normal operation in startup state	---

*1 If you can connect communications to the CPU Unit from the Sysmac Studio with a direct USB connection, the CPU Unit is in PROGRAM mode. If you cannot connect communications, the CPU Unit is being reset.*3

*2 If you can connect communications to the CPU Unit from the Sysmac Studio with a direct USB connection, a major fault level error has occurred. If you cannot connect communications, a watchdog timer error has occurred in the CPU Unit.*3

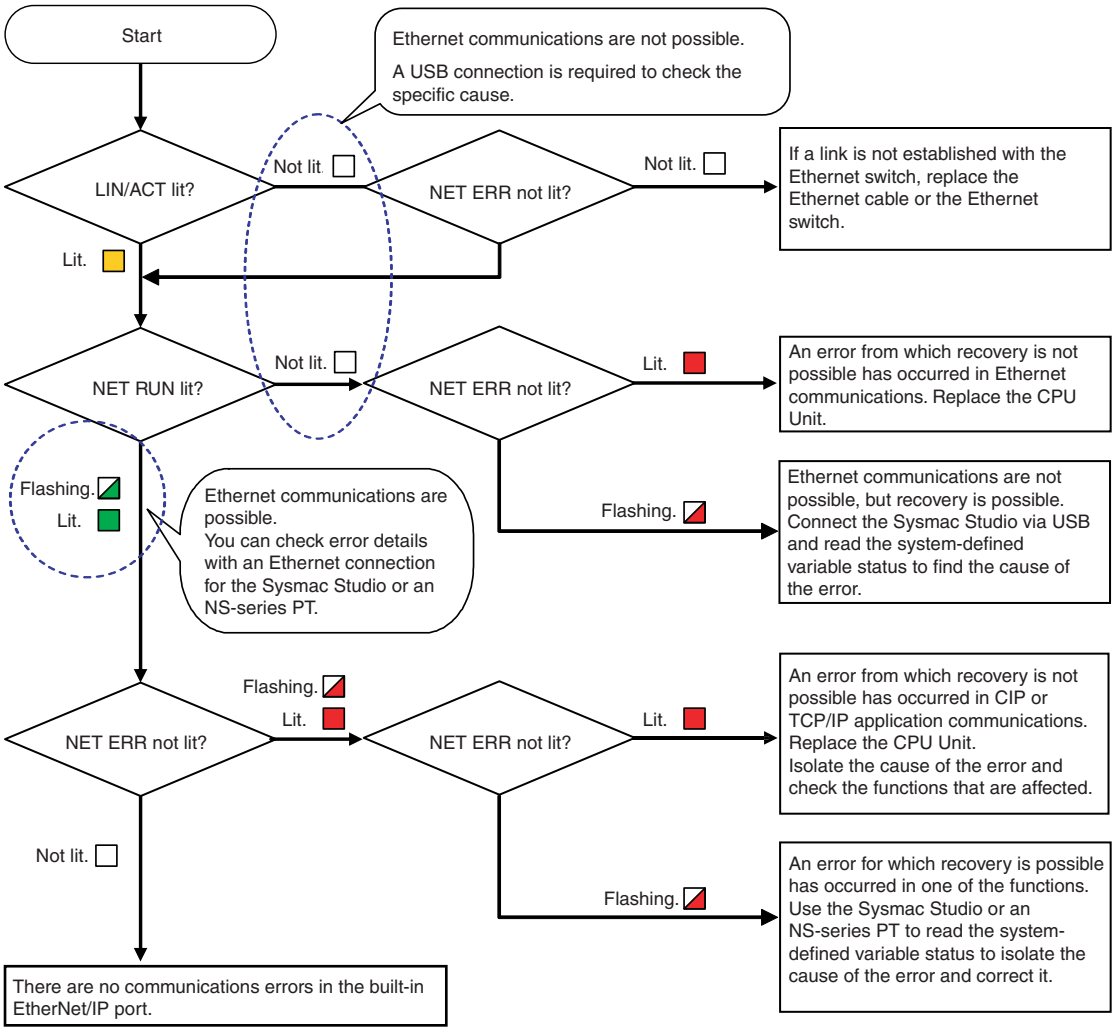
*3 If you cannot connect communications to the CPU Unit from the Sysmac Studio, it is also possible that the USB cable is faulty or that the network type on the Sysmac Studio is not set for a direct USB connection. Refer to the *NJ-series Troubleshooting Manual* (Cat. No. W503) if you cannot connect communications to the CPU Unit.

EtherNet/IP Function Module Error Status

If the EtherNet/IP NET ERR indicator is also lit in addition to the PWR, RUN, and ERROR indicators, then an error in the minor fault level or a higher level has occurred in the EtherNet/IP Function Module. The indicator lets you check the status given in the following table.

Indicators	Indicated status
EtherNet/IP NET ERR	EtherNet/IP Function Module Status • Lit: An error for which normal status cannot be recovered through user actions (i.e., errors for which you must replace the CPU Unit or contact your OMRON representative) has occurred. • Flashing: An error for which normal status can be recovered through user actions has occurred. • Not lit: There is no minor fault level or higher-level error.

Flowchart to Check Error Status on EtherNet/IP Function Module with Indicators



Checking with the Troubleshooting Function of Sysmac Studio

When an error occurs, you can connect the Sysmac Studio online to the Controller to check current Controller errors and the log of past Controller errors. You can also check the cause of the error and corrections.

Refer to the *NJ-series Troubleshooting Manual* (Cat. No. W503) for the procedures to check for errors with the Sysmac Studio.

Checking with the Troubleshooter of an NS-series PT

If you can connect communications between an NS-series PT and the Controller when an error occurs, you can check for current Controller errors and the log of past Controller errors. You can also check the cause of the error and corrections.

Refer to the *NJ-series Troubleshooting Manual* (Cat. No. W503) for the procedures to check for errors with an NS-series PT.

Checking with Instructions That Read Error Status

You can use instructions in the user program to check the error status of each function module. The following table gives the instruction that is used to get error information for the EtherNet/IP Function Module.

Instruction	Name	Outline of function
GetEIPError	Get EtherNet/IP Error Status	The GetEIPError instruction gets the highest level status (partial fault or minor fault) and highest level event code of the current Controller errors in the EtherNet/IP Function Module.

For details on the instructions that get error status, refer to the *NJ-series Instructions Reference Manual* (Cat. No. W502).

Checking with System-defined Variables

You can use system-defined variables to check for errors in the EtherNet/IP Master Function Module.

● Error Status Variables

You can check for errors in each function module of the NJ-series Controller with error status variables. The following variables show the error status of the EtherNet/IP Function Module.

Variable name	Data type	Meaning	Function
_EIP_ErrSta	WORD	Built-in EtherNet/IP Error	Gives the collective error status of all error status for the EtherNet/IP Error Function Module.
_EIP_PortErr	WORD	Communications Port Error	Gives the collective error status of all error status for the EtherNet/IP communications port.
_EIP_CipErr	WORD	CIP Communications Error	Gives the collective error status for CIP communications.
_EIP_TcpAppErr	WORD	TCP Application Communications Error	Gives the collective error status for TCP applications communications.

The meanings of the individual bits in the above error status variables are given below.

Bit	Name	Description	Value	Meaning
15	Master Detection*	This bit indicates whether the master detected an error in the slaves that it manages.	TRUE	Error
			FALSE	No error
14	Slave Summary*	Indicates whether there is an error at a level below the function module.	TRUE	Error
			FALSE	No error

Bit	Name	Description	Value	Meaning
8 to 13	Not used.			
7	Major Fault	Indicates if there is a major fault level error.	TRUE	Error
			FALSE	No error
6	Partial Fault	Indicates if there is a partial fault level error.	TRUE	Error
			FALSE	No error
5	Minor Fault	Indicates if there is a minor fault level error.	TRUE	Error
			FALSE	No error
4	Observation	Indicates if there is an observation level error.	TRUE	Error
			FALSE	No error
0 to 3	Not used.			

* This bit is not used in the error status variables for the EtherNet/IP Function Module.

● Other System-defined Variables Related to Errors

For the EtherNet/IP Function Module, there are other system-defined variables that show error status in addition to the error status variables. Refer to *3-1 System-defined Variables Related to the Built-in EtherNet/IP Port* for system-defined variables that are related to errors.

Checking Communications Status with the Network Configurator

You can use the Network Configurator to check the communications status (e.g., tag data link connection status) for each device on the EtherNet/IP network. Refer to *14-3 Checking Status with the Network Configurator*.

14-1-2 Errors Related to the EtherNet/IP Function Module

This section describes the errors that are related to the EtherNet/IP Function Module.

Classifications

There are the following three sources of errors in the EtherNet/IP Function Module.

Classification	Description
Communications port errors	If an error is detected in overall EtherNet/IP communications, the corresponding bit in the Communications Port Error status variable shows the error.
CIP communications errors	If an error that is related to the tag data links or CIP message communications is detected, the corresponding bit in the CIP Error Communications Error variable shows the error.
TCP application errors	If an error that is related to the FTP server, NTP, or SMNP client is detected, the corresponding bit in the TCP Application Communications Error variable shows the error.

* Other Ethernet communications are not affected.

Event Levels

This section describes the operation of the EtherNet/IP Function Module for each event level.

Event level of the error	Operation
Major fault	All NJ-series Controller control operations stop for errors in this event level. A major fault level error never occurs for the EtherNet/IP Function Module.
Partial fault	All control operations for one of the function modules in the NJ-series Controller stop for errors in this event level. If a partial fault level error occurs in the EtherNet/IP Function Module, all functions of the EtherNet/IP Function Module, such as axis operation, stop.
Minor fault	Some of the control operations for one of the function modules in the NJ-series Controller stop for errors in this event level. If a minor fault level error occurs for the EtherNet/IP Function Module, EtherNet/IP communications are possible, but control may be affected. User action is required.
Observation	Errors in the observation level do not affect NJ-series Controller control operations. Observations are reported in order to prevent them from developing into errors at the minor fault level or higher.
Information	Events that are classified as information provide information that do not indicate errors.

Event Levels for Built-in EtherNet/IP Errors

The event levels for errors that are related to built-in EtherNet/IP are given in the following table.

Source / Level	Communications port	CIP communications	TCP application
Major fault	• None	• None	• None
Partial fault	• Communications Controller Failure • MAC Address Error • IP Address Duplication Error • BOOTP Server Connection Error • Basic Ethernet Setting Error • TCP/IP Basic Setting Error • EtherNet/IP Processing Error	• None	• None

Source / Level	Communications port	CIP communications	TCP application
Minor fault	• TCP/IP Advanced Setting Error • DNS Server Connection Error	• Identity Error • Tag Data Link Setting Error • Tag Data Link Timeout • Tag Data Link Connection Failure • Tag Name Resolution Error	• FTP Server Setting Error • NTP Client Setting Error • SNMP Setting Error • NTP Server Connection Error
Observation	• Packet Discarded Due to Full Receive Buffer • Link OFF Detected • Variable Access Error	• None	• None
Information	• Link Detected • Restarting Ethernet Port • IP Address Fixed • BOOTP Client Started	• Tag Data Link Download Started • Tag Data Link Download Finished • Tag Data Link Stopped • Tag Data Link Started • Tag Data Link All Run	• FTP Server Started • NTP Client Started • SNMP Started

14-2 Troubleshooting

This section describes the errors that can occur and the corrections for them.

14-2-1 Error Table

The errors (i.e., events) that can occur in the EtherNet/IP Function Module are given on the following pages. Event levels are given as following in the tables:

Maj: Major fault level

Prt: Partial fault level

Min: Minor fault level

Obs: Observation

Info: Information

Refer to the *NJ-series Troubleshooting Manual* (Cat. No. W503) for all of the event codes that may occur in an NJ-series Controller.

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
10200000 hex	User Program/Controller Configurations and Setup Transfer Error	The user program or Controller Configurations and Setup were not transferred correctly.	• The user program or Controller Configurations and Setup are not correct because the power supply to the Controller was interrupted or communications with the Sysmac Studio were disconnected during a download of the user program or the Controller Configurations and Setup. • The user program or Controller Configurations and Setup are not correct because the power supply to the Controller was interrupted during online editing. • The user program or Controller Configurations and Setup are not correct because the power supply to the Controller was interrupted during a Clear All Memory operation. • Non-volatile memory failed.	√					page 14-15
04200000 hex	Communications Controller Failure	A hardware error was detected in the communications controller of the built-in EtherNet/IP port.	• Communications Controller hardware error		√				page 14-16
14200000 hex	MAC Address Error	The MAC address in non-volatile memory was not read correctly.	• Non-volatile memory failure		√				page 14-16
14220000 hex	EtherNet/IP Processing Error	A fatal error was detected in the EtherNet/IP Function Module.	• Hardware has failed.		√				page 14-17

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
34210000 hex	Basic Ethernet Setting Error	An error was detected in the Ethernet settings.	• Power was interrupted when a download was in progress for the Ethernet basic settings. • Memory error		√				page 14-17
34220000 hex	TCP/IP Basic Setting Error (Local Port IP Address)	An error was detected in the IP address settings.	• Power was interrupted when a download was in progress for the TCP/IP basic settings. • Memory error • The IP address acquired from BOOTP server is illegal.		√				page 14-18
84010000 hex	IP Address Duplication Error	The same IP address is used more than once.	• The IP address of the built-in EtherNet/IP port is also used as the IP address of another node.		√				page 14-19
84020000 hex	BOOTP Server Connection Error	Connection with the BOOTP server failed.	• Server is down. • An error occurred in the communications path. • The IP address acquired from BOOTP server is illegal.		√				page 14-20
14210000 hex	Identity Error	The CIP identity information in non-volatile memory was not read correctly.	• Non-volatile memory failure			√			page 14-20
34200000 hex	Tag Data Link Setting Error	An error was detected in the communications settings for tag data links.	• Power was interrupted when a download was in progress for the data link settings. • Memory error			√			page 14-21
34230000 hex	TCP/IP Advanced Setting Error (IP Router Table)	An error was detected in the hosts in the IP router table.	• Power was interrupted when a download was in progress for the TCP/IP advanced settings. • Memory error • There is a mistake in the IP router table settings or hosts settings.			√			page 14-22
34240000 hex	FTP Server Setting Error	An error was detected in the FTP server settings.	• Power was interrupted when a download was in progress for the FTP server settings. • Memory error			√			page 14-23
34250000 hex	NTP Client Setting Error	An error was detected in the NTP client settings.	• Power was interrupted when a download was in progress for the NTP client settings. • Memory error			√			page 14-24
34260000 hex	SNMP Setting Error	An error was detected in the SNMP agent/trap settings.	• Power was interrupted when a download was in progress for the SNMP agent/trap settings. • Memory error			√			page 14-25

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
34270000 hex	Tag Name Resolution Error	Resolution of a tag used in a tag data link failed.	• The size of the network-published variable is different from the tag settings. • The I/O direction set for a tag data link and the I/O direction of the Controller variable do not match. • There are no network-published variables for the Controller tag settings. • A variable in the Controller that is set for a tag data link has the Network Publish attribute set to Input but also has the Constant attribute.			√			page 14-26
84030000 hex	DNS Server Connection Error	Connection with the DNS server failed.	• Parameter error • Server is down. • An error occurred in the communications path.			√			page 14-27
84040000 hex	NTP Server Connection Error	Connection with the NTP server failed.	• Parameter error • Server is down. • An error occurred in the communications path.			√			page 14-27
84070000 hex	Tag Data Link Connection Failed	Establishing a tag data link connection failed.	• The tag data link connection information is not the same for the originator and target. • Insufficient connections			√			page 14-28
84080000 hex	Tag Data Link Timeout	A timeout occurred in a tag data link.	• The power supply to the target node is OFF. • Communications with the target node stop. • The Ethernet cable for EtherNet/IP is disconnected. • The Ethernet cable for EtherNet/IP is broken. • Noise			√			page 14-29
54E00000 hex	Variable Access Error	Accessing a tag variable that is used in a tag data link failed.	• An out-of-range value was written by an EtherNet/IP tag data link for a variable that specifies SUBRANGE.				√		page 14-29
84050000 hex	Packet Discarded Due to Full Reception Buffer	A packet was discarded.	• A network convergence occurred.				√		page 14-30

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	

84060000 hex	Link OFF Detected	The Ethernet link status turned OFF.	• An Ethernet cable is broken, disconnected, or loose.
			• The Ethernet switch's power supply is turned OFF.
			• Baud rate mismatch.
			• Noise
			• One of the following operations was performed.
			  • The Identify object was reset.
			  • Settings were downloaded from the Network Configurator and EtherNet/IP was restarted.
			  • Settings for EtherNet/IP were downloaded from the Sysmac Studio or the Memory All Clear operation was performed.
							√		page 14-31

94010000 hex	Tag Data Link Download Started	Changing the tag data link settings started.	• Changing the tag data link settings started.					√	page 14-31
94020000 hex	Tag Data Link Download Finished	Changing the tag data link settings finished.	• Changing the tag data link settings finished.					√	page 14-32
94030000 hex	Tag Data Link Stopped	Tag data links were stopped by Network Configurator or manipulation of a system-defined variable. Or, the data link table was downloaded from the Network Configurator again.	• Tag data links were stopped by Network Configurator or manipulation of a system-defined variable.					√	page 14-32
94040000 hex	Tag Data Link Started	Tag data links were started by Network Configurator or manipulation of a system-defined variable. Or, the data link table was downloaded from the Network Configurator again.	• Tag data links were started by Network Configurator or manipulation of a system-defined variable.					√	page 14-33
94050000 hex	Link Detected	Establishment of an Ethernet link was detected.	• Establishment of an Ethernet link was detected.					√	page 14-33
94060000 hex	Restarting Ethernet Port	The built-in EtherNet/IP port was restarted.	• The built-in EtherNet/IP port was restarted.					√	page 14-34
94070000 hex	Tag Data Link All Run	Tag data link connections to all nodes have been established.	• Tag data link connections to all target nodes have been established.					√	page 14-34
94080000 hex	IP Address Fixed	The correct IP address has been determined and Ethernet communications can start.	• The correct IP address has been determined and Ethernet communications can start.					√	page 14-35
94090000 hex	BOOTP Client Started	The BOOTP client started requesting an IP address.	• The BOOTP client started requesting an IP address.					√	page 14-35

Event code	Event name	Meaning	Assumed cause	Level					Reference
				Maj	Prt	Min	Obs	Info	
940A0000 hex	FTP Server Started	The FTP agent started normally.	• The FTP agent started normally.					√	page 14-35
940B0000 hex	NTP Client Started	The NTP client started normally and a request for the NTP server to obtain the time started.	• The NTP client started normally and a request for the NTP server to obtain the time started.					√	page 14-36
940C0000 hex	SNMP Started	The SNMP agent started normally.	• The SNMP agent started normally.					√	page 14-36

14-2-2 Error Descriptions

This section describes the information that is given for individual errors.

Controller Error Descriptions

The items that are used to describe individual errors (events) are described in the following copy of an error table.

Event name	Gives the name of the error.			Event code	Gives the code of the error.	
Meaning	Gives a short description of the error.					
Source	Gives the source of the error.		Source details	Gives details on the source of the error.	Detection timing	Tells when the error is detected.
Error attributes	Level	Tells the level of influence on control.*1	Recovery	Gives the recovery method.*2	Log category	Tells which log the error is saved in.*3
Effects	User program	Tells what will happen to execution of the user program.*4	Operation	Provides special information on the operation that results from the error.		
Indicators	Gives the status of the built-in EtherNet/IP port and built-in EtherCAT port indicators. Indicator status is given only for errors in the EtherCAT Master Function Module and the EtherNet/IP Function Module.					
System-defined variables	Variable		Data type		Name	
	Lists the variable names, data types, and meanings for system-defined variables that provide direct error notification, that are directly affected by the error, or that contain settings that cause the error.					
Cause and correction	Assumed cause		Correction		Prevention	
	Lists the possible causes, corrections, and preventive measures for the error.					
Attached information	This is the attached information that is displayed by the Sysmac Studio or an NS-series PT.					
Precautions/ Remarks	Provides precautions, restrictions, and supplemental information.					

*1 One of the following:
Major fault: Major fault level
Partial fault: Partial fault level
Minor fault: Minor fault level
Observation
Information

*2 One of the following:
Automatic recovery: Normal status is restored automatically when the cause of the error is removed.
Error reset: Normal status is restored when the error is reset after the cause of the error is removed.
Cycle the power supply: Normal status is restored when the power supply to the Controller is turned OFF and then back ON after the cause of the error is removed.
Controller reset: Normal status is restored when the Controller is reset after the cause of the error is removed.
Depends on cause: The recovery method depends on the cause of the error.

*3 One of the following:
System: System event log
Access: Access event log

*4 One of the following:
Continues: Execution of the user program will continue.
Stops: Execution of the user program stops.
Starts: Execution of the user program starts.

Error Descriptions

Event name	User Program/Controller Configurations and Setup Transfer Error			Event code	10200000 hex	
Meaning	The user program or Controller Configurations and Setup were not transferred correctly.					
Source	PLC Function Module Motion Control Function Module EtherCAT Master Function Module EtherNet/IP Function Module		Source details	None	Detection timing	At power ON or Controller reset
Error attributes	Level	Major fault	Recovery	Cycle the power supply or reset the Controller.	Log category	System
Effects	User program	Stops.	Operation	All outputs are stopped.		
System-defined variables	Variable		Data type		Name	
	None		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	The user program or Controller Configurations and Setup are not correct because the power supply to the Controller was interrupted or communications with the Sysmac Studio were disconnected during a download of the user program or the Controller Configurations and Setup.		Clear all of memory and then download the project from the Sysmac Studio. If attached information is registered, cycle the power supply to the Controller and then implement the above correction.		Do not turn OFF the power supply to the Controller or disconnect communications with the Sysmac Studio during a download of the user program or the Controller Configurations and Setup.	
	The user program or Controller Configurations and Setup are not correct because the power supply to the Controller was interrupted during online editing.				Do not interrupt the power supply to the Controller during online editing.	
	The user program or Controller Configurations and Setup are not correct because the power supply to the Controller was interrupted during a Clear All Memory operation.				Do not interrupt the power supply to the Controller during a Clear All Memory operation.	
	Non-volatile memory failed.		If the error persists even after you make the above correction, replace the CPU Unit.		None	
Attached information	Attached Information 1: Cause Details None: Power was interrupted or communications were disconnected during a download or power was interrupted during online editing. Downloading/Predownloading: For other causes, the timing of error occurrence (during download or during download preparations) is given.					
Precautions/ Remarks	None					

Event name	Communications Controller Failure		Event code	04200000 hex		
Meaning	A hardware error was detected in the communications controller of the built-in EtherNet/IP port.					
Source	EtherNet/IP Function Module		Source details	Communications port	Detection timing	Continuously
Error attributes	Level	Partial fault	Recovery	Cycle the power supply.	Log category	System
Effects	User program	Continues.	Operation	EtherNet/IP communications will not operate.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	OFF		Lights.		---	
System-defined variables	Variable		Data type		Name	
	_EIP_LanHwErr		BOOL		Communications Controller Error	
Cause and correction	Assumed cause		Correction		Prevention	
	Communications Controller hardware error		Replace the CPU Unit.		None	
Attached information	None					
Precautions/ Remarks	After the *_EIP_LanHwErr* system-defined variable changes to TRUE, it will not change to FALSE unless the power supply to the Controller is cycled.					

Event name	MAC Address Error		Event code	14200000 hex		
Meaning	The MAC address in non-volatile memory was not read correctly.					
Source	EtherNet/IP Function Module		Source details	Communications port	Detection timing	At power ON or Controller reset
Error attributes	Level	Partial fault	Recovery	Cycle the power supply.	Log category	System
Effects	User program	Continues.	Operation	EtherNet/IP communications will not operate.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	OFF		Lights.		---	
System-defined variables	Variable		Data type		Name	
	_EIP_MacAdrErr		BOOL		MAC Address Error	
Cause and correction	Assumed cause		Correction		Prevention	
	Non-volatile memory failure		Replace the CPU Unit.		None	
Attached information	None					
Precautions/ Remarks	After the *_EIP_MacAdrErr* system-defined variable changes to TRUE, it will not change to FALSE unless the power supply to the Controller is cycled.					

Event name	EtherNet/IP Processing Error			Event code	14220000 hex	
Meaning	A fatal error was detected in the EtherNet/IP Function Module.					
Source	EtherNet/IP Function Module		Source details	Communications port	Detection timing	Continuously
Error attributes	Level	Partial fault	Recovery	Cycle the power supply.	Log category	System
Effects	User program	Continues.	Operation	EtherNet/IP communications will not operate.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	OFF		Lights.		---	
System-defined variables	Variable		Data type		Name	
	None		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	Hardware has failed.		Replace the CPU Unit.		None	
Attached information	None					
Precautions/ Remarks	None					

Event name	Basic Ethernet Setting Error			Event code	34210000 hex	
Meaning	An error was detected in the Ethernet settings.					
Source	EtherNet/IP Function Module		Source details	Communications port	Detection timing	At power ON or Controller reset
Error attributes	Level	Partial fault	Recovery	Automatic recovery, cycle the power supply, or reset Controller	Log category	System
Effects	User program	Continues.	Operation	EtherNet/IP communications will not operate.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	OFF		Flashes at 1-s intervals.		---	
System-defined variables	Variable		Data type		Name	
	_EIP_EtnCfgErr		BOOL		Basic Ethernet Setting Error	
Cause and correction	Assumed cause		Correction		Prevention	
	Power was interrupted when a download was in progress for the Ethernet basic settings.		Cycle the power supply to the Controller or reset the Controller. If normal operation is restored by cycling the power supply to the Controller or resetting the Controller, a memory error has occurred and may occur again. Replace the CPU Unit. If operation is not restored even after you cycle the power supply or reset the Controller, perform either of the following operations and download the settings again. • Perform a Memory All Clear operation. • Identify the cause of error from those listed in the event log details and correct the settings. If operation is not recovered by the above, replace the CPU Unit.		Do not turn OFF the power supply to the Controller while a download is in progress for the Ethernet basic settings.
	Memory error				None	
Attached information	Attached information 1: Type of errors (01 hex: Non-volatile memory access error, 02 hex: Inconsistency in settings) Attached information 2: Error details (00 hex: Non-volatile memory access error, 11 hex: Incorrect baud rate setting, 12 hex: Unsupported baud rate)					
Precautions/ Remarks	The cause of error can be identified with the attached information.					

<table>
<tr><td>Event name</td><td colspan="2">TCP/IP Basic Setting Error (Local Port IP Address)</td><td>Event code</td><td colspan="2">34220000 hex</td></tr>
<tr><td>Meaning</td><td colspan="5">An error was detected in the IP address settings.</td></tr>
<tr><td>Source</td><td colspan="2">EtherNet/IP Function Module</td><td>Source details</td><td>Communications port</td><td>Detection timing</td><td>At power ON or Controller reset</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Partial fault</td><td>Recovery</td><td>Automatic recovery (after downloading the settings), cycle the power supply, or reset Controller</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">EtherNet/IP communications will not operate.</td></tr>
<tr><td rowspan="2">Indicators</td><td colspan="2">EtherNet/IP NET RUN</td><td colspan="2">EtherNet/IP NET ERR</td><td colspan="2">EtherNet/IP LINK/ACT</td></tr>
<tr><td colspan="2">OFF</td><td colspan="2">Flashes at 1-s intervals.</td><td colspan="2">---</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_EIP_IPAdrCfgErr</td><td colspan="2">BOOL</td><td colspan="2">Basic TCP/IP Setting Error</td></tr>
<tr><td rowspan="4">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Power was interrupted when a download was in progress for the TCP/IP basic settings.</td><td colspan="2" rowspan="2">Cycle the power supply to the Controller or reset the Controller. If normal operation is restored by cycling the power supply to the Controller or resetting the Controller, a memory error has occurred and may occur again. Replace the CPU Unit. If operation is not restored even after you cycle the power supply or reset the Controller, perform either of the following operations and download the settings again.
• Memory All Clear operation.
• Identify the cause of error from those listed in the event log details and correct the settings.
If operation is not recovered by the above, replace the CPU Unit.</td><td colspan="2">Do not turn OFF the power supply to the Controller while a download is in progress for the TCP/IP basic settings.</td></tr>
<tr><td colspan="2">Memory error</td><td colspan="2">None</td></tr>
<tr><td colspan="2">The IP address acquired from BOOTP server is illegal.</td><td colspan="2">Correct the IP address that was provided to this port by the BOOTP server so that it is within the range specified for an NJ-series Controller.</td><td colspan="2">Correct the IP address that was provided to this port by the BOOTP server so that it is within the range specified for an NJ-series Controller.</td></tr>
<tr><td>Attached information</td><td colspan="5">Attached information 1: Type of errors (01 hex: Non-volatile memory access error, 02 hex: Inconsistency in settings)
Attached information 2: Error details (01 hex: Non-volatile memory access error
When settings are inconsistent (11 hex: Illegal IP address, 12 hex: Illegal subnet mask, 13 hex: Illegal default gateway, 14 hex: Illegal priority DNS, 15 hex: Illegal exchange DNS, 16 hex: Illegal domain name)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="5">The cause of error can be identified with the attached information.</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="2">IP Address Duplication Error</td><td>Event code</td><td colspan="2">84010000 hex</td></tr>
<tr><td>Meaning</td><td colspan="5">The same IP address is used more than once.</td></tr>
<tr><td>Source</td><td colspan="2">EtherNet/IP Function Module</td><td>Source details</td><td>Communications port</td><td>Detection timing</td><td>After link is established</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Partial fault</td><td>Recovery</td><td>Cycle the power supply or reset the Controller.</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">EtherNet/IP communications will not operate. Packets addressed to the local IP address are discarded.</td></tr>
<tr><td rowspan="2">Indicators</td><td colspan="2">EtherNet/IP NET RUN</td><td colspan="2">EtherNet/IP NET ERR</td><td colspan="2">EtherNet/IP LINK/ACT</td></tr>
<tr><td colspan="2">OFF</td><td colspan="2">Flashes at 1-s intervals.</td><td colspan="2">---</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_EIP_IPAdrDupErr</td><td colspan="2">BOOL</td><td colspan="2">IP Address Duplication Error</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">The IP address of the built-in EtherNet/IP port is also used as the IP address of another node.</td><td colspan="2">Perform either of the following and then cycle the power supply to the Controller or reset the Controller.
• Check the IP addresses of other nodes and correct the IP address settings so that the same address is not used by more than one node.
• Remove the node that has the duplicate IP address from the network.</td><td colspan="2">Perform allocations so that IP addresses of nodes on the network are used for only one node.</td></tr>
<tr><td>Attached information</td><td colspan="5">Attached information 1: Duplicated IP address (example: C0A8FA01 hex = address 192.168.250.1)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="5">A duplicated address error occurs if an ARP is sent with the set IP address and there is an ARP response.</td></tr>
</table>

Event name	BOOTP Server Connection Error			Event code	84020000 hex	
Meaning	Connection with the BOOTP server failed.					
Source	EtherNet/IP Function Module		Source details	Communications port	Detection timing	At BOOTP operation
Error attributes	Level	Partial fault	Recovery	Automatic recovery	Log category	System
Effects	User program	Continues.	Operation	EtherNet/IP communications will not operate. Requests to the BOOTP server will continue until there is a response from the BOOTP server. Data refreshing with the PLC Function Module will continue. An IP address was not set for the EtherNet/IP port when it was supposed to be set from the BOOTP server.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	OFF		Flashes at 1-s intervals.		---	
System-defined variables	Variable		Data type		Name	
	_EIP_BootpErr		BOOL		BOOTP Server Error	
Cause and correction	Assumed cause		Correction		Prevention	
	Server is down.		Check if the server at the remote connection is operating normally and set it to operate normally if it is not.		Check to make sure that the server at the remote connection is operating normally.	
	An error occurred in the communications path.		Check the communications path to the server and take corrective measures if there are any problems.		None	
	The IP address acquired from BOOTP server is illegal.		Correct the IP address that was provided to this port by the BOOTP server so that it is within the range specified for an NJ-series Controller.		Correct the IP address that was provided to this port by the BOOTP server so that it is within the range specified for an NJ-series Controller.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Identity Error			Event code	14210000 hex	
Meaning	The CIP identity information in non-volatile memory was not read correctly.					
Source	EtherNet/IP Function Module		Source details	CIP	Detection timing	At power ON or Controller reset
Error attributes	Level	Minor fault	Recovery	Cycle the power supply.	Log category	System
Effects	User program	Continues.	Operation	EtherNet/IP communications will not operate.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	---		Lights.		---	
System-defined variables	Variable		Data type		Name	
	_EIP_IdentityErr		BOOL		Identify Information Error	
Cause and correction	Assumed cause		Correction		Prevention	
	Non-volatile memory failure		Replace the CPU Unit.		None	
Attached information	None					
Precautions/ Remarks	None					

<table>
<tr><td>Event name</td><td colspan="3">Tag Data Link Setting Error</td><td>Event code</td><td colspan="2">34200000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">An error was detected in the communications settings for tag data links.</td></tr>
<tr><td>Source</td><td colspan="2">EtherNet/IP Function Module</td><td>Source details</td><td>CIP</td><td>Detection timing</td><td>At power ON or Controller reset</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Automatic recovery (after downloading the tag data link settings), cycle the power supply, or reset Controller</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">Tag data link communications will not operate.</td></tr>
<tr><td rowspan="2">Indicators</td><td colspan="2">EtherNet/IP NET RUN</td><td colspan="2">EtherNet/IP NET ERR</td><td colspan="2">EtherNet/IP LINK/ACT</td></tr>
<tr><td colspan="2">Flashes at 1-s intervals.</td><td colspan="2">Flashes at 1-s intervals.</td><td colspan="2">---</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_EIP_TDLinkCfgErr</td><td colspan="2">BOOL</td><td colspan="2">Tag Data Link Setting Error</td></tr>
<tr><td rowspan="3">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Power was interrupted when a download was in progress for the data link settings.</td><td colspan="2" rowspan="2">Cycle the power supply to the Controller or reset the Controller. If normal operation is restored by cycling the power supply to the Controller or resetting the Controller, a memory error has occurred and may occur again. Replace the CPU Unit. If normal operation is not recovered by cycling the power supply or resetting the Controller, perform one of the following operations.
• Memory All Clear operation
• Download the tag data link settings again.
• Clear the tag data link settings.
If operation is not recovered by the above, replace the CPU Unit.</td><td colspan="2">Do not turn OFF the power supply to the Controller while a download is in progress for the tag data link settings.</td></tr>
<tr><td colspan="2">Memory error</td><td colspan="2">None</td></tr>
<tr><td>Attached information</td><td colspan="6">Attached information 1: Type of errors (01 hex: Non-volatile memory access error, 02 hex: Inconsistency in settings)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">None</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="2">TCP/IP Advanced Setting Error (IP Router Table)</td><td>Event code</td><td colspan="2">34230000 hex</td></tr>
<tr><td>Meaning</td><td colspan="5">An error was detected in the hosts in the IP router table.</td></tr>
<tr><td>Source</td><td colspan="2">EtherNet/IP Function Module</td><td>Source details</td><td>Communications port</td><td>Detection timing</td><td>At power ON or Controller reset</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Automatic recovery (after downloading the settings), cycle the power supply, or reset Controller</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">Communications cannot be performed with the host in the IP router table.</td></tr>
<tr><td rowspan="2">Indicators</td><td colspan="2">EtherNet/IP NET RUN</td><td colspan="2">EtherNet/IP NET ERR</td><td colspan="2">EtherNet/IP LINK/ACT</td></tr>
<tr><td colspan="2">---</td><td colspan="2">Flashes at 1-s intervals.</td><td colspan="2">---</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">_EIP_IPRTblErr</td><td colspan="2">BOOL</td><td colspan="2">TCP/IP Advanced Setting Error</td></tr>
<tr><td rowspan="4">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Power was interrupted when a download was in progress for the TCP/IP advanced settings.</td><td colspan="2" rowspan="2">Cycle the power supply to the Controller or reset the Controller. If normal operation is restored by cycling the power supply to the Controller or resetting the Controller, a memory error has occurred and may occur again. Replace the CPU Unit. If operation is not restored even after you cycle the power supply or reset the Controller, perform either of the following operations and download the settings again.
• Memory All Clear operation.
• Identify the cause of error from those listed in the event log details and correct the settings.
If operation is not recovered by the above, replace the CPU Unit.</td><td colspan="2">Do not turn OFF the power supply to the Controller while a download is in progress for the TCP/IP advanced settings.</td></tr>
<tr><td colspan="2">Memory error</td><td colspan="2">None</td></tr>
<tr><td colspan="2">There is a mistake in the IP router table settings or hosts settings.</td><td colspan="2">Correct the IP router table settings or hosts settings.</td><td colspan="2">Check that there are no mistakes in the IP router table settings or hosts settings.</td></tr>
<tr><td>Attached information</td><td colspan="5">Attached information 1: Type of errors (01 hex: Non-volatile memory access error, 02 hex: Inconsistency in settings)
Attached information 2: Error details (01 hex: Non-volatile memory access error)
When the settings are inconsistent (11 hex: Illegal IP router table settings, 12 hex: Illegal Hosts setting)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="5">The cause of error can be identified with the attached information.</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="2">FTP Server Setting Error</td><td>Event code</td><td colspan="2">34240000 hex</td></tr>
<tr><td>Meaning</td><td colspan="5">An error was detected in the FTP server settings.</td></tr>
<tr><td>Source</td><td colspan="2">EtherNet/IP Function Module</td><td>Source details</td><td>FTP</td><td>Detection timing</td><td>At power ON or Controller reset</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Automatic recovery (after downloading the FTP settings), cycle the power supply, or reset Controller</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">FTP will not operate.</td></tr>
<tr><td rowspan="2">Indicators</td><td colspan="2">EtherNet/IP NET RUN</td><td colspan="2">EtherNet/IP NET ERR</td><td colspan="2">EtherNet/IP LINK/ACT</td></tr>
<tr><td colspan="2">---</td><td colspan="2">Flashes at 1-s intervals.</td><td colspan="2">---</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">---</td><td colspan="2">---</td></tr>
<tr><td rowspan="3">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Power was interrupted when a download was in progress for the FTP server settings.</td><td colspan="2" rowspan="2">Cycle the power supply to the Controller or reset the Controller. If normal operation is restored by cycling the power supply to the Controller or resetting the Controller, a memory error has occurred and may occur again. Replace the CPU Unit. If operation is not restored even after you cycle the power supply or reset the Controller, perform either of the following operations and download the settings again.
• Memory All Clear operation.
• Identify the cause of error from those listed in the event log details and correct the settings.
If operation is not recovered by the above, replace the CPU Unit.</td><td colspan="2">Do not turn OFF the power supply to the Controller while a download is in progress for the FTP server settings.</td></tr>
<tr><td colspan="2">Memory error</td><td colspan="2">None</td></tr>
<tr><td>Attached information</td><td colspan="5">Attached information 1: Type of errors (01 hex: Non-volatile memory access error, 02 hex: Inconsistency in settings)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="5">The cause of error can be identified with the attached information.</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="2">NTP Client Setting Error</td><td>Event code</td><td colspan="2">34250000 hex</td></tr>
<tr><td>Meaning</td><td colspan="5">An error was detected in the NTP client settings.</td></tr>
<tr><td>Source</td><td colspan="2">EtherNet/IP Function Module</td><td>Source details</td><td>NTP</td><td>Detection timing</td><td>At power ON or Controller reset</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Automatic recovery (after downloading the NTP settings), cycle the power supply, or reset Controller</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">NTP operation stops.</td></tr>
<tr><td rowspan="2">Indicators</td><td colspan="2">EtherNet/IP NET RUN</td><td colspan="2">EtherNet/IP NET ERR</td><td colspan="2">EtherNet/IP LINK/ACT</td></tr>
<tr><td colspan="2">---</td><td colspan="2">Flashes at 1-s intervals.</td><td colspan="2">---</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">---</td><td colspan="2">---</td></tr>
<tr><td rowspan="3">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Power was interrupted when a download was in progress for the NTP client settings.</td><td colspan="2" rowspan="2">Cycle the power supply to the Controller or reset the Controller. If normal operation is restored by cycling the power supply to the Controller or resetting the Controller, a memory error has occurred and may occur again. Replace the CPU Unit. If operation is not restored even after you cycle the power supply or reset the Controller, perform either of the following operations and download the settings again.
• Memory All Clear operation.
• Identify the cause of error from those listed in the event log details and correct the settings.
If operation is not recovered by the above, replace the CPU Unit.</td><td colspan="2">Do not turn OFF the power supply to the Controller while a download is in progress for the NTP client settings.</td></tr>
<tr><td colspan="2">Memory error</td><td colspan="2">None</td></tr>
<tr><td>Attached information</td><td colspan="5">Attached information 1: Type of errors (01 hex: Non-volatile memory access error, 02 hex: Inconsistency in settings)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="5">The cause of error can be identified with the attached information.</td></tr>
</table>

<table>
<tr><td>Event name</td><td colspan="3">SNMP Setting Error</td><td>Event code</td><td colspan="2">34260000 hex</td></tr>
<tr><td>Meaning</td><td colspan="6">An error was detected in the SNMP agent/trap settings.</td></tr>
<tr><td>Source</td><td colspan="2">EtherNet/IP Function Module</td><td>Source details</td><td>SNMP</td><td>Detection timing</td><td>At power ON or Controller reset</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Minor fault</td><td>Recovery</td><td>Automatic recovery (after downloading the SNMP settings), cycle the power supply, or reset Controller</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">SNMP operation stops.</td></tr>
<tr><td rowspan="2">Indicators</td><td colspan="2">EtherNet/IP NET RUN</td><td colspan="2">EtherNet/IP NET ERR</td><td colspan="2">EtherNet/IP LINK/ACT</td></tr>
<tr><td colspan="2">---</td><td colspan="2">Flashes at 1-s intervals.</td><td colspan="2">---</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">---</td><td colspan="2">---</td></tr>
<tr><td rowspan="3">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">Power was interrupted when a download was in progress for the SNMP agent/trap settings.</td><td colspan="2" rowspan="2">Cycle the power supply to the Controller or reset the Controller. If normal operation is restored by cycling the power supply to the Controller or resetting the Controller, a memory error has occurred and may occur again. Replace the CPU Unit. If operation is not restored even after you cycle the power supply or reset the Controller, perform either of the following operations and download the settings again.
• Perform a Memory All Clear operation.
• Identify the cause of error from those listed in the event log details and correct the settings.
If operation is not recovered by the above, replace the CPU Unit.</td><td colspan="2">Do not turn OFF the power supply to the Controller while a download is in progress for the SNMP agent/trap settings.</td></tr>
<tr><td colspan="2">Memory error</td><td colspan="2">None</td></tr>
<tr><td>Attached information</td><td colspan="6">Attached information 1: Type of errors (01 hex: Non-volatile memory access error, 02 hex: Inconsistency in settings)
Attached information 2: Error location, when there is an inconsistency in the settings (01 hex: SNMP agent settings, 02 hex: SNMP trap settings)</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">The cause of error can be identified with the attached information.</td></tr>
</table>

Event name	Tag Name Resolution Error			Event code	34270000 hex	
Meaning	Resolution of a tag used in a tag data link failed.					
Source	EtherNet/IP Function Module		Source details	CIP	Detection timing	At power ON, at Controller reset, when variables are changed from the Sysmac Studio, or when the data link table is changed from the Network Configurator
Error attributes	Level	Minor fault	Recovery	Automatic recovery (after downloading the tag settings)	Log category	System
Effects	User program	Continues.	Operation	Data links will not operate for unresolved tags. Data links for other tags will operate.		

Indicators	EtherNet/IP NET RUN	EtherNet/IP NET ERR	EtherNet/IP LINK/ACT
	Flashes at 1-s intervals.	Flashes at 1-s intervals.	---

System-defined variables	Variable	Data type	Name
	_EIP_TagAdrErr	BOOL	Tag Resolution Error

Cause and correction	Assumed cause	Correction	Prevention
	The size of the network-published variable is different from the tag settings.	Correct the sizes in the tag settings to match the network-published variables.	Set the sizes in the tag settings to match the network-published variables.
	The I/O direction set for a tag data link and the I/O direction of the Controller variable do not match.	Correct the tag settings or the settings of the Controller variables so that the I/O direction for the tag data links match the I/O direction of the Controller variable.	Set the tag settings or the settings of the Controller variables so that the I/O directions for the tag data links match the I/O directions of the Controller variables.
	There are no network-published variables for the Controller tag settings.	Correct the tag settings so that existing network-published variables are set for the tags.	Set the tag settings so that existing network-published variables are set for the tags.
	A variable in the Controller that is set for a tag data link has the Network Publish attribute set to Input but also has the Constant attribute.	Remove the Constant attribute from the Controller variable that has the Network Publish attribute set to Input.	Do not set the Constant attribute for a Controller variable that has the Network Publish attribute set to Input.

Attached information	None
Precautions/ Remarks	None

Event name	DNS Server Connection Error			Event code	84030000 hex	
Meaning	Connection with the DNS server failed.					
Source	EtherNet/IP Function Module		Source details	Communications port	Detection timing	At DNS operation
Error attributes	Level	Minor fault	Recovery	Automatic recovery (after downloading the DNS settings)	Log category	System
Effects	User program	Continues.	Operation	Communications using DNS stop.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	---		Flashes at 1-s intervals.		---	
System-defined variables	Variable		Data type		Name	
	_EIP_DNSSrvErr		BOOL		DNS Server Connection Error	
Cause and correction	Assumed cause		Correction		Prevention	
	Parameter error		If there is a mistake with the specifications of the connected server, correct the server specifications and download them again.		Make sure that the connected server is specified correctly.	
	Server is down.		Check if the server at the remote connection is operating normally and set it to operate normally if it is not.		Check to make sure that the server at the remote connection is operating normally.	
	An error occurred in the communications path.		Check the communications path to the server and take corrective measures if there are any problems.		None	
Attached information	None					
Precautions/ Remarks	If *TCP Server Run* is recorded in the event log after the correction is made, then the CPU Unit is correctly connected to the DNS server.					

Event name	NTP Server Connection Error			Event code	84040000 hex	
Meaning	Connection with the NTP server failed.					
Source	EtherNet/IP Function Module		Source details	NTP	Detection timing	At NTP operation
Error attributes	Level	Minor fault	Recovery	Automatic recovery (after downloading the NTP settings)	Log category	System
Effects	User program	Continues.	Operation	Time cannot be acquired from NTP.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	---		Flashes at 1-s intervals.		---	
System-defined variables	Variable		Data type		Name	
	_EIP_NTPSrvErr		BOOL		NTP Server Connection Error	
Cause and correction	Assumed cause		Correction		Prevention	
	Parameter error		If there is a mistake with the specifications of the connected server, correct the server specifications and download them again.		Make sure that the connected server is specified correctly.	
	Server is down.		Check if the server at the remote connection is operating normally and set it to operate normally if it is not.		Check to make sure that the server at the remote connection is operating normally.	
	An error occurred in the communications path.		Check the communications path to the server and take corrective measures if there are any problems.		None	
Attached information	None					
Precautions/ Remarks	If *TCP Server Run* is recorded in the event log after the correction is made, then the CPU Unit is correctly connected to the DNS server.					

Event name	Tag Data Link Connection Failed			Event code	84070000 hex	
Meaning	Establishing a tag data link connection failed.					
Source	EtherNet/IP Function Module		Source details	CIP	Detection timing	When establishing tag data link connection
Error attributes	Level	Minor fault	Recovery	Automatic recovery	Log category	System
Effects	User program	Continues.	Operation	Data links will not operate for connections that could not be established. Data links for other connections will operate.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	Flashes at 1-s intervals.		Flashes at 1-s intervals.		---	
System-defined variables	Variable		Data type		Name	
	_EIP_TDLinkOpnErr		BOOL		Tag Data Link Connection Failure	
Cause and correction	Assumed cause		Correction		Prevention	
	The tag data link connection information is not the same for the originator and target.		Correct the tag data link connection information, and then download the device parameters from the Network Configurator.		Before you use the tag data links, make sure that the tag data link connection information in the originator and target are suitable.	
	Insufficient connections		Reduce the number of class-3 messages.		Reduce the number of data links and class-3 messages that are used.	
Attached information	Attached information 1: Target node IP address (example: C0A8FA01 hex = address 192.168.250.1)					
Precautions/ Remarks	You can investigate a detailed cause from the connection status. Refer to *14-3-2 Connection Status Codes and Troubleshooting*.					

Event name	Tag Data Link Timeout			Event code	84080000 hex	
Meaning	A timeout occurred in a tag data link.					
Source	EtherNet/IP Function Module		Source details	CIP	Detection timing	Continuously after starting tag data link communications
Error attributes	Level	Minor fault	Recovery	Automatic recovery	Log category	System
Effects	User program	Continues.	Operation	The relevant data link connection will stop. Reconnection processing is periodically repeated for the tag data link error target.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	Flashes at 1-s intervals.		Flashes at 1-s intervals.		---	
System-defined variables	Variable		Data type		Name	
	_EIP_TDLinkErr		BOOL		Tag Data Link Communications Error	
Cause and correction	Assumed cause		Correction		Prevention	
	The power supply to the target node is OFF.		Check the status of the target node and start it normally.		Use the tag data link after you confirm that the target node is normal.	
	Communications with the target node stop.					
	The Ethernet cable for EtherNet/IP is disconnected.		Reconnect the connector and make sure it is mated correctly.		Connect the connector securely.	
	The Ethernet cable for EtherNet/IP is broken.		Replace the Ethernet cable.		None	
	Noise		Implement noise countermeasures if there is excessive noise.		Implement noise countermeasures if there is excessive noise.	
Attached information	Attached information 1: Connection instance No. (0 to 255) Attached information 2: Target node IP address (example: C0A8FA01 hex = address 192.168.250.1)					
Precautions/ Remarks	The following cases are not included in this error. • Connections as a target • Connection timeouts due to a Link OFF Error with the Ethernet switch.					

Event name	Variable Access Error			Event code	54E00000 hex	
Meaning	Accessing a tag variable that is used in a tag data link failed.					
Source	EtherNet/IP Function Module		Source details	Communications port	Detection timing	At variable read/write
Error attributes	Level	Observation	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	Not affected.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	---		---		---	
System-defined variables	Variable		Data type		Name	
	None		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	An out-of-range value was written by an EtherNet/IP tag data link for a variable that specifies SUBRANGE.		Correct the value that is written to the variable that specifies SUBRANGE so that the value is in range.		Write a value that is in range for the variable that specifies SUBRANGE.	
Attached information	None					
Precautions/ Remarks	Reading/writing the variable will not end normally.					

<table>
<tr><td>Event name</td><td colspan="2">Packet Discarded Due to Full Reception Buffer</td><td>Event code</td><td colspan="2">84050000 hex</td></tr>
<tr><td>Meaning</td><td colspan="5">A packet was discarded.</td></tr>
<tr><td>Source</td><td colspan="2">EtherNet/IP Function Module</td><td>Source details</td><td>Communications port</td><td>Detection timing</td><td>After link is established</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Observation</td><td>Recovery</td><td>---</td><td>Log category</td><td>System</td></tr>
<tr><td>Effects</td><td>User program</td><td>Continues.</td><td>Operation</td><td colspan="3">Not affected.</td></tr>
<tr><td rowspan="2">Indicators</td><td colspan="2">EtherNet/IP NET RUN</td><td colspan="2">EtherNet/IP NET ERR</td><td colspan="2">EtherNet/IP LINK/ACT</td></tr>
<tr><td colspan="2">---</td><td colspan="2">---</td><td colspan="2">---</td></tr>
<tr><td rowspan="2">System-defined variables</td><td colspan="2">Variable</td><td colspan="2">Data type</td><td colspan="2">Name</td></tr>
<tr><td colspan="2">None</td><td colspan="2">---</td><td colspan="2">---</td></tr>
<tr><td rowspan="2">Cause and correction</td><td colspan="2">Assumed cause</td><td colspan="2">Correction</td><td colspan="2">Prevention</td></tr>
<tr><td colspan="2">A network convergence occurred.</td><td colspan="2">The load on the network is too high. Check whether there are nodes that send unnecessary broadcast frames on the network and remove them. After that, check that the received number of frames has reduced in the network statistical information.</td><td colspan="2">Make sure that unnecessary broadcast frames are not sent on the network. Do not connect the Ethernet cable in a loop.</td></tr>
<tr><td>Attached information</td><td colspan="5">None</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="5">None</td></tr>
</table>

Event name	Link OFF Detected			Event code	84060000 hex	
Meaning	The Ethernet link status turned OFF.					
Source	EtherNet/IP Function Module		Source details	Communications port	Detection timing	After link is established
Error attributes	Level	Observation	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	EtherNet/IP communications will not operate.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	---		---		OFF	
System-defined variables	Variable		Data type		Name	
	None		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	An Ethernet cable is broken, disconnected, or loose.		Connect the Ethernet cable securely. If the cable is broken, replace it.		Connect the Ethernet cable securely. Check the cable to make sure that it is not disconnected.	
	The Ethernet switch power supply is turned OFF.		Turn ON the power supply to the Ethernet switch. Replace the Ethernet switch if it fails.		Do not turn OFF the power supply to the Ethernet switch.	
	Baud rate mismatch.		Correct the settings so that the same baud rate is used as for the remote communications nodes.		Set the same baud rate as for the remote communications nodes.	
	Noise		Implement noise countermeasures if there is excessive noise.		Implement noise countermeasures.	
	One of the following operations was performed. • The Identify object was reset. • Settings were downloaded from the Network Configurator and EtherNet/IP was restarted. • Settings for EtherNet/IP were downloaded from the Sysmac Studio or the Memory All Clear operation was performed.		None This error occurs when the operations on the left are preformed.		None This error occurs when the operations on the left are preformed.	
Attached information	None					
Precautions/ Remarks	None					

Event name	Tag Data Link Download Started			Event code	94010000 hex	
Meaning	Changing the tag data link settings started.					
Source	EtherNet/IP Function Module		Source details	CIP	Detection timing	At user operation
Error attributes	Level	Information	Recovery	---	Log category	Access
Effects	User program	Continues.	Operation	Not affected.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	Flashes at 1-s intervals.		---		---	
System-defined variables	Variable		Data type		Name	
	None		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	Changing the tag data link settings started.		---		---	
Attached information	Attached information 1: Controller status (01 hex: PROGRAM mode, 02 hex: RUN mode)					
Precautions/ Remarks	None					

Event name	Tag Data Link Download Finished			Event code	94020000 hex	
Meaning	Changing the tag data link settings finished.					
Source	EtherNet/IP Function Module		Source details	CIP	Detection timing	At user operation
Error attributes	Level	Information	Recovery	---	Log category	Access
Effects	User program	Continues.	Operation	Not affected.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	Flashes at 1-s intervals.		---		---	
System-defined variables	Variable		Data type		Name	
	None		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	Changing the tag data link settings finished.		---		---	
Attached information	Attached information 1: Controller status (01 hex: PROGRAM mode, 02 hex: RUN mode)					
Precautions/ Remarks	None					

Event name	Tag Data Link Stopped			Event code	94030000 hex	
Meaning	Tag data links were stopped by Network Configurator or manipulation of a system-defined variable. Or, the data link table was downloaded from the Network Configurator again.					
Source	EtherNet/IP Function Module		Source details	CIP	Detection timing	At user operation
Error attributes	Level	Information	Recovery	---	Log category	Access
Effects	User program	Continues.	Operation	Not affected.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	Flashes at 1-s intervals.		---		---	
System-defined variables	Variable		Data type		Name	
	_EIP_TDLinkStopCmd		BOOL		Tag Data Link Communications Stop Switch	
Cause and correction	Assumed cause		Correction		Prevention	
	Tag data links were stopped by Network Configurator or manipulation of a system-defined variable.		---		---	
Attached information	Attached information 1: Controller status		01 hex: PROGRAM mode, 02 hex: RUN mode			
	Attached information 2: Operation method		01 hex: Operation from the Network Configurator, 02 hex: Operation with a system-defined variable			
Precautions/ Remarks	None					

Event name	Tag Data Link Started			Event code	94040000 hex	
Meaning	Tag data links were started by Network Configurator or manipulation of a system-defined variable. Or, the data link table was downloaded from the Network Configurator again.					
Source	EtherNet/IP Function Module		Source details	CIP	Detection timing	At user operation
Error attributes	Level	Information	Recovery	---	Log category	Access
Effects	User program	Continues.	Operation	Not affected.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	Flashes at 1-s intervals.		---		---	
System-defined variables	Variable		Data type		Name	
	_EIP_TDLinkStartCmd		BOOL		Tag Data Link Communications Start Switch	
Cause and correction	Assumed cause		Correction		Prevention	
	Tag data links were started by Network Configurator or manipulation of a system-defined variable.		---		---	
Attached information	Attached information 1: Controller status		01 hex: PROGRAM mode, 02 hex: RUN mode			
	Attached information 2: Operation method		01 hex: Operation from the Network Configurator, 02 hex: Operation with system-defined variable			
Precautions/ Remarks	None					

Event name	Link Detected			Event code	94050000 hex	
Meaning	Establishment of an Ethernet link was detected.					
Source	EtherNet/IP Function Module		Source details	Communications port	Detection timing	When establishing link
Error attributes	Level	Information	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	Not affected.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	---		---		Lights.	
System-defined variables	Variable		Data type		Name	
	None		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	Establishment of an Ethernet link was detected.		---		---	
Attached information	None					
Precautions/ Remarks	None					

Event name	Restarting Ethernet Port			Event code	94060000 hex	
Meaning	The built-in EtherNet/IP port was restarted.					
Source	EtherNet/IP Function Module		Source details	Communications port	Detection timing	At user operation
Error attributes	Level	Information	Recovery	---	Log category	Access
Effects	User program	Continues.	Operation	Not affected.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	---		---		---	
System-defined variables	Variable		Data type		Name	
	None		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	The built-in EtherNet/IP port was restarted.		---		---	
Attached information	None					
Precautions/ Remarks	None					

Event name	Tag Data Link All Run			Event code	94070000 hex	
Meaning	Tag data link connections to all nodes have been established.					
Source	EtherNet/IP Function Module		Source details	CIP	Detection timing	When establishing tag data link connection
Error attributes	Level	Information	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	Not affected.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	Lights.		---		---	
System-defined variables	Variable		Data type		Name	
	_EIP_TDLinkAllRunSta		BOOL		Tag Data Link All Run Status	
Cause and correction	Assumed cause		Correction		Prevention	
	Tag data link connections to all target nodes have been established.		---		---	
Attached information	None					
Precautions/ Remarks	None					

Event name	IP Address Fixed			Event code	94080000 hex	
Meaning	The correct IP address has been determined and Ethernet communications can start.					
Source	EtherNet/IP Function Module		Source details	Communications port	Detection timing	At power ON or Controller reset
Error attributes	Level	Information	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	Not affected.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	Lights.		---		---	
System-defined variables	Variable		Data type		Name	
	None		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	The correct IP address has been determined and Ethernet communications can start.		---		---	
Attached information	Attached Information 1: IP address (example: C0A8FA01 hex = address 192.168.250.1)					
Precautions/ Remarks	None					

Event name	BOOTP Client Started			Event code	94090000 hex	
Meaning	The BOOTP client started requesting an IP address.					
Source	EtherNet/IP Function Module		Source details	Communications port	Detection timing	At power ON or Controller reset
Error attributes	Level	Information	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	Not affected.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	---		---		---	
System-defined variables	Variable		Data type		Name	
	None		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	The BOOTP client started requesting an IP address.		---		---	
Attached information	None					
Precautions/ Remarks	None					

Event name	FTP Server Started			Event code	940A0000 hex	
Meaning	The FTP agent started normally.					
Source	EtherNet/IP Function Module		Source details	FTP	Detection timing	At power ON or Controller reset
Error attributes	Level	Information	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	Not affected.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	---		---		---	
System-defined variables	Variable		Data type		Name	
	None		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	The FTP agent started normally.		---		---	
Attached information	None					
Precautions/ Remarks	None					

Event name	NTP Client Started			Event code	940B0000 hex	
Meaning	The NTP client started normally and a request for the NTP server to obtain the time started.					
Source	EtherNet/IP Function Module		Source details	NTP	Detection timing	At power ON or Controller reset
Error attributes	Level	Information	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	Not affected.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	---		---		---	
System-defined variables	Variable		Data type		Name	
	None		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	The NTP client started normally and a request for the NTP server to obtain the time started.		---		---	
Attached information	None					
Precautions/ Remarks	None					

Event name	SNMP Started			Event code	940C0000 hex	
Meaning	The SNMP agent started normally.					
Source	EtherNet/IP Function Module		Source details	SNMP	Detection timing	At power ON or Controller reset
Error attributes	Level	Information	Recovery	---	Log category	System
Effects	User program	Continues.	Operation	Not affected.		
Indicators	EtherNet/IP NET RUN		EtherNet/IP NET ERR		EtherNet/IP LINK/ACT	
	---		---		---	
System-defined variables	Variable		Data type		Name	
	None		---		---	
Cause and correction	Assumed cause		Correction		Prevention	
	The SNMP agent started normally.		---		---	
Attached information	None					
Precautions/ Remarks	None					

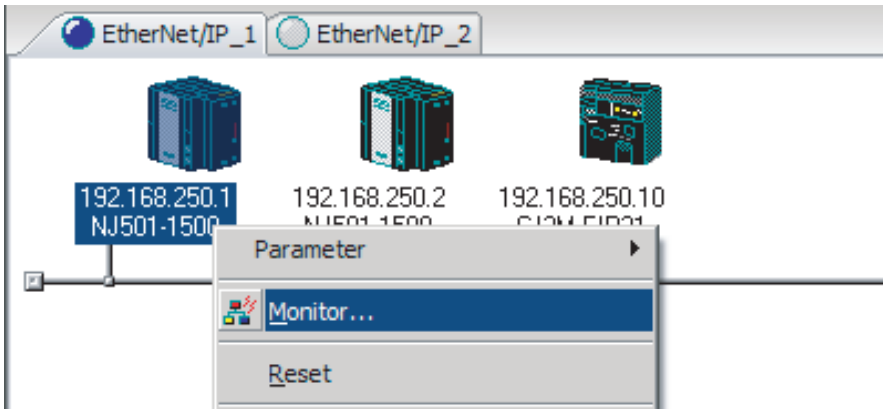
14-2-3 Troubleshooting

Problem	Correction
Tag data is not concurrent.	Check the following items and correct the user program. • Data concurrency is maintained for each connection between the CPU Unit and the built-in EtherNet/IP port. To maintain data concurrency for tag data links, set a refreshing task for the network variables that are assigned to tags. Refer to *7-1-7 Concurrency of Tag Data Link Data* for details. • Refer to the product manuals for products from other manufacturers.
At startup, some of the receive data is FALSE when it should be TRUE.	• If the user program uses receive data, make sure that the All Tag Data Link Communications Status in communications status 1 or the Controller Operating Mode for the target node is TRUE before you use the receive data. To use operation information from the Controller, use Controller status in the tag sets on both the sending and receiving nodes. • If the Fault Action setting of the output (produce) tag is enabled, The output (produce) data changes to FALSE when a fatal error occurs in the CPU Unit. Check the error status at the output (producing) Controller.
Tag data link communications are not stable.	• Use a 100 Mbps Ethernet switch if 100 Mbps is set or if you are using a 10 Mbps or 100 Mbps repeater hub. The performance of the tag data links assumes that an Ethernet switch is used to achieve a 1,000-pps bandwidth for full-duplex, 100-Mbps auto-negotiation communications. • Refer to *14-3 Checking Status with the Network Configurator* to access the error counters and discarded packed counters on the Ethernet Information Tab Page. Use the information to check for noise on the communications path, non-standard cables, damaged cables/connectors, unexpectedly high communications traffic, and incorrect loops in connections between Ethernet switches. • Contact the Ethernet switch manufacturer if there are problems with the transfer capacity of the Ethernet switches in the communications path. If Ethernet switches are cascaded, the load may be concentrated on the middle Ethernet switches. Change the network configuration so that the load is not concentrated. • Also, refer to *14-3 Checking Status with the Network Configurator* and use the connection status on the Connections Tab Page to remove the cause of the error according to information in *14-3-2 Connection Status Codes and Troubleshooting*.

14-3 Checking Status with the Network Configurator

14-3-1 The Network Configurator's Device Monitor Function

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

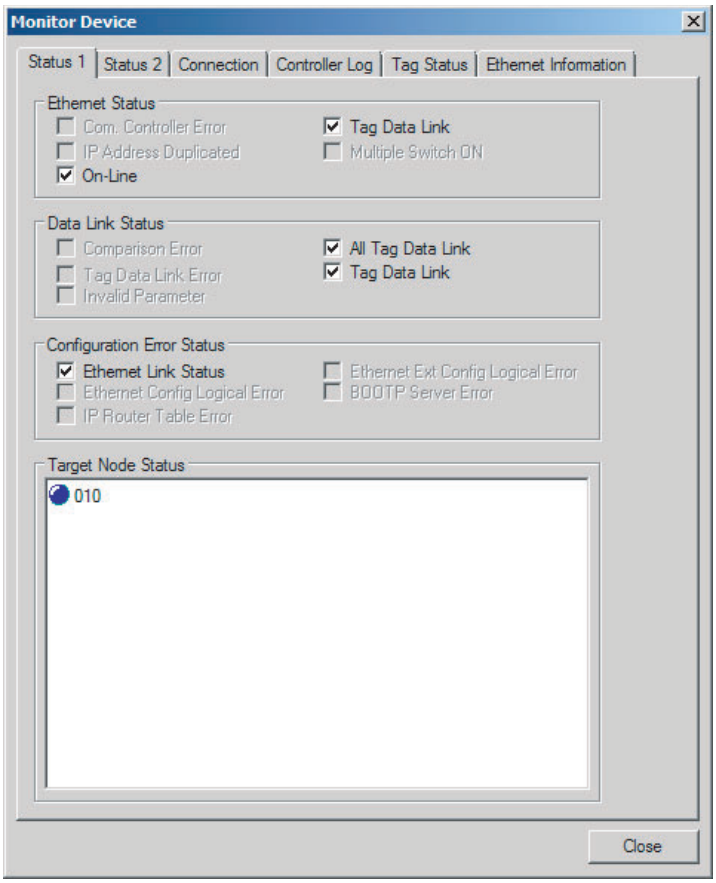
If a communications error occurs during monitoring, the dialog box will continue to show the last information that was collected. To start monitoring again, close the Monitor Device Dialog Box, and then open the dialog box again.

Status 1 Tab Page

The following check boxes are displayed for the status. If a check box is selected, the status is TRUE.

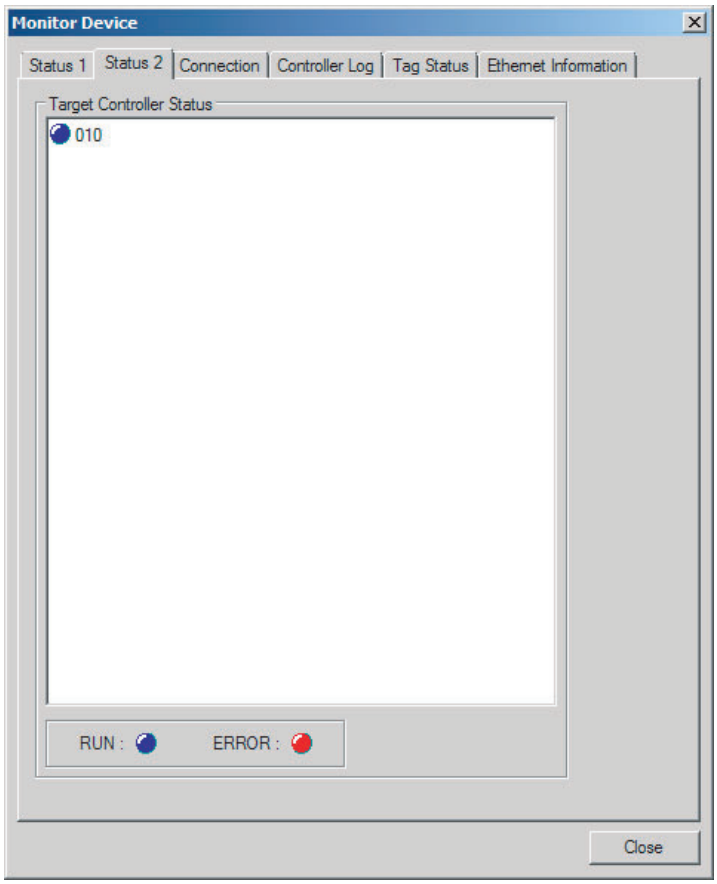
Classification	Item	Corresponding system-defined variable
Ethernet Status	Com. Controller Error	An error occurred in the communications controller.
	IP Address Duplicated	The same IP address is assigned to more than one node.
	On-Line	Indicates that the Unit is online. (The EtherNet/IP Unit can perform communications processing.)
	Tag Data Link	Indicates that the tag data link is in operation. This is TRUE in the following cases: • The originator is set up and the power supply is turned ON. • The originator is set up and the start data link switch is changed to TRUE.
	Multiple Switch ON	Indicates that more than one data link start/stop switch changed to TRUE at the same time.
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.


Monitor Device
Status 1
Status 2
Connection
Controller Log
Tag Status
Ethernet Information
Ethernet Status
Com. Controller Error
IP Address Duplicated
On-Line
Tag Data Link
Multiple Switch ON
Data Link Status
Comparison Error
Tag Data Link Error
Invalid Parameter
All Tag Data Link
Tag Data Link
Configuration Error Status
Ethernet Link Status
Ethernet Config Logical Error
IP Router Table Error
Ethernet Ext Config Logical Error
BOOTP Server Error
Target Node Status
010
Close

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.

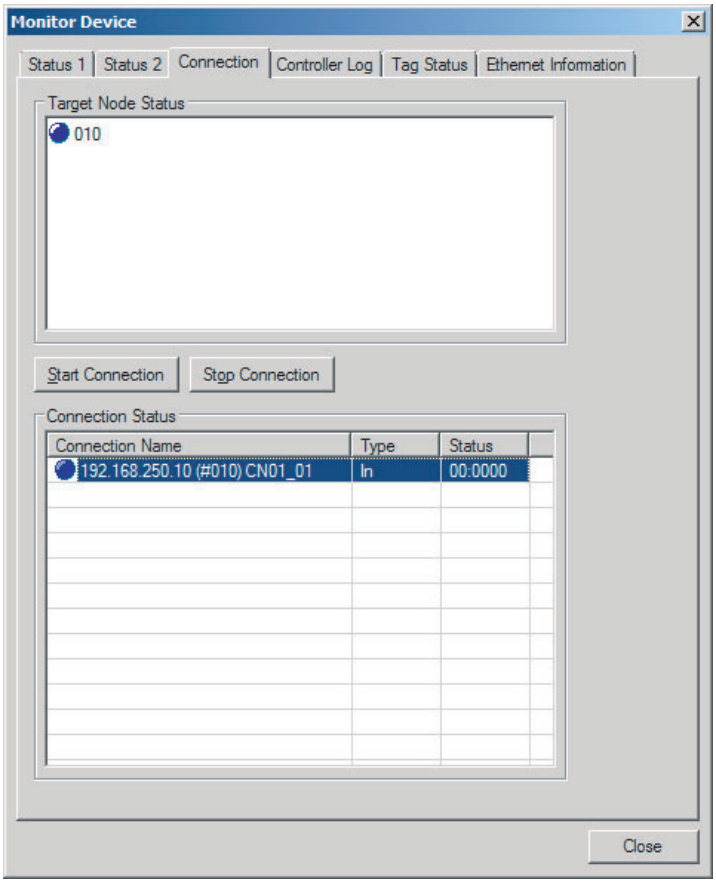


Additional Information

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.

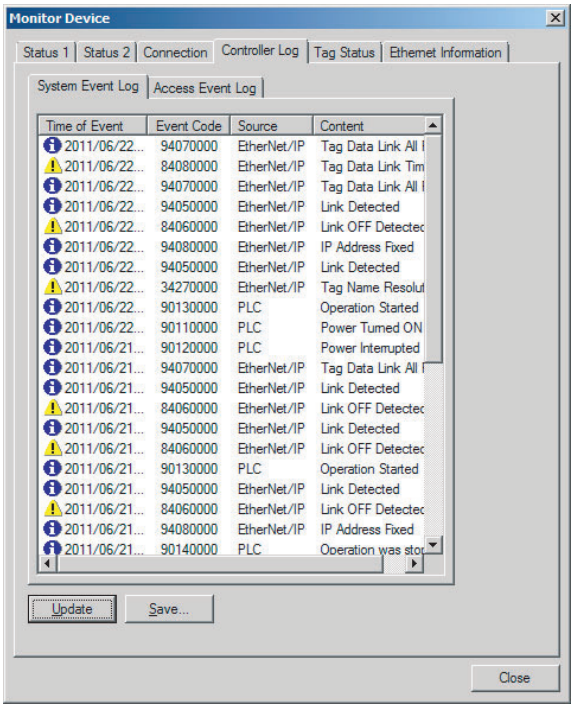
Connection Tab Page

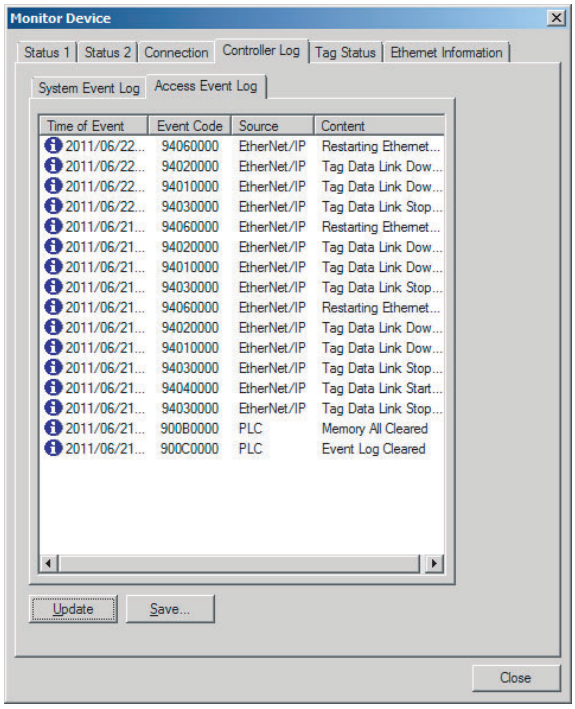
Information about the target node that acts as the originator is displayed. If all tag data link connections to the node are established and normal, this information is displayed in blue. However, if any connection is broken it is displayed in red. However, this information is displayed in gray if the connection to the node is stopped. In addition, the *Connection Status* Area shows the current status of each connection that is set as the originator. This information can be used to identify the cause of tag data link errors. Refer to *14-3-2 Connection Status Codes and Troubleshooting* for details on the connection status.



Controller Event Log Tab Page

This tab page displays the Controller event log that is stored in the CPU Unit. 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 CPU Unit for details on error information.



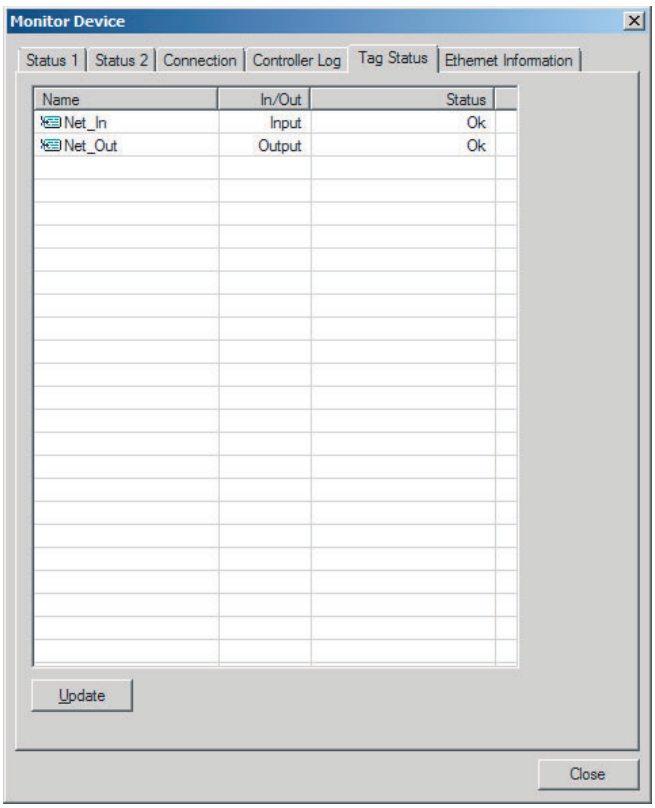


Tag Status Tab Page

This tab page displays if the tag settings for each tag for tag data links are set so that data can be exchanged with the CPU Unit. The following status is displayed depending on the status that is set.

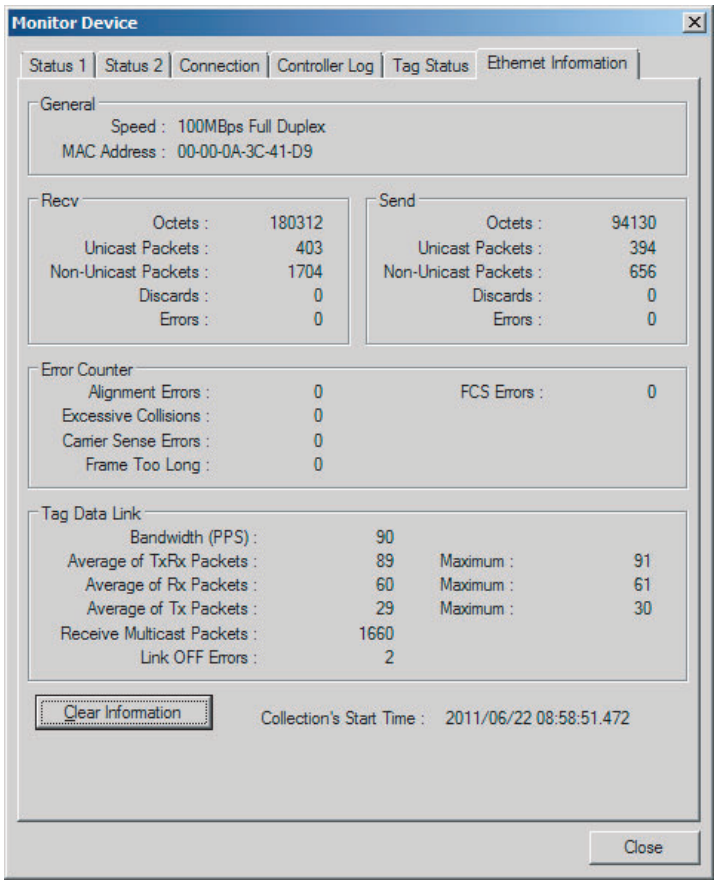
- Normal resolution completed: Normal data exchange is possible.
- Resolving: The variables with tags are being resolved. When the resolution is completed normally, a connection will be established and the data exchange will start.
- Size does not match error: Different sizes are set for the network 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 CPU Unit for the specified tag setting. A connection will not be established for a tag for which this error occurs.
- Attribute error: Writing is not possible for Read Only and Constant attributes.

If the status is not “Normal resolution completed,” check the tag data link settings or the network variable settings in the variable table in the NJ-series CPU Unit.



● 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).



14-3-2 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 *14-3-1 The Network Configurator's Device Monitor Function* 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□, and NJ501-1□00	CS1W-EIP21, CJ1W-EIP21, CJ2H-CPU□□-EIP, CJ2M-CPU3□, and NJ501-1□00
Configuration 2	CS1W-EIP21, CJ1W-EIP21, CJ2H-CPU□□-EIP, CJ2M-CPU3□, and NJ501-1□00	Products from other manufacturers
Configuration 3	Products from other manufacturers	CS1W-EIP21, CJ1W-EIP21, CJ2H-CPU□□-EIP, CJ2M-CPU3□, and NJ501-1□00

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 is does, contact the target device's manufacturer.)	Depends on the originator's specifications. (This error should not occur. If is 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.
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.

Connection status		Source of error	Handling		
General Status (hex)	Additional Status (hex)		Configuration 1	Configuration 2	Configuration 3
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 connectionÅfs 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.
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 linkÅfs 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.

Connection status		Source of error	Handling		
General Status (hex)	Additional Status (hex)		Configuration 1	Configuration 2	Configuration 3
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.)
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 *14-1-1 How to Check for Errors*.	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.)

Connection status		Source of error	Handling		
General Status (hex)	Additional Status (hex)		Configuration 1	Configuration 2	Configuration 3
OMRON error code					
01	0810	Error code returned from target: New data could not be obtained from the CPU Unit when opening connection. (The Unit will automatically recover, and attempt to open the connection again.)	This error may occur if the CPU Unit'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 CPU Unit.	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 CPU Unit when opening connection. (The Unit will automatically recover, and attempt to open the connection again.)	This error may occur if the CPU Unit'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.)

App

Appendices

A-1 Functional Comparison of EtherNet/IP Ports on NJ-series CPU Units and Other Series A-2

A-2 EDS File Management A-3

A-2-1 Installing EDS Files A-4
A-2-2 Creating EDS Files A-5
A-2-3 Deleting EDS Files A-5
A-2-4 Saving EDS Files A-6
A-2-5 Searching EDS Files A-6
A-2-6 Displaying EDS File Properties A-6
A-2-7 Creating EDS Index Files A-7

A-3 Precautions for Using the Network Configurator on Windows XP, Windows Vista, or Windows 7 . . . . . A-8

A-3-1 Changing Windows Firewall Settings A-8

A-4 Variable Memory Allocation Methods A-11

A-4-1 Variable Memory Allocation Rules A-11
A-4-2 Important Case Examples A-14

A-5 Precautions When Accessing External Outputs from NJ-series CPU Units A-15

A-1 Functional Comparison of EtherNet/IP Ports on NJ-series CPU Units and Other Series

OK: Supported., ---: Not supported.

Item	Built-in EtherNet/IP port on NJ-series CPU Unit	CJ-series Ethernet Unit	EtherNet/IP Unit (built-in port on CJ2 CPU Unit)	
			Unit version 1.0	Unit version 2.0
Tag data link communications service	OK	---	OK	OK
CIP message communications service	OK	---	OK	OK
Socket service	OK	OK	---	---
File transfer (FTP)	OK	OK	---	OK
Mail send/receive	---	OK	---	---
Web functions	---	OK	---	---
Automatic adjustment of PLC/Controller's internal clock	OK	OK	---	OK
Error history	OK*	OK	OK	OK
Response to PING command	OK	OK	OK	OK
SNMP/SNMP trap	OK	---	---	OK
CIDR function for IP addresses	OK	---	---	OK
Online connection via EtherNet/IP using CX-One/Sysmac Studio	OK	---	---	OK
Online connection via EtherNet/IP using Network Configurator	OK	---	OK	OK

* This is equivalent to the event log in the built-in EtherNet/IP of an NJ-series Controller.

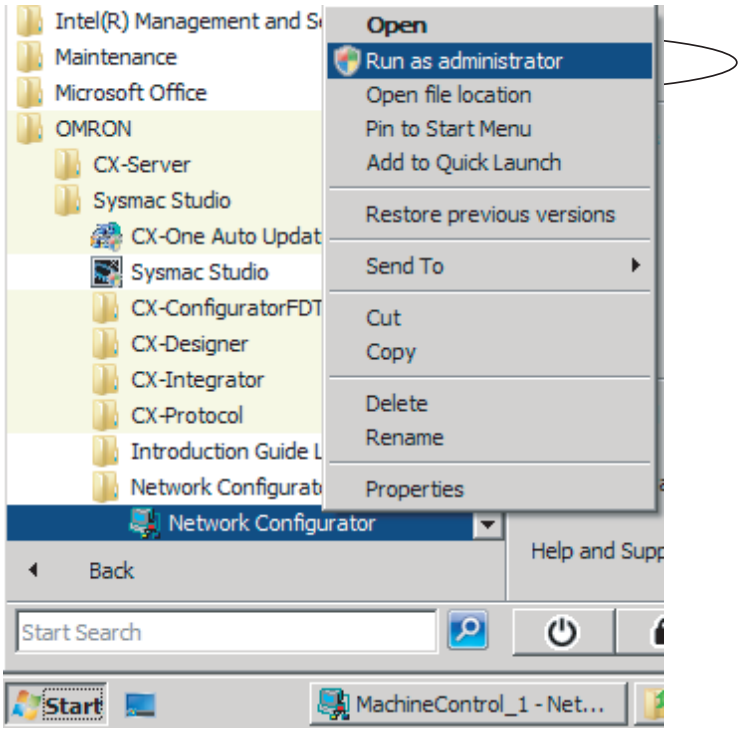
A-2 EDS File Management

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-2-1 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-2-2 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 and I/O 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

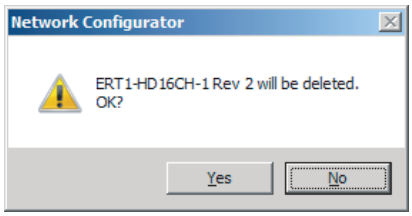
You cannot set device parameters with the Network Configurator's EDS file creation function. Obtain a proper EDS file from the manufacturer of the device to make device parameter settings for the device.

A-2-3 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-2-4 Saving EDS Files

EDS File – Save

To save the EDS file, use the following procedure.

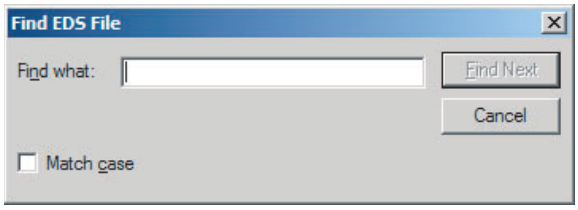
1. Select the target hardware device in the Hardware List, and then select ***EDS File*** – ***Save***.
   A Save EDS File Dialog Box is displayed.
2. Input the folder and file names and click the **Save** Button.
   The EDS file is saved.

A-2-5 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

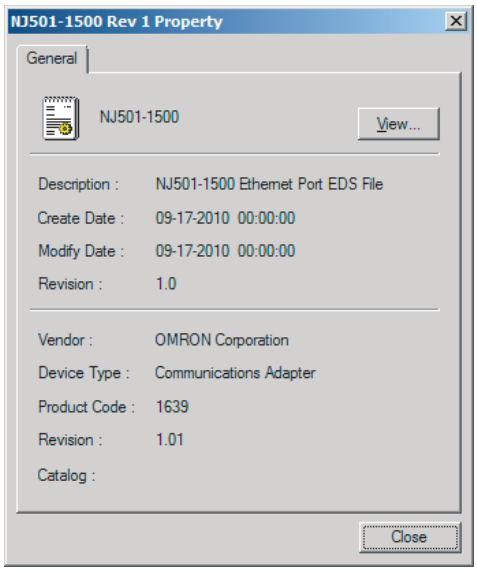
- The device is found only if it is located below the present cursor position in the Hardware List.
- To search all the devices, select *Hardware* in the Hardware List before you perform the search procedure.

A-2-6 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-2-7 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-3 Precautions for Using the Network Configurator on Windows XP, Windows Vista, or Windows 7

Better firewall security for Windows XP (SP2 or higher), Windows Vista, and Windows 7 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 NJ-series CPU Unit.

- If you select ***Option*** – ***Select Interface*** – ***Ethernet I/F***
- If you select ***Option*** – ***Select Interface*** – ***NJ Series Ethernet Direct I/F***
- If you select ***Option*** – ***Select Interface*** – ***NJ Series USB Port***

Precautions for Correct Use

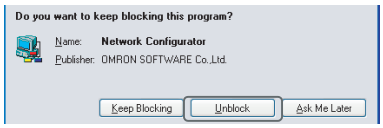
The main function of the firewall is to prevent unwanted access from external sources (e.g., the Internet). The changes that are made with the following procedures are to allow the Network Configurator and the NJ-series CPU Unit to connect. If your computer is on an inhouse network, make sure that security will not be jeopardized before you change the settings.

A-3-1 Changing Windows Firewall Settings

Windows XP

1 When you attempt to connect to the NJ-series CPU Unit from the Network Configurator, the Windows Security Warning Dialog Box is displayed.

2 Click the **Unblock** Button.

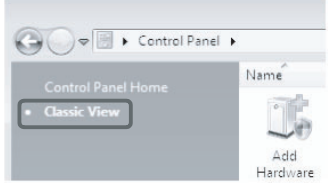


A USB or EtherNet/IP connection will be approved for the Network Configurator, and you will be able to connect the Network Configurator in the future.

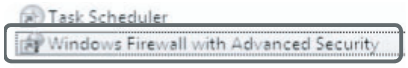
Windows Vista or Windows 7

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.

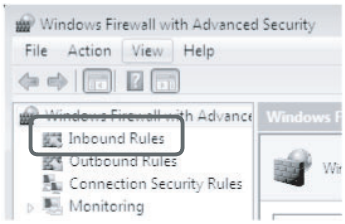
1 Select *Control Panel* from the Windows Start Menu and change the display to Classic View.



2 Open the Administrative Tools and select *Windows Firewall with Advanced Security* from the dialog box that is displayed.



3 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.



4 Select *New Rule* under *Inbound Rules* in the Actions Area on the right side of the dialog box.

5 Make the following settings for each step in the New Inbound Rule Wizard Dialog Box, and click the **Next** Button to move between steps.

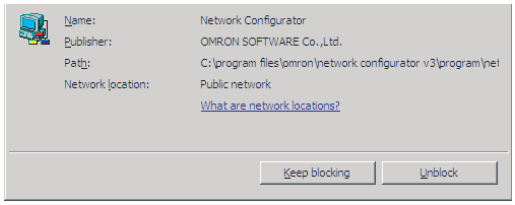
Rule Type	Select *Custom*.
Program	Select *All Programs*.
Protocol and support	Select ICMPv4 as the protocol type.
Scope	Select *Any IP address* for everything.
Action	Select *Allow the connection*.
Profile	Select *Domain*, *Private*, and *Public*.
Name	Enter any name, e.g., Omron_EIP.

6 Click the **Finish** Button. The rule that you defined will be registered in the Inbound Rules (e.g., Omron_EIP).

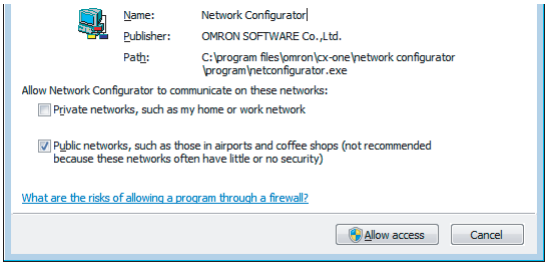
Close the *Windows Firewall with Advanced Security* Dialog Box.

7 When you attempt to connect to the NJ-series CPU Unit from the Network Configurator, the Windows Security Warning Dialog Box is displayed.

8 Click the **Unblock** Button.



(Windows Vista)



(Windows 7)

A USB or EtherNet/IP connection will be approved for the Network Configurator, and you will be able to connect the Network Configurator in the future.

A-4 Variable Memory Allocation Methods

When you assign an NJ-series CPU Unit structure variable to a tag and create a tag data link between an NJ-series CPU Unit and another CPU Unit, or when you use CIP messages to access variables, you must align the memory locations of the structure variable members with those of the other device.

A-4-1 Variable Memory Allocation Rules

Variables are stored at locations in memory that are multiples of the alignment values shown in the following table.

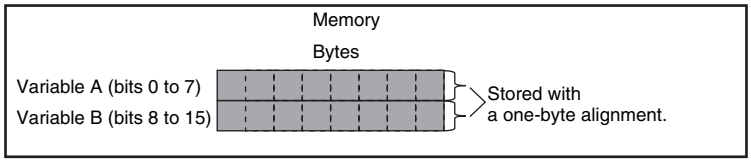
Data type	Size	Alignment
BOOL	16 bits	2 bytes
BYTE, USINT, or SINT	8 bits	1 byte
WORD, UINT, or INT	16 bits	2 bytes
DWORD, UDINT, or DINT	32 bits	4 bytes
LWORD, ULINT, or LINT	64 bits	8 bytes
REAL	32 bits	4 bytes
LREAL	64 bits	8 bytes
TIME, DATE, TIME_OF_DAY, or DATE_AND_TIME	64 bits	8 bytes
STRING[N]	(N+1) × 8 bits	1 byte
Enumerations	32 bits	4 bytes

Basic Data Types

- **Variables with One-Byte Alignments (e.g., BYTE)**

  These variables are stored in memory with a one-byte alignment.

  Example:



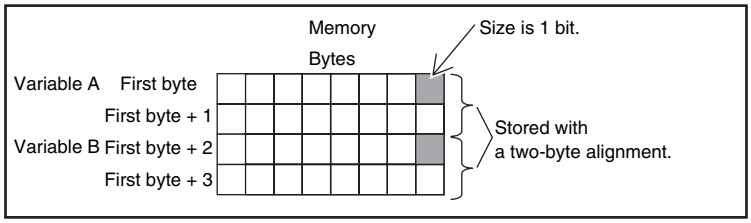
Variable Table

Name	Data type
A	BYTE
B	BYTE

- **Variables with Two-byte Alignments (e.g., BOOL and WORD)**

  These variables are stored in memory with a two-byte alignment.

  Example:



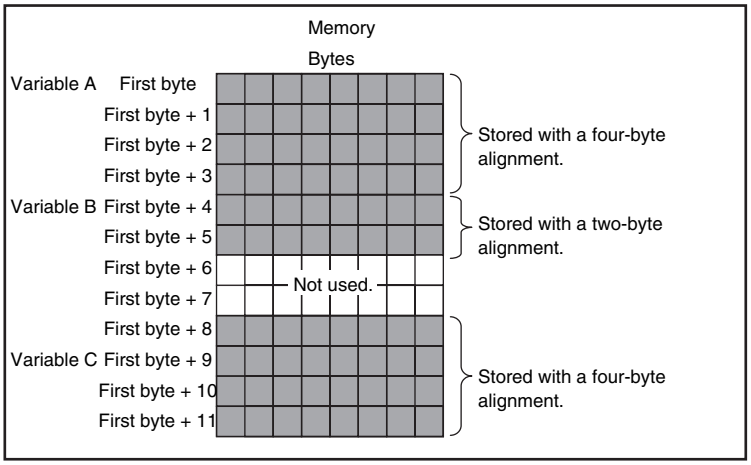
Variable Table

Name	Data type
A	BOOL
B	BOOL

Variables with Four-byte Alignments (e.g., DWORD)

These variables are stored in memory with a four-byte alignment. The first byte is the first of four bytes in memory. Therefore, if a variable with a two-byte alignment, such as WORD data, is inserted, two bytes of unused memory will remain.

Example:



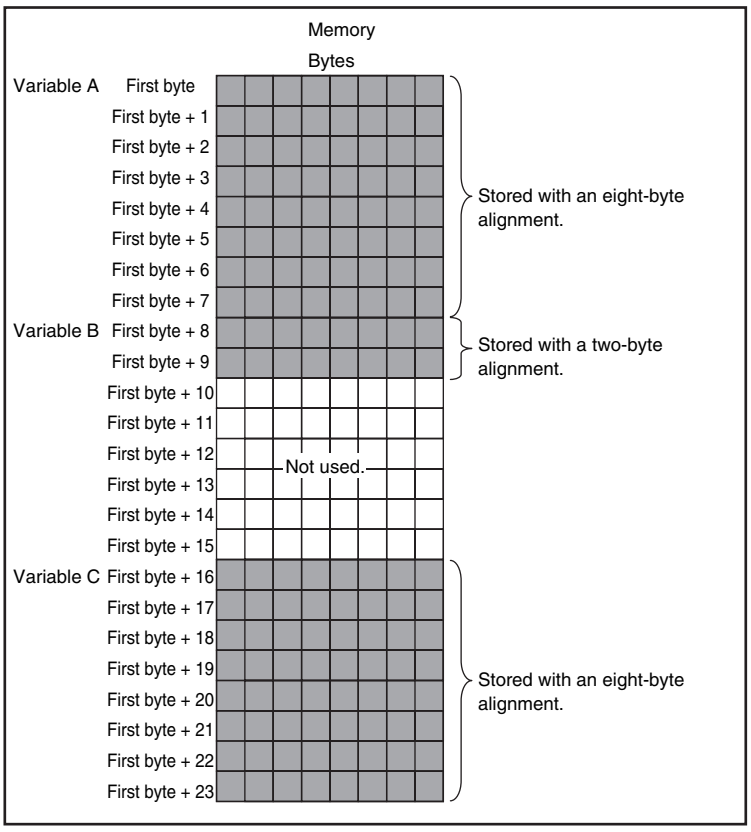
Variable Table

Name	Data type
A	DWORD
B	WORD
C	DWORD

Variables with Eight-byte Alignments (e.g., LWORD)

These variables are stored in memory with an eight-byte alignment. The first byte is the first of eight bytes in memory. 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:



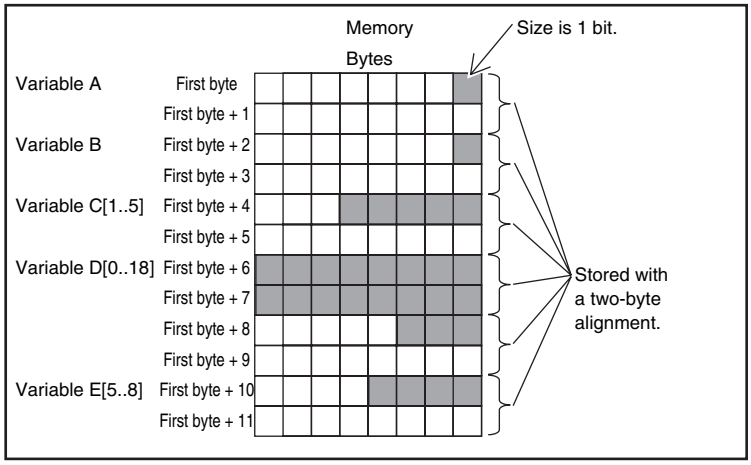
Variable Table

Name	Data type
A	LWORD
B	WORD
C	LWORD

Array Variables

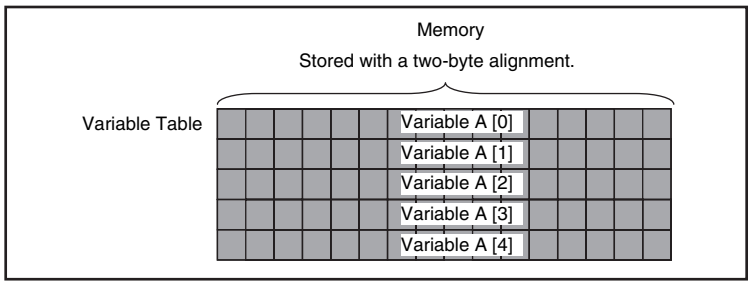
Array variables are stored in a continuous section of memory.

Example:



Variable Table

Name	Data type
A	BOOL
B	BOOL
C	ARRAY [1..5] OF BOOL
D	ARRAY [0..18] OF BOOL
E	ARRAY [5..8] OF BOOL



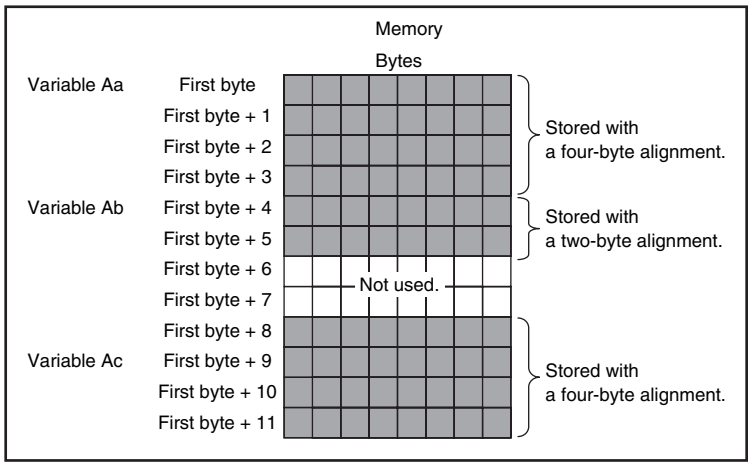
Variable Table

Name	Data type
Variable A	ARRAY[0..4] OF INT

Structure Variables

A continuous section of memory is allocated based on the alignment value of the data type of the structure variable to store structure data.

Example:



Data Type Definitions

Name	Data type
Structure A	STRUCT
a	DINT
b	INT
c	DINT

Variable Table

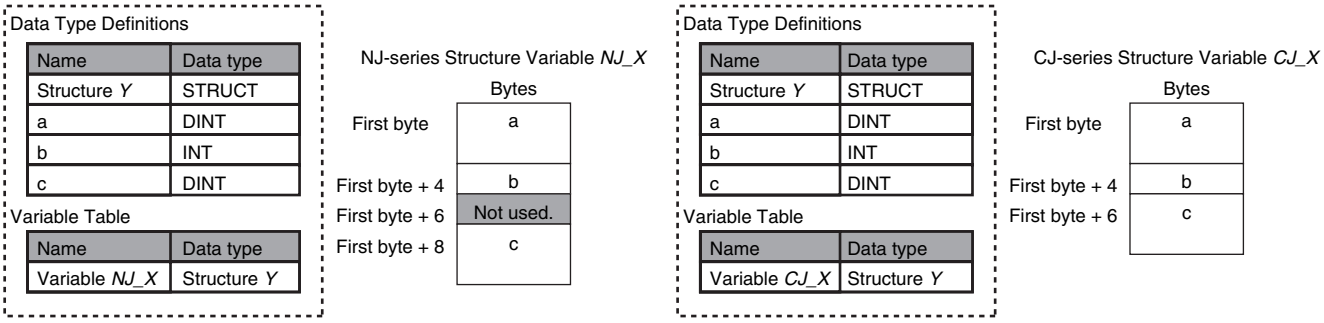
Name	Data type
Variable A	Structure A

A-4-2 Important Case Examples

When the object for a tag data link includes a structure variable, make sure that the locations in memory of the structure variable members match between the NJ-series CPU Unit and the CPU Unit you need to exchange data with. You need to be aware of the locations in memory of structure variable members in the following cases.

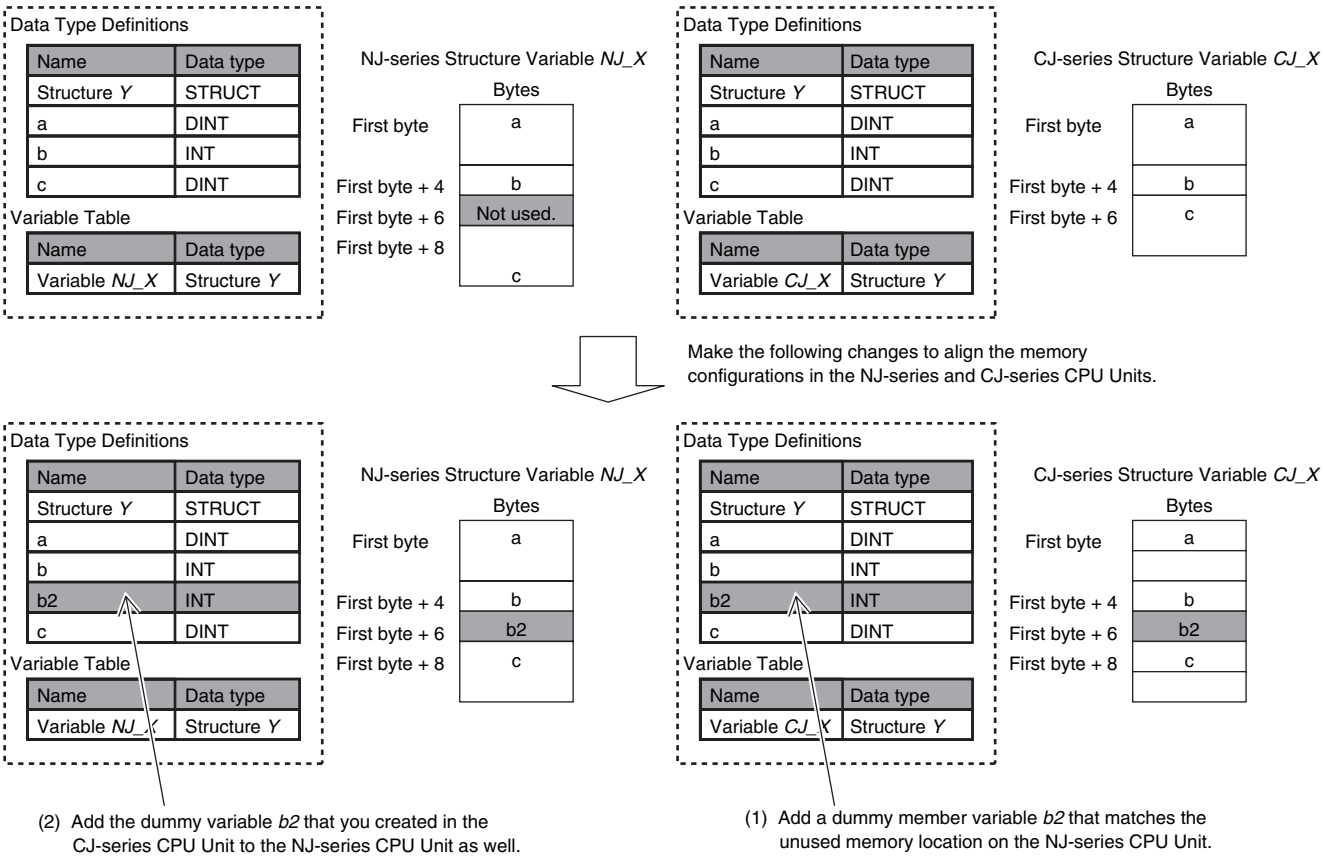
● Reading and Writing Variables through CIP Messages or EtherNet/IP Tag Data Links between an NJ-series CPU Unit and a CJ-series CPU Unit

The differences in memory configuration for structure variables between an NJ-series CPU Unit and a CJ-series CPU Unit are shown below.



In this case, use the following solution.

Solution: Align the CJ-series and NJ-series memory locations.
Example: Insert members to adjust memory locations.
You must match both the memory locations and the data types.
You need to create the alignment members in both the CJ-series and NJ-series CPU Units.



A-5 Precautions When Accessing External Outputs from NJ-series CPU Units

Observe the following precautions when you access variables or I/O memory addresses that are assigned to external outputs in an NJ-series CPU Unit.

- 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 NJ-series CPU Unit through a tag data link or communications instruction from an external device 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.

* This applies to global device variables that are assigned to I/O ports for CJ-series Basic Output Units or EtherCAT output slaves and to global variables with AT specifications to output bits that are assigned to CJ-series Basic Output Units.

- Precaution When Directly Writing to I/O Memory Addresses Assigned to Output Bits for CJ-series Basic Output Units
  Any value that is written to an I/O memory address that corresponds to an output bit that is assigned to a CJ-series Basic Output Unit through a tag data link will be overwritten by the execution results of the user program. The value that is written directly to the I/O memory address from the tag data link will therefore not be output to the external device.

I

Index

Index

A

Accept TCP Socket instruction 9-13
addresses 5-7, 12-22
All Tag Data Link Communications Status 3-7, 3-15
array variables
  preparing array variables to input and output service data and response data 8-11
assumed causes 14-9
Auto Connection Configuration 7-36
automatically setting connections 7-35

B

bandwidth usage
  requested packet intervals 13-3
  tag data links 13-7
Basic Ethernet Setting Error 3-4, 3-12
binary format 10-17
BOOTP client 1-12
BOOTP Server Error 3-4, 3-13
boots 2-2
broadcasting 9-8
built-in EtherCAT port 1-9
Built-in EtherNet/IP Error 3-3, 3-10, 14-5
built-in EtherNet/IP port 1-9
  settings 5-1
  socket services
    overview 9-9
  specifications 1-7

C

CIDR 4-3
CIP Communications 1-11
CIP Communications Error 3-4, 3-11, 14-5
CIP communications errors 14-6
CIP communications instructions 8-3
  using 8-4
CIP message communications
  using 8-3
CIP message communications service
  overview 8-2
  specifications 8-2
Clear UDP/TCP Socket Receive Buffer instruction 9-13
clock
  automatic adjustment 1-13, 11-2
    required settings 11-4
    specifications 11-3, 11-4
    troubleshooting 11-7
Close CIP Class 3 Connection instruction 8-4
Close UDP/TCP Socket instruction 9-13
Communications Controller Error 3-4, 3-12
communications load
  adjusting 13-6
Communications Port Error 3-3, 3-11, 14-5
communications port errors 14-6
community names 5-8, 5-9, 5-10, 12-23, 12-24, 12-25
Connect TCP Socket instruction 9-13
Connection I/O Type 7-32, 7-33
Connection Name 7-32, 7-33
connection settings
  automatically setting connections 7-35
  batch editing 7-32
  editing individual connections 7-31
  Register Device List 7-29
connection status
  codes and troubleshooting 14-45
Connection Tab Page 14-42
Connection Type 7-32, 7-33
connections
  checking 7-58
Controller Event Log Tab Page 14-43
Controller Object 8-38
Controller status 7-8
Create UDP Socket instruction 9-13
cyclic communications 1-11

D

data areas 7-3
data processing time
  calculation example 13-22
  overview 13-22
data transmissions
  timing 13-21
default gateway 5-2, 9-11
destination IP addresses 5-3, 9-12
destination mask IP addresses 5-3, 9-12
device bandwidth usage
  adjusting 13-9
Device Connection Structure Tree 7-37
Device Monitor 14-38
device parameters
  clearing 7-54
  verifying 7-50
device status
  displaying 7-61
devices
  changing 7-60
  registering 7-17
DNS 5-2
DNS Server Connection Error 3-5, 3-15
domain names 5-2

E

EDS files
  management A-3

_EIP_BootpErr 3-4, 3-13
_EIP_CipErr 3-4, 3-11, 14-5
_EIP_DNSSrvErr 3-5, 3-15
_EIP_ErrSta 3-3, 3-10, 14-5
_EIP_EstbTargetSta 14-41
_EIP_EstbTargetSta[255] 3-7, 3-16
_EIP_EtnCfgErr 3-4, 3-12
_EIP_EtnOnlineSta 3-7, 3-15
_EIP_IdentityErr 3-5, 3-13
_EIP_IPAdrCfgErr 3-4, 3-12
_EIP_IPAdrDupErr 3-4, 3-12
_EIP_IPRTblErr 3-5, 3-13
_EIP_LanHwErr 3-4, 3-12
_EIP_MacAdrErr 3-4, 3-12
_EIP_MultiSwONErr 3-5, 3-14
_EIP_NTPResult 3-8, 3-17
_EIP_NTPResult.ExecNormal 3-8
_EIP_NTPResult.ExecTime 3-8
_EIP_NTPSrvErr 3-5, 3-14
_EIP_PortErr 3-3, 3-11, 14-5
_EIP_RegTargetSta[255] 3-7, 3-15
_EIP_TagAdrErr 3-5, 3-14
_EIP_TargetNodeErr 3-16
_EIP_TargetNodeErr[255] 3-8
_EIP_TargetPLCErr 7-8, 14-41
_EIP_TargetPLCErr[255] 3-8, 3-16
_EIP_TargetPLCModeSta 7-8, 14-41
_EIP_TargetPLCModeSta[255] 3-7, 3-16
_EIP_TcpAppCfgErr 3-5, 3-14
_EIP_TcpAppErr 3-4, 3-11, 14-5
_EIP_TDLinkAllRunSta 3-7, 3-15
_EIP_TDLinkCfgErr 3-5, 3-13
_EIP_TDLinkErr 3-5, 3-14
_EIP_TDLinkOpnErr 3-5, 3-13
_EIP_TDLinkRunSta 3-7, 3-15
_EIP_TDLinkStartCmd 3-9, 3-17, 7-52
_EIP_TDLinkStopCmd 3-9, 3-17, 7-52
error status 3-6
error status variables 14-5
errors
  checking communications with Network Configurator 14-6
  checking for errors 14-2
  classifications 14-6
  descriptions 14-14
  event levels 14-6, 14-7
  indicators 14-3
  reading with instructions 14-5
  sources 14-6
  tables 14-9
  Troubleshooter of an NS-series PT 14-5
  troubleshooting with Sysmac Studio 14-4
Ethernet addresses 1-9
Ethernet connectors 2-8
Ethernet Information Tab Page 14-45
Ethernet Link Object 8-34
Ethernet switches 1-5, 2-2
  functions 2-3
  selection precautions 2-4
  types 2-3
event codes 14-9
event names 14-9
events 14-2

F

FINS/TCP Connection Status 3-9
FINSTCPConnSta 3-9
fragmenting
  send data 9-4
FTP commands
  append 10-12
  bye 10-14
  cd 10-12
  close 10-14
  delete 10-14
  dir 10-11
  get 10-13
  ls 10-10
  mdelete 10-14
  mdir 10-11
  mget 10-13
  mkdir 10-12
  mls 10-10
  mput 10-14
  open 10-10
  put 10-13
  pwd 10-12
  quit 10-15
  rename 10-11
  rmdir 10-12
  type 10-13
  user 10-10
FTP Display 5-5
FTP server 1-13, 5-5, 10-6
  application example 10-7
  application example from host computer 10-19
  application procedure 10-5
  commands 10-9
  overview 10-2
  specifications 10-2
FTP Settings Display 10-6
functions
  comparison between NJ Series and Other Series A-2

G

gateway addresses 5-3, 9-12
general status 8-28
general status codes 8-26
Get EtherNet/IP Error Status instruction 14-5
global addresses 4-10
global broadcasting 9-8

H

host names .....
..... 5-3, 5-6, 5-8, 5-9, 5-10, 11-5, 12-23, 12-24, 12-25
specifying 1-14

I

Identity Error 3-5, 3-13
Identity Object 8-30
indicators 1-9, 1-10, 14-3
information event level 14-7
input ON response time 13-25
instructions
CIPClose 8-4
CIPOpen 8-4
CIPRead 8-4
CIPSend 8-4
CIPUCMMRead 8-3
CIPUCMMSend 8-3
CIPUCMMWrite 8-3
CIPWrite 8-4
GetEIPError 14-5
reading errors 14-5
SktClearBuf 9-13
SktGetTCPStatus 9-13
SktTCPAccept 9-13
SktTCPConnect 9-13
SktTCPRcv 9-13
SktTCPSend 9-13
SktUDPCreate 9-13
SktUDPRcv 9-13
SktUDPSend 9-13
interval
NTP operation timing 5-6, 11-6
IOI 8-8
IP address configuration 4-2
IP Address Duplication Error 3-4, 3-12
IP addresses .....
.....5-2, 5-3, 5-6, 5-8, 5-9, 5-10, 9-11,
..... 11-5, 12-23, 12-24, 12-25
allocation 4-3
checking current addresses 4-9
setting 4-5
IP Router Table Display 9-12

K

Keep Alive Monitoring Time setting 5-3, 9-12
Keep Alive setting 5-3, 9-12

L

Linger option 5-3, 9-12
LINK settings 5-4
LINK/ACT indicator 1-10
local broadcasting 9-8
location
SNMP 5-7, 12-22
login names 5-5, 10-6

M

MAC Address Error 3-4, 3-12
major fault event level 14-7
message communications 1-12
message service
transmission delay 13-27
MIB
groups 12-4
system design 12-4
MIB objects
detailed descriptions 12-5
minor fault event level 14-7
multi-cast communications 7-7
multicast filtering 2-3
Multiple Switches ON Error 3-5, 3-14

N

NET ERR indicator 1-10
NET RUN indicator 1-10
network configuration
verifying 7-49
network configuration file
reading 7-57
saving 7-55
Network Configurator 1-6
connections through NJ-series CPU Unit's
USB port 7-40
connections via Ethernet to NJ-series CPU Unit ... 7-41
Ethernet connections 7-38
network transmission delay time 13-26
network variables 7-6
importing to Network Configurator 7-25
Normal Target Node Information 3-7, 3-16, 14-41
NTP Display 11-5
NTP Last Operation Time 3-8, 3-17
NTP Operation Information 3-8
NTP Operation Result 3-8, 3-17
NTP operation timing 5-6, 11-6
NTP server clock information 5-6, 11-5
NTP Server Connection Error 3-5, 3-14
NTP Settings Display 5-6
number of connections
calculating 13-4

O

observation event level 14-7
Online 3-7, 3-15
Open CIP Class 3 Connection instruction 8-4
Originator Variable 7-33
output ON response time 13-26
output variables
operation and timing 8-25, 9-16

P

Packet Interval (RPI) 7-32
partial fault event level 14-7
passwords 5-5, 10-6
PING command 6-2
port numbers ... 5-5, 5-6, 5-7, 5-9, 10-6, 11-5, 12-22, 12-24
  socket services 9-2
PPS 13-3
precautions
  socket services 9-29
priority DNS server 5-2
private addresses 4-10
programming
  CIP message communications 8-12
  ladder programming for tag data links 7-62
  socket services 9-18, 9-23

R

Read TCP Socket Status instruction 9-13
Read Variable Class 3 Explicit instruction 8-4
Read Variable UCMM Explicit instruction 8-3
receive data processing time 13-26
Recognition 1 settings 5-8, 12-23
Recognition 2 settings 5-8, 12-23
recognition method 5-8, 12-23
recognition trap
  sending 5-7, 12-22
Registered Target Node Information 3-7, 3-15
registering devices 7-17
request path 8-8
requested packet intervals 7-7, 13-26
  accuracy 13-5
  adjusting according to task periods 13-23
  bandwidth usage 13-3
  changing 13-10
  settings 13-2
response codes 8-26
route path 8-6
RPI setting 7-33

S

SD Memory Cards
  file types 10-17
  format of variable data 10-17
  initializing 10-17
  types 10-16
secondary DNS server 5-2
send data processing time 13-26
Send Explicit Message Class 3 instruction 8-4
Send Explicit Message UCMM instruction 8-3
server specifying method 5-6, 11-5
settings 5-1
SNMP 5-7, 12-22
SNMP agent 1-15, 12-2
  application procedure 12-20
  required settings 12-20
SNMP messages 12-3
SNMP Settings Display 5-7, 12-22
SNMP specifications 12-3
SNMP Trap Settings Display 5-9, 12-24
SNMP traps 5-9, 12-3, 12-24
socket service 1-14
socket service communications
  data reception processing 9-6
  fragmenting of send data 9-4
socket service instructions 9-13
socket services
  application procedure 9-14
  built-in EtherNet/IP port
    overview 9-9
  port numbers 9-2
  precautions 9-29
  required settings 9-10
sockets 9-2
specifying method 5-9, 5-10, 12-24, 12-25
Status 1 Tab Page 14-39
Status 2 Tab Page 14-41
structure variables
  providing to input request paths 8-9
subnet masks 4-3, 5-2, 9-11
Sysmac Studio 1-6
system-defined variables 3-2
  EtherNet/IP communications errors 3-3, 3-10
  EtherNet/IP communications status 3-7, 3-15
  EtherNet/IP communications switches 3-9, 3-17

T

tag data link
  functions 7-5
  specifications 7-5
Tag Data Link Communications Error 3-5, 3-14
Tag Data Link Communications Start Switch 3-9, 3-17
Tag Data Link Communications Status 3-7, 3-15
Tag Data Link Communications Stop Switch 3-17
Tag Data Link Connection Failed 3-5, 3-13
tag data link parameters
  downloading 7-43
  setting and downloading 7-6
  uploading all 7-46
  uploading from individual devices 7-47
Tag Data Link Setting Error 3-5, 3-13
Tag Data Link Start Switch 7-52
Tag Data Link Stop Switch 7-52
tag data links 1-11
  automatically starting 7-52
  bandwidth usage and requested packet intervals . 13-8
  communications method 13-2
  data concurrency 7-10
  effect on task periods 13-23
  introduction 7-2
  maximum response time 13-25
  models other than NJ-series CPU Units 7-67
  setting 7-14
  starting and stopping 7-8

starting and stopping for individual devices 7-52
verifying 7-49
Tag Name Resolution Error 3-5, 3-14
tag sets 7-3
creating 7-18
Tag Status Tab Page 14-44
tags 7-3
Target Device 7-33
Target Node Error Information 3-8, 3-16
Target PLC Error Information 3-8, 3-16, 7-8, 14-41
Target PLC Operating Mode 3-7, 3-16, 7-8, 14-41
Target Variable 7-33
task periods
relationship with requested packet intervals 13-23
TCP Application Communications Error 3-4, 3-11, 14-5
TCP application errors 14-6
TCP communications 9-3
procedure 9-3
TCP Socket Receive instruction 9-13
TCP Socket Send instruction 9-13
TCP/IP Advanced Setting Error 3-5, 3-13
TCP/IP Basic Setting Error 3-4, 3-12
TCP/IP Display 5-2, 9-11
TCP/IP Interface Object 8-32
TCP/IP Setting Error 3-5, 3-14
time
NTP operation timing 5-6, 11-6
timeout time
NTP operation timing 5-6, 11-6
Timeout Value 7-32, 7-33
Trap 1 settings 5-9, 12-24
Trap 2 settings 12-25
Troubleshooter 14-5
troubleshooting 14-4, 14-37
SNMP agent 12-26
twisted-pair cable 1-5
installation environment precautions 2-7
installation precautions 2-6

U

UDP communications 9-3
UDP Socket Receive instruction 9-13
UDP Socket Send instruction 9-13
unicast communications 7-7
USB port 1-9

V

variable memory allocation
alignment A-11
important examples A-14
rules A-11
versions 5-9, 5-10, 12-24, 12-25

W

Windows firewall settings
changing A-8
Write Variable Class 3 Explicit instruction 8-4
Write Variable UCMM Explicit instruction 8-3

OMRON Corporation **Industrial Automation Company**
Tokyo, 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
One Commerce Drive Schaumburg,
IL 60173-5302 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 2011 All Rights Reserved.
In the interest of product improvement,
specifications are subject to change without notice.

Cat. No. W506-E1-01 0711