OMRON

Machine Automation Controller

CJ-series

# EtherNet/IP™ Units

# Operation Manual for NJ-series CPU Unit

CJ1W-EIP21
CJ1W-EIP21S

EtherNet/IP Units

W495-E1-12

## NOTE

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

## Trademarks

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

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

## Copyrights

Microsoft product screen shots used with permission from Microsoft.

# Introduction

Thank you for purchasing an EtherNet/IP Unit.
This manual contains information that is necessary to use the EtherNet/IP 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.

- CJ-series EtherNet/IP Unit
  - CJ1W-EIP21
  - CJ1W-EIP21S

This manual contains information that is necessary to use the CJ1W-EIP21/EIP21S by connecting it to an NJ-series CPU Unit.

To connect the CJ-series EtherNet/IP Unit to an NJ-series CPU Unit, use a combination of the product versions listed in the table below.

| EtherNet/IP Unit | NJ-series CPU Unit | Sysmac Studio |
|---|---|---|
| CJ1W-EIP21 with unit version 2.1 or later | Unit version 1.01 or later | Version 1.02 or higher |
| CJ1W-EIP21 with unit version 3.0 or later | Unit version 1.01 or later | Version 1.11 or higher |
| CJ1W-EIP21S with unit version.1.0<br>Lot number 241001□ or later[*1] | Unit version 1.67 or later | Version 1.60 or higher |

[*1] For lot numbers 241001□ or later, "+NJ" is printed at the lower right of the Unit’s front panel. Refer to *1-3-2 Part Names and Functions* for information on the indication printed at the lower right corner of the front panel.

# Relevant Manuals

The following table provides the relevant manuals for the CJ-series EtherNet/IP Unit. Read all of the manuals that are relevant to your system configuration and application before you use the CJ-series EtherNet/IP Unit. Most operations are performed from the Sysmac Studio Automation Software. Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for information on the Sysmac Studio.

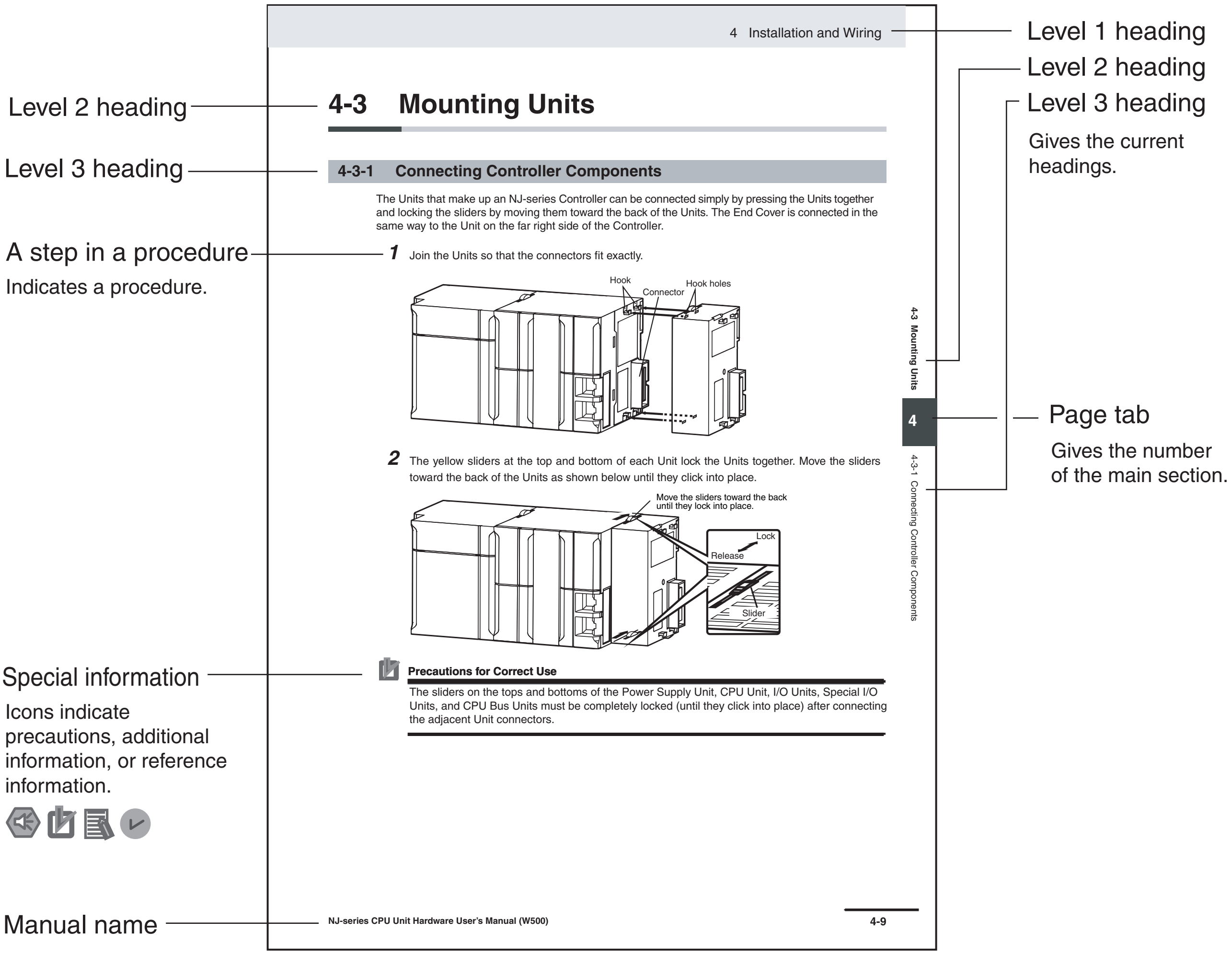
| Purpose of use | Manual | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Basic information | | | | | | | | | | | |
| | NX-series CPU Unit Hardware User's Manual | NJ-series CPU Unit Hardware User's Manual | NJ/NX-series CPU Unit Software User's Manual | NJ/NX-series Instructions Reference Manual | NJ/NX-series CPU Unit Motion Control User's Manual | NJ/NX-series Motion Control Instructions Reference Manual | NJ/NX-series CPU Unit Built-in EtherCAT Port User's Manual | NJ/NX-series CPU Unit Built-in EtherNet/IP Port User's Manual | NJ-series Database Connection CPU Unit User's Manual | NJ-series SECS/GEM CPU Units User's Manual | NJ/NX-series Troubleshooting Manual | CJ-series Special Unit Operation Manuals for NJ-series CPU Unit |
| Introduction to NX-series Controllers | ● | | | | | | | | | | | |
| Introduction to NJ-series Controllers | | ● | | | | | | | | | | |
| Setting devices and hardware | ● | ● | | | | | | | | | | |
| Using motion control | | | | | ● | | | | | | | |
| Using EtherCAT | | | | | | | ● | | | | | |
| Using EtherNet/IP | | | | | | | | ● | | | | |
| Using the database connection service | | | | | | | | | ● | | | |
| Using the GEM Services | | | | | | | | | | ● | | |
| Using CJ-series Units | | | | | | | | | | | | ● |
| Software settings | | | ● | | | | | | | | | |
| Using motion control | | | | | ● | | | | | | | |
| Using EtherCAT | | | | | | | ● | | | | | |
| Using EtherNet/IP | | | | | | | | ● | | | | |
| Using the database connection service | | | | | | | | | ● | | | |
| Using the GEM Services | | | | | | | | | | ● | | |
| Writing the user program | | | ● | ● | | | | | | | | |
| Using motion control | | | | | ● | ● | | | | | | |
| Using EtherCAT | | | | | | | ● | | | | | |
| Using EtherNet/IP | | | | | | | | ● | | | | |
| Using the database connection service | | | | | | | | | ● | | | |
| Using the GEM Services | | | | | | | | | | ● | | |
| Using CJ-series Units | | | | | | | | | | | | ● |
| Programming error processing | | | | | | | | | | | ● | |
| Testing operation and debugging | | | ● | | | | | | | | | |
| Using motion control | | | | | ● | | | | | | | |
| Using EtherCAT | | | | | | | ● | | | | | |
| Using EtherNet/IP | | | | | | | | ● | | | | |
| Using the database connection service | | | | | | | | | ● | | | |
| Using the GEM Services | | | | | | | | | | ● | | |
| Learning about error management and corrections[*1] | ▲ | ▲ | ▲ | | ▲ | | ▲ | ▲ | ▲ | ▲ | ● | ▲ |
| Maintenance | ● | ● | | | | | | | | | | |
| Using motion control | | | | | ● | | | | | | | |
| Using EtherCAT | | | | | | | ● | | | | | |
| Using EtherNet/IP | | | | | | | | ● | | | | |
| Using CJ-series Units | | | | | | | | | | | | ● |

[*1] Refer to the *NJ/NX-series Troubleshooting Manual* (Cat. No. W503) for the error management concepts and an overview of the error items. Refer to the manuals that are indicated with triangles for details on errors for the corresponding Units.

# Manual Structure

## Page Structure

The following page structure is used in this manual.



4 Installation and Wiring

4-3 Mounting Units

4-3-1 Connecting Controller Components

The Units that make up an NJ-series Controller can be connected simply by pressing the Units together and locking the sliders by moving them toward the back of the Units. The End Cover is connected in the same way to the Unit on the far right side of the Controller.

1 Join the Units so that the connectors fit exactly.

2 The yellow sliders at the top and bottom of each Unit lock the Units together. Move the sliders toward the back of the Units as shown below until they click into place.

Move the sliders toward the back until they lock into place.
Lock
Release
Slider

Precautions for Correct Use

The sliders on the tops and bottoms of the Power Supply Unit, CPU Unit, I/O Units, Special I/O Units, and CPU Bus Units must be completely locked (until they click into place) after connecting the adjacent Unit connectors.

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

4-9

Level 1 heading

Level 2 heading

Level 3 heading

Gives the current headings.

Page tab

Gives the number of the main section.

Level 2 heading

Level 3 heading

A step in a procedure

Indicates a procedure.

Special information

Icons indicate precautions, additional information, or reference information.

Manual name

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

## Special Information

Special information in this manual is classified as follows:

**Precautions for Safe Use**

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

**Precautions for Correct Use**

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

**Additional Information**

Additional information to read as required.
This information is provided to increase understanding or make operation easier.

**Version Information**

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

**Note** References are provided to more detailed or related information.

## Precaution on Terminology

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

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

# Sections in this Manual

| | |
|---|---|
| 1 | 9 |
| 2 | 10 |
| 3 | 11 |
| 4 | 12 |
| 5 | 13 |
| 6 | 14 |
| 7 | 15 |
| 8 | A |
| I | |

# CONTENTS

## Section 1 Introduction

## Section 2 Installing Ethernet Networks

## Section 3 Assigning Device Variables for CJ-series Units

## Section 4 Determining IP Addresses

## Section 5 Sysmac Studio Settings for the EtherNet/IP Unit

## Section 6 Testing Communications

## Section 7 Tag Data Link Functions

## Section 8 Message Communications

## Section 9 FTP Server

## Section 10 Automatic Clock Adjustment

## Section 11 SNMP Agent

## Section 12 Security Functions

## Section 13 Socket Services

## Section 14 Communications Performance and Communications Load

## Section 15 Troubleshooting

# Appendices

# Index

# Terms and Conditions Agreement

## Warranty, Limitations of Liability

### Warranties

- **Exclusive Warranty**

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

- **Limitations**

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

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

- **Buyer Remedy**

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

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

### Limitation on Liability; Etc

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

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

## Application Considerations

### Suitability of Use

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

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

### Programmable Products

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

## Disclaimers

### Performance Data

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

### Change in Specifications

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

### Errors and Omissions

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

# Safety Precautions

## Definition of Precautionary Information

The following notation is used in this manual to provide precautions required to ensure safe usage of a CJ-series EtherNet/IP 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.

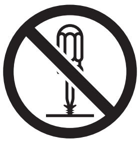
| | |
|---|---|
| ⚠ **WARNING** | Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury. Additionally, there may be severe property damage. |

| | |
|---|---|
| ⚠ **Caution** | Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury, or property damage. |

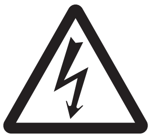
**Precautions for Safe Use**
Indicates precautions on what to do and what not to do to ensure safe usage of the product.

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

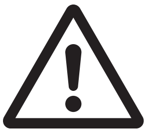
## Symbols



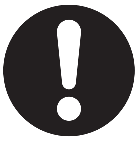
The circle and slash symbol indicates operations that you must not do.
The specific operation is shown in the circle and explained in text.
This example indicates prohibiting disassembly.



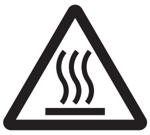
The triangle symbol indicates precautions (including warnings).
The specific operation is shown in the triangle and explained in text.
This example indicates a precaution for electric shock.



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.



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 high temperatures.

## Warnings

## ⚠ WARNING

### During Power Supply

Do not touch any of the terminals or terminal blocks while the power is being supplied. Doing so may result in electric shock.

Do not attempt to take any Unit apart. In particular, high-voltage parts are present in the Power Supply Unit while power is supplied or immediately after power is turned OFF. Touching any of these parts may result in electric shock. There are sharp parts inside the Unit that may cause injury.

### Fail-safe Measures

Provide safety measures in external circuits to ensure safety in the system if an abnormality occurs due to malfunction of the CPU Unit, slaves, or Units or due to other external factors affecting operation. Not doing so may result in serious accidents due to incorrect operation.

Emergency stop circuits, interlock circuits, limit circuits, and similar safety measures must be provided in external control circuits.

The Controller outputs may remain ON or OFF due to deposition or burning of the output relays or destruction of the output transistors. As a countermeasure for such problems, external safety measures must be provided to ensure safe operation of the system.

The CPU Unit will turn OFF all outputs from Basic Output Units in the following cases. The slaves will operate according to the settings in the slaves.

- If an error occurs in the power supply
- If the power supply connection becomes faulty
- If a CPU watchdog timer error or CPU reset occurs
- If a major fault level Controller error occurs
- While the CPU Unit is on standby until RUN mode is entered after the power is turned ON

External safety measures must be provided to ensure safe operation of the system in such cases.

If external power supplies for slaves or other devices are overloaded or shortcircuited, the voltage will drop, outputs will turn OFF, and the system may be unable to read inputs. Provide external safety measures in controls with monitoring of external power supply voltage as required so that the system operates safely in such a case.

Unintended outputs may occur when an error occurs in variable memory or in memory used for CJ-series Units. As a countermeasure for such problems, external safety measures must be provided to ensure safe operation of the system.

Provide measures in the communications system and user program to ensure safety in the overall system even if errors or malfunctions occur in data link communications or remote I/O communications.

If there is interference in remote I/O communications or if a major fault level error occurs, output status will depend on the products that are used.
Confirm the operation that will occur when there is interference in communications or a major fault level error, and implement safety measures.
Correctly set all of the settings in the slaves and Units.

The NJ-series Controller continues normal operation for a certain period of time when a momentary power interruption occurs. This means that the NJ-series Controller may receive incorrect signals from external devices that are also affected by the power interruption. Accordingly, take suitable actions, such as external fail-safe measures and interlock conditions, to monitor the power supply voltage of the external device as required.

You must take fail-safe measures to ensure safety in the event of incorrect, missing, or abnormal signals caused by broken signal lines, momentary power interruptions, or other causes. Not doing so may result in serious accidents due to incorrect operation.

## Voltage and Current Inputs

Make sure that the voltages and currents that are input to the slaves and Units are within the specified ranges.
Inputting voltages or currents that are outside of the specified ranges may cause accidents or fire.

## Downloading

Always confirm safety at the destination before you transfer a user program, configuration data, setup data, device variables, or values in memory used for CJ-series Units from the Sysmac Studio. The devices or machines may perform unexpected operation regardless of the operating mode of the CPU Unit.

## Actual Operation

Check the user program, data, and parameter settings for proper execution before you use them for actual operation.

## Cautions

# ⚠ Caution

### Application

Do not touch any Unit when power is being supplied or immediately after the power supply is turned OFF. Doing so may result in burn injury.

### Wiring

Be sure that all terminal screws and cable connector screws are tightened to the torque specified in the relevant manuals. The loose screws may result in fire or malfunction.

### Online Editing

Execute online editing only after confirming that no adverse effects will be caused by deviations in the timing of I/O. If you perform online editing, the task execution time may exceed the task period, I/O may not be refreshed with external devices, input signals may not be read, and output timing may change.

### Precaution on Error Message That Says an Instruction May Cause Unintended Operation

Instructions may results in unexpected operation and affect the system if you clear the selection of the *Detect an error when an in-out variable is passed to specific instruction argument* Check Box in the Program Check Area. Always confirm that the conditions for use that are given in the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502) are met before you clear the selection of this check box.

**Version Information**

This error message is displayed by and the above option setting is available on Sysmac Studio version 1.02.

# Precautions for Safe Use

## Disassembly and Dropping

- Do not attempt to disassemble, repair, or modify any Units. Doing so may result in malfunction or fire.
- Do not drop any Unit or subject it to abnormal vibration or shock. Doing so may result in Unit malfunction or burning.

## Mounting

- The sliders on the tops and bottoms of the Power Supply Unit, CPU Unit, I/O Units, and other Units must be completely locked (until they click into place) after connecting the adjacent Unit connectors.

## Installation

- Always connect to a ground of 100 Ω or less when installing the Units. A ground of 100 Ω or less must be installed when shorting the GR and LG terminals on the Power Supply Unit.

## Wiring

- Follow the instructions in this manual to correctly perform wiring.
  Double-check all wiring and switch settings before turning ON the power supply.
- Use crimp terminals for wiring.
  Do not connect bare stranded wires directly to terminals.
- Do not pull on the cables or bend the cables beyond their natural limit.
  Do not place heavy objects on top of the cables or other wiring lines. Doing so may break the cables.
- Mount terminal blocks and connectors only after checking the mounting location carefully.
  Be sure that the terminal blocks, expansion cables, and other items with locking devices are properly locked into place.
- Always remove any dustproof labels that are on the top of the Units when they are shipped before you turn ON the power supply. If the labels are not removed, heat will accumulate and malfunctions may occur.
- Before you connect a computer to the CPU Unit, disconnect the power supply plug of the computer from the AC outlet. Also, if the computer has an FG terminal, make the connections so that the FG terminal has the same electrical potential as the GR terminal on the Power Supply Unit. A difference in electrical potential between the computer and Controller may cause failure or malfunction.
- If the external power supply to an Output Unit or slave has polarity, connect it with the correct polarity. If the polarity is reversed, current may flow in the reverse direction and damage the connected devices regardless of the operation of the Controller.

## Power Supply Design

- Do not exceed the rated supply capacity of the Power Supply Units in the NJ-series Controller. The rated supply capacities are given in the *NJ-series CPU Unit Hardware User’s Manual* (Cat. No. W500).
  If the capacity is exceeded, operation may stop, malfunctions may occur, or data may not be backed up normally for power interruptions.
  Use NJ-series Power Supply Units for both the NJ-series CPU Rack and Expansion Racks.
  Operation is not possible if a CJ-series Power Supply Unit is used with an NJ-series CPU Unit or an NJ-series Power Supply Unit is used with a CJ-series CPU Unit.

- Do not apply voltages or connect loads to the Output Units or slaves in excess of the maximum ratings.
- Surge current occurs when the power supply is turned ON. When selecting fuses or breakers for external circuits, consider the above precaution and allow sufficient margin in shut-off performance. Refer to the relevant manuals for surge current specifications. Refer to the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) for surge current specifications.
- If the full dielectric strength voltage is applied or turned OFF using the switch on the tester, the generated impulse voltage may damage the Power Supply Unit. Use the adjustment on the tester to gradually increase and decrease the voltage.
- Apply the voltage between the Power Supply Unit's L1 or L2 terminal and the GR terminal when testing insulation and dielectric strength.
- Do not supply AC power from an inverter or other device with a square-wave output. Internal temperature rise may result in smoking or burning. Always input a sinusoidal wave with the frequency that is given in the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500).
- Install external breakers and take other safety measures against short-circuiting in external wiring.

## Turning ON the Power Supply

- It takes up to approximately 10 to 20 s to enter RUN mode after the power is turned ON. The outputs during this time behave according to the slave or Unit specifications. Use the RUN output on the Power Supply Unit, for example, to implement fail-safe circuits so that external devices do not operate incorrectly.
- Configure the external circuits so that the power supply to the control system turns ON only after the power supply to the Controller has turned ON. If the power supply to the Controller is turned ON after the control power supply, temporary errors may result in incorrect control system signals because the output terminals on Output Units may momentarily turn ON when power supply is turned ON to the Controller.
- If you transfer data from a backup file on an SD Memory Card to the Controller when the power supply is turned ON, properly select the data groups to transfer. If the data for an unintended data group is transferred to the Controller, it may cause the equipment to operate unpredictably.
- Never turn OFF the power supply to the Controller until RUN mode is entered after the power is turned ON. If the power supply is turned OFF, a Battery-backup Memory Check Error may occur at next time you start operation. If a Battery-backup Memory Check Error occurs, the variables retained are set to their initial values and the Holding, DM and EM Areas in memory used for CJ-series Units are cleared to all zeros. If you want to resume the operation, reload the correct data for the variables retained and CJ-series Unit memory, as required.

## Turning OFF the Power Supply

- Never turn OFF the power supply to the Controller when the BUSY indicator is flashing. While the BUSY indicator is lit, the user program and settings in the CPU Unit are being backed up in the built-in non-volatile memory. This data will not be backed up correctly if the power supply is turned OFF. Also, a major fault level Controller error will occur the next time you start operation, and operation will stop.
- Do not turn OFF the power supply or remove the SD Memory Card while SD Memory Card access is in progress (i.e., while the SD BUSY indicator flashes). Data may become corrupted, and the Controller will not operate correctly if it uses corrupted data. To remove the SD Memory Card from the CPU Unit while the power supply is ON, press the SD Memory Card power supply switch and wait for the SD BUSY indicator to turn OFF before you remove the SD Memory Card.
- Do not disconnect the cable or turn OFF the power supply to the Controller when downloading data or the user program from Support Software.
- Always turn OFF the power supply to the Controller before you attempt any of the following.
  - Mounting or removing I/O Units or the CPU Unit
  - Assembling the Units

- Setting DIP switches or rotary switches
- Connecting cables or wiring the system
- Connecting or disconnecting the connectors

The Power Supply Unit may continue to supply power to the rest of the Controller for a few seconds after the power supply turns OFF. The PWR indicator is lit during this time. Confirm that the PWR indicator is not lit before you perform any of the above.

## Operation

- Confirm that no adverse effect will occur in the system before you attempt any of the following.
  - Changing the operating mode of the CPU Unit (including changing the setting of the Operating Mode at Startup)
  - Changing the user program or settings
  - Changing set values or present values
  - Forced refreshing
- After you change any slave or Unit settings, carefully check the safety of the controlled system before you restart the Unit.
- If two different function modules are used together, such as when you use CJ-series Basic Units and EtherCAT slaves, take suitable measures in the user program and external controls to ensure that safety is maintained in the controlled system if one of the function modules stops. The relevant outputs will behave according to the slave or Unit specifications if a partial fault level error occurs in one of the function modules.
- Always confirm safety at the connected equipment before you reset Controller errors with an event level of partial fault or higher for the EtherCAT Master Function Module.
  When the error is reset, all slaves that were in any state other than Operational state due to a Controller error with an event level of partial fault or higher (in which outputs are disabled) will go to Operational state and the outputs will be enabled.
  Before you reset all errors or restart a slave, confirm that no Controller errors with an event level of partial fault have occurred for the EtherCAT Master Function Module.
- Always confirm safety at the connected equipment before you reset Controller errors for a CJ-series Special Unit. When a Controller error is reset, the Unit where the Controller error with an event level of observation or higher will be restarted.
  Before you reset all errors, confirm that no Controller errors with an event level of observation or higher have occurred for the CJ-series Special Unit. Observation level events do not appear on the Controller Error Tab Page, so it is possible that you may restart the CJ-series Special Unit without intending to do so.
  You can check the status of the *_CJB_UnitErrSta[0,0]* to *_CJB_UnitErrSta[3,9]* error status variables on a Watch Tab Page to see if an observation level Controller error has occurred.

## Battery Backup

- The user program and initial values for the variables are stored in non-volatile memory in the CPU Unit. The present values of variables with the Retain attribute and the values of the Holding, DM, and EM Areas in the memory used for CJ-series Units are backed up by a Battery. If the Battery is not connected or the Battery is exhausted, the CPU Unit detects a Battery-backup Memory Check Error. If that error is detected, variables with a Retain attribute are set to their initial values and the Holding, DM, and EM Areas in memory used for CJ-series Units are cleared to all zeros. Perform thorough verifications and provide sufficient measures to ensure that the devices perform safe operation for the initial values of the variables with Retain attributes and the resulting operation.
- The absolute encoder home offsets are backed up by a Battery. If the CPU Unit detects a low battery voltage or the absence of a mounted battery when the power supply to the Controller is turned ON, the absolute encoder home offsets are cleared to zeros and an Encoder Home Offset Read Error occurs. Reset the error and perform homing to define home. If you do not define home, unintended operation of the controlled system may occur.

## Debugging

- Forced refreshing ignores the results of user program execution and refreshes I/O with the specified values. If forced refreshing is used for inputs for which I/O refreshing is not supported, the inputs will first take the specified values, but they will then be overwritten by the user program. This operation differs from the force-set/reset functionality of the CJ-series PLCs.
- You cannot upload or download information for forced refreshing with the Sysmac Studio.
  After downloading data that contains forced refreshing, change to RUN mode and then use the Sysmac Studio to perform the operation for forced refreshing.
  Depending on the difference in the forced status, the control system may operate unexpectedly.
- Do not specify the same address for the AT specification for more than one variable.
  Doing so would allow the same entity to be accessed with different variable names, which would make the user program more difficult to understand and possibly cause programming mistakes.

## General Communications

- When you use data link communications, check the error information that is given in *_ErrSta* (Controller Error Status) to make sure that no error has occurred in the source device. Create a user program that uses reception data only when there is no error in the source device. If there is an error in the source device, the data for the data link may contain incorrect values.
- Unexpected operation may result if inappropriate data link tables are set. Even if appropriate data link tables have been set, confirm that the controlled system will not be adversely affected before you transfer the data link tables. The data links start automatically after the data link tables are transferred.
- All CPU Bus Units are restarted when routing tables are transferred from Support Software to the CPU Unit. Confirm that the system will not be adversely affected by restarting before you transfer the routing tables.
- Tag data links will stop between related nodes while tag data link parameters are transferred during Controller operation. Confirm that the system will not be adversely affected before you transfer the tag data link parameters.

## EtherNet/IP Communications

- All related EtherNet/IP nodes are reset when you transfer the settings for the CJ1W-EIP21 EtherNet/IP Unit (including IP addresses and tag data links settings) from the Support Software. Confirm that the system will not be adversely affected by resetting nodes before you transfer the settings.
- If EtherNet/IP tag data links (cyclic communications) are used with a repeating hub, the communications load on the network will increase. This will increase collisions and may prevent stable communications. Do not use repeating hubs on networks where tag data links are used. Use an Ethernet switch instead.

## EtherCAT Communications

- Make sure that the communications distance, number of nodes connected, and method of connection for EtherCAT are within specifications.
  Do not connect EtherCAT communications to EtherNet/IP, a standard in-house LAN, or other networks. An overload may cause the network to fail or malfunction.
- Malfunctions or unexpected operation may occur for some combinations of EtherCAT revisions of the master and slaves. If you disable the revision check in the network settings, use the Sysmac Studio to check the slave revision settings in the master and the actual slave revisions, and then make sure that functionality is compatible in the slave manuals or other references. You can check the actual slave revisions from the Sysmac Studio or on slave nameplates.

- After you transfer the user program, the CPU Unit is restarted and communications with the EtherCAT slaves are cut off. During that period, the slave outputs behave according to the slave specifications. The time that communications are cut off depends on the EtherCAT network configuration.
- If the Fail-soft Operation parameter is set to stop operation, process data communications will stop for all slaves when an EtherCAT communications error is detected in a slave. The Servo Drive will operate according to the Servo Drive specifications. Make sure that the Fail-soft Operation parameter setting results in safe operation when a device error occurs.
- EtherCAT communications are not always established immediately after the power supply is turned ON. Use the system-defined variables in the user program to confirm that communications are established before attempting control operations.
- If noise occurs or an EtherCAT slave is disconnected from the network, any current communications frames may be lost. If frames are lost, slave I/O data is not communicated, and unintended operation may occur. The slave outputs will behave according to the slave specifications. For details, refer to relevant manuals for each slave. If a noise countermeasure or slave replacement is required, perform the following processing.
    - Program the *_EC_InDataInvalid* (Input Data Disable) system-defined variable as an interlock condition in the user program.
    - Set the PDO communications timeout detection count setting in the EtherCAT master to at least 2. Refer to the *NJ/NX-series CPU Unit Built-in EtherCAT Port User’s Manual* (Cat. No. W505) for details.
- When an EtherCAT slave is disconnected or disabled, communications will stop and control of the outputs will be lost not only for the disconnected slave, but for all slaves connected after it. Confirm that the system will not be adversely affected before you disconnect or disable a slave.
- NX bus communications are not always established immediately after the power supply is turned ON. Use the system-defined variables and the EtherCAT Coupler Unit device variables in the user program to confirm that communications are established before attempting control operations.
- You cannot use standard Ethernet hubs or repeater hubs with EtherCAT communications. If you use one of these, a major fault level error or other error may occur.

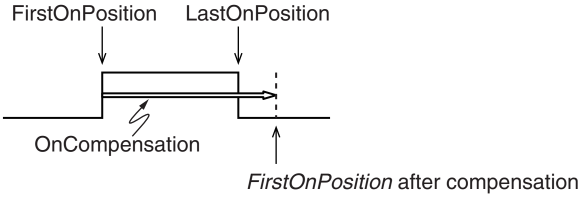
## Motion Control

- Confirm the axis number carefully before you perform an MC Test Run.
- The motor is stopped if communications are interrupted between the Sysmac Studio and the CPU Unit during an MC Test Run. Connect the communications cable between the computer and CPU Unit securely and confirm that the system will not be adversely affected before you perform an MC Test Run.
- Always execute the Save Cam Table instruction if you change any of the cam data from the user program in the CPU Unit or from the Sysmac Studio. If the cam data is not saved, the previous condition will be restored when the power is turned ON again, possibly causing unexpected machine operation.
- The positive drive prohibit input (POT), negative drive prohibit input (NOT), and home proximity input (DEC) of the Servo Drive are used by the MC Function Module as the positive limit input, negative limit input, and home proximity input. Make sure that the signal widths for all of these input signals are longer than the control period of the MC Function Module. If the input signal widths are shorter than the control period, the MC Function Module may not be able to detect the input signals, resulting in incorrect operation.
- If you make any changes in the Detailed Settings Area of the Axis Basic Settings Display of the Sysmac Studio, make sure that the devices or machines perform the expected operation before you start actual operation.

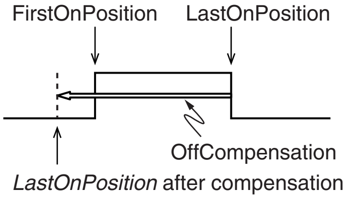
  If the relationship between the functions of the Motion Control Function Module and the EtherCAT slave process data that is assigned to the axes is not correct, the devices or machines may perform unexpected operation.
- Always use the axis at a constant velocity for the MC_DigitalCamSwitch (Enable Digital Cam Switch) instruction.

  If you set the Count Mode to Rotary Mode, the following operation will occur if you use *OnCompensation* or *OffCompensation* and the axis velocity changes abruptly.

  - If the value of *OnCompensation* or *OffCompensation* is equivalent to the time for half a rotation or more, *InOperation* will be FALSE.
  - If the value of *OnCompensation* results in exceeding *LastOnPosition*, the output timing will be unstable.



  - If the value of *OffCompensation* results in exceeding *FirstOnPosition*, the output timing will be unstable.



- Use the NX_AryDOutTimeStamp (Write Digital Output Array with Specified Time Stamp) instruction only after you confirm that *InOperation* from the MC_DigitalCamSwitch (Enable Digital Cam Switch) instruction is TRUE.

## Restoring Data

- You cannot back up, restore, or compare some or all of the settings for certain slaves and Units. Also, you cannot back up, restore, or compare data for disabled slaves or Units. After you restore data, sufficiently confirm that operation is correct before you start actual operation.
- The absolute encoder home offsets are backed up with a Battery in the CPU Unit as absolute encoder information. If any of the following conditions is met, clear the absolute encoder home offsets from the list of data items to restore, and then restore the data. Then, define the absolute encoder home again. If you do not define home, unintended operation of the controlled system may occur.
  - The Servomotor or Servo Drive was changed since the data was backed up.
  - The absolute encoder was set up after the data was backed up.
  - The absolute data for the absolute encoder was lost.

## Battery Replacement

- The Battery may leak, rupture, heat, or ignite. Never short-circuit, charge, disassemble, heat, or incinerate the Battery or subject it to strong shock.
- Dispose of any Battery that has been dropped on the floor or otherwise subjected to excessive shock. Batteries that have been subjected to shock may leak if they are used.
- UL standards require that only an experienced engineer replace the Battery. Make sure that an experienced engineer is in charge of Battery replacement.
- Apply power for at least five minutes before changing the Battery. Install a new Battery within five minutes (at 25°C) of turning OFF the power supply. If power is not supplied for at least 5 minutes, the saved data may be lost.
- Make sure that the required data, including the user program, configurations, settings, variables, and memory used for CJ-series Units, is transferred to a CPU Unit that was replaced and to externally connected devices before restarting operation.
  Be sure to include the tag data link settings, routing tables, and other CPU Bus Unit data, which are stored in the CPU Unit.

## Unit Replacement

- We recommend replacing the Battery with the power turned OFF to prevent the CPU Unit’s sensitive internal components from being damaged by static electricity and to prevent malfunctions. The Battery can be replaced without turning OFF the power supply. To do so, always touch a grounded piece of metal to discharge static electricity from your body before you start the procedure.
  After you replace the Battery, connect the Sysmac Studio and clear the Low Battery Voltage error.
- The absolute encoder home offsets are backed up with a Battery in the CPU Unit as absolute encoder information. When you change the combination of the CPU Unit and Servomotor, e.g., when you add or replace a Servomotor, define the absolute encoder home again.

## Disposal

- Dispose of the product and Batteries according to local ordinances as they apply.

  廢電池請回收
- The following information must be displayed for all products that contain primary lithium batteries with a perchlorate content of 6 ppb or higher when shipped to or transported through the State of California, USA.

  Perchlorate Material - special handling may apply.
  See www.dtsc.ca.gov/hazardouswaste/perchlorate.
- The CPU Unit contains a primary lithium battery with a perchlorate content of 6 ppb or higher. Place the above information on the individual boxes and shipping boxes when shipping finished products that contain a CPU Unit to the State of California, USA.

# Precautions for Correct Use

## Storage, Mounting, and Wiring

- Do not operate or store the Controller in the following locations. Operation may stop or malfunctions may occur.
  - Locations subject to direct sunlight
  - Locations subject to temperatures or humidity outside the range specified in the specifications
  - Locations subject to condensation as the result of severe changes in temperature
  - Locations subject to corrosive or flammable gases
  - Locations subject to dust (especially iron dust) or salts
  - Locations subject to exposure to water, oil, or chemicals
  - Locations subject to shock or vibration
- Take appropriate and sufficient countermeasures when installing the Controller in the following locations.
  - Locations subject to strong, high-frequency noise
  - Locations subject to static electricity or other forms of noise
  - Locations subject to strong electromagnetic fields
  - Locations subject to possible exposure to radioactivity
  - Locations close to power lines
- Before touching a Unit, be sure to first touch a grounded metallic object in order to discharge any static build-up.
- Install the Controller away from sources of heat and ensure proper ventilation. Not doing so may result in malfunction, in operation stopping, or in burning.
- An I/O bus check error will occur and the Controller will stop if an I/O Connecting Cable’s connector is disconnected from the Rack. Be sure that the connectors are secure.
- Do not allow foreign matter to enter the openings in the Unit. Doing so may result in Unit burning, electric shock, or failure.
- Do not allow wire clippings, shavings, or other foreign material to enter any Unit. Otherwise, Unit burning, failure, or malfunction may occur. Cover the Units or take other suitable countermeasures, especially during wiring work.
- For EtherCAT and EtherNet/IP, use the connection methods and cables that are specified in the *NJ/NX-series CPU Unit Built-in EtherCAT Port User’s Manual* (Cat. No. W505) and the *NJ/NX-series CPU Unit Built-in EtherNet/IP Port User’s Manual* (Cat. No. W506). Otherwise, communications may be faulty.
- Use the rated power supply voltage for the Power Supply Units. Take appropriate measures to ensure that the specified power with the rated voltage and frequency is supplied in places where the power supply is unstable.
- Make sure that the current capacity of the wire is sufficient. Otherwise, excessive heat may be generated. When cross-wiring terminals, the total current for all the terminals will flow in the wire. When wiring cross-overs, make sure that the current capacity of each of the wires is not exceeded.
- Do not touch the terminals on the Power Supply Unit immediately after turning OFF the power supply. Residual voltage may cause electrical shock.
- If you use reed switches for the input contacts for AC Input Units, use switches with a current capacity of 1 A or greater.
  If the capacity of the reed switches is too low, surge current may fuse the contacts.

## Error Processing

- In applications that use the results of instructions that read the error status, consider the affect on the system when errors are detected and program error processing accordingly. For example, even the detection of a minor error, such as Battery replacement during operation, can affect the system depending on how the user program is written.
- If you change the event level of a Controller error, the output status when the error occurs may also change. Confirm safety before you change an event level.

## Restoring and Automatically Transferring Data

- When you edit the restore command file or the automatic transfer command file, do not change anything in the file except for the “yes” and “no” specifications for the selectable data groups. If you change anything else in the file, the Controller may perform unexpected operation when you restore or automatically transfer the data.

## Replacing Slaves and Units

- If you replace a slave or Unit, refer to the operation manual for the slave or Unit for information on the data required for individual slaves or Units and redo the necessary settings.

## Task Settings

- If a Task Period Exceeded error occurs, shorten the programs to fit in the task period or increase the setting of the task period.

## Motion Control

- Use the system-defined variable in the user program to confirm that EtherCAT communications are established before you attempt to execute motion control instructions. Motion control instructions are not executed normally if EtherCAT communications are not established.
- Use the system-defined variables to monitor for errors in communications with the slaves that are controlled by the motion control function module. Motion control instructions are not executed normally if an error occur in slave communications.
- Before you start an MC Test Run, make sure that the operation parameters are set correctly.
- Do not download motion control settings during an MC Test Run.

## EtherCAT Communications

- If you need to disconnect the cable from an EtherCAT slave during operation, first disconnect the software connection to the EtherCAT slave or disable the EtherCAT slave and all of the EtherCAT slaves that are connected after it.
- Set the Servo Drives to stop operation if an error occurs in EtherCAT communications between the Controller and a Servo Drive.
- Make sure that all of the slaves to be restored are participating in the network before you reset a Network Configuration Verification Error, Process Data Communications Error, or Link OFF Error in the EtherCAT Master Function Module. If any slave is not participating when any of these errors is reset, the EtherCAT Master Function Module may access slave with a different node address than the specified node address or the error may not be reset correctly.
- Always use the specified EtherCAT slave cables. If you use any other cable, the EtherCAT master or the EtherCAT slaves may detect an error and one of the following may occur.
  - Continuous refreshing of process data communications will not be possible.
  - Continuous refreshing of process data communications will not end during the set cycle.

## Battery Replacement

- Be sure to install a replacement Battery within two years of the production date shown on the Battery label.
- Turn ON the power after replacing the battery for a CPU Unit that has been unused for a long time. Leaving the CPU Unit unused again without turning ON the power even once after the battery is replaced may result in a shorter battery life.
- When you replace the Battery, use the CJ1W-BAT01 Battery Set.

## SD Memory Cards

- Insert the SD Memory Card all the way.
- Do not turn OFF the power supply to the Controller during SD Memory Card access. The files may be corrupted.
  If there is a corrupted file in the SD Memory Card, the file is automatically deleted by the restoration function when the power supply is turned ON.

## Online Editing

When performing online editing in combination of a CPU Unit with a unit version of 1.04 or later and Sysmac Studio version 1.05 or higher, the CPU Unit saves a program updated by the online editing to built-in non-volatile memory. Sysmac Studio shows a message that it is in a backup operation. Do not turn OFF the power supply to the Controller while this message is displayed. If the power supply to the Controller is turned OFF, a Controller error will occur when the power supply is turned ON next time.

# Regulations and Standards

## Conformance to EMC and Electrical Safety Regulations

### Concepts

OMRON products are industrial electrical devices that are incorporated into various types of machines and manufacturing equipment. The products conform to the relevant standards so that the machines and equipment incorporating the OMRON products can comply with EMC and Electrical Safety Regulations more easily.

Refer to the OMRON website (www.ia.omron.com) or ask your OMRON representative for the most recent standards to which our products conform.

#### ● Conformance to EMC regulations

This product complies with EMC regulations when assembled in a PLC system or Machine Automation Controller.

To ensure that your machine or equipment complies with EMC regulations, please observe the following precautions.

- This product is defined as an in-panel device and must be installed within a control panel.
- This product complies with the emission standards. For the radiated emission requirements, in particular, please note that the actual emission varies depending on the configuration of the control panel to be used, the connected devices, and wiring methods. Therefore, customers themselves must confirm that the entire machine or equipment conforms to EMC regulations, even you are using a device that conforms to EMC regulations.
- You must use reinforced insulation or double insulation for the DC power supplies connected to DC Power Supply Units and I/O Units.

Caution:

This equipment is not intended for use in residential environments and may not provide adequate protection to radio reception in such environments.

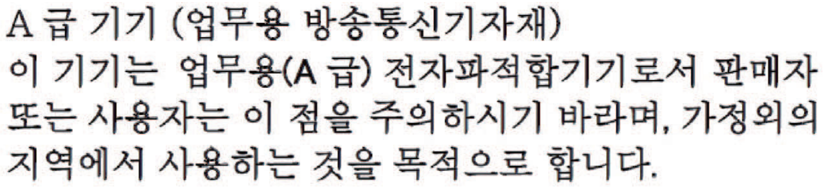
#### ● Conformance to Electrical Safety regulations

This product complies with Electrical Safety regulations required by specific laws and regulations such as the EU Directives and UKCA.

For precautions for each product, see the instruction manual included with the product.

## Conformance to KC Standards

Observe the following precaution if you use NX-series Units in Korea.



A 급 기기 (업무용 방송통신기자재)
이 기기는 업무용(A 급) 전자파적합기기로서 판매자
또는 사용자는 이 점을 주의하시기 바라며, 가정외의
지역에서 사용하는 것을 목적으로 합니다.

Class A Device (Broadcasting Communications Device for Office Use)

This device obtained EMC registration for office use (Class A), and it is intended to be used in places other than homes.

Sellers and/or users need to take note of this.

## 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 |

## 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/.

# Versions

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

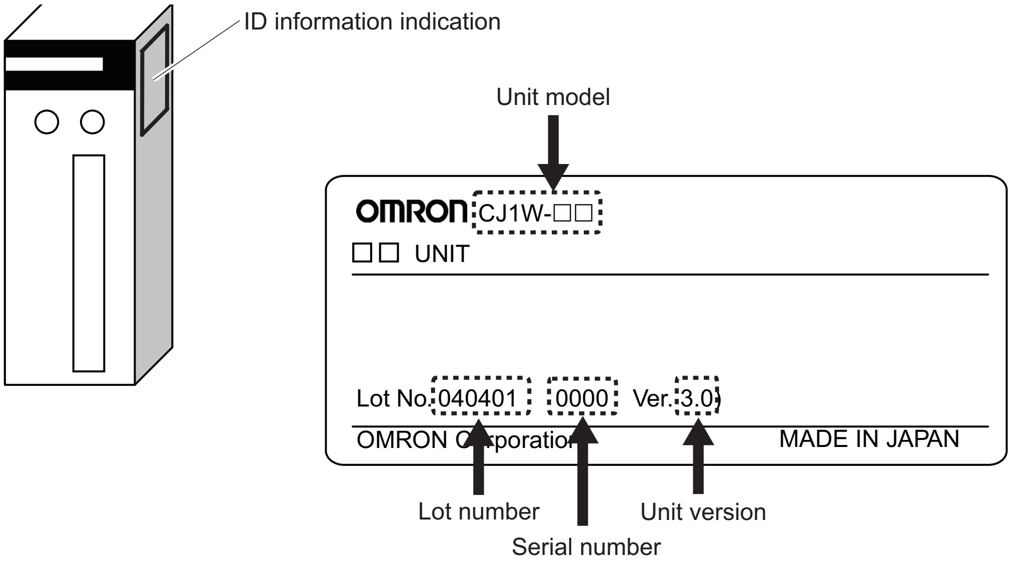
## Checking Versions

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

### Checking Unit Versions on ID Information Indications

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

**CJ1W-EIP21/EIP21S**



The following information is provided on the ID information indication.

| Item | Description |
|---|---|
| Unit model | Gives the model of the Unit. |
| Lot number and serial number | Gives the lot number and serial number of the Unit.<br>YYMMDD: Lot number, □: For use by OMRON, xxxx: Serial number<br>YY indicates the last two digits of the year, MM the month, and DD the day. |
| Unit version | Gives the unit version of the Unit. |

## Checking Unit Versions with the Sysmac Studio

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

### ● Checking the Unit Version of a Unit

You can use the Production Information while the Sysmac Studio is online to check the unit version of a Unit. You can do this for the CPU Unit, CJ-series Special I/O Units, and CJ-series CPU Bus Units. You cannot check the unit versions of CJ-series Basic I/O Units with the Sysmac Studio.

Use the following procedure to check the unit version.

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.

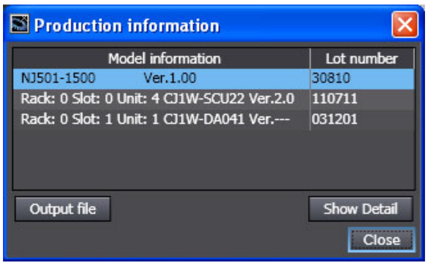
2. Right-click any open space in the Unit Editor and select ***Production Information***.

   The Production Information Dialog Box is displayed.

### ● Changing Information Displayed in Production Information Dialog Box

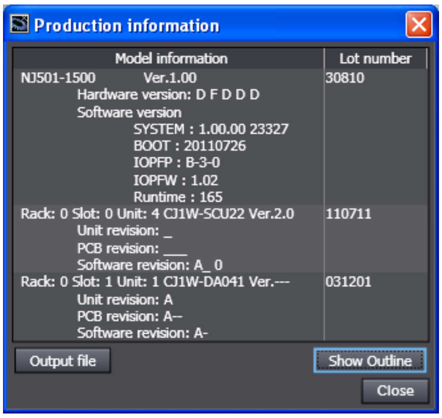
1. Click the **Show Detail** or **Show Outline** Button at the lower right of the Production Information Dialog Box.

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



| Model information | Lot number |
|---|---|
| NJ501-1500 Ver.1.00 | 30810 |
| Rack: 0 Slot: 0 Unit: 4 CJ1W-SCU22 Ver.2.0 | 110711 |
| Rack: 0 Slot: 1 Unit: 1 CJ1W-DA041 Ver.--- | 031201 |

Outline View



| Model information | Lot number |
|---|---|
| NJ501-1500 Ver.1.00<br>Hardware version: D F D D D<br>Software version<br>SYSTEM : 1.00.00 23327<br>BOOT : 20110726<br>IOPFP : B-3-0<br>IOPFW : 1.02<br>Runtime : 165 | 30810 |
| Rack: 0 Slot: 0 Unit: 4 CJ1W-SCU22 Ver.2.0<br>Unit revision: _<br>PCB revision: ___<br>Software revision: A_ 0 | 110711 |
| Rack: 0 Slot: 1 Unit: 1 CJ1W-DA041 Ver.---<br>Unit revision: A<br>PCB revision: A--<br>Software revision: A- | 031201 |

Detail View

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

### ● Checking the Unit Version of an EtherCAT Slave

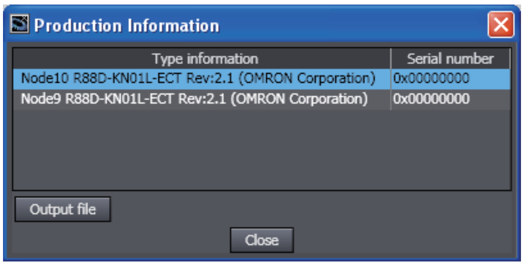
You can use the Production Information while the Sysmac Studio is online to check the unit version of an EtherCAT slave. Use the following procedure to check the unit version.

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

The EtherCAT Tab Page is displayed.

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

The Production Information Dialog Box is displayed.
The unit version is displayed after “Rev.”



Production Information

| Type information | Serial number |
|---|---|
| Node10 R88D-KN01L-ECT Rev:2.1 (OMRON Corporation) | 0x00000000 |
| Node9 R88D-KN01L-ECT Rev:2.1 (OMRON Corporation) | 0x00000000 |

## Changing the Unit Version

The following describes the procedure to change the unit version of the CJ1W-EIP21 EtherNet/IP Unit registered in the Sysmac Studio project file.

When you replace your EtherNet/IP Unit with one with a different unit version, you can transfer the settings of the previous Unit by following the steps below.

**1** Open a controller project that contains the settings for a CJ1W-EIP21 with a unit version of 2.1 on the Sysmac Studio.

**2** Right-click CJ1W-EIP21 Ver2.1 in the CPU/Expansion Racks Tab Page and select ***Save Special Unit Settings*** from the menu. This saves the settings in the parameter file with an UPF file name extension.

**3** The procedure in this step varies depending on the conditions below.

- You used other applications than the Sysmac Studio to make the EtherNet/IP connection settings (tag data link settings)

**(1) Open the I/O Map on the Sysmac Studio. Copy the CJ1W-EIP21 variables in the I/O Map and paste them to a Microsoft Excel worksheet. Then save the file.**

**(2) Right-click CJ1W-EIP21 Ver2.1 in the CPU/Expansion Racks Tab Page and select *Change Model* from the menu. This changes the CJ1W-EIP21 unit version in the configuration to 3.0.**

**(3) Copy the variables pasted on the Microsoft Excel worksheet. Open the I/O Map on the Sysmac Studio and paste the copied variables in the Variable field for the CJ1W-EIP21. This completes the restoration.**

- You used the Sysmac Studio to make the EtherNet/IP connection settings (tag data link settings), or you want to prevent from following the former procedures in this step

**(1) Prepare the Controller with the same CPU and Expansion Racks configuration after replacing the CJ1W-EIP21 with it with a unit version of 3.0. Connect the Controller and the Sysmac Studio online.**

**(2) Select Compare and Merge menu in the CPU/Expansion Racks Tab Page to change the CJ1W-EIP21 unit version registered in the controller project from 2.1 to 3.0.**

**4** Right-click CJ1W-EIP21(NJ) Ver3.0 in the CPU/Expansion Racks Tab Page and select ***Read Special Unit Settings*** from the menu. Select the parameter file with the UPF file name extension that is saved in step 2 to import it. This completes the restoration. This completes the unit version change.

## CIP Revision

The following table gives the CIP revision that is supported by the unit version of the EtherNet/IP Unit.

### ● CJ1W-EIP21

| Unit version | CIP revision |
|---|---|
| 2.1 | 2.04 |
| 3.0 | 3.01 |

### ● CJ1W-EIP21S

| Unit version | CIP revision |
|---|---|
| 1.0 | 4.01 |

## Unit Versions and Programming Device Versions

The following versions of the Sysmac Studio and Network Configuration are required to set the EtherNet/IP Unit.

- **CJ1W-EIP21**

| Unit version | Sysmac Studio | | |
|---|---|---|---|
| | Version 1.01 or lower | Version 1.02 | Version 1.11 or higher |
| 2.1 | Not supported. | Supported. | Supported. |
| 3.0 | Not supported. | Not supported. | Supported. |

| Unit version | Network Configurator for EtherNet/IP | | |
|---|---|---|---|
| | Version 3.40 or lower | Version 3.50 | Version 3.57 or higher |
| 2.1 | Not supported. | Supported. | Supported. |
| 3.0 | Not supported. | Not supported. | Supported. |

- **CJ1W-EIP21S**

| Unit version | Sysmac Studio | |
|---|---|---|
| | Version 1.59 or lower | Version 1.60 or higher |
| 1.0 | Not supported. | Supported. |

| Unit version | Network Configurator for EtherNet/IP | |
|---|---|---|
| | Version 3.75 or lower | Version 3.76 or higher |
| 1.0 | Not supported. | Supported. |

## Unit Versions and Applicable CPU Units

The following table gives the CPU Units to which you can connect the EtherNet/IP Unit according to the unit version of the EtherNet/IP Unit.

- **CJ1W-EIP21**

| Unit version | CPU Unit | |
|---|---|---|
| | CJ-series CPU Unit | NJ-series CPU Unit |
| 2.0 or earlier | Supported. | Not supported. |
| 2.1 | Supported. | Supported.* |
| 3.0 or later | Supported. | Supported.* |

* A CPU Unit with unit version 1.01 or later and Sysmac Studio version 1.02 (for unit version 2.1) or 1.11 (for unit version 3.0) are required.

- **CJ1W-EIP21S**

| Unit version | CPU Unit | |
|---|---|---|
| | CJ-series CPU Unit | NJ-series CPU Unit |
| 1.0 (Lot number earlier than 241001□) | Supported. | Not supported. |
| 1.0 (Lot number 241001□ or later)[*1] | Supported. | Supported.[*2] |

[*1] For lot numbers 241001□ or later, "+NJ" is printed at the lower right of the Unit’s front panel. Refer to *1-3-2 Part Names and Functions* for information on the indication printed at the lower right corner of the front panel.

[*2] Use a CPU Unit with unit version 1.67 or later and Sysmac Studio version 1.60 or higher for connection.

# Related Manuals

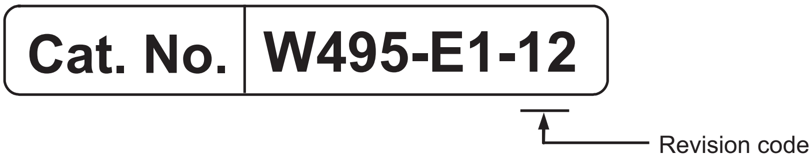
The followings are the manuals related to this manual. Use these manuals for reference.

| Manual name | Cat. No. | Model numbers | Application | Description |
|---|---|---|---|---|
| NX-series CPU Unit Hardware User's Manual | W535 | NX701-□□□□ | Learning the basic specifications of the NX-series CPU Units, including introductory information, designing, installation, and maintenance. Mainly hardware information is provided. | An introduction to the entire NX-series system is provided along with the following information on the CPU Unit.<br>• Features and system configuration<br>• Introduction<br>• Part names and functions<br>• General specifications<br>• Installation and wiring<br>• Maintenance and inspection<br>Use this manual together with the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501). |
| NJ-series CPU Unit Hardware User's Manual | W500 | NJ501-□□□□<br>NJ301-□□□□<br>NJ101-□□□□ | 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 the CPU Unit.<br>• Features and system configuration<br>• Introduction<br>• Part names and functions<br>• General specifications<br>• Installation and wiring<br>• Maintenance and inspection<br>Use this manual together with the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501). |
| NJ/NX-series CPU Unit Software User's Manual | W501 | NX701-□□□□<br>NJ501-□□□□<br>NJ301-□□□□<br>NJ101-□□□□ | Learning how to program and set up an NJ/NX-series CPU Unit. Mainly software information is provided. | The following information is provided on an NJ/NX-series CPU Unit.<br>• CPU Unit operation<br>• CPU Unit features<br>• Initial settings<br>• Programming based on IEC 61131-3 language specifications<br>Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) or *NX-series CPU Unit Hardware User's Manual* (Cat. No. W535). |
| NJ/NX-series Instructions Reference Manual | W502 | NX701-□□□□<br>NJ501-□□□□<br>NJ301-□□□□<br>NJ101-□□□□ | Learning detailed specifications on the basic instructions of an NJ/NX-series CPU Unit. | 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) or *NX-series CPU Unit Hardware User's Manual* (Cat. No. W535) and with the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501). |
| NJ/NX-series CPU Unit Motion Control User's Manual | W507 | NX701-□□□□<br>NJ501-□□□□<br>NJ301-□□□□<br>NJ101-□□□□ | Learning about motion control settings and programming concepts. | The settings and operation of the CPU Unit and programming concepts for motion control are described. When programming, use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) or *NX-series CPU Unit Hardware User's Manual* (Cat. No. W535) and with the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501). |

| Manual name | Cat. No. | Model numbers | Application | Description |
|---|---|---|---|---|
| NJ/NX-series Motion Control Instructions Reference Manual | W508 | NX701-□□□□<br>NJ501-□□□□<br>NJ301-□□□□<br>NJ101-□□□□ | 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) or *NX-series CPU Unit Hardware User's Manual* (Cat. No. W535), and with the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) and *NJ/NX-series CPU Unit Motion Control User's Manual* (Cat. No. W507). |
| NJ/NX-series CPU Unit Built-in EtherCAT® Port User's Manual | W505 | NX701-□□□□<br>NJ501-□□□□<br>NJ301-□□□□<br>NJ101-□□□□ | Using the built-in EtherCAT port on an NJ/NX-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.<br>Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) or *NX-series CPU Unit Hardware User's Manual* (Cat. No. W535) and with the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501). |
| NJ/NX-series CPU Unit Built-in EtherNet/IP$^{TM}$ Port User's Manual | W506 | NX701-□□□□<br>NJ501-□□□□<br>NJ301-□□□□<br>NJ101-□□□□ | Using the built-in EtherNet/IP port on an NJ/NX-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.<br>Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) or *NX-series CPU Unit Hardware User's Manual* (Cat. No. W535) and with the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501). |
| NJ-series Database Connection CPU Units User's Manual | W527 | NJ501-1□20 | Using the database connection service with NJ-series Controllers | Describes the database connection service. |
| NJ-series SECS/GEM CPU Unit User's Manual | W528 | NJ501-1340 | Using the GEM Services with NJ-series Controllers | Information is provided on the GEM Services. |
| NJ/NX-series Troubleshooting Manual | W503 | NX701-□□□□<br>NJ501-□□□□<br>NJ301-□□□□<br>NJ101-□□□□ | Learning about the errors that may be detected in an NJ/NX-series Controller. | Concepts on managing errors that may be detected in an NJ/NX-series Controller and information on individual errors are described.<br>Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) or *NX-series CPU Unit Hardware User's Manual* (Cat. No. W535) and with the *NJ/NX-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. |
| CJ-series EtherNet/IP$^{TM}$ Units Operation Manual for NJ-series CPU Unit | W495 | CJ1W-EIP21<br>CJ1W-EIP21S | Learning how to use the EtherNet/IP Unit | Information on using an EtherNet/IP Unit that is connected to an NJ-series CPU Unit is provided. Information is provided on the basic setup, tag data links, and other features.<br>Use this manual together with the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) and *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501). |

# 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 | March 2012 | Original production |
| 02 | May 2012 | • Added information on functional support for unit version 1.02 and later of the CPU Units.<br>• Corrected mistakes. |
| 03 | August 2012 | Made changes accompanying release of unit version 1.03 of the CPU Unit. |
| 04 | February 2013 | Made changes accompanying release of unit version 1.04 of the CPU Unit. |
| 05 | April 2013 | Corrected mistakes. |
| 06 | November 2014 | The allowable bandwidth was increased to 12,000 pps. |
| 07 | April 2015 | • Added information on the NJ101-□□□□ NJ-series CPU Units.<br>• Corrected mistakes. |
| 08 | April 2018 | Corrected mistakes. |
| 09 | January 2020 | Corrected mistakes. |
| 10 | December 2020 | Corrected mistakes. |
| 11 | May 2023 | Corrected mistakes. |
| 12 | October 2024 | • Made changes accompanying addition of the CJ1W-EIP21S.<br>• Made changes accompanying release of unit version 1.67 of the CPU Unit.<br>• Corrected mistakes. |

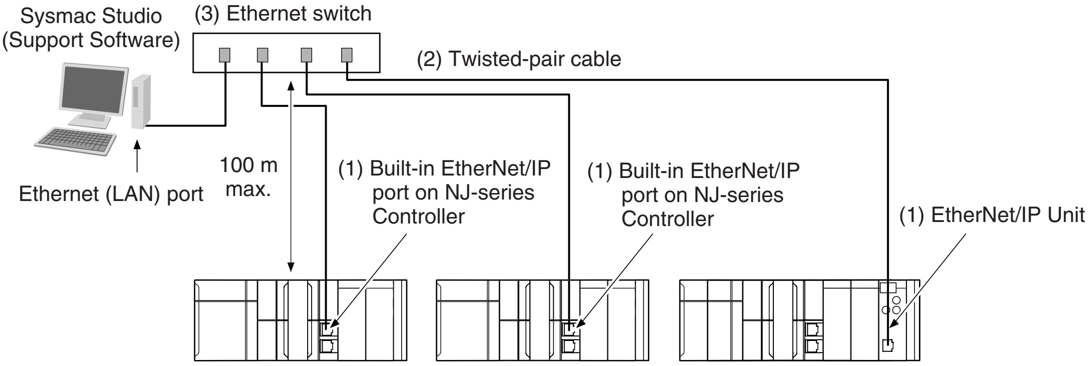
# 1

# Introduction

# 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.

### ● 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 the EtherNet/IP Unit

### ● 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 184,832 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 EtherNet/IP Unit 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 EtherNet/IP Units 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 SNTP 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 SNTP server is necessary to automatically adjust the Controller clocks.

### ● Host Names

You can directly specify IP addresses, but you can also use the host names instead of the IP addresses for SNTP 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 EtherNet/IP Unit 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

## ● Socket Services (CJ1W-EIP21S Only)

Socket services can be used to send and receive data between general-purpose applications and Controllers.

Through the communications services with sockets, you can send and receive data between the host computer and Controllers or between Controllers.

You can use the Send Command instruction in the program to execute various communications processing with the socket services.

There are two socket services, the UDP socket service and TCP socket service.

## ● Security Functions (CJ1W-EIP21S Only)

By opening and closing the port and using the packet filter, you can improve the security performance against intrusion from the outside.

**Additional Information**

CIP (Common Industrial Protocol)

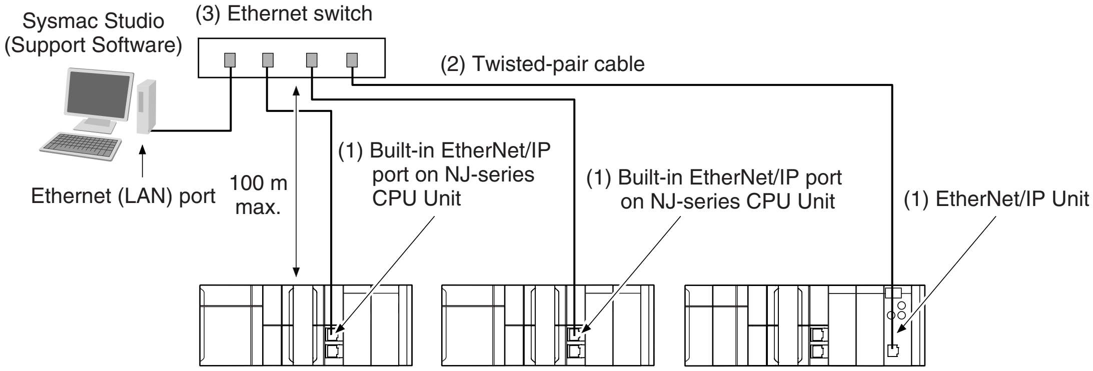
CIP is a shared industrial protocol for the OSI application layer. The CIP is used in networks such as EtherNet/IP, CompoNet, and DeviceNet. Data can be routed easily between networks that are based on the CIP. You can therefore easily configure a transparent network from the field device level to the 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<br>NJ501-□□□□, NJ301-□□□□, or NJ101-□□□□ NJ-series CPU Unit (built-in EtherNet/IP port)<br>CJ-series EtherNet/IP Unit<br>(CJ1W-EIP21[*1]/CJ1W-EIP21S[*2])<br><br>OMRON PLCs<br>CJ2H-CPU□□-EIP or CJ2M-CPU3□ CJ2 CPU Unit (built-in EtherNet/IP port)<br>CS-series EtherNet/IP Unit<br>(CS1W-EIP21/CS1W-EIP21S) | These Units are used to connect to an EtherNet/IP network. |
| (2) Twisted-pair cable | The twisted-pair cable has a RJ45 Modular Connector at each end. This cable is used to connect the built-in EtherNet/IP port or EtherNet/IP Unit to an Ethernet switch. Use an STP (shielded twisted-pair) cable of category 5, 5e, or higher. |
| (3) Ethernet switch | This is a relay device that connects multiple nodes in a star LAN. For details on recommended devices to configure a network, refer to *2-3-1 Recommended Network Devices*. |

[*1] The CJ1W-EIP21 can be used with an NJ-series CPU Unit with unit version 1.01 or later and Sysmac Studio version 1.02 or higher.

[*2] The CJ1W-EIP21S can be used with an NJ-series CPU Unit with unit version 1.67 or later and Sysmac Studio version 1.60 or higher.

### Precautions for Correct Use

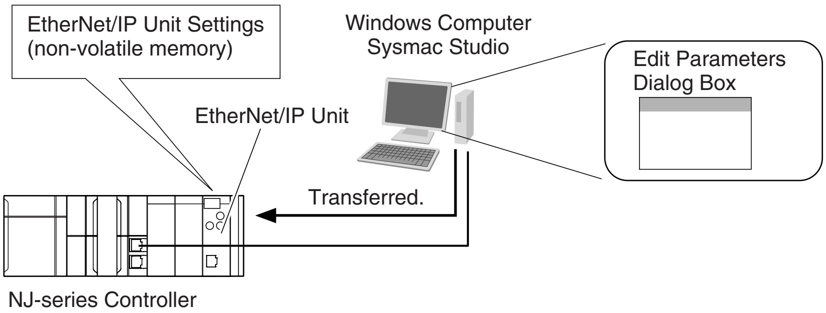
- You cannot place the Sysmac Studio online with an NJ-series CPU Unit if you connect through an EtherNet/IP Unit.
- You cannot perform troubleshooting from an HMI if you connect an HMI to the Controller through an EtherNet/IP Unit.

## 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 EtherNet/IP Unit has Ethernet Settings and Tag Data Link Settings, which are stored in the non-volatile memory in the EtherNet/IP Unit. Support Software is provided for each, as described below.

### ● EtherNet/IP Unit Settings: Sysmac Studio

Use the Sysmac Studio to set the basic settings, such as the IP address and subnet mask of the EtherNet/IP Unit. 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

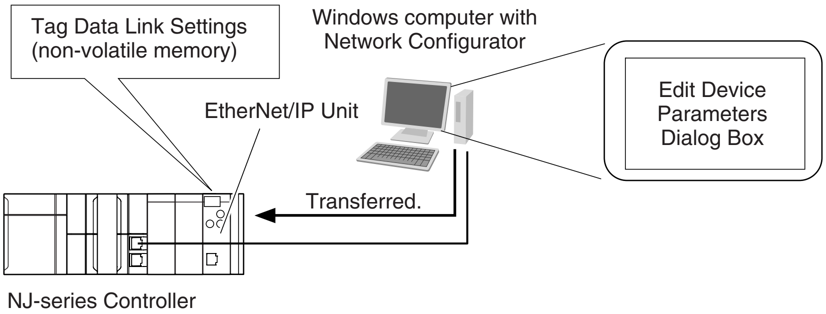
Use the Network Configurator to set the tag data links for the EtherNet/IP Unit. (The Network Configurator is included in the Sysmac Studio Standard Edition.) The main functions of the Network Configurator are given below.

### 1) Setting and Monitoring Tag Data Links (Connections)

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

### 2) Multi-vendor Device Connections

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



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

**Additional Information**

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

# 1-3 EtherNet/IP Unit

## 1-3-1 Specifications

| Item | | Specifications |
|---|---|---|
| **Product** | | EtherNet/IP Unit |
| **Unit classification** | | CPU Bus Unit |
| **Mounting location** | | CPU Rack or Expansion Rack |
| **Number of mountable Units** | | 4 max. |
| **Communications protocol** | | TCP/IP or UDP/IP |
| **Supported services** | | Sysmac Studio connection, tag data link, CIP message communications, FTP server, automatic clock adjustment (SNTP 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 an Ethernet switch is used. |
| **CIP service: Tag data links (cyclic communications)** | **Number of connections** | 256 |
| | **Packet interval (refresh cycle)** | 0.5 to 10,000 ms in 0.5-ms increments<br>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** | 6000 to 12000 pps *2<br>(CJ1W-EIP21 Units with unit version 2.1 or earlier: 6,000 pps)<br>This includes the heartbeat. |
| | **Number of registrable tags** | 256 |
| | **Tag types** | Network variables<br>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** | 369,664 bytes |
| | **Maximum data size per connection** | 1,444 bytes *3<br>**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** | 256 (1 connection = 1 tag set) |
| | **Maximum size of 1 tag set** | 722 words (The Controller status uses 1 word when the tag set contains the Controller status.) |
| | **Changing tag data link parameters when Controller is in RUN mode** | Supported *4 |
| | **Multi-cast packet filter *5** | Supported |

| Item | | Specifications |
|---|---|---|
| CIP message service: Explicit messages [*6] | Class 3 (connected) | Number of connections: 128 (clients + servers) |
| | UCMM (unconnected) | Number of clients that can communicate at one time: 32 max.<br>Number of servers that can communicate at one time: 32 max. |
| | CIP routing [*7] | Supported.<br>CIP routing is supported for the following remote Units: NJ501-□□□□, NJ301-□□□□, NJ101-□□□□, CS1W-EIP21/EIP21S, CJ1W-EIP21/EIP21S, CJ2H-CPU□□-EIP, and CJ2M-CPU3□ |
| SNMP | Agents | SNMPv1 or SNMPv2c |
| | MIB | MIB-II |
| EtherNet/IP conformance test | | Conforms to CT8 |
| Ethernet interface | | 10Base-T or 100Base-TX<br>Auto negotiation or fixed settings |
| Applicable CPU Units | | NJ101-□□□□ CPU Units (unit version 1.10 or later)<br>NJ301-□□□□ CPU Units (unit version 1.01 or later)<br>NJ501-□□□□ CPU Units (unit version 1.01 or later) |
| Unit current consumption | | 5 VDC 410 mA max. |
| Weight | | 94 g max. |
| Dimensions | | 31 × 90 × 65 mm (W × H × D) |

*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, and NJ-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 EtherNet/IP Unit are changed, the Unit is restarted. When other nodes are in communications with the affected node, the communications will temporarily time out and automatically recover after the restart.

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

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

| Service | Protocol | Port number | Remarks |
|---|---|---|---|
| EIP data links | UDP | 2222 | Fixed values |
| CIP messages | TCP/UDP | 44818 | |
| DNS client | UDP | 53 | |
| BOOTP client | UDP | 68 | |
| FTP server<br>(Data transfer port) | TCP | 20<br>Port number:<br>(Control port ) - 1 | You can change the port number in the Unit Settings on the Sysmac Studio. |
| FTP server<br>(Control port) | TCP | 21 | |
| SNTP client | UDP | 123 | |
| SNMP agent | UDP | 161 | |
| SNMP trap | UDP | 162 | |

*7 A CPU Unit with unit version 1.01 or later and Sysmac Studio version 1.02 or higher are required to use CPU routing.

## Restrictions for Installation (CJ1W-EIP21S)

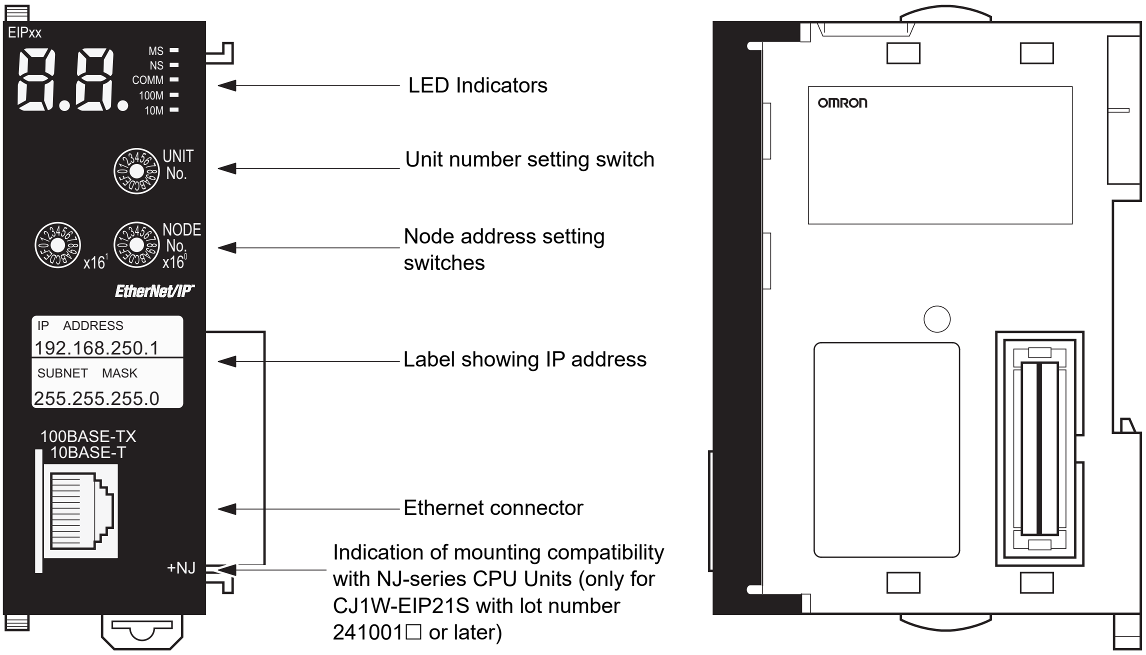
### Coping with Long Unit Startup Time

The Unit startup time of the CJ1W-EIP21S is several seconds longer than that of the CJ1W-EIP21.

After power ON, the CPU Unit can start operation when all the Special I/O Units and CPU Bus Units have been recognized. For this reason, the CPU Unit startup time is several seconds longer in a system that includes the CJ1W-EIP21S than in a system that does not include it.

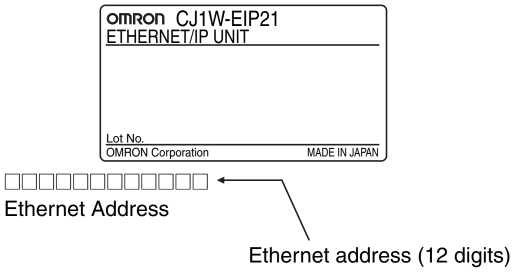
Check the effect on the system’s initial setup operation.

## 1-3-2 Part Names and Functions



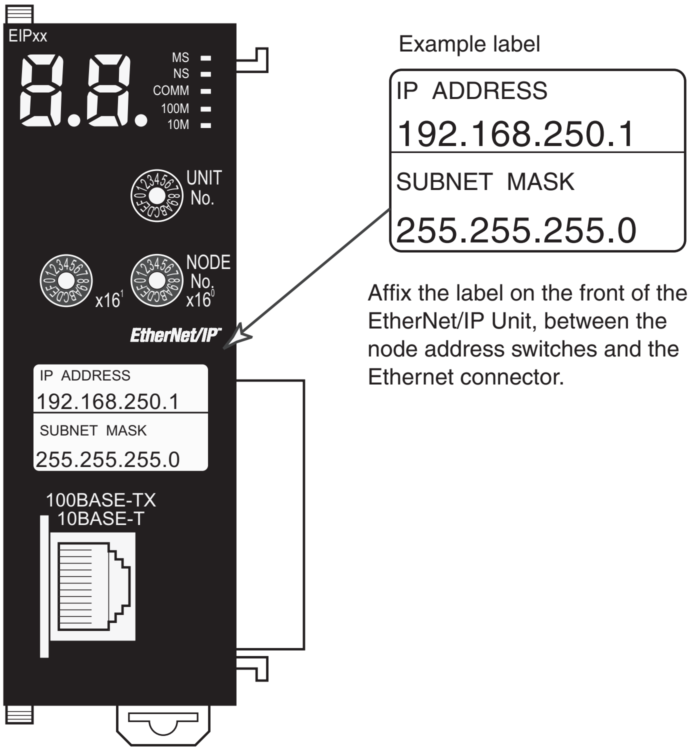
### Ethernet Address Notation

A specific Ethernet address is allocated to all devices connected to the Ethernet network. The EtherNet/IP Unit's address is listed in 12-digit hexadecimal on the right side of the Unit.



### Additional Information

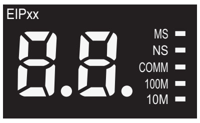
An IP address label is included with the EtherNet/IP Unit, so the user can record the user-set IP address and subnet mask on the label, and affix the label to the front of the Unit. When this label is affixed to the front of the Unit, it is easy to confirm the Unit's IP address and subnet mask.



Affix the label on the front of the EtherNet/IP Unit, between the node address switches and the Ethernet connector.

## Indicators (LEDs)

An EtherNet/IP Unit is equipped with the following indicators that indicate the operating status of the node itself and the overall network.



### Status Indicators: MS, NS, COMM, 100M, and 10M

The MS (Module Status) indicator indicates the status of the node itself and the NS (Network Status) indicator indicates the status of the network.

The COMM, 100M, and 10M indicators indicate the status of Ethernet communications.

The MS and NS indicators can be green or red. The COMM, 100M, and 10M indicators are yellow. These indicators can be lit, flashing, or not lit. The following table shows the meaning of these indicator conditions.

Refer to *Section 15 Troubleshooting* for details on using these indicators for troubleshooting.

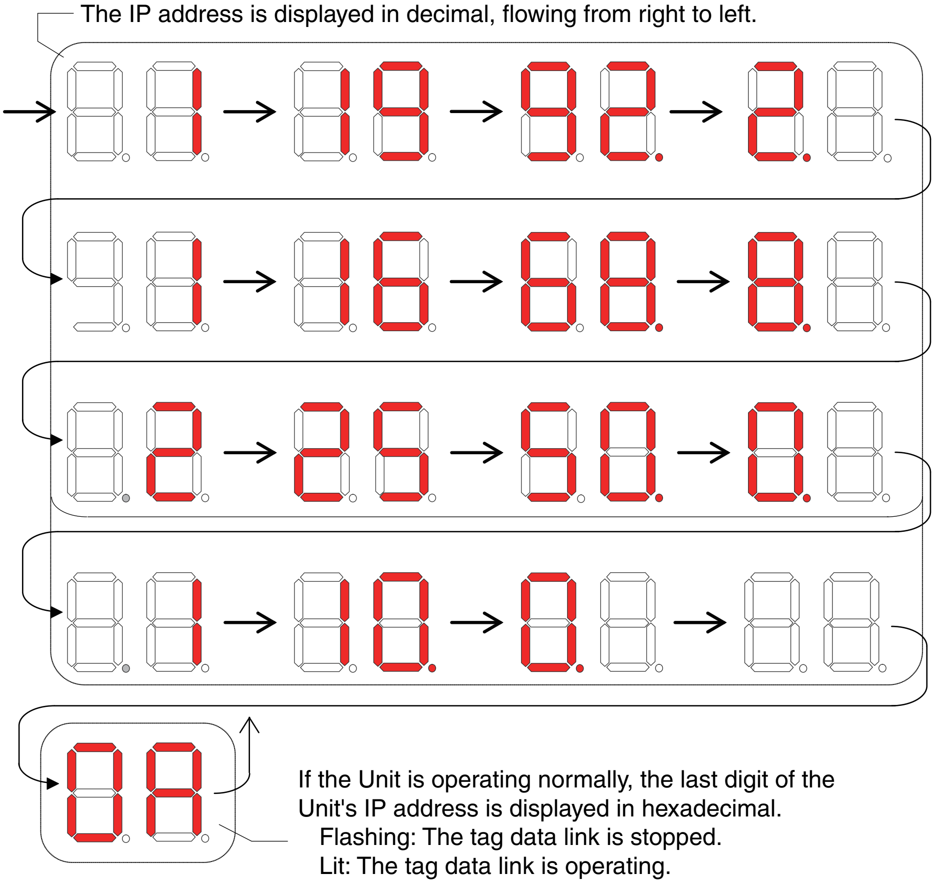
| Indicator | Name | Color | LED status | Indicated operating status |
|---|---|---|---|---|
| MS | Module Status | Red | Lit | Fatal error |
| | | | Flashing | Recoverable error |
| | | Green | Lit | Normal |
| | | | Flashing[*1] | IP address not set |
| | | --- | Not lit | Power supply OFF |
| NS | Network Status | Red | Lit | Fatal error |
| | | | Flashing | Recoverable error |
| | | Green | Lit | Tag data link and message connections established |
| | | | Flashing | Tag data link and message connections not established |
| | | --- | Not lit | Offline or power supply OFF |
| COMM | Communication | Yellow | Lit | Transferring data |
| | | | Not lit | Not transferring data |
| 100M | 100Mbps | Yellow | Lit | 100BASE-TX link established |
| | | | Not lit | 100BASE-TX link not established |
| 10M | 10Mbps | Yellow | Lit | 10BASE-TX link established |
| | | | Not lit | 10BASE-TX link not established |

[*1] This applies to the CJ1W-EIP21S only.

### ● Seven-segment Display

When the power is turned ON (or the Unit is restarted), all of the segments will flash twice, the IP address set in the EtherNet/IP Unit will be displayed on the 7-segment display just once, from right to left. Afterwards, the rightmost 8 bits of the IP address is displayed in hexadecimal during normal operation.

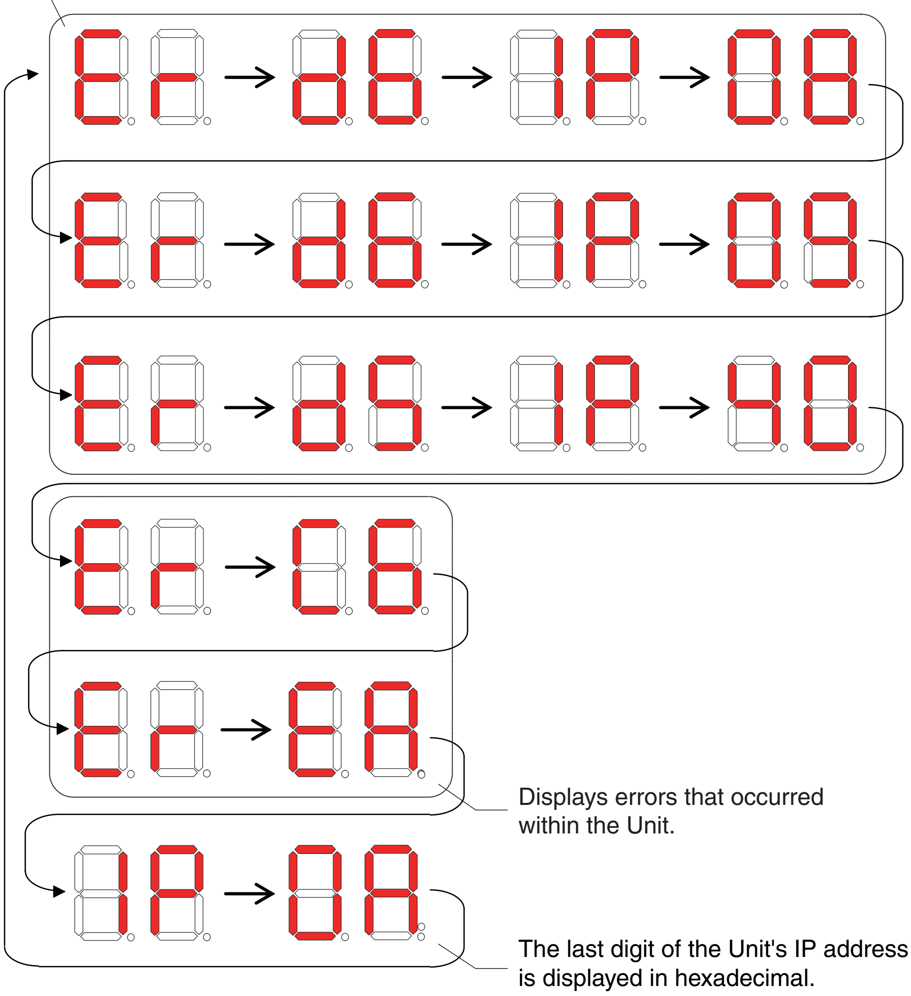
Example 1: Displaying IP Address 192.168.250.10


The IP address is displayed in decimal, flowing from right to left.
If the Unit is operating normally, the last digit of the Unit's IP address is displayed in hexadecimal.
Flashing: The tag data link is stopped.
Lit: The tag data link is operating.

If an error occurs, the error code will be displayed alternately with the rightmost byte of the affected device's IP address. For details on error codes, refer to *Section 15 Troubleshooting*.

Example 2: Displaying Multiple Error Sources

- A d6 error (failed to establish connection) occurred with IP address 192.168.250.8.
- A d6 error (failed to establish connection) occurred with IP address 192.168.250.9.
- A d5 error (verification error, target nonexistent) occurred with IP address 192.168.250.64.
- A C6 error (multiple switches ON) and EA error (EtherNet/IP Advanced Setting Error) occurred at the local EtherNet/IP Unit, IP address 192.168.250.10.



The error code is displayed and then the last digit of the target node's IP address is displayed in hexadecimal.

Displays errors that occurred within the Unit.

The last digit of the Unit's IP address is displayed in hexadecimal.

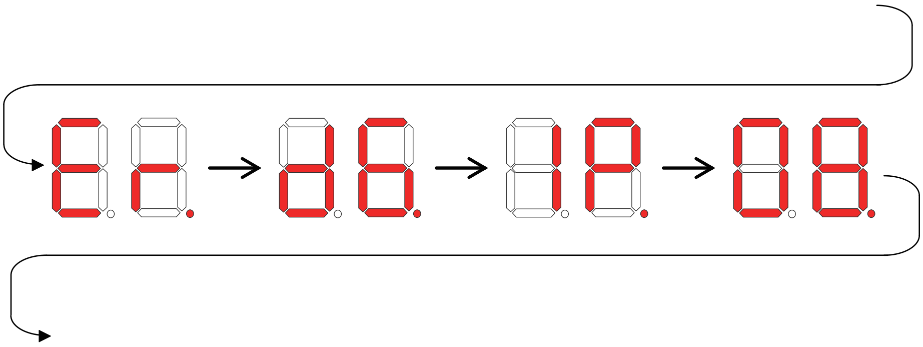
There is no particular priority to the order in which the errors are displayed. All of the errors are displayed repeatedly in order.

### Right and Left Dot LEDs

If an error occurred in two or more devices with the same rightmost byte in their IP addresses, the Right Dot LED will be lit while the device error is being displayed.

Example: Displaying the Following Errors

- A d6 error (failed to establish connection) occurred with IP address 10.0.1.8.
- A d6 error (failed to establish connection) occurred with IP address 10.0.2.8.



## Switch Settings

### Unit Number Setting Switch

The Unit Number Setting Switch sets the unit number of the EtherNet/IP Unit as a CPU Bus Unit.

| Setting method | Setting range |
|---|---|
| One-digit hexadecimal | 0 to F |

**Note** The unit number is factory-set to 0.

The unit number can be set to any number in the setting range (0 to F), as long as the same number is not set on another CPU Bus Unit in the same Controller.

**Precautions for Correct Use**

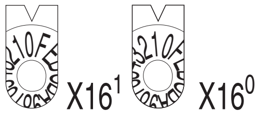
- Use a small screwdriver to make the setting, and be sure not to damage the rotary switch.
- Always turn OFF the Controller's power supply before setting the unit number.

**Additional Information**

- The unit number is factory-set to 0.
- If the same unit number is set on more than one CPU Bus Unit mounted in a Controller, a unit number duplication error will occur in the Controller and the EtherNet/IP Unit will not be able to start operating.

### ● Node Address Setting Switch

The Node Address Setting Switch sets the node address of the EtherNet/IP Unit.



| Setting method | Setting range |
|---|---|
| Two-digit hexadecimal | 01 to FE |

**Note** The node address is factory-set to 01. With the default settings, the values set on these switches become the last two digits of the IP address of the EtherNet/IP Unit.
Default IP address = 192.168.250.node address
With the factory-default node address setting of 01, the default IP address is 192.168.250.1.

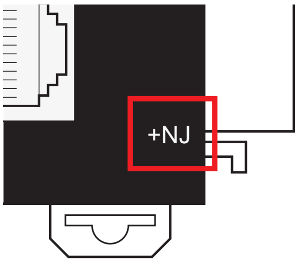
The node address can be set to any number in the setting range (01 to FE), as long as the same address is not set on another node in the network.

**Additional Information**

- If the node address setting is changed during operation, the MS Indicator will flash red.

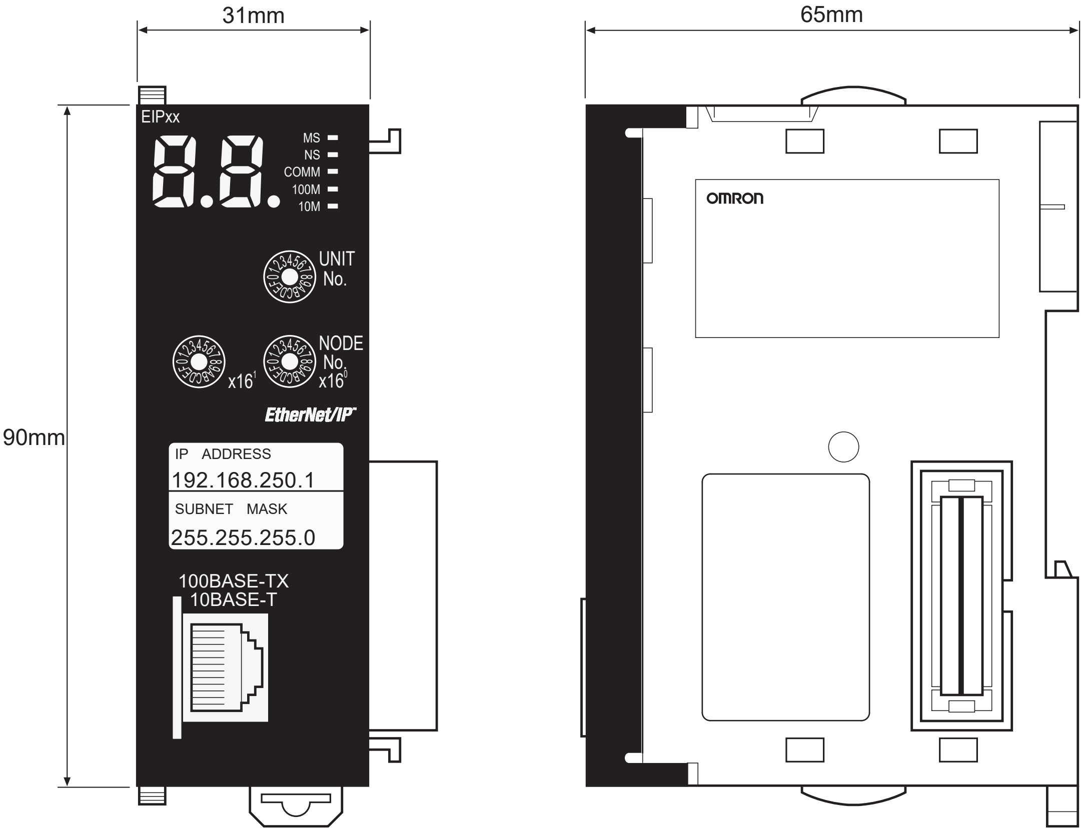
## Indication of Mounting Compatibility with NJ-series CPU Units (Only for CJ1W-EIP21S with Lot Number 241001□ or Later)

The CJ1W-EIP21S has "+NJ" printed at the lower right of the Unit's front panel, if it can be mounted on NJ-series CPU Units. This applies to the CJ1W-EIP21S with a lot number 241001□ or later.



## 1-3-3 Dimensions

- CJ1W-EIP21/EIP21S



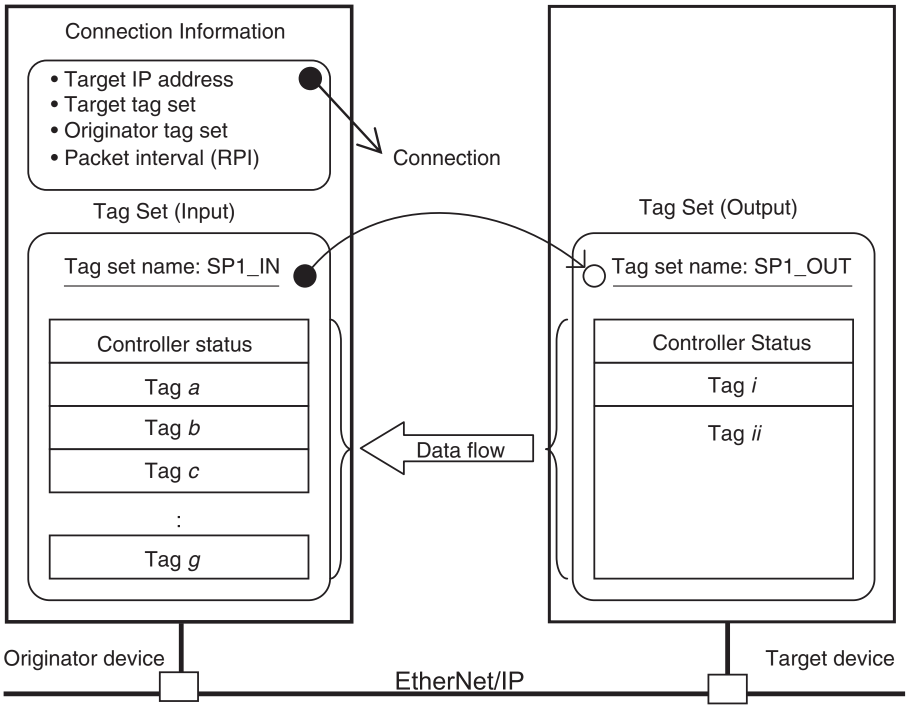
# 1-4 Introduction to Communications Services

## 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 an EtherNet/IP Unit. One connection is used per tag set. You can register up to 256 connections. The following table gives the EtherNet/IP Unit tag and tag set specifications.

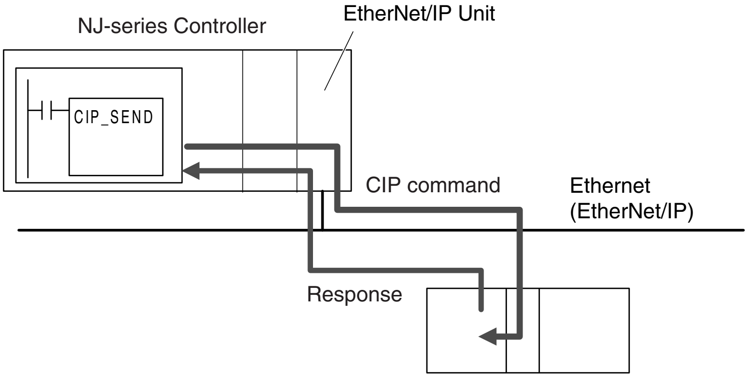
| Tags | Tag sets |
|---|---|
| Total size of all tags ≤ 184,832 words | Maximum size of 1 tag set ≤ 722 words<br>(The maximum size is 721 words when the tag set includes the Controller status.) |
| Maximum size of 1 tag ≤ 722 words<br>(The maximum size is 721 words when the tag set includes the Controller status.) | Number of tags per tag set ≤ 8<br>(7 tags/tag set when the tag set includes the Controller status)<br>**Note** Input and output variables cannot be combined. |
| Number of registrable tags ≤ 256 | Number of registrable tag sets ≤ 256 |



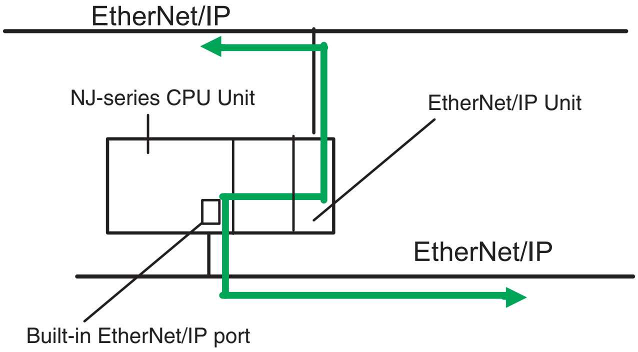
* In this example, a connection is established with the originator's tag list with tags a to g (inputs), which are in a tag set called SP1_IN, and the target's tag list with tags i and ii (outputs), which are in a tag set called SP1_OUT.

## CIP Message Communications

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



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



### Additional Information

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

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

OMRON products that support CIP subtract 5 seconds per hop.

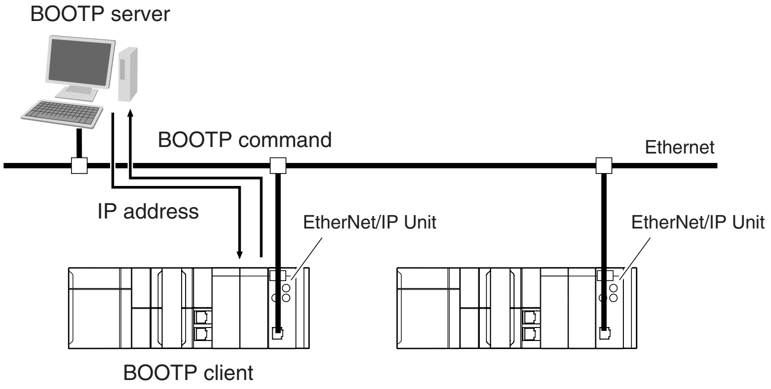
### Version Information

You can use the CJ1W-EIP21 EtherNet/IP Unit with a CPU Unit with unit version 1.01 or later and Sysmac Studio version 1.02 or higher.

You can use the CJ1W-EIP21S EtherNet/IP Unit with a CPU Unit with unit version 1.67 or later and Sysmac Studio version 1.60 or higher.

## 1-4-2 BOOTP Client

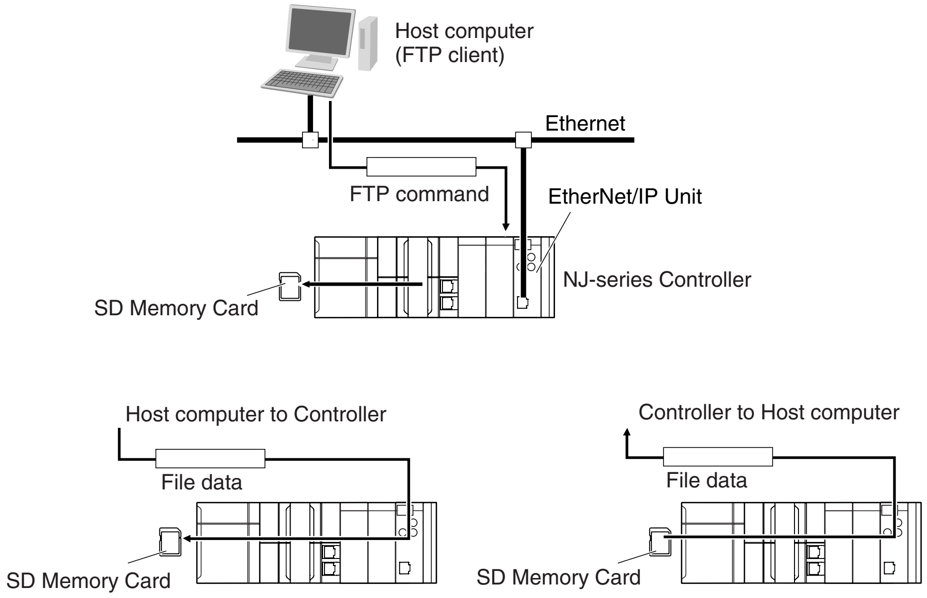
You set the EtherNet/IP Unit in the BOOTP settings to use the BOOTP client to obtain settings, such as the EtherNet/IP Unit IP address.



The EtherNet/IP Unit IP address is obtained from the BOOTP server when the power is turned ON.

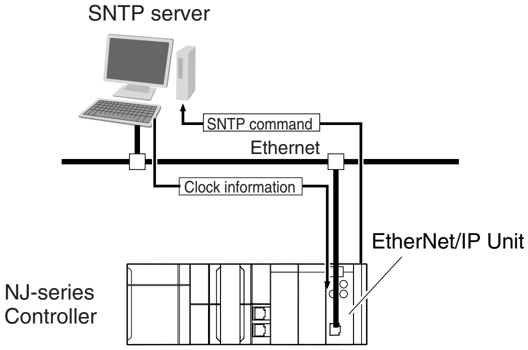
## 1-4-3 FTP Server

An FTP server is built into the EtherNet/IP Unit 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 EtherNet/IP Unit, clock information is read from the SNTP server at the specified times or when the Clock Information Adjustment Switch is changed to TRUE. The internal clock time in the CPU Unit is updated with the read time.



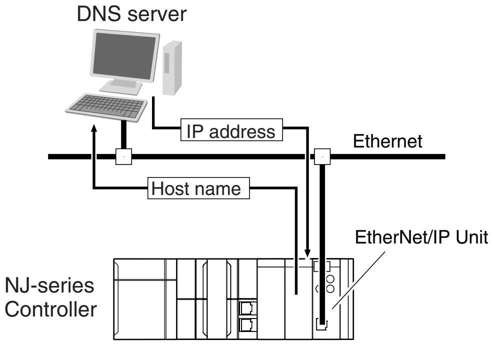
### Precautions for Correct Use

An SNTP server is required to use automatic clock adjustment.

## 1-4-5 Specifying Host Names

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

Example: Setting Host Names on the DNS Server

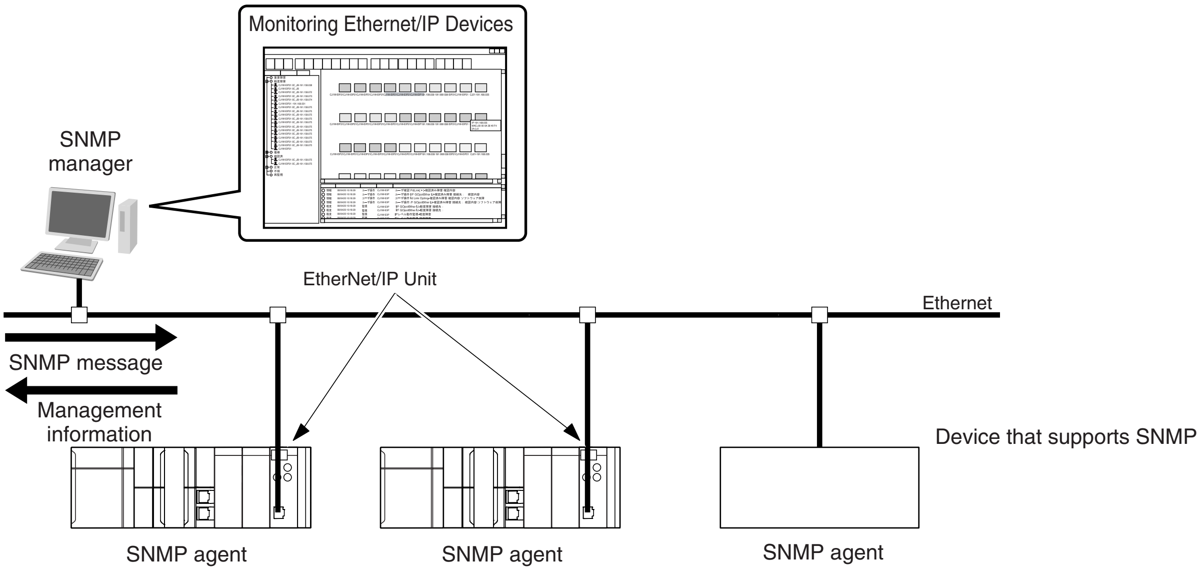


### Precautions for Correct Use

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

## 1-4-6 SNMP Agent

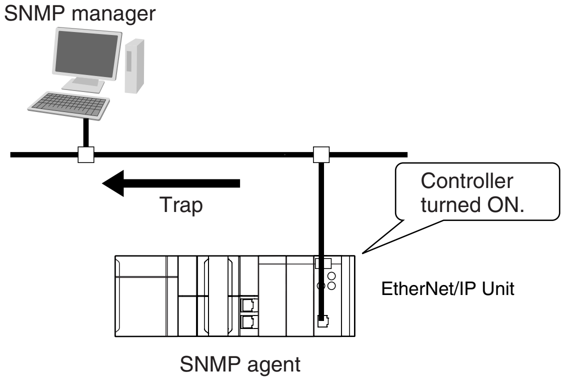
The SNMP agent passes internal status information from the EtherNet/IP Unit to network management software that uses an SNMP manager.



### SNMP Trap

When specific conditions occur, the EtherNet/IP Unit 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 EtherNet/IP Unit. 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



## 1-4-7 Socket Services (CJ1W-EIP21S Only)

You can send data to and receive data from any node on Ethernet with the UDP or TCP protocol.

In socket services, data is sent and received in communications processing, which is executed by using the Send Command instruction in the program or manipulating device variables assigned to I/O ports on the CJ1W-EIP21S.

After a connection with a remote device is established, the values of the specified variables are sent and the received data is stored in the specified variables. Then, the connection is closed and communications end.

For TCP, you can also read the socket status and received data.

You can use a total of eight ports, including both TCP and UDP ports.

# 1-5 EtherNet/IP Communications Procedures

## 1-5-1 Basic Operation

This section provides the basic procedures for the EtherNet/IP Unit.
Use the Sysmac Studio to create the programs and set the Unit.
Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for specific procedures on the Sysmac Studio.

**1** Set the unit number and node address on the rotary switches on the front of the Unit.

*2-1 Switch Settings on the Front Panel of the EtherNet/IP Unit*

↓

**2** Mount the Unit to the CPU Rack or Expansion Rack.

*2-2 Mounting the Unit to the CPU Rack or Expansion Rack*

↓

**3** Wire the Ethernet network with twisted-pair cable.

*2-5 Connecting to the Network*

↓

**4** Register the EtherNet/IP Unit in the Sysmac Studio.

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

1. Create a project.
2. Register the EtherNet/IP Unit in the Unit Configuration.

   Register the EtherNet/IP Unit in the Unit Configuration in **CPU/Expansion Racks** under **Configurations and Setup**. When you register the Unit, set the device name and unit number as well.

↓

**5** Assign the device variables for the CJ-series Unit to the I/O ports on the Sysmac Studio.

- Select existing variables and assign them.
- Enter new variable names.
- Automatically create the device variable names.

↓

**6** Create the initial settings for the EtherNet/IP Unit on the Sysmac Studio.

*Section 5 Sysmac Studio Settings for the EtherNet/IP Unit*

- Set up the EtherNet/IP Unit in the Edit Special Unit Settings Tab Page.
- Set the TCP/IP settings and Ethernet settings as required.

↓

**7** Set the IP address of the EtherNet/IP Unit with the Sysmac Studio.

*Section 4 Determining IP Addresses*

Set the IP address in one of the following ways:

- Default: 192.168.250.1 (subnet mask = 255.255.255.0)
- Set a user-specified IP address.
- Obtain the IP address from the BOOTP server.

↓

**8** Turn ON the power supply to the Controller.

Turn ON the power supply to the Controller. If there is a Unit Configuration registered in the CPU Unit and it does not agree with the actual configuration of the Units, an I/O Setting Check Error will occur. If that occurs, reset the Controller according to step 10, below, to clear the error.

↓

**9** Transfer the user program.

Transfer the user program, Unit Configuration and Setup, and variable information from the Sysmac Studio to the CPU Unit. At this point, the Special Unit Setup is not transferred to the EtherNet/IP Unit.

↓

**10** Update the Special Unit Setup.

Use one of the following methods to transfer the Special Unit Setup that you made in the Edit Special Unit Settings Tab Page from the CPU Unit to the EtherNet/IP Unit.

- Reset the Controller.
- Restart the EtherNet/IP Unit.
  - Click the **Restart** Button in the Edit Special Unit Setup Dialog Box.
  - Change the Unit Restart Flag in the device variables for the CJ-series Unit to TRUE.

### Restart Flag

| System-defined variable | Data type | R/W | Meaning | Description |
|---|---|---|---|---|
| _CJB_CBU**<br>Restart | BOOL | RW | CPU Bus Unit Restart Flag | TRUE: The CPU Bus Unit is restarted.<br>FALSE: The Unit is not restarted.<br>Default: FALSE |

("**" is replaced by the unit number.)

↓

**11** Execute a PING command from the computer to test communications.

*Section 6 Testing Communications*

## 1-5-2 Procedure for Using Tag Data Links

This section gives the setup procedure up to starting tag data links. Use the Network Configurator to set the tag data link parameters to use for tag data links.

**1** Import the settings of the variables that you specified for the tags on the Sysmac Studio to the Network Configurator.

*7-2-4 Creating Tags and Tag Sets*

↓

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

*Section 7 Tag Data Link Functions*

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

↓

**3** Connect the Network Configurator online.

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

↓

**4** Download the tag data link settings.

*7-2-9 Downloading Tag Data Link Parameters*

↓

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

*7-2-12 Starting and Stopping Tag Data Links*

↓

**6** Check operation.

*1-3-2 Part Names and Functions*

*Section 15 Troubleshooting*

- Check the indicators on the EtherNet/IP Unit.
- 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.

**Additional Information**

Status information on the target nodes for tag data links is provided in device variables for the CJ-series Unit. You can also store it in user-defined variables.

*User-defined Variables to Which the Status Areas Are Assigned* on page 3-5.

## 1-5-3 Using EtherNet/IP Network Functions

### ● Using the Message Communications Service

- CIP Communications Instructions

**1** Execute CIP communications instructions in the user program.

*Section 8 Message Communications*

↓

**2** Check operation.

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

*1-3-2 Part Names and Functions*

*Section 15 Troubleshooting*

### ● Using the FTP Server

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

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

*Section 9 FTP Server*

↓

**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 EtherNet/IP Unit.
- Check the event log to see if FTP started.

### ● Using the Automatic Clock Adjustment

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

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

*Section 10 Automatic Clock Adjustment*

↓

**2** Execute automatic clock adjustment.

- Execute automatic adjustment at specified times.

### ● Using the SNMP Agent

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

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

*Section 11 SNMP Agent*

↓

**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 EtherNet/IP Unit*

- Set the BOOTP settings.

↓

**2** Check operation.

- Check the event log to see if BOOTP started.
- Check online status with the device variables for the CJ-series Unit.

### ● Using the Socket Services

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

*Section 5 Sysmac Studio Settings for the EtherNet/IP Unit*

- Configure the Status Area Settings.

↓

**2** Execute socket services in the user program.

*Section 13 Socket Services*

- Set the socket service parameters.
- Write the program to execute the Send Command instruction or manipulate device variables.

↓

**3** Check operation.

*Section 13 Socket Services*

*3-2-5 Device Variables for the CJ-series Unit Socket Service-related Data (CJ1W-EIP21S Only)*

- Check the socket status.
- Check the response codes in the Socket Service Parameter Area.

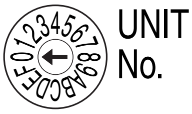
# 2

# Installing Ethernet Networks

# 2-1 Switch Settings on the Front Panel of the EtherNet/IP Unit

## ● Setting the Unit Number

The unit number is used to identify individual CPU Bus Units when more than one CPU Bus Unit is mounted to the same Controller. Use a small screwdriver to make the setting, taking care not to damage the rotary switch. The unit number is factory-set to 0.



| Setting range: |
|---|
| 0 to F |

**Precautions for Correct Use**

- Turn OFF the power supply before setting the unit number.
- If you set the unit number of the EtherNet/IP Unit in the Unit Configuration on the Sysmac Studio, make sure that the rotary switches are set to the same unit number. If different unit numbers are set in the Unit Configuration and on the rotary switches, an I/O Setting Check Error will occur.

## ● Setting the Node Address

The node addresses are used by the FINS communications service to identify individual EtherNet/IP Units (nodes).

Use the node address switches to set the node address as a hexadecimal value. Do not set the same node address as any other node on the same Ethernet network.

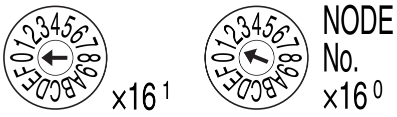
If the FINS communications service is not used on the Ethernet network, then it is all right for the same node address to be set on two or more EtherNet/IP Units.

The node address must meet the following conditions.

If the conditions are not met, an error will occur (the MS indicator will flash red and the 7-segment display will show H4 (Node Address Setting Error)) and the EtherNet/IP Unit will stop operating.

- Set the node address to between 01 and FE hex.
- Set the node address on the rotary switches and the lowest field of the IP address (i.e., the host ID) of the EtherNet/IP Unit to the same value.

If you cannot use the same value for the node address and the lowest field of the IP address of the EtherNet/IP Unit, you must use the IP address table method or the combined method for address conversion. Refer to *Section 5 Sysmac Studio Settings for the EtherNet/IP Unit* for details.



| Setting range: |
|---|
| 01 to FE (1 to 254 decimal) |

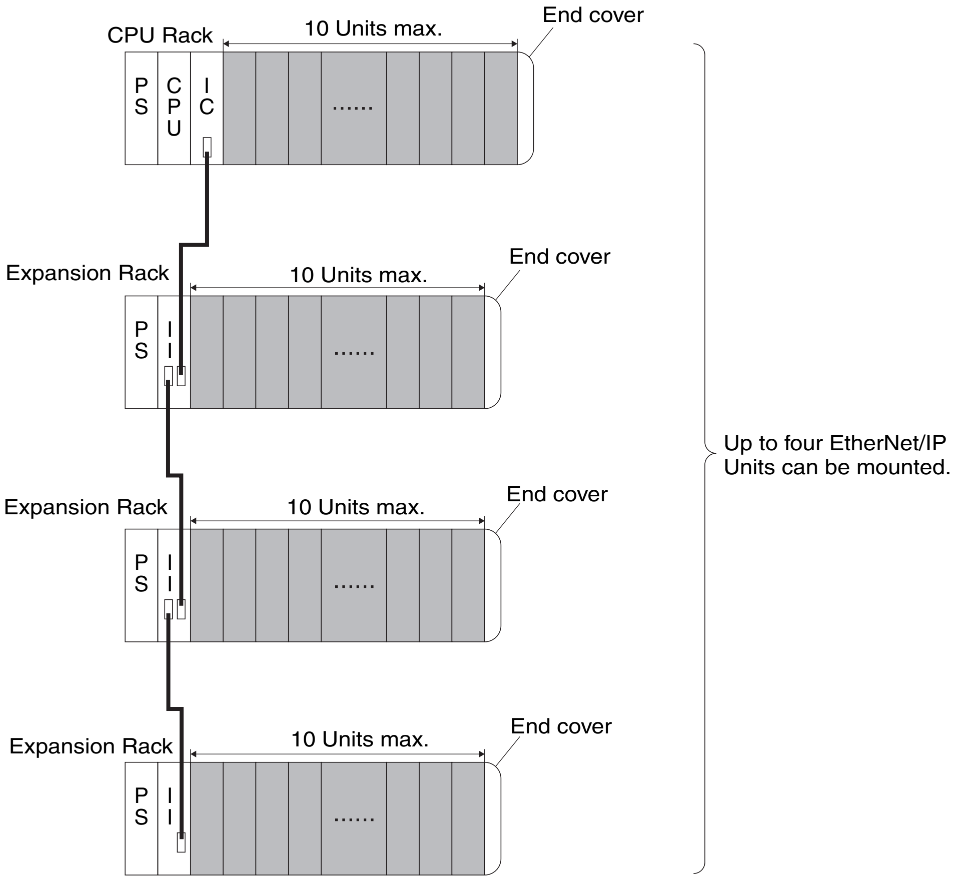
The left switch sets the sixteens digit (most significant digit) and the right switch sets the ones digit (least significant digit). The node address is factory-set to 01.

**Precautions for Correct Use**

Turn OFF the power supply before setting the node address.

# 2-2 Mounting the Unit to the CPU Rack or Expansion Rack

You can mount EtherNet/IP Units to an NJ-series CPU Rack or a CJ-series Expansion Rack. Mount an EtherNet/IP Unit in any of the positions shown below using the sliders on the top and bottom of the Unit. You can mount a maximum of four EtherNet/IP Units in a single Controller. If EtherNet/IP Units are mounted in combination with other CPU Bus Units, the maximum total number of CPU Bus Units that can be mounted is 16.



PS: Power Supply Unit
CPU: CPU Unit
IC: I/O Control Unit
II: I/O Interface Unit

### Additional Information

The maximum current consumption of the EtherNet/IP Units is as follows.

- Current consumption of the CJ1W-EIP21: 410 mA at maximum
- Current consumption of the CJ1W-EIP21S: 620 mA at maximum

Be sure that the total current consumption of all the Units that are mounted to the same CPU Rack or Expansion Rack does not exceed the output capacity of the Power Supply Unit.

# 2-3 Selecting the Network Devices

## 2-3-1 Recommended Network Devices

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

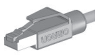
### ● Ethernet Switches

| Manufacturer | Model | Description |
|---|---|---|
| OMRON | W4S1-03B | • Packet priority control (QoS): EtherNet/IP control data priority<br>• Failure detection: Broadcast storm, LSI error detection<br>• 10/100Base-TX<br>• Auto negotiation<br>• Number of ports (three for the W4S1-03B, or five each for the W4S1-05B, W4S1-05C, and W4S1-05D)<br>• Failure detection output (W4S1-05C only) |
| | W4S1-05B | |
| | W4S1-05C | |
| | W4S1-05D | |
| Cisco Systems, Inc. | Consult the manufacturer.<br>http://www.cisco.com/ | |
| CONTEC USA Inc. | Consult the manufacturer.<br>http://www.contec.com/ | |
| Phoenix Contact USA | Consult the manufacturer.<br>https://www.phoenixcontact.com | |

### ● Twisted-pair Cables and Connectors

Applicable EtherNet/IP communications cables and connectors vary depending on the used baud rate. Use an STP (shielded twisted-pair) cable of category 5 or higher. You can use either a straight or cross cable.

Cabling materials used for EtherNet/IP communication cables are shown in the table below. "100Base-TX" in the "Product" column of the table below indicates that either 100Base-TX or 10Base-T can be used.

| Product | | Manufacturer | Model |
|---|---|---|---|
| Sizes and conductor pairs: AWG 24 × 4 pairs | Cables | Tonichi Kyosan Cable, Ltd. | NETSTAR-C5E<br>SAB 0.5 × 4P CP |
| | | Kuramo Electric Co., Ltd. | KETH-SB |
| | RJ45 Connectors | Panduit Corporation | MPS588 |
| Sizes and conductor pairs: AWG 22 × 2 pairs *1 | Cables | Kuramo Electric Co., Ltd. | KETH-PSB-OMR |
| | | Nihon Electric Wire & Cable Co., Ltd. | PNET/B |
| | RJ45 Assembly Connectors  | OMRON | XS6G-T421-1 |
| Sizes and conductor pairs: 0.5 mm × 4 pairs | Cables | Fujikura Ltd. | F-LINK-E 0.5mm × 4P |
| | RJ45 Connectors | Panduit Corporation | MPS588 |

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

- **Boots**

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

**Precautions for Correct Use**

- Always use an 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.

## 2-3-2 Ethernet Switch Types

### Unmanaged Layer 2 (L2) Ethernet Switches

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

### Managed Layer 2 (L2) Ethernet Switches

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

## 2-3-3 Ethernet Switch Functions

This section describes the Ethernet switch functions that are important for an EtherNet/IP network. For an EtherNet/IP Unit, 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 EtherNet/IP Units 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.

## 2-3-4 Precautions for Ethernet Switch Selection

The functions supported by the Ethernet switch may affect tag data link transmission delays and the settings in the Controller configurations and setup. In addition, if the Ethernet switch supports advanced functions, special settings are required for those functions. When you select an Ethernet switch, it is necessary to consider whether to select the Ethernet switch based on the kind and amount of communications you want to execute in the network. Refer to the following precautions when you select an Ethernet switch. Refer to *14-2 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.

# 2-4 Network Installation

## 2-4-1 Basic Installation Precautions

- Take the greatest care when you install the Ethernet System. Be sure to follow ISO 8802-3 specifications. Be sure you understand them before attempting to install an Ethernet System.
- Unless you are already experienced in installation of communications systems, we strongly recommend that you employ a professional to install your system.
- Do not install Ethernet equipment near sources of noise. If a noisy environment is unavoidable, take adequate measures against noise interference, such as installation of network components in metal cases or the use of optical cable in the system.
- When using a shielded cable with the shields on both ends of the cable connected to connector hoods, ground loops induced by improper grounding methods may decrease noise immunity and cause device damage. To prevent ground loops caused by differences in potential between device grounding points, the reference potential between the devices must be stabilized. Design grounding appropriately so that noise current does not flow to ground lines between the devices. For grounding methods, refer to the *NJ-series CPU Unit Hardware User's Manual* (Cat. No. W500) and the *NX-series CPU Unit Hardware User's Manual* (Cat. No. W535).
- To obtain information on laying EtherNet/IP cable, contact ODVA.
  ODVA web site: http://www.odva.org
- When you install an EtherNet/IP network that combines an information network with the control system, and the communications load may be heavy due to tag data links, we recommend that you set up the network so that the load does not affect communications. For example, install the tag data links in a segment that is separate from the information network.

## 2-4-2 Recommended Network Devices

Refer to *2-3 Selecting the Network Devices* for the devices recommended for use with the EtherNet/IP Unit.

## 2-4-3 Precautions When Laying Twisted-pair Cable

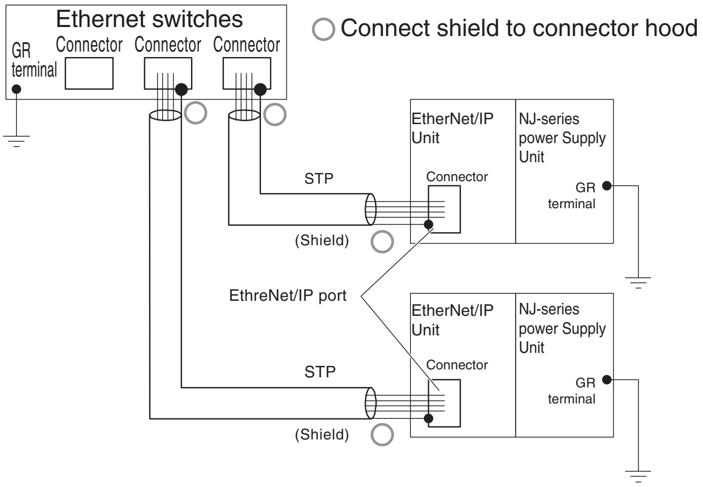
### Connecting the Shield to Connector Hoods

- **Between an EtherNet/IP Port and an Ethernet Switch**

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

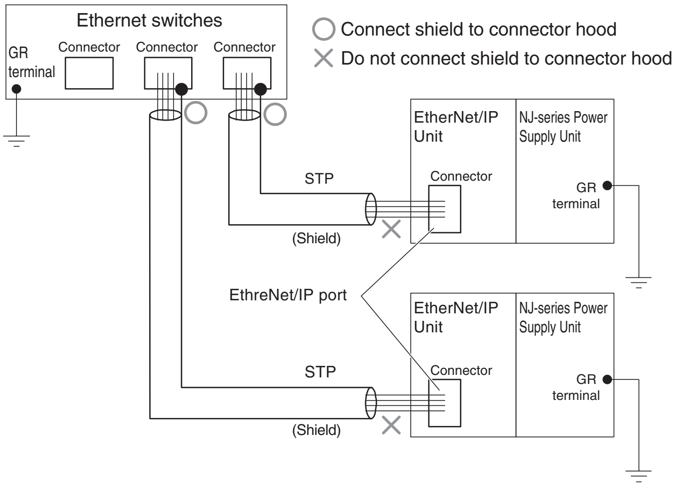
a) Connecting the shields at both ends of the cable

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



b) Connecting the shields on the Ethernet switch side only

Connect only the shield at the end of the cable on the Ethernet switch side to the connector hood.



**Additional Information**

Noise immunity may be reduced and device damage may occur due to ground loops, which can occur due to improper shield connections and grounding methods. It may be possible to alleviate this problem by connecting only the Ethernet switch side as described in b), rather than connecting both ends as described in a).

- **Between two Ethernet switches**

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

### Other Precautions When Laying the Twisted-pair Cable

- Press the cable connector in firmly until it locks into place at both the Ethernet switch and the EtherNet/IP Unit.
- 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.

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

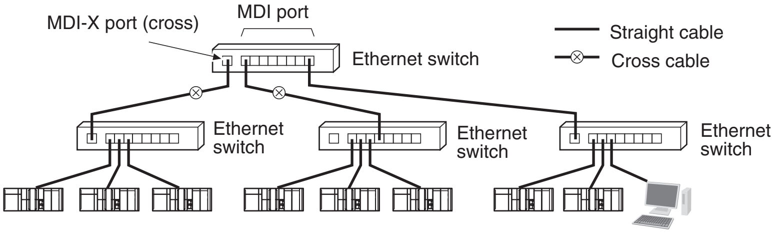
### Precautions When Installing Ethernet Switches

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

### Ethernet Switch Connection Methods

- 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 EtherNet/IP Unit'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 | | EtherNet/IP Unit | | | | |
|---|---|---|---|---|---|---|
| | | 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 |

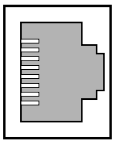
Best = Recommended; OK = Allowed; --- = Not allowed.

# 2-5 Connecting to the Network

## 2-5-1 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)

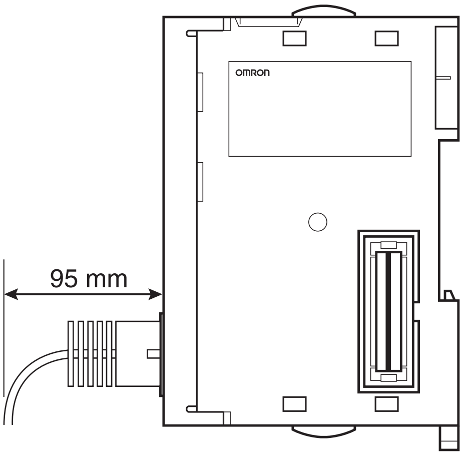


| Connector pin | Signal name | Abbr. | Signal direction |
|---|---|---|---|
| 1 | Transmission data + | TD+ | Output |
| 2 | Transmission data – | TD– | Output |
| 3 | Reception data + | RD+ | Input |
| 4 | Not used. | --- | --- |
| 5 | Not used. | --- | --- |
| 6 | Reception data – | RD– | Input |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

## 2-5-2 Connecting the Cable

**Precautions for Correct Use**

- Turn OFF the power supply to the Controller before you connect or disconnect twisted-pair cable.
- Allow more than enough space for the bending radius of the twisted-pair cable. The space that is required depends on the communications cable and connector that you use. Contact the manufacturer or sales agent.



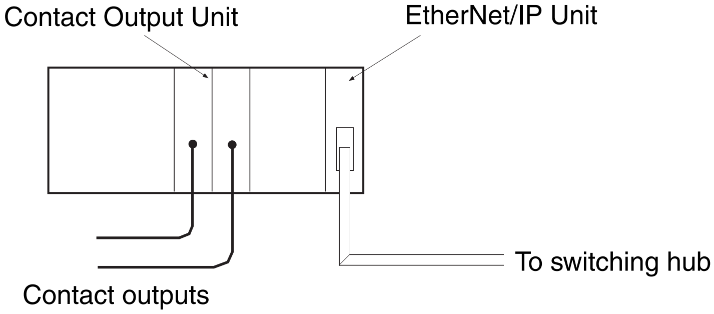
1. Lay the twisted-pair cable.
2. Connect the cable to the Ethernet switch.
3. Connect the twisted-pair cable to the connector on the EtherNet/IP Unit. Be sure to press the connectors (both the Ethernet switch side and Ethernet side) until they lock into place.

## 2-5-3 Noise Countermeasures for Contact Output Units

When an EtherNet/IP Unit and Contact Output Unit are mounted in the same Controller, communications errors may occur due to the noise that is generated by the contact outputs. Take either of the following countermeasures.

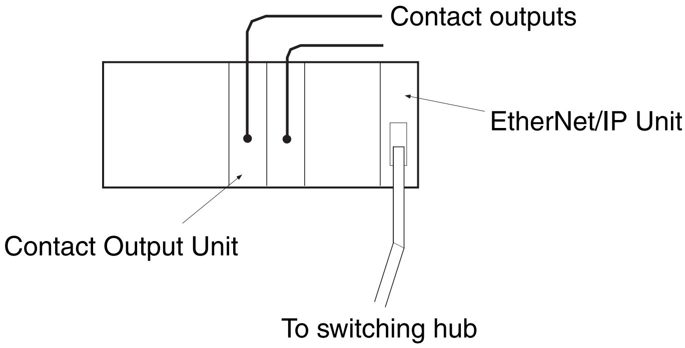
### ● Separating the Units

Separate the EtherNet/IP Unit as far as possible from the Contact Output Units.



### ● Separating the Cables

Separate the twisted-pair cable that is connected to the EtherNet/ IP Unit as far as possible from the wiring to the Contact Output Units.



# 3

# Assigning Device Variables for CJ-series Units

# 3-1 Data Exchange with the CPU Unit

Data exchange between the EtherNet/IP Unit and CPU Unit is performed with the I/O ports that are assigned to the EtherNet/IP Unit and, if required, the memory used for CJ-series Units.

Refer to *3-2 Device Variables for the CJ-series Unit* for information on I/O ports.

## 3-1-1 Data Flow

The exchange of data between the CPU Unit and the EtherNet/IP Unit is described in the following table and figure.

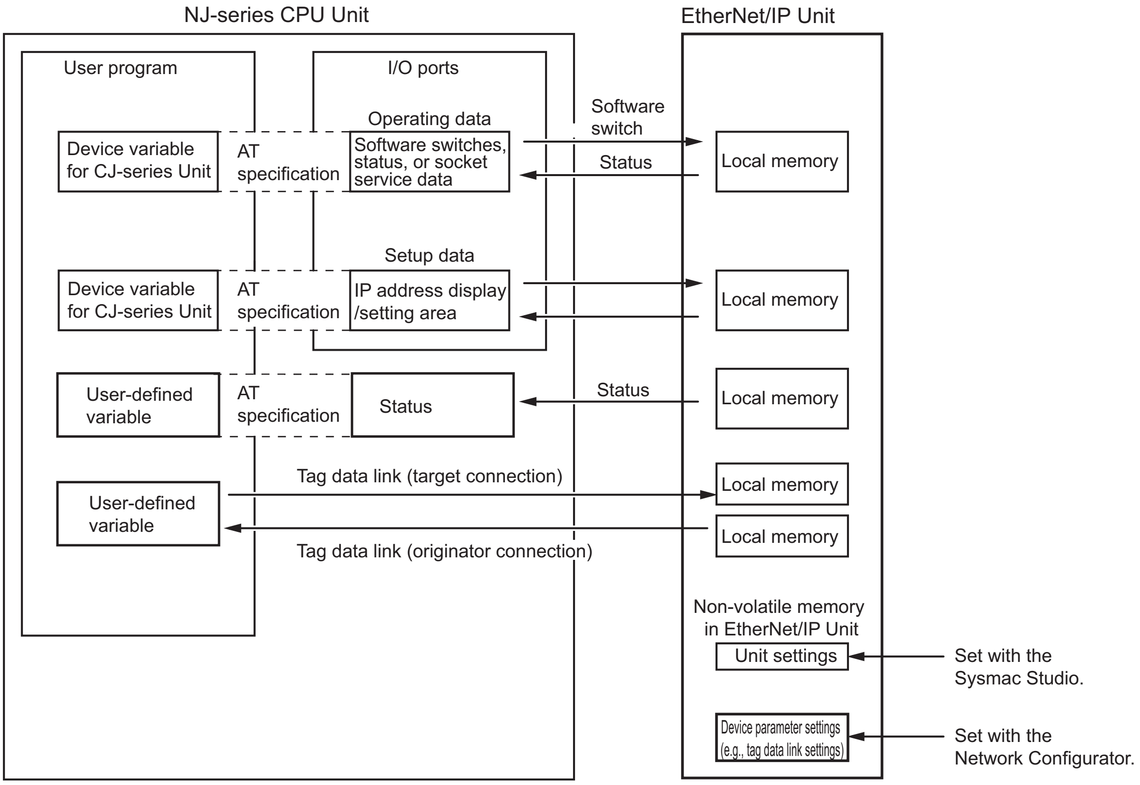
Types of Data Exchange

| Access method from the user program | AT specification | Timing of data exchange | Classification of Unit data |
|---|---|---|---|
| Device variables for CJ-series Unit | I/O ports for operating data | During I/O refreshing | Software switches |
| | | | Status *1<br>Socket service data*2 |
| | I/O ports for setup data | During I/O refreshing | IP address display/setting area |
| User-defined variables | Memory used for CJ-series Units | During I/O refreshing | Status *3 |

*1 The target PLC operating and error information, Unit status, communications status, registered/normal target nodes, and FINS/TCP connection status are sent as status.

*2 This applies to the CJ1W-EIP21S when the status area layout in the Status Area Settings Display is set to *User setting*. In this case, among the status information provided by default, only Unit Status and Communications Status will be sent.

*3 The target PLC operating and error information and the registered/normal target node table are sent as status.


NJ-series CPU Unit
EtherNet/IP Unit
User program
I/O ports
Operating data
Device variable for CJ-series Unit
AT specification
Software switches, status, or socket service data
Software switch
Status
Local memory
Setup data
Device variable for CJ-series Unit
AT specification
IP address display /setting area
Local memory
User-defined variable
AT specification
Status
Status
Local memory
Tag data link (target connection)
User-defined variable
Local memory
Local memory
Tag data link (originator connection)
Non-volatile memory in EtherNet/IP Unit
Unit settings
Set with the Sysmac Studio.
Device parameter settings (e.g., tag data link settings)
Set with the Network Configurator.

## Device Variables for CJ-series Unit

A device variable for a CJ-series Unit is a variable with an AT specification to an I/O port. In the user program, you can use device variables for a CJ-series Unit to access Configuration Units, such as the EtherNet/IP Unit.

Refer to *Specifying Device Variables for CJ-series Units* on page 3-5 for information on assigning I/O ports for device variables for CJ-series Units.

### ● I/O Ports

An I/O port is a logical interface that is used by the CPU Unit to exchange data with an EtherNet/IP Unit or other Configuration Unit.

The names of I/O ports are defined in advance for each Unit model number and function.

I/O ports are automatically created when you create the Unit Configuration on the Sysmac Studio.

Refer to *3-2 Device Variables for the CJ-series Unit* for information on the I/O ports that are defined for the EtherNet/IP Unit.

There are two types of I/O ports for an EtherNet/IP Unit: I/O ports for operating data and I/O ports for setup data.

### ● Operating Data

- Software Switches
  Software switches for output from the CPU Unit to the EtherNet/IP Unit are provided, such as the Tag Data Link Start Bit.
  Software switch data is stored in the I/O ports for operating data that are assigned to the EtherNet/IP Unit.
  From the user program, you can use a device variable for the CJ-series Unit to access the data or manipulate the switch. The software switches are updated during I/O refreshing.
- Status
  Status data for input from the EtherNet/IP Unit to the CPU Unit is provided, such as the Communications Status. Status data is stored in the I/O ports for operating data that is assigned to the EtherNet/IP Unit.
  From the user program, you can use a device variable for the CJ-series Unit to access the status data.
  The status data is updated during I/O refreshing.
- Socket Service Data (CJ1W-EIP21S Only)
  When the Status Area Layout Is Set to *User setting* in the Status Area Settings Display

  Socket service data includes Socket Service Request switches, UDP/TCP socket status, etc. for each socket.
  This data is referenced and used for switch operations from the user program via device variables for CJ-series Units.

### ● Setup Data

- IP Address Display/Setting Area
  You normally set the setup data from the Edit Special Unit Settings Tab Page of the Sysmac Studio. You can also set the setup data from the user program by using the setup device variables for the CJ-series Unit with AT specifications to the applicable I/O ports.

## User-defined Variables

User-defined variables are used to perform tag data links and other communications for an EtherNet/IP Unit that is connected to an NJ-series CPU Unit. For details on user-defined variables, refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501).

## Access Methods from the User Program

The CPU Unit and EtherNet/IP Unit exchange data through the memory used for CJ-series Units in the CPU Unit.

Refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) for details on the memory used for CJ-series Units.

To exchange data, use the data that is listed in the following table in the user program.

| Type of data | Access method |
|---|---|
| Software switches | Device variables for the CJ-series Unit |
| Status data | |
| Setup data | |
| Status data | User-defined variables |
| Data that is updated with tag data links | |

## 3-1-2 Specifying and Creating Variables

### Specifying Device Variables for CJ-series Units

Assign the device variables for the CJ-series Unit to the I/O ports in the I/O Map on the Sysmac Studio.

Specify the assigned variable names as given below.

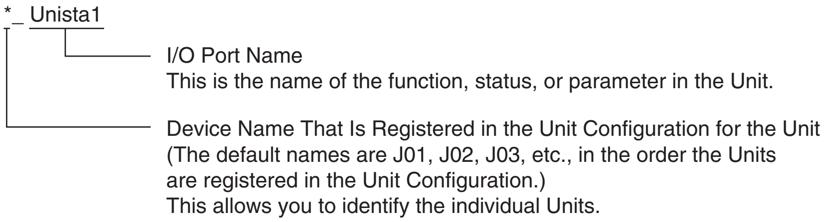
**(1) Select existing variables and assign them.**

**(2) Enter new variable names.**

**(3) Automatically create the device variable names.**

The configuration of the automatically created variable names (3, above) is shown below.

Device Variable Names for CJ-series Units



*_ Unista1

I/O Port Name
This is the name of the function, status, or parameter in the Unit.

Device Name That Is Registered in the Unit Configuration for the Unit
(The default names are J01, J02, J03, etc., in the order the Units
are registered in the Unit Configuration.)
This allows you to identify the individual Units.

Refer to *3-2 Device Variables for the CJ-series Unit* for details information on the device variables for the CJ-series Unit.

In further descriptions, the variable names that are automatically generated for the device variables for the CJ-series Unit are used, such as **_TDLStartCmd*.

### User-defined Variables to Which the Status Areas Are Assigned

If you assign the status areas to user-defined variables, there is a data type for a structure that is defined in advance for that purpose.

To use that data type, register the structure data type for the EtherNet/IP Unit in advance in the CPU/Expansion Racks Tab Page of the Controller Configurations and Setup.

Right-click the Unit in the CPU/Expansion Racks Tab Page of the Controller Configurations and Setup, and then select ***Register Data Types for Units*** from the menu to register the data type.

After you register the structure data type for the EtherNet/IP Unit, you create a user-defined variable that uses that structure data type to assign the status areas.

You can then use that user-defined variable in the user program to read information from the status area.

Refer to *3-3 User Definition Settings for the Status Area* for information on setting the status area.

- **Data Type for User-defined Status Area**

| Structure data type name | Member | Data type |
|---|---|---|
| _sCJEIP21_User_StaArea | RegTargetSta (Registered Target Node Table) | _uCJEIP21_TargetSta |
| | EstbTargetSta (Normal Target Node Table) | _uCJEIP21_TargetSta |
| | TargetPLCMdSta (Target Node PLC Operating Flags) | _uCJEIP21_TargetSta |
| | TargetPLCErrSta (Target Node PLC Error Flags) | _uCJEIP21_TargetSta |

# 3-2 Device Variables for the CJ-series Unit

The section describes the device variables for the CJ-series Unit individually.

In this section, the variable names that are created automatically in the I/O Map are used.

## 3-2-1 Assigning Device Variables for CJ-series Units

An NJ-series Controller predefines the following three types of data as I/O ports for the operating data and setup data of an EtherNet/IP Unit.

The device variables for the CJ-series Unit are assigned to I/O ports to use them.

**(1) Software switches (Serve as commands from the CPU Unit to the EtherNet/IP Unit.)**

**(2) Status (Gives the status of the EtherNet/IP Unit, error information, network status, and tag data link target status.)**

**(3) Socket service-related data (Socket Service Request Switches, TCP/UDP socket status, etc.)**

**(4) Setup (Stores the IP address that is set for the EtherNet/IP Unit.)**

The three types of device variables for the EtherNet/IP Unit are described individually below.

Refer to *A-7-2 Differences in Access Methods from the User Program* for the operating data and setup data addresses that correspond to the device variables for the CJ-series Unit.

## 3-2-2 Device Variables for the CJ-series Unit for Software Switches

These variables are used for outputs from the CPU Unit to the EtherNet/IP Unit.

| Device variable name for CJ-series Unit | Data type | R/W | Meaning | Description |
|---|---|---|---|---|
| *_TDLStartCmd | BOOL | RW | Tag Data Link Start Bit | The tag data links start when this variable is changed to TRUE.<br>FALSE: Tag data link operation has started.<br>Default: FALSE |
| *_TDLStopCmd | BOOL | RW | Tag Data Link Stop Bit | The tag data links stop when this variable is changed to TRUE.<br>FALSE: Tag data link operation has stopped.<br>Default: FALSE |
| *_AdjTmCmd | BOOL | RW | Adjust Clock Bit | The clock is automatically adjusted when this variable is changed to TRUE.<br>Default: FALSE |
| *_SktForceCloseCmd (CJ1W-EIP21S Only) | BOOL | RW | Force Socket Close Switch | Sockets are forcibly closed when this variable is changed to TRUE.<br>Default:FALSE |

### ● Tag Data Link Start Bit (**_*_TDLStartCmd_**)

Start the tag data links by changing this variable to TRUE. If the tag data links are already operating, the signal is ignored. The tag data links start operating automatically after the tag data link parameter settings are downloaded from the Network Configurator, the power to the Controller is turned ON, or the Unit is restarted.

If the Tag Data Link Stop Bit was changed to TRUE to stop the tag data links, you can restart the tag data links by changing the Tag Data Link Start Bit to TRUE.

After the tag data links start, the EtherNet/IP Unit automatically changes the Tag Data Link Start Bit to FALSE. Do not force this variable to FALSE. It is automatically changed to FALSE by the Unit.

### ● Tag Data Link Stop Bit (**_*_TDLStopCmd_**)

Stop the tag data links by changing this variable to TRUE. After the tag data links stop, they will remain stopped until the Unit is restarted or the Tag Data Link Start Bit is changed to TRUE. (The tag data links will also start automatically when the tag data link parameter settings are downloaded from the Network Configurator.)

If the tag data links are already stopped, the signal will be ignored.

Message communications can be performed while the tag data links are stopped.

After the tag data links stop, the EtherNet/IP Unit automatically changes the Tag Data Link Stop Bit to FALSE. Do not force this variable to FALSE. It is automatically changed to FALSE by the Unit.

### ● Adjust Clock Bit (**_*_AdjTmCmd_**)

Automatically adjust the time on the clock by changing this variable to TRUE. The SNTP server used to adjust the time is set in the Unit Setup.

After the clock time is adjusted, the EtherNet/IP Unit automatically changes the Adjust Clock Bit to FALSE. Do not force this variable to FALSE. It is automatically changed to FALSE by the Unit.

### ● Force Socket Close Switch (**_*_SktForceCloseCmd_**) (CJ1W-EIP21S Only)

All UDP and TCP sockets used for socket services can be force-closed by changing this variable to TRUE. This can be used for operations such as error processing.

Be careful not to force-close sockets during communications, or an error will occur.

After all sockets have been force-closed, the CJ1W-EIP21S EtherNet/IP Unit will automatically change this variable to FALSE again. Do not attempt to forcibly manipulate this switch before it is automatically returned to FALSE by the Unit.

Ports used internally by the CJ1W-EIP21S EtherNet/IP Unit will not be closed.

## 3-2-3 Device Variables for the CJ-series Unit for Status

These variables are used for inputs from the EtherNet/IP Unit to the CPU Unit.

Socket service data is also described in this section.

### Target PLC Information

The following variables are available only when the status area layout in the Status Area Settings Display is set to *Default*.
When the status area layout in the Status Area Settings Display is set to *User setting*, instead of this variable, the user-defined variable set in *Mapped area* in the Status Area Settings Display should be referenced. Refer to *3-3 User Definition Settings for the Status Area* for details.

These variables provide information on the target PLCs that are connected to the EtherNet/IP Unit when the EtherNet/IP Unit is the originator.

**Additional Information**

- This status information is enabled when the Controller status is included in the communications data in both the originator and target node.
- This device variable for the CJ-series Unit provides information for only nodes 0 to 63.
- If it is necessary to get the error status of nodes higher than node 63, refer to *3-3 User Definition Settings for the Status Area*.

#### ● Target Node PLC Operating Flags (EtherNet/IP Unit to CPU Unit) (**_TargetPLCMdSta [0]* to *[63]*)

This variable provides information on the target PLCs that are connected to the EtherNet/IP Unit when the EtherNet/IP Unit is the originator.

The information in this variable is valid only when the corresponding element in the Normal Target Node Table is TRUE.

If the corresponding element in the Normal Target Node Table is FALSE, the Target Node PLC Operating Flags the previous operating status.

| Device variable name for CJ-series Unit | Data type | R/W | Meaning | Description |
|---|---|---|---|---|
| *_TargetPLCMdSta [0] to [63] | ARRAY [0..63] OF BOOL | R | Target Node PLC Operating Flags | TRUE: The target PLC is operating.<br>FALSE: Other than the above.<br>Default: 0000 |

#### ● Target Node PLC Error Flags (EtherNet/IP Unit to CPU Unit) (**_TargetPLCErrSta [0]* to *[63]*)

This variable indicates Controller errors (an OR of major faults, partial faults, and minor faults) in the target PLCs that are connected to the EtherNet/IP Unit when the EtherNet/IP Unit is the originator.

The information in this variable is valid only when the corresponding element in the Normal Target Node Table is TRUE.

If the corresponding element in the Normal Target Node Table is FALSE, the Target Node PLC Error Flags retains the previous operating status.

| Device variable name for CJ-series Unit | Data type | R/W | Meaning | Description |
|---|---|---|---|---|
| *_TargetPLCErrSta [0] to [63] | ARRAY [0..63] OF BOOL | R | Target Node PLC Error Flags | TRUE: There is a major, partial, or minor fault in the CPU Unit of the corresponding target node.<br>FALSE: Other than the above. |

## Unit Status Information

This variable gives the status of the EtherNet/IP Unit.

### ● Unit Status 1 (EtherNet/IP Unit to CPU Unit) (**_*_UnitSta1_**)

| Device variable name for CJ-series Unit | Data type | R/W | Meaning | Description |
|---|---|---|---|---|
| *_UnitSta1 | WORD | R | Unit Status 1 | Bit 00: Unit Error<br>Bit 01: Network Error<br>Bit 04: Unit Memory Error<br>Bit 05: Communications Controller Error<br>Bit 06: IP Address Duplication<br>Bit 09: Link OFF Error<br>Bit 14: Status Area Layout Setting Error |
| *_UnitErr | BOOL | R | Unit Error Occurred | TRUE: An error occurred in the operation of the EtherNet/IP Unit. This variable changes to TRUE when any of the following variables in Unit Status 1 changes to TRUE. (That is, it is an OR of the status of the following variables.)<br>*_NetEr<br>*_UnitMemErr<br>*_LANHwErr<br>*_IPAdrDupErr<br>*_LkOffErr<br>*_UserStaAreaCfgErr<br>FALSE: None of the above errors currently exists. (If any of the errors did occur, it has been cleared.)<br>Default: FALSE |
| *_NetErr | BOOL | R | Network Error Occurred | TRUE: There is one or more network-related errors. (This is an OR of the Communications Status 1 and 3 variables. )<br>FALSE: None of the above errors currently exists. (If any of the errors did occur, it has been cleared.)<br>Default: FALSE |
| *_UnitMemErr | BOOL | R | Unit Memory Error | TRUE: An error occurred in accessing the Unit's internal non-volatile memory (device error).<br>FALSE: A non-volatile memory error has not occurred. Once this error occurs, it is not cleared. (The variable remains TRUE.)<br>Default: FALSE |
| *_LANHwErr | BOOL | R | Communications Controller Error | TRUE: An error occurred in the communications controller.<br>FALSE: A communications controller error has not occurred. This variable remains TRUE until the power supply is cycled.<br>Default: FALSE |

| Device variable name for CJ-series Unit | Data type | R/W | Meaning | Description |
|---|---|---|---|---|
| *_IPAdrDupErr | BOOL | R | IP Address Duplication Error | TRUE: An ARP was sent with the specified IP address, indicating that an IP address duplication was detected. An address duplication is detected if there is an ARP response. If there is an ARP response, it is assumed that there is an address duplication. This variable changes to FALSE if the power is cycled, the Controller is reset, or the EtherNet/IP Unit is restarted. (The Ethernet interface stops.)<br>FALSE: Other than the above.<br>Default: FALSE |
| *_LkOffErr | BOOL | R | Link OFF Error | TRUE: An error occurred in establishing a link with the Ethernet switch.<br>FALSE: A link was normally established with the Ethernet switch.<br>Default: FALSE |
| *_UserStaAreaCfgErr | BOOL | R | Status Area Layout Setting Error | TRUE: There is an error in the layout settings for the allocated words in the CIO Area. In this case, operation uses the default layout of the allocated words in the CIO Area. However, if a word that does not exist is specified, the user-set pattern of the layout of the allocated words in the CIO Area is used for operation and the user settings area is not refreshed.<br>FALSE: There are no problems in the layout settings for the allocated words in the CIO Area.<br>Default: FALSE |

### ● Unit Status 2 (EtherNet/IP Unit to CPU Unit) (*_*UnitSta2*)

| Device variable name for CJ-series Unit | Data type | R/W | Meaning | Description |
|---|---|---|---|---|
| *_UnitSta2 | WORD | R | Unit Status 2 | Bit 00: Online<br>Bit 01: Tag Data Link in Operation<br>Bit 02: Operating IP Address Change<br>Bit 11: User Settings Area Enabled<br>Bit 14: Multiple Switches ON Error<br>Bit 15: Error Log Stored |
| *_ETNOnlineSta | BOOL | R | Online | TRUE: The Unit is online. (The EtherNet/IP Unit can perform communications processing.)<br>FALSE: The Unit is not online. The EtherNet/IP Unit goes offline in the following cases:<br>• IP Address Duplication Error<br>• Ethernet Communications Controller Error (hardware error)<br>• BOOTP Server Error |

| Device variable name for CJ-series Unit | Data type | R/W | Meaning | Description |
|---|---|---|---|---|
| *_TDLOprSta | BOOL | R | Tag Data Link Operating | TRUE: The tag data links are operating. This variable changes to TRUE in the following cases:<br>• The EtherNet/IP Unit was set as an originator and then the power was cycled, the Controller was reset, or the EtherNet/IP Unit was restarted.<br>• The Unit is set as the originator and the Tag Data Link Start Bit was changed to TRUE.<br>FALSE: The tag data links are stopped. This variable changes to FALSE when communications stop in the following cases:<br>• The tag Data Link Stop Bit was set to TRUE.<br>• Hardware error<br>• IP Address Duplication Error<br>• BOOTP Server Error<br>• Basic Ethernet Settings Error<br>• Memory Error (MAC Address Error) |
| *_IPAdrChgErr | BOOL | R | Operating IP Address Change | TRUE: The setting on the node address switches was changed during CPU Unit operation.<br>FALSE: The setting on the node address switches is the same as when CPU Unit operation started. |
| *_UserStaAreaEnblSta | BOOL | R | User Setting Area Enabled | TRUE: The data in the user settings area is enabled.<br>This variable changes to TRUE when the user settings are specified to use for the layout of the allocated words in the CIO Area and refreshing of the user settings area starts.<br>FALSE: The data in the user settings area is disabled.<br>Communications stop in the following cases and this variable changes to FALSE.<br>• When the default layout is specified for the allocated words in the CIO Area<br>• When the user settings specified for the allocated words in the CIO Area and there is an error in the layout settings |
| *_MultiSwOnErr | BOOL | R | Multiple Switches ON Error | TRUE: Two or more software switches where changed to TRUE simultaneously. (Unused software switches are ignored.)<br>FALSE: The next control operation started. |
| *_ErrLogStoreSta | BOOL | R | Error Log Stored | TRUE: An error record was registered in the error log.<br>FALSE: There are no errors in the error log. This variable also changes to FALSE when an error log clear request is received. |

## Communications Status Information

The status of the tag data links and Ethernet is given in the following variables.

### ● Communications Status 1 (EtherNet/IP Unit to CPU Unit) (*_CommSta1)

The status of the tag data links is given in the following variable.

| Device variable name for CJ-series Unit | Data type | R/W | Meaning | Description |
|---|---|---|---|---|
| *_CommSta1 | WORD | R | Communications Status 1 | Bit 00: Verification Error<br>Bit 02: Tag Data Link Error<br>Bit 04: Invalid Communications Parameter<br>Bit 05: Tag Refresh Error<br>Bit 06: Tag Database Error<br>Bit 14: All Tag Data Link Communications Status<br>Bit 15: Tag Data Link Communications Status |
| *_TDLOpnErr | BOOL | R | Verification Error | TRUE: The information registered for a target node in the tag data link parameters is different from the actual node information. Main causes:<br>• The specified target does not exist.<br>• The variable name does not match.<br>• The connection size is different.<br>• Connection resources are insufficient.<br>FALSE: A verification error has not occurred. This variable also changed to FALSE when a verification error is cleared. |
| *_TDLErr | BOOL | R | Tag Data Link Error | TRUE: There were one or more errors in a connection as an originator. This status does not indicate the following errors.<br>• Errors in connections as a target<br>• Connection timeout due to a Link OFF Error with the Ethernet switch<br>FALSE: An error has not occurred in any connection as an originator. |
| *_CommParamErr | BOOL | R | Invalid Communications Parameter | TRUE: Changes to TRUE when there was an error in the validation check of tag data link parameters stored in the Unit's non-volatile memory, and a checksum error occurred. (Includes parameters related to basic Ethernet settings.) The tag data links will stop.<br>FALSE: The validation check of parameters in non-volatile memory was normal. |
| *_TagRefreshErr | BOOL | R | Tag Refresh Error | TRUE: Changes to TRUE when a specified data area or address range is not supported in tag data links.<br>FALSE: The specified data areas and addresses are supported in tag data links. |
| *_TagDbErr | BOOL | R | Tag Database Error | TRUE: A tag database error occurred in the CPU Unit when a variable name is used incorrectly in a setting for the EtherNet/IP Unit (tag data link, status area allocations setting, etc.). (CJ2H-CPU6□-EIP or CJ2M-CPU3□ only)<br>FALSE: Change to FALSE when a variable name is not used in a setting for the EtherNet/IP Unit, when a tag database error has not occurred, or when a previous error has been cleared. |

| Device variable name for CJ-series Unit | | Data type | R/W | Meaning | Description |
|---|---|---|---|---|---|
| | *_TDLAllRunSta | BOOL | R | All Tag Data Links Operating | TRUE: Tag data links are communicating in all connections as the originator.<br>FALSE: A tag data link failed in one or more connections as the originator.<br>(FALSE even if some tag data links are communicating.) |
| | *_TDLRunSta | BOOL | R | Tag Data Links Operating | TRUE: Tag data links are communicating in one or more connections as the originator.<br>FALSE: Not even one tag data link is communicating in connections as the originator.<br>(FALSE even if the Unit is communicating as a target.) |

- **Communications Status 2 (EtherNet/IP Unit to CPU Unit) (*_*CommSta2*)**

The status of Ethernet is given in the following variable.

| Device variable name for CJ-series Unit | | Data type | R/W | Meaning | Description |
|---|---|---|---|---|---|
| *_CommSta2 | | WORD | R | Communications Status 2 | Bit 00: FTP Status<br>Bit 14: Link Status |
| | *_FTPSta | BOOL | R | FTP Status | TRUE: The FTP server is operating (i.e., when there is an FTP client connection).<br>FALSE: The FTP is on standby (i.e., waiting for a client connection). |
| | *_LkSta | BOOL | R | Link Status | TRUE: A link was established with the Ethernet switch.<br>FALSE: The link with the Ethernet switch was stopped. |

- **Communications Status 3 (EtherNet/IP Unit to CPU Unit) (*_*CommSta3*)**

The error status of Ethernet is given in the following variable.

| Device variable name for CJ-series Unit | | Data type | R/W | Meaning | Description |
|---|---|---|---|---|---|
| *_CommSta3 | | WORD | R | Communications Status 3 | Bit 02: Basic Ethernet Setting Error<br>Bit 03: IP Address Table Error<br>Bit 04: IP Router Table Error<br>Bit 05: DNS Server Error<br>Bit 06: Routing Table Error<br>Bit 09: Ethernet Advanced Setting Error<br>Bit 10: BOOTP Server Error<br>Bit 11: SNTP Server Error<br>Bit 14: Address Mismatch<br>Bit 15: Non-volatile Memory Error |
| | *_ETNBaseSetErr | BOOL | R | Basic Ethernet Settings Error | TRUE: One of the following Ethernet settings is not correct.<br>• TCP/IP settings (IP address, subnet mask, or link settings)<br>FALSE: The settings are normal. |
| | *_IPAdrTblErr | BOOL | R | IP Address Table Error | TRUE: The IP address table information is not correct.<br>FALSE: The IP address table information is correct. |
| | *_IPRouterTblErr | BOOL | R | IP Router Table Error | TRUE: The IP router table information is not correct.<br>FALSE: The IP router table information is correct. |

| Device variable name for CJ-series Unit | Data type | R/W | Meaning | Description |
|---|---|---|---|---|
| *_DNSSvrErr | BOOL | R | DNS Server Error | TRUE: The DNS client failed to connect to the DNS server (timeout).<br>FALSE: DNS is not set up. Or, DNS server information is set and the connection was successful. |
| *_RTblErr | BOOL | R | Routing Table Error | TRUE: The routing table information is not correct.<br>FALSE: The routing table information is correct. |
| *_ETNAdvSetErr | BOOL | R | Ethernet Advanced Setting Error | TRUE: One of the following Ethernet settings is not correct.<br>• FINS settings<br>FALSE: The settings are normal. |
| *_BootpSvrErr | BOOL | R | BOOTP Server Error | TRUE: There was a failure to connect to the BOOTP server (timeout).<br>FALSE: The BOOTP is not set up, or the BOOTP server is set up and an IP address was normally obtained from the BOOTP server. |
| *_SNTPSvrErr | BOOL | R | SNTP Server Error | TRUE: The SNTP client failed to connect to the server (timeout).<br>FALSE: SNTP is not set up or SNTP is set up and the connection was successful. |
| *_AdrMismatchErr | BOOL | R | Address Mismatch | TRUE: The target IP address conversion method is set to Automatic generation, but the local IP address' host ID does not match the FINS node address.<br>FALSE: Other than the above setting status. |
| *_MemErr | BOOL | R | Non-volatile Memory Error | TRUE: There is an error in the non-volatile memory in EtherNet/IP Unit.<br>FALSE: The non-volatile memory in EtherNet/IP Unit is normal. |

## Target Node Information

The following variables are available only when the status area layout in the Status Area Settings Display is set to *Default*.
When the status area layout in the Status Area Settings Display is set to *User setting*, instead of this variable, the user-defined variable set in *Mapped area* in the Status Area Settings Display should be referenced. Refer to *3-3 User Definition Settings for the Status Area* for details.

The following variables provide information on the target nodes that are connected to the EtherNet/IP Unit when the EtherNet/IP Unit is the originator.

**Additional Information**

These variables give the status of only nodes 0 to 63. If it is necessary to get the error status of nodes higher than node 63, refer to *3-3 User Definition Settings for the Status Area*.

### ● Registered Target Node Table (EtherNet/IP Unit to CPU Unit) (**_RegTargetSta [0] to [63]*)

This variable provides registration information on tag data links of the target nodes that are connected to the EtherNet/IP Unit when the EtherNet/IP Unit is the originator.

| Device variable name for CJ-series Unit | Data type | R/W | Meaning | Description |
|---|---|---|---|---|
| *_RegTargetSta [0] to [63] | ARRAY [0..63] OF BOOL | R | Registered Target Node Table | TRUE: Indicates that the node's tag data link is registered.<br>FALSE: Indicates that the node's tag data link is not registered. |

### ● Normal Target Node Table (EtherNet/IP Unit to CPU Unit) (**_EstbTargetSta [0] to [63]*)

This variable shows the connection status of the target nodes that are connected with the EtherNet/IP Unit as the originator. The elements of the array change to TRUE after all data for multiple connections for individual target devices is refreshed in the CPU Unit.

| Device variable name for CJ-series Unit | Data type | R/W | Meaning | Description |
|---|---|---|---|---|
| *_EstbRegTargetSta [0] to [63] | ARRAY [0..63] OF BOOL | R | Normal Target Node Table | TRUE: The connection for the corresponding node was established normally.<br>FALSE: The connection for the corresponding node was not established normally. |

3-2 Device Variables for the CJ-series Unit

3

3-2-3 Device Variables for the CJ-series Unit for Status

## 3-2-4 Device Variables for the CJ-series Unit for Setup

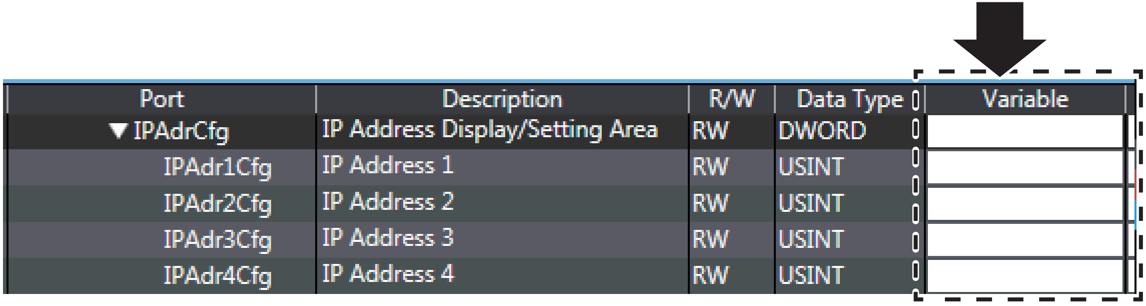
### IP Address Display/Setting Area (*_*IPAdrCfg*)

When confirming and configuring the local IP address, define the variables in the following procedure.

**1** Confirm that device variables are not assigned to the IP Address Display/Setting Area (*_IPAdrCfg) of the I/O port.

If they are already assigned, delete the relevant device variables to reset the assignment.

Confirm that device variables are not assigned.

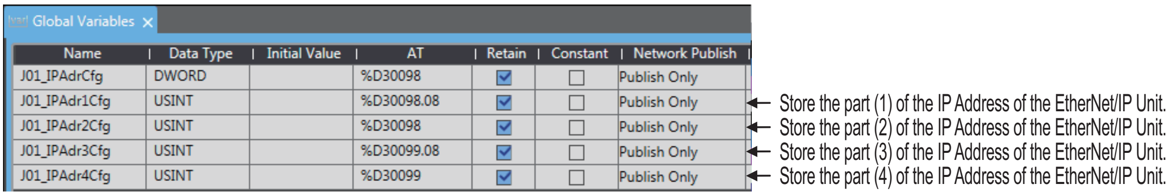


| Port | Description | R/W | Data Type | Variable |
|---|---|---|---|---|
| ▼IPAdrCfg | IP Address Display/Setting Area | RW | DWORD | |
| IPAdr1Cfg | IP Address 1 | RW | USINT | |
| IPAdr2Cfg | IP Address 2 | RW | USINT | |
| IPAdr3Cfg | IP Address 3 | RW | USINT | |
| IPAdr4Cfg | IP Address 4 | RW | USINT | |

**2** Make the following settings.

- Register the IP Address Display/Setting Area in the global variable table.
- Set the address for the AT specification to the AT attribute of the registered variable.
  The address can be calculated below.
  IP Address Display/Setting Area = D30000 + Unit number × 100 + 98, and 99 (Unit number: 0 to 15)
- Check the Retain attribute of the registered variable.
- To be monitored from external devices, set the Network Publish attribute to Publish Only.

For example, if the variable name is "J01_IPAdr * Cfg" and the Unit number of the EtherNet/IP Unit is 0, the address of the IP Address Display/Setting Area is D30098 and D30099, and the global variable is defined as shown below.



Global Variables

| Name | Data Type | Initial Value | AT | Retain | Constant | Network Publish |
|---|---|---|---|---|---|---|
| J01_IPAdrCfg | DWORD | | %D30098 | ☑ | ☐ | Publish Only |
| J01_IPAdr1Cfg | USINT | | %D30098.08 | ☑ | ☐ | Publish Only |
| J01_IPAdr2Cfg | USINT | | %D30098 | ☑ | ☐ | Publish Only |
| J01_IPAdr3Cfg | USINT | | %D30099.08 | ☑ | ☐ | Publish Only |
| J01_IPAdr4Cfg | USINT | | %D30099 | ☑ | ☐ | Publish Only |

← Store the part (1) of the IP Address of the EtherNet/IP Unit.
← Store the part (2) of the IP Address of the EtherNet/IP Unit.
← Store the part (3) of the IP Address of the EtherNet/IP Unit.
← Store the part (4) of the IP Address of the EtherNet/IP Unit.

The IP address is stored as shown below.

| Byte | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| First byte | (1) | | | | | | | | (2) | | | | | | | |
| First byte + 1 | (3) | | | | | | | | (4) | | | | | | | |

IP Address: (1).(2).(3).(4) (hex)

If the IP address is set to a value other than 0.0.0.0 in the TCP/IP Configuration, the variable set in Step 2 will act as an IP Address Display Area and the IP address set in the TCP/IP Configuration will be read and stored here when the power is turned ON or the Unit is restarted.

If the IP address in the TCP/IP Configuration is set to 0.0.0.0, the value of the variable (set in Step 2) is read by the EtherNet/IP Unit when the power is turned ON or the Unit is restarted and is used as the IP address.

If the variable (set in Step 2) and the TCP/IP Configuration are both set to 0.0.0.0, the default IP address (192.168.250.Node_address) will be used. For details on the IP address settings, refer to *Section 4 Determining IP Addresses*.

### Precautions for Correct Use

- If the IP address is set to a value other than 0.0.0.0 in the TCP/IP Settings, that address is written to this variable even if you set the variable (set in Step 2) to a different IP address.
- It is not possible to set the following IP addresses.
  - IP addresses where all network number bits are 0 or 1.
  - IP addresses where all host number bits are 0 or 1.
  - IP addresses where all subnet number bits are 1.
  - IP addresses that start with 127 (7F hexadecimal, e.g., 127.35.21.16).

## 3-2-5 Device Variables for the CJ-series Unit Socket Service-related Data (CJ1W-EIP21S Only)

The following variables are available only when the **Layout type** is set to *User setting* in the **Status area layout** category in the Status Area Settings Display.

Refer to *3-3 User Definition Settings for the Status Area* for the user setting of the layout type in the **Status area layout** category in the Status Area Settings Display.

### Socket Service Request Switches 1 to 8 (CPU Unit to EtherNet/IP Unit) (*_Skt1Cmd* to *_Skt8Cmd*)

When you request socket services by manipulating variables, the following variables can be manipulated.
Refer to *Section 13 Socket Services* for details.

Socket Service Request Switch 1

| Device variable for CJ-series Units | Type | R/W | Meaning | Function |
|---|---|---|---|---|
| *_Skt1Cmd | BYTE | R/W | Socket Service Request Switch 1 | Requests socket services. |
| *_Skt1UdpOpenCmd | BOOL | R/W | UDP Open Request Switch | Opens UDP socket when switch is changed to TRUE.<br>FALSE: Open processing completed<br>Default: FALSE |
| *_Skt1TcpPassiveCmd | BOOL | R/W | TCP Passive Open Request Switch | Opens TCP socket when switch is changed to TRUE.<br>FALSE: Open processing completed (connection established)<br>Default: FALSE |
| *_Skt1TcpActiveCmd | BOOL | R/W | TCP Active Open Request Switch | Opens TCP socket when switch is changed to TRUE.<br>FALSE: Open processing completed (connection established)<br>Default: FALSE |
| *_Skt1SendCmd | BOOL | R/W | Send Request Switch | Executes send processing when switch is changed to TRUE.<br>(The protocol (TCP/UDP) is determined when the socket is opened.)<br>FALSE: Send processing completed<br>Default: FALSE |
| *_Skt1RcvCmd | BOOL | R/W | Receive Request Switch | Executes receive processing when switch is changed to TRUE.<br>(The protocol (TCP/UDP) is determined when the socket is opened.)<br>FALSE: Receive processing completed<br>Default: FALSE |
| *_Skt1CloseCmd | BOOL | R/W | Close Request Switch | Closes TCP socket when switch is changed to TRUE.<br>FALSE: Close processing completed<br>Default: FALSE |

Subsequent Socket Service Request Switches 2 to 8 are assigned similar variables.
(This table is omitted for Socket Service Request Switches 2 to 7.)

Socket Service Request Switch 8

| Device variable for CJ-series Units | | Type | R/W | Meaning | Function |
|---|---|---|---|---|---|
| *_Skt8Cmd | | BYTE | R/W | Socket Service Request Switch 8 | Requests socket services. |
| | *_Skt8UdpOpenCmd | BOOL | R/W | UDP Open Request Switch | Opens UDP socket when switch is changed to TRUE.<br>FALSE: Open processing completed<br>Default: FALSE |
| | *_Skt8TcpPassiveCmd | BOOL | R/W | TCP Passive Open Request Switch | Opens TCP socket when switch is changed to TRUE.<br>FALSE: Open processing completed (connection established)<br>Default: FALSE |
| | *_Skt8TcpActiveCmd | BOOL | R/W | TCP Active Open Request Switch | Opens TCP socket when switch is changed to TRUE.<br>FALSE: Open processing completed (connection established)<br>Default: FALSE |
| | *_Skt8SendCmd | BOOL | R/W | Send Request Switch | Executes send processing when switch is changed to TRUE.<br>(The protocol (TCP/UDP) is determined when the socket is opened.)<br>FALSE: Send processing completed<br>Default: FALSE |
| | *_Skt8RcvCmd | BOOL | R/W | Receive Request Switch | Executes receive processing when switch is changed to TRUE.<br>(The protocol (TCP/UDP) is determined when the socket is opened.)<br>FALSE: Receive processing completed<br>Default: FALSE |
| | *_Skt8CloseCmd | BOOL | R/W | Close Request Switch | Closes TCP socket when switch is changed to TRUE.<br>FALSE: Close processing completed<br>Default: FALSE |

## UDP Socket No. 1 to No. 8 Status (EtherNet/IP Unit to CPU Unit) (*_Skt1UdpSta to *_Skt8UdpSta)

Refer to *Section 13 Socket Services* for details.

Socket No. 1

| Device variable for CJ-series Units | | Type | R/W | Meaning | Function |
|---|---|---|---|---|---|
| *_Skt1UdpSta | | BYTE | R | Status of UDP Sockets 1 | Indicates the status information on the socket service. |
| | *_Skt1UdpOpening | BOOL | R | Opening Flag | TRUE: Open processing in progress (Always TRUE when an open request is received)<br>FALSE: Open processing completed<br>Default: FALSE |
| | *_Skt1UdpRcving | BOOL | R | Receiving Flag | TRUE: Receive processing in progress (Always TRUE if a receive request is received with the high-speed service option disabled, or remaining FALSE if it is received with the option enabled)<br>FALSE: Receive processing completed<br>Default: FALSE |
| | *_Skt1UdpSending | BOOL | R | Sending Flag | TRUE: Send processing in progress (Always TRUE if a send request is received with the high-speed service option disabled, or remaining FALSE if it is received with the option enabled)<br>FALSE: Send processing completed<br>Default: FALSE |
| | *_Skt1UdpClosing | BOOL | R | Closing Flag | TRUE: Close processing in progress (Always TRUE when a close request is received)<br>FALSE: Close processing completed<br>Default: FALSE |
| | *_Skt1UdpAreaErr | BOOL | R | Results Storage Error Flag | TRUE: Error in specifying the Results Storage Area when using the socket service function with the SendCmd instruction<br>(The flag changes to TRUE when one of the Opening Flag, Receiving Flag, Sending Flag, or Closing Flag is changed to FALSE (when processing is completed.))<br>FALSE: Next request received<br>Default: FALSE |
| | *_Skt1UdpOpened | BOOL | R | UDP Open Flag | TRUE: Open processing completed<br>FALSE: Close processing completed<br>(The flag remains FALSE if the open processing is completed abnormally.)<br>Default: FALSE |

Subsequent sockets, Socket No. 2 to No. 8, are assigned similar variables.
(This table is omitted for Socket No. 2 to No. 7.)

Socket No. 8

| Device variable for CJ-series Units | | Type | R/W | Meaning | Function |
|---|---|---|---|---|---|
| *_Skt8UdpSta | | BYTE | R | Status of UDP Sockets 8 | Indicates the status information on the socket service. |
| | *_Skt8UdpOpening | BOOL | R | Opening Flag | TRUE: Open processing in progress (Always TRUE when an open request is received)<br>FALSE: Open processing completed<br>Default: FALSE |
| | *_Skt8UdpRcving | BOOL | R | Receiving Flag | TRUE: Receive processing in progress (Always TRUE if a receive request is received with the high-speed service option disabled, or remaining FALSE if it is received with the option enabled)<br>FALSE: Receive processing completed<br>Default: FALSE |
| | *_Skt8UdpSending | BOOL | R | Sending Flag | TRUE: Send processing in progress (Always TRUE if a send request is received with the high-speed service option disabled, or remaining FALSE if it is received with the option enabled)<br>FALSE: Send processing completed<br>Default: FALSE |
| | *_Skt8UdpClosing | BOOL | R | Closing Flag | TRUE: Close processing in progress (Always TRUE when a close request is received)<br>FALSE: Close processing completed<br>Default: FALSE |
| | *_Skt8UdpAreaErr | BOOL | R | Results Storage Error Flag | TRUE: Error in specifying the Results Storage Area when using the socket service function with the SendCmd instruction<br>(The flag changes to TRUE when one of the Opening Flag, Receiving Flag, Sending Flag, or Closing Flag is changed to FALSE (when processing is completed.))<br>FALSE: Next request received<br>Default: FALSE |
| | *_Skt8UdpOpened | BOOL | R | UDP Open Flag | TRUE: Open processing completed<br>FALSE: Close processing completed<br>(The flag remains FALSE if the open processing is completed abnormally.)<br>Default: FALSE |

## TCP Socket No. 1 to No. 8 Status (EtherNet/IP Unit to CPU Unit) (*_Skt1TcpSta* to *_Skt8TcpSta*)

Refer to *Section 13 Socket Services* for details.

Socket No. 1

| Device variable for CJ-series Units | Type | R/W | Meaning | Function |
|---|---|---|---|---|
| *_Skt1TcpSta | BYTE | R | Status of TCP Sockets 1 | Indicates the status information on the socket service. |
| *_Skt1TcpOpening | BOOL | R | Opening Flag | TRUE: Open processing in progress (Always TRUE when an open request is received)<br>FALSE: Open processing completed<br>Default: FALSE |
| *_Skt1TcpRcving | BOOL | R | Receiving Flag | TRUE: Receive processing in progress (Always TRUE if a receive request is received with the high-speed service option disabled, or remaining FALSE if it is received with the option enabled)<br>FALSE: Receive processing completed<br>Default: FALSE |
| *_Skt1TcpSending | BOOL | R | Sending Flag | TRUE: Send processing in progress (Always TRUE if a send request is received with the high-speed service option disabled, or remaining FALSE if it is received with the option enabled)<br>FALSE: Send processing completed<br>Default: FALSE |
| *_Skt1TcpClosing | BOOL | R | Closing Flag | TRUE: Close processing in progress (Always TRUE when a close request is received)<br>FALSE: Close processing completed<br>Default: FALSE |
| *_Skt1TcpDatRcved | BOOL | R | Data Received Flag | TRUE: Data received from the remote node using an open TCP socket<br>FALSE: Receive processing request executed using an open TCP socket<br>Default: FALSE |
| *_Skt1TcpAreaErr | BOOL | R | Results Storage Error Flag | TRUE: Error in specifying the Results Storage Area when using the socket service function with the SendCmd instruction<br>(The flag changes to TRUE when one of the Opening Flag, Receiving Flag, Sending Flag, or Closing Flag is changed to FALSE (when processing is completed.))<br>FALSE: Next request received<br>Default: FALSE |
| *_Skt1TcpOpened | BOOL | R | TCP Open Flag | TRUE: Open process completed (during connection)<br>FALSE: Close processing completed<br>(The flag remains FALSE if the open processing is completed abnormally.)<br>Default: FALSE |

Subsequent sockets, Socket No. 2 to No. 8, are assigned similar variables.
(This table is omitted for Socket No. 2 to No. 7.)

Socket No. 8

| Device variable for CJ-series Units | Type | R/W | Meaning | Function |
|---|---|---|---|---|
| *_Skt8TcpSta | BYTE | R | Status of TCP Sockets 8 | Indicates the status information on the socket service. |
| *_Skt8TcpOpening | BOOL | R | Opening Flag | TRUE: Open processing in progress (Always TRUE when an open request is received)<br>FALSE: Open processing completed<br>Default: FALSE |
| *_Skt8TcpRcving | BOOL | R | Receiving Flag | TRUE: Receive processing in progress (Always TRUE if a receive request is received with the high-speed service option disabled, or remaining FALSE if it is received with the option enabled)<br>FALSE: Receive processing completed<br>Default: FALSE |
| *_Skt8TcpSending | BOOL | R | Sending Flag | TRUE: Send processing in progress (Always TRUE if a send request is received with the high-speed service option disabled, or remaining FALSE if it is received with the option enabled)<br>FALSE: Send processing completed<br>Default: FALSE |
| *_Skt8TcpClosing | BOOL | R | Closing Flag | TRUE: Close processing in progress (Always TRUE when a close request is received)<br>FALSE: Close processing completed<br>Default: FALSE |
| *_Skt8TcpDatRcved | BOOL | R | Data Received Flag | TRUE: Data received from the remote node using an open TCP socket<br>FALSE: Receive processing request executed using an open TCP socket<br>Default: FALSE |
| *_Skt8TcpAreaErr | BOOL | R | Results Storage Error Flag | TRUE: Error in specifying the Results Storage Area when using the socket service function with the SendCmd instruction<br>(The flag changes to TRUE when one of the Opening Flag, Receiving Flag, Sending Flag, or Closing Flag is changed to FALSE (when processing is completed.))<br>FALSE: Next request received<br>Default: FALSE |
| *_Skt8TcpOpened | BOOL | R | TCP Open Flag | TRUE: Open process completed (during connection)<br>FALSE: Close processing completed<br>(The flag remains FALSE if the open processing is completed abnormally.)<br>Default: FALSE |

## Number of Bytes Received at TCP Socket No. 1 to 8 (EtherNet/IP Unit to CPU Unit) (*_*Skt1TcpRcvDatByte* to *_*Skt8TcpRcvDatByte*)

These variables store the number of bytes of received data in the reception buffer for each TCP socket. Up to 4,096 bytes of data are stored in the reception buffer, but the value stored is within the range (maximum: 1,984 bytes) that can be set by manipulating the variable or sending the receive request in the SendCmd instruction.

0000 hex: 0 bytes

07C0 hex: 1,984 bytes

In conjunction with these variables, the Data Received Flag of the UDP/TCP Socket No. 1 to No. 8 Status is changed to TRUE/FALSE. When the receive request is sent by manipulating the variable or by executing the SendCmd instruction, the value of this variable is temporarily set to 0000 hex.

If the data remains in the reception buffer after the receive request processing is complete, the number of remaining bytes is stored in the Number of Bytes Received at TCP Socket and the Data Received Flag changes to TRUE again.

Receive requests should be executed after confirming that the required data is contained in the number of bytes received.

| Device variable for CJ-series Units | Type | R/W | Meaning | Function |
|---|---|---|---|---|
| *_Skt1TcpRcvDatByte | UINT | R | TCP Socket No.1 Number of Bytes Received | Indicates the number of bytes of data received at TCP Socket No. 1. |
| *_Skt2TcpRcvDatByte | UINT | R | TCP Socket No.2 Number of Bytes Received | Indicates the number of bytes of data received at TCP Socket No. 2. |
| *_Skt3TcpRcvDatByte | UINT | R | TCP Socket No.3 Number of Bytes Received | Indicates the number of bytes of data received at TCP Socket No. 3. |
| *_Skt4TcpRcvDatByte | UINT | R | TCP Socket No.4 Number of Bytes Received | Indicates the number of bytes of data received at TCP Socket No. 4. |
| *_Skt5TcpRcvDatByte | UINT | R | TCP Socket No.5 Number of Bytes Received | Indicates the number of bytes of data received at TCP Socket No. 5. |
| *_Skt6TcpRcvDatByte | UINT | R | TCP Socket No.6 Number of Bytes Received | Indicates the number of bytes of data received at TCP Socket No. 6. |
| *_Skt7TcpRcvDatByte | UINT | R | TCP Socket No.7 Number of Bytes Received | Indicates the number of bytes of data received at TCP Socket No. 7. |
| *_Skt8TcpRcvDatByte | UINT | R | TCP Socket No.8 Number of Bytes Received | Indicates the number of bytes of data received at TCP Socket No. 8. |

## TCP Socket No. 1 to 8 Connection Status (EtherNet/IP Unit to CPU Unit) (*_Skt1TcpConSta to *_Skt8TcpConSta)

These variables store the connection status of each TCP socket as a code.

| Device variable for CJ-series Units | Type | R/W | Meaning | Function |
|---|---|---|---|---|
| *_Skt1TcpConSta | WORD | R | TCP Socket No.1 Connection Status | Indicates the connection status of TCP Socket No. 1. |
| *_Skt2TcpConSta | WORD | R | TCP Socket No.2 Connection Status | Indicates the connection status of TCP Socket No. 2. |
| *_Skt3TcpConSta | WORD | R | TCP Socket No.3 Connection Status | Indicates the connection status of TCP Socket No. 3. |
| *_Skt4TcpConSta | WORD | R | TCP Socket No.4 Connection Status | Indicates the connection status of TCP Socket No. 4. |
| *_Skt5TcpConSta | WORD | R | TCP Socket No.5 Connection Status | Indicates the connection status of TCP Socket No. 5. |
| *_Skt6TcpConSta | WORD | R | TCP Socket No.6 Connection Status | Indicates the connection status of TCP Socket No. 6. |
| *_Skt7TcpConSta | WORD | R | TCP Socket No.7 Connection Status | Indicates the connection status of TCP Socket No. 7. |
| *_Skt8TcpConSta | WORD | R | TCP Socket No.8 Connection Status | Indicates the connection status of TCP Socket No. 8. |

The meanings of the connection status codes are shown in the following table.

| Code | Status | Meaning |
|---|---|---|
| 0 | CLOSED | Connection closed. |
| 1 | LISTEN | Waiting for connection. |
| 2 | SYN SENT | SYN sent in active status. |
| 3 | SYN RECEIVED | SYN received and sent. |
| 4 | ESTABLISHED | Already established. |
| 5 | CLOSE WAIT | FIN received and waiting for completion. |
| 6 | FIN WAIT1 | Completed and FIN sent. |
| 7 | CLOSING | Completed and exchanged FIN.<br>Awaiting ACK. |
| 8 | LAST ACK | FIN sent and completed.<br>Awaiting ACK. |
| 9 | FIN WAIT2 | Completed and ACK received.<br>Awaiting FIN. |
| A | TIME WAIT | After closing, pauses twice the maximum segment life (2MSL). |

## Socket Service Parameter Area 1 to 8 (EtherNet/IP Unit to CPU Unit) (*_Skt1ParamArea* to *_Skt8ParamArea*)

To request socket services by manipulating variables, you need to configure in advance the required settings in these parameter areas.

However, the parameters to be used differ depending on the service request.

Refer to *Section 13 Socket Services* for the parameter areas to be used for each service request, as well as their settings and usage.

Socket Service Parameter Area 1

| Device variable for CJ-series Units | Type | R/W | Meaning | Function |
|---|---|---|---|---|
| *_Skt1ParamArea | ARRAY[0..9] OF WORD | R/W | Socket Services Parameter Area 1 | Set the socket service parameters. |
| *_Skt1No | USINT | R/W | UDP/TCP Socket Number | Specify the number of the UDP or TCP socket to open. |
| *_Skt1Option | BYTE | R/W | Socket Option | Set the specifications of the keep-alive function for the TCP OPEN REQUEST command.<br>Specify the number of the UDP or TCP socket to open. |
| *_Skt1KeepAlive | BOOL | R/W | Keep-alive Function Used | For the TCP OPEN REQUEST (ACTIVE or PASSIVE) command, specify whether or not the keep-alive function is to be used.<br>TRUE:Use<br>FALSE:Do not use |
| *_Skt1Linger | BOOL | R/W | Linger Function Used | Specify whether or not the linger function is to be used.<br>TRUE:Use<br>FALSE:Do not use |
| *_Skt1LocalPortNo | UINT | R/W | Local UDP/TCP Port Number (0000 to FFFF hex) | Specify the number of the UDP or TCP port for the socket to use for communications. |
| *_Skt1RemoteIPAdr | DWORD | R/W | Remote IP Address (00000000 to FFFFFFFF hex) | Specify the IP address of the remote device. |
| *_Skt1RemoteIPAdr1 | USINT | R/W | 1st Byte of Remote IP Address (16#00 to 16#FF) | Specify the first byte of the remote IP address. |
| *_Skt1RemoteIPAdr2 | USINT | R/W | 2nd Byte of Remote IP Address (16#00 to 16#FF) | Specify the second byte of the remote IP address. |
| *_Skt1RemoteIPAdr3 | USINT | R/W | 3rd Byte of Remote IP Address (16#00 to 16#FF) | Specify the third byte of the remote IP address. |
| *_Skt1RemoteIPAdr4 | USINT | R/W | 4th Byte of Remote IP Address (16#00 to 16#FF) | Specify the fourth byte of the remote IP address. |
| *_Skt1RemotePortNo | UINT | R/W | Remote UDP/TCP Port Number (0000 to FFFF hex) | Specify the UDP or TCP port number used by the remote device. |
| *_Skt1SendRcvByte | UINT | R/W | Number Of Send/Receive Bytes (0000 to 07C0 hex (1984)) | Specify the number of bytes to send or receive when a send or receive request is made. |
| *_Skt1SendRcvAdr | DWORD | R/W | Send/Receive Data Address | Specify the first word of the destination address for the data to send when a send request is made, or the first word of the storage address for the data to receive when a receive request is made. |

| Device variable for CJ-series Units | Type | R/W | Meaning | Function |
|---|---|---|---|---|
| *_Skt1SendRcvAdrType | BYTE | R/W | Area Type (#16#00 to #16#FF) | Specify the area type. |
| *_Skt1SendRcvAdrCh1 | BYTE | R/W | Channel Address(Upper 2 digits) | Specify the upper two digits of the word address. |
| *_Skt1SendRcvAdrCh2 | BYTE | R/W | Channel Address(Lower 2 digits) | Specify the lower two digits of the word address. |
| *_Skt1Timeout | UINT | R/W | Timeout Value (0000 to FFFF hex) | Specify the time limit for completion of communications from the time that the Receive Request Switch (TCP or UDP) or the TCP Passive Open Request Switch is changed to TRUE. |
| *_Skt1RespCode | WORD | R | Response Code | The results of processing will be stored when processing of a socket service request is completed. |

Subsequent Socket Service Parameter Areas 2 to 8 are assigned similar variables.

(This table is omitted for Socket Services Parameter Areas 2 to 7.)

Socket Service Parameter Area 8

| Device variable for CJ-series Units | Type | R/W | Meaning | Function |
|---|---|---|---|---|
| *_Skt8ParamArea | ARRAY[0..9] OF WORD | R/W | Socket Services Parameter Area 8 | Set the socket service parameters. |
| *_Skt8No | USINT | R/W | UDP/TCP Socket Number | Specify the number of the UDP or TCP socket to open. |
| *_Skt8Option | BYTE | R/W | Socket Option | Set the specifications of the keep-alive function for the TCP OPEN REQUEST command.<br>Specify the number of the UDP or TCP socket to open. |
| *_Skt8KeepAlive | BOOL | R/W | Keep-alive Function Used | For the TCP OPEN REQUEST (ACTIVE or PASSIVE) command, specify whether or not the keep-alive function is to be used.<br>TRUE:Use<br>FALSE:Do not use |
| *_Skt8Linger | BOOL | R/W | Linger Function Used | Specify whether or not the linger function is to be used.<br>TRUE:Use<br>FALSE:Do not use |
| *_Skt8LocalPortNo | UINT | R/W | Local UDP/TCP Port Number (0000 to FFFF hex) | Specify the number of the UDP or TCP port for the socket to use for communications. |
| *_Skt8RemoteIPAdr | DWORD | R/W | Remote IP Address (00000000 to FFFFFFFF hex) | Specify the IP address of the remote device. |
| *_Skt8RemoteIPAdr1 | USINT | R/W | 1st Byte of Remote IP Address (16#00 to 16#FF) | Specify the first byte of the remote IP address. |
| *_Skt8RemoteIPAdr2 | USINT | R/W | 2nd Byte of Remote IP Address (16#00 to 16#FF) | Specify the second byte of the remote IP address. |
| *_Skt8RemoteIPAdr3 | USINT | R/W | 3rd Byte of Remote IP Address (16#00 to 16#FF) | Specify the third byte of the remote IP address. |
| *_Skt8RemoteIPAdr4 | USINT | R/W | 4th Byte of Remote IP Address (16#00 to 16#FF) | Specify the fourth byte of the remote IP address. |
| *_Skt8RemotePortNo | UINT | R/W | Remote UDP/TCP Port Number (0000 to FFFF hex) | Specify the UDP or TCP port number used by the remote device. |
| *_Skt8SendRcvByte | UINT | R/W | Number Of Send/Receive Bytes (0000 to 07C0 hex (1984)) | Specify the number of bytes to send or receive when a send or receive request is made. |
| *_Skt8SendRcvAdr | DWORD | R/W | Send/Receive Data Address | Specify the first word of the destination address for the data to send when a send request is made, or the first word of the storage address for the data to receive when a receive request is made. |
| *_Skt8SendRcvAdrType | BYTE | R/W | Area Type (#16#00 to #16#FF) | Specify the area type. |
| *_Skt8SendRcvAdrCh1 | BYTE | R/W | Channel Address(Upper 2 digits) | Specify the upper two digits of the word address. |

| Device variable for CJ-series Units | Type | R/W | Meaning | Function |
|---|---|---|---|---|
| *_Skt8SendRcvAdrCh2 | BYTE | R/W | Channel Address(Lower 2 digits) | Specify the lower two digits of the word address. |
| *_Skt8Timeout | UINT | R/W | Timeout Value (0000 to FFFF hex) | Specify the time limit for completion of communications from the time that the Receive Request Switch (TCP or UDP) or the TCP Passive Open Request Switch is changed to TRUE. |
| *_Skt8RespCode | WORD | R | Response Code | The results of processing will be stored when processing of a socket service request is completed. |

## Force Socket Close Switch

Refer to *Force Socket Close Switch* (*_SktForceCloseCmd) in *3-2-2 Device Variables for the CJ-series Unit for Software Switches*.

# 3-3 User Definition Settings for the Status Area

## 3-3-1 Introduction to User Definition Settings

If the EtherNet/IP Unit is the originator of a connection, you can store the corresponding target node status information in a user-defined variable instead of the default device variable for the CJ-series Unit.

Although the default device variable for CJ-series Unit provides information only for nodes 0 to 63, you can obtain the information for 256 nodes if you use user definition settings for an area.

### Additional Information

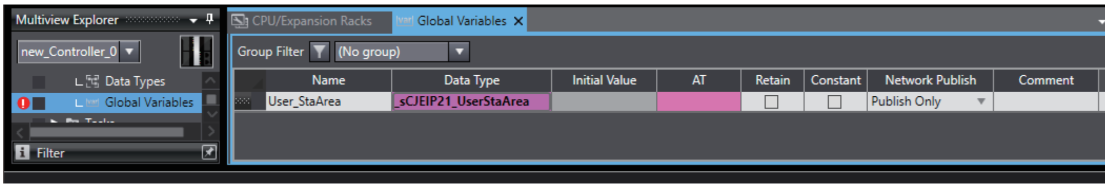
The information in this variable is valid only when the corresponding element in the Normal Target Node Table is TRUE. If the corresponding element in the Normal Target Node Table is FALSE, the Target Node PLC Operating Flags retains the previous operating status.

## 3-3-2 Setting User Definitions

**1** Double-click **CPU/Expansion Racks** under **Configuration and Setup** in the Sysmac Studio, right-click the EtherNet/IP Unit, and select **Register Data Types for Units** from the menu. This operation makes the data type *_sCJEIP21_User_StaArea* available.

**2** In the Multiview Explorer of the Sysmac Studio, select ***Programming* – *Data* – *Global Variables***.

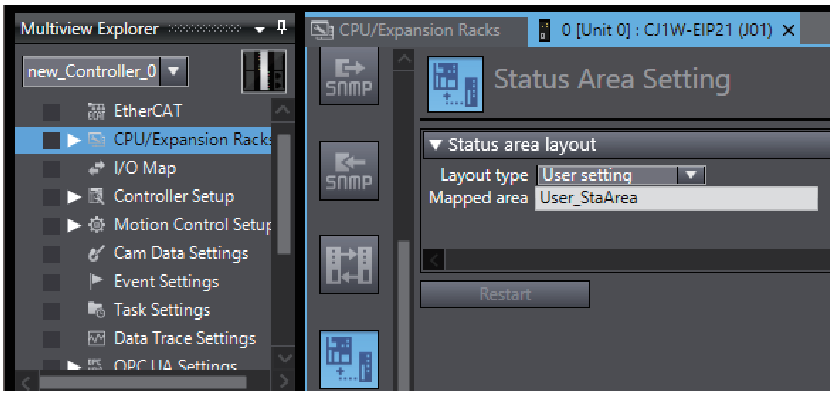
**3** In the global variables table, set a user-defined variable with a Name (any name), Data Type (*_sCJEIP21_User_StaArea*), and Network Publish (*Publish Only* or *Output*).



| Name | Data Type | Initial Value | AT | Retain | Constant | Network Publish | Comment |
|---|---|---|---|---|---|---|---|
| User_StaArea | _sCJEIP21_UserStaArea | | | □ | □ | Publish Only | |

**4** Double-click **Special Unit Setup** under **Configurations and Setup** in the Sysmac Studio and, in the Status Area Settings Display, set the *Layout type* to *User setting*.

**5** Specify the variable set in step 3 in **Mapped area**.



Layout type: User setting
Mapped area: User_StaArea

## 3-3-3 Accessing the User-defined Status Area

In the user program, you can specify the user-defined variable that is specified for the status area to access the status area.

The user-defined variable that is specified for the status area is defined by using the preregistered data type for the user-defined status area (_*sCJEIP21_User_StaArea*).

To use this data type, you must make the following setting in advance in the Sysmac Studio to register the data type so that you can use it.

Right-click the Unit in the CPU/Expansion Racks Tab Page of the Controller Configurations and Setup, and then select ***Register Data Types for Units*** from the menu.

### Data Type for the Status Area

- **Data Type for the Status Area (_*sCJEIP21_User_StaArea*)**

The data type contains a different member for each type of status information.

Each member uses the data type for storing target node information (_*uCJEIP21_TargetSta*).

| Structure data type name | Member | Data type |
|---|---|---|
| _sCJEIP21_User_StaArea | RegTargetSta (Registered Target Node Table) | _uCJEIP21_TargetSta |
| | EstbTargetSta (Normal Target Node Table) | _uCJEIP21_TargetSta |
| | TargetPLCMdSta (Target Node PLC Operating Flags) | _uCJEIP21_TargetSta |
| | TargetPLCErrSta (Target Node PLC Error Flags) | _uCJEIP21_TargetSta |

Data Type for Storing Target Node Information (_uCJEIP21_TargetSta)

Information is stored for nodes 0 to 255.

This variable is a union, which allows access with different data types.

You can specify different data types to access the data: a BOOL array variable with 256 elements, 256 BOOL variables, or WORD variables.

| Structure data type name | Member | Data type |
|---|---|---|
| _uCJEIP21_TargetSta | TargetSta | array[0..255] of BOOL |
| | TargetStaWd | array[0..15] of WORD |

## 3-3-4 Status Information Assigned to the User-defined Area

### Registered Target Node Table (EtherNet/IP Unit to CPU Unit)

This variable gives the registration status of the target nodes. It is valid only when the EtherNet/IP Unit is the originator of the connection.

x. RegTargetSta.TargetSta [0] to [255]

x. RegTargetSta.TargetStaWd [0] to [15]

x: The name of the user-defined variable that is specified for the status area in the user settings.

| Bit | Meaning | Status | Manipulated by | Unit operation | Reference |
|---|---|---|---|---|---|
| --- | Registered Target Node Table | TRUE | EtherNet/IP Unit | The tag data links are registered. | *3-2-3 Device Variables for the CJ-series Unit for Status* |
| | | FALSE | EtherNet/IP Unit | The tag data links are not registered. | |

## Normal Target Node Table (EtherNet/IP Unit to CPU Unit)

This variable gives the connection status of the target nodes. The elements of this variable change to TRUE after all data for multiple connections for individual target devices is refreshed in the CPU Unit.

x.EstbTargetSta.TargetSta [0] to [255]

x. EstbTargetSta.TargetStaWd [0] to [15]

x: The name of the user-defined variable that is specified for the status area in the user settings.

| Bit | Meaning | Status | Manipulated by | Unit operation | Reference |
|---|---|---|---|---|---|
| --- | Normal Target Node Table | TRUE | EtherNet/IP Unit | The corresponding element changes to TRUE after all data for multiple connections for the target device is refreshed in the CPU Unit. | *3-2-3 Device Variables for the CJ-series Unit for Status* |
| | | FALSE | EtherNet/IP Unit | Other than the above. | |

## Target Node PLC Operating Flags (EtherNet/IP Unit to CPU Unit)

This variable gives the operating status of the target PLCs. It is valid only when the EtherNet/IP Unit is the originator. The elements of this variable are valid only when the corresponding elements in the Normal Target Node Table is TRUE. If the corresponding element in Normal Target Node Table is FALSE, the element in the Target Node PLC Operating Flags gives the previous operating status.

x.TargetPLCMdSta.TargetSta [0] to [255]

x. TargetPLCMdSta.TargetStaWd [0] to [15]

x: The name of the user-defined variable that is specified for the status area in the user settings.

| Bit | Meaning | Status | Manipulated by | Unit operation | Reference |
|---|---|---|---|---|---|
| --- | Target Node PLC Operating Flags | TRUE | EtherNet/IP Unit | The target PLC is operating. | *3-2-3 Device Variables for the CJ-series Unit for Status* |
| | | FALSE | EtherNet/IP Unit | Other than the above. | |

## Target Node PLC Error Flags (EtherNet/IP Unit to CPU Unit)

This variable indicates Controller errors (an OR of major faults, partial faults, and minor faults) in the target PLCs.

The elements of this variable are valid only when the corresponding elements in the Normal Target Node Table is TRUE. If the corresponding element in Normal Target Node Table is FALSE, the element in the Target Node PLC Error Flags indicates the previous error status.

x.TargetPLCErrSta.TargetSta [0] to [255]

x. TargetPLCErrSta.TargetStaWd [0] to [15]

x: The name of the user-defined variable that is specified for the status area in the user settings.

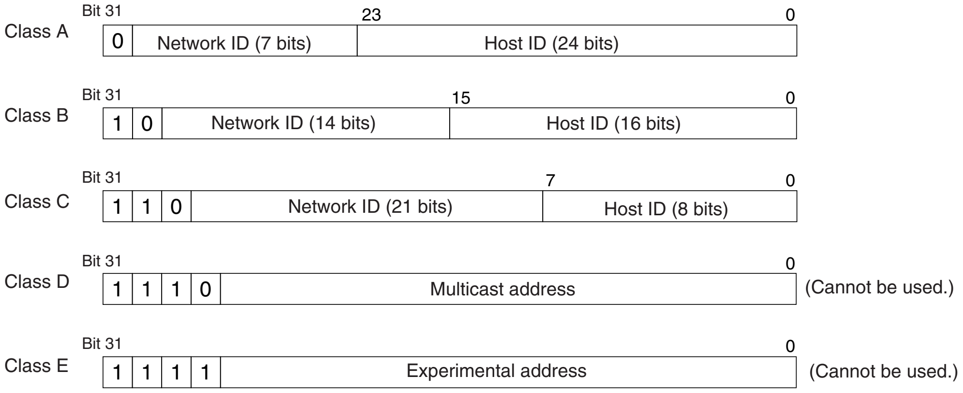
| Bit | Meaning | Status | Manipulated by | Unit operation | Reference |
|---|---|---|---|---|---|
| --- | Target Node PLC Error Flags | TRUE | EtherNet/IP Unit | A major fault, partial fault, or minor fault occurred in the corresponding PLC. | *3-2-3 Device Variables for the CJ-series Unit for Status* |
| | | FALSE | EtherNet/IP Unit | Other than the above. | |

# Determining IP Addresses

# 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.

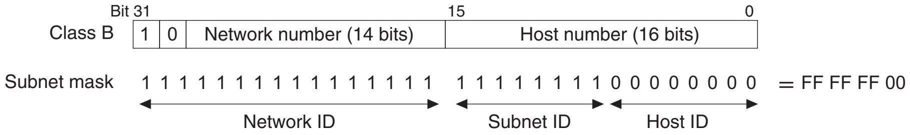
## 4-1-2 Allocating IP Addresses

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 EtherNet/IP Unit 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 Setting the IP Address of the EtherNet/IP Unit

## 4-2-1 Determining IP Addresses

Use one of the following methods to set the IP address of the EtherNet/IP Unit.

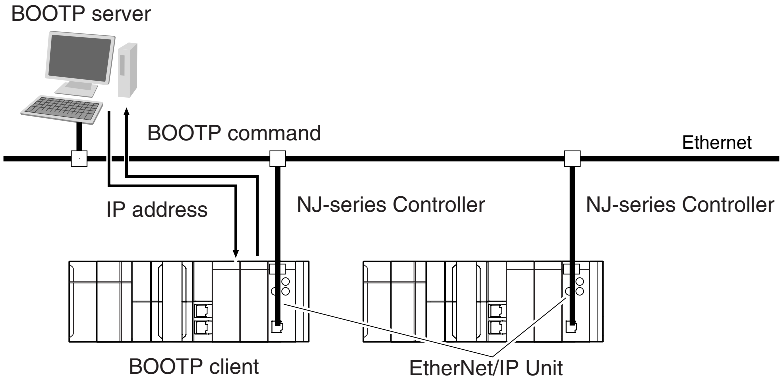
### ● Setting a User-specified IP Address

If you need to change the default IP address of the EtherNet/IP Unit or if you need to use the EtherNet/IP Unit 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.

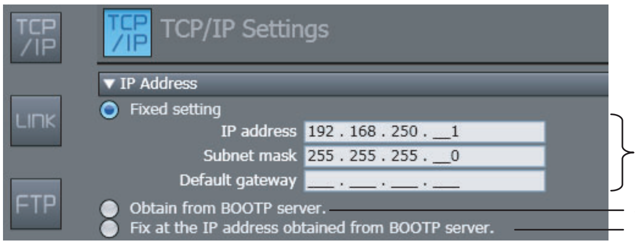


## 4-2-2 Setting IP Addresses

Use the Sysmac Studio to set the IP address of the EtherNet/IP Unit.

**1** Select the setting method for IP addresses.

Set the IP address of the EtherNet/IP Unit in the Special Unit Setup that is registered in the Unit Configuration of the Sysmac Studio.



TCP/IP Settings

▼ IP Address

● Fixed setting

IP address 192 . 168 . 250 . __1

Subnet mask 255 . 255 . 255 . __0

Default gateway ___ . ___ . ___ . ___

○ Obtain from BOOTP server.

○ Fix at the IP address obtained from BOOTP server.

Used to set a user-specified IP address.

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

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

Refer to *5-2 TCP/IP Settings Display* for details.

**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** Restart the EtherNet/IP Unit.

Click the **Restart** Button in the Edit Special Unit Setup Dialog Box.

**6** When the EtherNet/IP Unit restarts, check the status of the 7-segment display on the EtherNet/IP Unit.

If the 7-segment display goes out, goes through the test sequence, and then displays the IP address, the EtherNet/IP Unit has recognized the new TCP/IP settings (the IP address in this case).

**7** IP address is reflected in the EtherNet/IP 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 EtherNet/IP Unit.

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

After the data is downloaded, the IP address that was obtained from the BOOTP server is automatically saved in the EtherNet/IP Unit. Each time the power supply is turned ON, the IP address from the BOOTP server is automatically saved in the EtherNet/IP 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 EtherNet/IP Unit 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 the Controller power supply is turned ON when the IP address was not normally obtained from the BOOTP server, the setting remains at the default setting (i.e., at a fixed setting).
- After the IP address is obtained from the BOOTP server, the EtherNet/IP Unit IP address setting is automatically set to *Fixed setting*.

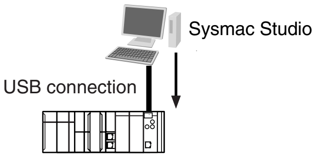
## 4-2-3 Online Connection

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

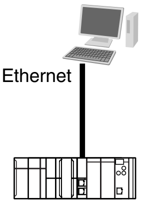
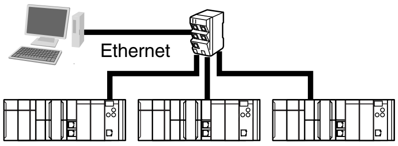
### ● Types of Connections between the CPU Unit and Computer That Runs the Sysmac Studio

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

#### USB Connection

| USB Direct Connection |
|---|
|  |

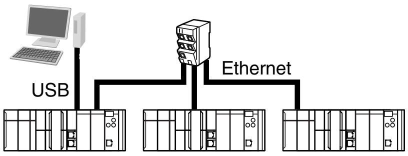
#### Ethernet Connection

| 1:1 Ethernet Direct Connection through AutoIP | 1:N Ethernet Switch Connection |
|---|---|
|  |  |
| **Note 1** An Ethernet switch is not necessarily required.<br>**2** You can use a straight or cross Ethernet cable to connect. | **Note** An Ethernet switch is required to connect. Refer to *2-3-4 Precautions for Ethernet Switch Selection* for details. |

**Precautions for Correct Use**

You cannot place the Sysmac Studio online with an NJ-series CPU Unit if you connect through an EtherNet/IP Unit.

#### Connection from USB Across Ethernet

| 1:N USB Remote Connection |
|---|
|  |
| **Note** An Ethernet switch is required to connect. Refer to *2-3-4 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.

### Precautions for Correct Use

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

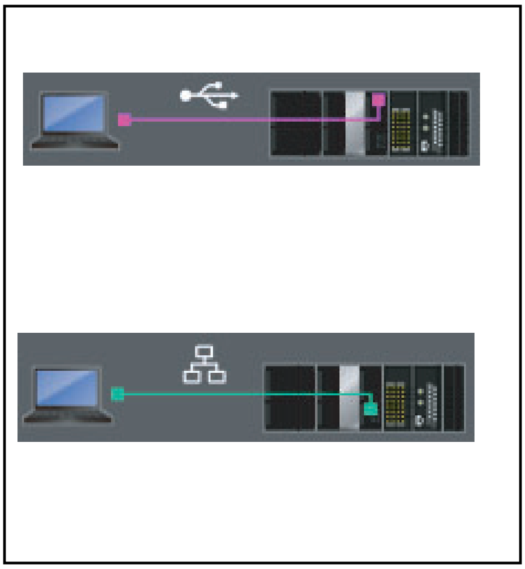
## ● Online Connection Procedure

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.

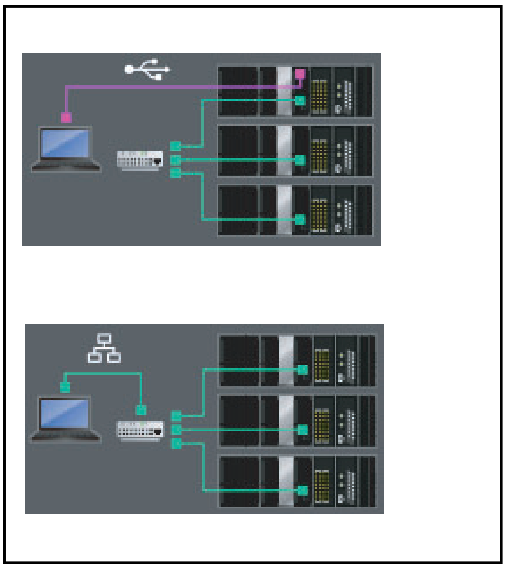
1:1 Connection

Direct Connection



1:N Connection

EtherNet/IP Connection



### Precautions for Correct Use

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

### ● Connecting from a Saved Project

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

### ● Checking the Current IP Address

For the method of reading the local IP address, refer to *3-2-4 Device Variables for the CJ-series Unit for Setup*.

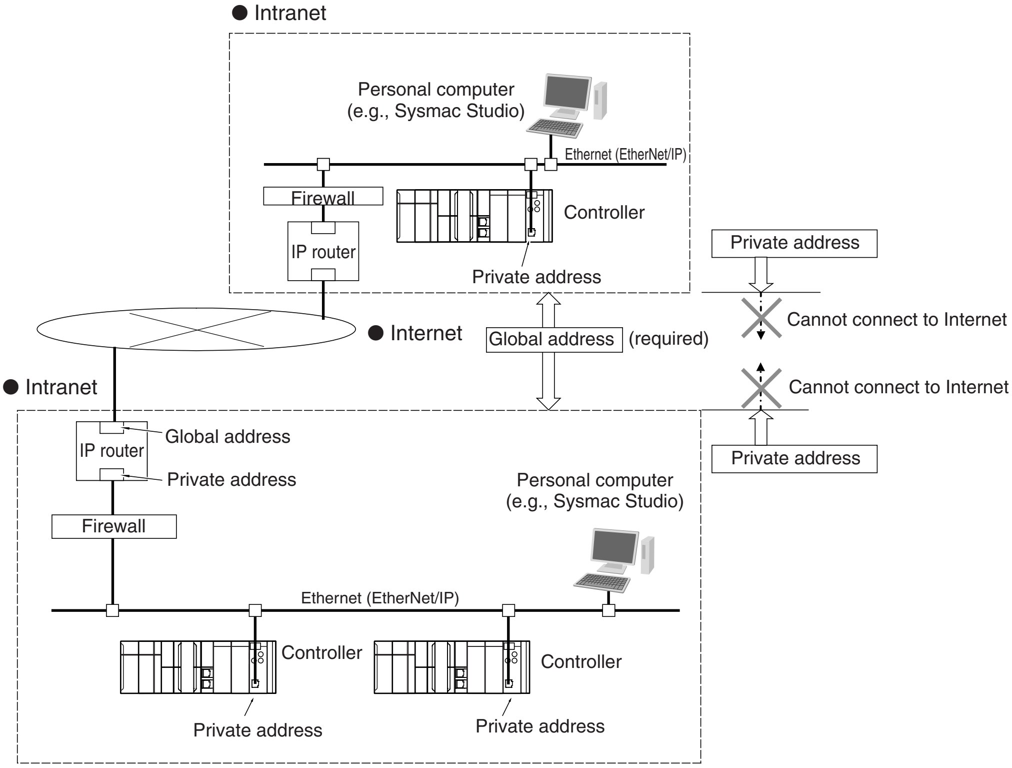
# 4-3 Private and Global Addresses

## 4-3-1 Private and Global Addresses

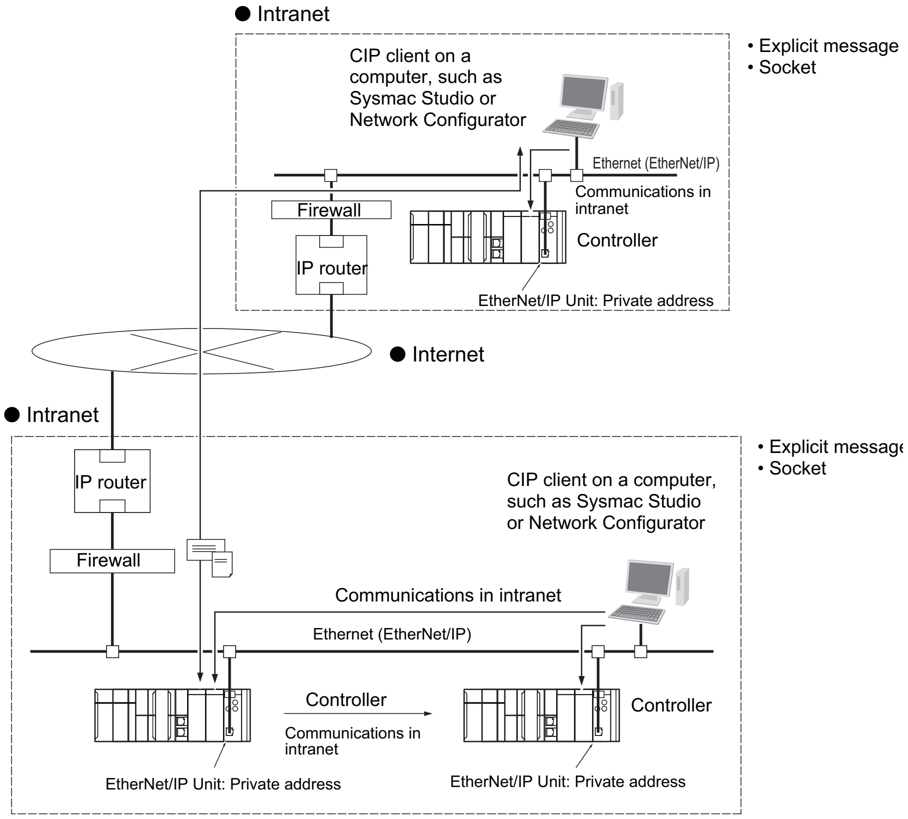
There are two kinds of IP addresses, private and global.

| 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 worldwide. |
| 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 EtherNet/IP Unit, are allocated private addresses.



## 4-3-2 Using a Private Address for the EtherNet/IP Unit


● Intranet
CIP client on a computer, such as Sysmac Studio or Network Configurator
• Explicit message
• Socket
Ethernet (EtherNet/IP)
Communications in intranet
Firewall
Controller
IP router
EtherNet/IP Unit: Private address
● Internet
● Intranet
• Explicit message
• Socket
IP router
CIP client on a computer, such as Sysmac Studio or Network Configurator
Firewall
Communications in intranet
Ethernet (EtherNet/IP)
Controller
Communications in intranet
Controller
EtherNet/IP Unit: Private address
EtherNet/IP Unit: Private address

### ● Conditions for Communications Applications

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

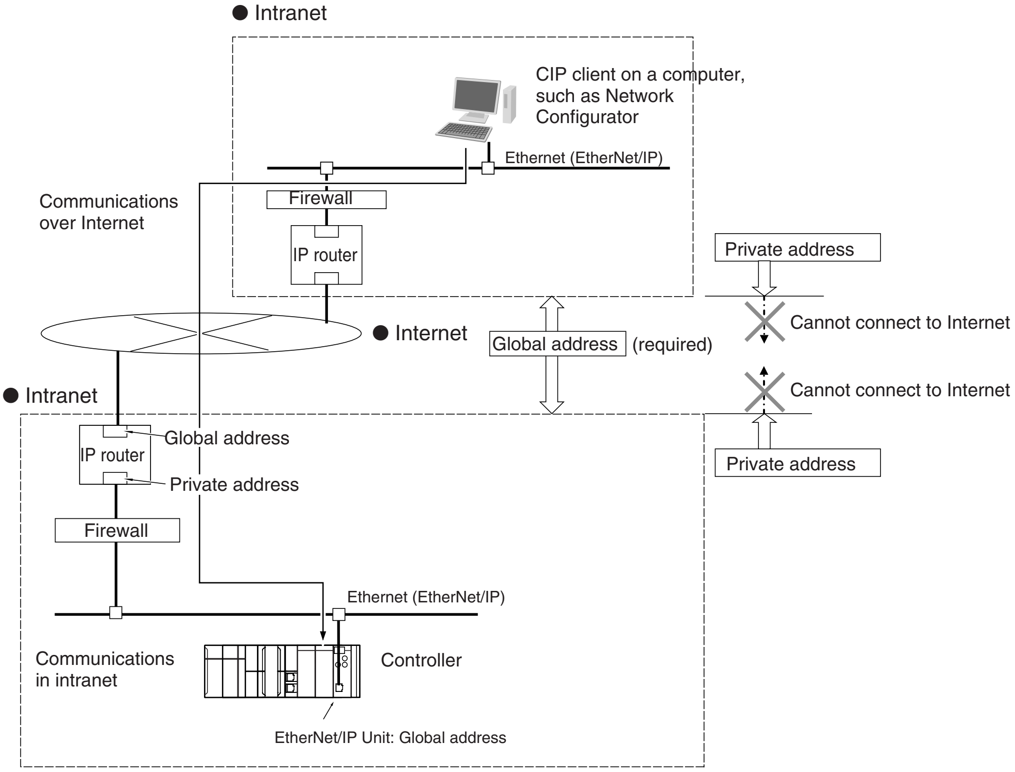
- The explicit message communications service can be executed on the intranet between EtherNet/IP Units with private addresses only.
- A device such as a personal computer (CIP applications including the Network Configurator) cannot connect online and communicate over the Internet with an EtherNet/IP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



### ● 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 an EtherNet/IP Unit 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 an EtherNet/IP Unit 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 EtherNet/IP Unit

# 5-1 Unit Settings for the EtherNet/IP Unit

Set up the EtherNet/IP Unit on the Edit Special Unit Settings Tab Page on the Sysmac Studio.

## 5-1-1 Updating the Unit Settings

To update the Unit settings in the EtherNet/IP Unit, first transfer them from the Sysmac Studio to the CPU Unit and then perform one of the following methods.
As a result, the Unit settings are transferred from the CPU Unit to the EtherNet/IP Unit.

### ● Resetting the Controller

When you reset the Controller, the Unit settings are transferred from the CPU Unit to the EtherNet/IP Unit when the CPU Unit restarts.

### ● Restarting the EtherNet/IP Unit

When you restart the EtherNet/IP Unit with one of the following methods, the Unit settings are transferred from the CPU Unit to the EtherNet/IP Unit.

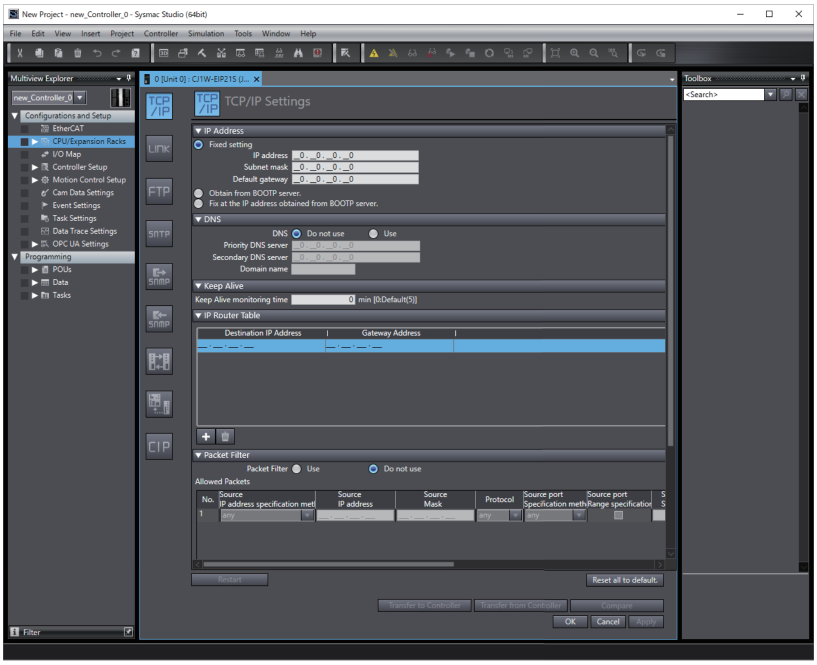
- Click the **Restart** Button in the Edit Special Unit Settings Tab Page.
- Change the Restart Flag (_CJB_CBU**Restart) in the system-defined variables for the EtherNet/IP Unit to TRUE.

## 5-1-2 Restoring the Default Unit Settings

You can restore the Unit settings to their default values.

Click the **Restart** Button in the Edit Special Unit Settings Tab Page, set the restart type for the EtherNet/IP Unit to *Reset all to default*, and then restart the EtherNet/IP Unit.

# 5-2 TCP/IP Settings Display



TCP/IP Settings

▼ IP Address

● Fixed setting

IP address: 0.0.0.0

Subnet mask: 0.0.0.0

Default gateway: 0.0.0.0

○ Obtain from BOOTP server.

○ Fix at the IP address obtained from BOOTP server.

▼ DNS

DNS: ● Do not use ○ Use

Priority DNS server: 0.0.0.0

Secondary DNS server: 0.0.0.0

Domain name:

▼ Keep Alive

Keep Alive monitoring time: 0 min [0:Default(5)]

▼ IP Router Table

| Destination IP Address | Gateway Address |
|---|---|
| _._._._ | _._._._ |

▼ Packet Filter

Packet Filter: ○ Use ● Do not use

Allowed Packets

| No. | Source IP address specification met | Source IP address | Source Mask | Protocol | Source port Specification meth | Source port Range specificatior |
|---|---|---|---|---|---|---|
| 1 | any | _._._._ | _._._._ | any | any | □ |

## ● IP Address

| Setting | Description | Default |
|---|---|---|
| IP address setting method | Select one of the following IP address setting methods for the EtherNet/IP Unit.[1]<br>• Fixed setting<br>• Obtain from BOOTP server<br>• Fix at the IP address obtained from BOOTP server | Fixed setting |
| IP Address [2] | Set the IP address for the EtherNet/IP Unit. | None [3] |
| Subnet mask *2 | Set the subnet mask for the EtherNet/IP Unit. | None |
| Default gateway *4 | Set the IP address of the default gateway for the EtherNet/IP Unit.<br>This setting is not required when a default gateway is not used. | None |

*1 For details on setting IP addresses, refer to *4-2 Setting the IP Address of the EtherNet/IP Unit*.

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

*3 The node address that is set on the rotary switches on the front panel of the EtherNet/IP Unit is set as the default.

*4 This setting is valid if you set IP address setting method to *Fixed setting*.

## ● DNS

| Setting | Description | Default |
|---|---|---|
| DNS | Enable using DNS if a DNS is used to resolve host names when host names are specified for the remote communications nodes in CIP communications and socket instructions.<br>A DNS server is required to use DNS. | Do not use. |
| Priority DNS server [*1*2] | Set the IP address of the DNS server.<br>You can set priority and secondary IP addresses. | None |
| Secondary DNS server | | None |
| Domain name[*1] | Sets the name of the domain to which the EtherNet/IP Unit belongs.<br>The EtherNet/IP Unit does not use the domain name in actual communications.<br>(Single-byte alphanumeric characters, dots, and hyphens: 48 characters max.) | None |

*1 These settings are required when DNS is used.

*2 Due to Ethernet restrictions, you cannot specify the IP addresses that are described below.

- IP addresses that start with 127, 0, or 255 (decimal)
- Class-D IP addresses (224.0.0.0 to 239.255.255.255)
- Class-E IP addresses (240.0.0.0 to 255.255.255.255)

## ● Keep Alive (CJ1W-EIP21S Only)

| Setting | Description | Default |
|---|---|---|
| Keep Alive monitoring time | Set the keep-alive monitoring time.<br>When using socket services with FINS/TCP or TCP/IP, if the remote node (server or client) remains idle (not responding) for more than the time set here, the EtherNet/IP Unit disconnects the connection (only for socket services with FINS/TCP or TCP/IP).<br>Setting range: 0 to 65,535 minutes (The default is 0, meaning that the checking time is 5 minutes.)<br>This setting is common to keep-alive setting (whether or not the function is enabled for the remote node) for each connection number in the *FINS/TCP* category. | 0 (Default: 5 minutes) |

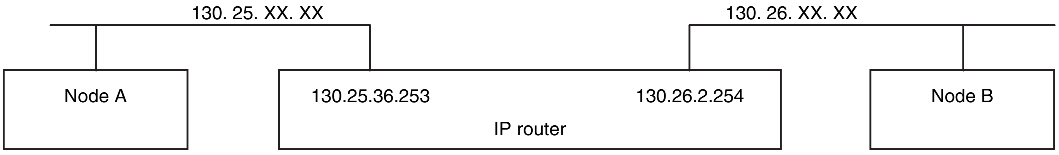
## ● IP Router Table

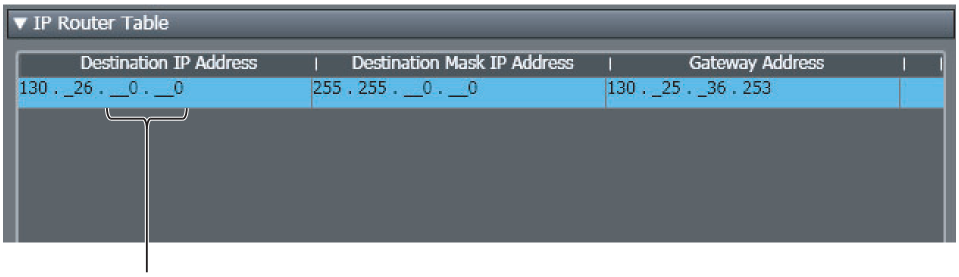
| Setting | Description | Default |
|---|---|---|
| Destination IP address | Set these settings when the EtherNet/IP Unit communicates with nodes on other IP network segments via an IP router.<br>You can set up to eight combinations of IP addresses and gateway addresses.<br>Specify 0 for the host portions of the IP addresses.<br>Refer to the following section for a setting example for the IP router table. | None |
| Gateway address | | None |

### Additional Information

#### IP Router Table Setting Example

Set the following IP router table in node A to use tag data links or CIP message communications between node A and node B through the IP router. When you set the IP router table, node A sends packets to the gateway IP address (130.25.36.253) if communications instructions are executed on node A and addressed to node B.




▼ IP Router Table

| Destination IP Address | Destination Mask IP Address | Gateway Address |
|---|---|---|
| 130 . 26 . 0 . 0 | 255 . 255 . 0 . 0 | 130 . 25 . 36 . 253 |

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

## ● Packet Filter (CJ1W-EIP21S Only)

| Setting | Description | Default |
|---|---|---|
| Packet Filter | Select whether or not to use the packet filter. | Do not use |
| Allowed Packets | Set the packets allowed to enter.<br>Up to 32 entries can be set. | 1 entry |
| Source IP address specification method | Select how to specify the source IP address.<br>• any: Allow all IP addresses.<br>• IP address specification: Allow the specified IP address. | IP address specification |
| Source IP address | When the **Source IP address specification method** is *IP Address specification*, set the IP address.<br>Set the bits set for the mask in *Source Mask* and the following bits to 0s. | None |
| Source Mask | When the **Source IP address specification method** is *IP Address setting*, set the mask. | None |
| Protocol | Select the protocol that you want to allow.<br>• any: Allow all protocols.<br>• TCP: Allow TCP.<br>• UDP: Allow UDP. | any |
| Source port Specification method | When the **Protocol** is *TCP* or *UDP*, select how to specify the source port you want to allow.<br>• any: Allow all ports.<br>• Port specification: Allow the specified port. | any |
| Source port Range specification | Select whether to enable range specification for the source port.<br>• Not selected: Disable range specification.<br>• Selected: Enable range specification. | Not selected |
| Source port Start number | When the **Source port Specification method** is *Port specification*, set the start number of the source port.<br>This can be specified in the range of 1 to 65535. | None |
| Source port End number | When the **Source port Specification method** is *Port specification* and the **Source port Range specification** Check Box is selected, set the end number of the destination port.<br>This can be specified in the range of 1 to 65535. | None |

| Setting | Description | Default |
|---|---|---|
| Destination port Specification method | When the **Protocol** is *TCP* or *UDP*, select how to specify the destination port you want to allow.<br>• any: Allow all ports.<br>• Port specification: Allow the specified port. | any |
| Destination port Range specification | When the **Destination port Specification method** is *Port specification*, select whether to enable range specification for the destination port.<br>• Not selected: Disable range specification.<br>• Selected: Enable range specification. | Not selected |
| Destination port Start number | When the **Destination port Specification method** is *Port specification*, set the start number of the destination port.<br>This can be specified in the range of 1 to 65535. | None |
| Destination port End number | When the **Destination port Specification method** is *Port specification* and the **Destination port Range specification** Check Box is selected, set the end number of the destination port.<br>This can be specified in the range of 1 to 65535. | None |

A. How the Packet Filter Works When Filtering Packets by IP Address

The packet filter settings allow packets reception at the set IP address(es).

If *Any* is set in **Source IP address specification method**, packet reception is allowed for all IP addresses.

If *IP address specification* is set in **Source IP address specification method**, packet reception is allowed for the IP address specified in the lower-level settings: **Source IP address** and **Source Mask**.

For **Source IP address**, set the bits set for the mask in **Source Mask** and the following bits to 0s.

Example of setting

- To allow packet reception for one IP address
  Set the IP address at which to allow reception in **Source IP address** and 255.255.255.255 in **Source Mask**.
  For example, to allow reception at 192.168.250.100, set *192.168.250.100* in **Source IP address** and 255.255.255.255 in **Source Mask**.
- To allow packet reception for multiple IP addresses
  Set the IP address at which to allow reception in **Source IP address** and **Source Mask**.
  For example, to allow reception at 192.168.***.***, set 192.168.0.0 in **Source IP address** and 255.255.0.0 in **Source Mask**.

B. How the Packet Filter Works When Filtering Packets by Protocol

The packet filter settings allow packet reception with the set protocol and port.

If *any* is set in **Protocol**, reception of IGMP and ICMP packets is allowed for all TCP/UDP ports.

If *TCP* or *UDP* is set in **Protocol**, and *any* is set in **Source port Specification method**, reception of all TCP or UDP packets is allowed, regardless of the port number settings.

If *TCP* or *UDP* is set in **Protocol**, and *Port specification* is set in **Source port Specification method**, packet reception is allowed for all TCP or UDP port numbers in the specified range.

Example of setting

- To allow packet reception for one TCP/UDP port number
  Clear the **Source port Range specification** Check Box, and set the TCP/UDP port numbers for which to allow packet reception for **Source port Start number**.

- To allow packet reception for multiple TCP/UDP port numbers
  Select the **Source port Range specification** Check Box, and set a range of TCP/UDP port numbers for which to allow packet reception for **Source port Start number** and **Source port End number**.
  Packet reception is allowed for the range of port numbers determined by the **Source port Start number** and **Source port End number** settings.

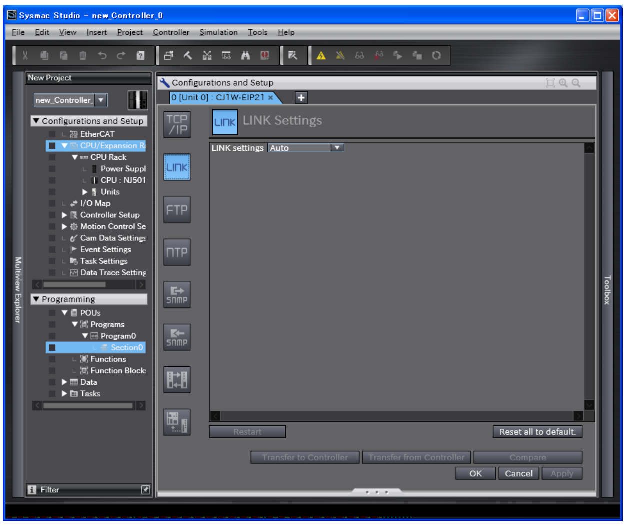
### ● Broadcasting (CJ1W-EIP21S Only)

| Setting | Description | Default |
|---|---|---|
| All Broadcast | Set how to specify the IP address for broadcasting using FINS/UDP.<br>• All 1 (4.3BSD): Broadcasting is performed with all bits of the host ID set to 1.<br>• All 0 (4.2BSD): Broadcasting is performed with all bits of the host ID set to 0.<br>Normally use *All 1 (4.3BSD)*, which is the default . | All 1 (4.3BSD) |

### ● Socket Services (CJ1W-EIP21S Only)

| Setting | Description | Default |
|---|---|---|
| Speed Up<br>(High-speed Socket Service) | Selecting the *Speed Up* Option improves the performance of processing for sending and receiving by manipulating variables for socket services. Refer to *13-8-5 Times Required for Sending and Receiving for Socket Services* for communications performance. When this option is checked, executing the SendCmd instruction will result in a socket service error. | Not selected |

# 5-3 Link Settings Display

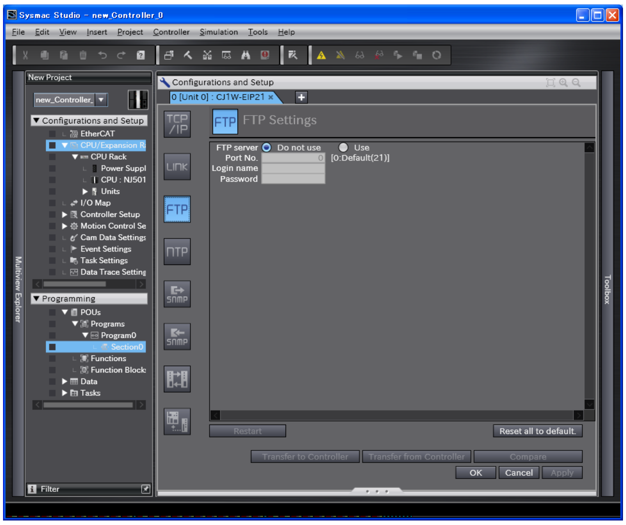


LINK Settings

LINK settings: Auto

| Setting | Description | Default |
|---|---|---|
| LINK settings | Set the baud rate for the EtherNet/IP Unit.<br>• Auto<br>• 10 Mbps Half Duplex<br>• 10 Mbps Full Duplex<br>• 100 Mbps Half Duplex<br>• 100 Mbps Full Duplex | Auto |

# 5-4 FTP Settings Display



FTP Settings

FTP server ● Do not use ○ Use

Port No. 0 [0:Default(21)]

Login name

Password

| Setting | Description | Default |
|---|---|---|
| FTP server | Specify whether to use the FTP server.<br>FTP connections from external devices will not be possible if the *Do not use* Option is selected. | Do not use. |
| Port No. *1*2 | Set the FTP port number of the EtherNet/IP Unit. It is normally not necessary to change this setting.<br>The port number that is one less than the specified port number is used for data transfer. | 0 (port No. 21) |
| Login name *1 | Set the login name to externally connect to the EtherNet/ IP Unit via FTP.<br>CJ1W-EIP21: 1 to 12 single-byte alphanumeric characters*3<br>CJ1W-EIP21S: 1 to 16 single-byte alphanumeric characters*3 | None |
| Password *1 | Set the password to externally connect to the EtherNet/IP Unit via FTP.<br>CJ1W-EIP21: 1 to 8 single-byte alphanumeric characters*3<br>CJ1W-EIP21S: 8 to 16 single-byte alphanumeric characters*3 | None |

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

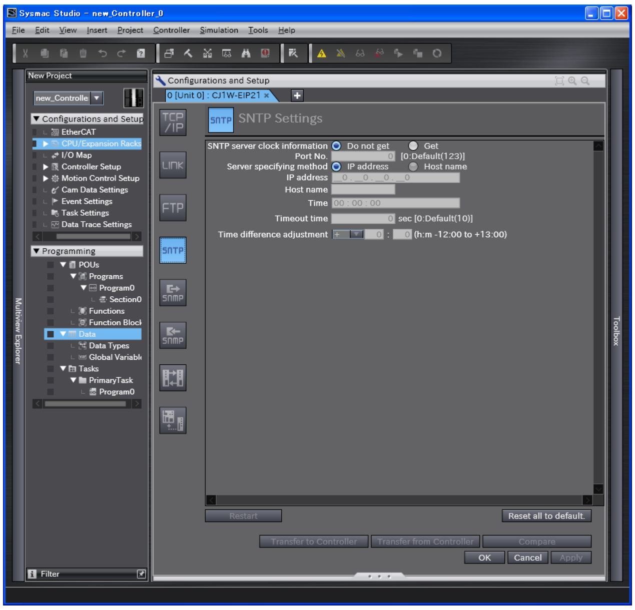
*2 The following ports are used by the system and cannot be set by the user: 9600 and 44818.

*3 The login name and password are case sensitive.

### Additional Information

Refer to *Section 9 FTP Server* for details on the FTP server.

# 5-5 SNTP Settings Display



SNTP Settings

SNTP server clock information ● Do not get ○ Get

Port No. 0 [0:Default(123)]

Server specifying method ● IP address ○ Host name

IP address 0 . 0 . 0 . 0

Host name

Time 00 : 00 : 00

Timeout time 0 sec [0:Default(10)]

Time difference adjustment + 0 : 0 (h:m -12:00 to +13:00)

| Setting | Description | Default |
|---|---|---|
| SNTP server clock information | Specify whether to get the clock information from the SNTP 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 SNTP server to obtain clock information. It is normally not necessary to change this setting. | 0 (port No. 123) |
| Server specifying method *1 | Set the method to use to specify the SNTP server to obtain clock information.<br>• IP address<br>• Host name | IP address |
| IP address *3 | Set the IP address of the SNTP server.<br>(Set this setting if the server specifying method is set to the *IP address* Option.) | None |
| Host name | Set the host name of the SNTP server (i.e., the domain name of the host).<br>(Set this setting if server specifying method is set to the *Host name* Option.)<br>(You can use up to 200 single-byte alphanumeric characters, dots, and hyphens with up to 63 single-byte alphanumeric characters between dots.) | None |
| Time [hours:minutes:seconds] | Set the time at which the SNTP server is accessed to synchronize the clock.<br>(Setting range: 00:00:00 to 23:59:59) | 00:00:00 |
| Timeout time (seconds) *1 | Set the timeout detection time.<br>(Setting range: 1 to 255 seconds)<br>If the remote host does not respond, retry processing is performed four times within the time interval that is set here. | 0 (10 s) |
| Time difference adjustment | Set the time to offset the clock in the CPU Unit when setting the clock in the CPU Unit to the time obtained from the SNTP server. To use the time from the SNTP server as is, enter 0 for the time difference adjustment. | +0: 0 (h: m) |

*1 These settings are required to get the clock information from the SNTP server.

*2 The following ports are used by the system and cannot be set by the user: 53, 68, 161, 162, 2222, 9600, and 44818.

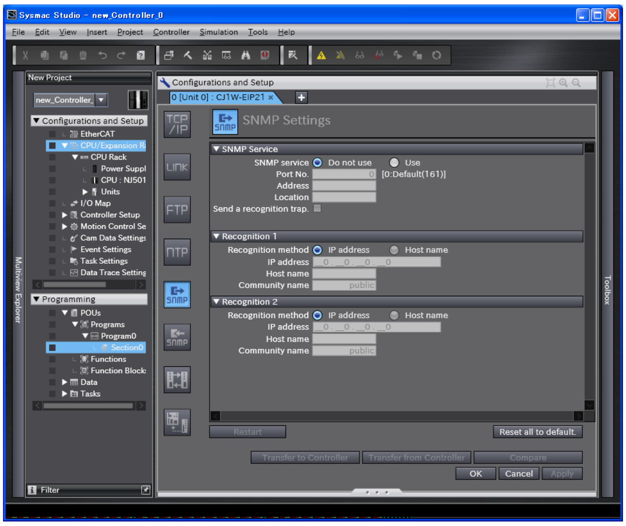
*3 Due to Ethernet restrictions, you cannot specify the IP addresses that are described below.

- IP addresses that start with 127, 0, or 255 (decimal)
- Class-D IP addresses (224.0.0.0 to 239.255.255.255)
- Class-E IP addresses (240.0.0.0 to 255.255.255.255)

**Additional Information**

Refer to *Section 10 Automatic Clock Adjustment* for details on obtaining clock information from the SNTP server.

# 5-6 SNMP Settings Display



SNMP Settings

▼ SNMP Service

SNMP service ● Do not use ○ Use

Port No. 0 [0:Default(161)]

Address

Location

Send a recognition trap. □

▼ Recognition 1

Recognition method ● IP address ○ Host name

IP address 0 . 0 . 0 . 0

Host name

Community name public

▼ Recognition 2

Recognition method ● IP address ○ Host name

IP address 0 . 0 . 0 . 0

Host name

Community name public

## ● SNMP Service

| Setting | | Description | Default |
|---|---|---|---|
| SNMP Service | | Specify whether to use the SNMP monitor service.<br>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 SNMP server that is used to connect from an SNMP manager.<br>It is normally not necessary to change this setting. | 0 (port No. 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 200 alphanumeric characters for each item.) | None |
| | Location | | None |
| | Send a recognition trap | Set whether to send an authentication trap.<br>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.<br>If you select *Send a recognition trap*, specify the SNMP trap settings on the SNMP Trap Tab Page. | Not selected. (Not used.) |

* The following ports are used by the system and cannot be set by the user: 53, 68, 123, 162, 2222, 9600, and 44818.

### Additional Information

Refer to *Section 11 SNMP Agent* for details on the SNMP service.

### ● Recognition 1 *1

| Setting | Description | Default |
|---|---|---|
| Recognition method | Set the method to use to specify SNMP managers for which access is permitted.<br>• IP address<br>• Host name<br>Make these settings to permit access by only certain SNMP managers.<br>Access is not allowed unless an IP address or host name is set. | IP address |
| IP address *2 | Set the IP address of the SNMP manager.<br>If the default setting of 0.0.0.0 is used, access is permitted from all SNMP managers.<br>(Set this setting if the recognition method in the recognition 1 settings is set to the *IP address* Option.) | 0.0.0.0 |
| Host name | Set the host name of the SNMP manager.<br>(Set this setting if the recognition method in the recognition 1 settings is set to the *Host name* Option.)<br>(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 EtherNet/IP Unit.<br>(Single-byte alphanumeric characters, dots, and hyphens: 200 characters max.) | public |

### ● Recognition 2 *2

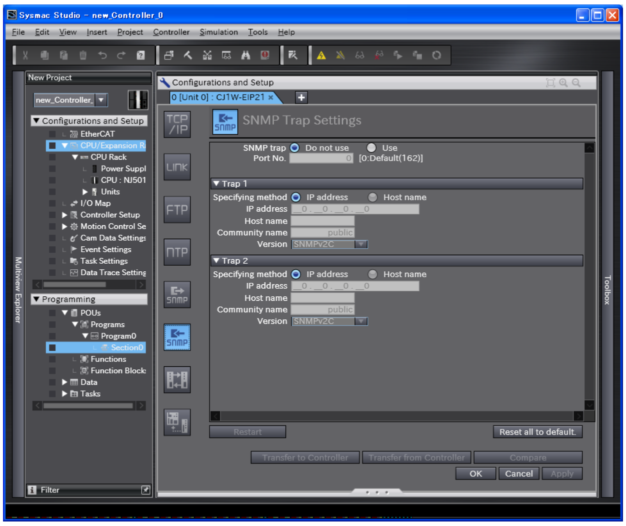
| Setting | Description | Default |
|---|---|---|
| Recognition method | Set the method to use to specify SNMP managers for which access is permitted.<br>• IP address<br>• Host name<br>Make these settings to permit access by only certain SNMP managers.<br>Access is not allowed unless an IP address or host name is set. | IP address |
| IP address *2 | Set the IP address of the SNMP manager.<br>If the default setting of 0.0.0.0 is used, access is permitted from all SNMP managers.<br>(Set this setting if the recognition method in the recognition 2 settings is set to the *IP address* Option.) | 0.0.0.0 |
| Host name | Set the host name of the SNMP manager.<br>(Set this setting if the recognition method in the recognition 2 settings is set to the *Host name* Option.)<br>(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 EtherNet/IP Unit.<br>(Single-byte alphanumeric characters, dots, and hyphens: 200 characters max.) | public |

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

*2 Due to Ethernet restrictions, you cannot specify the IP addresses that are described below.

- IP addresses that start with 127, 0, or 255 (decimal)
- Class-D IP addresses (224.0.0.0 to 239.255.255.255)
- Class-E IP addresses (240.0.0.0 to 255.255.255.255)

# 5-7 SNMP Trap Settings Display



SNMP Trap Settings

SNMP trap ● Do not use ○ Use

Port No. 0 [0:Default(162)]

▼ Trap 1

Specifying method ● IP address ○ Host name

IP address 0 . 0 . 0 . 0

Host name

Community name public

Version SNMPv2C

▼ Trap 2

Specifying method ● IP address ○ Host name

IP address 0 . 0 . 0 . 0

Host name

Community name public

Version SNMPv2C

## ● SNMP Trap

| Setting | Description | Default |
|---|---|---|
| SNMP trap | Specify whether to use the SNMP trap (network error detection).<br>If the SNMP trap service is not enabled, SNMP traps are not sent to the SNMP manager. | Do not use. |
| Port No. *1 | Set the port number to use to connect to the SNMP server.<br>It is normally not necessary to change this setting. | 0 (port No. 162) |

*1 The following ports are used by the system and cannot be set by the user: 53, 68, 123, 161, 2222, 9600, and 44818.

### Additional Information

Refer to *11-1-1 Overview* for details on the SNMP trap.

### ● Trap 1 *1

| Setting | Description | Default |
|---|---|---|
| Specifying method | Set the specifying method for the SNMP manager destination for SNMP traps.<br>• IP address<br>• Host name | IP address |
| IP address *2 | Set the IP address of the SNMP manager.<br>(Set this setting if the specifying method in the trap 1 settings is set to the *IP address* Option.) | 0.0.0.0 |
| Host name | Set the host name of the SNMP manager.<br>(Set this setting if the specifying method in the trap 1 settings is set to the *Host name* Option.)<br>(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.<br>(You can use up to 200 alphanumeric characters.) | public |
| Version | Set the version of the SNMP manager.<br>• SNMPv1<br>• SNMPv2C | SNMPv2C |

### ● Trap 2 *1

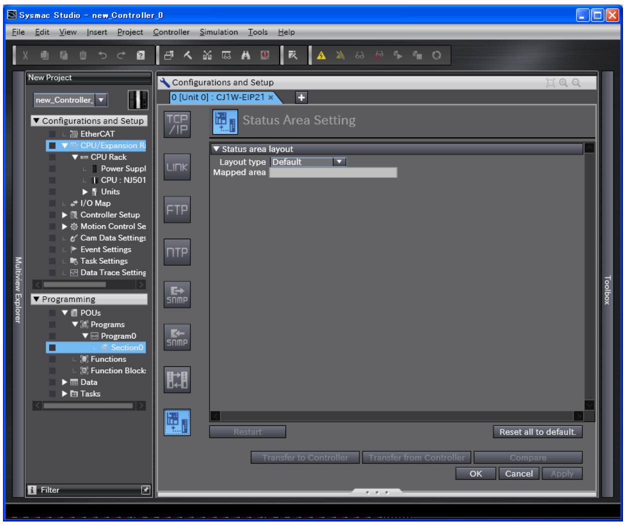
| Setting | Description | Default |
|---|---|---|
| Specifying method | Set the specifying method for the SNMP manager destination for SNMP traps.<br>• IP address<br>• Host name | IP address |
| IP address *2 | Set the IP address of the SNMP manager.<br>(Set this setting if the specifying method in the trap 2 settings is set to the *IP address* Option.) | 0.0.0.0 |
| Host name | Set the host name of the SNMP manager.<br>(Set this setting if the specifying method in the trap 2 settings is set to the *Host name* Option.)<br>(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.<br>(You can use up to 200 alphanumeric characters.) | public |
| Version | Set the version of the SNMP manager.<br>• SNMPv1<br>• SNMPv2C | SNMPv2C |

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

*2 Due to Ethernet restrictions, you cannot specify the IP addresses that are described below.

- IP addresses that start with 127, 0, or 255 (decimal)
- Class-D IP addresses (224.0.0.0 to 239.255.255.255)
- Class-E IP addresses (240.0.0.0 to 255.255.255.255)

# 5-8 Status Area Settings Display



Status area layout

Layout type: Default

Mapped area

## ● Status Area Layout

| Setting | Description | Default |
|---|---|---|
| Layout Type | Select the variables in which to store the status information on the target nodes that are connected to the EtherNet/IP Unit when the EtherNet/IP Unit is the originator.<br>• Default<br>The following device variables for the CJ-series Unit are used.<br>• *_*TargetPLCMdSta* (Target Node PLC Operating Flags)<br>• *_*TargetPLCErrSta* (Target Node PLC Error Flags)<br>• *_*RegTargetSta* (Registered Target Node Table)<br>• *_*EstbTargetSta* (Normal Target Node Table)<br>• User Setting<br>Select this setting to store the status information in a user-defined variable. | Default |
| Mapped area | If the *Layout Type* is set to *User Definition*, enter the name of the user-defined variable in which to store the status information on the target nodes.<br>Refer to *3-3-2 Setting User Definitions* for the setting procedure for user-defined variables. | None |

# 6

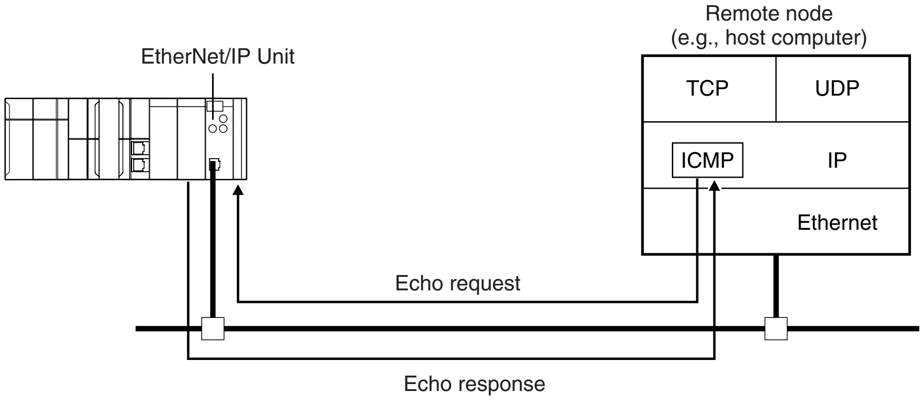
# Testing Communications

# 6-1 Testing Communications

If the basic settings (in particular the IP address and subnet mask) have been made correctly for the EtherNet/IP Unit, 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 EtherNet/IP Unit.

## 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 EtherNet/IP Unit 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 with the EtherNet/IP Unit

The EtherNet/IP Unit 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 an EtherNet/IP Unit. 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

## 7-1-1 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.

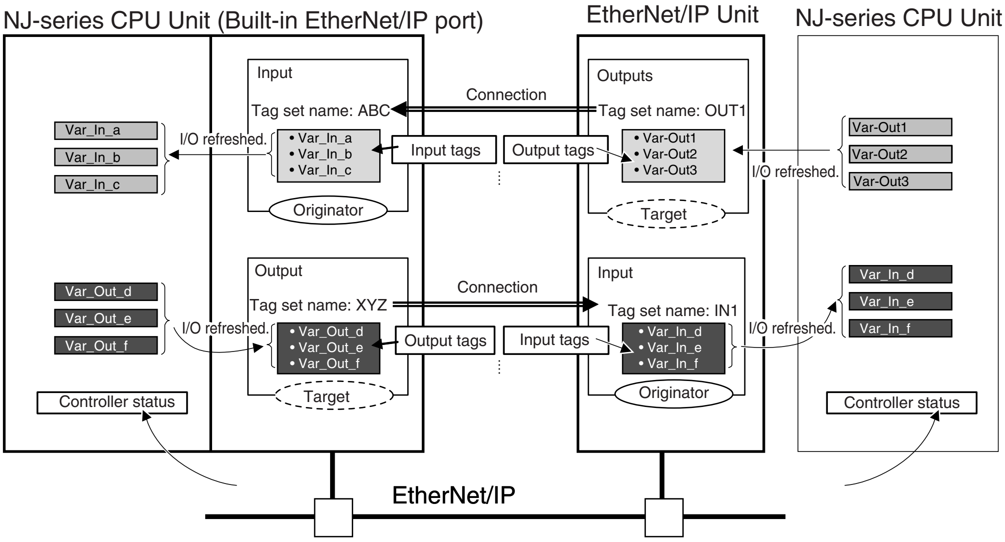
### Precautions for Correct Use

If the *Do not use* Option is selected for **CIP Message Server**, tag data links cannot be used. Select the *Use* Option for **CIP Message Server**. Refer to *CIP Message Server* in *12-2 Opening and Closing the Port* for information on the CIP message server settings.

### Additional Information

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

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



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

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

## 7-1-2 Data Link Data Areas

### Tags

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

**Precautions for Correct Use**

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

### Tag Sets

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

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

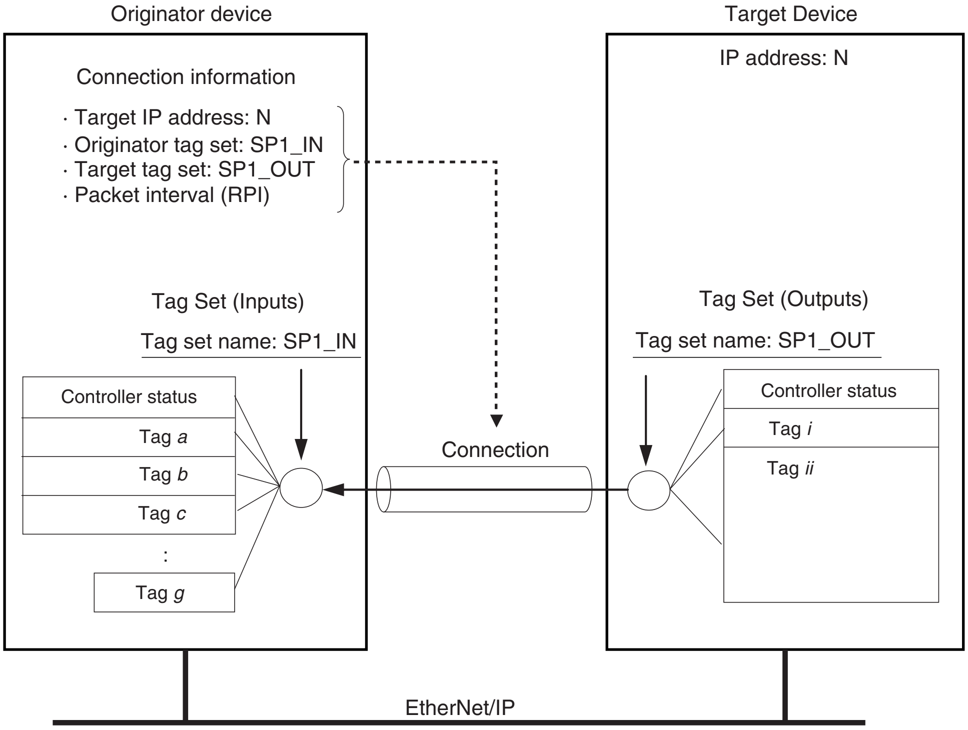
**Note** A connection is used to exchange data as a unit within which data concurrency is maintained. Thus, data concurrency is maintained for all the data exchanged for the tags in one data set.

**Precautions for Correct Use**

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

## ● Example

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



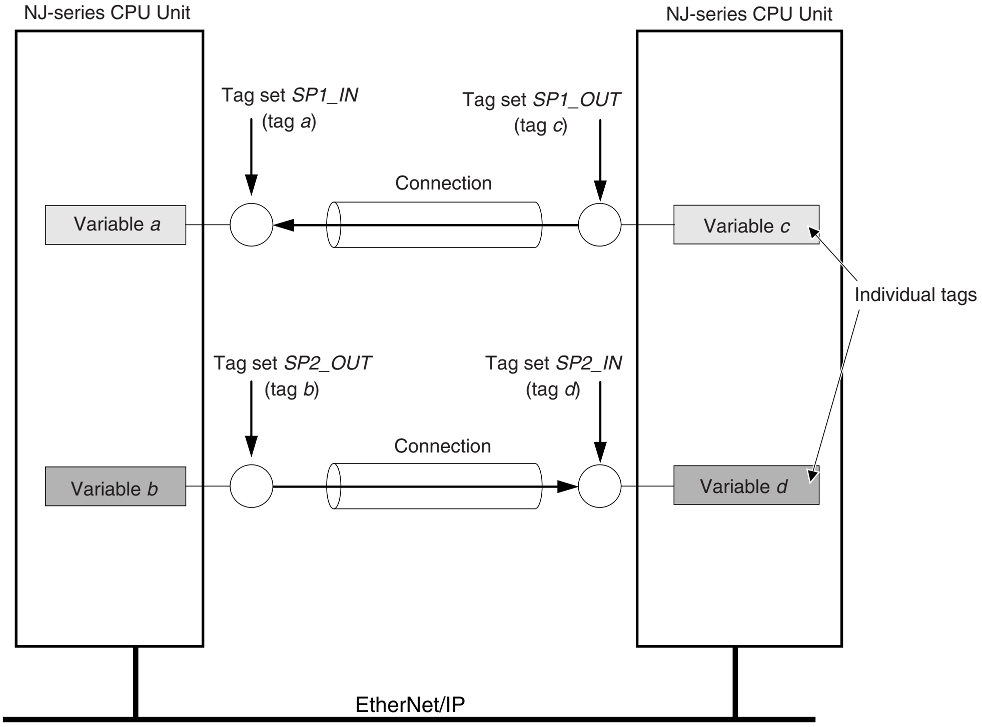
There are both input (consume) and output (produce) tag sets. Each tag set can contain only input tags or only output tags. The same input tag cannot be included in more than one input tag set.

## ● Number of Tags in Tag Sets

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

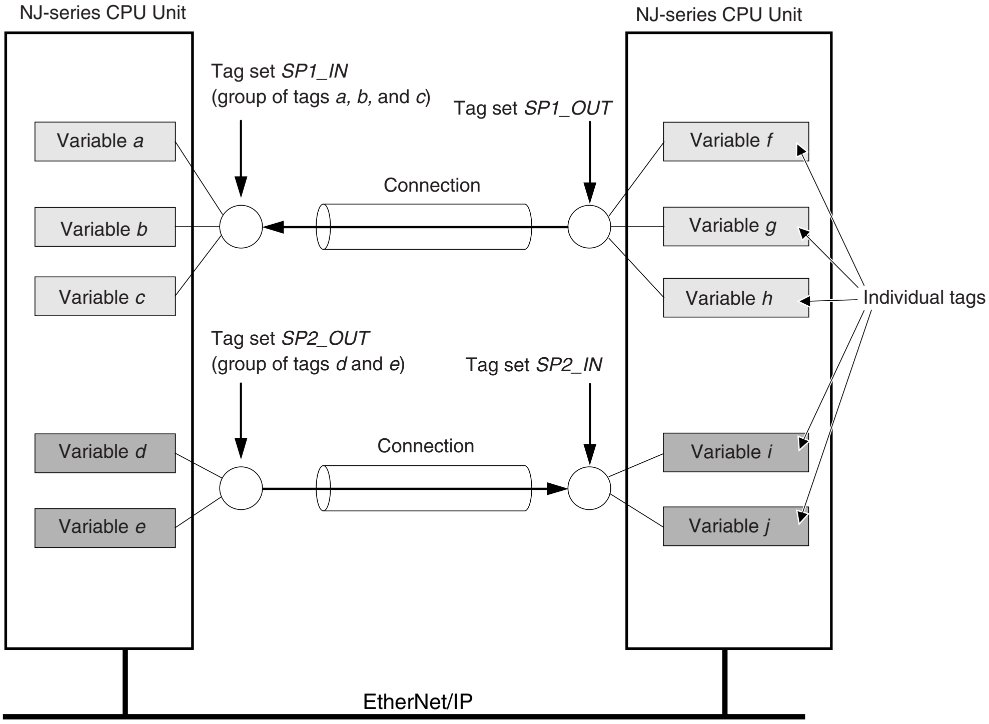
### Tag Sets with Only One Tag Each

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



### Tag Sets with Multiple Tags Each

As shown below, tags can be grouped. You can place up to eight tags (with a total of up to 722 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.<br>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.<br>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]<br>• Global variables |
| | Maximum number of words per tag | 722 words (1,444 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 | 722 words (1,444 bytes) |
| | Maximum number of tag sets | 256 |
| Connections | | Maximum number of connections per Unit: 256 |
| Connection type | | Each connection can be set for 1-to-1 (unicast) or 1-to-N (multi-cast) communications. |
| Packet interval (RPI) | | 0.5 to 10,000 ms in 0.5-ms increments<br>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 |
|---|---|---|---|
| **Variables with basic data types** | | aaa | Supported. |
| **Enumerated variables** | | bbb | Supported. |
| **Array variables** | **Arrays** | bbb | Supported. |
| | **Elements** | ccc[2] | Supported. |
| **Structure variables** | **Structures** | ddd | Supported. |
| | **Member** | ddd.xxx | Supported. |
| **Union variables** | **Unions** | eee | Not supported. |
| | **Member** | eee.yyy | Supported. |

## 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.*

**Note** 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.

Refer to *7-1-7 Concurrency of Tag Data Link Data* for details on the concurrency of tag data link data.

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

**Precautions for Correct Use**

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

1) Variable names that contain only single-byte numerals (Example: 001)

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

## Setting and Downloading Tag Data Link Parameters (Network Configurator)

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 EtherNet/IP Unit, 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 an EtherNet/IP Unit on an NJ-series CPU Unit. (There is a maximum data size of 1,444 bytes (722 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 256 tag sets for an EtherNet/IP Unit on an NJ-series CPU Unit. (The maximum data size of 1 tag set is 1,444 bytes (722 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 EtherNet/IP Unit.

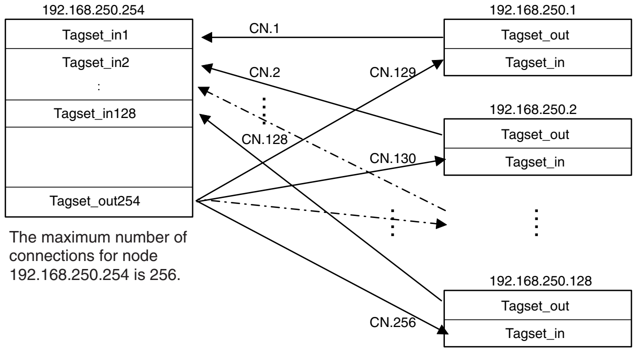
## Additional Information

### 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 EtherNet/IP Unit with IP address of 192.168.250.254 in bidirectional connection with 128 nodes

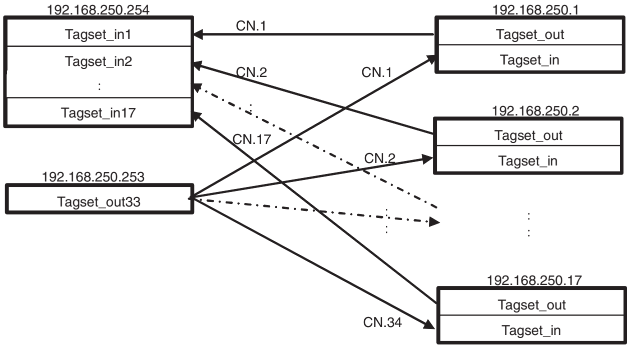


An EtherNet/IP Unit must be mounted to the Controller to increase the maximum number of connections. (Refer to the following figure.)

Example of Calculating the Number of Connections

The maximum number of connections (32) per port would be exceeded if a built-in EtherNet/IP port with an IP address of 192.168.250.254 is used in a bidirectional connection with 17 nodes.

In this case, bidirectional communications can be performed with 17 nodes or more by adding an EtherNet/IP Unit with the IP address of, for example, 192.168.250.253 to the same Controller, creating an output tag set in the new EtherNet/IP Unit, and creating connections.



## Version Information

- You can use the CJ1W-EIP21 EtherNet/IP Unit mounted to an NJ-series Controller with a CPU Unit with unit version 1.01 or later and Sysmac Studio version 1.02 or higher.
- You can use the CJ1W-EIP21S EtherNet/IP Unit mounted to an NJ-series Controller with a CPU Unit with unit version 1.67 or later and Sysmac Studio version 1.60 or higher.

### Setting the Requested Packet Interval (RPI)

The RPI is the I/O data refresh cycle on the Ethernet line when performing tag data links. The RPI can be set separately for each connection. You can set the EtherNet/IP Unit to between 0.5 and 10,000 ms (in 0.5-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 Connection Type

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 an Ethernet switch that has multi-cast filtering, unless the tag set is received by all nodes in the network. If an Ethernet switch without multi-cast filtering is used, the multi-cast packets are broadcast to the entire network, and so packets are sent to nodes that do not require them, which will cause the communications load on those nodes to increase. 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**

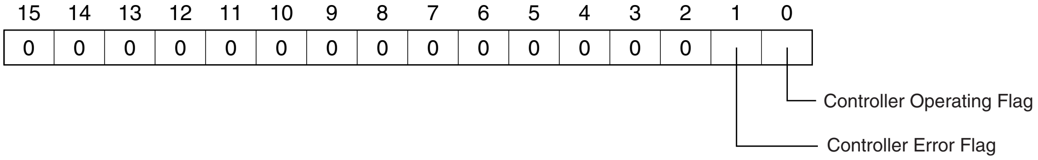
The performance of communications devices is limited to some extent by the limitations of each product’s specifications. Consequently, there are limits to the packet interval (RPI) settings. Refer to *14-2 Adjusting the Communications Load* and set an appropriate packet interval (RPI).

## 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 the device variables for the CJ-series Unit to start and stop the entire network. Refer to *7-2-12 Starting and Stopping Tag Data Links* for details.

## 7-1-6 Controller Status

You can include the Controller status as a member of a tag set in the data sent and received.
The Controller status consists of flags that show the operating status of the CPU Unit to which the EtherNet/IP Unit is mounted. It includes operating information and error information. 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.)

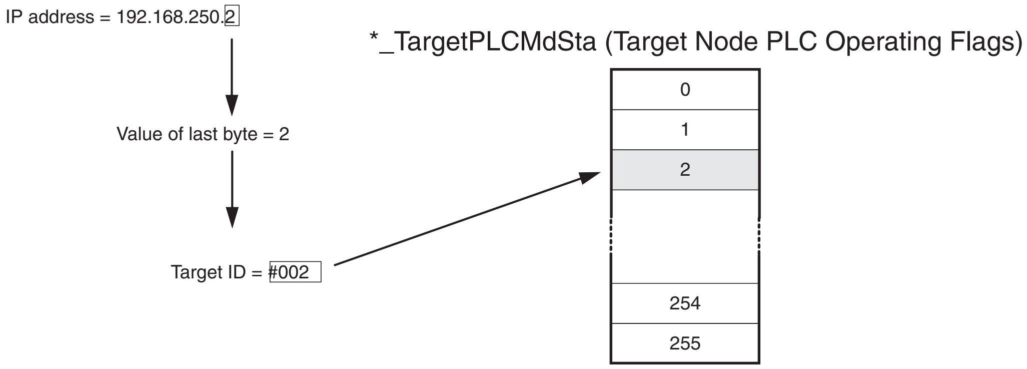


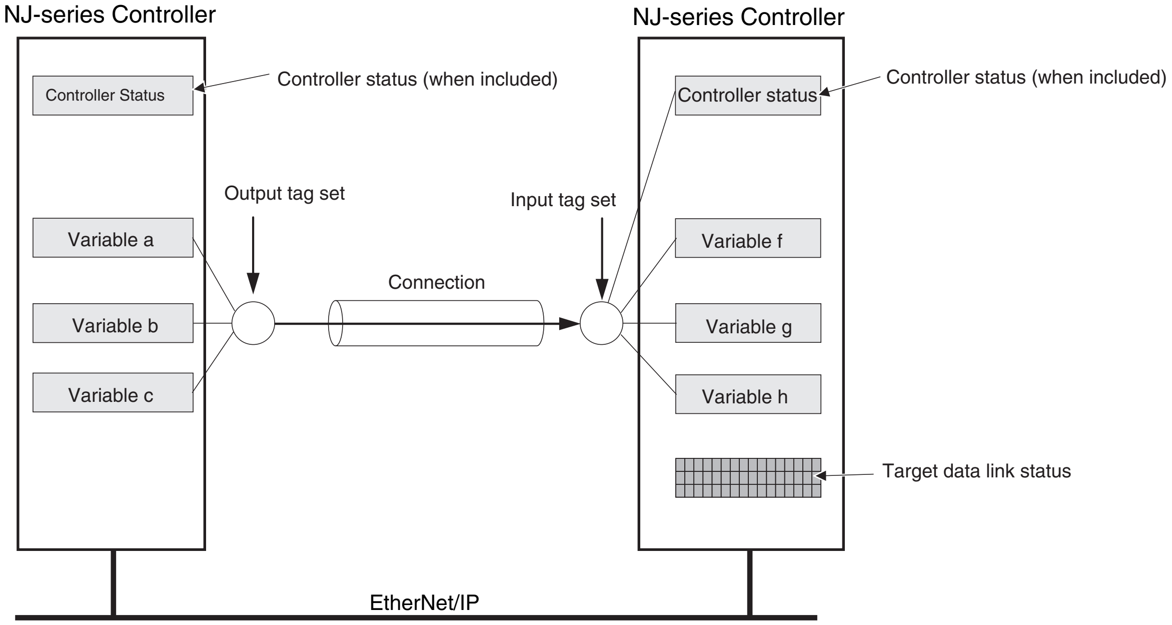
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |

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 device variables for the CJ-series Unit that are given below.

- *_*TargetPLCMdSta* (Target Node PLC Operating Flags)
- *_*TargetPLCErrSta* (Target Node PLC Error Flags)

Sending the Target Node PLC Operating Flags of the Target Node with an IP Address of 192.168.250.2.


IP address = 192.168.250.2
Value of last byte = 2
Target ID = #002
*_TargetPLCMdSta (Target Node PLC Operating Flags)
0
1
2
254
255



### 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 |
|---|---|---|
| Target Node PLC Operating Flags | *_*TargetPLCMdSta* (Target Node PLC Operating Flags) | This variable shows the operation information of the Controller at the target node.<br>When the EtherNet/IP Unit Is the Originator of the Connection<br>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. |

**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 Node PLC Operating Flags and Target Node PLC Error Flags 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/NX-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 four conditions.

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

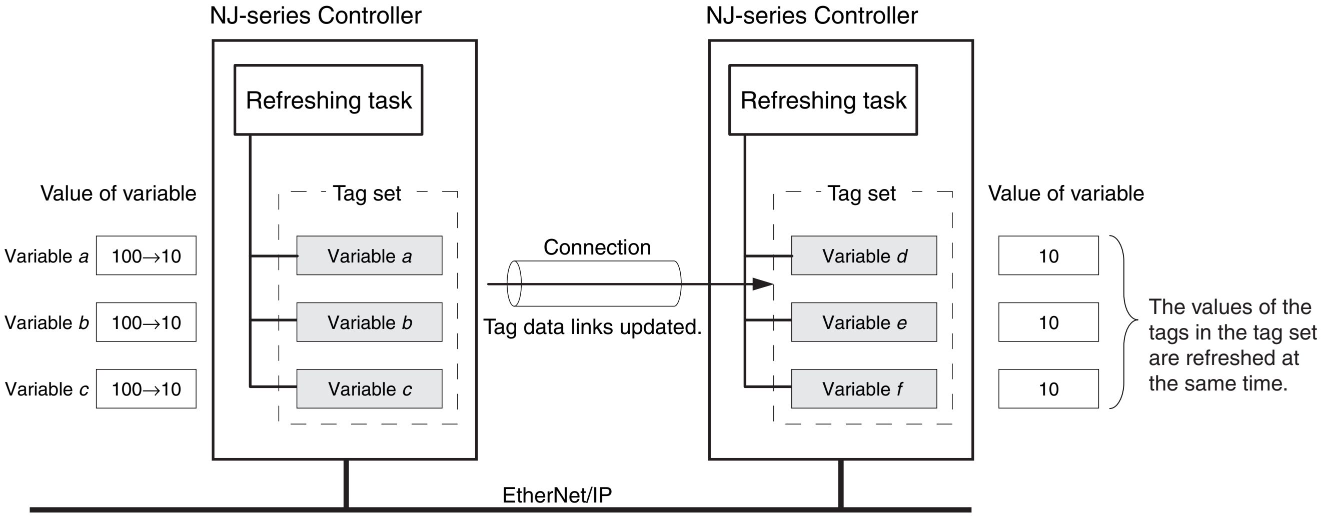
**(2) A refreshing task must be set for the network variables assigned to the tags, and the refreshing task must be the same for all tags in the tag set.***

**Note** 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.

**(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 *14-3-3 Effect of Tag Data Links on Task Periods* for details on the variable access time and data transfer.**

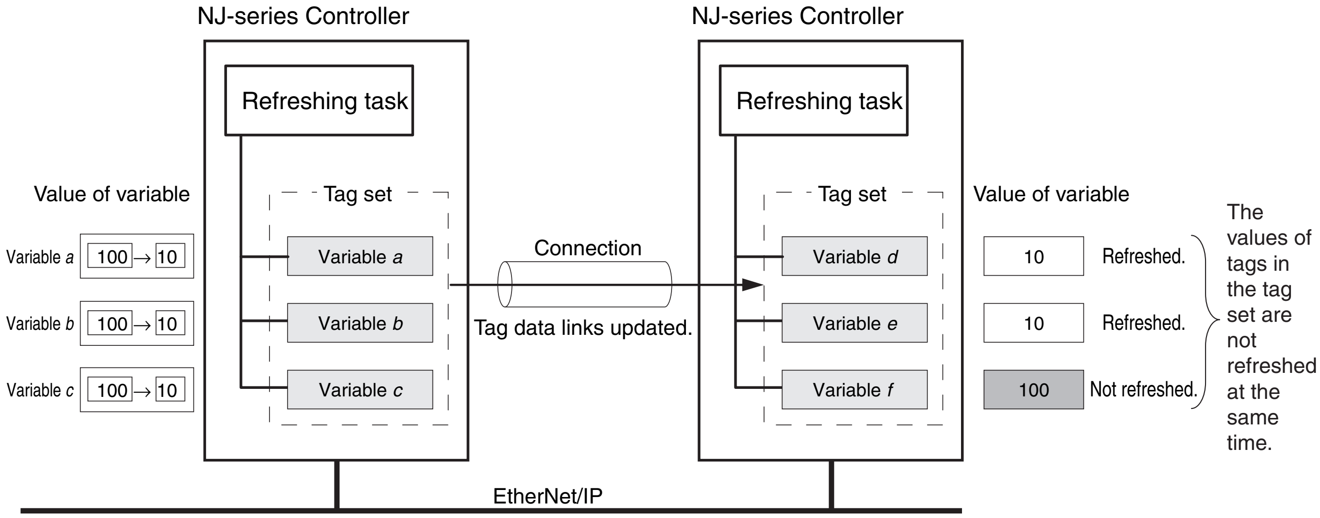
- Setting Refreshing Tasks for Tags (Network Variables)
  Concurrency of the tags in the tag set is maintained.



**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.

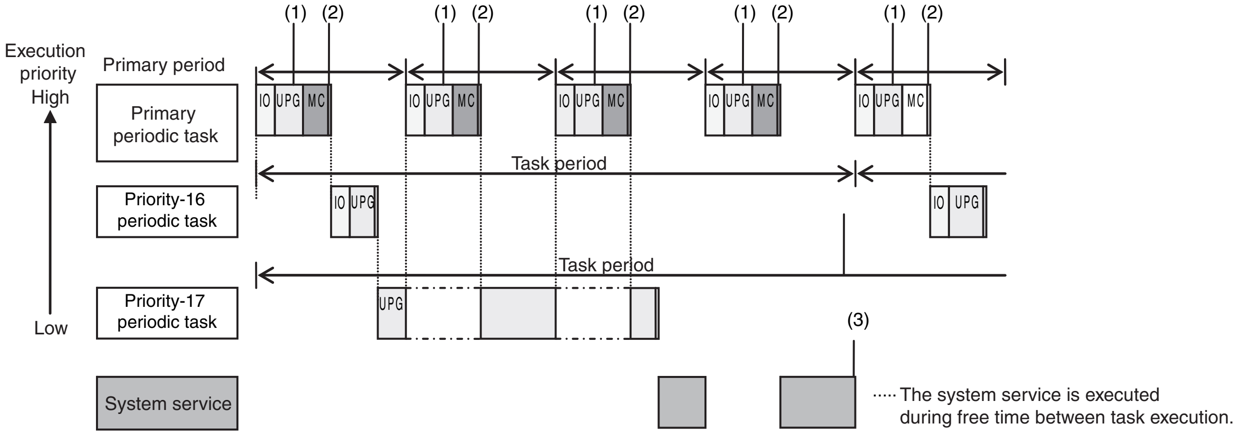


## ● The timing of updating network variables that are assigned to tags is synchronized with the execution period of the program that accesses the network variables.

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.

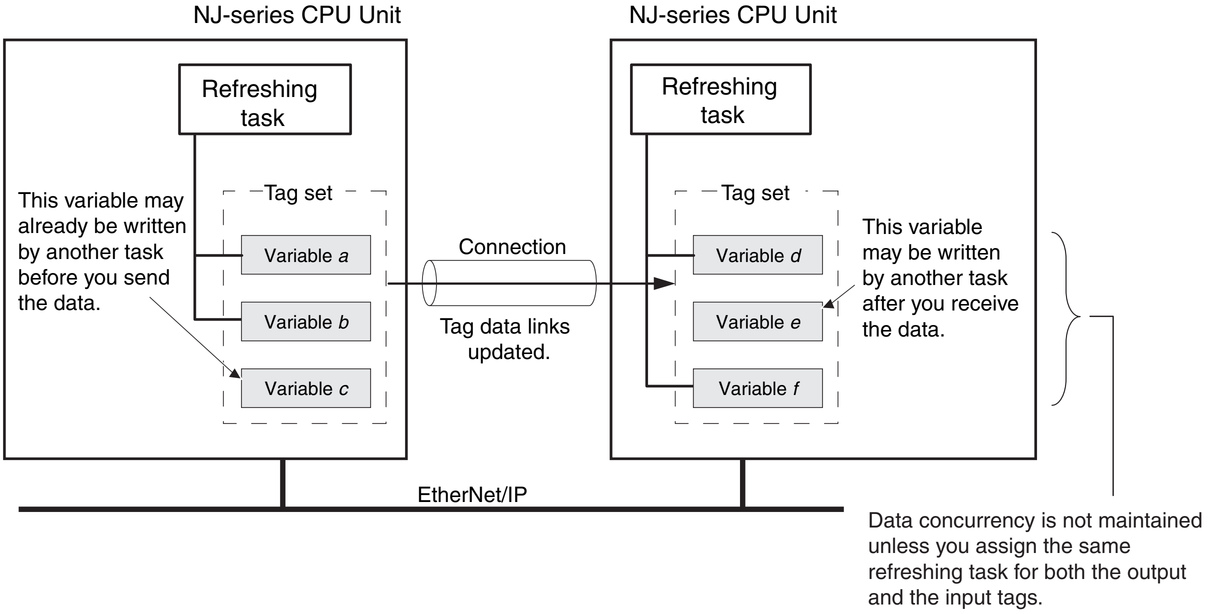
**Additional Information**

**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.


NJ-series CPU Unit
NJ-series CPU Unit
Refreshing task
Refreshing task
Tag set
Tag set
This variable may already be written by another task before you send the data.
Variable a
Variable b
Variable c
Connection
Tag data links updated.
Variable d
Variable e
Variable f
This variable may be written by another task after you receive the data.
EtherNet/IP
Data concurrency is not maintained unless you assign the same refreshing task for both the output and the input tags.

## ● 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 *14-3-3 Effect of Tag Data Links on Task Periods* 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/NX-series CPU Unit Software User’s Manual* (Cat. No. W501).

# 7-2 Setting Tag Data Links

**Additional Information**

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

## 7-2-1 Starting the Network Configurator

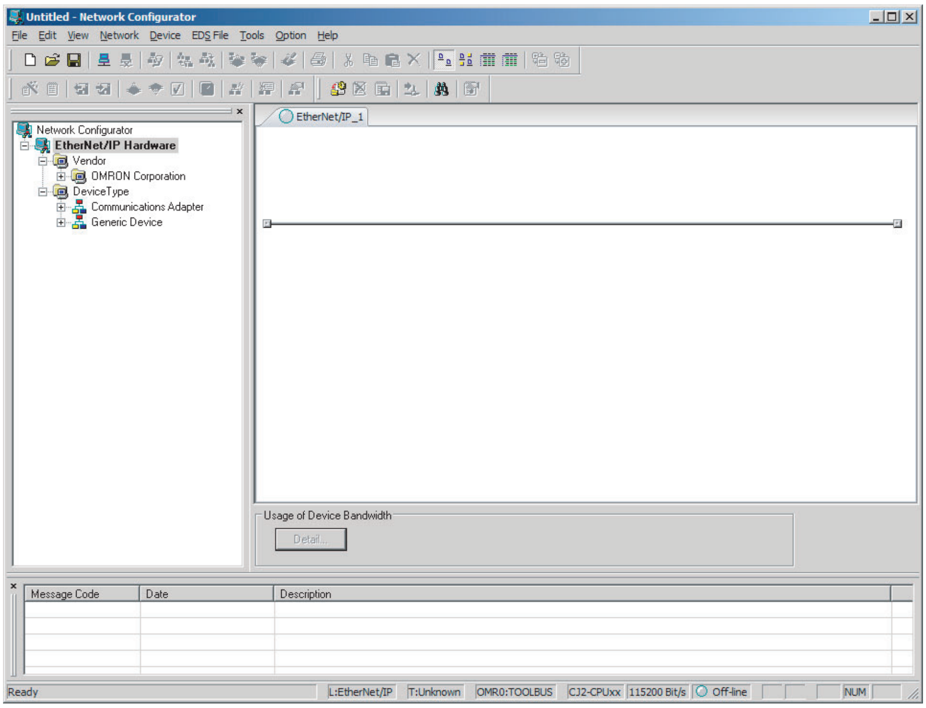
### Procedure

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

- **Using the Windows Start Menu**

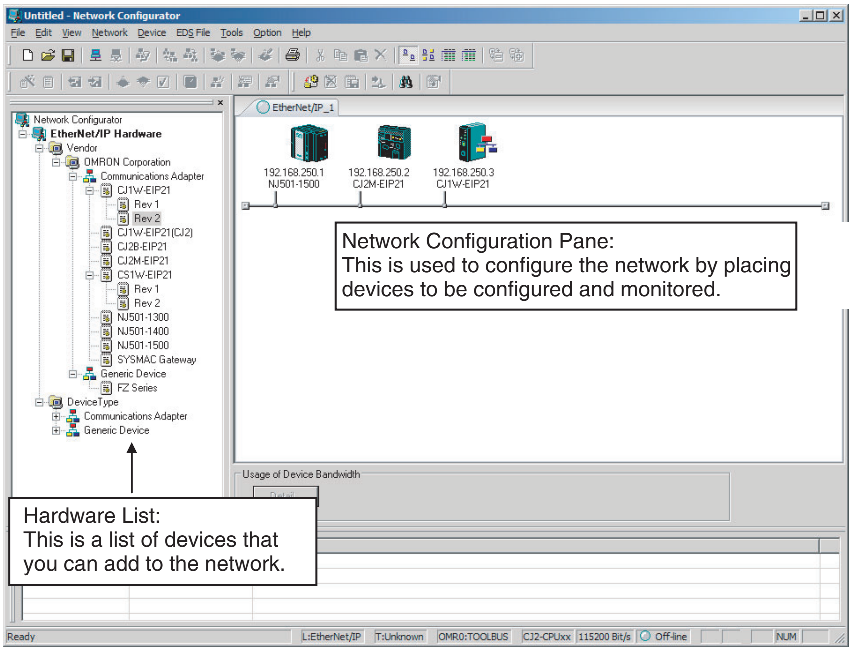
To start the Network configurator, select ***OMRON*** – ***Sysmac Studio*** – ***Network Configurator for EtherNetIP*** – ***Network Configurator*** from the Windows Start Menu.

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



## Main Window

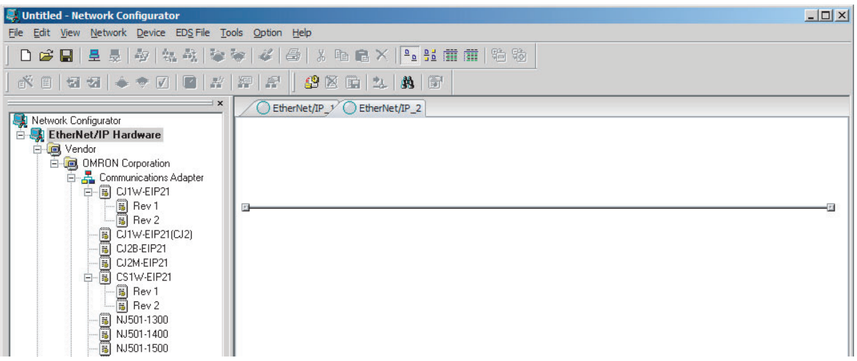
The Main Window consists of a Hardware List and a Network Configuration Pane, as shown in the following diagram.



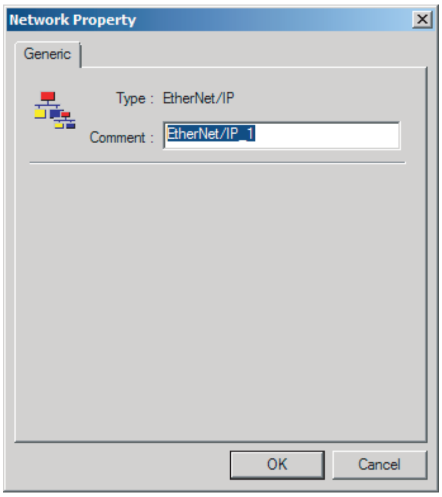
Network Configuration Pane:
This is used to configure the network by placing devices to be configured and monitored.

Hardware List:
This is a list of devices that you can add to the network.

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.



Type : EtherNet/IP

Comment : EtherNet/IP_1

## 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 EtherNet/IP Unit for which to create connections in the EtherNet/IP Network Configuration Pane. (Refer to *7-2-3 Registering Devices*.)**

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

↓

**2 Create the tag and tag set connections.**

**(1) Create tags and tag sets for all registered devices (EtherNet/IP Units). (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*.)**

↓

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

↓

**4 Make sure that the tag data links are operating normally by using the indicators for the EtherNet/IP Unit (refer to *15-3 Connection Status Codes and Error Processing*) and the Network Configurator monitor functions. (Refer to *15-1 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.**

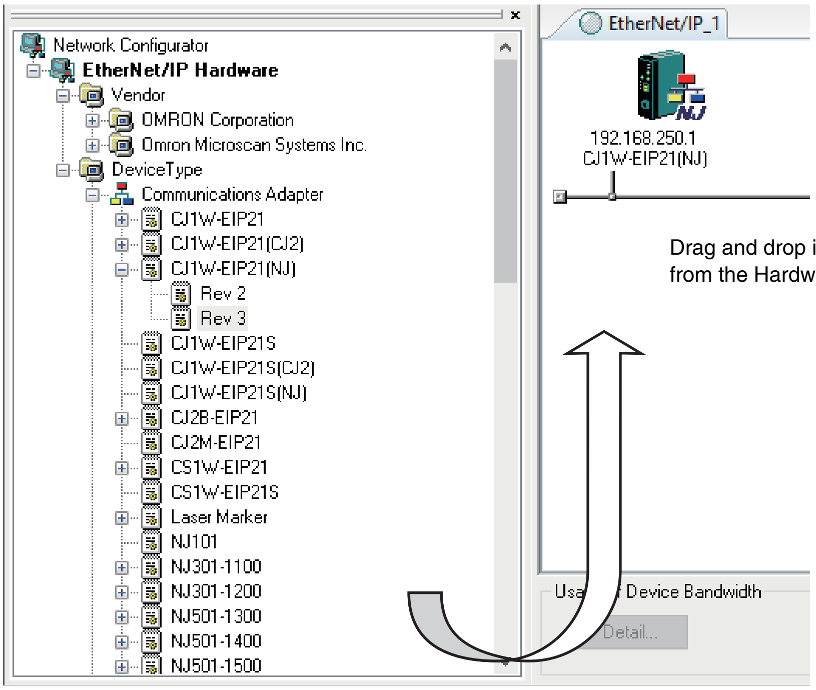
**Note** Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for the procedure.

## 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.



Drag and drop icons from the Hardware List.

The device names and major CIP revisions (Rev □) are displayed in the hardware list.

For EtherNet/IP Units, device names and major CIP revisions are as shown in the following table.

| Name in Hardware List | CIP revision | Unit version |
|---|---|---|
| CJ1W-EIP21(NJ) | Rev. 2 | Ver. 2.1 |
| | Rev. 3 | Ver. 3.0 |
| CJ1W-EIP21S(NJ) | Rev. 4 | Ver. 1.0 |

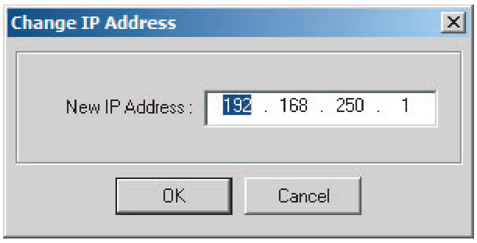
### Precautions for Correct Use

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

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

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

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

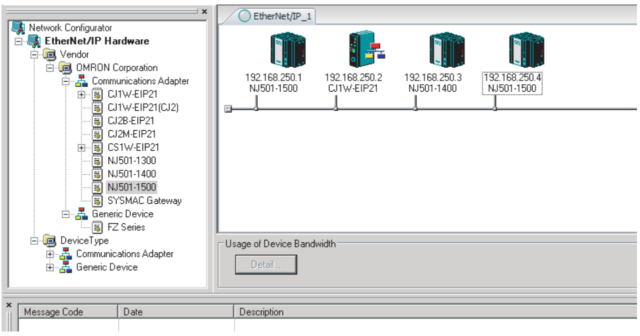


Change IP Address

New IP Address : 192 . 168 . 250 . 1

**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.



## Relationship Between Major CIP Revisions of Registered Device and Device Used in Actual Operation

Whether or not downloading, uploading, and comparison are supported depends on the combination of major CIP revisions of the registered device and the device that you use in the actual operation. The relationships are given in the following table.

| Major CIP revision of the registered device | Major CIP revision of the device used in actual operation | | |
|---|---|---|---|
| | 2 | 3 | 4 or later |
| 2 | Supported | Supported | Supported. |
| 3 | Not supported | Supported | Supported. |
| 4 | Not supported. | Not supported. | Supported. |

## 7-2-4 Creating Tags and Tag Sets

You must create the tag sets and set member tags required to create connections for a registered EtherNet/IP Unit. You can set the network variables used in control programs for tags. This section first describes the basic procedure to create tags and tag sets (1, below). Then it explains how to import variables with a Network Publish attribute from the Sysmac Studio to the Network Configurator (2, below).

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

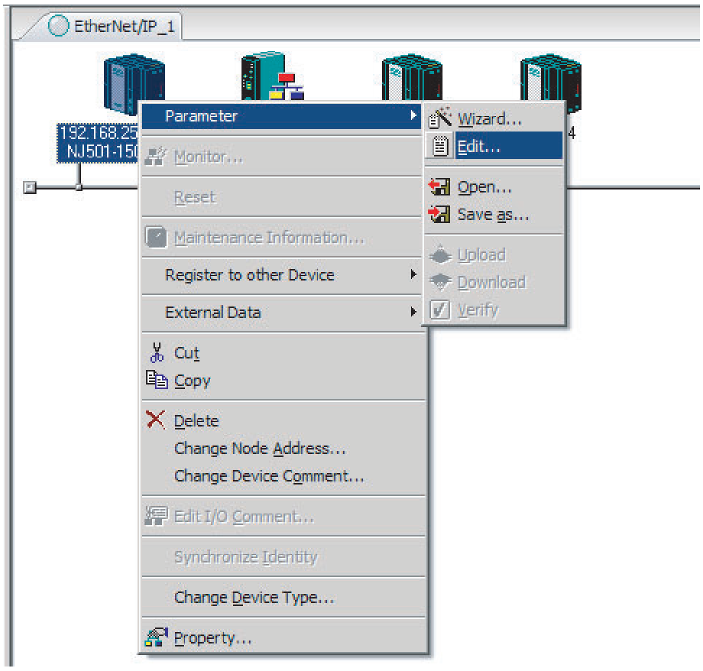
**(1) Creating Tags and Tag Sets with the Network Configurator's Device Parameter Editing Function**

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

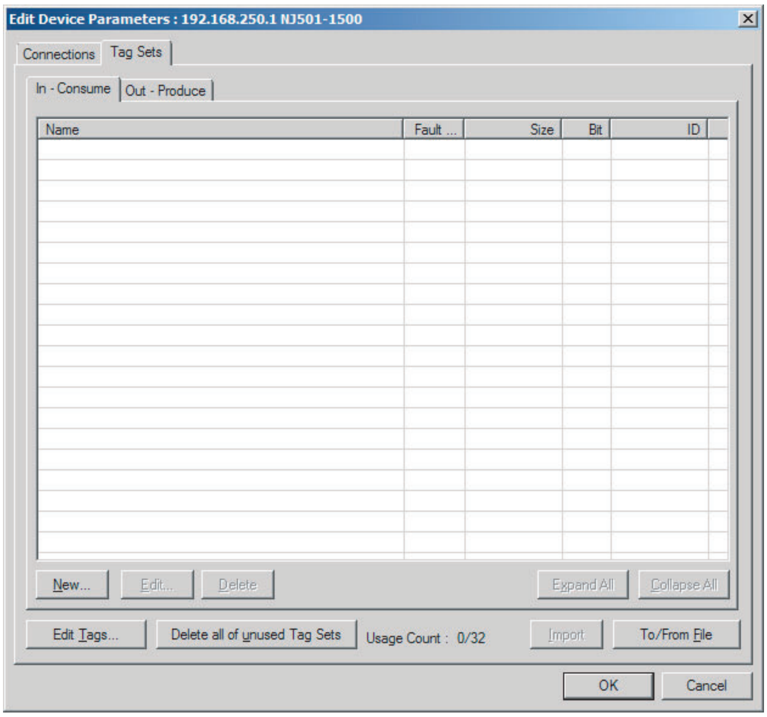
### (1) Creating Tags and Tag Sets with the Network Configurator's Device Parameter Editing Function

#### ● Creating a Tag Set

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



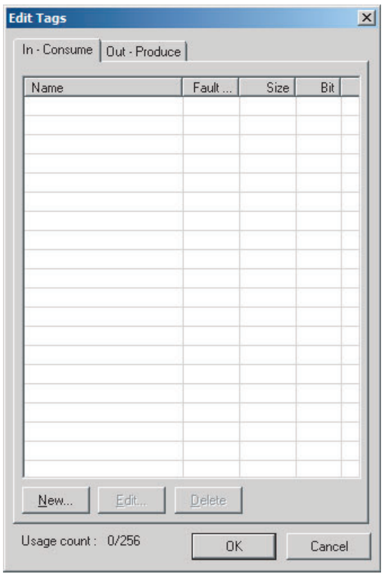
**2** Click the **Tag Sets** Tab at the top of the Edit Device Parameters Dialog Box. There are two kinds of tag sets: input (consume) and output (produce).



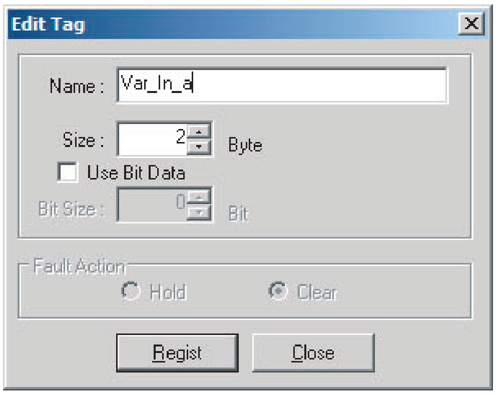
## ● Creating and Adding Tags

**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.



Edit Tag

Name : Var_In_a

Size : 2 Byte

□ Use Bit Data

Bit Size : 0 Bit

Fault Action

○ Hold ● Clear

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

### Additional Information

- You can use the following characters in tag names. 0 to 9, A to Z, a to z, single-byte kana, _ (underbar), and multi-byte characters (e.g., Japanese)
- You cannot use the following characters in tag names. ! “ # $ & ‘ ( ) * + , - . / : ; < = > ? @ [ ] ^ ‘ % spaces or text strings that start with numerals (0 to 9)
- The maximum length of a tag name is 255 bytes.
- Specify array 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

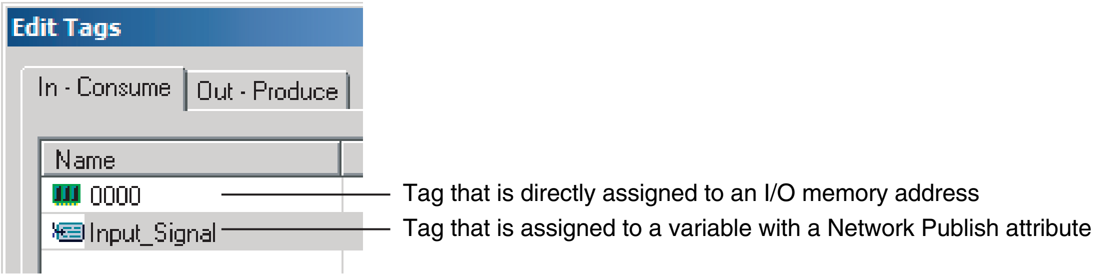
Specify the variable names for tags.
To specify an I/O memory address for a tag, do not specify the I/O memory address for the tag directly. Instead, create a variable, set an AT specification of the I/O memory address on the Sysmac Studio, and then specify the variable with the AT specification for the tag. If you enter I/O memory addresses for tag names on the Network Configurator, as shown below, the tags are directly assigned to the I/O memory addresses in the CPU Unit, and not to the variables. Always specify variable names for tags.

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)

You can check the memory address or variable to which a tag is assigned with the icons in the Edit Tags Dialog Box.


Edit Tags
In - Consume
Out - Produce
Name
0000
Input_Signal
Tag that is directly assigned to an I/O memory address
Tag that is assigned to a variable with a Network Publish attribute

**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.

### Precautions for Correct Use

Tag sizes can be set to odd numbers of bytes for any EtherNet/IP Unit that was manufactured in October 2012 or later. However, the following precaution must be observed.

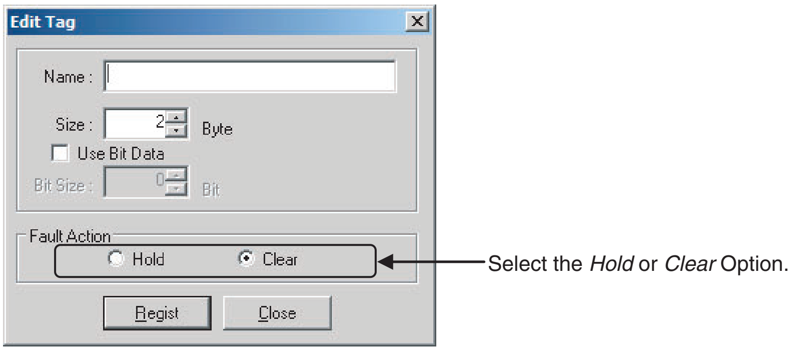
- Memory in the CPU Unit is used in 2-byte increments. Therefore, specify a tag with the desired odd number of bytes on the Network Configurator and then define a variable with a data size that is one byte larger on the Sysmac Studio and specify it for the tag.

**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.


Edit Tag

Name :

Size : 2 Byte

□ Use Bit Data

Bit Size : 0 Bit

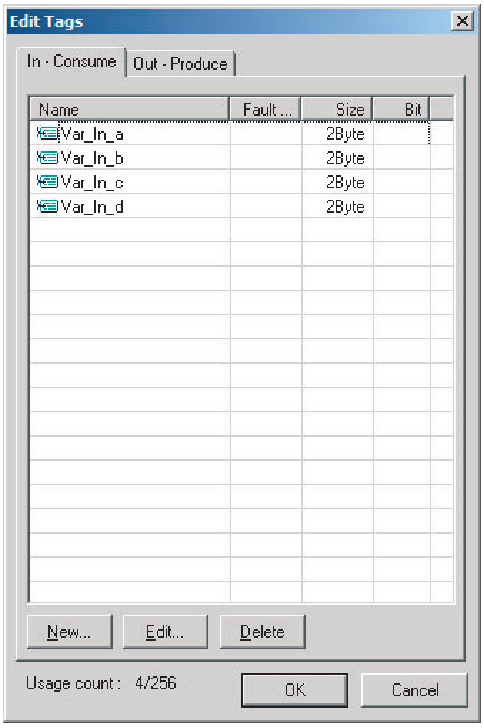
Fault Action: ○ Hold ● Clear

### 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.



| Name | Fault ... | Size | Bit |
|---|---|---|---|
| Var_In_a | | 2Byte | |
| Var_In_b | | 2Byte | |
| Var_In_c | | 2Byte | |
| Var_In_d | | 2Byte | |

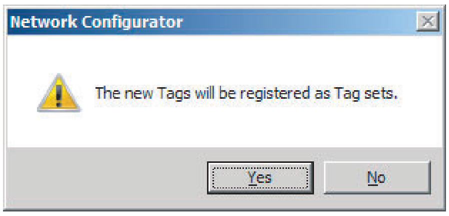
Usage count : 4/256

### 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.

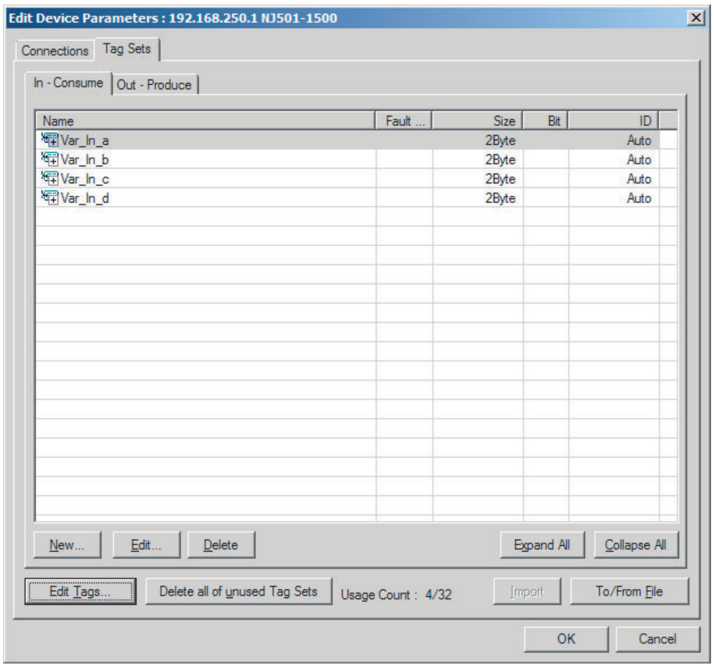


The new Tags will be registered as Tag sets.

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.

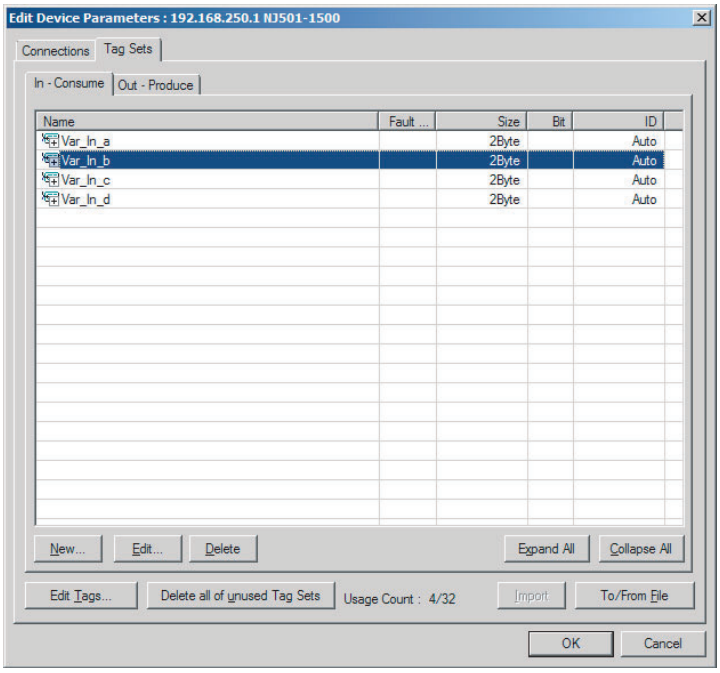


Edit Device Parameters : 192.168.250.1 NJ501-1500

| Name | Fault ... | Size | Bit | ID |
|---|---|---|---|---|
| Var_In_a | | 2Byte | | Auto |
| Var_In_b | | 2Byte | | Auto |
| Var_In_c | | 2Byte | | Auto |
| Var_In_d | | 2Byte | | Auto |

Usage Count : 4/32

**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.)

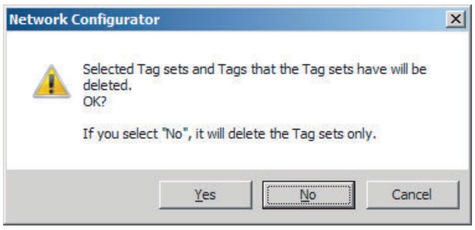


Edit Device Parameters : 192.168.250.1 NJ501-1500

| Name | Fault ... | Size | Bit | ID |
|---|---|---|---|---|
| Var_In_a | | 2Byte | | Auto |
| Var_In_b | | 2Byte | | Auto |
| Var_In_c | | 2Byte | | Auto |
| Var_In_d | | 2Byte | | Auto |

Usage Count : 4/32

A confirmation message is displayed.



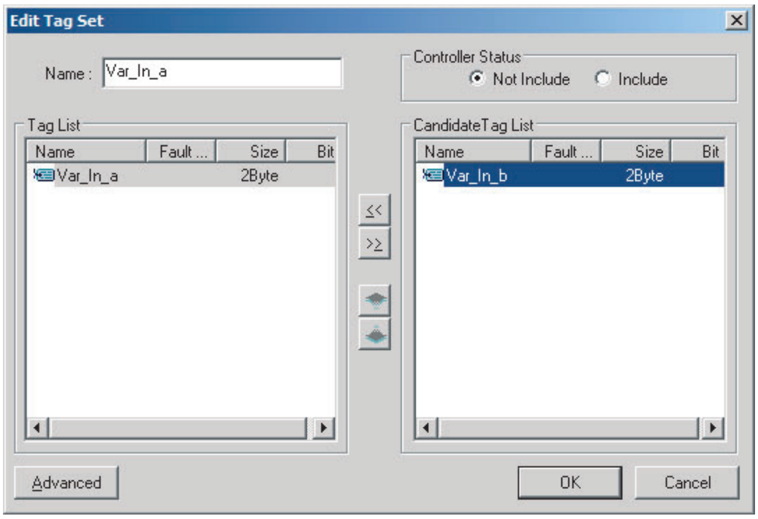
Network Configurator

Selected Tag sets and Tags that the Tag sets have will be deleted.
OK?

If you select "No", it will delete the Tag sets only.

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.



Edit Tag Set

Name : Var_In_a

Controller Status: ● Not Include ○ Include

Tag List

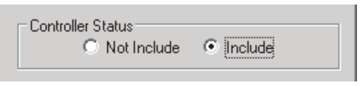
| Name | Fault ... | Size | Bit |
|---|---|---|---|
| Var_In_a | | 2Byte | |

CandidateTag List

| Name | Fault ... | Size | Bit |
|---|---|---|---|
| Var_In_b | | 2Byte | |

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.

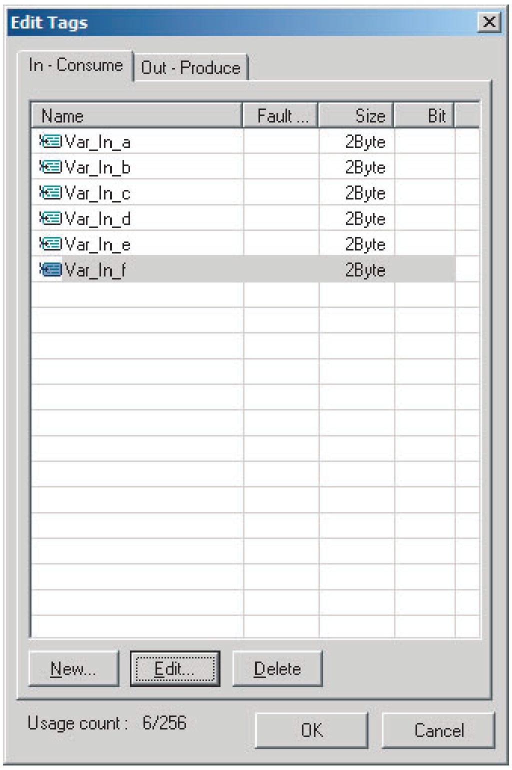


Controller Status: ○ Not Include ● Include

**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.



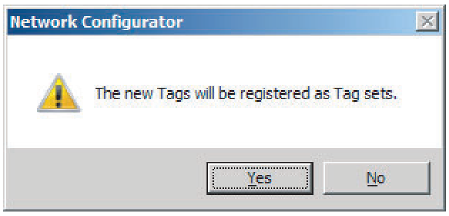
Edit Tags

| Name | Fault ... | Size | Bit |
|---|---|---|---|
| Var_In_a | | 2Byte | |
| Var_In_b | | 2Byte | |
| Var_In_c | | 2Byte | |
| Var_In_d | | 2Byte | |
| Var_In_e | | 2Byte | |
| Var_In_f | | 2Byte | |

Usage count : 6/256

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

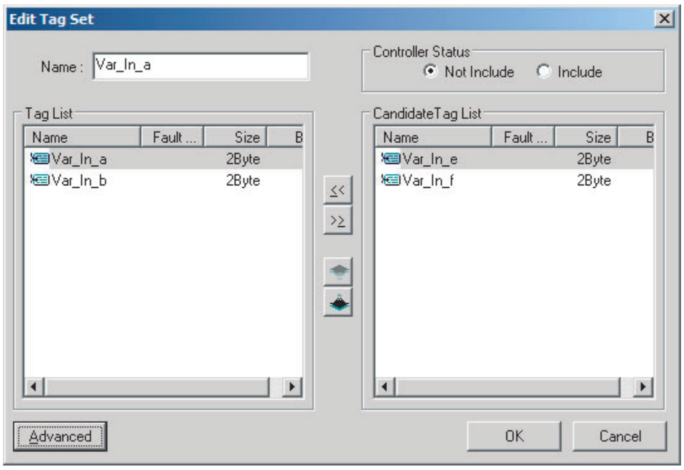
**19** At this point, a confirmation dialog box is displayed to check whether the registered tag names are used as the tag set names. Tags are just added in this case, so click the **No** Button. Just the tags are registered. The tags are not registered as tag sets.



Network Configurator

The new Tags will be 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.



Edit Tag Set

Name : Var_In_a

Controller Status: ● Not Include ○ Include

Tag List

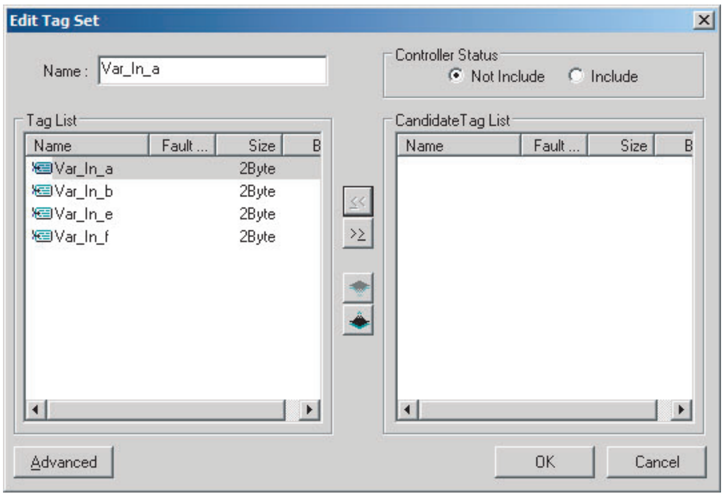
| Name | Fault ... | Size | B |
|---|---|---|---|
| Var_In_a | | 2Byte | |
| Var_In_b | | 2Byte | |

CandidateTag List

| Name | Fault ... | Size | B |
|---|---|---|---|
| Var_In_e | | 2Byte | |
| Var_In_f | | 2Byte | |

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.



Edit Tag Set

Name : Var_In_a

Controller Status: ● Not Include ○ Include

Tag List

| Name | Fault ... | Size | B |
|---|---|---|---|
| Var_In_a | | 2Byte | |
| Var_In_b | | 2Byte | |
| Var_In_e | | 2Byte | |
| Var_In_f | | 2Byte | |

CandidateTag List

| Name | Fault ... | Size | B |
|---|---|---|---|

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.

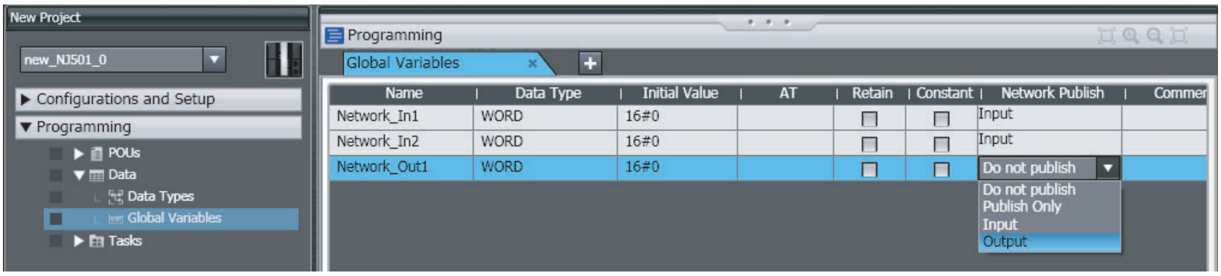
**23** Click the **OK** Button in the Edit Device Parameters Dialog Box.

## (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.



| Name | Data Type | Initial Value | AT | Retain | Constant | Network Publish | Commer |
|---|---|---|---|---|---|---|---|
| Network_In1 | WORD | 16#0 | | □ | □ | Input | |
| Network_In2 | WORD | 16#0 | | □ | □ | Input | |
| Network_Out1 | WORD | 16#0 | | □ | □ | Do not publish | |

**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)

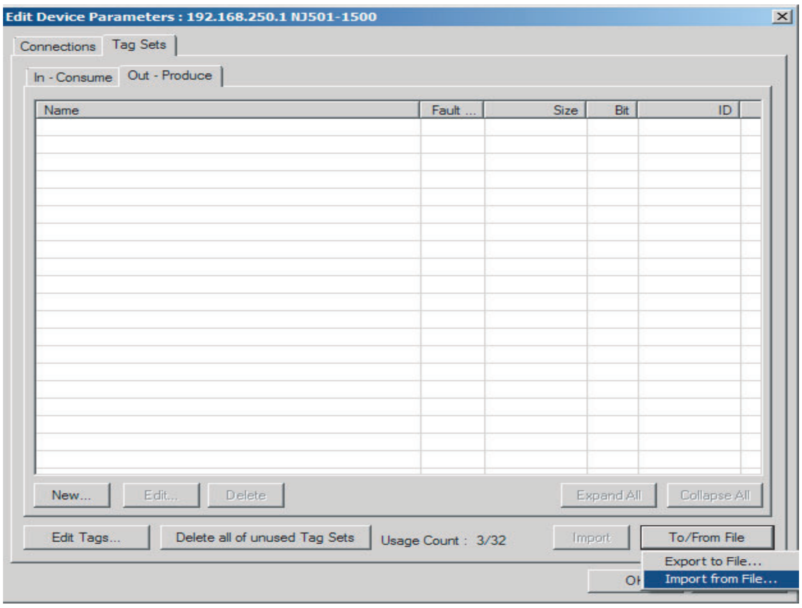
2) Variable names with the following single-byte letters (uppercase or lowercase) followed by single-byte numerals

- 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.

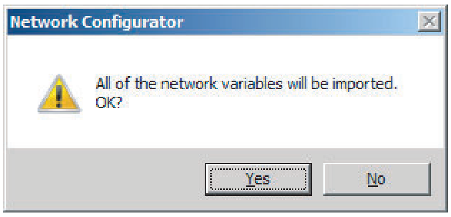
Right-click the icon to display the pop-up menu, and select ***Device*** – ***Parameter*** – ***Edit***.

**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.



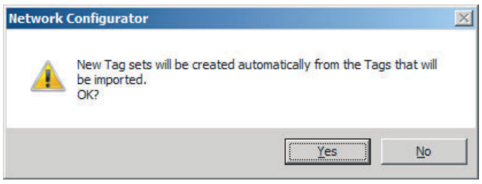
A confirmation dialog box is displayed that asks you how you want to import the variables as shown below.

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



All of the network variables will be imported.
OK?

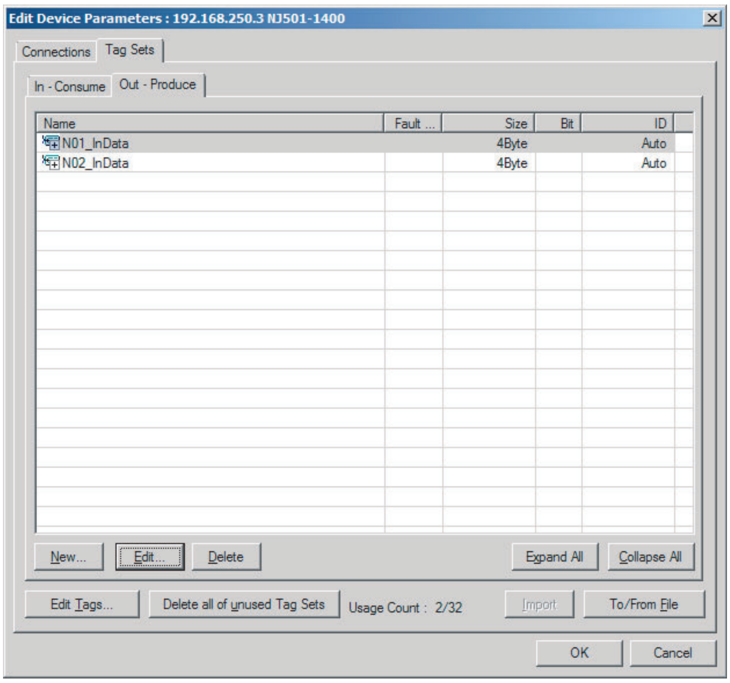
After you import the variables to the tags, click the **Yes** Button to automatically create tag sets, or click the **No** Button to set up tag sets manually.



Network Configurator

New Tag sets will be created automatically from the Tags that will be imported.
OK?

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.)

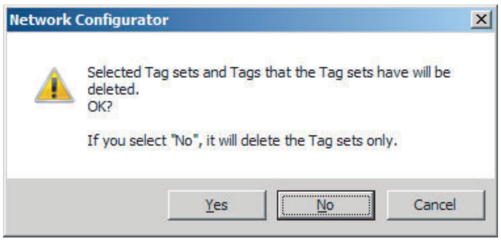


Edit Device Parameters : 192.168.250.3 NJ501-1400

| Name | Fault ... | Size | Bit | ID |
|---|---|---|---|---|
| N01_InData | | 4Byte | | Auto |
| N02_InData | | 4Byte | | Auto |

Usage Count : 2/32

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.

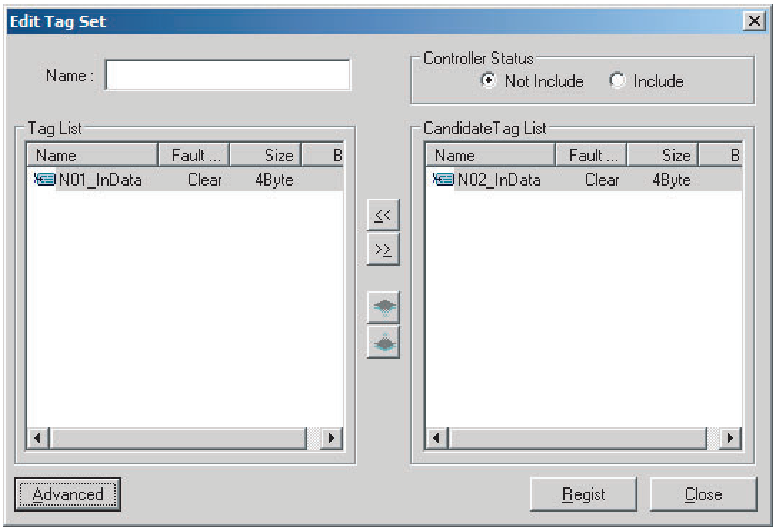


Network Configurator

Selected Tag sets and Tags that the Tag sets have will be deleted.
OK?

If you select "No", it will delete the Tag sets only.

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.



Controller Status: ● Not Include ○ Include

Tag List

| Name | Fault ... | Size | B |
|---|---|---|---|
| N01_InData | Clear | 4Byte | |

CandidateTag List

| Name | Fault ... | Size | B |
|---|---|---|---|
| N02_InData | Clear | 4Byte | |

**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
- The connection type (multi-cast or unicast)
- The length of the packet intervals (RPI)
- Connection name (optional)

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

### Precautions for Correct Use

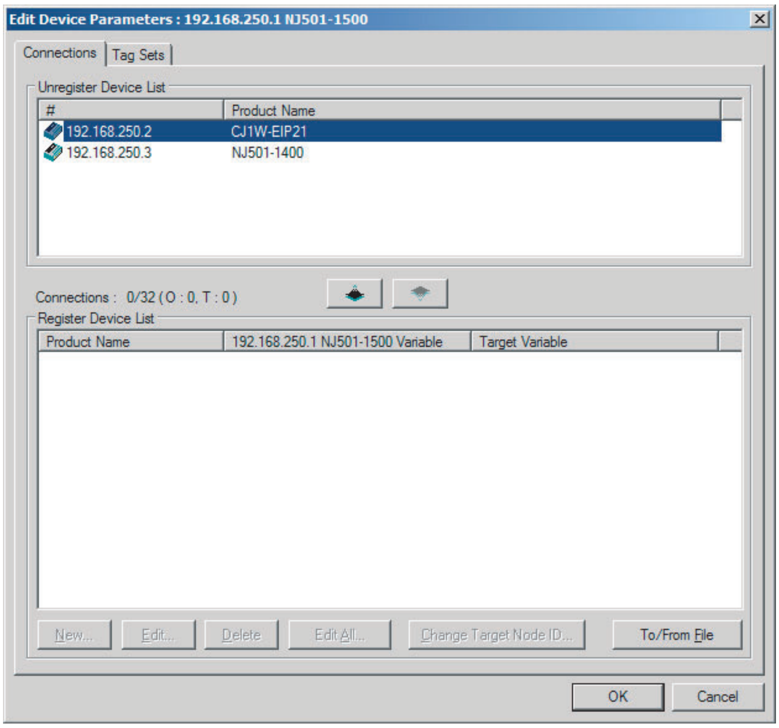
Make the connections settings after you create tag sets for all of the devices involved in tag data links.

## Connection Settings (Connections Tab Page)

### ● Registering Devices in the Register Device List

1. Double-click the icon of the device for which to make originator settings in the Network Configuration Pane of the Network Configurator. The Edit Device Parameters Dialog Box is displayed. Right-click the icon to display the pop-up menu, and select ***Parameter – Edit***.

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



Edit Device Parameters : 192.168.250.1 NJ501-1500

Unregister Device List

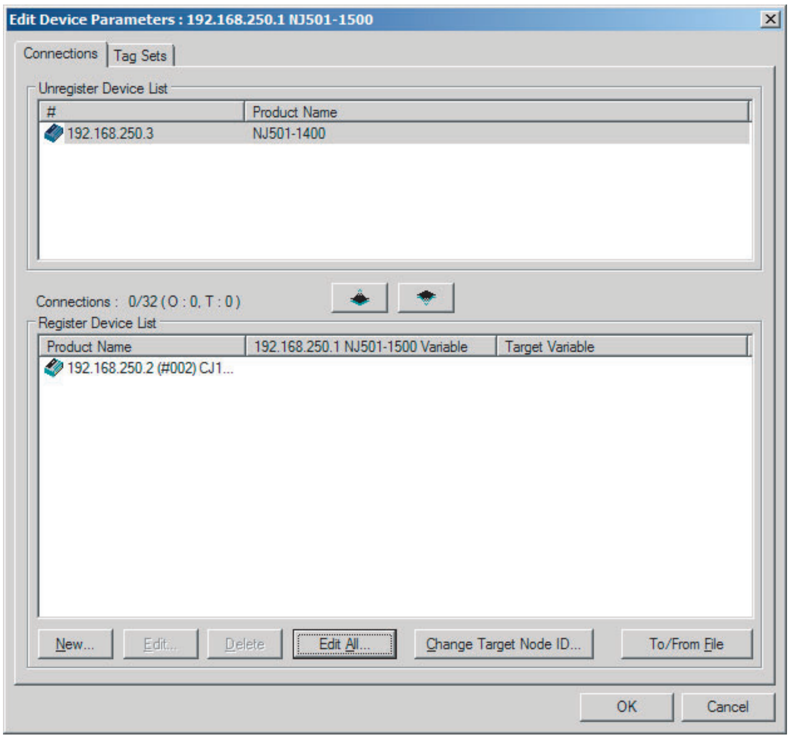
| # | Product Name |
|---|---|
| 192.168.250.2 | CJ1W-EIP21 |
| 192.168.250.3 | NJ501-1400 |

Connections : 0/32 ( O : 0, T : 0 )

Register Device List

| Product Name | 192.168.250.1 NJ501-1500 Variable | Target Variable |
|---|---|---|

**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.



Edit Device Parameters : 192.168.250.1 NJ501-1500

Unregister Device List

| # | Product Name |
|---|---|
| 192.168.250.3 | NJ501-1400 |

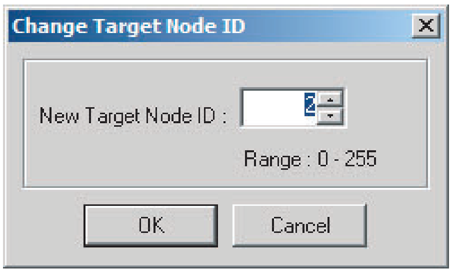
Connections : 0/32 ( O : 0, T : 0 )

Register Device List

| Product Name | 192.168.250.1 NJ501-1500 Variable | Target Variable |
|---|---|---|
| 192.168.250.2 (#002) CJ1... | | |

**4** Target node IDs are assigned to the devices that are registered in the Register Device List.

The target node ID serves as the bit array position for the following variables in the originator Controller: Target Node PLC Operating Flags, Target Node PLC Error Flags, Target Node Error Information, Registered Target Node Table, and Normal Target Node Table. 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.



Change Target Node ID

New Target Node ID : 2

Range : 0 - 255

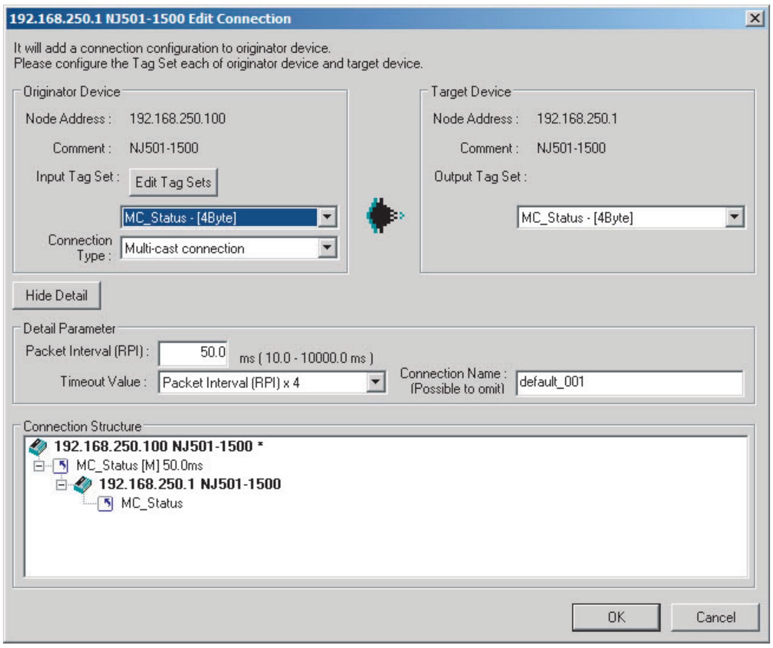
## ● Editing Settings for Individual Connections

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)



192.168.250.1 NJ501-1500 Edit Connection

It will add a connection configuration to originator device.
Please configure the Tag Set each of originator device and target device.

Originator Device
Node Address : 192.168.250.100
Comment : NJ501-1500
Input Tag Set : Edit Tag Sets
MC_Status - [4Byte]
Connection Type : Multi-cast connection

Target Device
Node Address : 192.168.250.1
Comment : NJ501-1500
Output Tag Set :
MC_Status - [4Byte]

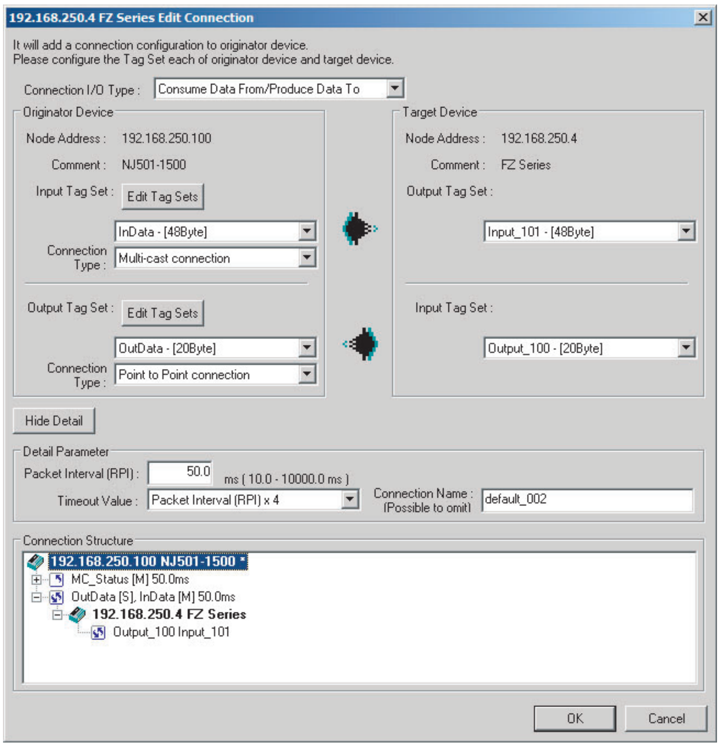
Hide Detail

Detail Parameter
Packet Interval (RPI) : 50.0 ms ( 10.0 - 10000.0 ms )
Timeout Value : Packet Interval (RPI) x 4
Connection Name : (Possible to omit) default_001

Connection Structure
192.168.250.100 NJ501-1500 *
MC_Status [M] 50.0ms
192.168.250.1 NJ501-1500
MC_Status

OK Cancel

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



192.168.250.4 FZ Series Edit Connection

It will add a connection configuration to originator device.
Please configure the Tag Set each of originator device and target device.

Connection I/O Type : Consume Data From/Produce Data To

Originator Device
Node Address : 192.168.250.100
Comment : NJ501-1500
Input Tag Set : Edit Tag Sets
InData - [48Byte]
Connection Type : Multi-cast connection
Output Tag Set : Edit Tag Sets
OutData - [20Byte]
Connection Type : Point to Point connection

Target Device
Node Address : 192.168.250.4
Comment : FZ Series
Output Tag Set :
Input_101 - [48Byte]
Input Tag Set :
Output_100 - [20Byte]

Hide Detail

Detail Parameter
Packet Interval (RPI) : 50.0 ms ( 10.0 - 10000.0 ms )
Timeout Value : Packet Interval (RPI) x 4
Connection Name : (Possible to omit) default_002

Connection Structure
192.168.250.100 NJ501-1500 *
MC_Status [M] 50.0ms
OutData [S], InData [M] 50.0ms
192.168.250.4 FZ Series
Output_100 Input_101

OK Cancel

The settings are as follows:

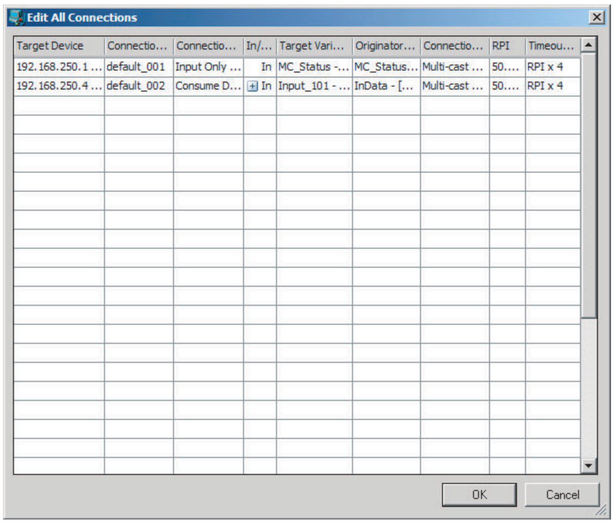
| Setting | Description |
|---|---|
| Connection I/O Type | Select *Input Only (Tag type)* to use tag data links with a CS1W-EIP21, CS1W-EIP21S, CJ1W-EIP21, CJ2B-EIP21, CJ2M-EIP21, CJ1W-EIP21(CJ2), CJ1W-EIP21(NJ) , CJ1W-EIP21S, CJ1W-EIP21S(CJ2), CJ1W-EIP21S(NJ), NX701, NJ501-□□□□, NJ301-□□□□, or NJ101 CPU Unit.<br>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.<br>• Multi-cast connection: Select when the same data is shared by multiple nodes. This setting is usually used.<br>• 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.<br>**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.<br>You can set an interval to between 0.5 and 10,000 ms in 0.5-ms increments (NJ-series Controllers: between 10 and 10,000 ms in 1-ms increments).<br>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.



Edit All Connections

| Target Device | Connectio… | Connectio… | In/… | Target Vari… | Originator… | Connectio… | RPI | Timeou… |
|---|---|---|---|---|---|---|---|---|
| 192.168.250.1 … | default_001 | Input Only … | In | MC_Status -… | MC_Status… | Multi-cast … | 50.… | RPI x 4 |
| 192.168.250.4 … | default_002 | Consume D… | ⊞ In | Input_101 - … | InData - [… | Multi-cast … | 50.… | RPI x 4 |

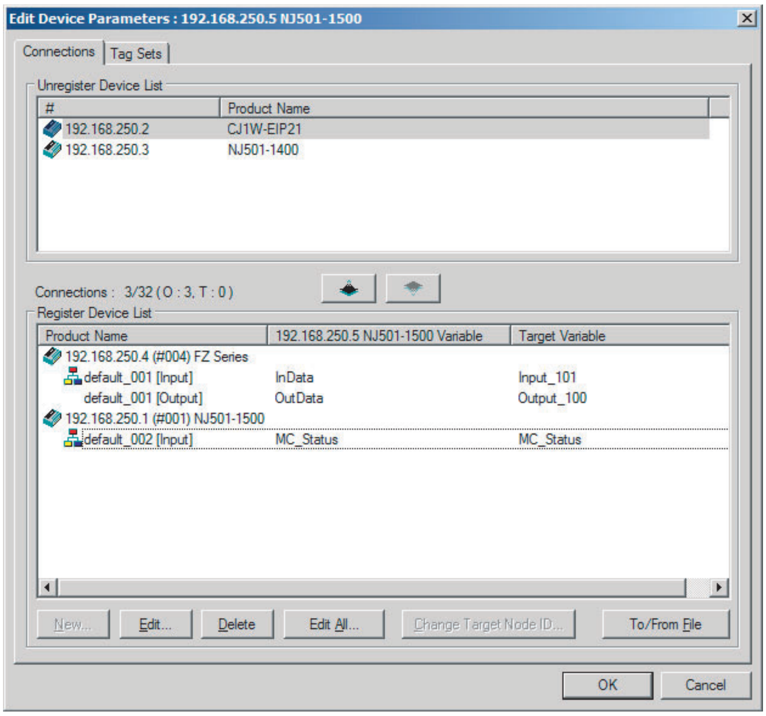
The settings are as follows:

| 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 *Input Only (Tag type)* to use tag data links with a CS1W-EIP21, CS1W-EIP21S, CJ1W-EIP21, CJ2B-EIP21, CJ2M-EIP21, CJ1W-EIP21(CJ2), CJ1W-EIP21(NJ) , CJ1W-EIP21S, CJ1W-EIP21S(CJ2), CJ1W-EIP21S(NJ), NX701, NJ501-□□□□, NJ301-□□□□, or NJ101 CPU Unit.<br>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.<br>• Input Only: Just *In* is displayed. |
| Target Variable | Select the target node's tag set to assign it.<br>• In: Select the target's output (produce) tag set.<br>• Out: Select the target's input (consume) tag set. |
| Originator Variable | Select the originator node's tag set to assign it.<br>• In: Select the originator's input (consume) tag set.<br>• 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.<br>• Multi-cast connection: Select when the same data is shared by multiple nodes. This setting is usually used.<br>• 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.<br>**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 0.5 and 10,000 ms in 0.5-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.



Edit Device Parameters : 192.168.250.5 NJ501-1500

Connections | Tag Sets

Unregister Device List

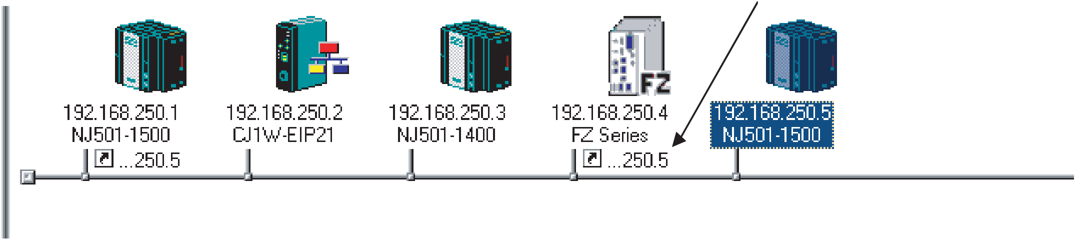
| # | Product Name |
|---|---|
| 192.168.250.2 | CJ1W-EIP21 |
| 192.168.250.3 | NJ501-1400 |

Connections : 3/32 (O : 3, T : 0)

Register Device List

| Product Name | 192.168.250.5 NJ501-1500 Variable | Target Variable |
|---|---|---|
| 192.168.250.4 (#004) FZ Series | | |
| default_001 [Input] | InData | Input_101 |
| default_001 [Output] | OutData | Output_100 |
| 192.168.250.1 (#001) NJ501-1500 | | |
| default_002 [Input] | MC_Status | MC_Status |

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



Indicates the IP address of the originator where the connection was set.

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

### Precautions for Correct Use

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

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

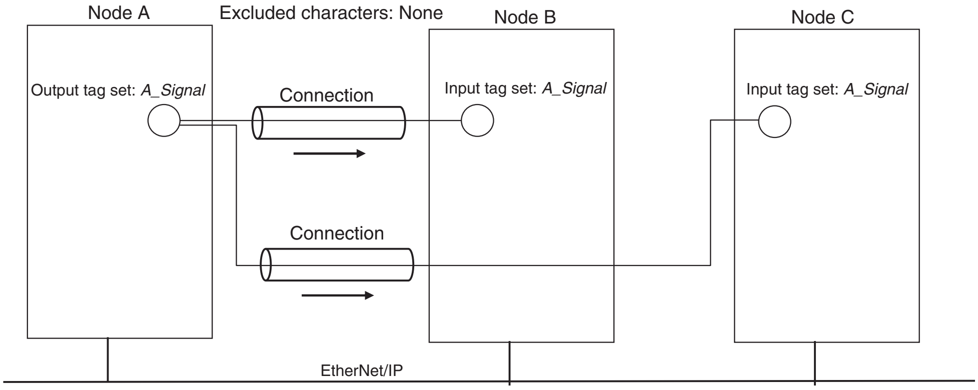
## Automatically Setting Connections (*Network* – *Auto Connection*)

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

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

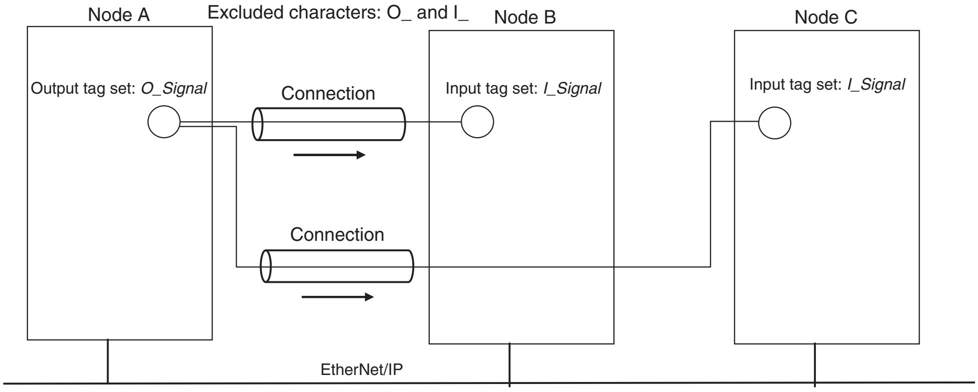
### Example 1: Automatic Connections with the Same Tag Set Names

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


Node A
Excluded characters: None
Node B
Node C
Output tag set: *A_Signal*
Connection
Input tag set: *A_Signal*
Input tag set: *A_Signal*
Connection
EtherNet/IP

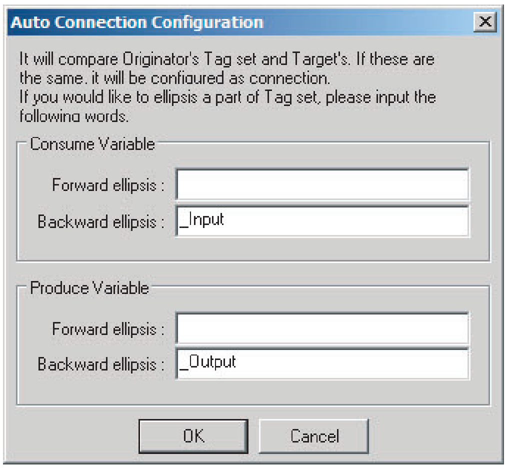
### Example 2: Automatic Connections with the Ellipses

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.


Node A
Excluded characters: O_ and I_
Node B
Node C
Output tag set: *O_Signal*
Connection
Input tag set: *I_Signal*
Input tag set: *I_Signal*
Connection
EtherNet/IP

**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.



It will compare Originator's Tag set and Target's. If these are the same, it will be configured as connection.
If you would like to ellipsis a part of Tag set, please input the following words.

Consume Variable

Forward ellipsis :

Backward ellipsis : _Input

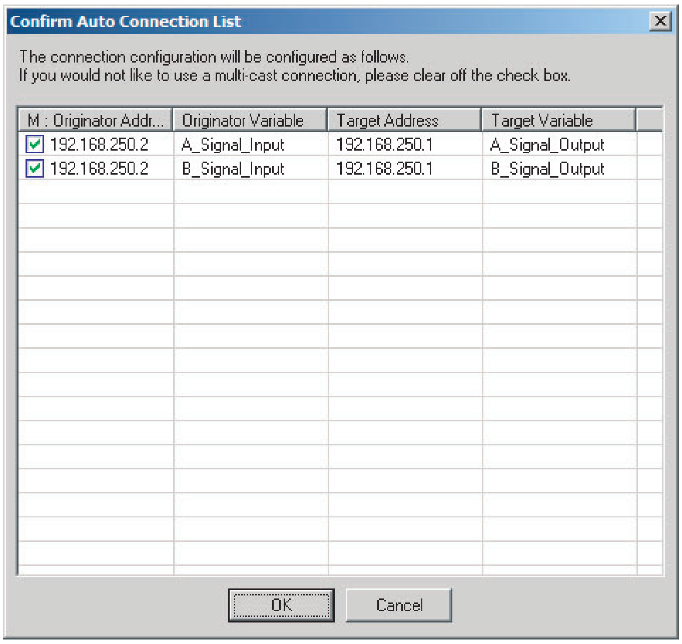
Produce Variable

Forward ellipsis :

Backward ellipsis : _Output

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.

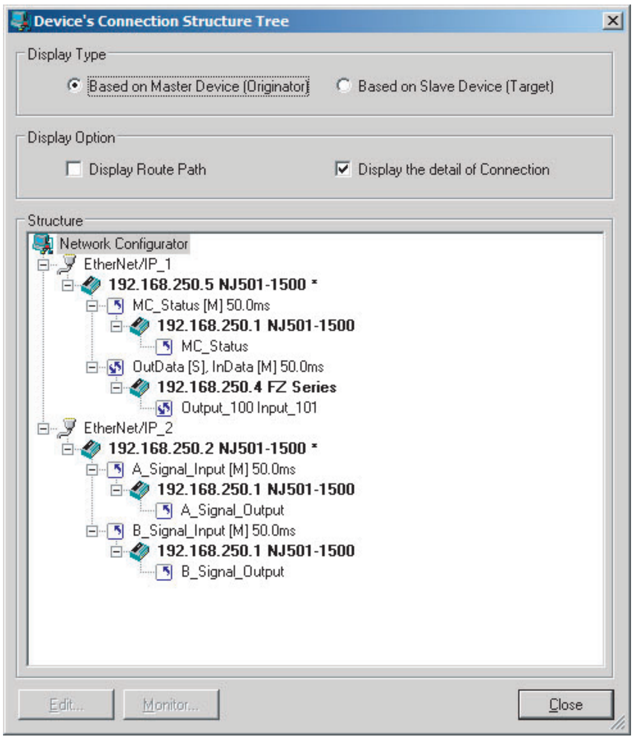


The connection configuration will be configured as follows.
If you would not like to use a multi-cast connection, please clear off the check box.

| M : Originator Addr... | Originator Variable | Target Address | Target Variable |
|---|---|---|---|
| ☑ 192.168.250.2 | A_Signal_Input | 192.168.250.1 | A_Signal_Output |
| ☑ 192.168.250.2 | B_Signal_Input | 192.168.250.1 | B_Signal_Output |

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

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



Device's Connection Structure Tree

Display Type

● Based on Master Device (Originator) ○ Based on Slave Device (Target)

Display Option

□ Display Route Path ☑ Display the detail of Connection

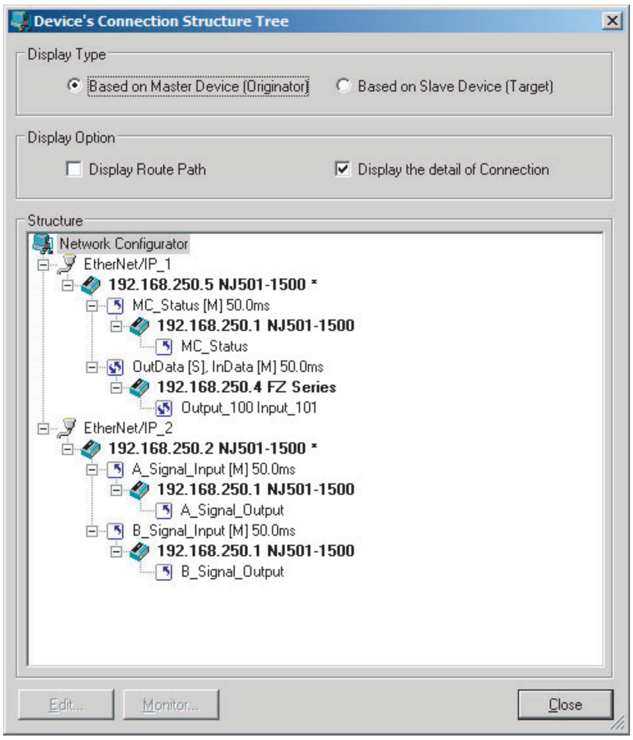
Structure

```
Network Configurator
⊟ EtherNet/IP_1
  ⊟ 192.168.250.5 NJ501-1500 *
    ⊟ MC_Status [M] 50.0ms
      ⊟ 192.168.250.1 NJ501-1500
          MC_Status
    ⊟ OutData [S], InData [M] 50.0ms
      ⊟ 192.168.250.4 FZ Series
          Output_100 Input_101
⊟ EtherNet/IP_2
  ⊟ 192.168.250.2 NJ501-1500 *
    ⊟ A_Signal_Input [M] 50.0ms
      ⊟ 192.168.250.1 NJ501-1500
          A_Signal_Output
    ⊟ B_Signal_Input [M] 50.0ms
      ⊟ 192.168.250.1 NJ501-1500
          B_Signal_Output
```

**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.



Device's Connection Structure Tree

Display Type

● Based on Master Device (Originator) ○ Based on Slave Device (Target)

Display Option

□ Display Route Path ☑ Display the detail of Connection

Structure

```
Network Configurator
⊟ EtherNet/IP_1
  ⊟ 192.168.250.5 NJ501-1500 *
    ⊟ MC_Status [M] 50.0ms
      ⊟ 192.168.250.1 NJ501-1500
          MC_Status
    ⊟ OutData [S], InData [M] 50.0ms
      ⊟ 192.168.250.4 FZ Series
          Output_100 Input_101
⊟ EtherNet/IP_2
  ⊟ 192.168.250.2 NJ501-1500 *
    ⊟ A_Signal_Input [M] 50.0ms
      ⊟ 192.168.250.1 NJ501-1500
          A_Signal_Output
    ⊟ B_Signal_Input [M] 50.0ms
      ⊟ 192.168.250.1 NJ501-1500
          B_Signal_Output
```

- You can use the *Display the detail of Connection* Check Box to switch between device-level and connection-level displays of tag data link communications.
- An asterisk is displayed after the device name of the originator set for the connection.

- The Edit Device Parameters Dialog Box is displayed if you select a connection and click the **Edit** Button. You can edit the connections in this dialog box.

## 7-2-6 Creating Connections Using the Wizard

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

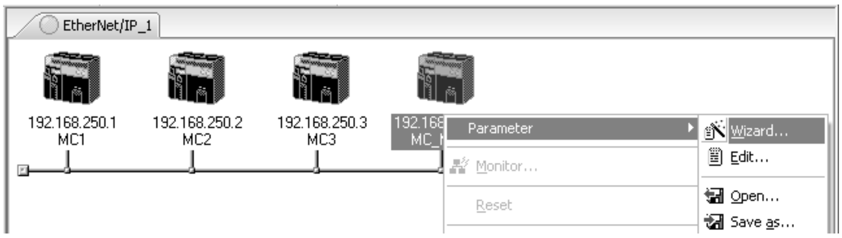
**Note** The Wizard can be used only with the following OMRON EtherNet/IP devices.

| Device name | Remarks |
|---|---|
| CJ1W-EIP21(NJ) | CJ1W-EIP21 mounted to NJ-series CPU Unit |
| CJ1W-EIP21 | CJ1W-EIP21 mounted to CJ1 CPU Unit |
| CJ1W-EIP21(CJ2) | CJ1W-EIP21 mounted to CJ2 CPU Unit |
| CJ1W-EIP21S(NJ) | CJ1W-EIP21S mounted to NJ-series CPU Unit |
| CJ1W-EIP21S | CJ1W-EIP21S mounted to CJ1 CPU Unit |
| CJ1W-EIP21S(CJ2) | CJ1W-EIP21S mounted to CJ2 CPU Unit |
| CJ2B-EIP21 | Built-in EtherNet/IP port in CJ2H CPU Unit |
| CJ2M-EIP21 | Built-in EtherNet/IP port in CJ2M CPU Unit |
| CS1W-EIP21 | CJ1W-EIP21 mounted to CS1 CPU Unit |
| CS1W-EIP21S | CS1W-EIP21S mounted to CS1 CPU Unit |
| NJ101<br>NJ301-□□□□<br>NJ501-□□□□ | Built-in EtherNet/IP port on the NJ-series CPU Unit |
| NX701 | Built-in EtherNet/IP port on the NX-series CPU Unit |

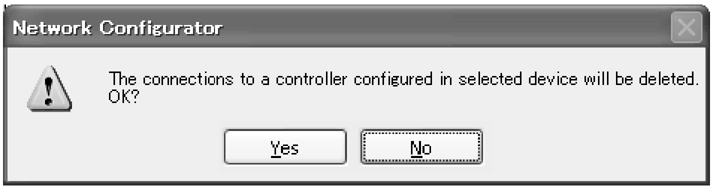
Use the following procedure to create connections (i.e., data links) with the Wizard.

**1** Set tags and tag sets for all devices before starting the Wizard. Refer to *7-2-4 Creating Tags and Tag Sets* for the setting procedure.

**2** For tag data links between OMRON PLCs, a connection is created in the PLC (i.e., the originator device) that receives data as input data.
First, select the registered device for which you want to create a connection in the Network Configuration Window of the Network Configurator, and then select ***Device - Parameters - Wizard*** from the menus.



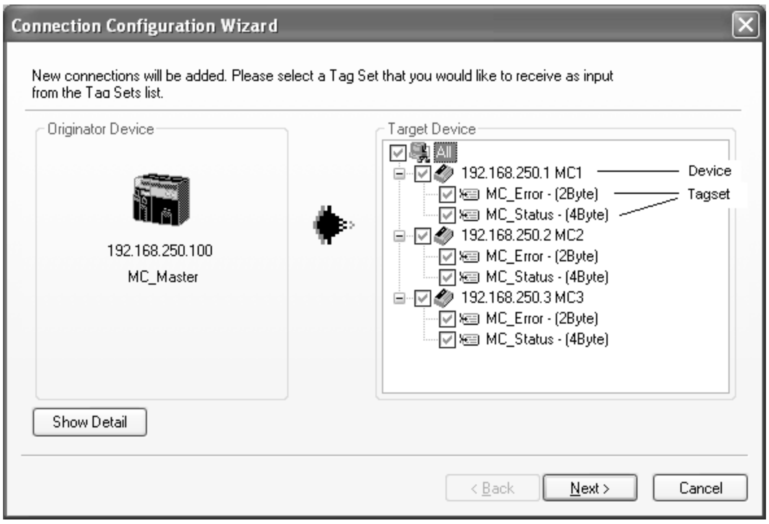
The following dialog box will be displayed before the Wizard starts.


Network Configurator

The connections to a controller configured in selected device will be deleted.
OK?

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

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



Connection Configuration Wizard

New connections will be added. Please select a Tag Set that you would like to receive as input from the Taq Sets list.

Originator Device

192.168.250.100
MC_Master

Target Device

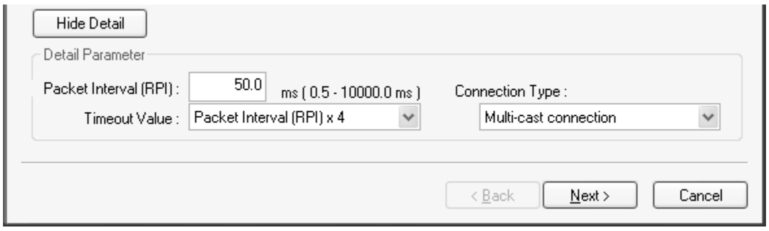
- ☑ All
  - ☑ 192.168.250.1 MC1
    - ☑ MC_Error - (2Byte)
    - ☑ MC_Status - (4Byte)
  - ☑ 192.168.250.2 MC2
    - ☑ MC_Error - (2Byte)
    - ☑ MC_Status - (4Byte)
  - ☑ 192.168.250.3 MC3
    - ☑ MC_Error - (2Byte)
    - ☑ MC_Status - (4Byte)

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

| Icon | Display position | Status |
|---|---|---|
| ☑ | All | All output tag sets for all devices are selected. |
| | Device | All output tag sets for the applicable device are selected. |
| | Tag set | The applicable output tag sets are selected. These are the tag sets that will be set in the connection. |
| ▣ | All | All or some output tag sets for some devices are selected. |
| | Device | Some output tag sets for applicable devices are selected. |
| □ | All | All output tag sets for all devices are not selected. |
| | Device | All output tag sets for applicable devices are not selected. |
| | Tag set | The applicable output tag sets are not selected. The connections for this tag set will be deleted. |
| ▨ | Device | No applicable tag sets. |

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

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

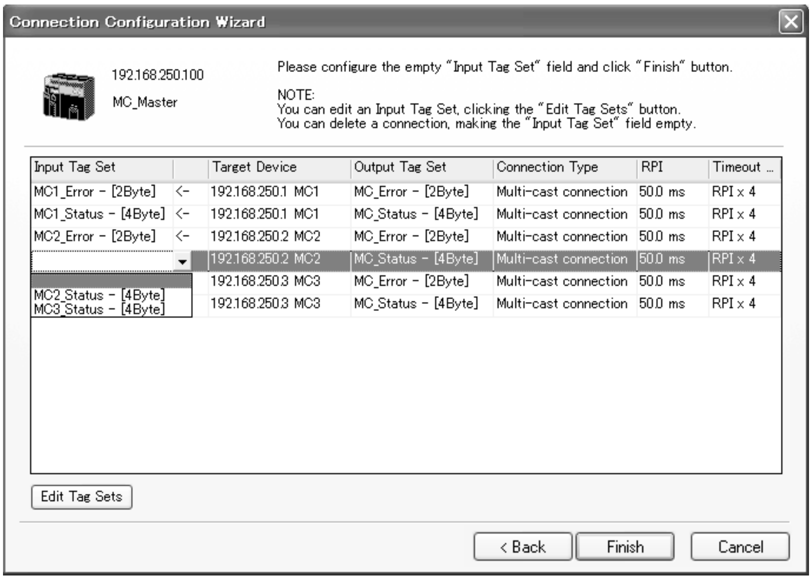


Detail Parameter

Packet Interval (RPI) : 50.0 ms ( 0.5 - 10000.0 ms )
Timeout Value : Packet Interval (RPI) x 4
Connection Type : Multi-cast connection

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

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



Connection Configuration Wizard

192.168.250.100

MC_Master

Please configure the empty "Input Tag Set" field and click "Finish" button.

NOTE:
You can edit an Input Tag Set, clicking the "Edit Tag Sets" button.
You can delete a connection, making the "Input Tag Set" field empty.

| Input Tag Set | | Target Device | Output Tag Set | Connection Type | RPI | Timeout ... |
|---|---|---|---|---|---|---|
| MC1_Error - [2Byte] | <- | 192.168.250.1 MC1 | MC_Error - [2Byte] | Multi-cast connection | 50.0 ms | RPI x 4 |
| MC1_Status - [4Byte] | <- | 192.168.250.1 MC1 | MC_Status - [4Byte] | Multi-cast connection | 50.0 ms | RPI x 4 |
| MC2_Error - [2Byte] | <- | 192.168.250.2 MC2 | MC_Error - [2Byte] | Multi-cast connection | 50.0 ms | RPI x 4 |
| | | 192.168.250.2 MC2 | MC_Status - [4Byte] | Multi-cast connection | 50.0 ms | RPI x 4 |
| | | 192.168.250.3 MC3 | MC_Error - [2Byte] | Multi-cast connection | 50.0 ms | RPI x 4 |
| | | 192.168.250.3 MC3 | MC_Status - [4Byte] | Multi-cast connection | 50.0 ms | RPI x 4 |

- The blank area in the Input Tag Set Column is the connection that you are creating.
- The rows in which there are input tag sets are connections that are already set.
- To prevent duplicate settings, input tag sets that have been used are not displayed in the list box for input tag sets.
- If there is no applicable input tag set, you can edit a tag set or create a new one by using the **Edit Tag Sets** Button and **Edit Tag** Button.

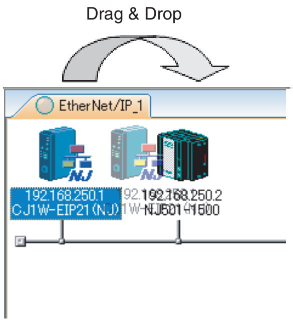
**6** Once the input tag set settings have been completed, click the **Finish** Button. You can check the set connection by selecting ***Network - View Devices Connection Structure Tree*** from the menus.

- The Wizard can be ended even if the input tag set includes a blank row. In that case, a connection is not created for the blank row.
- You can delete a connection by deleting the input tag sets that were previously set.

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

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

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



### Additional Information

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

| Device name | Remarks |
|---|---|
| CJ1W-EIP21(NJ) | CJ1W-EIP21 mounted to NJ-series CPU Unit |
| CJ1W-EIP21 | CJ1W-EIP21 mounted to CJ1 CPU Unit |
| CJ1W-EIP21(CJ2) | CJ1W-EIP21 mounted to CJ2 CPU Unit |
| CJ1W-EIP21S(NJ) | CJ1W-EIP21S mounted to NJ-series CPU Unit |
| CJ1W-EIP21S | CJ1W-EIP21S mounted to CJ1 CPU Unit |
| CJ1W-EIP21S(CJ2) | CJ1W-EIP21S mounted to CJ2 CPU Unit |
| CJ2B-EIP21 | Built-in EtherNet/IP port in CJ2H CPU Unit |
| CJ2M-EIP21 | Built-in EtherNet/IP port in CJ2M CPU Unit |
| CS1W-EIP21 | CJ1W-EIP21 mounted to CS1 CPU Unit |
| CS1W-EIP21S | CS1W-EIP21S mounted to CS1 CPU Unit |
| NJ101<br>NJ301-□□□□<br>NJ501-□□□□ | Built-in EtherNet/IP port in the NJ-series CPU Unit |
| NX701 | Built-in EtherNet/IP port on the NX-series CPU Unit |

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

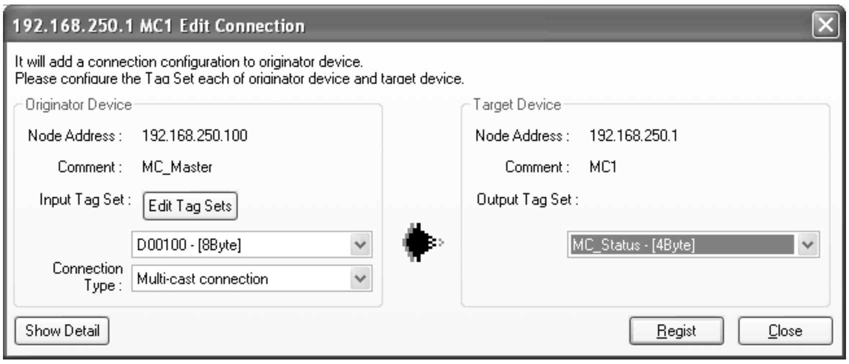
**1** Set the tags and tag sets for the target device that will be dragged.

**(1) Refer to *7-2-4 Creating Tags and Tag Sets* for information on creating the settings if the target is one of the OMRON EtherNet/IP devices given above.**

**(2) If the target is another EtherNet/IP device, refer to the manual of that device and perform settings as required.**

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

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



192.168.250.1 MC1 Edit Connection

It will add a connection configuration to originator device.
Please configure the Tag Set each of originator device and target device.

Originator Device

Node Address : 192.168.250.100

Comment : MC_Master

Input Tag Set : Edit Tag Sets

D00100 - [8Byte]

Connection Type : Multi-cast connection

Target Device

Node Address : 192.168.250.1

Comment : MC1

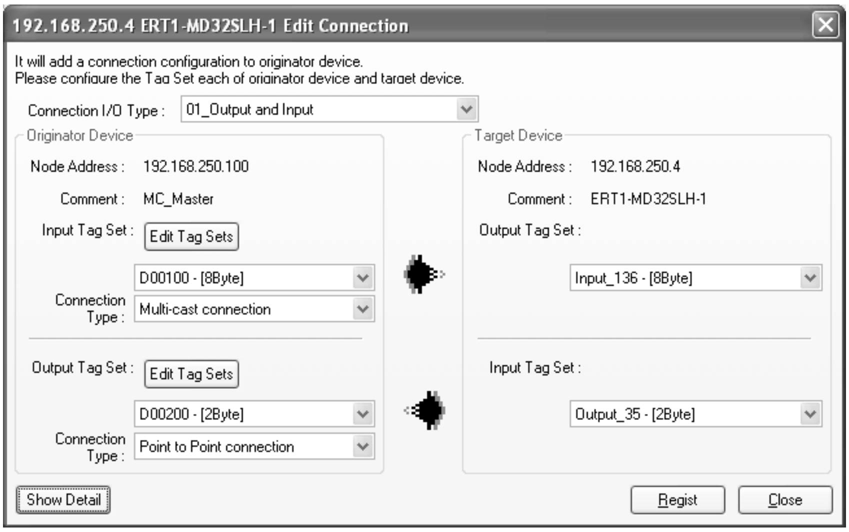
Output Tag Set :

MC_Status - [4Byte]

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

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

**(2) Using Other EtherNet/IP Devices as Target**



192.168.250.4 ERT1-MD32SLH-1 Edit Connection

It will add a connection configuration to originator device.
Please configure the Tag Set each of originator device and target device.

Connection I/O Type : 01_Output and Input

Originator Device

Node Address : 192.168.250.100

Comment : MC_Master

Input Tag Set : Edit Tag Sets

D00100 - [8Byte]

Connection Type : Multi-cast connection

Output Tag Set : Edit Tag Sets

D00200 - [2Byte]

Connection Type : Point to Point connection

Target Device

Node Address : 192.168.250.4

Comment : ERT1-MD32SLH-1

Output Tag Set :

Input_136 - [8Byte]

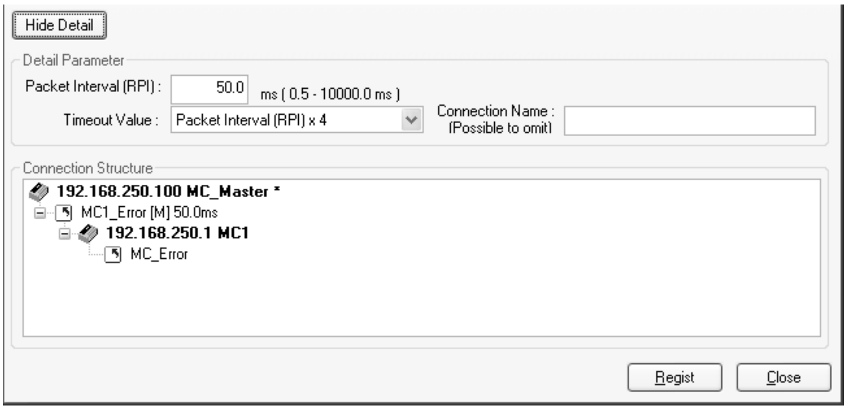
Input Tag Set :

Output_35 - [2Byte]

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

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

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



Packet Interval (RPI) : 50.0 ms ( 0.5 - 10000.0 ms )

Timeout Value : Packet Interval (RPI) x 4

Connection Name : (Possible to omit)

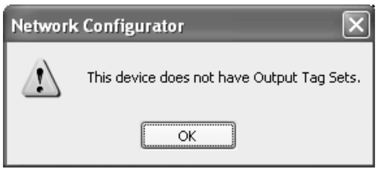
```
192.168.250.100 MC_Master *
  MC1_Error [M] 50.0ms
    192.168.250.1 MC1
      MC_Error
```

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

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

### Additional Information

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



This device does not have Output Tag Sets.

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

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

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

This section describes how to Connect the Network Configurator online.

### Connecting through Ethernet

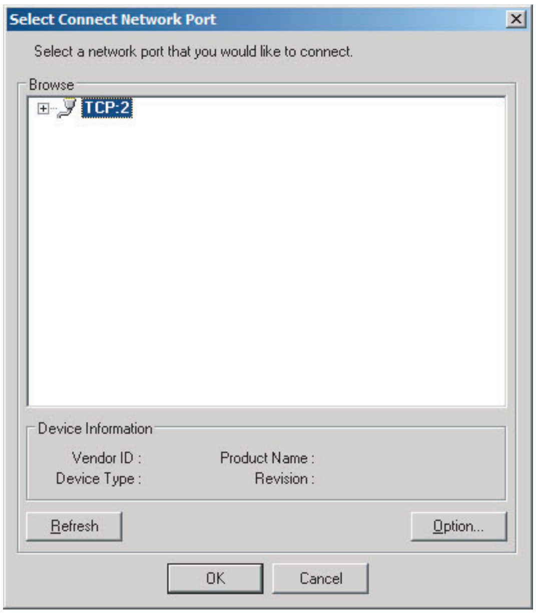
#### Precautions for Correct Use

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-4 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.



Select Connect Network Port

Select a network port that you would like to connect.

Browse

TCP:2

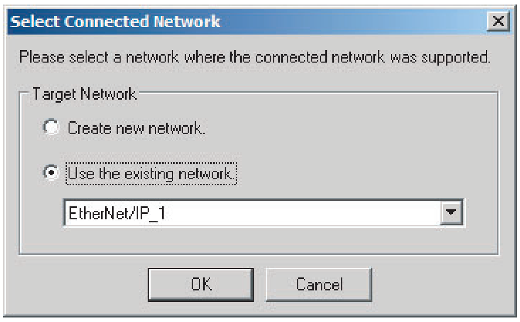
Device Information

Vendor ID :　　Product Name :

Device Type :　　Revision :

**3** Click the **OK** Button.

Select the network to connect.



Select Connected Network

Please select a network where the connected network was supported.

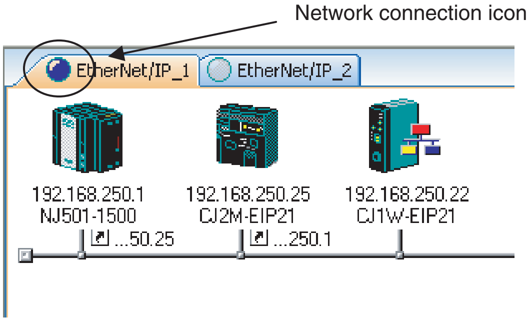
Target Network

○ Create new network.

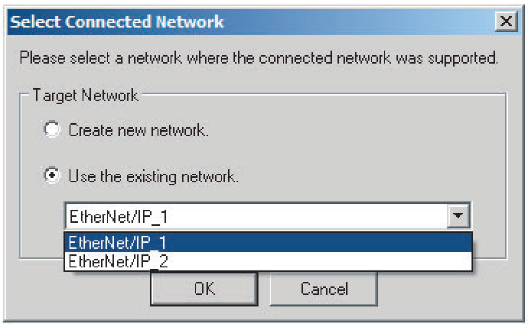
● Use the existing network.

EtherNet/IP_1

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.



Select Connected Network

Please select a network where the connected network was supported.

Target Network

○ Create new network.

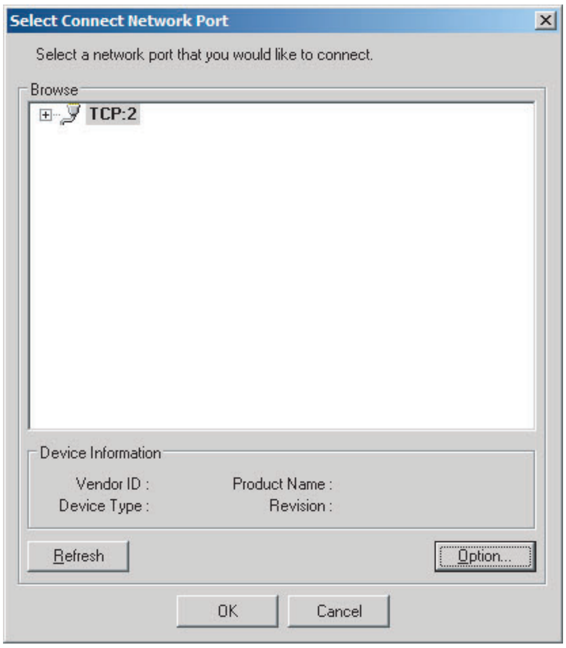
● Use the existing network.

EtherNet/IP_1

EtherNet/IP_1

EtherNet/IP_2

**4** The following dialog box is displayed.



Select Connect Network Port

Select a network port that you would like to connect.

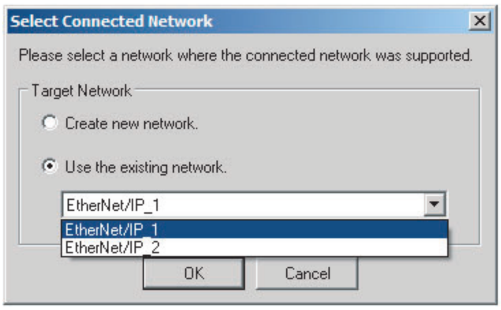
Browse

TCP:2

Device Information

Vendor ID :
Product Name :
Device Type :
Revision :

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



Select Connected Network

Please select a network where the connected network was supported.

Target Network

○ Create new network.

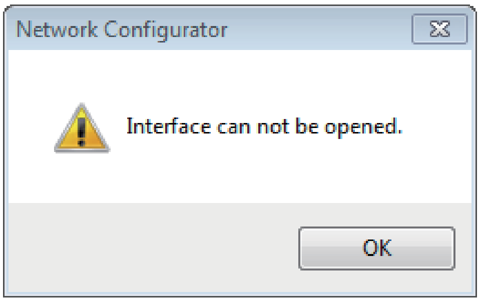
● Use the existing network.

EtherNet/IP_1

EtherNet/IP_1
EtherNet/IP_2

### Additional Information

If the following dialog box appears in the Network Configurator when you go online with an NJ/NX-series CPU Unit, refer to the following table for possible causes and corrections.



Network Configurator

Interface can not be opened.

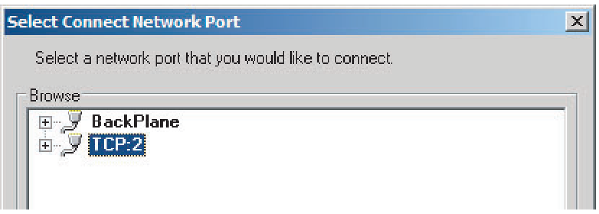
| Assumed cause | Correction |
|---|---|
| The cable is not connected correctly. | Check if the cable is disconnected or loose. |
| Connection with the Controller is blocked due to the firewall settings. | If connection with the Controller is blocked due to the firewall settings, disable the blocking.<br>For the firewall settings, refer to *A-4 Precautions for Using the Network Configurator on Windows XP, Windows Vista, or Windows 7 A-49*. |

## Connections through NJ-series CPU Unit’s USB Port

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

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

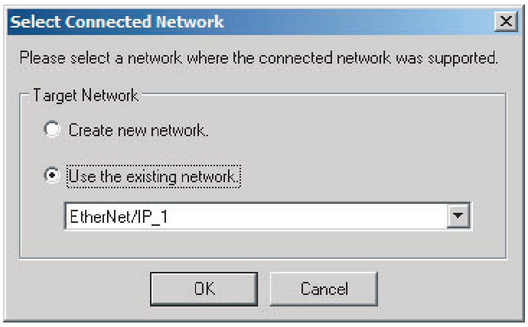
**2** Select *TCP:2* and then click the **OK** Button.



Select Connect Network Port

Select a network port that you would like to connect.

**3** Select the network to connect.



Select Connected Network

Please select a network where the connected network was supported.

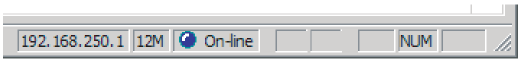
Target Network

○ Create new network.

● Use the existing network.

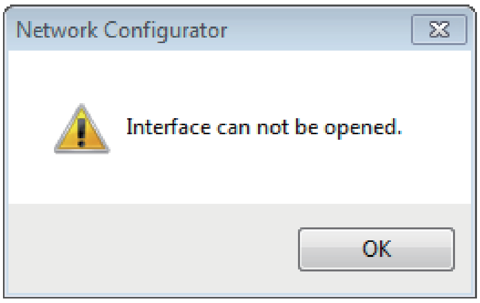
EtherNet/IP_1

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



### Additional Information

If the following dialog box appears in the Network Configurator when you go online with an NJ/NX-series CPU Unit, refer to the following table for possible causes and corrections.



Network Configurator

Interface can not be opened.

| Assumed cause | Correction |
|---|---|
| The cable is not connected correctly. | Check if the cable is disconnected or loose. |
| Connection with the Controller is blocked due to the firewall settings. | If connection with the Controller is blocked due to the firewall settings, disable the blocking.<br>For the firewall settings, refer to *A-4 Precautions for Using the Network Configurator on Windows XP, Windows Vista, or Windows 7 A-49*. |
| The USB driver is not installed correctly. | Install the USB driver correctly.<br>For how to install the USB driver, refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504). |

## 7-2-9 Downloading Tag Data Link Parameters

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

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

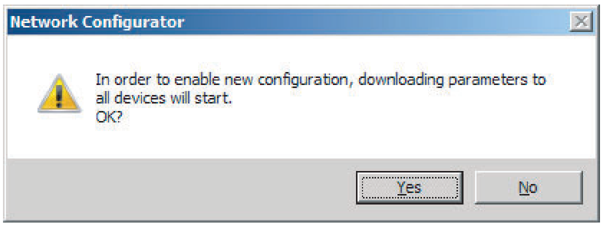
### Precautions for Correct Use

- If the node addresses (IP addresses) are not set correctly, you may connect to the wrong Controller and set incorrect device parameters. Download data only after you confirm that you are connected to the correct Controller.
- If incorrect tag data link parameters are set, it may cause equipment to operate unpredictably. Even when the correct tag data link parameters are set, make sure that there will be no effect on equipment before you transfer the data.
- When network variables are used in tag settings, a connection error will result if the variables are not also set in the 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 *15-1-1 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 communications port for the EtherNet/IP Unit is automatically restarted after the parameters are downloaded. This restart is required to enable the tag set and connection information. Before you download the parameters, check to confirm that problems will not occur with the equipment when the port is restarted.
- Make sure that the major CIP revision of the device registered with the Network Configurator is the same as the major CIP revision of the EtherNet/IP Unit that you use. If major CIP revisions are not the same, the parameters may not be downloaded. To determine whether downloading is possible, refer to *7-2-3 Registering Devices.*
- Do not disconnect the Ethernet cable 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 2 or later, the CPU Unit can download tag data link parameters in RUN mode. (They can also be downloaded in PROGRAM mode.) 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.

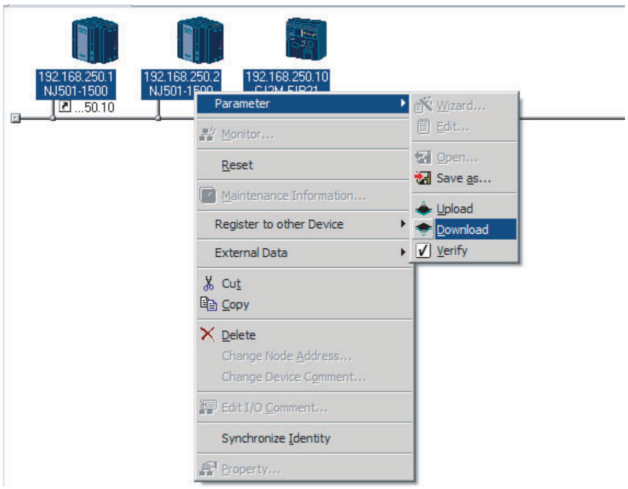


Network Configurator

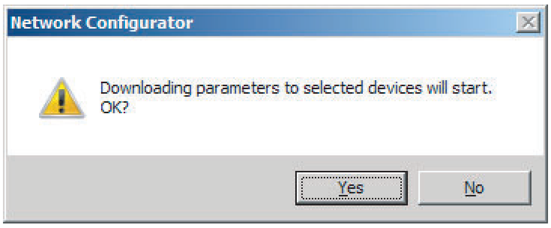
In order to enable new configuration, downloading parameters to all devices will start.
OK?

### Downloading Individually to Particular Devices

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



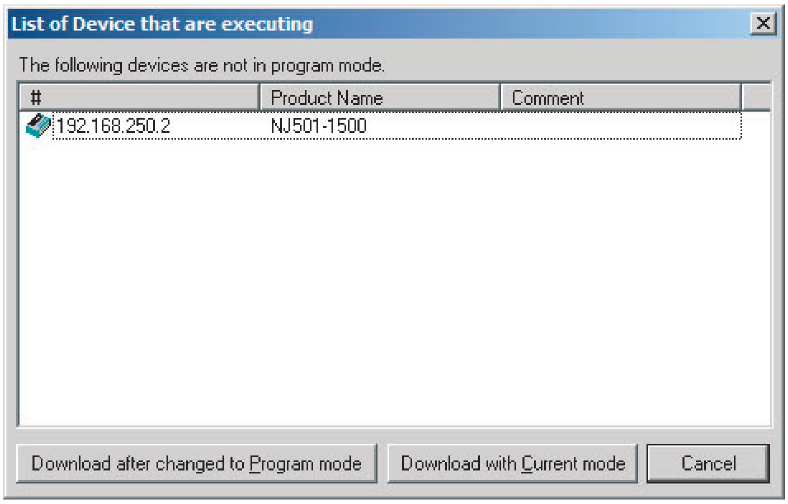
The following dialog box is displayed.



Network Configurator

Downloading parameters to selected devices will start.
OK?

**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.

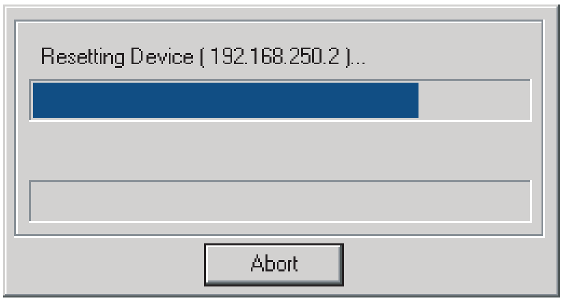

List of Device that are executing
Download after changed to Program mode
Download with Current mode
Cancel

The following devices are not in program mode.

| # | Product Name | Comment |
|---|---|---|
| 192.168.250.2 | NJ501-1500 | |

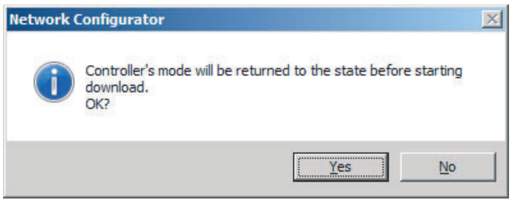
If the **Download after changed to Program mode** Button is clicked, all CPU Units are changed to PROGRAM mode and the parameters are downloaded. Confirm safety for all controlled equipment before you change the CPU Units to PROGRAM mode. You can restore the operating modes after the parameters are downloaded.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



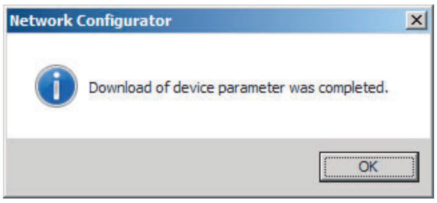
Resetting Device ( 192.168.250.2 )...

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.



Controller's mode will be returned to the state before starting download.
OK?

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



Download of device parameter was completed.

## 7-2-10 Uploading Tag Data Link Parameters

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

### Precautions for Correct Use

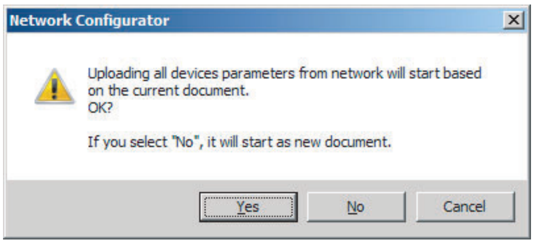
Make sure that the major CIP revision of the device registered with the Network Configurator is the same as the major CIP revision of the EtherNet/IP Unit that you use. If major CIP revisions are not the same, the parameters may not be uploaded. To determine whether uploading is possible, refer to *7-2-3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

There are two ways to upload the parameters.

**Uploading from All Devices in the Network**

**1** Connect the Network Configurator online, and then select ***Upload*** from the Network Menu.

**2** The following dialog box is displayed.



Network Configurator

Uploading all devices parameters from network will start based on the current document.
OK?

If you select "No", it will start as new document.

**Clicking the Yes Button:**

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

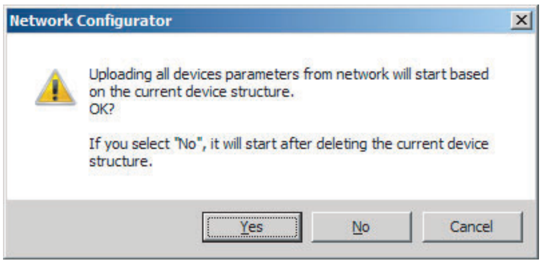
**Clicking the No Button:**

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.



Network Configurator

Uploading all devices parameters from network will start based on the current device structure.
OK?

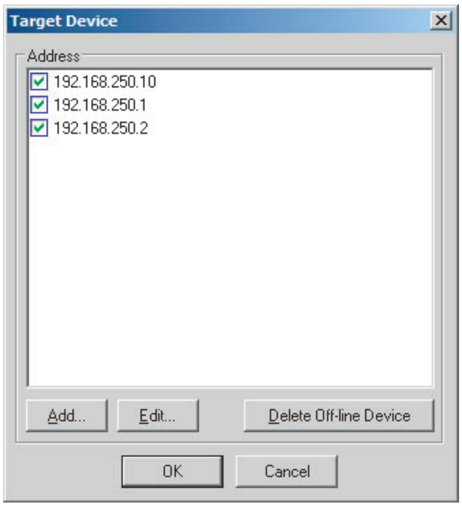
If you select "No", it will start after deleting the current device structure.

**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:**

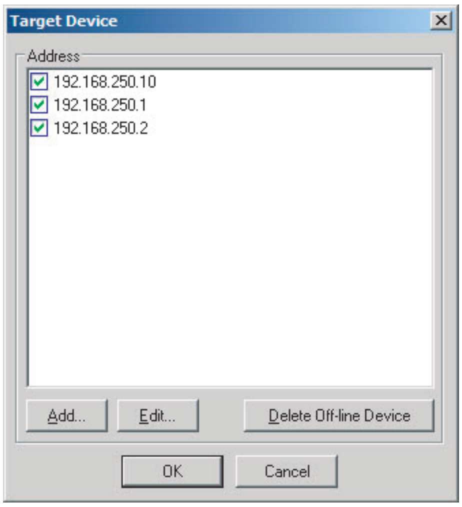
Performing a Batch Upload over the Network
Parameters are uploaded from all devices on the network. The current Network Configuration Information will be lost.
The following dialog box will be displayed. Select the devices for which to upload parameters and click the **OK** Button.



**Clicking the Cancel Button:**

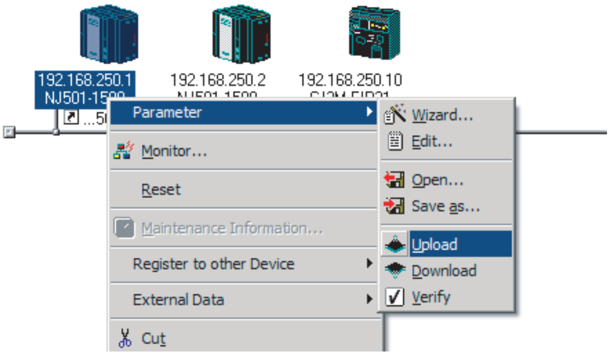
The upload operation is cancelled. The upload is not performed.

**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.

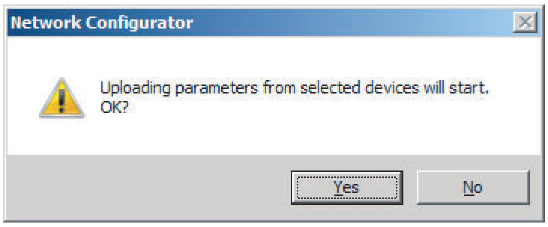


### Uploading Individually from Particular Devices

**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, hold down the **Shift** Key or the **Ctrl** Key while you click the icons. (In the following example, 2 nodes are selected: 192.168.250.1 and 192.168.250.2.)
Right-click the icon to display the pop-up menu, and select ***Parameter*** – ***Upload***.



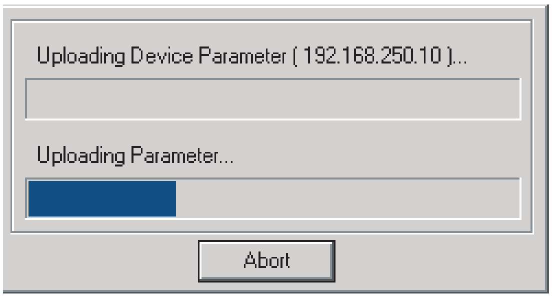
**2** The following dialog box is displayed.



Uploading parameters from selected devices will start.
OK?

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

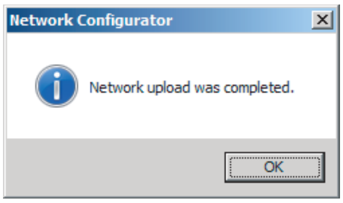
**3** During the upload, the following progress monitor is displayed to show the progress of the upload.



Uploading Device Parameter ( 192.168.250.10 )...

Uploading Parameter...

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



Network upload was completed.

## 7-2-11 Verifying the Tag Data Links

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

**Precautions for Correct Use**

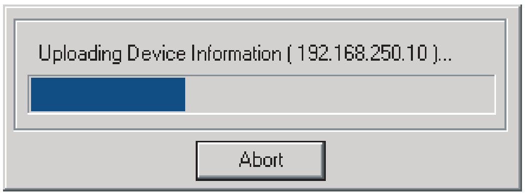
Make sure that the major CIP revision of the device registered with the Network Configurator is the same as the major CIP revision of the EtherNet/IP Unit CPU Unit that you use. If the major CIP revisions are not the same, the parameters may not be compared. To determine whether comparison is possible,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

### Verifying the Network Configuration

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

**1** Connect the Network Configurator online.

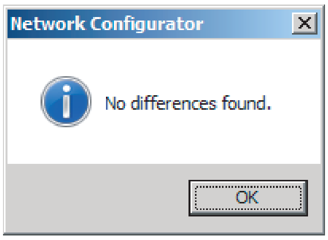
**2** Select ***Network*** – ***Verify Structure***.
The following progress monitor is displayed to show the progress as data is read from the network and compared.



Uploading Device Information ( 192.168.250.10 )...

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

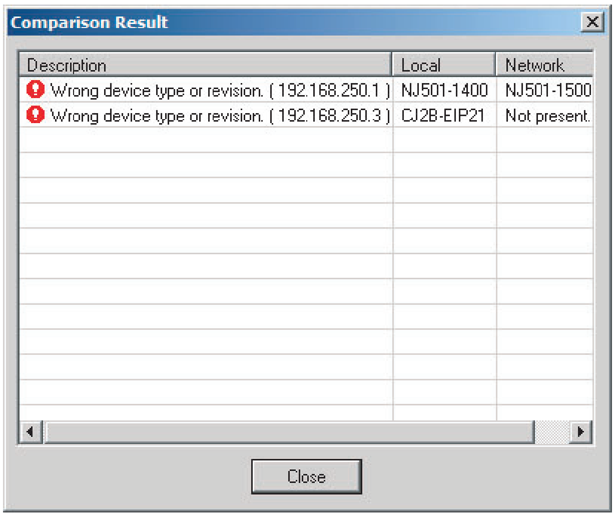
**Differences Not Found in the Comparison**



Network Configurator

No differences found.

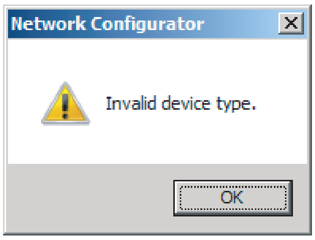
**Differences Found in the Comparison**



Comparison Result

| Description | Local | Network |
|---|---|---|
| Wrong device type or revision. ( 192.168.250.1 ) | NJ501-1400 | NJ501-1500 |
| Wrong device type or revision. ( 192.168.250.3 ) | CJ2B-EIP21 | Not present. |

**Differences Found in the Device Type**



Network Configurator

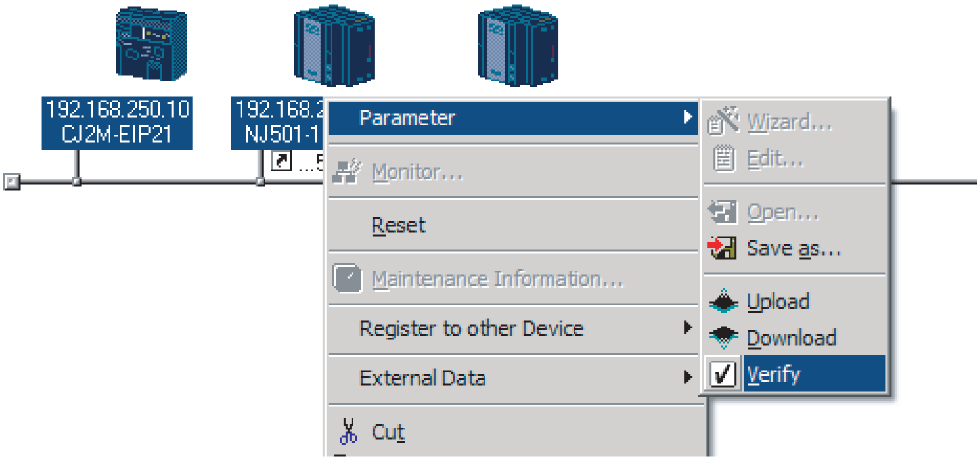
Invalid device type.

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

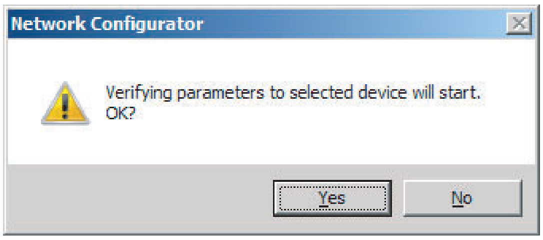
## Verifying the Device Parameters

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



**3** The following dialog box is displayed.



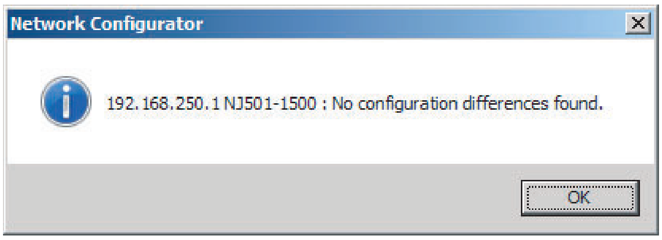
Network Configurator

Verifying parameters to selected device will start.
OK?

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

**4** The following dialog box is displayed.

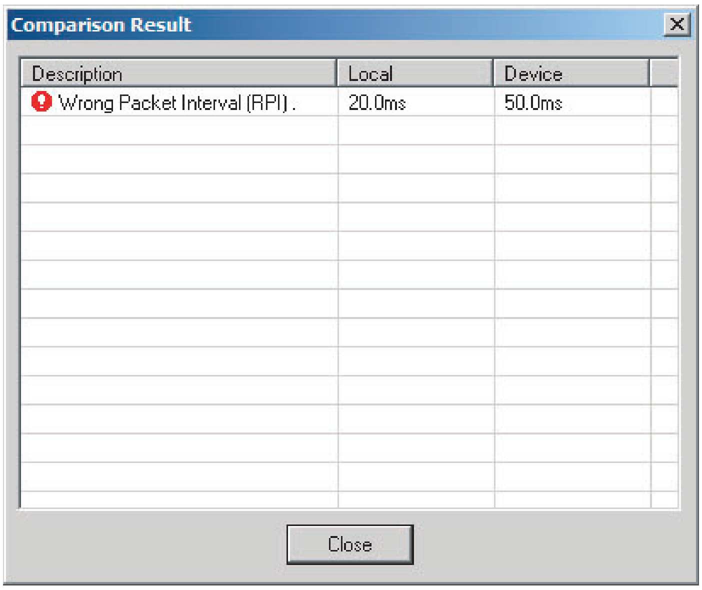
**Differences Not Found in the Comparison**



Network Configurator

192.168.250.1 NJ501-1500 : No configuration differences found.

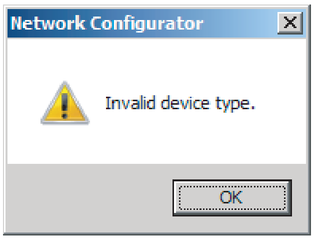
**Differences Found in the Comparison**



Comparison Result

| Description | Local | Device |
|---|---|---|
| Wrong Packet Interval (RPI) . | 20.0ms | 50.0ms |

**Differences Found in the Device Type**

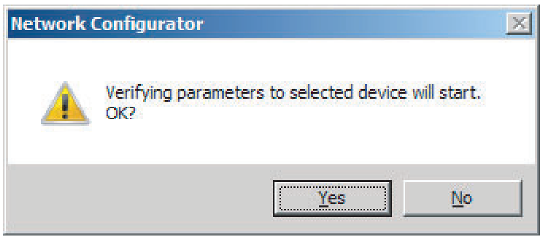


Network Configurator

Invalid 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.



Network Configurator

Verifying parameters to selected device will start. OK?

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

## 7-2-12 Starting and Stopping Tag Data Links

### Automatically Starting Tag Data Links

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

### Starting and Stopping Tag Data Links for the Entire Network

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

**Precautions for Correct Use**

Use the same method (i.e., either the user program or the Network Configurator) to both start and stop tag data links. For example, if you use the *_*TDLStopCmd* (Tag Data Link Stop Bit) device variable for the CJ-series Unit to stop tag data links, you cannot start them from the Network Configurator.

#### ● Using Commands in the User Program

You can change the corresponding elements in the following the device variables for the CJ-series Unit to TRUE in the user program to start and stop tag data links for individual devices. (Refer to *Section 3 Assigning Device Variables for CJ-series Units.*)

- *_TDLStartCmd (Tag Data Link Start Bit)
- *_TDLStopCm (Tag Data Link Stop Bit)

**Additional Information**

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

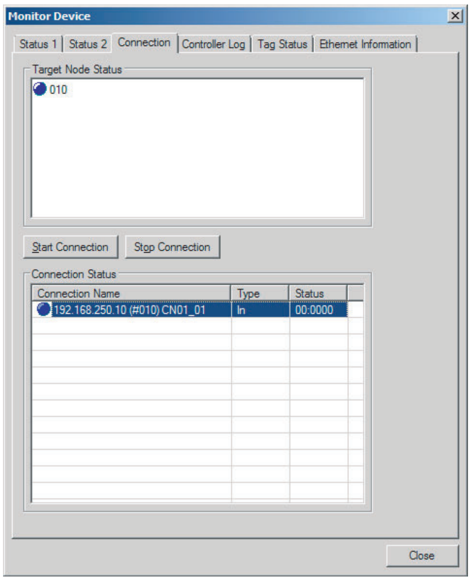
#### ● Using the Network Configurator

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

## Starting and Stopping Tag Data Links for Individual Devices

### ● Using the Network Configurator

You can start and stop tag data links for individual devices using the Connection Tab Page in the Monitor Device Dialog Box. This applies only to tag data links for which the device is the originator. Select ***Monitor*** from the Device Menu to access the Monitor Device Dialog Box.



Target Node Status

010

Connection Status

| Connection Name | Type | Status |
|---|---|---|
| 192.168.250.10 (#010) CN01_01 | In | 00:0000 |

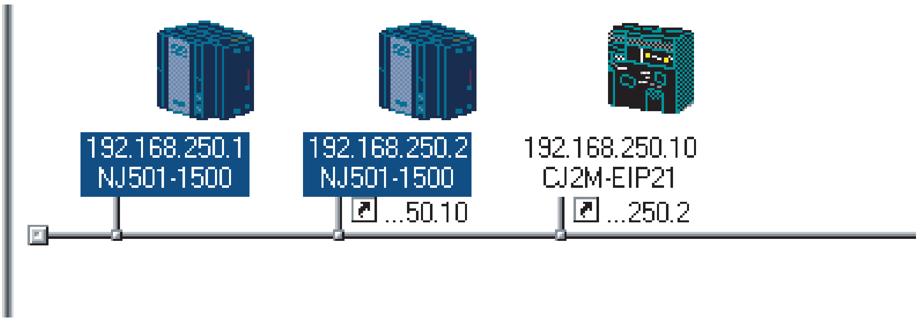
**Start Connection Button:**
Starts all connections for which the device is the originator.

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

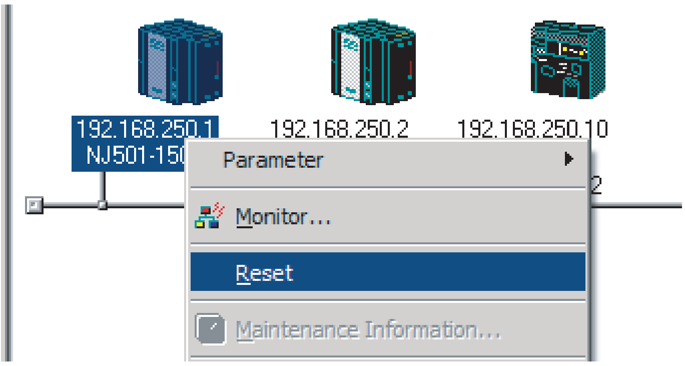
## 7-2-13 Clearing the Device Parameters

You can clear the tag data link parameters that are saved in the EtherNet/IP Unit 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 *7-2-8 Connecting the Network Configurator to the Network*.

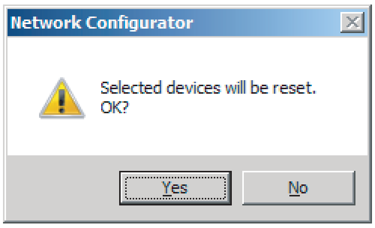
1. Connect the Network Configurator online.
2. Select the icon of the EtherNet/IP Unit from which you want to clear the device parameters. In the following example, 2 nodes are selected: 192.168.250.1 and 192.168.250.2. To select multiple nodes, press and hold the **Shift** Key while you select additional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



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



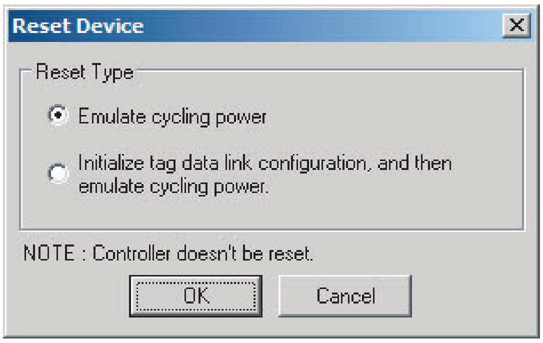
**4** The following dialog box is displayed.


Network Configurator

Selected devices will be reset.
OK?

**Yes Button:**

The following dialog box is displayed.


Reset Device

Reset Type

● Emulate cycling power

○ Initialize tag data link configuration, and then emulate cycling power.

NOTE : Controller doesn't be reset.

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 EtherNet/IP Unit is restarted.

**No Button:**

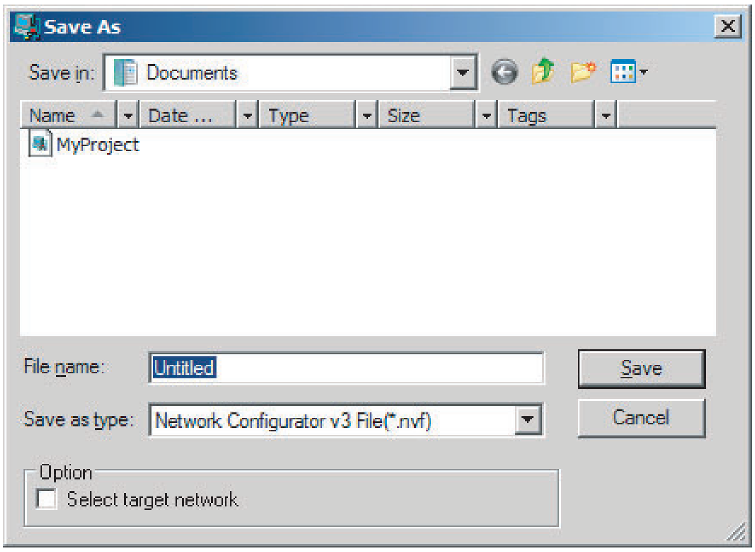
The device parameters are not cleared or reset.

## 7-2-14 Saving the Network Configuration File

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

**1** Select ***File*** – ***Save As***.

The following dialog box is displayed.



Save in: Documents

File name: Untitled

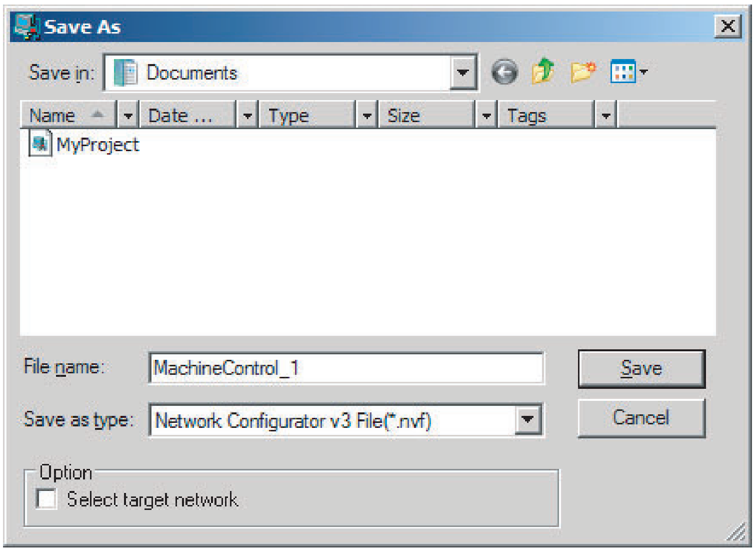
Save as type: Network Configurator v3 File(*.nvf)

Option

□ Select target network

Untitled.nvf is displayed as the default file name.

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



Save in: Documents

File name: MachineControl_1

Save as type: Network Configurator v3 File(*.nvf)

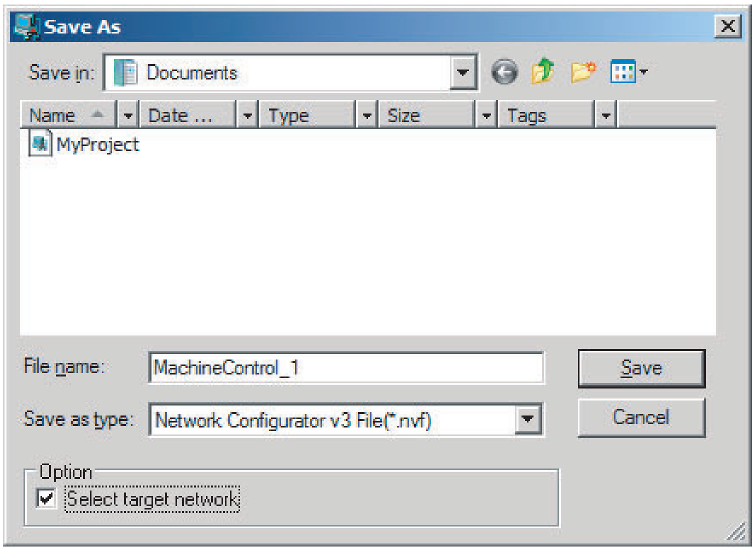
Option

□ Select target network

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.



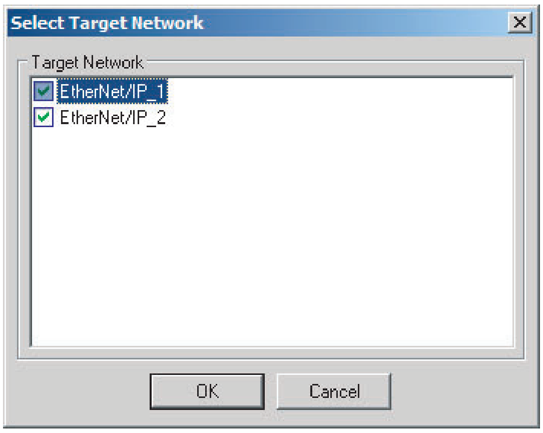
Save in: Documents

File name: MachineControl_1

Save as type: Network Configurator v3 File(*.nvf)

☑ Select target network

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



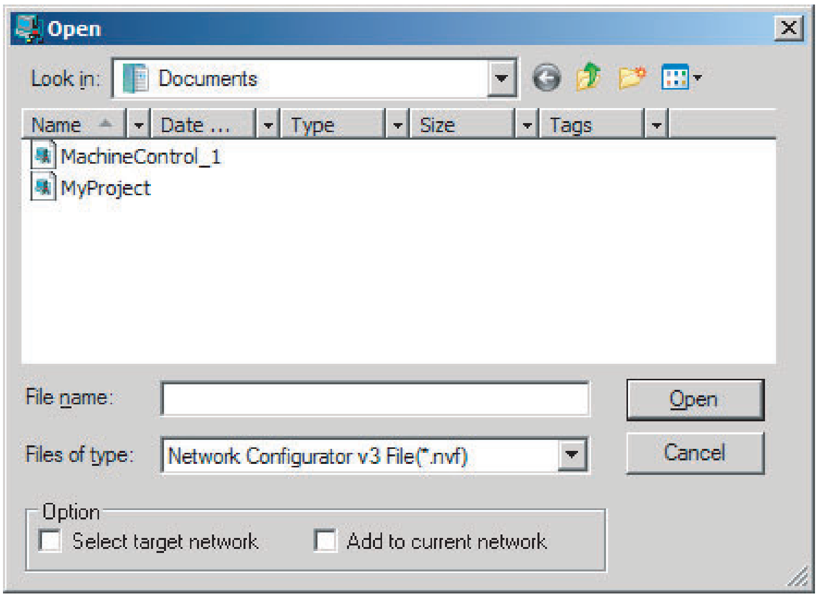
☑ EtherNet/IP_1

☑ EtherNet/IP_2

## 7-2-15 Reading a Network Configuration File

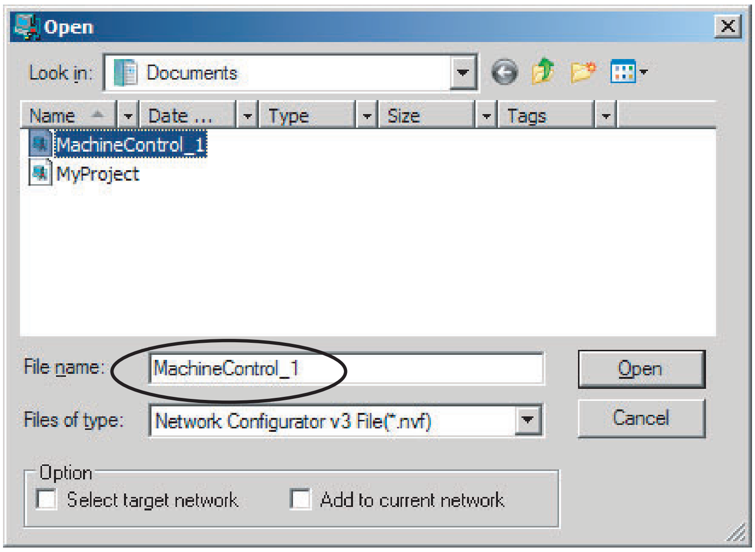
You can read a previously saved network configuration file into the Network Configurator.

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



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

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

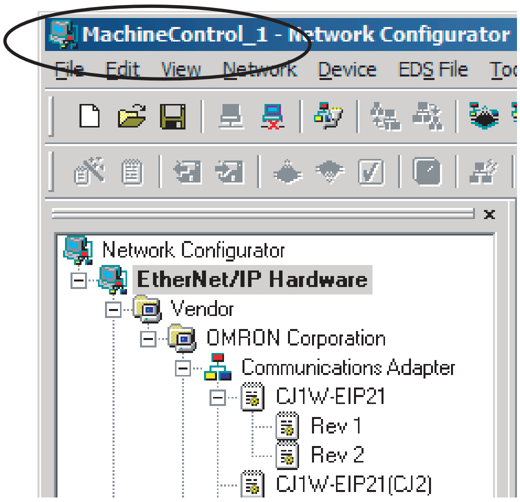


File name: MachineControl_1

Files of type: Network Configurator v3 File(*.nvf)

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

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



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

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

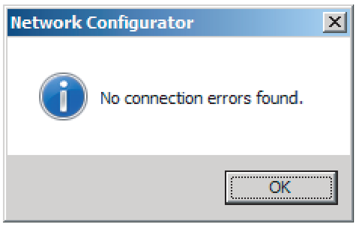
### Additional Information

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

## 7-2-16 Checking Connections

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

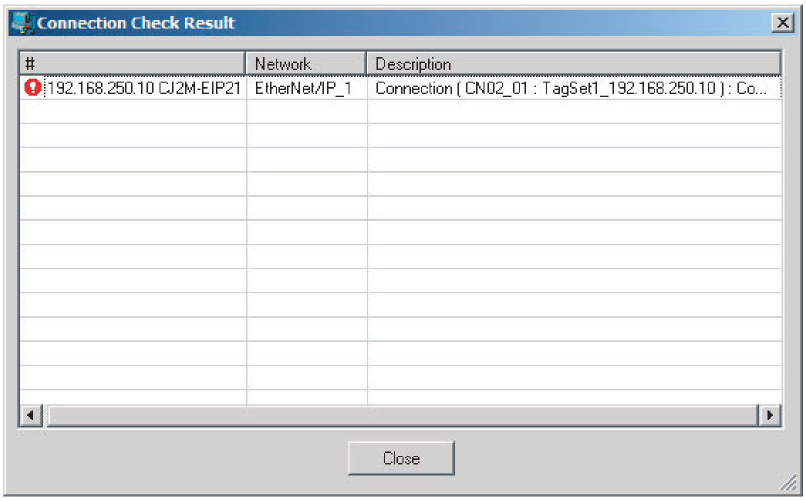
**1** Select ***Check Connection*** from the Network Menu.
The following dialog box is displayed if parameters are normal.



Network Configurator

No connection errors found.

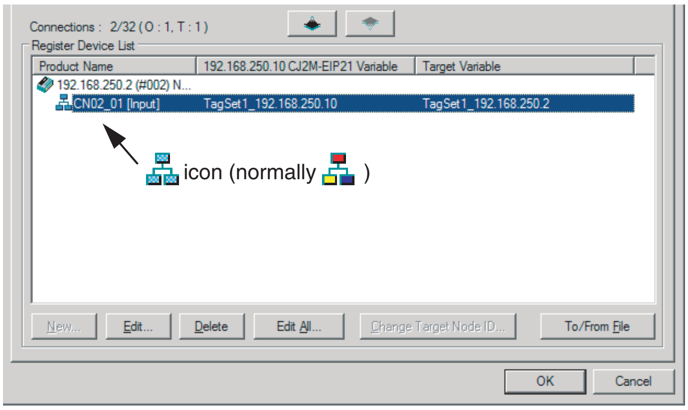
The following dialog box is displayed if there are parameter errors. Check the displayed details and review the settings.



Connection Check Result

| # | Network | Description |
|---|---|---|
| 192.168.250.10 CJ2M-EIP21 | EtherNet/IP_1 | Connection ( CN02_01 : TagSet1_192.168.250.10 ) : Co... |

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.



Connections : 2/32 ( O : 1, T : 1 )

Register Device List

| Product Name | 192.168.250.10 CJ2M-EIP21 Variable | Target Variable |
|---|---|---|
| 192.168.250.2 (#002) N... | | |
| CN02_01 [Input] | TagSet1_192.168.250.10 | TagSet1_192.168.250.2 |

## 7-2-17 Changing Devices

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

### Device Changes

| **Model after change** | | **CS1W -EIP21** | **CJ1W-EIP21** | **CS1W -EIP21** | **CJ1W-EIP21** | **CJ1W-EIP21 (CJ2)** | **CJ2B-EIP21** | **CJ1W-EIP21 (NJ)** | **CJ2M** | **CS1W -EIP21** | **CJ1W-EIP21** | **CJ1W-EIP21 (CJ2)** | **CJ2B-EIP21** | **CJ1W-EIP21 (NJ)** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model before change** | **Rev** | **1** | **1** | **2** | **2** | **2** | **2** | **2** | **2** | **3** | **3** | **3** | **3** | **3** |
| **CS1W-EIP21** | **1** | | Yes | Yes | Yes | Yes | Yes | Yes | *3 | Yes | Yes | Yes | Yes | Yes |
| **CJ1W-EIP21** | **1** | Yes | | Yes | Yes | Yes | Yes | Yes | *3 | Yes | Yes | Yes | Yes | Yes |
| **CS1W-EIP21** | **2** | No | No | | Yes | Yes | Yes | *5 | *3 | Yes | Yes | Yes | Yes | *5 |
| **CJ1W-EIP21** | **2** | No | No | Yes | | Yes | Yes | *5 | *3 | Yes | Yes | Yes | Yes | *5 |
| **CJ1W-EIP21(CJ2)** | **2** | No | No | *1 | *1 | | Yes | *5 | *3 | *1 | *1 | Yes | Yes | *5 |
| **CJ2B-EIP21** | **2** | No | No | *1 | *1 | Yes | | *5 | *3 | *1 | *1 | Yes | Yes | *5 |
| **CJ1W-EIP21(NJ)** | **2** | No | No | *1<br>*2 | *1<br>*2 | *2 | *2 | | *2<br>*6 | *1<br>*2 | *1<br>*2 | *2 | *2 | Yes |
| **CJ2M** | **2** | No | No | *1 | *1 | Yes | Yes | *5 | | *1 | *1 | Yes | Yes | *5 |
| **CS1W-EIP21** | **3** | No | No | Yes | Yes | Yes | Yes | *5 | *3 | | Yes | Yes | Yes | *5 |
| **CJ1W-EIP21** | **3** | No | No | Yes | Yes | Yes | Yes | *5 | *3 | Yes | | Yes | Yes | *5 |
| **CJ1W-EIP21(CJ2)** | **3** | No | No | *1 | *1 | Yes | Yes | *5 | *3 | *1 | *1 | | Yes | *5 |
| **CJ2B-EIP21** | **3** | No | No | *1 | *1 | Yes | Yes | *5 | *3 | *1 | *1 | Yes | | *5 |
| **CJ1W-EIP21(NJ)** | **3** | No | No | *1<br>*2 | *1<br>*2 | *2 | *2 | Yes | *2<br>*6 | *1<br>*2 | *1<br>*2 | *2 | *2 | |
| **CS1W-EIP21S** | **4** | No | No | Yes | Yes | Yes | Yes | *5 | *3 | Yes | Yes | Yes | Yes | *5 |
| **CJ1W-EIP21S** | **4** | No | No | Yes | Yes | Yes | Yes | *5 | *3 | Yes | Yes | Yes | Yes | *5 |
| **CJ1W-EIP21S (CJ2)** | **4** | No | No | *1 | *1 | Yes | Yes | *5 | *3 | *1 | *1 | Yes | Yes | *5 |
| **CJ1W-EIP21S (NJ)** | **4** | No | No | *1<br>*2 | *1<br>*2 | *2 | *2 | Yes | *2<br>*6 | *1<br>*2 | *1<br>*2 | *2 | *2 | Yes |
| **NJ501-□□□□<br>NJ301-□□□□<br>NJ101** | **1** | No | No | *1<br>*2 | *1<br>*2 | *2 | *2 | Yes | *2<br>*6 | *1<br>*2 | *1<br>*2 | *2 | *2 | Yes |
| | **2** | No | No | *1<br>*2 | *1<br>*2 | *2 | *2 | Yes | *2<br>*6 | *1<br>*2 | *1<br>*2 | *2 | *2 | Yes |
| **NX701** | **2** | No | No | No | No | *2 | *2 | Yes | *2<br>*6 | No | No | *2 | *2 | Yes |
| **NX1P2** | **2** | No | No | No | No | *2 | *2 | Yes | *2<br>*6 | No | No | *2 | *2 | Yes |
| **NX102** | **2** | No | No | No | No | *2 | *2 | Yes | *2<br>*6 | No | No | *2 | *2 | Yes |
| **NY512/532** | **2** | No | No | No | No | No | No | Yes | No | No | No | No | No | Yes |
| **NX-CSG** | **2** | No | No | No | No | No | No | No | No | No | No | No | No | No |
| **NX502** | **2** | No | No | No | No | *2<br>*7 | *2<br>*7 | *2<br>*7 | *2<br>*6<br>*7 | No | No | *2<br>*7 | *2<br>*7 | *2<br>*7 |

| Model after change | | CS1W-EIP21 | CJ1W-EIP21 | CS1W-EIP21 | CJ1W-EIP21 | CJ1W-EIP21 (CJ2) | CJ2B-EIP21 | CJ1W-EIP21 (NJ) | CJ2M | CS1W-EIP21 | CJ1W-EIP21 | CJ1W-EIP21 (CJ2) | CJ2B-EIP21 | CJ1W-EIP21 (NJ) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model before change** | **Rev** | **1** | **1** | **2** | **2** | **2** | **2** | **2** | **2** | **3** | **3** | **3** | **3** | **3** |
| **NX-EIP201** | **2** | No | No | No | No | *2<br>*6<br>*7 | *2<br>*6<br>*7 | *2<br>*6<br>*7 | *2<br>*4<br>*7 | No | No | *2<br>*6<br>*7 | *2<br>*6<br>*7 | *2<br>*4<br>*7 |

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 the maximum size of a tag name or tag set name (size after conversion into UTF-8) exceeds 48 bytes.

*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 the physical addresses are allocated to one of the tags and the tag size is an odd number of bytes.

*6 Cannot be changed if any of tags, tag sets, and refreshing sizes exceeds the permissible settings. (A refreshing size refers to a total size of multiple tag sets that a single node exchanges. There are limits to the permissible refresh size settings.)

*7 Cannot be changed if the maximum number of tags per tag set exceeds the permissible setting.

(Continued)

| Model after change | | CS1W-EIP21S | CJ1W-EIP21S | CJ1W-EIP21S (CJ2) | CJ1W-EIP21S (NJ) | NJ501-□□□□<br>NJ301-□□□□<br>NJ101 | | NX701 | NX1P2 | NX102 | NY512/532 | NX-CSG | NX502 | NX-EIP201 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model before change** | **Rev** | **4** | **4** | **4** | **4** | **1** | **2** | **2** | **2** | **2** | **2** | **2** | **2** | **2** |
| **CS1W-EIP21** | **1** | Yes | Yes | Yes | Yes | *4 | *4 | No | No | No | No | No | No | No |
| **CJ1W-EIP21** | **1** | Yes | Yes | Yes | Yes | *4 | *4 | No | No | No | No | No | No | No |
| **CS1W-EIP21** | **2** | Yes | Yes | Yes | *5 | *4<br>*5 | *4<br>*5 | No | No | No | No | No | No | No |
| **CJ1W-EIP21** | **2** | Yes | Yes | Yes | *5 | *4<br>*5 | *4<br>*5 | No | No | No | No | No | No | No |
| **CJ1W-EIP21 (CJ2)** | **2** | *1 | *1 | Yes | *5 | *4<br>*5 | *4<br>*5 | *5 | *4<br>*5 | *4<br>*5 | No | No | *4<br>*5 | *5 |
| **CJ2B-EIP21** | **2** | *1 | *1 | Yes | *5 | *4<br>*5 | *4<br>*5 | *5 | *4<br>*5 | *4<br>*5 | No | No | *4<br>*5 | *5 |
| **CJ1W-EIP21 (NJ)** | **2** | *1<br>*2 | *1<br>*2 | *2 | Yes | *4 | *4 | Yes | *4 | *4 | *4 | No | *4 | Yes |
| **CJ2M** | **2** | *1 | *1 | Yes | *5 | *4<br>*5 | *4<br>*5 | *5 | *4<br>*5 | *4<br>*5 | No | No | *5 | *5 |
| **CS1W-EIP21** | **3** | Yes | Yes | Yes | *5 | *4<br>*5 | *4<br>*5 | No | No | No | No | No | No | No |
| **CJ1W-EIP21** | **3** | Yes | Yes | Yes | *5 | *4<br>*5 | *4<br>*5 | No | No | No | No | No | No | No |
| **CJ1W-EIP21 (CJ2)** | **3** | *1 | *1 | Yes | *5 | *4<br>*5 | *4<br>*5 | *5 | *4<br>*5 | *4<br>*5 | No | No | *4<br>*5 | *5 |
| **CJ2B-EIP21** | **3** | *1 | *1 | Yes | *5 | *4<br>*5 | *4<br>*5 | *5 | *4<br>*5 | *4<br>*5 | No | No | *4<br>*5 | *5 |
| **CJ1W-EIP21 (NJ)** | **3** | *1<br>*2 | *1<br>*2 | *2 | Yes | *4 | *4 | Yes | *4 | *4 | *4 | No | *4 | Yes |

| Model after change | | CS1W-EIP21S | CJ1W-EIP21S | CJ1W-EIP21S (CJ2) | CJ1W-EIP21S (NJ) | NJ501-□□□□ NJ301-□□□□ NJ101 | | NX701 | NX1P2 | NX102 | NY512/532 | NX-CSG | NX502 | NX-EIP201 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model before change** | **Rev** | **4** | **4** | **4** | **4** | **1** | **2** | **2** | **2** | **2** | **2** | **2** | **2** | **2** |
| **CS1W-EIP21S** | **4** | | Yes | Yes | *5 | *4<br>*5 | *4<br>*5 | No | No | No | No | No | No | No |
| **CJ1W-EIP21S** | **4** | Yes | | Yes | *5 | *4<br>*5 | *4<br>*5 | No | No | No | No | No | No | No |
| **CJ1W-EIP21S (CJ2)** | **4** | *1 | *1 | | *5 | *4<br>*5 | *4<br>*5 | *5 | *4<br>*5 | *4<br>*5 | No | No | *4<br>*5 | *5 |
| **CJ1W-EIP21S (NJ)** | **4** | *1<br>*2 | *1<br>*2 | *2 | | *4 | *4 | Yes | *4 | *4 | *4 | No | *4 | Yes |
| **NJ501-□□□□ NJ301-□□□□ NJ101** | **1** | *1<br>*2 | *1<br>*2 | *2 | Yes | | Yes | No | No | No | No | No | No | No |
| | **2** | *1<br>*2 | *1<br>*2 | *2 | Yes | Yes | | Yes | *4 | Yes | Yes | No | Yes | Yes |
| **NX701** | **2** | No | No | *2 | Yes | No | *4 | | *4 | *4 | *4 | No | *4 | Yes |
| **NX1P2** | **2** | No | No | *2 | Yes | No | Yes | Yes | | Yes | *4 | No | Yes | Yes |
| **NX102** | **2** | No | No | *2 | Yes | No | Yes | Yes | *4 | | *4 | No | Yes | Yes |
| **NY512/532** | **2** | No | No | No | Yes | No | *4 | Yes | Yes | Yes | | No | *4 | Yes |
| **NX-CSG** | **2** | No | No | No | No | No | No | No | No | No | No | | No | No |
| **NX502** | **2** | No | No | *2<br>*7 | *2<br>*7 | No | *4<br>*7 | *7 | *4<br>*7 | *4<br>*7 | *7 | No | | Yes |
| **NX-EIP201** | **2** | No | No | *2<br>*6<br>*7 | *2<br>*4<br>*7 | No | *4<br>*7 | *6<br>*7 | *4<br>*7 | *4<br>*7 | *4<br>*7 | No | *4 | |

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 the maximum size of a tag name or tag set name (size after conversion into UTF-8) exceeds 48 bytes.

*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).

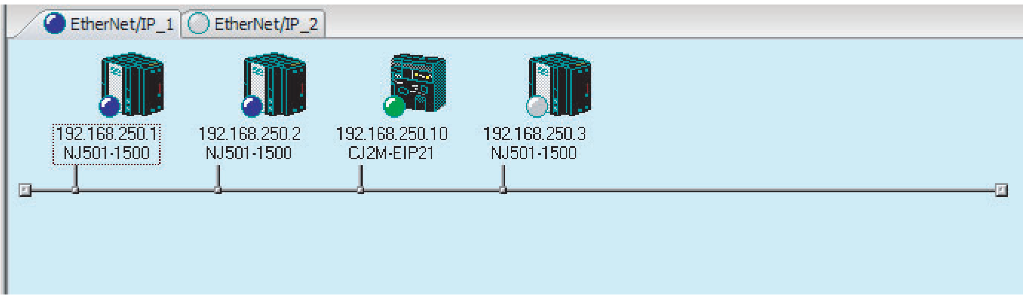
*5 Cannot be changed if the physical addresses are allocated to one of the tags and the tag size is an odd number of bytes.

*6 Cannot be changed if any of tags, tag sets, and refreshing sizes exceeds the permissible settings. (A refreshing size refers to a total size of multiple tag sets that a single node exchanges. There are limits to the permissible refresh size settings.)

*7 Cannot be changed if the maximum number of tags per tag set exceeds the permissible setting.

## 7-2-18 Displaying Device Status

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



| Icon | Status |
|---|---|
| ○<br>(white) | Offline |
| ○<br>(gray) | Default (including no Controller Configurations and Setup) |
| ●<br>(green) | Idle (including when CPU Unit of Controller is in PROGRAM mode) |
| ●<br>(blue) | Normal communications state (including when Controller is in RUN mode) |
| ●<br>(yellow) | Warning status (including when there is a partial fault or non-fatal error) |
| ●<br>(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 PLC information in the input conditions, add conditions 4 and 5.

### ● Conditions to enable the NJ-series EtherNet/IP Unit’s tag data links:

**(1) The following error bits in the Unit Error Occurred (*_*UnitErr*) are FALSE.**

**(2) Also, the Online variable (*_*ETNOnlineSta*) is TRUE.**

### ● Conditions showing that connections are established with the target device, and tag data links are operating:

**(3) The element corresponding to the target node address in the Normal Target Node Table variable (*_*EstbTargetSta*) *[0] to [63]* is TRUE.**

### ● Condition for the Target Node PLC Operating Flags (operating or stopped) (*_*TargetPLCMdSta*) (valid for OMRON Controllers only):

**(4) The Target PLC Operating Mode of the target node is TRUE.**

### ● Condition for the Target PLC Error Information of the target node (valid for OMRON Controllers only):

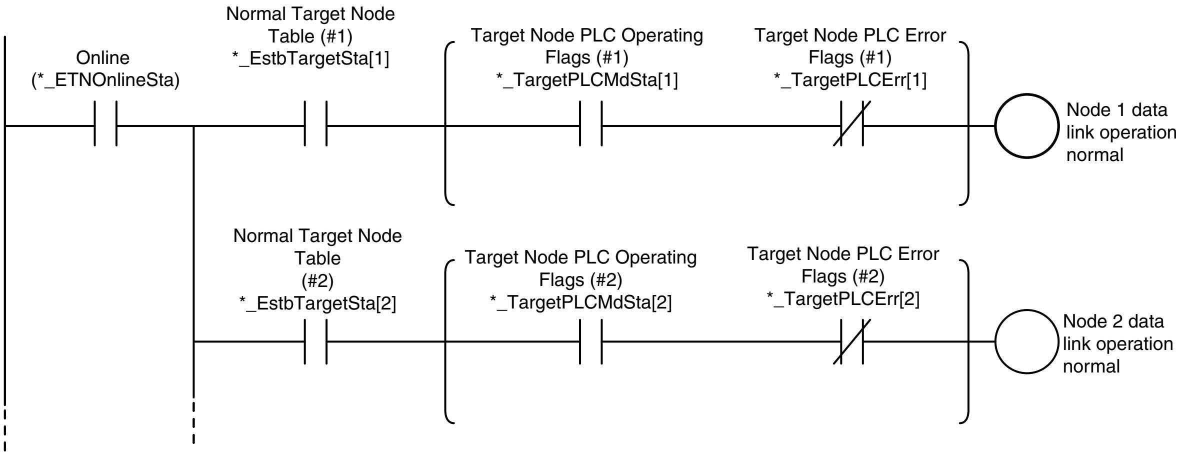
**(5) The corresponding element in the Target Node PLC Error Flags (*_*TargetPLCErrSta*) is FALSE.**

When you want to use the Target Node PLC Error Flags, 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.

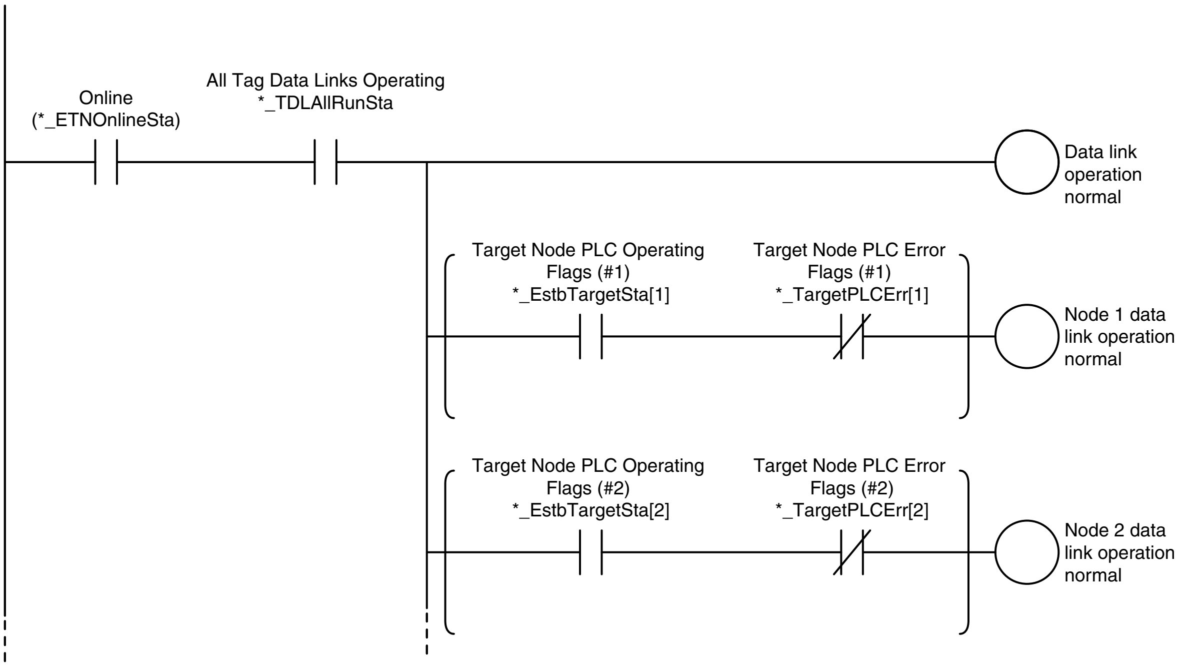
### Programming Example to Detect Normal Operation

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

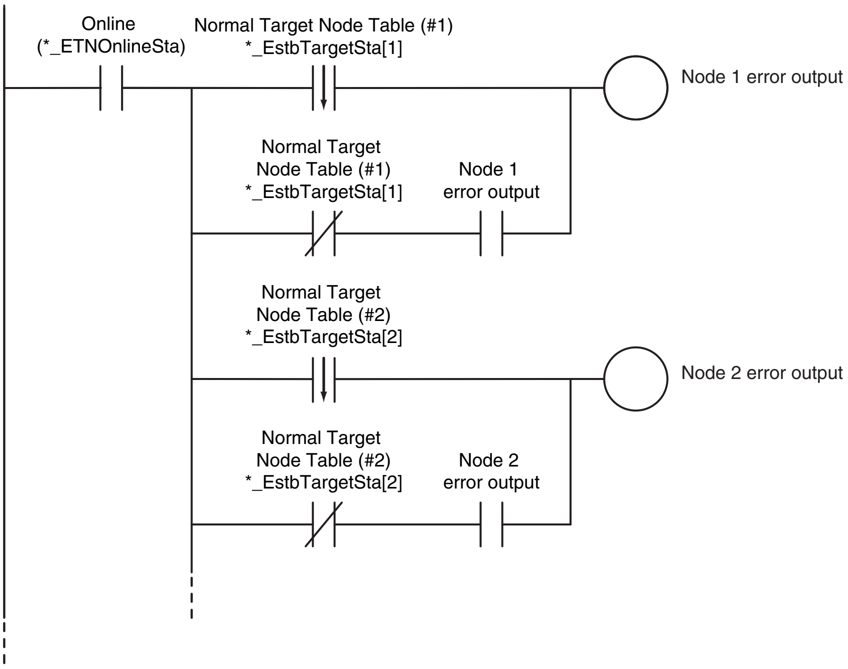

Online
(*_ETNOnlineSta)
Normal Target Node Table (#1)
*_EstbTargetSta[1]
Target Node PLC Operating Flags (#1)
*_TargetPLCMdSta[1]
Target Node PLC Error Flags (#1)
*_TargetPLCErr[1]
Node 1 data link operation normal
Normal Target Node Table (#2)
*_EstbTargetSta[2]
Target Node PLC Operating Flags (#2)
*_TargetPLCMdSta[2]
Target Node PLC Error Flags (#2)
*_TargetPLCErr[2]
Node 2 data link operation normal

- Programming Example 2 to Detect Normal Operation


Online
(*_ETNOnlineSta)
All Tag Data Links Operating
*_TDLAllRunSta
Data link operation normal
Target Node PLC Operating Flags (#1)
*_EstbTargetSta[1]
Target Node PLC Error Flags (#1)
*_TargetPLCErr[1]
Node 1 data link operation normal
Target Node PLC Operating Flags (#2)
*_EstbTargetSta[2]
Target Node PLC Error Flags (#2)
*_TargetPLCErr[2]
Node 2 data link operation normal

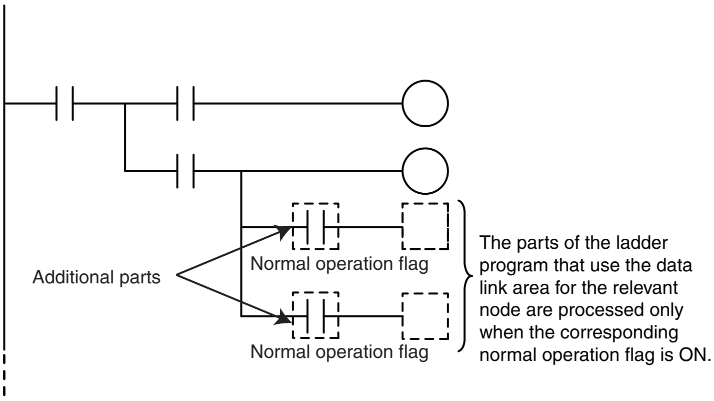
## ● Programming Example to Detect Errors

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.

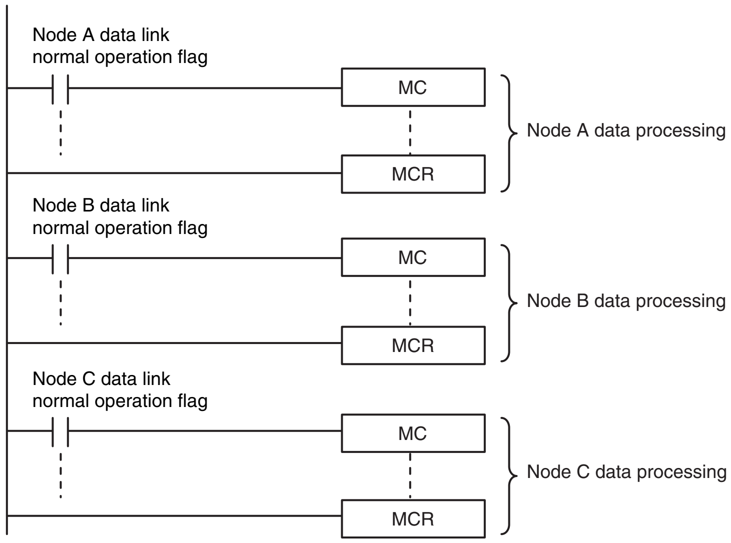


## ● Data Processing Programming Example

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


Additional parts
Normal operation flag
Normal operation flag
The parts of the ladder program that use the data link area for the relevant node are processed only when the corresponding normal operation flag is ON.

- You can use MC – MCR instructions and JMP instructions to process data only when the data links are operating normally as shown below.


Node A data link normal operation flag
MC
MCR
Node A data processing
Node B data link normal operation flag
MC
MCR
Node B data processing
Node C data link normal operation flag
MC
MCR
Node C data processing

### 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 Unit Error Occurred (*_*UnitErr*) are TRUE.

## 7-3-2 Status Flags Related to Tag Data Links

The status of the tag data links is reflected in the following device variables for the CJ-series Unit.

| Variable name | Description |
|---|---|
| *_*TargetPLCMdSta [0] to [63]* (Target Node PLC Operating Flags)<br>* Corresponds to the operating information in the Controller status. | This variable shows the operating status of the target PLCs that are connected with the EtherNet/IP Unit as the originator. The information in this area is valid only when the corresponding Normal Target Node Table is TRUE. If the corresponding Normal Target Node Table is FALSE, the Target Node PLC Operating Flags indicates the previous operating status.<br>*Array[x]* is TRUE: The target PLC with a node address of x is in operating status.<br>*Array[x]* is FALSE: Other than the above. |
| *_*TargetPLCErrSta [0] to [63]* (Target Node PLC Error Flags)<br>* Corresponds to the Controller Error Information in the Controller status. | This variable indicates that the connection for the Registered Target Node Table was not established or that an error occurred in the target PLC. The information in this area is valid only when the corresponding element in the Registered Target Node Table is TRUE.<br>*Array[x]* is TRUE: The Registered Target Node Table for a node address of x is TRUE, and the Normal Target Node Table is FALSE or the Target Node PLC Error Flags is TRUE.<br>*Array[x]* is FALSE: When the Registered Target Node Table for a node address of x is FALSE, or when the Registered Target Node Table is TRUE and the Normal Target Node Error Information is TRUE.<br>This is when the Controller Error Information is FALSE. |
| *_*EstbTargetSta [0] to [63]* (Normal Target Node Table)<br>* This status is not included in the Controller status. | This variable gives a list of nodes that have normally established EtherNet/IP connections.<br>*Array[x]* is TRUE: The connection to the node with a node address of x is established normally.<br>*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 EtherNet/IP Unit and another EtherNet/IP Unit or CPU Unit, use tag data link settings that match the Unit with the lower communications performance.

### ● Differences in Tag Data Link Performance Specifications

| Item | | CS1W-EIP21/EIP21S, CJ1W-EIP21/EIP21S, or CJ2H-CPU6□-EIP | CJ2M-CPU3□ | | NJ-series CPU Unit | |
|---|---|---|---|---|---|---|
| | | | Unit version | | | |
| | | | 2.0 | 2.1 or later | Ver. 1.02 or earlier | Ver. 1.03 or later |
| Tags | Total size of all tags | 184,832 words | 640 words | | 9,600 words | |
| | Maximum size of 1 tag | 722 words (721 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) | 300 words (299 words when the tag set includes the Controller status) | |
| | Number of registrable tags | 256 | 32 | | 256[*1] | |
| Tag sets | Maximum size of 1 tag set | 722 words (721 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) | 300 words (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)<br>**Note** Input and output variables cannot be combined. | | | | |
| | Number of registrable tag sets | 256 | 32 | | 32 | |
| Connections | Number of connections | 256 | 32 | | 32 | |
| | Maximum data size per connection | 252 words or 722 words[*2] (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<br>(Data concurrency is maintained within each connection.) | 640 words | 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.) | |
| Packet intervals (RPIs) | | 0.5 to 10,000 ms in 0.5-ms increments | 1 to 10,000 ms in 0.5-ms increments | | 10 to 10,000 ms in 1-ms increments | 1 to 10,000 ms in 1-ms increments |
| Allowed communications bandwidth per Unit (pps) | | 6000 to 12,000 pps[*3, *4] | 3,000 pps[*3] | | 1,000 pps[*3] | 3,000 pps[*3] |

[*1] 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.

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

*4 For the Units with unit version 2.1 or earlier, this is 6,000 pps.

### ● 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 | | | |
| NX-series CPU Unit | --- | NX701 | Yes | No |
| NJ-series CPU Unit | --- | NJ101, NJ301-1100, NJ301-1200, NJ501-1300, NJ501-1400, or NJ501-1500 | Yes | Yes*1 |
| | CJ1W-EIP21 | CJ1W-EIP21 (NJ) | Yes | Yes*1 |
| | CJ1W-EIP21S | CJ1W-EIP21S(NJ) | Yes | Yes*1 |
| CJ2H-CPU6□-EIP | --- | CJ2B-EIP21 | Yes | Yes |
| | CJ1W-EIP21 | CJ1W-EIP21(CJ2) | Yes | Yes |
| | CJ1W-EIP21S | CJ1W-EIP21S(CJ2) | Yes | Yes |
| CJ2H-CPU6□ | CJ1W-EIP21 | CJ1W-EIP21(CJ2) | Yes*2 | Yes |
| | CJ1W-EIP21S | CJ1W-EIP21S(CJ2) | Yes*2 | Yes |
| CJ2M-CPU3□ | --- | CJ2M-EIP21 | Yes | Yes |
| | CJ1W-EIP21 | CJ1W-EIP21(CJ2) | Yes | Yes |
| | CJ1W-EIP21S | CJ1W-EIP21S(CJ2) | Yes | Yes |
| CJ2M-CPU1□ | CJ1W-EIP21 | CJ1W-EIP21(CJ2) | Yes*3 | Yes |
| | CJ1W-EIP21S | CJ1W-EIP21S(CJ2) | Yes*3 | Yes |
| CJ1 CPU Unit | CJ1W-EIP21 | CJ1W-EIP21 | No | Yes |
| | CJ1W-EIP21S | CJ1W-EIP21S | No | Yes |
| CS1 CPU Unit | CS1W-EIP21 | CS1W-EIP21 | No | Yes |
| | CS1W-EIP21S | CS1W-EIP21S | No | Yes |

*1 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.

*2 The CJ2H-CPU6□ CPU Unit with unit version 1.6 or later is supported.

*3 The CJ2M-CPU1□ CPU Unit with unit version 2.2 or later is supported.

# 8

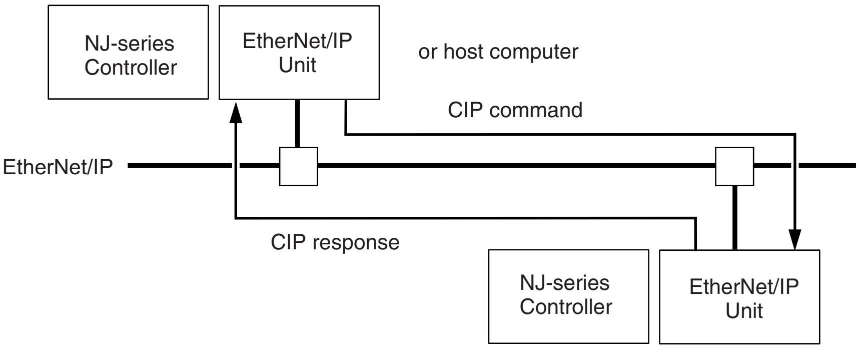
# Message Communications

# 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.



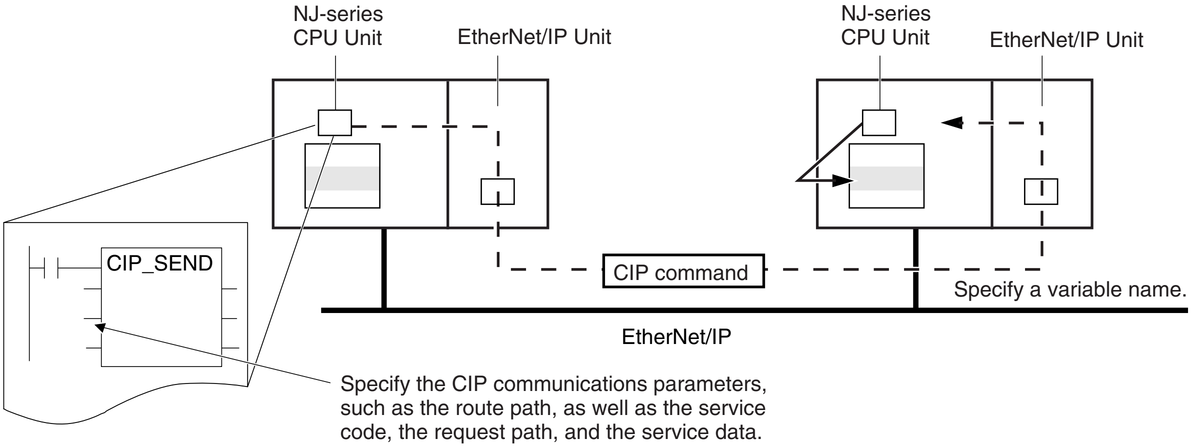
## 8-1-2 Message Communications Service Specifications

| Item | | Specification |
|---|---|---|
| Message type | | Either of the following can be selected.<br>CIP UCMM connectionless messages<br>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 |
| Maximum length per connection | Non-connection type (UCMM) | 502 bytes |
| | Connection type (class 3) | 502 bytes for Forward_Open<br>1,994 bytes for Large_Forward_Open |

# 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.


NJ-series CPU Unit
EtherNet/IP Unit
NJ-series CPU Unit
EtherNet/IP Unit
CIP_SEND
CIP command
Specify a variable name.
EtherNet/IP
Specify the CIP communications parameters, such as the route path, as well as the service code, the request path, and the service data.

## 8-2-2 CIP Communications Instructions

The following CIP communications instructions are available. For details on CIP communications instructions, refer to the *NJ/NX-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.<br>Refer to *8-2-9 Response Codes* and *8-3 CIP Objects Sent to the EtherNet/IP Unit* for information on the service codes and response codes that are used with the NJ/NX-series CPU Units. | |
| **CIPOpen** | Open CIP Class 3 Connection (Large_Forward_Open) | Opens a CIP class 3 connection (Large_Forward_Open) with the specified remote node. | CIP class 3 connection message |
| **CIPOpenWithDataSize** | Open CIP Class 3 Connection with Specified Data Size | Opens a CIP class 3 connection with the specified remote node that allows class 3 explicit messages of the specified data length or shorter to be sent and received. | |
| **CIPRead** | Read Variable Class 3 Explicit | Reads the value of a variable with a Network Publish attribute from the specified remote Controller on the CIP network and stores the value in a variable at the local Controller. | |
| **CIPWrite** | Write Variable Class 3 Explicit | Writes the value of a variable at the local controller to a variable with a Network Publish attribute at the specified remote Controller on the CIP network. | |
| **CIPSend** | Send Explicit Message Class 3 | Sends a specified class-3 CIP command to the specified remote Controller on the CIP network.<br>Refer to *8-2-9 Response Codes* and *8-3 CIP Objects Sent to the EtherNet/IP Unit* for information on the service codes and response codes that are used with the NJ/NX-series CPU Units. | |
| **CIPClose** | Close CIP Class 3 Connection | This instruction closes the CIP class 3 connection that is specified by the handle. | |

**Version Information**

A CPU Unit with unit version 1.06 or later and Sysmac Studio version 1.07 or higher are required to use the CIPOpenWithDataSize instruction.

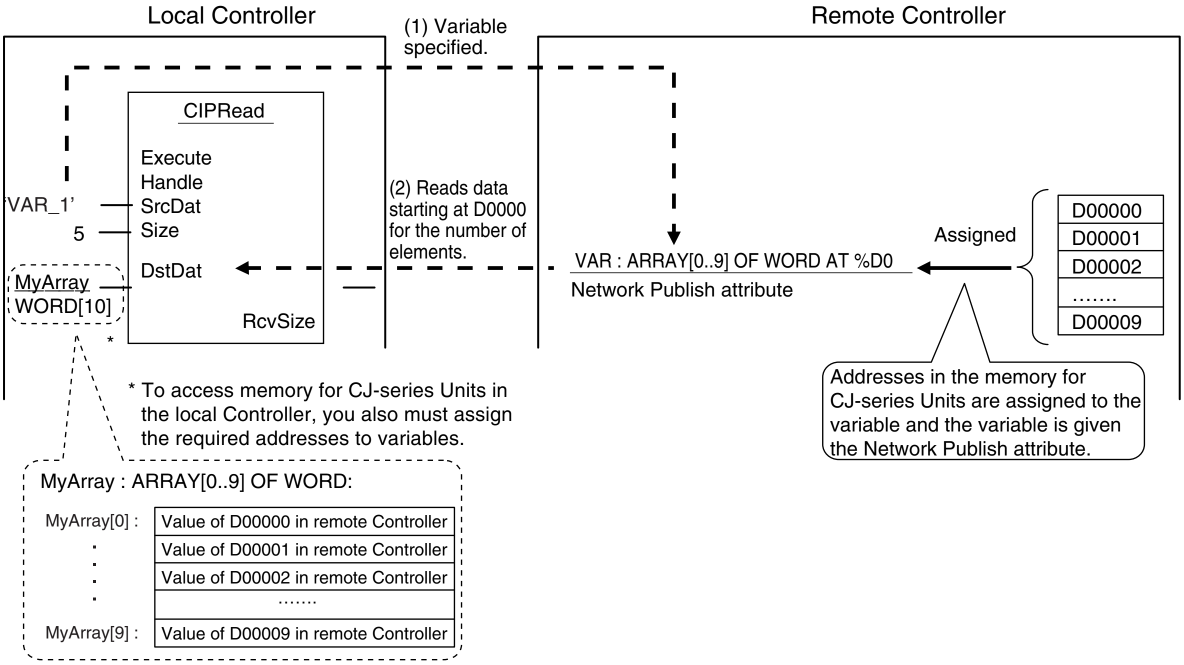
## 8-2-3 Using CIP Communications Instructions

CIP message communications include the following processes. If CIP class 3 connections are used, the open and close processes are required before and after the data is sent and received.

| Process | Description | Instruction |
|---|---|---|
| Open process (only for CIP class 3 connections)*1 | Execute this process before you use a CIP message. Open processing is continued until a CIP class 3 connection is established. | CIPOpen<br>CIPOpenWithDataSize |
| Sending and receiving variable data*2 | This process is used to read and write data for specified variables with the Network Publish attributes. | CIPUCMMRead<br>CIPUCMMWrite<br>CIPRead<br>CIPWrite |
| Sending CIP commands | You can set the required CIP command. | CIPUCMMSend<br>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.



Local Controller
Remote Controller
(1) Variable specified.
CIPRead
Execute
Handle
SrcDat
Size
DstDat
RcvSize
'VAR_1'
5
MyArray
WORD[10]
*
(2) Reads data starting at D0000 for the number of elements.
VAR : ARRAY[0..9] OF WORD AT %D0
Network Publish attribute
Assigned
D00000
D00001
D00002
…….
D00009
Addresses in the memory for CJ-series Units are assigned to the variable and the variable is given the Network Publish attribute.
* To access memory for CJ-series Units in the local Controller, you also must assign the required addresses to variables.
MyArray : ARRAY[0..9] OF WORD:
MyArray[0] :
MyArray[9] :
Value of D00000 in remote Controller
Value of D00001 in remote Controller
Value of D00002 in remote Controller
…….
Value of D00009 in remote Controller

### 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.

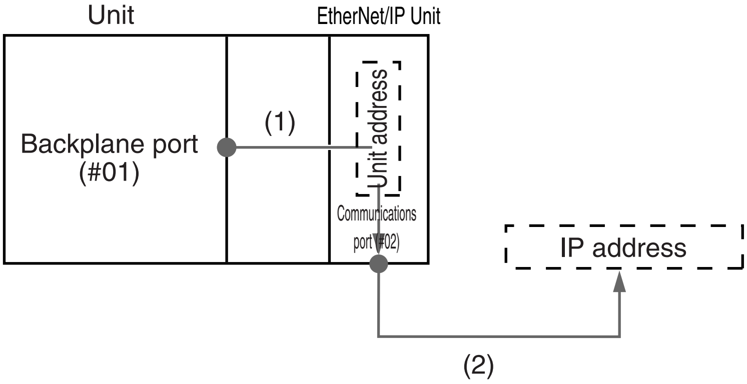
### Route Path Notation

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 (EtherNet/IP Unit) | 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.

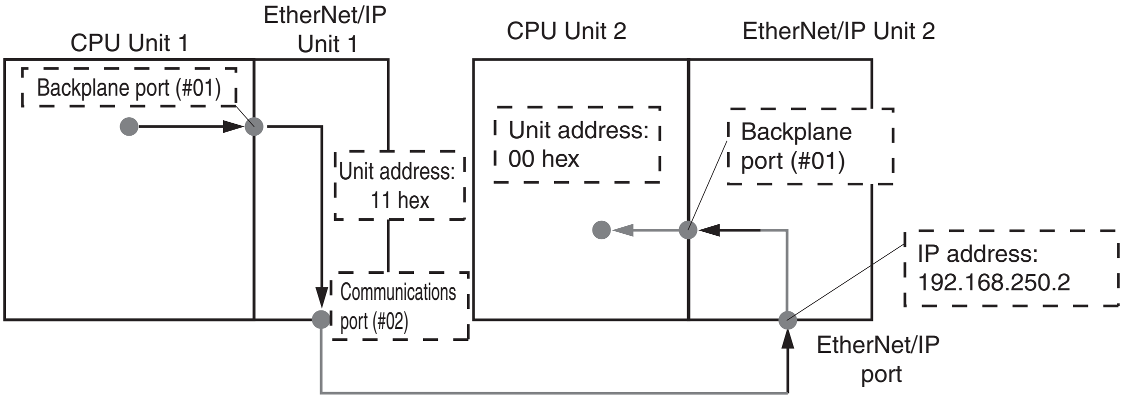
| | |
|---|---|
| • CPU Unit: | 00 hex, 01 hex |
| • CPU Bus Units (EtherNet/IP Units): | Unit number + 10 hex |

## 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 with an EtherNet/IP Unit

Example: Communicating from the EtherNet/IP Unit mounted to 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 1**

- Network type number: “01” (Output the command via internal backplane port.)
- Remote address: “#11” (unit address of EtherNet/IP Unit (unit number: 1+10 hex))

**(2) EtherNet/IP 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.

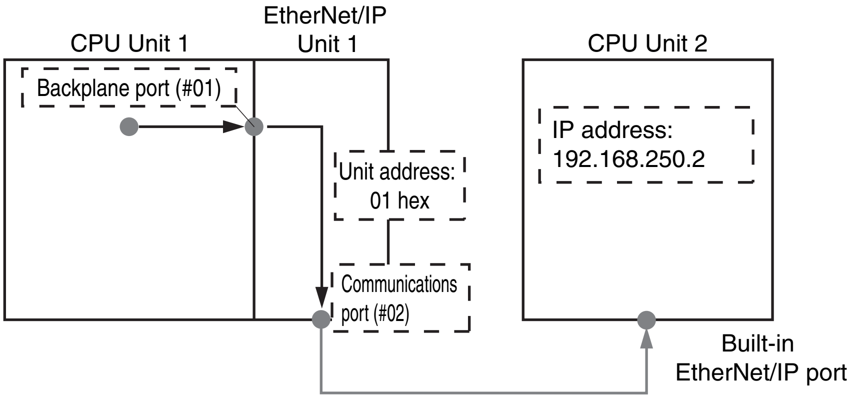
**(3) EtherNet/IP Unit 2 to CPU Unit 2**

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

Route path: $\underbrace{\text{\textbackslash 01\textbackslash\#11}}_{(1)}\underbrace{\text{\textbackslash 02\textbackslash 192.168.250.2}}_{(2)}\underbrace{\text{\textbackslash 01\textbackslash\#00}}_{(3)}$

### ● Communicating with a Built-in EtherNet/IP Port

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



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

- Network type number: “01” (Output the command via internal backplane port.)
- Remote address: “#01” (unit address of EtherNet/IP Unit)

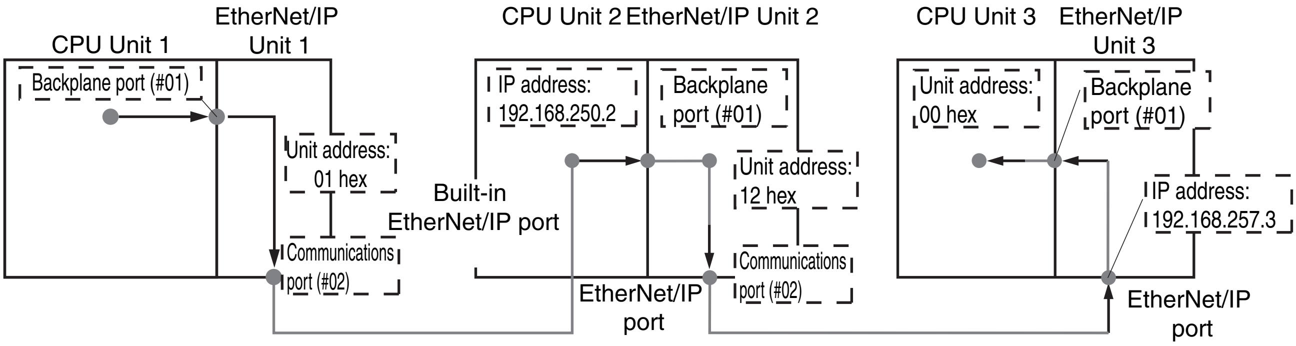
**(2) EtherNet/IP Unit 1 to CPU Unit 2 (built-in EtherNet/IP port)**

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

Route path:\01\#01\02\192.168.250.2

## ● Access via a Relay Node

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



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

- Network type number: “01” (Output the command via internal backplane port.)
- Remote address: “#01” (unit address of EtherNet/IP Unit)

**(2) EtherNet/IP Unit 1 to CPU Unit 2 (built-in EtherNet/IP port)**

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

**(3) 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))

**(4) 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.

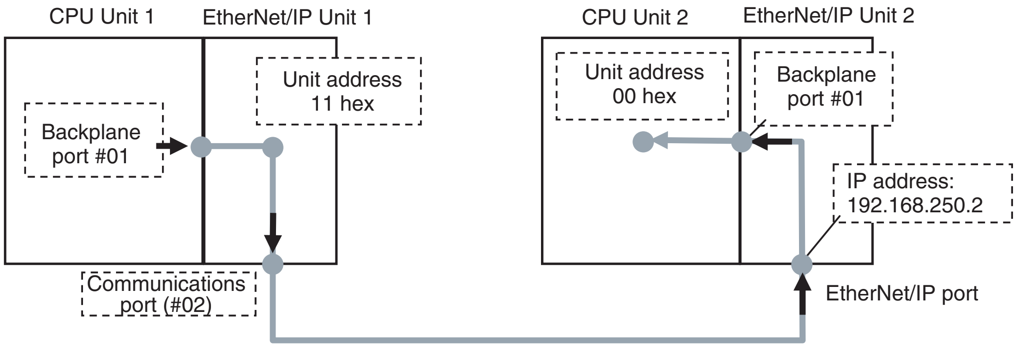
**(5) EtherNet/IP Unit 3 to CPU Unit 3**

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

Route path: $\underbrace{\text{\textbackslash 01\textbackslash \#01}}_{(1)}\underbrace{\text{\textbackslash 02\textbackslash 192.168.250.2}}_{(2)}\underbrace{\text{\textbackslash 01\textbackslash \#12}}_{(3)}\underbrace{\text{\textbackslash 02\textbackslash 192.168.257.3}}_{(4)}\underbrace{\text{\textbackslash 01\textbackslash \#00}}_{(5)}$

### ● Communicating between EtherNet/IP Units

Example: Communicating via EtherNet/IP Units Mounted to CPU Unit 1 and CPU Unit 2



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

- Network type number: "#01" (Output the command via internal backplane port.)
- Remote address: "#11" (Unit address of EtherNet/IP Unit (Unit number: 1+10 hex))

**(2) EtherNet/IP 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.

**(3) 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 : $\text{\textbackslash 01\textbackslash}\underbrace{\text{\#11}}_{(1)}\text{\textbackslash 02\textbackslash}\underbrace{\text{192.168.250.2}}_{(2)}\text{\textbackslash 01\textbackslash}\underbrace{\text{\#00}}_{(3)}$

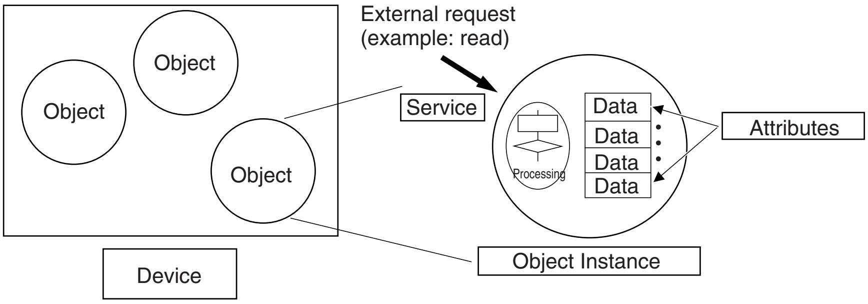
(1) Unit address of Local EtherNet/IP Unit (Unit number: 1+10 hex)
(2) IP address of remote EtherNet/IP Unit
(3) Unit address of remote CPU Unit

**Version Information**

- You can use the CJ1W-EIP21 EtherNet/IP Unit mounted to an NJ-series Controller with a CPU Unit with unit version 1.01 or later and Sysmac Studio version 1.02 or higher.
- You can use the CJ1W-EIP21S EtherNet/IP Unit mounted to an NJ-series Controller with a CPU Unit with unit version 1.67 or later and Sysmac Studio version 1.60 or higher.

## 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.

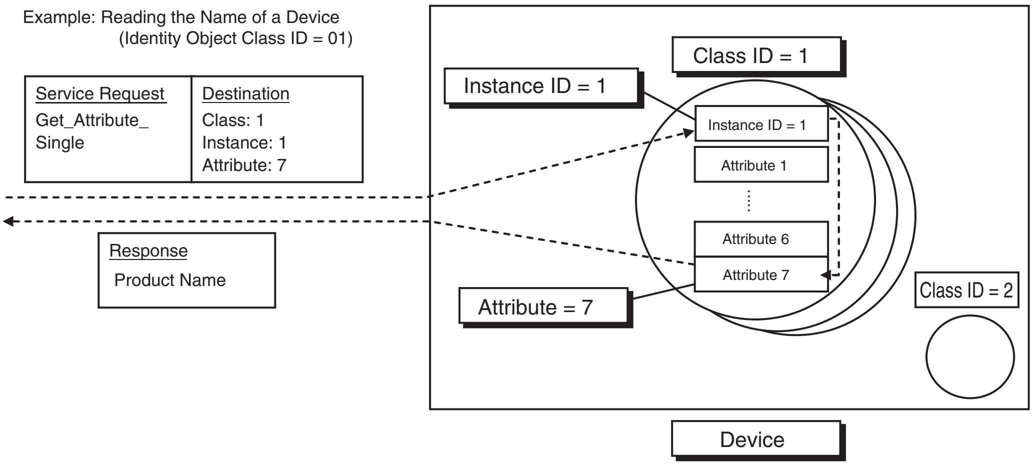


In the CIP Common Specification, “Object,” “Class,” “Instance,” “Attribute,” and “Service” are defined as follows: (Source: CIP Common Specification)

| 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 EtherNet/IP Unit* 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/NX-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.

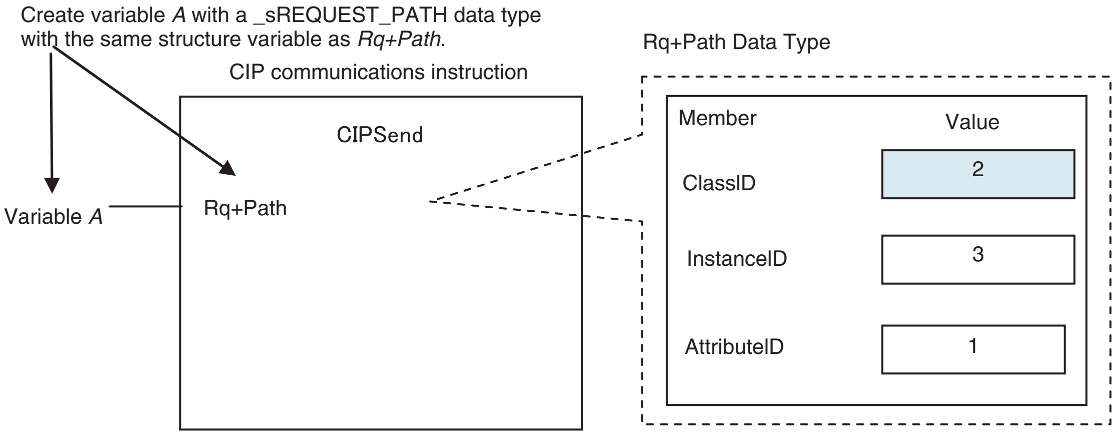
### Providing the Structure Variables to Input Request Paths

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

- **Creating Structure Variables**

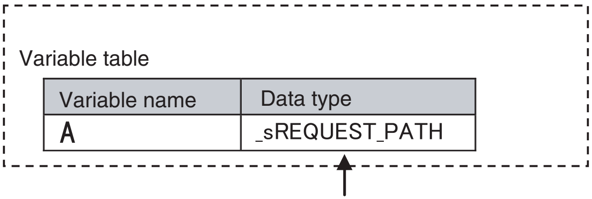
To input a value into the structure variable of a CIP communications instruction, you must create a user-defined variable with the same configuration in advance.

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


Create variable *A* with a _sREQUEST_PATH data type with the same structure variable as *Rq+Path*.
CIP communications instruction
CIPSend
Variable *A*
Rq+Path
Rq+Path Data Type
Member
Value
ClassID
2
InstanceID
3
AttributeID
1

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:

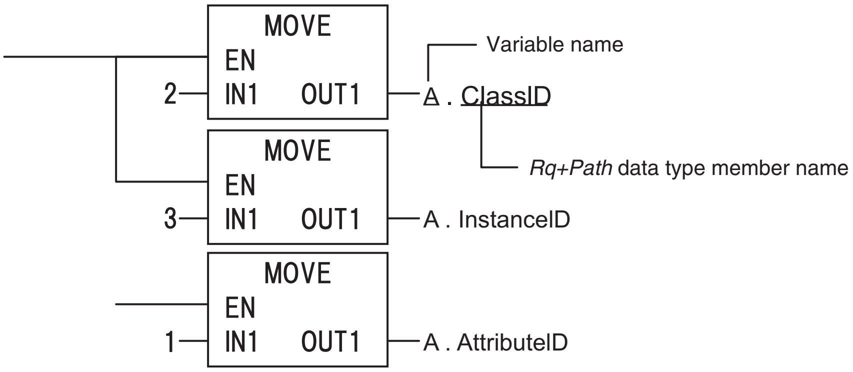

Variable table

| Variable name | Data type |
|---|---|
| A | _sREQUEST_PATH |

Select the same data type as *Rq+Path* for the data type of variable *A*.

### ● 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.


MOVE
EN
2 IN1 OUT1 A . ClassID
Variable name
Rq+Path data type member name
MOVE
EN
3 IN1 OUT1 A . InstanceID
MOVE
EN
1 IN1 OUT1 A . AttributeID

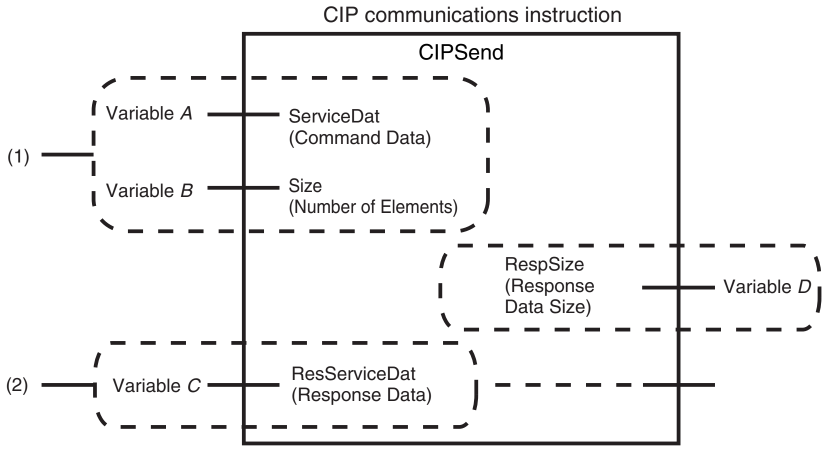
## Preparing Array Variables to Input and Output Service Data and Response Data

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.

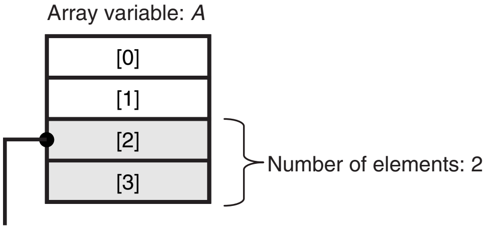
Example: Creating a Variable to Input Data to the CIPSend Instruction Array Variables



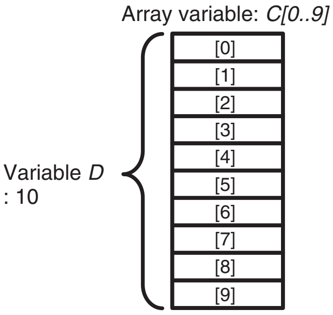
(1) Input the service data to send.

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.



If the service data (*ServiceDat*) is *Array[2]* and number of elements (*Size*) = 2, *Array[2]* and *Array[3]* are sent.

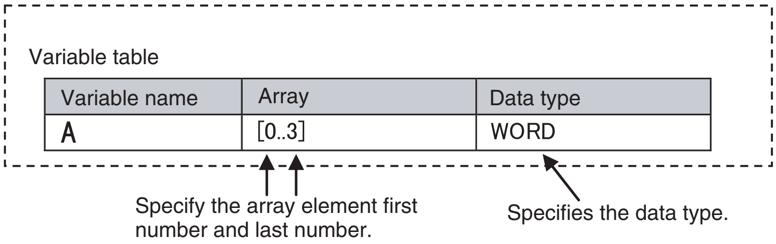


Use the following procedure to create a variable in the variable table.

**1** Select the *Array* Check Box.

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

Example: UINT Array



Variable table

| Variable name | Array | Data type |
|---|---|---|
| A | [0..3] | WORD |

Specify the array element first number and last number.

Specifies the data type.

## ● 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

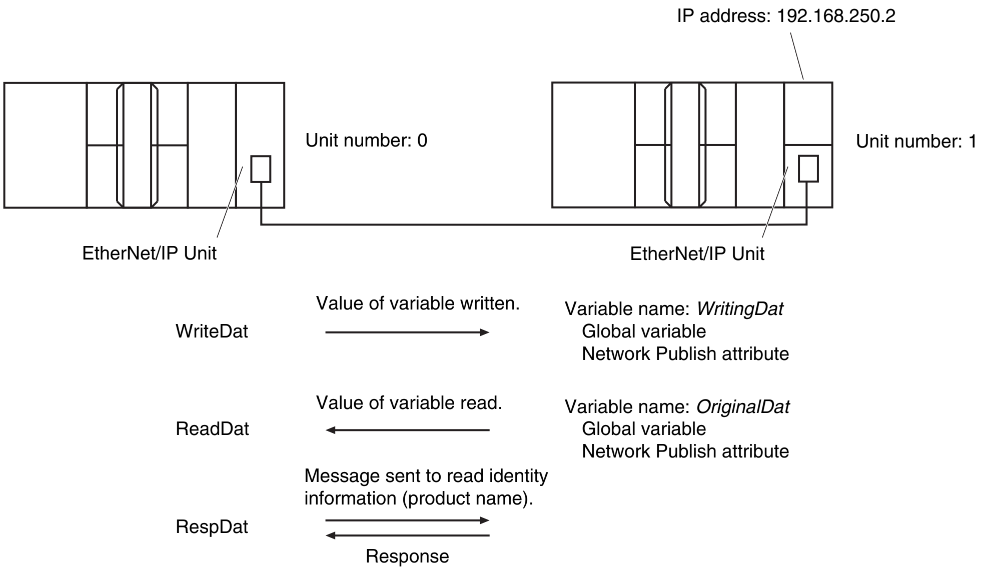
This sample uses CIP UCMM messages to write a variable, read a variable, and send a message. The Controllers are connected to an EtherNet/IP network. The IP address of the remote node is 192.168.250.2. The following procedure is used.

**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.

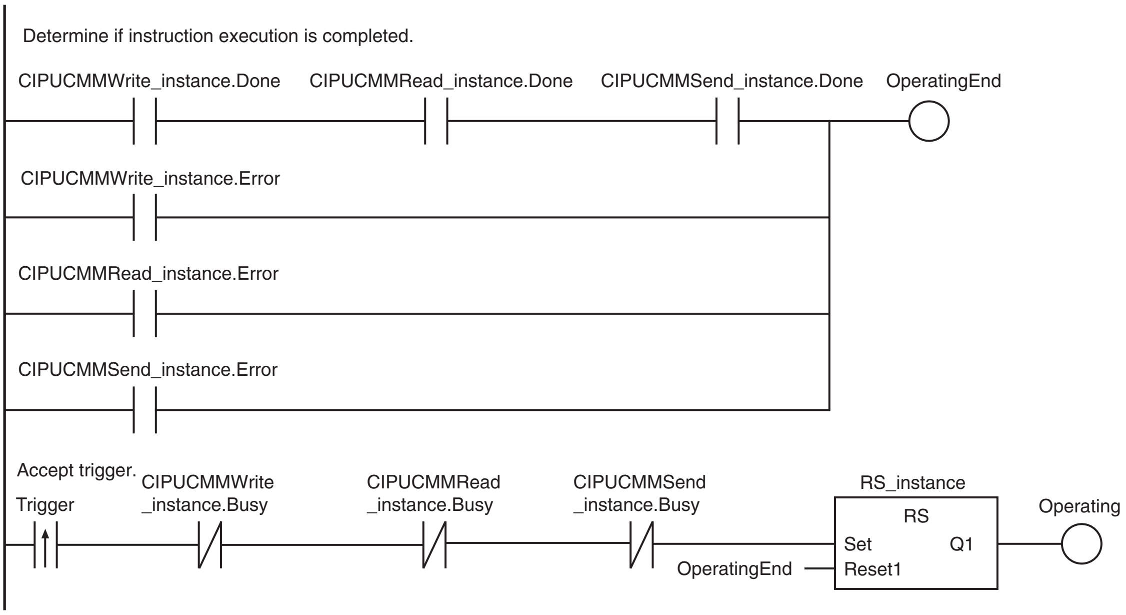
**3** The CIPUCMMSend instruction is used to send an explicit message to a remote node. The contents of the message is to read identity information (product name). The class ID, instance ID, attribute ID, and service code are as follows. The response data is stored in the RespDat variable.

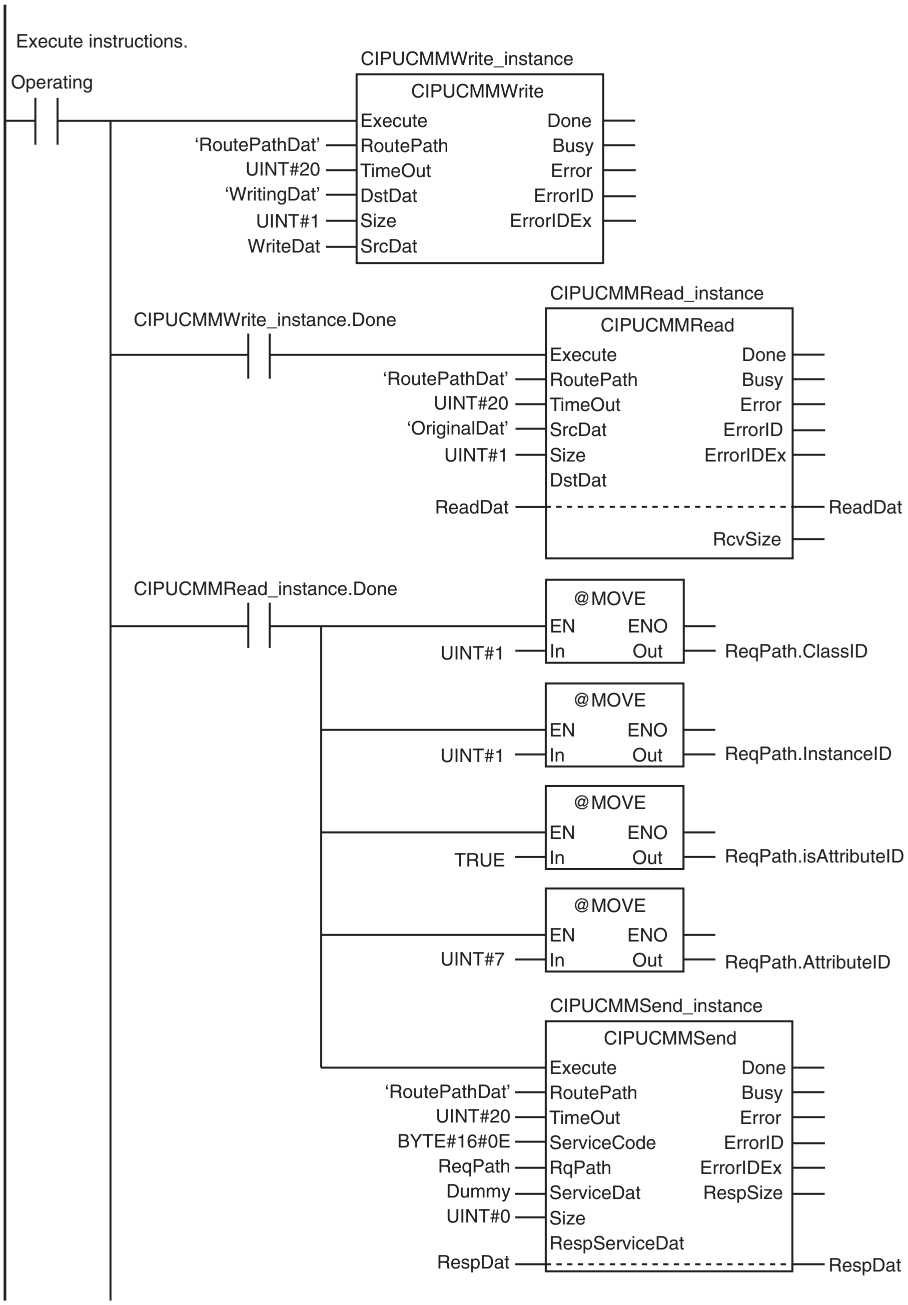
| Item | Value |
|---|---|
| Class ID | 1 |
| Instance ID | 1 |
| Attribute ID | 7 |
| Service code | 16#0E |

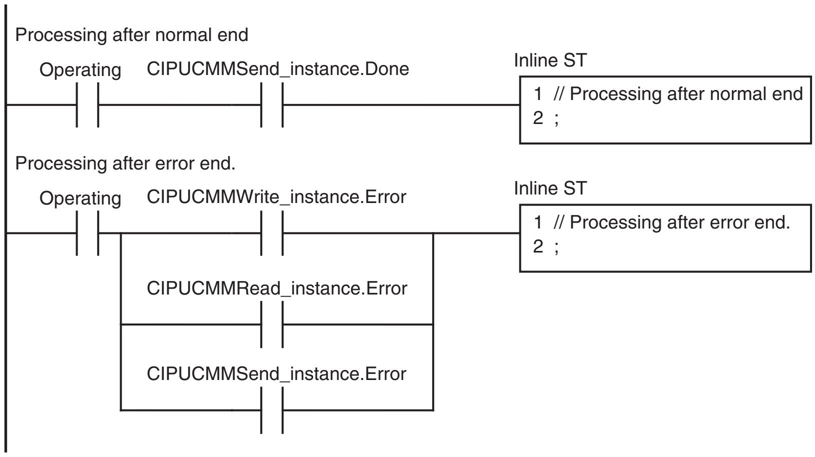

IP address: 192.168.250.2
Unit number: 0
Unit number: 1
EtherNet/IP Unit
EtherNet/IP Unit
WriteDat
Value of variable written.
Variable name: *WritingDat*
Global variable
Network Publish attribute
ReadDat
Value of variable read.
Variable name: *OriginalDat*
Global variable
Network Publish attribute
Message sent to read identity information (product name).
RespDat
Response

## 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 |
| RoutePathDat | string[256] | 01\#10\02\192.168.250.2\01\#00 | Route path |
| ReqPath | _sREQUEST_ PATH | (ClassID:=0, InstanceID:=0, isAttributeID:=False, AttributeID:=0) | Request path |
| RespDat | ARRAY[0..10] OF BYTE | [11(16#0)] | Response data |
| Dummy | BYTE | 16#0 | Dummy |
| RS_instance | RS | | |
| CIPUCMMWrite_instance | CIPUCMMWrite | | |
| CIPUCMMRead_instance | CIPUCMMRead | | |
| CIPUCMMSend_instance | CIPUCMMSend | | |


Determine if instruction execution is completed.
CIPUCMMWrite_instance.Done
CIPUCMMRead_instance.Done
CIPUCMMSend_instance.Done
OperatingEnd
CIPUCMMWrite_instance.Error
CIPUCMMRead_instance.Error
CIPUCMMSend_instance.Error
Accept trigger.
Trigger
CIPUCMMWrite _instance.Busy
CIPUCMMRead _instance.Busy
CIPUCMMSend _instance.Busy
RS_instance
RS
Set
Q1
OperatingEnd
Reset1
Operating


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


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 | 1234 | Write data |
| | ReadDat | INT | 0 | Read data |
| | ReqPath | _sREQUEST_ PATH | (ClassID:=0, InstanceID:=0, isAttributeID:=False, AttributeID:=0) | Request path |
| | RoutePathDat | string[256] | 01\#10\02\192.168.250.2\01\#00 | Route path |
| | RespDat | ARRAY[0..10] OF BYTE | [11(16#0)] | Response data |
| | Dummy | BYTE | 16#0 | Dummy |
| | CIPUCMMWrite_instance | CIPUCMMWrite | | |
| | CIPUCMMRead_instance | CIPUCMMRead | | |
| | CIPUCMMSend_instance | CIPUCMMSend | | |

| External Variables | Variable | Data type | Constant | Comment |
|---|---|---|---|---|
| | J 01_ETNOnlineSta | BOOL | ☑ | Online |

```
// Start sequence when Trigger changes to TRUE.
IF ((Trigger=TRUE) AND (DoUCMMTrigger=FALSE) AND (J 01_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        :='RoutePathDat',      // 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           :='RoutePathDat',       // 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           :='RoutePathDat',       // Route path
            TimeOut             :=UINT#20,              // Timeout time
            ServiceCode         :=BYTE#16#0E,           // Service code
            RqPath              :=ReqPath,              // Request path
            ServiceDat          := Dummy,               // Service data
            Size                :=UINT#0,               // Number of elements
            RespServiceDat      :=RespDat);             // Response data

        IF (CIPUCMMSend_instance.Done=TRUE) THEN
            Stage               :=INT#0;                // Normal end
        ELSIF (CIPUCMMSend_instance.Error=TRUE) THEN
            Stage               :=INT#30;               // Error end
        END_IF;

    0:                                                  // Processing after normal end
        DoUCMMTrigger:=FALSE;
        Trigger             :=FALSE;

    ELSE                                                // Processing after error end
        DoUCMMTrigger:=FALSE;
        Trigger             :=FALSE;
    END_CASE;
END_IF;
```

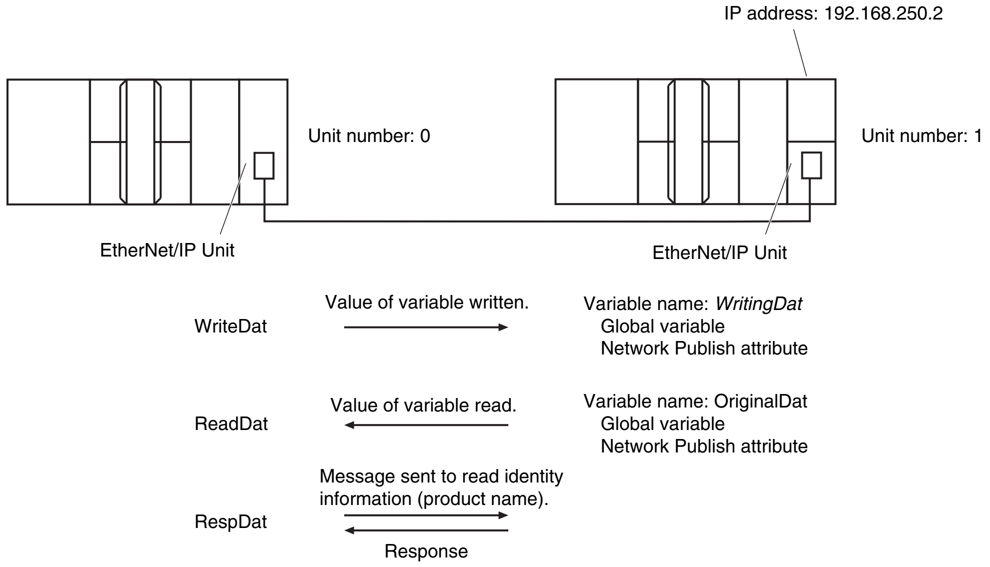
## 8-2-7 Sample Programming for CIP Connection (Class 3) Message Communications

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 RespDat variable.

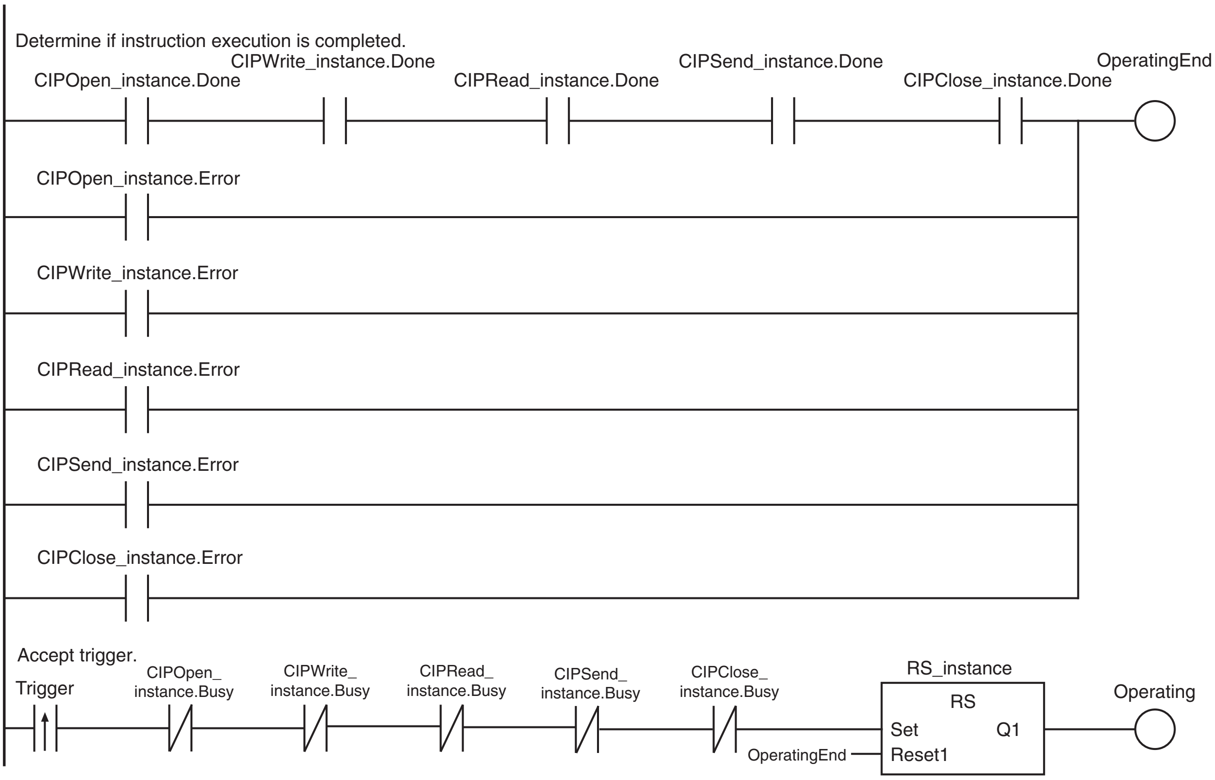
| Item | Value |
|---|---|
| Class ID | 1 |
| Instance ID | 1 |
| Attribute ID | 7 |
| Service code | 16#0E |

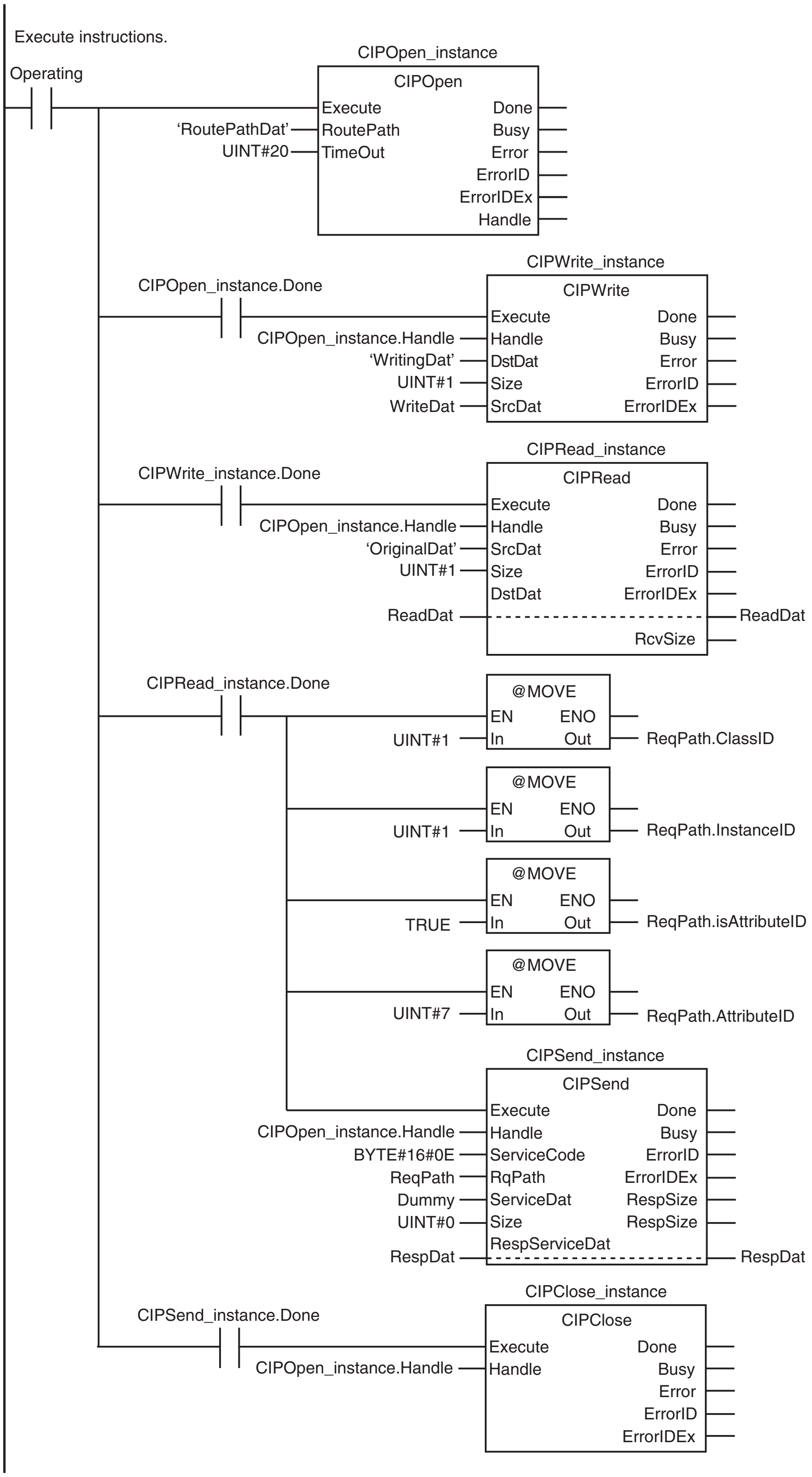
5. The CIPClose instruction is used to close the class 3 connection.

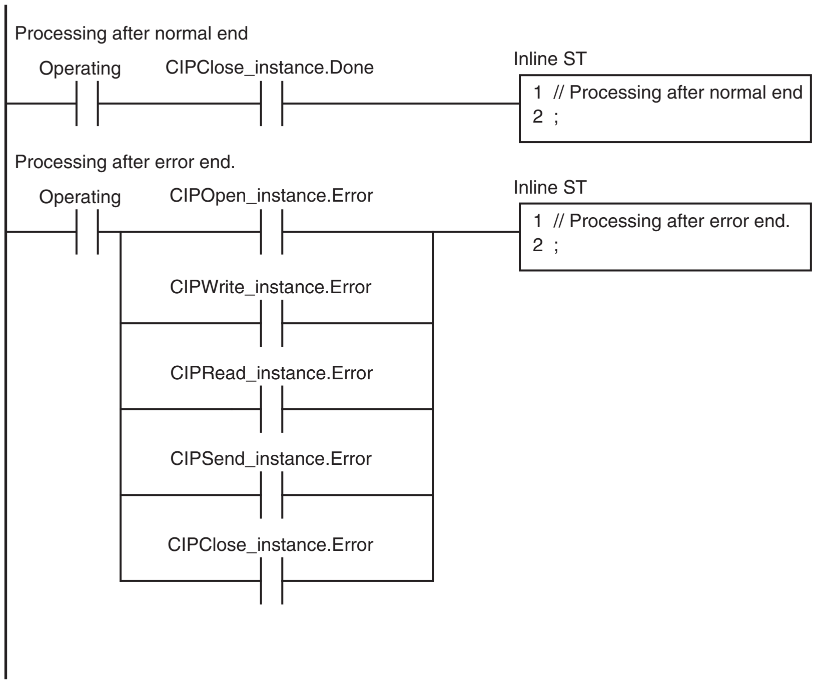

IP address: 192.168.250.2
Unit number: 0
Unit number: 1
EtherNet/IP Unit
EtherNet/IP Unit
WriteDat
Value of variable written.
Variable name: *WritingDat*
Global variable
Network Publish attribute
ReadDat
Value of variable read.
Variable name: OriginalDat
Global variable
Network Publish attribute
RespDat
Message sent to read identity information (product name).
Response

### 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 |
| RoutePathDat | string[256] | 01\#10\02\192.168.250.1\01\#00 | Route path |
| ReqPath | _sREQUEST_PATH | (ClassID:=0, InstanceID:=0, isAttributeID:=False, AttributeID:=0) | Request path |
| RespDat | ARRAY[0..10] OF BYTE | [11(16#0)] | Response data |
| Dummy | BYTE | 16#0 | Dummy |
| RS_instance | RS | | |
| CIPOpen_instance | CIPOpen | | |
| CIPWrite_instance | CIPWrite | | |
| CIPRead_instance | CIPRead | | |
| CIPSend_instance | CIPSend | | |
| CIPClose_instance | CIPClose | | |




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


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 | 1234 | Write data |
| | ReadDat | INT | 0 | Read data |
| | RoutePathDat | string[256] | 01\#10\02\192.168.250.1\01\#00 | Route path |
| | ReqPath | _sREQUEST_ PATH | (ClassID:=0, InstanceID:=0, isAttributeID:=False, AttributeID:=0) | Request path |
| | RespDat | ARRAY[0..10] OF BYTE | [11(16#0)] | Response data |
| | Dummy | BYTE | 16#0 | Dummy |
| | CIPOpen_instance | CIPOpen | | |
| | CIPWrite_instance | CIPWrite | | |
| | CIPRead_instance | CIPRead | | |
| | CIPSend_instance | CIPSend | | |
| | CIPClose_instance | CIPClose | | |

| External Variables | Variable | Data type | Constant | Comment |
|---|---|---|---|---|
| | J01_ETNOnlineSta | BOOL | ☑ | Online |

```
// Start sequence when Trigger changes to TRUE.
IF ((Trigger=TRUE) AND (DoCIPTrigger=FALSE) AND (J01_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 :='RoutePathDat');              // Route path

    IF (CIPOpen_instance.Done=TRUE) THEN
      Stage     :=INT#2;                        // Normal end
    ELSIF (CIPOpen_instance.Error=TRUE) THEN
      Stage     :=INT#10;                       // Error end
    END_IF;

  2 :                                           // Request writing value of variable.
    CIPWrite_instance(
      Execute   :=TRUE,
      Handle    :=CIPOpen_instance.Handle,      // Handle
      DstDat    :='WritingDat',                 // Source variable name
      Size      :=UINT#1,                       // Number of elements to write
      SrcDat    :=WriteDat);                    // Write data

    IF (CIPWrite_instance.Done=TRUE) THEN
      Stage     :=INT#3;                        // Normal end
    ELSIF (CIPWrite_instance.Error=TRUE) THEN
      Stage     :=INT#20;                       // Error end
    END_IF;

  3 :                                           // Request reading value of variable.
    CIPRead_instance(
      Execute   :=TRUE,
      Handle    :=CIPOpen_instance.Handle,      // Handle
      SrcDat    :='OriginalDat',                // Source variable name
      Size      :=UINT#1,                       // Number of elements to read
      DstDat    :=ReadDat);                     // Read data

    IF (CIPRead_instance.Done=TRUE) THEN
      Stage     :=INT#4;                        // Normal end
    ELSIF (CIPRead_instance.Error=TRUE) THEN
      Stage     :=INT#30;                       // Error end
    END_IF;
    4 :                                                // Send message
        ReqPath.ClassID   :=UINT#01;
        ReqPath.InstanceID :=UINT#01;
        ReqPath.isAttributeID:=TRUE;
        ReqPath.AttributeID :=UINT#07;
        CIPSend_instance(
            Execute     :=TRUE,
            Handle      :=CIPOpen_instance.Handle,     // Handle
            ServiceCode:=BYTE#16#0E,                   // Service code
            RqPath      :=ReqPath,                     // Request path
            ServiceDat :=Dummy,                        // Service data
            Size        :=UINT#0,                      // Number of elements
            RespServiceDat:=RespDat);                  // Response data

        IF (CIPSend_instance.Done=TRUE) THEN
            Stage       :=INT#5;                       // Normal end
        ELSIF (CIPSend_instance.Error=TRUE) THEN
            Stage       :=INT#40;                      // Error end
        END_IF;

    5 :                                                // Request closing CIP class 3 connection.
        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 Using CIP Message Communications

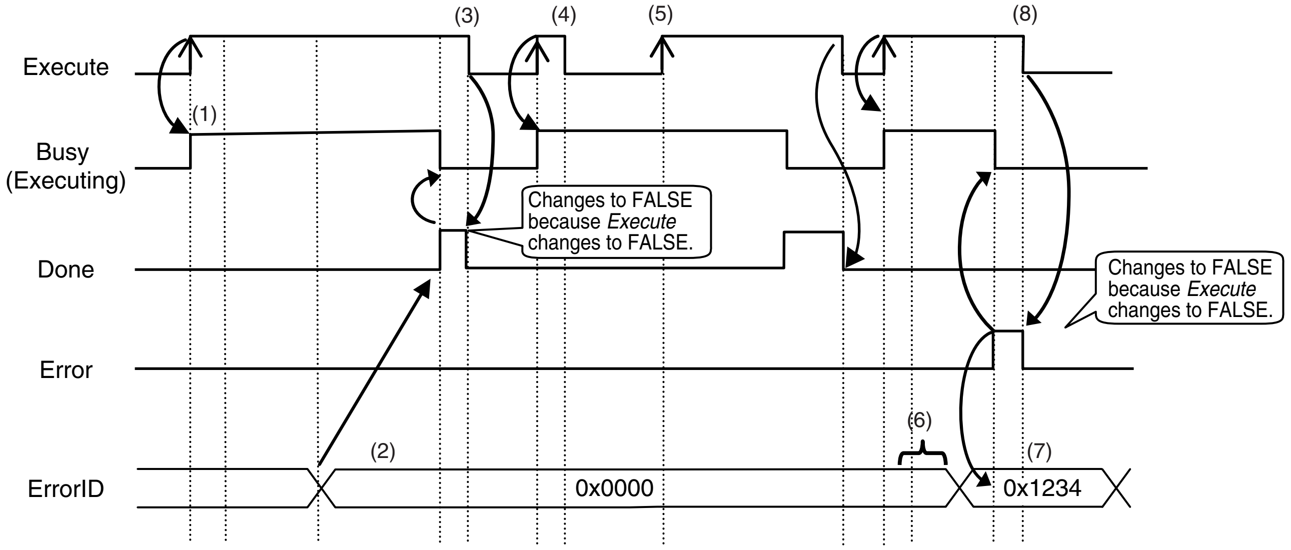
8

8-2-7 Sample Programming for CIP Connection (Class 3) Message Communications

## 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.

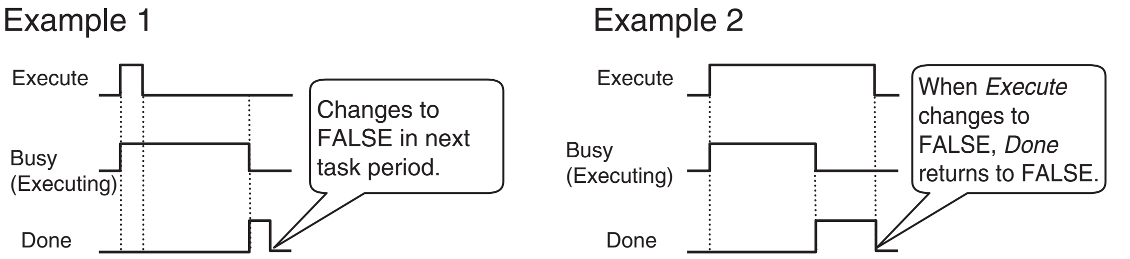


(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
Execute
Busy (Executing)
Done
Error
ErrorID
0x0000
0x1234
Changes to FALSE because *Execute* changes to FALSE.
Changes to FALSE because *Execute* changes to FALSE.

1. When *Execute* changes to TRUE, the instruction is executed and *Busy* changes to TRUE.
2. After the results of instruction execution are stored in the output variables, *Done* changes to TRUE and *Busy* changes to FALSE.
3. When *Execute* changes to FALSE, *Done* returns to FALSE.
4. When *Execute* changes to TRUE again, *Busy* changes to TRUE.
5. *Execute* is ignored if it changes to TRUE during instruction executed (i.e., when *Busy* is TRUE).
6. If an error occurs, several retries are attempted internally. The error code in the *ErrorID* is not updated during the retries.
7. When a communications error occurs, *Error* changes to TRUE and the *ErrorID* is stored. Also, *Busy* and *Done* change to FALSE.
8. When *Execute* changes to FALSE, *Error* changes to FALSE.

**Precautions for Correct Use**

If *Execute* changes back to FALSE before *Done* changes to TRUE, *Done* stays TRUE for only one task period. (Example 1)
If you want to see if *Done* is TRUE at any time, make sure to keep *Execute* TRUE until you confirm that *Done* is TRUE. If *Execute* is TRUE until *Done* changes to TRUE, *Done* stays TRUE until *Execute* changes to FALSE. (Example 2)

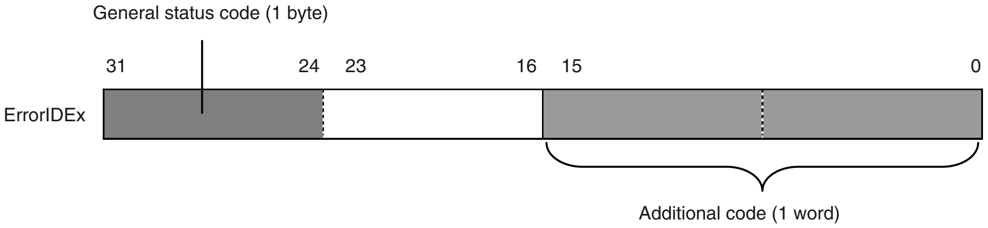

Example 1
Execute
Busy
(Executing)
Done
Changes to FALSE in next task period.
Example 2
Execute
Busy
(Executing)
Done
When *Execute* changes to FALSE, *Done* returns to FALSE.

## 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. |

| General status code (hex) | Status name | Description of status |
|---|---|---|
| 08 | Service not supported | The requested service was not supported or was not defined for this object class/instance. |
| 09 | Invalid attribute value | Invalid attribute data was detected. |
| 0A | Attribute list error | An attribute in the Get_Attribute_List or Set_Attribute_List response has a non-zero status. |
| 0B | Already in requested mode/state | The object is already in the mode/state being requested by the service. |
| 0C | Object state conflict | The object cannot perform the requested service in its current mode/state. |
| 0D | Object already exists | The requested instance of object to be created already exists. |
| 0E | Attribute not settable | A request to modify a non-modifiable attribute was received. |
| 0F | Privilege violation | A permission/privilege check failed. |
| 10 | Device state conflict | The device’s current mode/state prohibits the execution of the requested service. |
| 11 | Reply data too large | The data to be transmitted in the response buffer is larger than the allocated response buffer. |
| 12 | Fragmentation of a primitive value | The service specified an operation that is going to fragment a primitive data value, i.e. half a REAL data type. |
| 13 | Not enough data | The requested service did not supply enough data to perform the specified operation. |
| 14 | Attribute not supported | The attribute specified in the request is not supported. |
| 15 | Too much data | The service supplied more data than was expected. |
| 16 | Object does not exist | An object that does not exist was specified for the requested service. |
| 17 | Service fragmentation sequence not in progress | The fragmentation sequence for this service is not currently active for this data. |
| 18 | No stored attribute data | The attribute data of this object was not saved prior to the requested service. |
| 19 | Store operation failure | The attribute data of this object was not saved due to a failure during the attempt. |
| 1A | Routing failure (request packet too large) | The service request packet was too large for transmission on a network in the path to the destination. The routing device was forced to abort the service. |
| 1B | Routing failure (response packet too large) | The service response packet was too large for transmission on a network in the path from the destination. The routing device was forced to abort the service. |
| 1C | Missing attribute list entry data | The service did not supply an attribute in a list of attributes that was needed by the service to perform the requested behavior. |
| 1D | Invalid attribute value list | The service is returning the list of attributes supplied with status information for those attributes that were invalid. |
| 1E | Embedded service error | An embedded service resulted in an error. |
| 1F | Vendor specific error | A vendor-specific error occurred. The Additional Code Field of the error response defines the error. This is a general error code that is used only for errors that do not correspond to any of the error codes in this table and are not in an object class definition. |
| 20 | Invalid parameter | A parameter for the requested service is invalid. This code is used when a parameter does not meet the requirements of the specification and/or the requirements defined in an application object specification. |

| General status code (hex) | Status name | Description of status |
|---|---|---|
| 21 | Write-once value or medium already written | An attempt was made to write to a write-once medium (e.g. WORM drive or PROM) that was previously written or cannot be changed. |
| 22 | Invalid Reply Received | An invalid reply was received. (For example, the reply service code does not match the request service code, or the reply message is shorter than the minimum expected reply size.) This status code is used for other causes of invalid replies. |
| 23 and 24 | | Reserved by CIP for future extensions. |
| 25 | Key Failure in path | The key segment that was included as the first segment in the path does not match the destination module. The object specific status must indicate which part of the key check failed. |
| 26 | Path Size Invalid | The size of the path that was sent with the service request is either too large or too small for the request to be routed to an object. |
| 27 | Unexpected attribute in list | An attempt was made to set an attribute that is not able to be set at this time. |
| 28 | Invalid Member ID | The member ID specified in the request does not exist in the specified class, instance, and attribute. |
| 29 | Member not settable | A request to modify a non-modifiable member was received. |
| 2A | Group 2 only server general failure | This error code is reported only by group 2 only servers with 4K or less of code space and only in place of *Service not supported*, *Attribute not supported*, or *Attribute not settable*. |
| 2B to CF | | Reserved by CIP for future extensions. |
| D0 to FF | Reserved for Object Class and service errors | This range of error codes is to be used to indicate object class-specific errors. This code range is used only when none of the error codes in this table accurately reflect the error that occurred. The additional code field is used to describe the general error code in more detail. |

### ● Examples of Additional Status When General Status Is 01 Hex

(Status of Connection Manager Object)

| General Status (hex) | Additional Status (hex) | Description |
|---|---|---|
| 01 | 0100 | Connection in use or duplicate forward open. |
| 01 | 0103 | Transport class and trigger combination not supported. |
| 01 | 0106 | Ownership conflict. |
| 01 | 0107 | Connection not found at target application. |
| 01 | 0108 | Invalid connection type. There is a problem with either the connection type or priority of the connection. |
| 01 | 0109 | Invalid connection size. |
| 01 | 0100 | Device not configured. |
| 01 | 0111 | RPI not supported. May also indicate problem with connection time-out multiplier, or production inhibit time. |
| 01 | 0113 | Connection Manager cannot support any more connections. |
| 01 | 0114 | Either the vendor ID or the product code in the key segment does not match the device. |
| 01 | 0115 | Device type in the key segment does not match the device. |

| General Status (hex) | Additional Status (hex) | Description |
|---|---|---|
| 01 | 0116 | Major or minor revision information in the key segment does not match the device. |
| 01 | 0117 | Invalid connection point. |
| 01 | 0118 | Invalid configuration format. |
| 01 | 0119 | Connection request failed because there is no controlling connection currently open. |
| 01 | 011A | Target application cannot support any more connections. |
| 01 | 011B | RPI is smaller than the production inhibit time. |
| 01 | 0203 | Connection cannot be closed because the connection has timed out. |
| 01 | 0204 | Unconnected_Send service timed out while waiting for a response. |
| 01 | 0205 | Parameter error in Unconnected_Send service. |
| 01 | 0206 | Message too large for unconnected message service. |
| 01 | 0207 | Unconnected acknowledgement without reply. |
| 01 | 0301 | No buffer memory available. |
| 01 | 0302 | Network bandwidth not available for data. |
| 01 | 0303 | No tag filters available. |
| 01 | 0304 | Not configured to send real-time data. |
| 01 | 0311 | Port that was specified in port segment is not available. |
| 01 | 0312 | Link address that was specified in port segment is not available. |
| 01 | 0315 | Invalid segment type or segment value in path. |
| 01 | 0316 | Path and connection were not equal when closing the connection. |
| 01 | 0317 | Either the segment is not present or the encoded value in the network segment is invalid. |
| 01 | 0318 | Link address to self is invalid. |
| 01 | 0319 | Resources on secondary are unavailable. |
| 01 | 031A | Connection is already established. |
| 01 | 031B | Direct connection is already established. |
| 01 | 031C | Others |
| 01 | 031D | Redundant connection mismatch. |
| 01 | 031E | There are no more reception resources available on the sending module. |
| 01 | 031F | No connection resources exist for the target path. |
| 01 | 0320 to 07FF | Vendor specific. |

# 8-3 CIP Objects Sent to the EtherNet/IP Unit

## 8-3-1 CIP Objects Sent to the EtherNet/IP Unit

The following CIP objects can be sent to an EtherNet/IP Unit.

| Object name | Function | Reference |
|---|---|---|
| Identity object | • Reads ID information from the CPU Unit.<br>• Resets the EtherNet/IP Unit. | page 8-31 |
| TCP/IP interface object | • Writes and reads TCP/IP settings. | page 8-33 |
| Ethernet link object | • Reads Ethernet settings.<br>• Reads Ethernet status. | page 8-35 |
| Controller object | • Gets the Controller status.<br>• Changes the operating mode of the Controller. | page 8-39 |

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

This object reads the ID information of the CPU Unit and resets the EtherNet/IP Unit.

### ● Class ID

Specify 01 hex.

### ● Instance ID

Specify 00 or 01 hex.

### ● Attribute ID

The attribute ID specifies the information to read.

### Class Attribute ID

The class attribute ID specifies the attribute of the entire object.

| Attribute ID | Parameter name | Description | Attribute | Read data | |
|---|---|---|---|---|---|
| | | | | Data type | Value |
| 01 hex | Revision | Revision of the object | Read | UINT | 0001 hex |
| 02 hex | Max Instance | The maximum instance number | Read | UINT | 0001 hex |

### Instance Attribute ID

The instance attribute ID specifies the attribute of the instance.

| Attribute ID | Parameter name | Description | Attribute | Read data | |
|---|---|---|---|---|---|
| | | | | Data type | Value |
| 01 hex | Vendor ID | Vendor ID | Read | UINT | 002F hex |
| 02 hex | Device Type | Device type | Read | UINT | 000C hex |
| 03 hex | Product Code | Product code | Read | UINT | 0668 hex |
| 04 hex | Revision | Device revision | Read | Struct | --- |
| | Max Instance | Major revision | Read | USINT | Refer to *(1) Major CIP Revision and Minor CIP Revision* |
| | Revision | Minor revision | Read | USINT | |
| 05 hex | Status | Status of the EtherNet/IP Unit | Read | WORD | Refer to *(2) Status Details of the EtherNet/IP Unit* |

8-3 CIP Objects Sent to the EtherNet/IP Unit

8

8-3-1 CIP Objects Sent to the EtherNet/IP Unit

| 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) Major CIP Revision and Minor CIP Revision

| Unit version | CIP revision | |
|---|---|---|
| | Major revision | Minor revision |
| Unit Ver. 2.1 | 02 hex | 04 hex |
| Unit Ver. 3.0 | 03 hex | 01 hex |

### (2) Status Details of the EtherNet/IP Unit

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>0</td><td>Owned</td><td>Indicates when the EtherNet/IP Unit has an open connection as the target of a tag data link.</td></tr>
<tr><td>1</td><td>Reserved</td><td>Always FALSE.</td></tr>
<tr><td>2</td><td>Configured</td><td>Tag data link settings exist.</td></tr>
<tr><td>3</td><td>Reserved</td><td>Always FALSE.</td></tr>
<tr><td>4 to 7</td><td>Extended Device Status</td><td>Indicates the status of the EtherNet/IP Unit.
<table>
<tr><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th></th></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>There is a major fault.</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>A timeout occurred in one or more target connections.</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>Indicates that there are no tag data link settings.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>Indicates that one or more connections are performing communications normally.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>Other than the above.</td></tr>
</table></td></tr>
<tr><td>8</td><td>Minor Recoverable Fault</td><td>TRUE when any of the following errors occurs.<br>• TCP/IP Advanced Setting Error<br>• DNS Server Connection Error<br>• Tag Data Link Setting Error<br>• Tag Data Link Error<br>• Tag Data Link Connection Failed<br>• FTP Server Settings Error<br>• NTP Client Setting Error<br>• SNMP Settings Error<br>• NTP Server Connection Error<br>• Tag Resolution Error</td></tr>
<tr><td>9</td><td>Minor Unrecoverable Fault</td><td>TRUE when the following error occurs.<br>• Identity Error</td></tr>
<tr><td>10</td><td>Major Recoverable Fault</td><td>TRUE when any of the following errors occurs.<br>• IP Address Duplication Error<br>• BOOTP Server Error<br>• Basic Ethernet Setting Error<br>• TCP/IP Basic Setting Error</td></tr>
<tr><td>11</td><td>Major Unrecoverable Fault</td><td>TRUE when any of the following errors occurs.<br>• Communications Controller Error<br>• MAC Address Error</td></tr>
<tr><td>12 to 15</td><td>Reserved</td><td>Always FALSE.</td></tr>
</table>

### 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 EtherNet/IP Unit. This parameter is used to reset the EtherNet/IP Unit 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.<br>00 hex: Restart the EtherNet/IP Unit.<br>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 | Not required. |
| 0E hex | Get_Attribute_Single | | • Specifying a service for an instance: Always 01 hex | • Reading a class attribute: 01 or 02 hex<br>• 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

Specify 00 or 01 hex.

### Attribute ID

The attribute ID specifies the information to read.

### Class Attribute ID

The class attribute ID specifies the attribute of the entire object.

| Attribute ID | Parameter name | Description | Attribute | Read data | |
|---|---|---|---|---|---|
| | | | | Data type | Value |
| 01 hex | Revision | Revision of the object | Read | UINT | 0001 hex |
| 02 hex | Max Instance | The maximum instance number | Read | UINT | 0001 hex |

### Instance Attribute ID

The instance attribute ID specifies the attribute of the instance.

| Attribute ID | Parameter name | Description | Attribute | Read/write data | |
|---|---|---|---|---|---|
| | | | | Data type | Value |
| 01 hex | Interface Configuration Status | Indicates the IP address settings status of the EtherNet/IP Unit. | Read | DWORD | Bits 0 to 3: Interface Configuration Status<br>0 = IP address is not set. (This includes when BOOTP is starting.)<br>1 = IP address is set.<br>Bits 4 to 31: Reserved (always FALSE). |
| 02 hex | Configuration Capability | Indicates a Controller Configurations and Setup that can be set to the EtherNet/IP Unit. | Read | DWORD | Bit 0: BOOTP Client: Always TRUE.<br>Bit 1: DNS Client: Always TRUE.<br>Bit 2: DHCP Client: Always FALSE.<br>Bit 3: DHCP-DNS Update: Always FALSE.<br>Bit 4: Configuration Settable: Always TRUE.<br>Bits 5 to 31: Reserved (always FALSE). |
| 03 hex | Configuration Control | Sets the method used to set the IP address when the EtherNet/IP Unit starts. | Write | DWORD | Bit 0: Static IP address.<br>Bit 1: Set by BOOTP. |
| 04 hex | Physical Link Object | The path to the link object in the physical layer. | Read | Struct | --- |
| | Path size | The path size (WORD size). | | UINT | 0002 hex |
| | Path | The path to the link object in the physical layer (static). | | EPATH | 20 F6 24 01 hex |
| 05 hex | Interface Configuration | The EtherNet/IP Unit 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 . | | UDINT | Set value |
| | Domain Name | The domain name. | | STRING | Set value |
| 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. | No 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 EtherNet/IP Unit 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<br>• Specifying a service for an instance: 01 hex | Not required. |
| 0E hex | Get_Attribute_Single | | | • Reading a class attribute: 01 or 02 hex<br>• 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

Specify 00 or 01 hex.

### Attribute ID

The attribute ID specifies the information to read.

#### Class Attribute ID

The class attribute ID specifies the attribute of the entire object.

| Attribute ID | Parameter name | Description | Attribute | Read data | |
|---|---|---|---|---|---|
| | | | | Data type | Value |
| 01 hex | Revision | Revision of the object | Read | UINT | 0002 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 EtherNet/IP Unit. | Read | UDINT | Reads the current value. |
| 02 hex | Interface Flags | Gives the status of the EtherNet/IP Unit. | Read | DWORD | Refer to *(1) Interface Flag Details*, below. |
| 03 hex | Physical Address | Gives the MAC address of the EtherNet/IP Unit. | 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.<br>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 EtherNet/IP Unit. | Read | Struct | --- |
| | Alignment Errors | Number of frames received that were not octets in length. | | UDINT | Reads the current value. |
| | FCS Errors | Number of frames received that did not pass the FCS check. | | UDINT | Reads the current value. |
| | Single Collisions | Number of frames sent successfully with only one collision. | | UDINT | Reads the current value. |
| | Multiple Collisions | Number of frames sent successfully with two or more collisions. | | UDINT | Reads the current value. |
| | SQE Test Errors | Number of times a SQE test error message was generated. | | UDINT | Reads the current value. |
| | Deferred Transmissions | The number of frames for which the first attempt to send was delayed because the media was busy. | | UDINT | Reads the current value. |
| | Late Collisions | The number of collisions detected in packets that were sent after 512 bit times. | | UDINT | Reads the current value. |
| | Excessive Collisions | The number of frames that failed to be sent because of excessive collisions. | | UDINT | Reads the current value. |
| | MAC Transmit Errors | The number of frames that failed to be sent due to an internal MAC sublayer transmission error. | | UDINT | Reads the current value. |
| | Carrier Sense Errors | The number of times the carrier sense condition was lost or the number of times an assertion did not occur when an attempt was made to send the frame. | | UDINT | Reads the current value. |
| | Frame Too Long | The number of frames received that exceeded the maximum allowed frame size. | | UDINT | Reads the current value. |
| | MAC Receive Errors | The number of frames that could not be received through the interface due to an internal MAC sublayer reception error. | | UDINT | Reads the current value. |
| 06 hex | Interface Control | Counter for the EtherNet/IP Unit. | Write | Struct | --- |
| | Software Switches | *Auto Nego* for Ethernet communications that specifies full duplex. | | WORD | Refer to *(2) Software Switch 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.<br>TRUE: The link is up. |
| 1 | Half/FullDuplex | FALSE: Half duplex<br>TRUE: Full duplex |
| 2 to 4 | Negotiation Status | 00 hex: Auto-negotiation is in progress.<br>01 hex: Auto-negotiation and speed detection failed.<br>02 hex: Auto-negotiation failed, but speed detection succeeded.<br>03 hex: Speed and duplex mode negotiation succeeded.<br>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) Software Switch Details

| Bit | Name | Description |
|---|---|---|
| 0 | Auto-negotiate | FALSE: Auto-negotiation is disabled (communications setup is always set to 10 Mbps).<br>TRUE: Auto-negotiation is enabled (communications setup is automatically set to 100 or 10 Mbps). |
| 1 | ForcedDuplex Mode | FALSE: Half duplex<br>TRUE: Full duplex<br>* 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<br>• Specifying a service for an instance: Always 01 hex | Not required. |
| 10 hex | Set_Attribute_Single | | | • Reading a class attribute: 01 or 02 hex<br>• 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

Specify 00 hex.

### ● Attribute ID

The attribute ID specifies the information to read.

#### Class Attribute ID

The class attribute ID specifies the attribute (value) of the entire object.

| Attribute ID | Parameter name | Description | Attribute | Read/write data | |
|---|---|---|---|---|---|
| | | | | Data type | Value |
| 01 hex | Revision | Revision of the object | Read | UINT | Always 0002 hex. |
| 02 hex | Max Instance | The maximum instance number | Read | UINT | Always 0001 hex |
| 64 hex | PLC Mode | This can be used to read and modify the Controller operating mode. | Write | UINT | Specify this when you want to write to an attribute.<br>0000 hex: PROGRAM mode<br>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.<br>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. |

### ● 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 | | | |

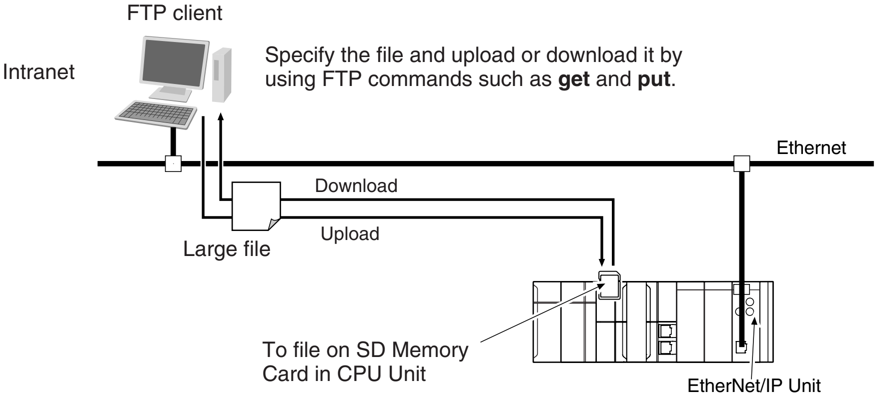
9

# FTP Server

# 9-1 Overview and Specifications

## 9-1-1 Overview

The EtherNet/IP Unit 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.


FTP client
Intranet
Specify the file and upload or download it by using FTP commands such as **get** and **put**.
Ethernet
Download
Upload
Large file
To file on SD Memory Card in CPU Unit
EtherNet/IP Unit

## 9-1-2 Specifications

| Item | Specification |
|---|---|
| Executable commands | open: Connects the specified host FTP server.<br>user: Specifies the user name for the remote FTP server.<br>ls: Displays file names in the remote host.<br>dir: Displays file names and details in the remote host.<br>rename: Changes a file name.<br>mkdir: Creates a new directory in the working directory in the remote host.<br>rmdir: Deletes a directory from the working directory in the remote host.<br>cd: Changes the work directory in the remote host to the specified directory.<br>pwd: Displays the work directory on the remote host.<br>type: Specifies the data type of transferred files.<br>get: Transfers the specified remote file to the local host.<br>mget: Transfers the specified multiple remote files to the local host.<br>put: Transfers the specified local file to the remote host.<br>mput: Transfers the specified multiple local files to the remote host.<br>delete: Deletes the specified file from the remote host.<br>mdelete: Deletes the specified multiple files from the remote host.<br>close: Disconnects the FTP server.<br>bye: Closes the FTP client.<br>quit: Closes the FTP client. |
| Protection | Login name<br>• CJ1W-EIP21: 1 to 12 single-byte alphanumeric characters<br>• CJ1W-EIP21S: 1 to 16 single-byte alphanumeric characters<br>Password (Up to 8 characters)<br>• CJ1W-EIP21: 1 to 8 single-byte alphanumeric characters<br>• CJ1W-EIP21S: 8 to 16 single-byte alphanumeric characters |
| Protocol used | FTP (port number: 20/TCP, 21/TCP) |
| Number of connections | 1 |

# 9-2 FTP Server Function Details

## 9-2-1 Supported Files

The file system in the Controller that can be accessed by the EtherNet/IP Unit includes files in any SD Memory Card mounted in the CPU Unit. The directory tree is shown below.

```
/: root
 └── MEMCARD: SD
```

A connection is initially made to the root directory.

**Additional Information**

- The date of the MEMCARD directory displayed for **ls, dir,** or **mkdir** commands in the root directory is the date of the file system volume label.
- The login date is displayed for MEMCARD if a volume label has not been created.

## 9-2-2 Connecting to the FTP Server

The login name and password set in the Unit settings for the CPU Bus Unit will be used to connect.

**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 EtherNet/IP Unit Settings to set any login name and password.

### ● Login Messages

| Status | Message |
|---|---|
| Normal connection | 220 xxx.xx.xx.xx yyyyyyyyyy FTP server (FTP Version z.zz) ready.<br>xxx.xx.xx.xx: IP address of EtherNet/IP Unit<br>yyyyyyyyyy: EtherNet/IP Unit model number (example: CJ1W-EIP21)<br>z.zz: Firmware version of the EtherNet/IP Unit |
| FTP server busy | 221 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.
- Login name
  CJ1W-EIP21: 1 to 12 single-byte alphanumeric characters
  CJ1W-EIP21S: 1 to 16 single-byte alphanumeric characters
- Password
  CJ1W-EIP21: 1 to 8 single-byte alphanumeric characters
  CJ1W-EIP21S: 8 to 16 single-byte alphanumeric 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">First access<br>Later 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="4">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>

# 9-3 Using the FTP Server Function

## 9-3-1 Procedure

**1** Make the basic settings.

Refer to *1-5-1 Basic Operation* for basic operations.

**2** Set up the FTP server on the Sysmac Studio. (Refer to *5-4 FTP Settings Display*.)

**3** Select **EtherNet/IP Unit 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 EtherNet/IP Unit from an FTP client.

**7** Input the FTP login name and password that you set in the EtherNet/IP Unit Settings to log in to the EtherNet/IP Unit.

**8** After you are logged in, you can use the ftp commands, such as cd (Change Directory) and get (Obtain File) for the MEMCARD1 directory in the SD Memory Card in the Controller.

**9** Close the connection.

## 9-3-2 List of Settings Required for the FTP Server Function

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

| EtherNet/IP Unit Settings Tab Page on Sysmac Studio | Setting | Setting conditions | Reference |
|---|---|---|---|
| FTP | FTP server | Required | page 5-9 |
| | Port No. | Optional*<br>**Note** Required when changing the default value of 21. | |
| | Login name | Required* | |
| | Password | Required* | |

* Settings are not required if the FTP server is not used.

**Additional Information**

Make the settings in the FTP Settings Dialog Box if the FTP server is used. Refer to *5-4 FTP Settings Display* for information on the FTP Settings Dialog Box.

# 9-4 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.

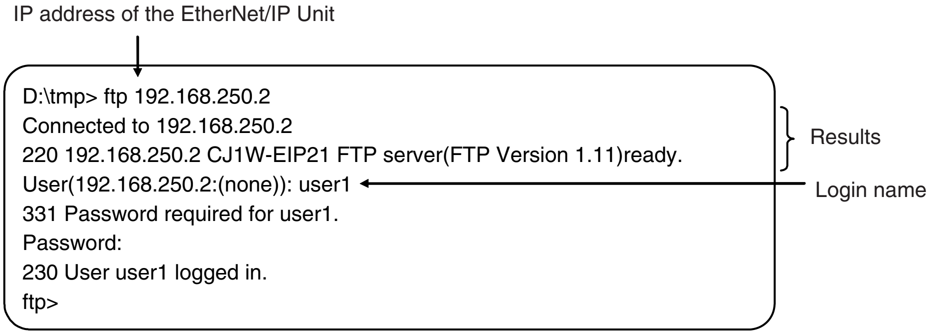
**Additional Information**

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

## ● Step

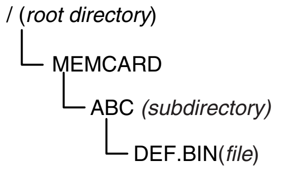
**1** Make sure that an SD Memory Card is inserted in the CPU Unit and turn ON the power supply to the Controller.

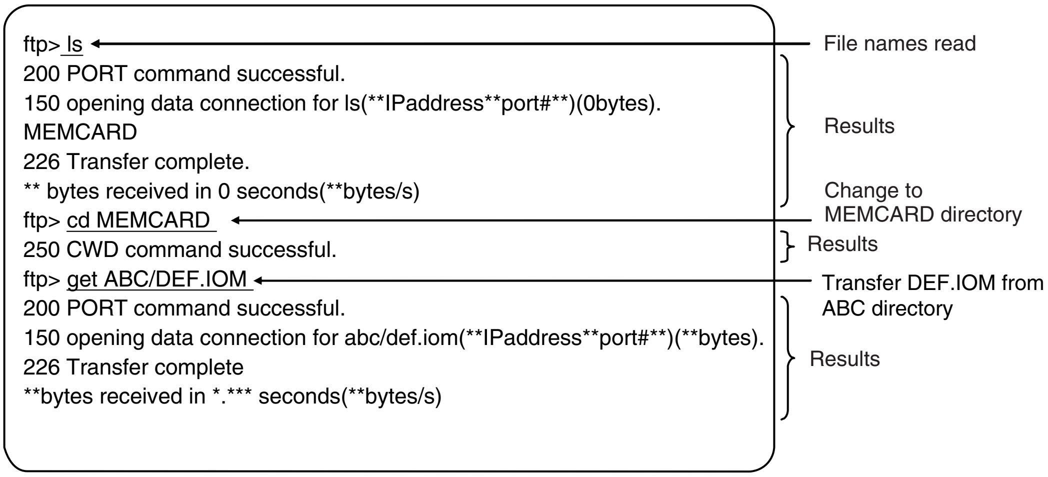
**2** Connect to the FTP server from a computer on the Ethernet by entering the text that is underlined in the following diagram.



```
D:\tmp> ftp 192.168.250.2
Connected to 192.168.250.2
220 192.168.250.2 CJ1W-EIP21 FTP server(FTP Version 1.11)ready.
User(192.168.250.2:(none)): user1
331 Password required for user1.
Password:
230 User user1 logged in.
ftp>
```

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





```
ftp> ls
200 PORT command successful.
150 opening data connection for ls(**IPaddress**port#**)(0bytes).
MEMCARD
226 Transfer complete.
** bytes received in 0 seconds(**bytes/s)
ftp> cd MEMCARD
250 CWD command successful.
ftp> get ABC/DEF.IOM
200 PORT command successful.
150 opening data connection for abc/def.iom(**IPaddress**port#**)(**bytes).
226 Transfer complete
**bytes received in *.*** seconds(**bytes/s)
```

# 9-5 Using FTP Commands

This section describes the FTP commands which the host computer (FTP client) can send to the FTP server of the EtherNet/IP Unit. The descriptions should also apply to most workstations, but slight differences may arise. Refer to your workstation's operation manuals for details.

## 9-5-1 Table of Commands

The FTP commands which can be sent to the EtherNet/IP Unit 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. |
| dir | 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. |
| close | Disconnects the FTP server. |
| bye | Closes the FTP client. |
| quit | Closes the FTP client. |

**Note 1** "Remote host" refers to the EtherNet/IP Unit.

**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).

## 9-5-2 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 EtherNet/IP Unit system setup.

### 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.

### 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.

## 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.

## 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 MEMCARD directory under the root directory (/).
- The root directory (/) is the directory that is used when you log onto the EtherNet/IP Unit.
  The MEMCARD 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.

- If an error occurs during file transfer, the file being transferred is deleted and the transmission of that file ends in an error.

## 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.
- If an error occurs during file transfer, the file being transferred is deleted and the transmission of that file ends in an error. However, *mput* execution continues and the remaining files are transferred.

## 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 EtherNet/IP Unit.

## bye

### Format

bye

### Function

- Ends the FTP sessions.

## quit

### Format

quit

### Function

- Ends the FTP sessions.

# 9-6 FTP Server Status

## 9-6-1 FTP Status

You can use the following device variable for the CJ-series Unit to see whether the FTP server is operating.

| Device variable name for CJ-series Unit | | Data type | R/W | Meaning | Description |
|---|---|---|---|---|---|
| *_CommSta2 | | WORD | R | Communications Status 2 | Bit 00: FTP Status<br>Bit 14: Link Status |
| | *_FTPSta | BOOL | R | FTP Status | TRUE: FTP server is operating. (The FTP client is connected.)<br>FALSE: FTP server is not connected. (The FTP client is waiting for a connection.) |

### Precautions for Correct Use

File operations for files on the SD Memory Card are performed during FTP communications. Do not remove the SD Memory Card or turn OFF power to the Controller while FTP is being used.

### Additional Information

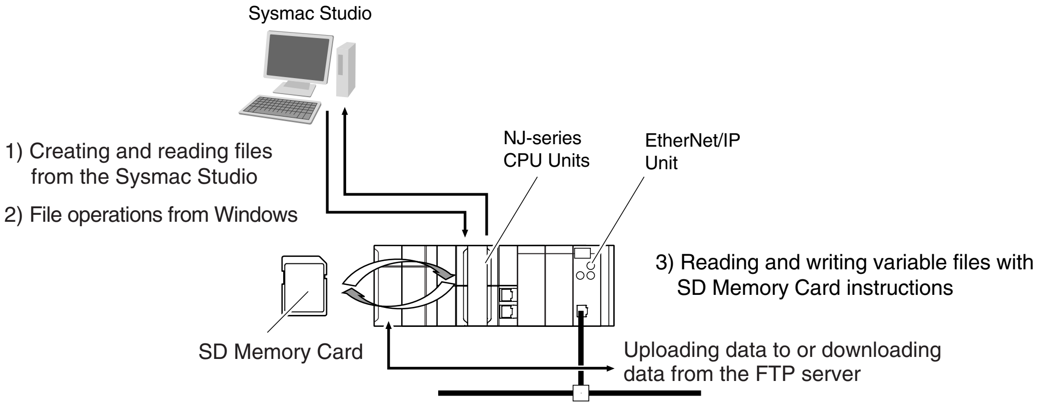
When using File Memory Instruction from the program in the CPU Unit, program exclusive control using the FTP Status variable so that the same data is not manipulated simultaneously by more than one instruction.

# 9-7 Using SD Memory Card Operations

The EtherNet/IP Unit 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.


Sysmac Studio
1) Creating and reading files from the Sysmac Studio
2) File operations from Windows
NJ-series CPU Units
EtherNet/IP Unit
3) Reading and writing variable files with SD Memory Card instructions
SD Memory Card
Uploading data to or downloading data from the FTP server

## 9-7-1 SD Memory Card Types

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

| Model number | Card type | Capacity [GB] | Format | Number of overwrites | Weight |
|---|---|---|---|---|---|
| HMC-SD292 | SD | 2 | FAT16 | 50,000 writes | 2 g max. |
| HMC-SD492 | SDHC | 4 | FAT32 | 100,000 writes | |
| HMC-SD1A2 | | 16 | | 100,000 writes | |

## 9-7-2 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. (Lowercase characters are converted to uppercase characters.)

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 is eight characters for the name and three characters for the extension. 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 *9-7-2 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/NX-series CPU Unit Software User’s Manual* (Cat. No. W501) for details.

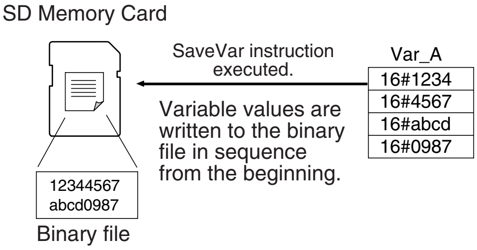
## 9-7-3 Initializing SD Memory Cards

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

## 9-7-4 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.


SD Memory Card
SaveVar instruction executed.
Var_A
16#1234
16#4567
16#abcd
16#0987
Variable values are written to the binary file in sequence from the beginning.
12344567
abcd0987
Binary file

**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 *9-5-2 Using the Commands*.)
- For details on how to use ladder diagram instructions to process files, refer to the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502).

# 9-8 FTP File Transfer Time

File transfers using FTP can require 30 or 40 minutes depending on the capacity of the file. Approximate file transfer time are provided in the following table for reference. All times are in seconds unless otherwise specified.

Example: Transfer Times in RUN Mode with a Task Period of 1 ms

| Process type | Transfer size | Transfer time |
|---|---|---|
| *put* transfer length (bytes) | 1K | 2.0 |
| | 5K | 10.1 |
| | 10K | 19.6 |
| | 40K | 79.7 |
| *get* transfer length (bytes) | 1K | 10.3 |
| | 5K | 51.4 |
| | 10K | 107.0 |
| | 40K | 409.7 |

(Unit: s)

# 9-9 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 EtherNet/IP Unit 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 EtherNet/IP Unit.

```
$ ftp nj
connected to nj
220 **IPaddress** NJ501-1300 FTP server(FTP**version**)ready
Name(nj:root): ■
```

FTP started.

**2** Enter the login name.

```
Name(nj:root):LogIn
331 Password required for LogIn.
Password:
230 LogIn logged in.
ftp> ■
```

Enter the login name.

Enter the password.

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

```
ftp> ls
200 PORT command successful.
150 opening data connection for ls(**IPaddress**port#**)(0 bytes).
MEMCARD
226 Transfer complete.
15 bytes received in 0 seconds(**bytes/s)
ftp> ■
```

Make sure the Memory Card is inserted.

**4** Change to the MEMCARD directory.

```
ftp> cd MEMCARD
250 CWD command successful.
ftp> ■
```

Change the directory.

**5** Change data type to binary.

```
ftp> type binary
200 Type set to I.
ftp> ■
```

Set binary data type.

**6** Read the file RESULT.BIN and transfer it to the workstation.

```
ftp> get RESULT.BIN
200 PORT command successful.
150 opening data connection for result.bin (**IPaddress**port#**) (**bytes).
226 Transfer complete.
** bytes received in *.*** seconds (**bytes/s)
ftp> ■
```

Read file.

**7** Write the file PLAN.BIN to the 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.

# 10

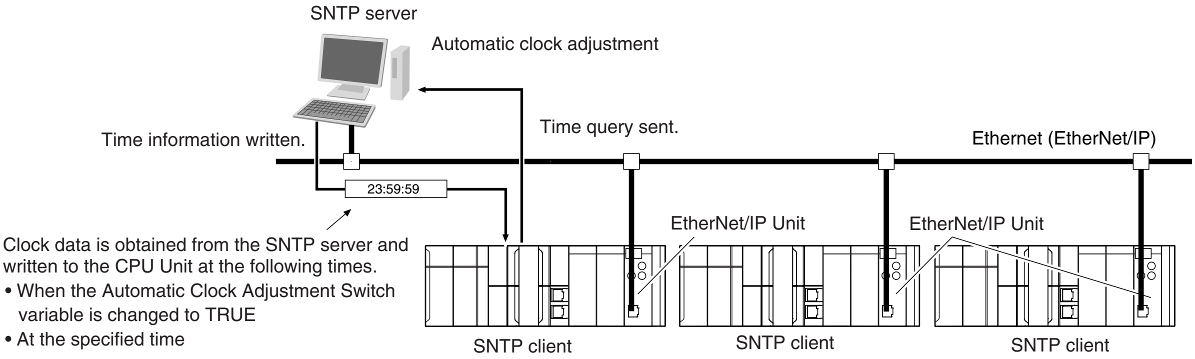
# Automatic Clock Adjustment

# 10-1 Automatic Clock Adjustment

## 10-1-1 Overview

With the EtherNet/IP Unit, clock information is read from the SNTP server at the specified time or when the Clock Information Adjustment Switch variable is changed to TRUE. The internal clock time in the CPU Unit of the EtherNet/IP Unit is updated with the read time.

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


SNTP server
Automatic clock adjustment
Time information written.
Time query sent.
Ethernet (EtherNet/IP)
23:59:59
EtherNet/IP Unit
EtherNet/IP Unit
Clock data is obtained from the SNTP server and written to the CPU Unit at the following times.
• When the Automatic Clock Adjustment Switch variable is changed to TRUE
• At the specified time
SNTP client
SNTP client
SNTP client

**Note** In accordance with SNTP protocol specifications, automatic adjustment will not be possible from February 7, 2036. This function will no longer operate in the EtherNet/IP Unit from February 7, 2036 (an error message will not be displayed).

## 10-1-2 Specifications

| Item | Specification | |
|---|---|---|
| Protocol | SNTP | |
| Port No. | 123 (UDP)<br>However, you can change the port number in the EtherNet/IP Unit Settings on the Sysmac Studio. | |
| Access to SNTP server | Writes the clock information from the SNTP 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. |
| SNTP Operation Timing | Clock information is automatically updated at the following times if the SNTP function is used.<br>• When the Adjust Clock Bit variable is changed to TRUE<br>• At the specified time | |

# 10-2 Procedure to Use the Automatic Clock Adjustment Function

## 10-2-1 Procedure

**1** Make the basic settings.

Refer to *1-5 EtherNet/IP Communications Procedures* for the flow of basic operations.

**2** Make the following settings in the Special Unit Setup in the Controller Configurations and Setup of the Sysmac Studio. Set the following on the SNTP Settings Display.

- SNTP server settings (required)
- Specified time to access the SNTP server

**3** To manually adjust the clock information, change the Adjust Clock Bit (**_AdjTmCmd*) variable to TRUE.

**4** Select ***Synchronization*** from the Controller Menu. The EtherNet/IP Unit settings are transferred to the CPU Unit.

## 10-2-2 Settings Required for Automatic Clock Adjustment

The following EtherNet/IP Unit Settings are made from the Sysmac Studio to use automatic clock adjustment.

| Tab page | Setting | | Setting conditions | Reference |
|---|---|---|---|---|
| SNTP | SNTP server clock information | | Required. | page 5-10 |
| | Port No. | | Specified by user.<br>**Note** Required to change from the default value of 123. | |
| | Server specifying method | | Required. | |
| | | IP address | One of these must be set, depending on the *Server specification type* setting. | |
| | | Host name | | |
| | Time | | Required. | |
| | Timeout time | | Specified by user.<br>**Note** Required to change from the default value of 10 seconds. | |
| | Time difference adjustment | | Specified by user.<br>**Note** Required to change from the default value of +0:0. | |

**Additional Information**

Make the settings in the SNTP Settings Dialog Box if automatic clock adjustment is used. Refer to *5-5 SNTP Settings Display* for information on the SNTP Settings Dialog Box.

## 10-2-3 Updating the Clock Information

When you change the following device variable for the CJ-series Unit to TRUE, the EtherNet/ IP Unit obtains the clock data from the SNTP server on the network, and applies it to the local CPU Unit. After applying the data, the variable automatically returns to FALSE.

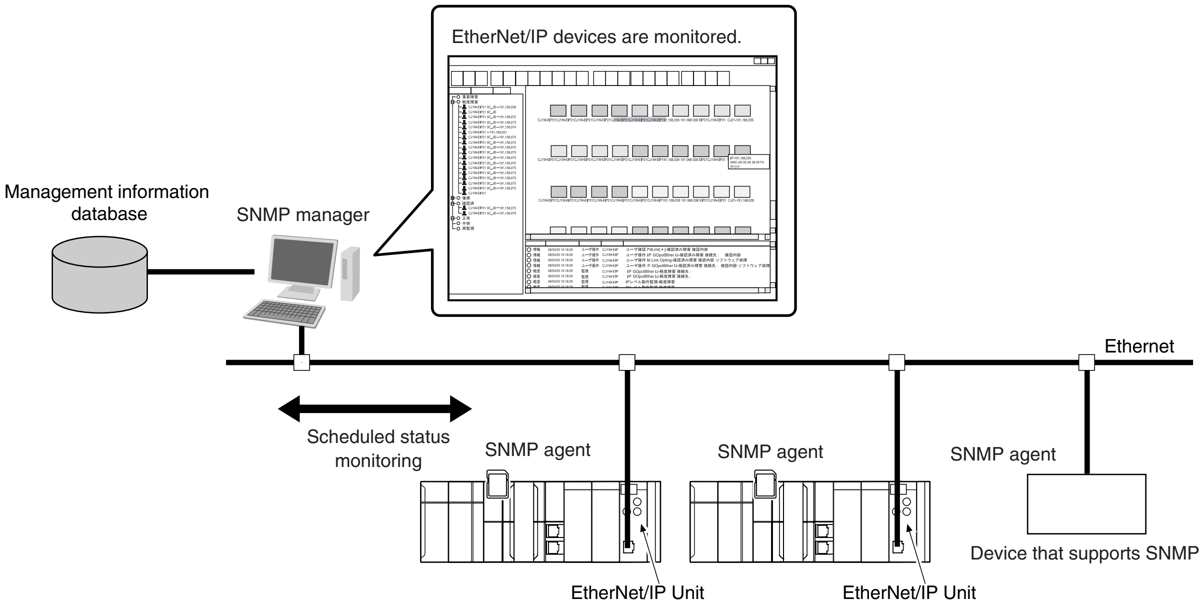
- **_AdjTmCmd* (Adjust Clock Bit)

# 11

# SNMP Agent

# 11-1 SNMP Agent

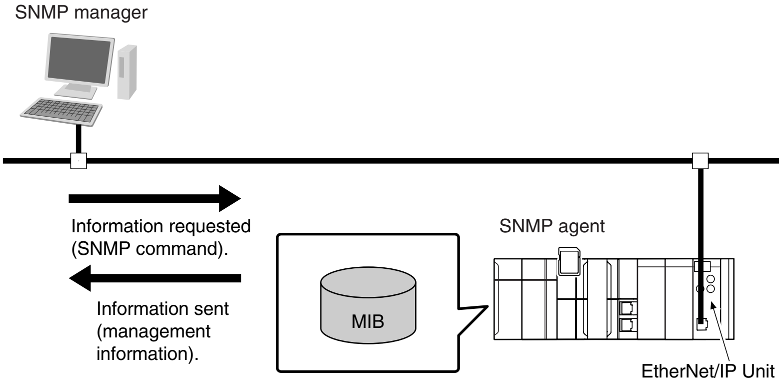
The SNMP (simple network management protocol) is a network management protocol. You can use the SNMP to manage any network that consists of devices that support SNMP. The server that manages the network is called the SNMP manager. The managed network devices are called SNMP agents.


EtherNet/IP devices are monitored.
Management information database
SNMP manager
Ethernet
Scheduled status monitoring
SNMP agent
SNMP agent
SNMP agent
Device that supports SNMP
EtherNet/IP Unit
EtherNet/IP Unit

## 11-1-1 Overview

### SNMP Agent

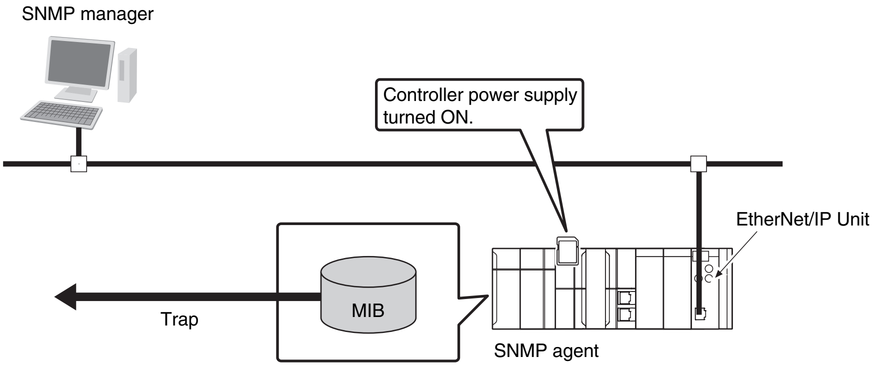
The EtherNet/IP Unit has its own management information called the MIB (management information base). This information can be provided to the SNMP manager. The SNMP manager is software that gathers and processes information about devices on the SNMP network and provides that information to the network administrator. You can use the SNMP manager to monitor the EtherNet/IP Unit.


SNMP manager
Information requested (SNMP command).
Information sent (management information).
MIB
SNMP agent
EtherNet/IP Unit

The SNMP manager has a SNMP command to request MIB information. The EtherNet/IP Unit SNMP agent function supports SNMPv1 (RFC1157) and SNMPv2C (RFC1901). Use the SNMPv1 or SNMPv2C protocol to manage the EtherNet/IP Unit 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 EtherNet/IP Unit 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.

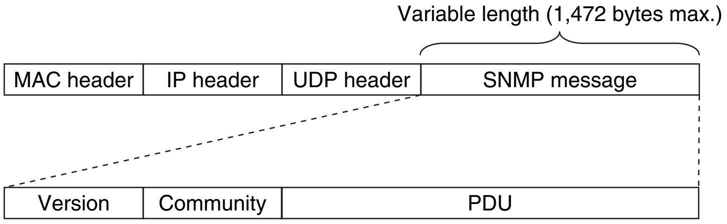


## 11-1-2 Specifications

| Item | Specification |
|---|---|
| Protocol | SNMP |
| Agent | SNMPv1, SNMPv2c |
| MIB | MIB-II |
| Port No. | SNMP agent: 161 (UDP)<br>SNMP trap: 162 (UDP)<br>These can be changed in the EtherNet/IP Unit Settings from the Sysmac Studio. |
| Timing of SNMP trap operation | Status reports are sent to the SNMP manager at the following times.<br>• When the Controller is turned ON<br>• When links are established<br>• When an SNMP agent fails to be authorized |
| Supported MIB commands | GetRequest/GetNextRequest |

## 11-1-3 SNMP Messages

The structure of SNMP messages is as follows:



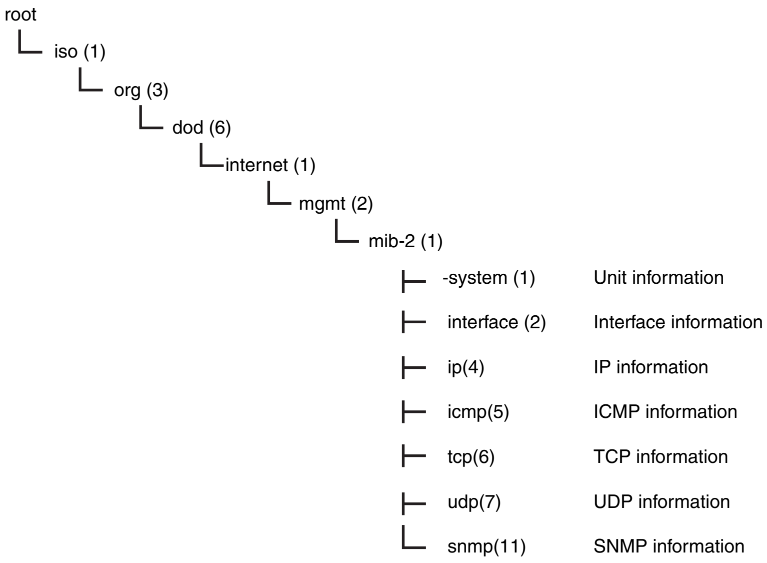
| Item | Set value |
|---|---|
| Version | This value gives the SNMP version.<br>SNMPv1: 0<br>SNMpv2c: 1 |
| Community | Community name for verification |
| PDU | This depends on the PDU type. |

## 11-1-4 MIB Specifications

This section describes the specifications of the MIB that is supported by the EtherNet/IP Unit.

### MIB System Diagram

The EtherNet/IP Unit 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<br>Device information (including hardware, OS, software names, and versions)<br>ASCII characters only. | Supported. | “OMRON Corporation” + Unit model |
| sysObjectID | (2) RO<br>Vendor OID.<br>Tells where this device information was assigned in the private MIB. | Supported. | 1.3.6.1.4.1.16838.1.1025.2 |
| sysUpTime | (3) RO<br>The time elapsed since the system was started (unit: 1/100 s). | Supported. | According to the standard. |
| sysContact | (4) RW<br>How to contact the administrator and information on the administrator. | Supported. | Set by the user. |
| sysName | (5) RW<br>The name for management. Sets the full domain name of the device. | Supported. | Unit name |
| sysLocation | (6) RW<br>The physical location of the device. | Supported. | Set by the user. |
| sysServices | (7) RO<br>The value of the provided service. | Supported. | 64 |

### ● *Interfaces* Group

| Subtree name | | | Standard [(*identifier*) *attribute*] | Support | Implementation specifications |
|---|---|---|---|---|---|
| ifNumber | | | (1) RO<br>The number of network interfaces. | Supported. | 1 |
| | ifTable | | (2) NA<br>Interface entity table | --- | |
| | | ifEntry | (1) NA<br>Row data for interface information<br>The index is *ifIndex*. | --- | |
| | | ifIndex | (1) RO<br>A number used to identify the interface. | Supported. | 1 |
| | | ifDescr | (2) RO<br>Information related to the interface (includes manufacturer name, product name, and hardware interface version). | Supported. | 10/100M Fast Ethernet Port |
| | | ifType | (3) RO<br>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<br>MTU value<br>The maximum size (in octets) of datagrams that can be sent and received through this interface. | Supported. | 1500 |
| | | ifSpeed | (5) RO<br>Estimated bandwidth<br>If a stable, accurate value cannot be obtained for the bandwidth, a nominal value is set instead. | Supported. | 10000000 |
| | | ifPhysAddress | (6) RO<br>MAC address<br>The physical address under the network layer of the interface. | Supported. | The MAC address of the EtherNet/IP port. |
| | | ifAdminStatus | (7) RW<br>The preferred status of the interface.<br>You cannot send normal packets in the testing state.<br>up(1)<br>down(2)<br>testing(3) | Supported. | According to the standard. |
| | | ifOperStatus | (8) RO<br>The current status of the interface.<br>You cannot send normal packets in the testing state.<br>up(1)<br>down(2)<br>testing(3) | Supported. | According to the standard. |
| | | ifLastChange | (9) RO<br>The *sysUpTime* (in 0.01seconds) at the last change in *ifOperStatus* for this interface. | Supported. | According to the standard. |
| | | ifInOctets | (10) RO<br>The number of octets received through this interface. This includes framing characters. | Supported. | According to the standard. |
| | | ifInUcastPkts | (11) RO<br>The number of unicast packets reported to a higher level protocol. | Supported. | According to the standard. |
| | | ifInNUcastPkts | (12) RO<br>The number of non-unicast packets (broadcast or multicast packets) reported to a higher level protocol. | Supported. | According to the standard. |
| | | ifInDiscards | (13) RO<br>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<br>The number of packets discarded because they contained errors. | Supported. | According to the standard. |
| | | ifInUnknown Protos | (15) RO<br>The number of packets received, but discarded because they were of an illegal or unsupported protocol.<br>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<br>The number of octets of packets sent through this interface.<br>This includes framing characters. | Supported. | According to the standard. |
| | | ifOutUcast Pkts | (17) RO<br>The number of unicast packets sent by higher level protocols.<br>This includes discarded packets and unsent packets. | Supported. | According to the standard. |
| | | ifOutNUcast Pkts | (18) RO<br>The number of non-unicast packets sent by higher level protocols.<br>This includes discarded packets and unsent packets. | Supported. | According to the standard. |
| | | ifOutDiscards | (19) RO<br>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<br>The number of packets that could not be sent because of an error. | Supported. | According to the standard. |
| | | ifOutQLen | (21) RO<br>The size of the send packet queue (i.e., the number of packets). | Supported. | Always 0. |
| | | ifSpecific | (22) RO<br>The object ID that represents a reference to the media-specific MIB for the interface.<br>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<br>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.<br>Forwarding (1)<br>Not-forwarding (2) | Supported. | Not-forwarding (2) |
| IpDefaultTTL | (2) RW<br>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<br>The number of all IP datagrams that reached the interface, including errors. | Supported. | According to the standard. |
| IpInHdrErrors | (4) RO<br>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<br>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<br>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<br>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<br>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<br>The number of datagrams delivered to an IP user protocol (any higher level protocol, including ICMP). | Supported. | According to the standard. |
| ipOutRequests | (10) RO<br>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<br>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<br>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<br>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<br>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<br>The number of IP datagrams received that were successfully reassembled. | Supported. | According to the standard. |
| ipReasmFails | (16) RO<br>The number of IP datagrams received that were not successfully reassembled. | Supported. | According to the standard. |
| ipFragOKs | (17) RO<br>The number of IP datagrams that were successfully fragmented. | Supported. | According to the standard. |
| ipFragFails | (18) RO<br>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<br>The number of IP datagrams created as a result of fragmentation. | Supported. | According to the standard. |
| ipAddrTable | (20) NA<br>An address information table for IP addresses. | --- | --- |

| Subtree name | Standard [(*identifier*) *attribute*] | Support | Implementation specifications |
|---|---|---|---|
| ipAddrEntry | (1) NA<br>Row data of address information for IP addresses. The index is *ipAdEntAddr*. | --- | --- |
| ipAdEntAddr | (1) RO<br>The IP address. | Supported. | According to the standard. |
| ipAdEntIfIndex | (2) RO<br>The index value of the interface that this entry applies to.<br>This is the same value as *ifIndex*. | Supported. | According to the standard. |
| ipAdEntNet Mask | (3) RO<br>The subnet mask for the IP address of this entry. | Supported. | According to the standard. |
| ipAdEntBcastAddr | (4) RO<br>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<br>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<br>The IP routing table for this entity. | --- | --- |
| ipRouteEntry | (1) NA<br>Route information for a specific destination.<br>The index is *ipRouteDest*. | --- | --- |
| ipRouteDest | (1) RW<br>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<br>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<br>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<br>The alternative routing metric for this route. | Supported. | According to the standard. |
| | | ipRouteMetric3 | (5) RW<br>The alternative routing metric for this route. | Supported. | According to the standard. |
| | | ipRouteMetric4 | (6) RW<br>The alternative routing metric for this route. | Supported. | According to the standard. |
| | | ipRouteNext Hop | (7) RW<br>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<br>The type of route.<br>Other (1): Not any of the following types.<br>Invalid (2): An invalid route.<br>Direct (3): A direct connection.<br>Indirect (4): An indirect connection (not connected to LOCAL). | Supported. | According to the standard. |
| | | ipRouteProto | (9) RO<br>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.<br>Other (1): Other than the following items.<br>Local (2): A route set on the local machine.<br>Netmgmt (3): A route set by network management.<br>Icmp (4): A route set by an ICMP redirect or some other ICMP function.<br>Egp (5): EGP<br>The following are gateway protocols:<br>Ggp (6): GGP<br>Hello (7): HELLO<br>Rip (8): RIP<br>is-is (9)<br>es-is (10)<br>ciscolgrp (11)<br>bbnSpflgp (12)<br>ospf (13): OSPF<br>bgp (14) | Supported. | According to the standard. |
| | | ipRouteAge | (10) RW<br>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<br>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<br>The alternative routing metric. | Supported. | According to the standard. |
| ipRouteInfo | (13) RO<br>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<br>The IP address conversion table used to map IP addresses to physical addresses. | --- | --- |
| ipNetToMediaEntry | (1) NA<br>Row data for the conversion table. The indices are *ipNetToMediaIfIndex* and *ipNetToMediaNetAddress*. | --- | --- |
| ipNetToMediaIfIndex | (1) RW<br>The interface ID number for this entry. The value of *ifIndex* is used for this value. | Supported. | According to the standard. |
| ipNetToMediaPhysAddress | (2) RW<br>The media-dependent physical address. | Supported. | According to the standard. |
| ipNetToMediaNetAddress | (3) RW<br>The IP address that corresponds to the media-dependent physical address. | Supported. | According to the standard. |
| ipNetToMediaType | (4) RW<br>The address conversion method.<br>Other (1): A method other than the following items.<br>Invalid (2): An invalid value.<br>Dynamic (3): Dynamic conversion.<br>Static (4): Static conversion. | Supported. | According to the standard. |
| ipRoutingDiscards | (23) RO<br>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<br>The total number of received ICMP messages. This includes messages counted by *icmpInErrors*. | Supported. | According to the standard. |
| icmpInErrors | (2) RO<br>The number of received ICMP message errors. (Checksum errors, frame length errors, etc.) | Supported. | According to the standard. |
| icmpInDestUnreachs | (3) RO<br>The number of *Destination Unreachable* messages received. | Supported. | According to the standard. |
| icmpInTimeExcds | (4) RO<br>The number of *Time Exceed* messages received. | Supported. | According to the standard. |
| icmpInParmProbs | (5) RO<br>The number of *Parameter Problem* messages received. | Supported. | According to the standard. |
| icmpInSrcQuenchs | (6) RO<br>The number of *Source Quench* messages received. | Supported. | According to the standard. |
| icmpInRedirects | (7) RO<br>The number of *Redirect* messages received. | Supported. | According to the standard. |
| icmpInEchos | (8) RO<br>The number of *Echo (request)* messages received. | Supported. | According to the standard. |
| icmpInEchoReps | (9) RO<br>The number of *Echo Reply* messages received. | Supported. | According to the standard. |
| icmpInTimestamps | (10) RO<br>The number of *Timestamp* messages received. | Supported. | According to the standard. |
| icmpInTimestampReps | (11) RO<br>The number of *Timestamp Reply* messages received. | Supported. | According to the standard. |
| icmpInAddrMasks | (12) RO<br>The number of *Address Mask Request* messages received. | Supported. | According to the standard. |
| icmpInAddrMaskReps | (13) RO<br>The number of *Address Mask Reply* messages received. | Supported. | According to the standard. |
| icmpOutMsgs | (14) RO<br>The total number of ICMP messages sent. This includes messages counted by *icmpOutErrors*. | Supported. | According to the standard. |
| icmpOutErrors | (15) RO<br>The number of ICMP messages that could not be sent because of an error. | Supported. | According to the standard. |
| icmpOutDestUnreachs | (16) RO<br>The number of *Destination Unreachable* messages sent. | Supported. | According to the standard. |
| icmpOutTimeExcds | (17) RO<br>The number of *Time Exceed* messages sent. | Supported. | According to the standard. |
| icmpOutParmProbs | (18) RO<br>The number of *Parameter Problem* messages sent. | Supported. | According to the standard. |
| icmpOutSrcQuenchs | (19) RO<br>The number of *Source Quench* messages sent. | Supported. | According to the standard. |

| Name | Standard [(*identifier*) *attribute*] | Support | Implementation specifications |
|---|---|---|---|
| icmpOutRedirects | (20) RO<br>The number of *Redirect* messages sent. | Supported. | According to the standard. |
| icmpOutEchos | (21) RO<br>The number of *Echo (request)* messages sent. | Supported. | According to the standard. |
| icmpOutEchoReps | (22) RO<br>The number of *Echo Reply* messages sent. | Supported. | According to the standard. |
| icmpOutTimestamps | (23) RO<br>The number of *Timestamp* messages sent. | Supported. | According to the standard. |
| icmpOutTimestampReps | (24) RO<br>The number of *Timestamp Reply* messages sent. | Supported. | According to the standard. |
| icmpOutAddrMasks | (25) RO<br>The number of *Address Mask Request* messages sent. | Supported. | According to the standard. |
| icmpOutAddrMaskReps | (26) RO<br>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<br>The algorithm used to determine the timeout value for resending.<br>Other (1): Other than the following items.<br>Constant (2): A constant RTO value.<br>Rsre (3): The algorithm specified by the MIL-STD-1778 standard.<br>Vanj (4): The Van Jacobson algorithm. | Supported. | According to the standard. |
| tcpRtoMin | (2) RO<br>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<br>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<br>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<br>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<br>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<br>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<br>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<br>The total number of TCP connections currently in the ESTABLISHED or the CLOSE-WAIT state. | Supported. | According to the standard. |
| tcpInSegs | (10) RO<br>The total number of received segments. This includes the number of error segments. | Supported. | According to the standard. |
| tcpOutSegs | (11) RO<br>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<br>The total number of resent segments. | Supported. | According to the standard. |

| Name | Standard [(*identifier*) *attribute*] | Support | Implementation specifications |
|---|---|---|---|
| tcpConnTable | (13) NA<br>The information table specific to the TCP connection. | --- | According to the standard. |
| tcpConnEntry | (1) NA<br>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<br>The status of the TCP connection.<br>closed(1)<br>listen(2)<br>synSent(3)<br>synReceived(4)<br>established(5)<br>finWait1(6)<br>finWait2(7)<br>closeWait(8)<br>lastAck(9)<br>closing(10)<br>timeWait(11) | Supported. | According to the standard. |
| tcpConnLocalAddress | (2) RO<br>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<br>The local port number for this TCP connection. | Supported. | According to the standard. |
| tcpConnRemAddress | (4) RO<br>The remote IP address for this TCP connection. | Supported. | According to the standard. |
| tcpConnRemPort | (5) RO<br>The remote port number for this TCP connection. | Supported. | According to the standard. |
| tcpInErrs | (14) RO<br>The total number of error segments received (TCP checksum errors, etc.). | Supported. | According to the standard. |
| tcpOutRsts | (15) RO<br>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<br>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<br>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<br>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<br>The total number of sent UDP datagrams. | Supported. | According to the standard. |
| udpTable | (5) NA<br>The information table for the UDP listener. | --- | According to the standard. |
| udpEntry | (1) NA<br>An entry related to a specific UDP listener. The indices are *udpLocalAddress* and *udpLocalPort*. | --- | According to the standard. |
| udpLocal Address | (1) RO<br>The 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<br>The local port number for this UDP listener. | Supported. | According to the standard. |

## ● Ip Group: Snmp

| Name | | | Standard [(*identifier*) *attribute*] |
|---|---|---|---|
| snmpInPkts | (1) RO<br>The total number of SNMP messages received. | Supported. | According to the standard. |
| snmpOutPkts | (2) RO<br>The total number of SNMP messages sent. | Supported. | According to the standard. |
| snmpInBadVersions | (3) RO<br>The total number of messages received of an unsupported version. | Supported. | According to the standard. |
| snmpInBadCommunityNames | (4) RO<br>The total number of messages received from an unregistered community. | Supported. | According to the standard. |
| snmpInBadCommunityUses | (5) RO<br>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<br>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<br>The total number of PDUs received with an error status of *tooBig*. | Supported. | According to the standard. |
| snmpInNoSuchNames | (9) RO<br>The total number of PDUs received with an error status of *noSuchName*. | Supported. | According to the standard. |

| **Name** | | | **Standard [(*identifier*) *attribute*]** |
|---|---|---|---|
| snmpInBadValues | (10) RO<br>The total number of PDUs received with an error status of *badValue*. | Supported. | According to the standard. |
| snmpInReadOnlys | (11) RO<br>The total number of PDUs received with an error status of *readOnly*. | Supported. | According to the standard. |
| snmpInGenErrs | (12) RO<br>The total number of PDUs received with an error status of *genErr*. | Supported. | According to the standard. |
| snmpInTotalReqVars | (13) RO<br>The total number of MIB objects read normally after receiving *GetRequest* or *GetNextRequest*. | Supported. | According to the standard. |
| snmpInTotalSetVars | (14) RO<br>The total number of MIB objects updated normally after receiving *SetRequest*. | Supported. | According to the standard. |
| snmpInGetRequests | (15) RO<br>The total number of *GetRequest* PDUs received. | Supported. | According to the standard. |
| snmpInGetNexts | (16) RO<br>The total number of *GetNextRequest* PDUs received. | Supported. | According to the standard. |
| snmpInSetRequests | (17) RO<br>The total number of *SetRequest* PDUs received. | Supported. | According to the standard. |
| snmpInGetResponses | (18) RO<br>The total number of *GetResponse* PDUs received. | Supported. | According to the standard. |
| snmpInTraps | (19) RO<br>The total number of trap PDUs received. | Supported. | According to the standard. |
| snmpOutTooBigs | (20) RO<br>The total number of PDUs sent with an error status of *tooBig*. | Supported. | According to the standard. |
| snmpOutNoSuchNames | (21) RO<br>The total number of PDUs sent with an error status of *noSuchName*. | Supported. | According to the standard. |
| snmpOutBadValues | (22) RO<br>The total number of PDUs sent with an error status of *badValue*. | Supported. | According to the standard. |
| snmpOutGenErrs | (24) RO<br>The total number of PDUs sent with an error status of *genErr*. | Supported. | According to the standard. |
| snmpOutGetRequests | (25) RO<br>The total number of *GetRequest* PDUs sent. | Supported. | According to the standard. |
| snmpOutGetNexts | (26) RO<br>The total number of *GetNextRequest* PDUs sent. | Supported. | According to the standard. |
| snmpOutSetRequests | (27) RO<br>The total number of *SetRequest* PDUs sent. | Supported. | According to the standard. |
| snmpOutGetResponses | (28) RO<br>The total number of *GetResponse* PDUs sent. | Supported. | According to the standard. |
| snmpOutTraps | (29) RO<br>The total number of trap PDUs sent. | Supported. | According to the standard. |
| snmpEnableAuthenTraps | (30) RW<br>Determines if the agent generates verification failed traps.<br>Enabled (1)<br>Disabled (2) | Supported. | According to the standard. |

# 11-2 Procedure to Use the SNMP Agent

## 11-2-1 Procedures

**1** Make the basic settings.

Refer to *1-5 EtherNet/IP Communications Procedures* for the flow of basic operations.

**2** Make the following settings in the Special Unit Setup in the Controller Configurations and Setup of 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 EtherNet/IP Unit settings are transferred to the CPU Unit.

## 11-2-2 Settings Required for the SNMP Agent

The following EtherNet/IP Unit settings are made from the Sysmac Studio to use the SNMP agent.

| Tab page | Setting | | Setting conditions | Reference |
|---|---|---|---|---|
| SNMP Settings | SNMP service | | Required. | page 5-12 |
| | Port No. | | Specified by user.<br>**Note** Required to change from the default value of 161. | |
| | Address, location | | Specified by user. | |
| | Send a recognition trap | | Specified by user.<br>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.<br>Make these settings to permit access by only certain SNMP managers. | page 5-13 |
| | | IP address | | |
| | | Host name | | |
| | | Community name | | |
| SNMP Trap Settings | SNMP trap | | Required. | page 5-14 |
| | Port No. | | Specified by user.<br>**Note** Required to change from the default value of 162. | |
| | Trap 1 and trap 2 | | | page 5-15 |
| | | IP address | Required.<br>Set an IP address or a host name as the SNMP trap destination. | |
| | | Host name | | |
| | | Community name | Specified by user. | |
| | | Version | Required.<br>Set the version of the SNMP manager. | |

**Additional Information**

Make the settings in the SNMP Settings Dialog Box and SNMP Trap Dialog Box if the SNMP agent is used.
Refer to *5-6 SNMP Settings Display* for information on the SNMP Settings Dialog Box. Refer to *5-7 SNMP Trap Settings Display* for information on the SNMP Trap Dialog Box.

# 12

# Security Functions

This section provides information on security functions provided by CJ1W-EIP21S EtherNet/IP Units. The functions described in this section are not provided for the CJ1W-EIP21.

# 12-1 Overview of Security Functions

This section provides an overview of security functions provided by CJ1W-EIP21S EtherNet/IP Units.

## 12-1-1 List of Security Functions

CJ1W-EIP21S EtherNet/IP Units provide a set of security functions that are intended to protect user assets from network access.

These functions protect user programs and various data stored in the CPU Unit.

| Function name | Function overview | Reference |
|---|---|---|
| Opening and closing the port | Blocks and allows packets to pass through the TCP/UDP ports assigned to individual communications functions according to the settings. | *12-2 Opening and Closing the Port* |
| Packet filter | Selectively permits packets that EtherNet/IP ports to receive to pass through the filter according to the preset conditions. | *12-3 Packet Filter* |

Refer to *12-4 General Security Use Cases* for the configuration example and settings for each use case.

### Supported Support Software

The following Support Software is supported.

| Support Software | Version |
|---|---|
| Sysmac Studio | Ver. 1.60 or higher |

### Supported CPU Units

The following CPU Units are supported.

| CPU Units | Version |
|---|---|
| NJ-series CPU Units | Ver. 1.67 or later |

# 12-2 Opening and Closing the Port

This section describes the function of opening and closing the port.

## 12-2-1 Function Overview

Opening and closing the port is a function that blocks and allows packets to pass through the TCP/UDP ports assigned to individual communications functions according to the settings.

By setting any communications functions that you will not use to *Do not use*, you can reduce the number of entry points for external attacks to improve the security performance of the system.

**Precautions for Correct Use**

Set all communications functions that you will or may use to *Use*.

This function enables or disables the communications functions for a node based on the settings. For example, when CIP Message Server is set to *Do not use*, the tag data link for that node will not work.

## 12-2-2 Function Details

The table below shows whether or not it is possible to use TCP/UDP and other communications functions.

Here, the functions supported only by the CJ1W-EIP21S EtherNet/IP Units are described.

### Settings

You can make the settings for the following communications functions.

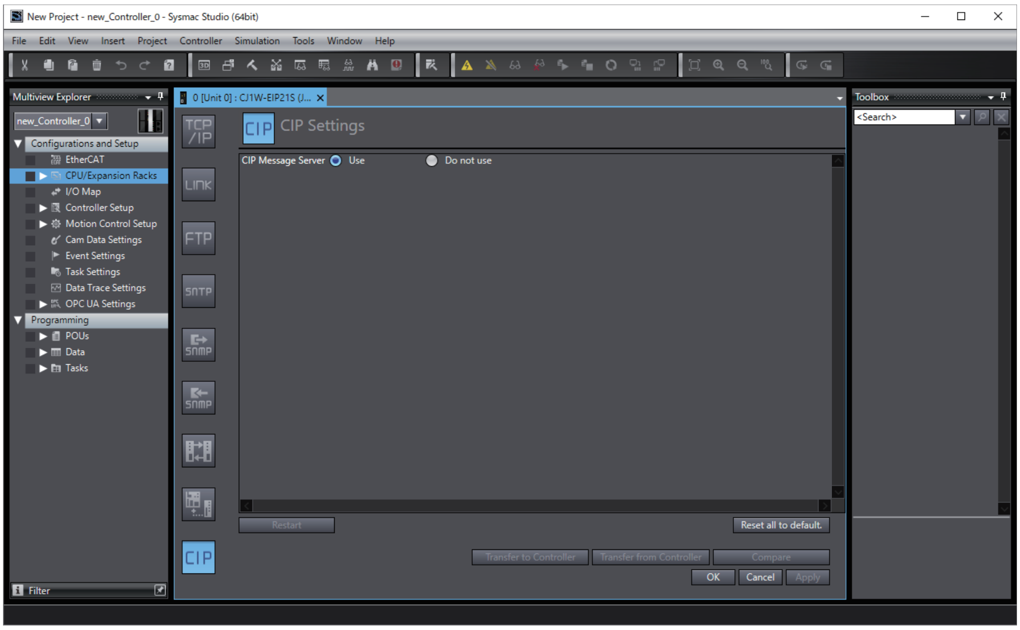
- CIP message server
- FINS/UDP
- FINS/TCP

The settings will be reflected and enabled when the Unit is restarted after completion of the transfer.

### ● CIP Message Server

Set this in **CIP** – **CIP Message Server** in the Special Unit Settings Tab Page for the CJ1W-EIP21S EtherNet/IP Unit in the Sysmac Studio.

| Setting | Description | Default |
|---|---|---|
| CIP Message Server | Select whether or not to use the CIP message server.<br>• Do not use<br>• Use | Use |



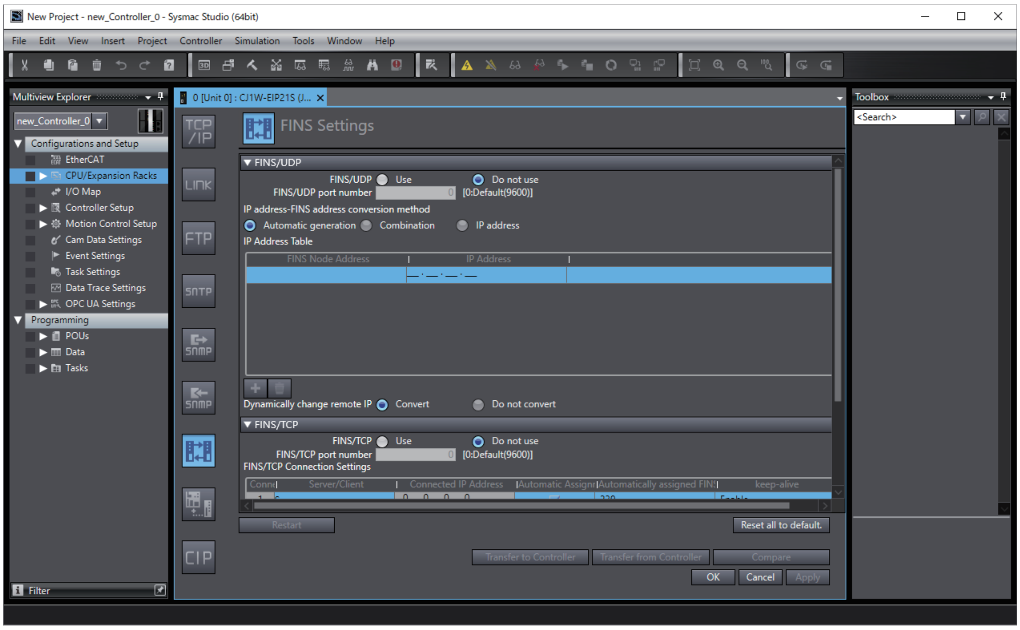
CIP Settings

CIP Message Server ● Use ○ Do not use

## ● FINS/UDP

Set this in **FINS** – **FINS/UDP** in the Special Unit Settings Tab Page for the CJ1W-EIP21S EtherNet/IP Unit in the Sysmac Studio.

| Setting | Description | Default |
|---|---|---|
| FINS/UDP | Select whether or not to use FINS/UDP.<br>• Do not use<br>• Use | Do not use |



FINS Settings

▼ FINS/UDP

FINS/UDP ○ Use ● Do not use

FINS/UDP port number 0 [0:Default(9600)]

IP address-FINS address conversion method

● Automatic generation ○ Combination ○ IP address

IP Address Table

| FINS Node Address | IP Address |
|---|---|
| | —.—.—.— |

Dynamically change remote IP ● Convert ○ Do not convert

▼ FINS/TCP

FINS/TCP ○ Use ● Do not use

FINS/TCP port number 0 [0:Default(9600)]

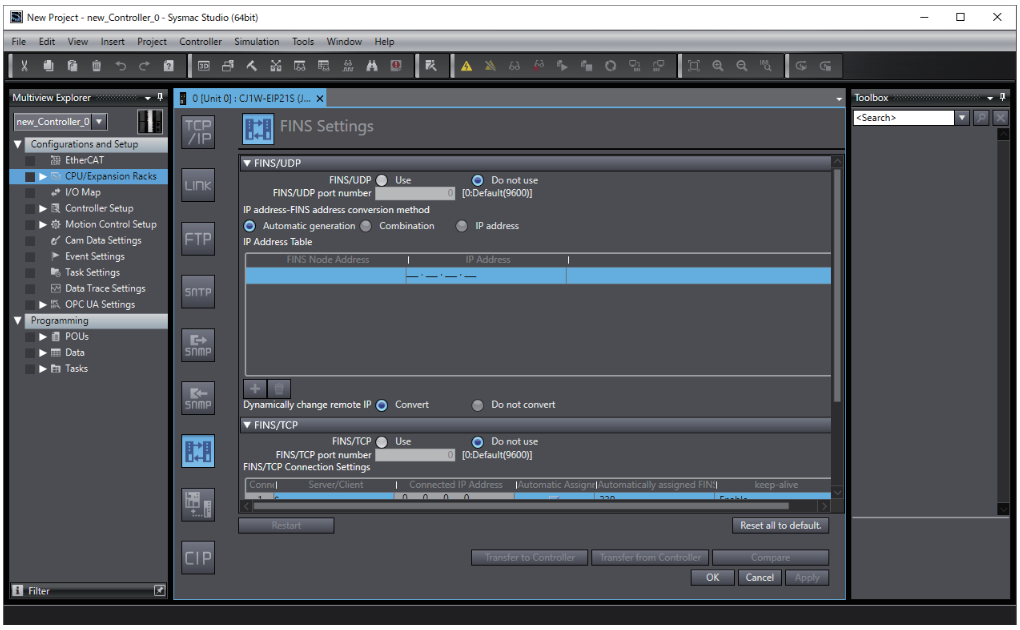
FINS/TCP Connection Settings

| Conne | Server/Client | Connected IP Address | Automatic Assignr | Automatically assigned FIN | keep-alive |
|---|---|---|---|---|---|

- **FINS/TCP**

Set this in **FINS** – **FINS/TCP** in the Special Unit Settings Tab Page for the CJ1W-EIP21S EtherNet/IP Unit in the Sysmac Studio.

| Setting | Description | Default |
|---|---|---|
| FINS/TCP | Select whether or not to use FINS/TCP.<br>• Do not use<br>• Use | Do not use |



New Project - new_Controller_0 - Sysmac Studio (64bit)

0 [Unit 0] : CJ1W-EIP21S (J...

FINS Settings

FINS/UDP

FINS/UDP ○ Use ● Do not use

FINS/UDP port number 0 [0:Default(9600)]

IP address-FINS address conversion method

● Automatic generation ○ Combination ○ IP address

IP Address Table

| FINS Node Address | IP Address |
|---|---|
| | __.__.__.__ |

Dynamically change remote IP ● Convert ○ Do not convert

FINS/TCP

FINS/TCP ○ Use ● Do not use

FINS/TCP port number 0 [0:Default(9600)]

FINS/TCP Connection Settings

| Conne | Server/Client | Connected IP Address | Automatic Assignr | Automatically assigned FINS | keep-alive |
|---|---|---|---|---|---|

## Specifications

- **CIP Message Server**

Set **CIP Message Server** to *Use* to use tag data links and Explicit messages.

| Setting | Operation |
|---|---|
| Do not use | • Tag data links and Explicit messages cannot be used.<br>• Support Software or NS/NA series programmable terminals cannot connect to the CPU when the Network Type is *EtherNet/IP*. |
| Use [*1] | • Tag data links and Explicit messages can be used.<br>• Support Software or NS/NA series programmable terminals can connect to the CPU when the Network Type is *EtherNet/IP*. (This is the same as the operation of CJ1W-EIP21 EtherNet/IP Units.) |

*1 Default

When tag data link settings exist and **CIP Message Server** is set to *Do not use* for a node, the behavior of the function will be as follows.

| When set for local node | | When set for remote node | |
|---|---|---|---|
| Originator | Target | Originator | Target |
| A verification error (target nonexistent) will occur.<br>• 7-segment display: d5<br>• Error code: Not recorded. | No error will be detected. | A verification error (target nonexistent) will occur.<br>• 7-segment display: d5<br>• Error code: Not recorded. | No error will be detected. |

## ● FINS/UDP and FINS/TCP

| Service | Setting | Operation |
|---|---|---|
| FINS/UDP | Do not use [*1] | The Network Configurator cannot connect via Ethernet.<br>NS/NA/NB-series programmable terminals cannot connect to the CPU via Ethernet.<br>When both **FINS/UDP** and **FINS/TCP** are set to *Do not use*, executing the SEND, RECV, CMND, SEND2, RECV2, or CMND2 instruction results in an abnormal completion. [*2] |
| | Use | FINS/UDP services can be used. |
| FINS/TCP | Do not use [*1] | The Network Configurator cannot connect via Ethernet.<br>When both FINS/UDP and **FINS/TCP** are set to *Do not use*, executing the SEND, RECV, CMND, SEND2, RECV2, or CMND2 instruction results in an abnormal completion. [*2] |
| | Use | FINS/TCP services can be used. |

[*1] Default

[*2] The behavior of the instruction differs depending on the response required/not required setting in its control data.
If response not required, the instruction will be normally completed.
If response required, the instruction will be completed abnormally. The response code is 0205 hex (Timeout error: No response returned from remote node. Monitoring timer timed out.).

# 12-3 Packet Filter

This section describes the packet filter function.

## 12-3-1 Function Overview

The packet filter function selectively permits packets that EtherNet/IP ports receive to pass through the filter according to the preset conditions. This improves the security performance of the system.

### Precautions for Correct Use

Register all devices and conditions for which to permit packet reception in **Packet Filter**.

If the Packet Filter settings do not match the specifications of the system, or if the settings are incorrect, the devices may not either communicate or connect to Support Software. Therefore, we recommend using only the use cases described in *12-4 General Security Use Cases*.

Alternatively, ask someone who fully understands the network specifications of the system and has knowledge of Ethernet to perform setup and management.

The network specifications here refer to the IP addresses of the devices and the TCP/UDP port numbers used for communications.

## 12-3-2 Function Details

In **TCP/IP** – **Packet Filter** in the Special Unit Settings Tab Page for the CJ1W-EIP21S EtherNet/IP Unit in the Sysmac Studio, set whether or not to use the packet filter function and the filter conditions for packets permitted to pass through the filter.

### Precautions for Correct Use

- Enabling the Packet Filter settings prevents the Unit from communicating with devices via the target EtherNet/IP port. Be careful of the settings.
- The settings in **Allowed Packets** are retained when you change the setting of **IP Packet Filter** from *Use* to *Do not use* in the Sysmac Studio. The settings in **Allowed Packets** are transferred to the CJ1W-EIP21S EtherNet/IP Unit regardless of the setting of whether or not to use the function.
- The packet filter function supports the stateful inspection. Therefore, the CJ1W-EIP21S EtherNet/IP Unit can receive responses from the remote node without filter conditions added to the allowed packet when it makes a request to a remote node as a client.
  The protocols supported by stateful inspection are TCP, UDP, and ICMP.

### Additional Information

The following function is not supported by the packet filter function. This is because it is a client function that opens ports during use and closes them at completion.
DNS/SNTP

## Settings

Refer to *Packet Filter (CJ1W-EIP21S Only)* on page 5-5 for the Packet Filter setting category and settings.

The settings will be reflected and enabled when the Unit is restarted after completion of the transfer.

## Settings for Various Use Scenarios

The allowed packet settings to use the packet filter function are shown for each device or function that you will use.

### ● Settings for Connecting Support Software

The settings for using Support Software are as shown in the table below.

| Support Software | Connection method | Applicable protocol | *Protocol Filter* settings | | | |
|---|---|---|---|---|---|---|
| | | | Protocol | *Destination Port* settings | | |
| | | | | Range specification | Start port No. | End port No. |
| Network Configurator | *Ethernet I/F* is selected as the interface. | CIP | TCP | Use | 44818 | 44818 |
| | | CIP | UDP | | 44818 | 44818 |
| | | ICMP | ICMP | Do not use | --- | --- |

### ● Settings for Using EtherNet/IP Communications

The settings for using EtherNet/IP communications are as follows.

| Communications name | Applicable protocol | Condition | *Protocol Filter* settings | | | |
|---|---|---|---|---|---|---|
| | | | Protocol | *Destination Port* settings | | |
| | | | | Range specification | Start port No. | End port No. |
| EtherNet/IP messages | UCMM | Server | TCP | Use | 44818 | 44818 |
| | Class 3 | Server | TCP | | 44818 | 44818 |
| EtherNet/IP tag data links | Class 1 | Originator | IGMP *1 | Do not use | --- | --- |
| | | Target | TCP | Use | 44818 | 44818 |

*1 Set this to use Multicast.

**Additional Information**

If you use SYSMAC Gateway or CX-Compolet, in addition to the above settings, make settings to also permit ICMP.

The settings are as follows:

| *Protocol Filter* settings | | | |
|---|---|---|---|
| Protocol | *Destination Port* settings | | |
| | Range specification | Start port No. | End port No. |
| ICMP | Do not use | --- | --- |

### ● Settings for Connecting OMRON Programmable Terminals

The settings for connecting OMRON Programmable Terminals are as shown in the table below.

- NA-series Programmable Terminals

| Connection method | Applicable protocol | ***Protocol Filter* settings** | | | |
|---|---|---|---|---|---|
| | | **Protocol** | ***Destination Port* settings** | | |
| | | | **Range specification** | **Start port No.** | **End port No.** |
| Connection via EtherNet/IP | CIP | TCP | Use | 44818 | 44818 |
| Connection via Ethernet | FINS | UDP | | 9600 [*1] | 9600 [*1] |

[*1] If you enter a port number value that is not the default in the FINS/UDP settings, the value will be set.

- NB-series Programmable Terminals

| Connection method | Applicable protocol | ***Protocol Filter* settings** | | | |
|---|---|---|---|---|---|
| | | **Protocol** | ***Destination Port* settings** | | |
| | | | **Range specification** | **Start port No.** | **End port No.** |
| Connection via Ethernet | FINS | UDP | Use | 9600 [*1] | 9600 [*1] |

[*1] If you enter a port number value that is not the default in the FINS/UDP settings, the value will be set.

- NS-series Programmable Terminals

| Connection method | Applicable protocol | ***Protocol Filter* settings** | | | |
|---|---|---|---|---|---|
| | | **Protocol** | ***Destination Port* settings** | | |
| | | | **Range specification** | **Start port No.** | **End port No.** |
| Connection via EtherNet/IP | CIP | TCP | Use | 44818 | 44818 |
| Connection via Ethernet | FINS | UDP | | 9600 [*1] | 9600 [*1] |

[*1] If you enter a port number value that is not the default in the FINS/UDP settings, the value will be set.

### ● Settings for Using FINS Message Communications

The settings for using FINS message communications are as shown in the table below.

| Connection method | Applicable protocol | ***Protocol Filter* settings** | | | |
|---|---|---|---|---|---|
| | | **Protocol** | ***Destination Port* settings** | | |
| | | | **Range specification** | **Start port No.** | **End port No.** |
| FINS/UDP | FINS | UDP | Use | 9600 [*1] | 9600 [*1] |
| FINS/TCP | FINS | TCP | | 9600 [*1] | 9600 [*1] |

[*1] If you enter a port number value that is not the default in the FINS/UDP or FINS/TCP settings, the value will be set.

### Settings for Using SNMP

The *Protocol Filter* and *Destination Port* settings for using SNMP are as follows.

**Protocol Filter** – **Protocol**: UDP

**Destination Port**

Range specification: Use

Start port No.: 161 [*1]

End port No.: 162 [*1]

[*1] If you enter an SNMP/SNMP trap port number, the value will be set.

### Settings for Using FTP Server

The *Protocol Filter* and *Destination Port* settings for using the FTP server function are as follows.

**Protocol Filter** – **Protocol**: TCP

**Destination Port**

Range specification: Use

Start port No.: 20 [*1]

End port No.: 21 [*1]

[*1] If you enter an FTP port number, the value will be set.

## Troubleshooting

### If You Forget the Packet Filter Settings for a CJ1W-EIP21S EtherNet/IP Unit

Place the Sysmac Studio online with the CPU Unit on which the EtherNet/IP Unit is mounted and check or change the Packet Filter settings.

### If Access from a Remote Node Fails

Follow the steps below to narrow down the cause and take corrective action.

| Step | Item | Corrective action |
|---|---|---|
| 1 | Determine whether the cause is the Packet Filter settings. | • Set **Packet Filter** to *Do not use* and check if it is possible to access the remote node.<br>• If possible, the cause is the Packet Filter settings. Go to step 2.<br>• If not possible, the cause may be other than the Packet Filter settings. Review the communications path, connection destination settings, etc. |
| 2 | Review the Packet Filter settings. | • Reset the IP address and port number settings for the devices and Support Software connected to the CJ1W-EIP21S.<br>• Monitor the communications line data and check if the source IP address, source port, and destination port are registered in the Packet Filter settings (only if possible). |

# 12-4 General Security Use Cases

This section describes general security use cases and shows a configuration example and settings for each of them.

## 12-4-1 Use Cases

The following use cases are described.

| Case | Use case | Function to use | Reference |
|---|---|---|---|
| 1 | Permitting packet reception for specific protocols | Opening and closing the port | *12-2 Opening and Closing the Port* |
| 2 | Permitting packet reception from specific source IP addresses | Packet Filter | *12-3 Packet Filter* |

## 12-4-2 Case 1: Permitting Packet Reception for Specific Protocols

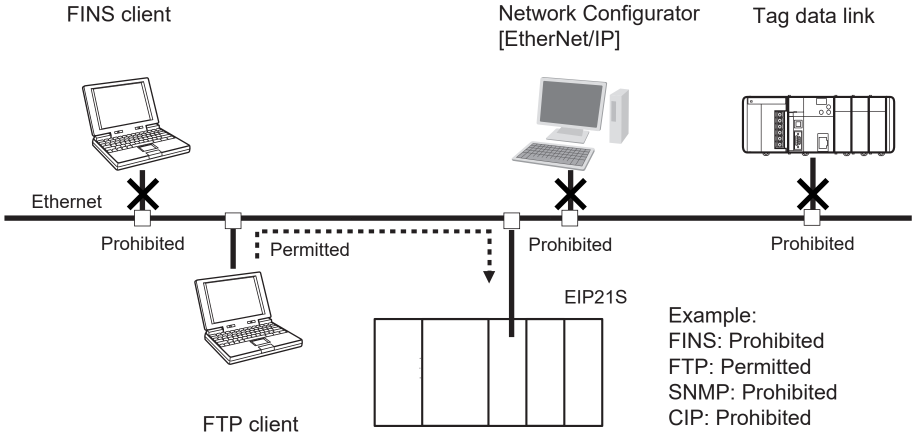
This use case is for permitting access from a specific protocol.

Use it to prohibit communications for protocols that are not used.

In this use case, the function of opening and closing the port is used.

### Configuration Example

This configuration example prohibits communications for protocols other than FTP.



Example:
FINS: Prohibited
FTP: Permitted
SNMP: Prohibited
CIP: Prohibited

### Settings for This Configuration Example

Make the settings as shown in the following table.

| TCP/IP communications functions | Setting | Reference |
|---|---|---|
| CIP Message Server | Do not use | *12-2 Opening and Closing the Port* |
| FINS/UDP | Do not use | |
| FINS/TCP | Do not use | |
| FTP server | Use | *9-3 Using the FTP Server Function* |
| SNMP | Do not use | *11-2 Procedure to Use the SNMP Agent* |

## 12-4-3 Case 2: Permitting Packet Reception from Specific Source IP Addresses

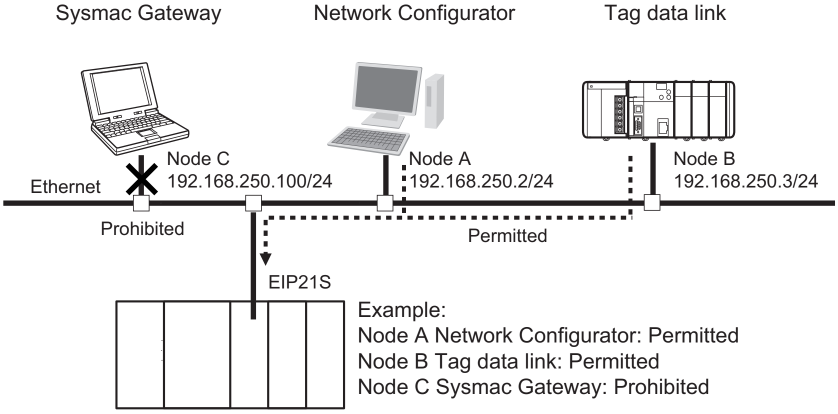
This use case is for permitting access from specific nodes.

Use it to prohibit connections from unauthorized nodes, such as computers brought into the site without permission.

In this use case, the packet filter function is used.

### Configuration Example

This configuration example permits only communications from specific computers and external devices, and prohibits communications from other client devices.



Example:
Node A Network Configurator: Permitted
Node B Tag data link: Permitted
Node C Sysmac Gateway: Prohibited

### Settings for This Configuration Example

Make the settings in the **Packet Filter** category as shown in the following table.

Set the IP addresses of nodes A and B individually. Therefore, set *255.255.255.255* as the mask.

| No. | IP Filter | | | Protocol Filter | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Source Settings | | | Protocol Filter | Source Port | | | Destination Port | | |
| | Setting method | IP Address | Mask | | Range specification | Start port No. | End port No. | Range specification | Start port No. | End port No. |
| 1 | IP address specification | 192.168.250.2 | 255.255.255.255 | Any | --- | --- | --- | --- | --- | --- |
| 2 | IP address specification | 192.168.250.3 | 255.255.255.255 | Any | --- | --- | --- | --- | --- | --- |

#### Precautions for Correct Use

Register the IP addresses of all external devices to use because communications from IP addresses that are not registered in the Packet Filter settings are blocked.

**Additional Information**

If you set the mask to other than 255.255.255.255, communications from multiple IP addresses will be permitted.

For example, the settings below permit communications from devices with IP addresses between 192.168.250.0 and 192.168.250.255.

| No. | IP Filter | | | Protocol Filter | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Source Settings | | | Protocol Filter | Source Port | | | Destination Port | | |
| | Setting method | IP Address | Mask | | Range specification | Start port No. | End port No. | Range specification | Start port No. | End port No. |
| 1 | IP address specification | 192.168.250.0 | 255.255.255.0 | Any | --- | --- | --- | --- | --- | --- |

# 12-5 Protective Measures to Prevent Security Threats

Using the security function of CJ1W-EIP21S EtherNet/IP Units is useful for preventing a network from security threats. To do so, you need to use the function of the Units properly.

This section describes the operational measures for using the security function of the Units properly.

### ● Arranging Communications Devices to a Reliable Network

Arrange communications devices, including the EtherNet/IP Units, in a reliable network.

If they are connected through an unreliable network, take appropriate measures such as using VPN.

Refer to *1-3-1 Specifications* for the services supported by CJ1W-EIP21S EtherNet/IP Units.

### ● Deleting Data before Discarding the Units

When you discard CJ1W-EIP21S EtherNet/IP Units and CPU Units connecting them, delete the information inside the Units by the following methods to prevent information leakage.

| Target | Description | Procedure |
|---|---|---|
| CPU Units | Clear all memory areas. | Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504). |
| CJ1W-EIP21S | Initialize them to the defaults. | In the Sysmac Studio, click the **Restart** Button at the lower left of the Special Unit Settings Tab Page for the CJ1W-EIP21S. Then, in the Restart Unit Dialog Box displayed, select **Restart after returning to factory settings** and click the **Restart** Button to restart the Unit. |

### ● Putting a Physical Access Restriction to the Units

Implement measures that restrict anyone other than authorized personnel not to enter the place where the system including the CPU Units and the CJ1W-EIP21S is installed.

The measures include restricting such people not to enter that place and locking the entrance.

### ● Putting an Access Restriction to the Network Configuration File and Managing the Configuration

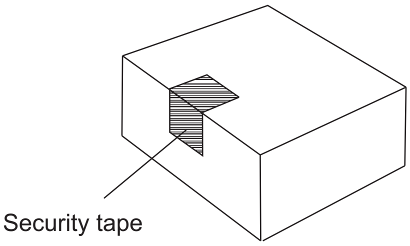
Implement measures that prevent anyone other than authorized personnel from obtaining or tampering the network configuration file that the Network Configurator creates.

The measures include keeping the network information file in an access-restricted place and managing the configuration of that.

Refer to the *CS/CJ-series EtherNet/IP Units Operation Manual* (Cat. No. W465) for the network information file.

### Using Unpacked Items

For CJ1W-EIP21S EtherNet/IP Units and CPU Units connecting them, use those with security tape unpeeled from the purchased item packaging boxes.



**Precautions for Correct Use**

To prevent tampering, the product’s item packaging boxes are sealed with tape that indicates that they have not been opened. Confirm that they have not been opened before use.

# 13

# Socket Services

This section provides information on socket services provided by CJ1W-EIP21S EtherNet/IP Units. The functions described in this section are not provided for the CJ1W-EIP21.

# 13-1 Overview of Socket Communications from EtherNet/IP Units

## 13-1-1 What are Sockets?

Sockets are interfaces that allow TCP and UDP protocols to be used directly from the user program.

For host computers such as personal computers, socket are provided as C language interface libraries, which allow TCP or UDP protocols to be programmed using library functions.

For UNIX computers, socket interfaces are supported in the form of system calls.

NJ-series Controllers allow the user program to use the socket services by manipulating device variables of the EtherNet/IP Unit.

Socket communications services can be used to transfer arbitrary data with remote nodes between a Controller and a host computer or between two Controllers.

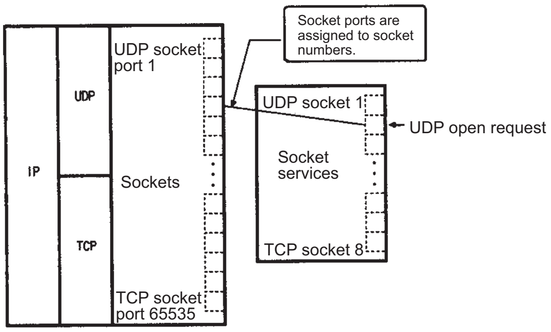
The EtherNet/IP Unit supports two socket services: a UDP socket service and a TCP socket service.

### Using Sockets with the EtherNet/IP Unit

The EtherNet/IP Unit supports up to 16 simultaneous socket connections for the socket services, 8 each for UDP and TCP sockets.
Socket numbers 1 to 8 are assigned to sockets for both UDP and TCP sockets.

Sockets are managed from the ladder-diagram program by assigning a socket port for each socket number.

The socket port number is assigned when the socket is opened.



## 13-1-2 Socket Service Port Numbers

Port numbers up to 1023 on a UNIX workstation can be used by the superuser only. Port numbers 0 to 255 are reserved for well-known ports.

Consequently, port numbers 1024 and above should be used for socket services.

The EtherNet/IP Unit does not support port #0.

Some port numbers over 1024 may be reserved on some workstations (for example, the X-window server is port #6000). Do not use port numbers that are already reserved for other processing.

The setting status of the UNIX workstation port numbers can be checked in /etc/services.

# 13-2 Protocol Overview

## 13-2-1 Differences between TCP and UDP

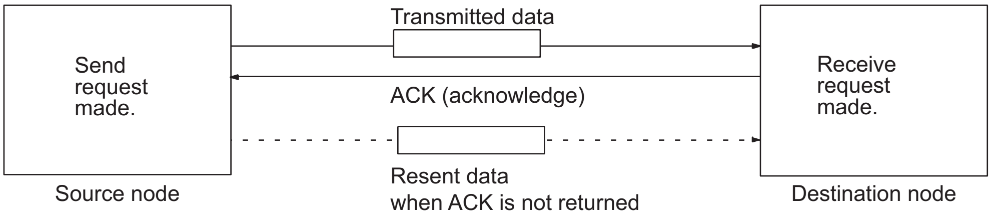
There are differences in the socket services between TCP and UDP.

### TCP Communications

The following procedure is followed each time data is transmitted to ensure that the data arrives normally at the destination node:

The destination node returns ACK when data is received normally.

The source node sends the next data after it receives ACK, or it resends the same data if ACK is not returned within the specified time.



With the TCP protocol, the remote IP address and remote TCP port number are specified when an open request is made for a socket. When a send request is made, the number of bytes to send and the send data are specified.

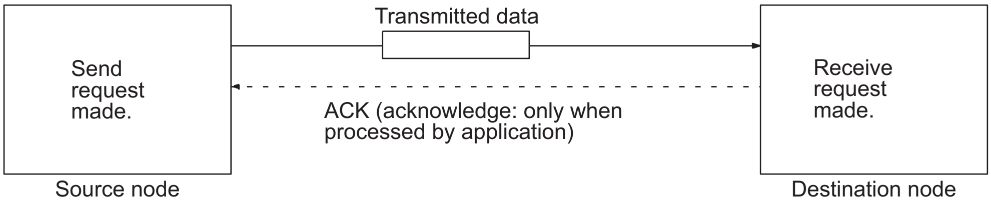
When a receive request is made, the number of bytes to receive is specified.

With the TCP protocol, communications with another remote device are not possible until the socket that was opened has been closed.

### UDP Communications

Data is simply sent to the destination node. Unlike TCP, the reception of data is not checked and data is not resent.

To increase communications reliability, data resends must be programmed by the user in user application.



With the UDP protocol, the remote IP address and remote UDP port number are not specified when an open request is made for a socket. When a send request is made, the remote IP address, the remote UDP port number, the number of bytes to send, and the send data are specified.

When a receive request is made, the number of bytes to receive is specified. (The response data shows from which IP address and UDP port number the received data was sent.)

With the UDP protocol, communications with another remote device are possible even if the socket that was opened is not closed.

## 13-2-2 Opening TCP Sockets

To achieve highly reliable data communications, TCP establishes a virtual communications circuit between the two nodes before starting data transmissions.

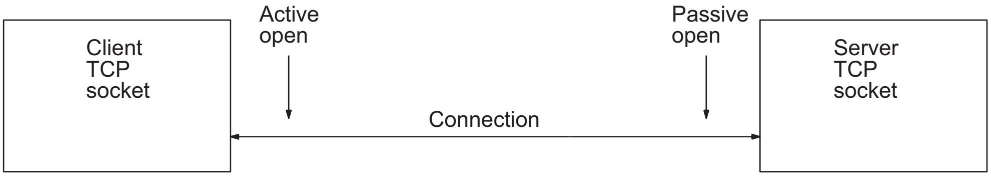
The virtual communications circuit is known as a “connection.”

### Passive Open and Active Open

Open processing is executed at each node to establish a connection.

The open method differs depending on whether the node is a client or server.

The passive open method is used to open the node as a server and the active open method is used to open the node as a client.



**Additional Information**

- TCP sockets must be closed once a connection has been made before communications are possible with other TCP sockets. This is true for other server and client sockets. Up to eight TCP sockets can be open simultaneously.
- With UDP sockets, communications are possible with more than one other UDP socket.

**Additional Information**

**Client and Server**

When a connection is made between two nodes, the processing at the node providing a service is called the server, and the processing at the node requesting the service is called the client.

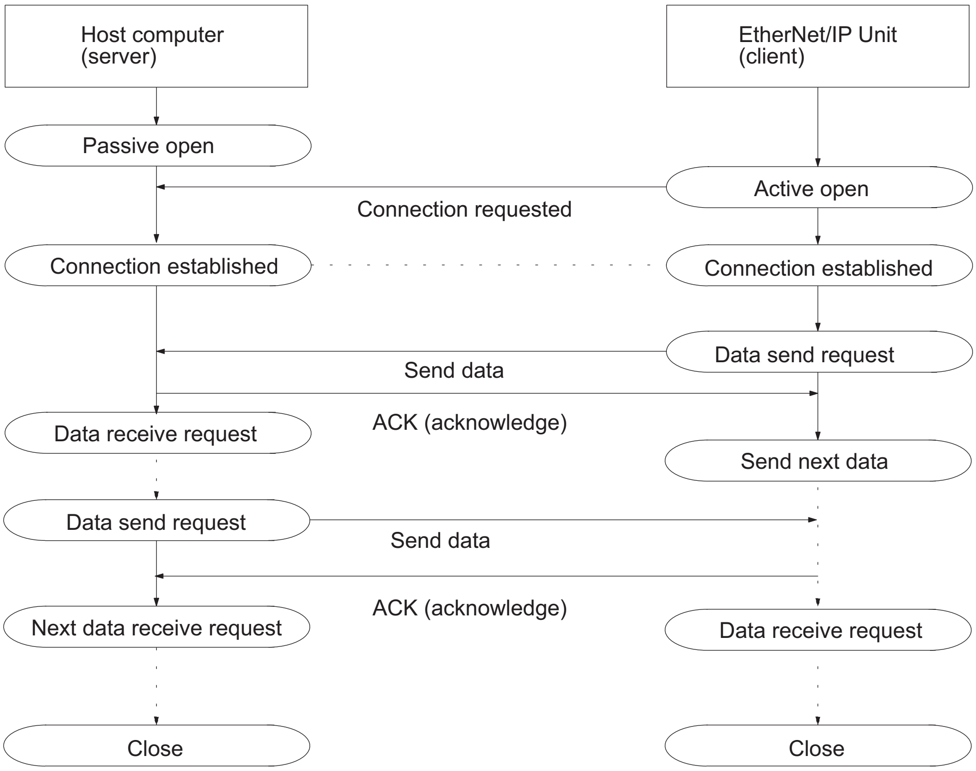
The server is started first and waits for a service request from a client.

The client requests to the server that a connection be opened and then transfers data.

When the TCP protocol is used, however, the client–server relationship does not need to be programmed in the application because it is automatically handled by the protocol.

## TCP Communications Procedure

The communications procedure is shown below for communications between a host computer and EtherNet/IP Unit using a TCP socket. In this example, the host computer is the server and the EtherNet/IP Unit is the client.



## 13-2-3 Fragmentation of Send Data

The EtherNet/IP Unit fragments data for TCP transmission into units of 1,460 bytes and data for UDP transmission into units of 1,472 bytes.

Therefore, the following points must be considered to receive data at the receiving node.

### Cautions when Using TCP

An example of the fragmentation and transmission of data using the TCP is shown in the following illustration.

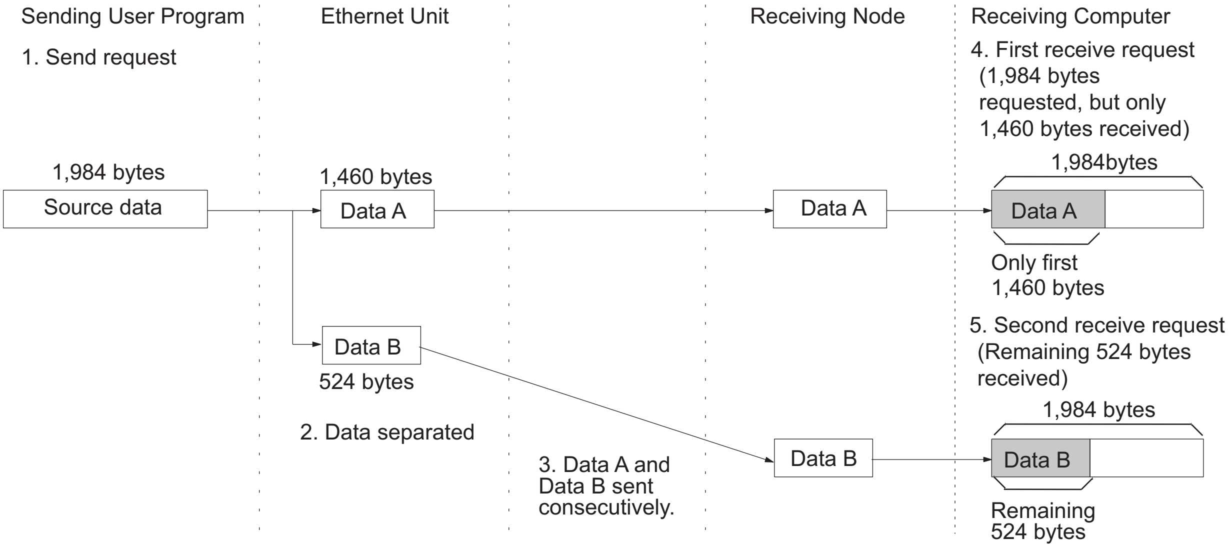
**(1) The sending user program sends a request to send 1,984 bytes of data.**

**(2) The EtherNet/IP Unit fragments the send data into Data A with 1,460 bytes and Data B with 524 bytes.**

**(3) Data A and Data B are sent consecutively.**

**(4) The receiving user program sends a request to receive 1,984 bytes of data. However, only data A is sent in the first packet, and data B is not received.**

**(5) Another receive request to receive data must be made before the remaining data, Data B, is received.**



When using TCP protocol, the fragmented data is passed to the user program.

Therefore, the receiving user program must be able to evaluate the end of the data transmission, and repeatedly send receive requests until all data has been received.

The receive request is sent twice in the example shown above, but the data would be even more fragmented if a router was included in the communications path, and the number of receive requests would need to be increased accordingly.

When making the receive request, it is not necessary to specify the same data length as the sent data length.

For example, if the length setting is shorter than the actual length of the data, all the data can be received by repeating the receive requests.

**Additional Information**

If communications are with a different segment and data is sent via the TCP protocol, data will be fragmented into units of 536 bytes.

## Cautions when Using UDP

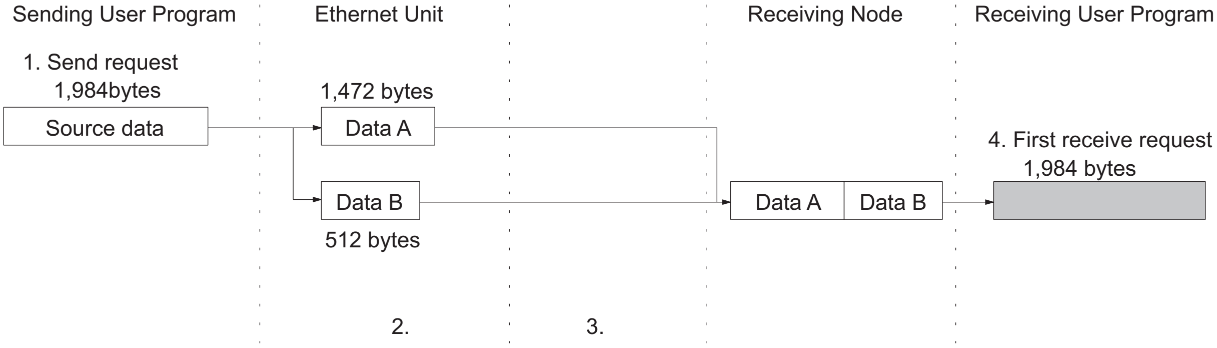
An example of fragmentation and transmission of data using the UDP is shown in the following illustration.

**(1) The transmission user program sends a request to send 1,984 bytes of data.**

**(2) The EtherNet/IP Unit fragments the send data into Data A with 1,472 bytes and Data B with 512 bytes.**

**(3) Data A and Data B are sent consecutively.**

**(4) When the receiving user program sends a request to receive 1,984 bytes of data, Data A and Data B are linked to restore the original data, which is passed to the user program.**



As shown above, the UDP protocol handles data communications as datagrams, so that the send data is restored to the original data before being passed to the user program.

Consequently, if the data length in the receive request is set to the length of the send data, the entire data can be received using a single receive data request.

However, if the data length in the receive data request is set smaller than the actual length of the data, all received data exceeding the set data length will be discarded.

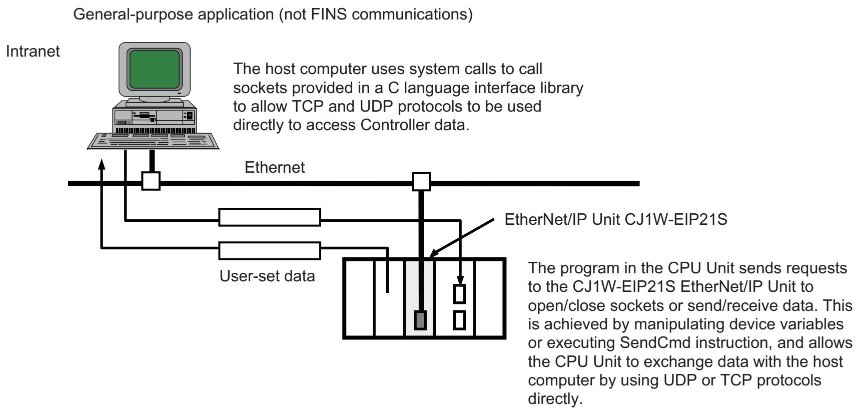
# 13-3 Overview of CJ1W-EIP21S EtherNet/IP Unit Socket Services

This section provides an overview of socket services provided by CJ1W-EIP21S EtherNet/IP Units.

## 13-3-1 Overview

The CJ1W-EIP21S EtherNet/IP Unit's socket services are used to exchange data between the Controller and general-purpose applications that use UDP or TCP.

The socket services can be requested by the Controller by manipulating device variables or by using the SendCmd instruction in the user program.


General-purpose application (not FINS communications)
Intranet
The host computer uses system calls to call sockets provided in a C language interface library to allow TCP and UDP protocols to be used directly to access Controller data.
Ethernet
EtherNet/IP Unit CJ1W-EIP21S
User-set data
The program in the CPU Unit sends requests to the CJ1W-EIP21S EtherNet/IP Unit to open/close sockets or send/receive data. This is achieved by manipulating device variables or executing SendCmd instruction, and allows the CPU Unit to exchange data with the host computer by using UDP or TCP protocols directly.

The two methods of using the socket services are as follows:

- Manipulating device variables
  Requests can be made to a socket service by only setting parameters and manipulating device variables.
- Using the SendCmd instruction
  Requests can be made to a socket service by sending service request commands to the CJ1W-EIP21S EtherNet/IP Unit.

To use the CJ1W-EIP21S's socket services with tag data links on the CJ1W-EIP21S, use the SendCmd instruction. Do not use socket services by manipulating device variables. Doing so may result in deterioration in the response performance of the tag data links.

**Note** One of the main functional differences between manipulating device variables and using the SendCmd instruction is in the number of sockets that can be connected simultaneously, as shown in the following table.

| Protocol | Manipulating device variables | Using the SendCmd instruction |
|---|---|---|
| UDP | Total of 8 sockets max. | 8 sockets max. |
| TCP | | 8 sockets max. |

## 13-3-2 Using Socket Services by Manipulating Device Variables

Using socket services by manipulating device variables is a method of using socket services by only setting parameters and manipulating device variables. Socket services can be used by setting the required parameters in device variables in a Socket Service Parameter Area and then changing the device variables for Socket Service Request Switches to TRUE.

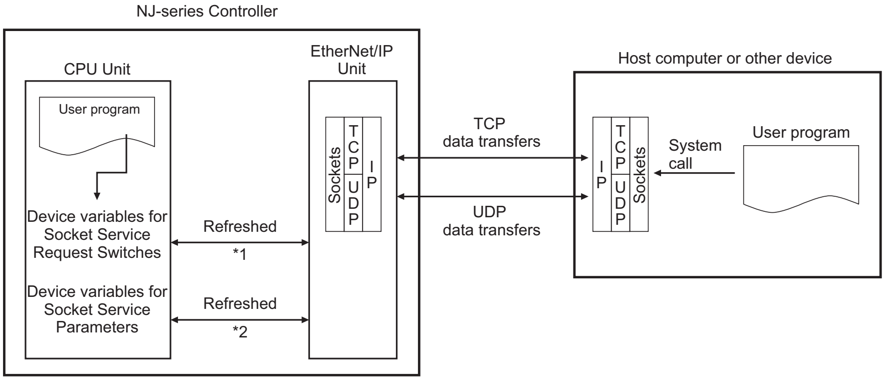
When using Socket Service Request Switches, a maximum of 8 sockets can be opened simultaneously for the UDP and TCP combined.

Also, the same socket number cannot be used simultaneously for both UDP and TCP. (There is only one Socket Service Parameter Area for each socket, i.e., the same area must be used for both UDP and TCP.)

### Precautions for Correct Use

To use the CJ1W-EIP21S's socket services when tag data links are used on the CJ1W-EIP21S, use the SendCmd instruction. Do not use socket services by manipulating device variables. Doing so may result in deterioration in the response performance of the tag data links.

An illustration of using Socket Service Request Switches to execute socket services is provided below.



Note 1. Device variables for Socket Service Request Switches are used to send a service request from the CPU Unit to the EtherNet/IP Unit.
2. Socket service parameter settings with the device variable are used for the CPU Unit to send a service request to the EtherNet/IP Unit, and for the EtherNet/IP Unit to return the processing result to the CPU Unit.

After setting the required parameters in device variables in a Socket Service Parameter Area, use the device variables for Socket Service Request Switches in the above diagram to execute UDP/TCP open, UDP/TCP send, UDP/TCP receive, or UDP/TCP close request.

In UDP/TCP send processing, send data in the variables at the send/receive data addresses set in the Socket Service Parameter Area is sent.

In UDP/TCP receive processing, data is received in the variables at the send/receive data addresses set in the Socket Service Parameter Area.

## Manipulating Device Variables

### Description

The EtherNet/IP Unit’s socket services are used by only setting parameters and manipulating device variables.

### Point

This method is used by setting the required parameters in device variables in a Socket Service Parameter Area of the EtherNet/IP Unit and then changing the device variables for Socket Service Request Switches to TRUE.

### Advantages/Disadvantages

A total of eight ports (UDP and TCP combined) can be used for socket services.

## 13-3-3 Using Socket Services with SendCmd Instruction

Service request commands can be sent to the EtherNet/IP Unit by executing the SendCmd instruction in the ladder diagram.

Up to 16 sockets can be connected using SendCmd instruction: 8 UDP sockets and 8 TCP sockets.

Requests sent to the EtherNet/IP Unit by sending commands through execution of SendCmd, and when the Unit receives a command, it will return a response.

The response does not, however, indicate that processing has been completed, and the status of the flags in the Socket Status device variables must be used to determine when processing has been completed.

The results of processing will be stored in the results storage area specified when the SendCmd instruction was executed once the requested processing has been completed.

## Using the SendCmd Instruction

### Description

The socket services are used by sending service request commands to the EtherNet/IP Unit.

### Point

Service request commands can be sent to the EtherNet/IP Unit by executing the SendCmd instruction in the ladder diagram.

### Advantages/Disadvantages

- A total of 16 sockets, comprising eight TCP ports and eight UDP ports, can be used.
- Using the SendCmd instruction is slower than manipulating device variables.

## 13-3-4 Specific Socket Service Functions

The socket service functions listed in the following table can be executed either by manipulating device variables for Socket Service Request Switches or using SendCmd.

| Protocol | Socket service request |
|---|---|
| UDP | Open UDP socket |
| | Receive via UDP socket |
| | Send via UDP socket |
| | Close UDP socket |
| TCP | Open TCP socket, passive |
| | Open TCP socket, active |
| | Receive via TCP socket |
| | Send via TCP socket |
| | Close TCP socket |

# 13-4 Settings Required for Socket Service Function

The following settings must be made in the Unit Setup when using socket services.

| Sysmac Studio Unit Settings Tab | Setting | Setting requirements | Reference |
|---|---|---|---|
| TCP/IP | IP Address | Optional | *5-2 TCP/IP Settings Display* |
| | Subnet mask | Optional | |
| | IP Router Table | Optional (Set when EtherNet/IP Unit will communicate through the IP router with a socket on another IP network segment) | |
| | TCP/IP keep-alive | Optional (Change when the default setting of 5 min is unacceptable.) | |
| | Socket Service - Speed Up<br>(High-speed socket service) | Optional (Set to improve the sending and receiving performance of manipulating device variables for socket services. Using socket services with the SendCmd instruction will result in an error at high speeds, however.) | |
| Status Area | Layout Type | Required<br>Set this to *User defined*. | *5-8 Status Area Settings Display* |
| | Allocation Area | Required<br>Set the first word of the user setting area to allocate. | |

# 13-5 Socket Service Status

## 13-5-1 UDP/TCP Socket No. 1 to No. 8 Status (EtherNet/IP Unit to CPU Unit)

The status of UDP and TCP sockets is reflected in the UDP/TCP socket status device variables. Refer to the following information in *3-2-5 Device Variables for the CJ-series Unit Socket Service-related Data (CJ1W-EIP21S Only)* for the device variables.

*UDP Socket No. 1 to No. 8 Status (EtherNet/IP Unit to CPU Unit)*

*TCP Socket No. 1 to No. 8 Status (EtherNet/IP Unit to CPU Unit)*

## 13-5-2 Number of Bytes Received at TCP Socket (EtherNet/IP Unit to CPU Unit)

The number of bytes of received data saved in the reception buffer at a TCP socket is stored in the corresponding Number of Bytes Received at TCP Socket device variable. Refer to the following section for the device variables.

*Number of Bytes Received at TCP Socket No. 1 to 8 (EtherNet/IP Unit to CPU Unit) (*_Skt1TcpRcvDatByte to *_Skt8TcpRcvDatByte)* on page 3-24

The Data Received Flag device variable is changed to TRUE/FALSE in response to this variable. When the receive request is sent by manipulating the device variable or by executing the SendCmd instruction, the value of this device variable is temporarily set to 0000 hex.

If the data remains in the reception buffer after the receive request processing is complete, the number of remaining bytes is stored in the Number of Bytes Received at TCP Socket and the Data Received Flag changes to TRUE again.

Depending on the timing of data reception, the number of received data bytes may be 0 even if the Data Received Flag is changed to TRUE. To prevent this, use flag control to receive data according to the sample programs.

Refer to *A-9 Sample Programming for Socket Services* for sample programming.

Receive requests should be executed after confirming that the required data is contained in the number of bytes received.

Up to 4,096 bytes of data are stored in the reception buffer, but the value stored is within the range (maximum: 1,984 bytes) that can be set by manipulating the device variable or sending the receive request in the SendCmd instruction.

0000 hex: 0 bytes

07C0 hex: 1,984 bytes

## 13-5-3 TCP Connection Status (EtherNet/IP Unit to CPU Unit)

The TCP Connection Status shows the status of a port that has been opened using the TCP socket.

This port status is stored even after the port is closed, and remains until the socket is used to open the port again.

The TCP Connection Status is not synchronized with the Socket Status Area, however, so the status conversion timing is slightly different.

Refer to the following section for the device variables.

*TCP Socket No. 1 to 8 Connection Status (EtherNet/IP Unit to CPU Unit) (*_Skt1TcpConSta to *_Skt8TcpConSta)* on page 3-25

The value indicates the connection status as follows.

| Value | Status | Meaning |
|---|---|---|
| 0 | CLOSED | Connection closed. |
| 1 | LISTEN | Waiting for connection. |
| 2 | SYN SENT | SYN sent in active status. |
| 3 | SYN RECEIVED | SYN received and sent. |
| 4 | ESTABLISHED | Already established. |
| 5 | CLOSE WAIT | FIN received and waiting for completion. |
| 6 | FIN WAIT1 | Completed and FIN sent. |
| 7 | CLOSING | Completed and exchanged FIN.<br>Awaiting ACK. |
| 8 | LAST ACK | FIN sent and completed.<br>Awaiting ACK. |
| 9 | FIN WAIT2 | Completed and ACK received.<br>Awaiting FIN. |
| A | TIME WAIT | After closing, pauses twice the maximum segment life (2MSL). |

# 13-6 Using Socket Services by Manipulating Device Variables

## 13-6-1 Application Procedure

### Procedure

**1** Make the settings that are required for the socket service function.

Refer to *13-4 Settings Required for Socket Service Function* for the required settings.

**2** Set socket service parameters in device variables in a Socket Service Parameter Area.

Device variables for Socket Service Parameter Area 1

| Device variable for CJ-series Units | Name |
|---|---|
| *_Skt1ParamArea | Socket Services Parameter Area 1 |
| *_Skt1No | UDP/TCP Socket Number |
| *_Skt1Option | Socket Option |
| *_Skt1KeepAlive | Keep-alive Function Used |
| *_Skt1Linger | Linger Function Used |
| *_Skt1LocalPortNo | Local UDP/TCP Port Number |
| *_Skt1RemoteIPAdr | Remote IP Address |
| *_Skt1RemoteIPAdr1 | 1st Byte of Remote IP Address |
| *_Skt1RemoteIPAdr2 | 2nd Byte of Remote IP Address |
| *_Skt1RemoteIPAdr3 | 3rd Byte of Remote IP Address |
| *_Skt1RemoteIPAdr4 | 4th Byte of Remote IP Address |
| *_Skt1RemotePortNo | Remote UDP/TCP Port Number |
| *_Skt1SendRcvByte | Number Of Send/Receive Bytes |
| *_Skt1SendRcvAdr | Send/Receive Data Address |
| *_Skt1SendRcvAdrType | Area Type |
| *_Skt1SendRcvAdrCh1 | Channel Address(Upper 2 digits) |
| *_Skt1SendRcvAdrCh2 | Channel Address(Lower 2 digits) |
| *_Skt1Timeout | Timeout Value |
| *_Skt1RespCode | Response Code |

**3** Create variables to store send or receive data and specify AT specifications to the memory addresses set in the send/receive data address in a Socket Service Parameter Area.

Refer to *Send/Receive Data Address* on page 13-22 for details on the specification method.

**4** Change the device variables for Socket Service Request Switches from FALSE to TRUE.

Device variables for Socket Service Request Switch 1

| Device variable for CJ-series Units | | Name |
|---|---|---|
| *_Skt1Cmd | | Socket Service Request Switch 1 |
| | *_Skt1UdpOpenCmd | UDP Open Request Switch |
| | *_Skt1TcpPassiveCmd | TCP Passive Open Request Switch |
| | *_Skt1TcpActiveCmd | TCP Active Open Request Switch |
| | *_Skt1SendCmd | Send Request Switch |
| | *_Skt1RcvCmd | Receive Request Switch |
| | *_Skt1CloseCmd | Close Request Switch |

**5** When a send or receive request is made, data of the variable with an AT specification specified will be automatically sent to or received from the send/receive data address in the Socket Service Parameter Area.

When processing has been completed, a response code will be automatically stored in the Socket Service Parameters.

## Precautions

A Socket Service Parameter Area cannot be used for other sockets once open processing has been successfully completed for it.

Check the socket status according to the TCP connection status of each socket. The performance of sending and receiving has been improved using optional settings for the TCP or UDP socket services using device variables. Also, a linger socket option can be used with the TCP socket services. Selecting this option enables immediate open processing using the same ports without having to wait (approximately 1 min.) until the port number opens after the socket closes.

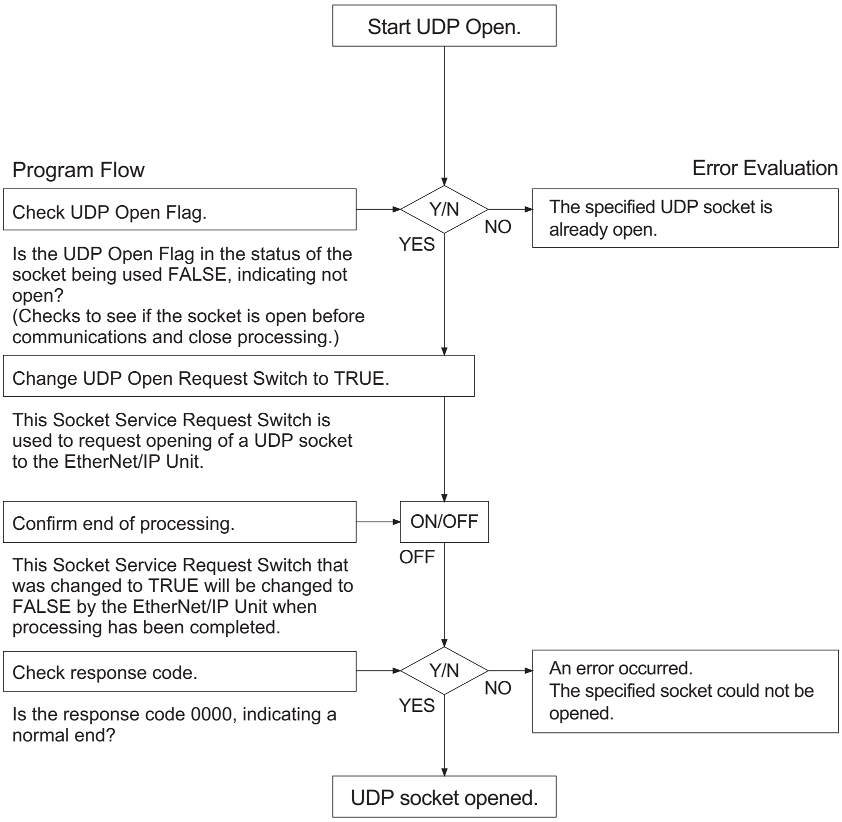
## 13-6-2 Using Socket Services and Socket Status

When using socket services, it is important to consider the timing of the status changes in the Socket Status Area.

The diagram below shows a flowchart for opening UDP.

The flow is similar for other socket services.

Replace the names of the appropriate flags in the flowchart to adapt it to other socket services.


Start UDP Open.
Program Flow
Error Evaluation
Check UDP Open Flag.
Y/N
NO
YES
The specified UDP socket is already open.
Is the UDP Open Flag in the status of the socket being used FALSE, indicating not open?
(Checks to see if the socket is open before communications and close processing.)
Change UDP Open Request Switch to TRUE.
This Socket Service Request Switch is used to request opening of a UDP socket to the EtherNet/IP Unit.
Confirm end of processing.
ON/OFF
OFF
This Socket Service Request Switch that was changed to TRUE will be changed to FALSE by the EtherNet/IP Unit when processing has been completed.
Check response code.
Y/N
NO
YES
An error occurred.
The specified socket could not be opened.
Is the response code 0000, indicating a normal end?
UDP socket opened.

## 13-6-3 Socket Service Parameters (between CPU Unit and EtherNet/IP Unit)

The device variables for socket service parameters used in a socket service request are shown below

Device variables for Socket Service Parameter Area 1

| Device variable for CJ-series Units | Name |
|---|---|
| *_Skt1ParamArea | Socket Services Parameter Area 1 |
| *_Skt1No | UDP/TCP Socket Number |
| *_Skt1Option | Socket Option |
| *_Skt1KeepAlive | Keep-alive Function Used |
| *_Skt1Linger | Linger Function Used |
| *_Skt1LocalPortNo | Local UDP/TCP Port Number |
| *_Skt1RemoteIPAdr | Remote IP Address |
| *_Skt1RemoteIPAdr1 | 1st Byte of Remote IP Address |
| *_Skt1RemoteIPAdr2 | 2nd Byte of Remote IP Address |
| *_Skt1RemoteIPAdr3 | 3rd Byte of Remote IP Address |
| *_Skt1RemoteIPAdr4 | 4th Byte of Remote IP Address |
| *_Skt1RemotePortNo | Remote UDP/TCP Port Number |
| *_Skt1SendRcvByte | Number Of Send/Receive Bytes |
| *_Skt1SendRcvAdr | Send/Receive Data Address |
| *_Skt1SendRcvAdrType | Area Type |
| *_Skt1SendRcvAdrCh1 | Channel Address(Upper 2 digits) |
| *_Skt1SendRcvAdrCh2 | Channel Address(Lower 2 digits) |
| *_Skt1Timeout | Timeout Value |
| *_Skt1RespCode | Response Code |

to

Device variables for Socket Service Parameter Area 8

| Device variable for CJ-series Units | Name |
|---|---|
| *_Skt8ParamArea | Socket Services Parameter Area 8 |
| *_Skt8No | UDP/TCP Socket Number |
| *_Skt8Option | Socket Option |
| *_Skt8KeepAlive | Keep-alive Function Used |
| *_Skt8Linger | Linger Function Used |
| *_Skt8LocalPortNo | Local UDP/TCP Port Number |
| *_Skt8RemoteIPAdr | Remote IP Address |
| *_Skt8RemoteIPAdr1 | 1st Byte of Remote IP Address |
| *_Skt8RemoteIPAdr2 | 2nd Byte of Remote IP Address |
| *_Skt8RemoteIPAdr3 | 3rd Byte of Remote IP Address |
| *_Skt8RemoteIPAdr4 | 4th Byte of Remote IP Address |
| *_Skt8RemotePortNo | Remote UDP/TCP Port Number |
| *_Skt8SendRcvByte | Number Of Send/Receive Bytes |
| *_Skt8SendRcvAdr | Send/Receive Data Address |
| *_Skt8SendRcvAdrType | Area Type |
| *_Skt8SendRcvAdrCh1 | Channel Address(Upper 2 digits) |
| *_Skt8SendRcvAdrCh2 | Channel Address(Lower 2 digits) |
| *_Skt8Timeout | Timeout Value |
| *_Skt8RespCode | Response Code |

## Required Parameters for Each Socket Service

The following table shows the parameters that are required for each service and the use of the parameters by the socket service.

| Parameter | Type | Range (decimal values in parentheses) | Socket service | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | UDP open | UDP receive | UDP send | UDP close | TCP passive open | TCP active open | TCP receive | TCP send | TCP close |
| UDP/TCP socket No. | USINT | 0001 to 0008 hex<br>(1 to 8) | W | W | W | W | W | W | W | W | W |
| Socket Option | BYTE | Specified bit | --- | --- | --- | --- | W | W | --- | --- | --- |
| Local UDP/TCP port No. | WORD | 0000 to FFFF hex<br>(0 to 65,535) | W | --- | --- | --- | W | RW | --- | --- | --- |
| Remote IP address | DWORD | 00000000 to FFFFFFFF hex<br>(0.0.0.0 to 255.255.255.255) | --- | R | W | --- | RW | W | --- | --- | --- |
| Remote UDP/TCP port No. | WORD | 0000 to FFFF hex<br>(0 to 65,535) | --- | R | W | --- | RW | W | --- | --- | --- |
| Number of bytes to send/receive | WORD | 0000 to 07C0 hex<br>(0 to 1,984 bytes) | --- | RW | RW | --- | --- | --- | RW | RW | --- |
| Send/Receive data address | DWORD | Refer to *13-6-4 Parameters*. | --- | W | W | --- | --- | --- | W | W | --- |
| Timeout time (Unit: 100 ms) | WORD | 0000 to FFFF hex<br>(0 to 65,535)<br>0: No limit<br>0.1 to 6,553.5 s | --- | W | | --- | W | --- | W | --- | --- |
| Response code | WORD | | R | R | R | R | R | R | R | R | R |

W: Required for user to write parameters

RW: Required for user to write parameters at start of execution and read results at completion

R: Required for user to read results at completion

---: Not required

## 13-6-4 Parameters

### UDP/TCP Socket No.

Specify the number of the UDP or TCP socket to open.

### Socket Option

For the TCP OPEN REQUEST (ACTIVE or PASSIVE) command, specify whether or not the keep-alive function is to be used.

When the keep-alive function is used, change *_*KeepAlive* to TRUE (set to 1).

When the linger function is used, change *_*Linger* to TRUE (set to 1).

### Local UDP/TCP Port No.

Specify the number of the UDP or TCP port for the socket to use for communications.

However, the following precautions apply.

- At the time of UDP OPEN REQUEST, do not specify the ports that are used as FINS UDP port number (default value: 9600) and UDP port number for CIP communications (default value: 2222,44818).
- At the time of TCP OPEN REQUEST, do not specify FTP server TCP port numbers 20 and 21, and FINS TCP port number (default value: 9600).
- As a rule, use port numbers 1,024 and higher.

If port number 0 is specified when for an active TCP open, the TCP port number will be automatically allocated and the number of the port that was opened will be stored in the local UDP/TCP port number in the Socket Service Parameter Area (i.e., the actual port number will be overwritten on the value of 0 set by the user).

### Remote IP Address

Specify the IP address of the remote device.

However, the following precautions apply.

- This parameter is not used when making a receive request for a UDP socket. The source IP address will be stored with the response data and will be written to the device variable for the Remote IP Address in the Socket Service Parameter Area.
- When opening a passive TCP socket, the combination of the remote IP address and the remote TCP port number can be used to affect processing as shown in the following table.

| Remote IP Address | Remote TCP Port No. | Processing |
|---|---|---|
| 0 | 0 | All connection requests accepted. |
| 0 | Not 0 | Connection requests accepted only for the same port number. |
| Not 0 | 0 | Connection requests accepted only for the same IP address. |
| Not 0 | Not 0 | Connection requests accepted only for the same port number and IP address. |

If the Remote IP Address is set to 0, a connection can be made to any remote node and the remote IP address of the node that is connected will be stored as the Remote IP Address in the Socket Service Parameter Area.

If a specific remote I/O address is set, then a connection can be made only to the node with the specified address.

If the Remote TCP Port No. is set to 0, a connection can be made to any remote node regardless of the TCP port number it is using.

If a specific remote TCP port number is set, then a connection can be made only to a node using the specified TCP port number.

## Remote UDP/TCP Port No.

Specify the UDP or TCP port number used by the remote device.

However, the following precautions apply.

- This parameter is not used when making a receive request for a UDP socket. The remote UDP/TCP port number will be stored with the response data and will be written as the Remote UDP/TCP Port No. in the Socket Service Parameter Area.
- When opening a passive TCP socket, the combination of the remote IP address and the remote TCP port number can be used to affect processing as shown in the table for the Remote IP Address, above. If the Remote UDP/TCP Port No. is set to 0, the UDP/TCP port number of the remote device will be written as the Remote UDP/TCP Port No. in the Socket Service Parameter Area.

## Number of Bytes to Send/Receive

Specify the number of bytes to send or receive when a send or receive request is made.

When the transfer has been completed, the actual number of bytes that have been sent or received will be written here.

## Send/Receive Data Address

Specify the memory address by using an AT specification for the variable to store the data to send or receive when a send or receive request is made, as follows.

| Setting | Description |
|---|---|
| Area type | Set the **Area type** value in the following table. |
| Channel Address(Upper 2 digits) | Set the upper two digits of the **Addresses for communications** value in the following table. |
| Channel Address(Lower 2 digits) | Set the lower two digits of the **Addresses for communications** value in the following table. |

The following table shows the areas and setting values that you can specify.

| Memory area | Data type | | Area Type | Word addresses | Addresses for communications |
|---|---|---|---|---|---|
| Bit Areas | Current value of word | CIO | B0 | 0000 to 6143 | 0000 to 17FF |
| | | HR | B2 | H000 to H511 | 0000 to 01FF |
| DM Area | Current value of word | DM | 82 | D00000 to D32767 | 0000 to 7FFF |
| EM Area | Current value of word | Bank 0 | A0 | E0_E00000 to E0_E32767 | 0000 to 7FFF |
| | | Bank 1 | A1 | E1_E00000 to E1_E32767 | 0000 to 7FFF |
| | | Bank 2 | A2 | E2_E00000 to E2_E32767 | 0000 to 7FFF |
| | | Bank 3 | A3 | E3_E00000 to E3_E32767 | 0000 to 7FFF |
| | | Bank 4 | A4 | E4_E00000 to E4_E32767 | 0000 to 7FFF |
| | | Bank 5 | A5 | E5_E00000 to E5_E32767 | 0000 to 7FFF |
| | | Bank 6 | A6 | E6_E00000 to E6_E32767 | 0000 to 7FFF |
| | | Bank 7 | A7 | E7_E00000 to E7_E32767 | 0000 to 7FFF |
| | | Bank 8 | A8 | E8_E00000 to E8_E32767 | 0000 to 7FFF |
| | | Bank 9 | A9 | E9_E00000 to E9_E32767 | 0000 to 7FFF |
| | | Bank A | AA | EA_E00000 to EA_E32767 | 0000 to 7FFF |
| | | Bank B | AB | EB_E00000 to EB_E32767 | 0000 to 7FFF |
| | | Bank C | AC | EC_E00000 to EC_E32767 | 0000 to 7FFF |
| | | Current bank | 98 | E00000 to E32767 | 0000 to 7FFF |

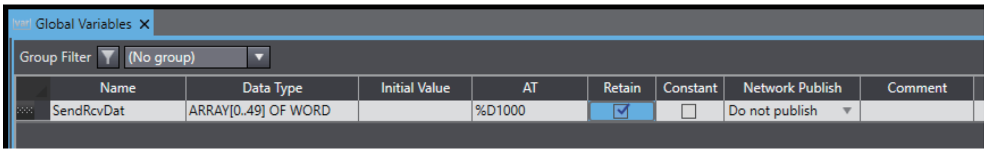
Socket services function handles data in memory used for the CJ-series Units in units of words. Therefore, for variables with AT specifications, specify two-byte data types such as WORD. Refer to *Section 6 Programming* in the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) for information on data types.

Example of using word D1000 of the DM Area words as the send/receive data address and *SendRcvDat* as the name of the variable to store the send/receive data, which is 100 bytes (50 words) of WORD data

- Since the area type of the DM area is 82 hex and the word address is 1000 (03E8 hex), configure the settings as follows.

| Setting | Set value |
|---|---|
| Area Type | 82 |
| Channel Address(Upper 2 digits) | 03 |
| Channel Address(Lower 2 digits) | E8 |

- Create a global variable *SendRcvDat* to store 100 bytes (50 words) of WORD data and specify its AT specification to the word address D1000 hex.



| Name | Data Type | Initial Value | AT | Retain | Constant | Network Publish | Comment |
|---|---|---|---|---|---|---|---|
| SendRcvDat | ARRAY[0..49] OF WORD | | %D1000 | ☑ | □ | Do not publish | |

## ● Timeout Time

Set the time limit in units of 0.1 s for completion of communications from the time that the Receive Request Switch (TCP or UDP) or the TCP Passive Open Request Switch is changed to TRUE.

A response code of 0080 hex (timeout) will be stored if communications time out.

If 0 is set, the requested service will not be timed.

## 13-6-5 Socket Service Request Switches (between CPU Unit and EtherNet/IP Unit)

The following device variables for Socket Service Request Switches can be manipulated to request socket services.

Socket No. 1

| Device variable for CJ-series Units | Name | Function |
|---|---|---|
| *_Skt1Cmd | Socket Service Request Switch 1 | Requests socket services. |
| *_Skt1UdpOpenCmd | UDP Open Request Switch | Opens UDP socket when switch is changed to TRUE.<br>FALSE: Open processing completed<br>Default: FALSE |
| *_Skt1TcpPassiveCmd | TCP Passive Open Request Switch | Opens TCP socket when switch is changed to TRUE.<br>FALSE: Open processing completed (connection established)<br>Default: FALSE |
| *_Skt1TcpActiveCmd | TCP Active Open Request Switch | Opens TCP socket when switch is changed to TRUE.<br>FALSE: Open processing completed (connection established)<br>Default: FALSE |
| *_Skt1SendCmd | Send Request Switch | Executes send processing when switch is changed to TRUE. (The protocol (TCP/UDP) is determined when the socket is opened.)<br>FALSE: Send processing completed<br>Default: FALSE |
| *_Skt1RcvCmd | Receive Request Switch | Executes receive processing when switch is changed to TRUE. (The protocol (TCP/UDP) is determined when the socket is opened.)<br>FALSE: Receive processing completed<br>Default: FALSE |
| *_Skt1CloseCmd | Close Request Switch | Closes TCP socket when switch is changed to TRUE.<br>FALSE: Close processing completed<br>Default: FALSE |

to

Socket No. 8

| Device variable for CJ-series Units | Name | Function |
|---|---|---|
| *_Skt8Cmd | Socket Service Request Switch 8 | Requests socket services. |
| *_Skt8UdpOpenCmd | UDP Open Request Switch | Opens UDP socket when switch is changed to TRUE.<br>FALSE: Open processing completed<br>Default: FALSE |
| *_Skt8TcpPassiveCmd | TCP Passive Open Request Switch | Opens TCP socket when switch is changed to TRUE.<br>FALSE: Open processing completed (connection established)<br>Default: FALSE |
| *_Skt8TcpActiveCmd | TCP Active Open Request Switch | Opens TCP socket when switch is changed to TRUE.<br>FALSE: Open processing completed (connection established)<br>Default: FALSE |
| *_Skt8SendCmd | Send Request Switch | Executes send processing when switch is changed to TRUE. (The protocol (TCP/UDP) is determined when the socket is opened.)<br>FALSE: Send processing completed<br>Default: FALSE |
| *_Skt8RcvCmd | Receive Request Switch | Executes receive processing when switch is changed to TRUE. (The protocol (TCP/UDP) is determined when the socket is opened.)<br>FALSE: Receive processing completed<br>Default: FALSE |
| *_Skt8CloseCmd | Close Request Switch | Closes TCP socket when switch is changed to TRUE.<br>FALSE: Close processing completed<br>Default: FALSE |

As shown in the above table, the Request Switches are changed to TRUE by the EtherNet/IP Unit when the requested processing has been completed.

### Additional Information

- In addition to the Socket Service Request Switches, there is also a Force Socket Close Switch (*_SktForceCloseCmd*).
  When this device variable is changed to TRUE, all sockets that are open will be force-closed. Refer to *Section 3 Assigning Device Variables for CJ-series Units* for details.
- When using socket services by manipulating device variables, the ladder diagram should be programmed to check the response codes when Socket Service Request Switches are changed to FALSE.

## 13-6-6 Response Codes

When processing of a request has been completed for socket services executed by manipulating device variables for Socket Service Request Switches, a response code will be stored in the Response Code in the Socket Service Parameter Area.

The following response codes will be stored depending on the service that was requested.

### ● UDP Socket Open Request

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0105 | Local IP address setting error. |
| 0302 | CPU Unit error; cannot execute. |
| 1100 | UDP socket number is not 1 to 8 or local UDP port number is 0. |
| 110C | Request Switch changed to TRUE during other processing. |
| 220F | Specified socket is already open. |
| 2211 | Unit is busy; cannot execute service. |
| 2606 | Specified socket is already open as TCP socket; cannot open UDP socket. |
| 2607 | Specified Socket Service Parameter Area is already being used for another socket. |
| 003E | Internal buffer cannot be obtained due to high reception traffic (ENOBUFS). |
| 0049 | Duplicate UDP port number (EADDRINUSE). |
| 0081 | Specified socket was closed during open processing. |

### ● UDP Socket Receive Request

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0302 | CPU Unit error; cannot execute. |
| 1100 | Number of bytes to receive is not in allowable range. |
| 1101 | Area type for the Send/Receive Data Address is not in allowable range. |
| 1103 | Bit number in the Send/Receive Data Address is not 00. |
| 110C | Request Switch changed to TRUE during other processing. |
| 220F | Specified socket is already processing a receive request. |
| 2210 | Specified socket is not open. |
| 2211 | Unit is busy; cannot execute service. |
| 2607 | Specified Socket Service Parameter Area is already being used for another socket. |
| 003E | Internal buffer cannot be obtained due to high reception traffic (ENOBUFS). |
| 0066 | Internal memory cannot be obtained; cannot execute service. |
| 0080 | Receive request timed out. |
| 0081 | Specified socket was closed during reception processing. |

## ● UDP Socket Send Request

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0302 | CPU Unit error; cannot execute. |
| 1100 | Number of bytes to send is not in allowable range or the destination IP address is 0. |
| 1101 | Area type for the Send/Receive Data Address is not in allowable range. |
| 1103 | Bit number in the Send/Receive Data Address is not 00. |
| 110C | Request Switch changed to TRUE during other processing. |
| 220F | Specified socket is already processing a send request. |
| 2210 | Specified socket is not open. |
| 2211 | Unit is busy; cannot execute service. |
| 2607 | Specified Socket Service Parameter Area is already being used for another socket. |
| 003E | Internal buffer cannot be obtained due to high reception traffic (ENOBUFS). |
| 0042 | Destination IP address is a broadcast address and the number of bytes sent is greater than 1,472 bytes (EMSGSIZE). |
| 004C | Network ID is incorrect or the destination IP address is incorrect (EADDRNOTAVAIL). |
| 004E | Network ID is not in the IP router table, router settings are incorrect, or the destination IP address is incorrect (ENETUNREACH). |
| 0051 | Router settings are incorrect or the destination IP address is incorrect (EHOSTUNREACH). |
| 0081 | Specified socket was closed during send processing. |

## ● UDP Socket Close Request

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 2210 | Specified socket is not open. |
| 2211 | Unit is busy; cannot execute service. |
| 2607 | Specified Socket Service Parameter Area is already being used for another socket. |
| 0302 | CPU Unit error; cannot execute. |

## ● TCP Socket Passive Open Request

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0105 | Local IP address setting error. |
| 0302 | CPU Unit error; cannot execute. |
| 1100 | TCP socket number is not 1 to 8 or local TCP port number is 0. |
| 110C | Request Switch changed to TRUE during other processing. |
| 220F | Specified socket (connection) is already open or already processing an open request. |
| 2211 | Unit is busy; cannot execute service. |
| 2606 | Specified socket is already open as UDP socket; cannot open TCP socket. |
| 2607 | Specified Socket Service Parameter Area is already being used for another socket. |
| 003E | Internal buffer cannot be obtained due to high reception traffic (ENOBUFS). |
| 0042 *1 | Error occurred. (EMSGSIZE). |
| 0045 | Error in communications with remote node (ECONNABORTED). |
| 0049 | Duplicate TCP port number (EADDRINUSE). |

| Response code | Meaning |
|---|---|
| 004A [*1] | Error occurred. (ECONNREFUSED). |
| 004B [*1] | Error in communications with remote node (ECONNRESET). |
| 004E [*1] | Remote IP address parameter error (ENETUNREACH). |
| 0051 [*1] | Remote IP address parameter error (EHOSTUNREACH). |
| 0053 | Error in communications with remote node (ETIMEDOUT) or remote node does not exist. |
| 0066 | Internal memory cannot be obtained; cannot execute service. |
| 0080 | Open request timed out. |
| 0081 | Specified socket was closed during open processing. |
| 0082 | Connection could not be established with specified remote node. |

*1 These response codes will be returned only on large, multilevel networks.

### ● TCP Socket Active Open Request

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0105 | Local IP address setting error. |
| 0302 | CPU Unit error; cannot execute. |
| 1100 | TCP socket number is not 1 to 8 or remote IP address is 0. |
| 110C | Request Switch changed to TRUE during other processing. |
| 220F | Specified socket is already open or already processing an open request. |
| 2211 | Unit is busy; cannot execute service. |
| 2606 | Specified socket is already open as UDP socket; cannot open TCP socket. |
| 2607 | Specified Socket Service Parameter Area is already being used for another socket. |
| 000D | Remote IP address parameter error (EACCES). |
| 003E | Internal buffer cannot be obtained due to high reception traffic (ENOBUFS). |
| 0042 [*1] | Error occurred. (EMSGSIZE). |
| 0044 | ICMP data received (ENOPROTOOPT). |
| 0045 | Local socket was closed (ECONNABORTED). |
| 0049 | Duplicate TCP port number (EADDRINUSE). |
| 004A | Error (ECONNREFUSED) or the remote node has not been opened as passive socket. |
| 004B [*1] | Error in communications with remote node (ECONNRESET). |
| 004C | Remote IP address parameter error (EADDRNOTAVAIL).<br>Wrong parameter designation.<br>Attempt was made to set the local TCP port of the local node to Active Open. |
| 004E | Remote IP address parameter error (ENETUNREACH).<br>Network ID is not in the IP router table or router settings are incorrect. |
| 0051 | Remote IP address parameter error (EHOSTUNREACH).<br>Router settings are incorrect. |
| 0053 | Error in communications with remote node (ETIMEDOUT) or remote node does not exist. |
| 0081 | Specified socket was closed during open processing. |

*1 These response codes will be returned only on large, multilevel networks.

### TCP Socket Receive Request

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0302 | CPU Unit error; cannot execute. |
| 1100 | Number of bytes to receive is not in allowable range. |
| 1101 | Area type for the Send/Receive Data Address is not in allowable range. |
| 1103 | Bit number in the Send/Receive Data Address is not 00. |
| 110C | Request Switch changed to TRUE during other processing. |
| 220F | Specified socket is already processing a receive request. |
| 2210 | Specified socket has not been connected. |
| 2211 | Unit is busy; cannot execute service. |
| 2607 | Specified Socket Service Parameter Area is already being used for another socket. |
| 003E | Internal buffer cannot be obtained due to high reception traffic (ENOBUFS). |
| 0042 [*1] | ICMP data received (EMSGSIZE). |
| 0044 [*1] | ICMP data received (ENOPROTOOPT). |
| 0045 [*1] | Error in communications with remote node (ECONNABORTED). |
| 004B | Error in communications with remote node (ECONNRESET). |
| 004E [*1] | ICMP data received (ENETUNREACH). |
| 004F [*1] | ICMP data received (EHOSTDOWN). |
| 0051 [*1] | ICMP data received (EHOSTUNREACH). |
| 0053 | Error in communications with remote node (ETIMEDOUT). |
| 0066 | Internal memory cannot be obtained; cannot execute service. |
| 0080 | Receive request timed out. |
| 0081 | Specified socket was closed during reception processing. |

[*1] These response codes will be returned only on large, multilevel networks.

### TCP Socket Send Request

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0302 | CPU Unit error; cannot execute. |
| 1100 | Number of bytes to send is not in allowable range. |
| 1101 | Area type for the Send/Receive Data Address is not in allowable range. |
| 1103 | Bit number in the Send/Receive Data Address is not 00. |
| 110C | Request Switch changed to TRUE during other processing. |
| 220F | Specified socket is already processing a send request. |
| 2210 | Specified socket is not been connected. |
| 2211 | Unit is busy; cannot execute service. |
| 2607 | Specified Socket Service Parameter Area is already being used for another socket. |
| 0020 | Connection with remote socket broken during send (EPIPE). |
| 003E | Internal buffer cannot be obtained due to high reception traffic (ENOBUFS). |
| 0042 [*1] | Error occurred. (EMSGSIZE). |
| 0044 [*1] | ICMP data received (ENOPROTOOPT). |
| 0045 [*1] | Error in communications with remote node (ECONNABOTED). |
| 004A | Error in communications with remote node (ECONNREFUSED). |

| Response code | Meaning |
|---|---|
| 004B [*1] | Error in communications with remote node (ECONNRESET). |
| 004E [*1] | Remote IP address parameter error (ENETUNREACH). |
| 004F [*1] | ICMP data received (EHOSTDOWN). |
| 0051 [*1] | Remote IP address parameter error (EHOSTUNREACH). |
| 0053 [*1] | Error in communications with remote node (ETIMEDOUT). |
| 0081 | Specified socket was closed during send processing. |

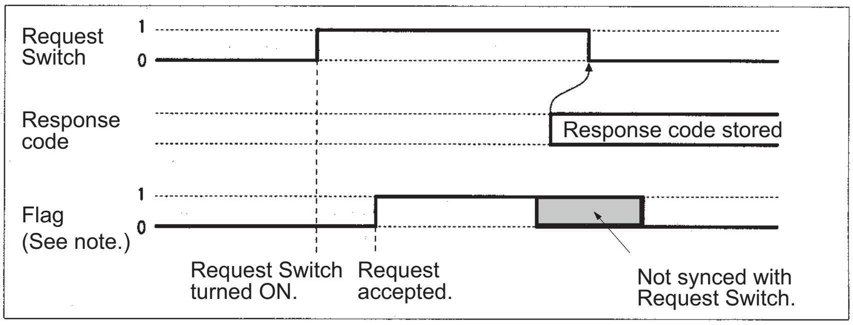
[*1] These response codes will be returned only on large, multilevel networks.

### ● TCP Socket Close Request

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 2210 | Specified socket has not been connected. |
| 2211 | Unit is busy; cannot execute service. |
| 2607 | Specified Socket Service Parameter Area is already being used for another socket. |
| 0302 | CPU Unit error; cannot execute. |

## 13-6-7 Timing Charts

The timing of flags for socket services (Opening, Receiving, Sending, or Closing Flag) when device variables for the Request Switches are manipulated and the changes in the response code are shown in the following chart.



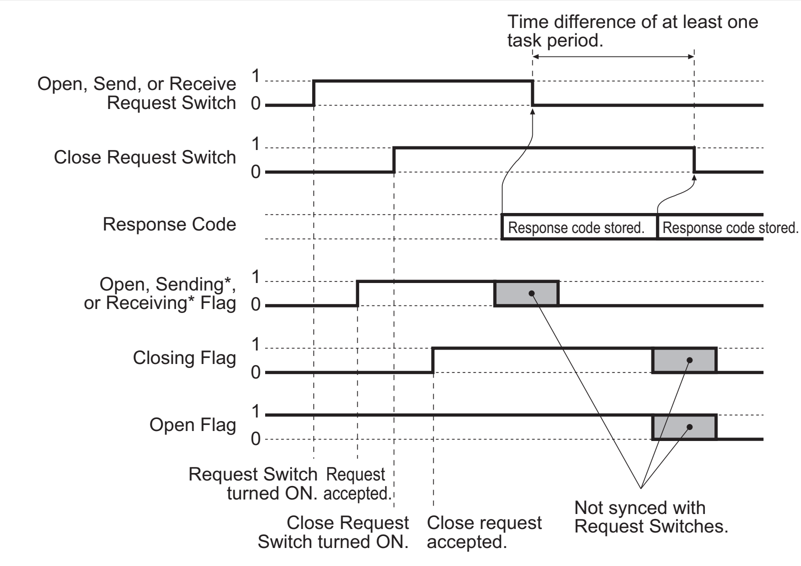
**Note** The Sending Flag and Receiving Flag will not be changed to TRUE if the high-speed socket service option is selected. Therefore, the program can be controlled only by changing the Send Request Switch and Receive Request Switch to FALSE.

## Close Processing When Processing Other Requests

The Close Request Switch or Force-close Switch can be used to close a socket even when open, receive, or send processing is being executed. Only close processing is possible when other requests are being processed.

### ● Close Request Switch

Using the Close Request Switch cancels the processing being executed and stores the processing results as a response code. Therefore, there is always a time difference of at least one task period between when the Request Switch for canceled processing is changed to FALSE and when the Close Request Switch is changed to FALSE, allowing the response codes for both to be read.


Time difference of at least one task period.
Open, Send, or Receive Request Switch
Close Request Switch
Response Code
Response code stored.
Response code stored.
Open, Sending*, or Receiving* Flag
Closing Flag
Open Flag
1
0
Request Switch turned ON.
Request accepted.
Close Request Switch turned ON.
Close request accepted.
Not synced with Request Switches.

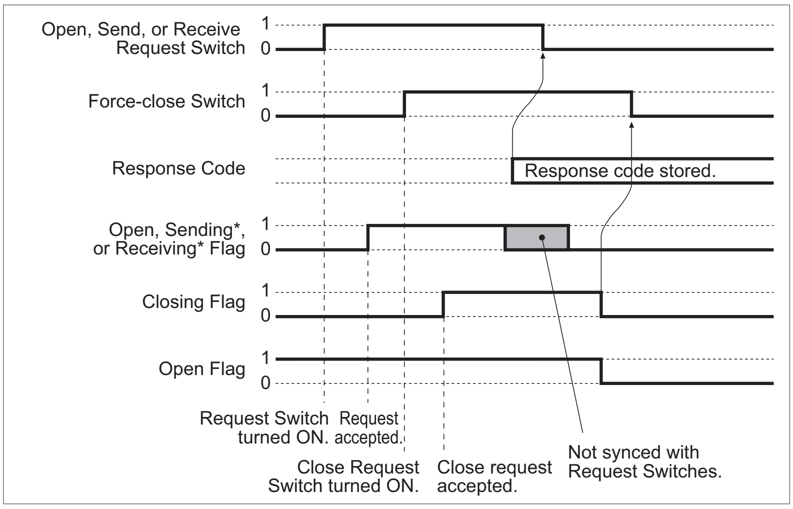
**Note** The Sending Flag and Receiving Flag will not be changed to TRUE if the high-speed socket service option is selected.

**Additional Information**

The Open Flag will remain FALSE if a close request is made during open processing.

## ● Force Socket Close Switch

Using the Force-close Switch cancels the requested processing being executed and stores a response code.


Open, Send, or Receive Request Switch
Force-close Switch
Response Code
Response code stored.
Open, Sending*, or Receiving* Flag
Closing Flag
Open Flag
1
0
Request Switch turned ON.
Request accepted.
Close Request Switch turned ON.
Close request accepted.
Not synced with Request Switches.

**Note** The Sending Flag and Receiving Flag will not be changed to TRUE if the high-speed socket service option is selected.

**Additional Information**

The Open Flag will remain FALSE if a force socket close request is made during open processing.

## 13-6-8 Sample Programming

Refer to *A-9 Sample Programming for Socket Services* for sample programming for socket services by manipulating device variables.

# 13-7 Using Socket Services with SendCmd Instruction

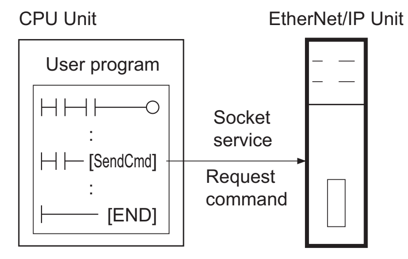
## 13-7-1 Using Socket Services

Each EtherNet/IP Unit has eight TCP sockets and eight UDP sockets. To use these sockets for communications, four types of processing, i.e., open, close, send, and receive, are available.

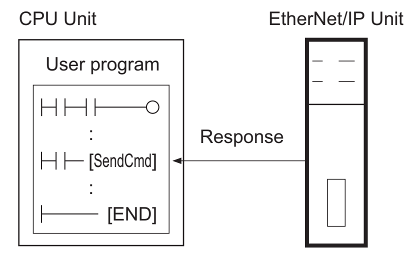
| Processing | Description |
|---|---|
| Open | Enables communications on a specified socket. A socket must be opened before it can be used for socket services. Opening a TCP socket establishes a connection. |
| Close | Ends use of the socket. Breaks the connection for a TCP socket. |
| Send | Sends data from a specified open socket. |
| Receive | Specifies an open socket and receives data from that socket. |

These processes are carried out by sending the SendCmd instruction to the EtherNet/IP Unit. The processing flow from the start to completion of the SendCmd instruction is shown in the following illustrations.

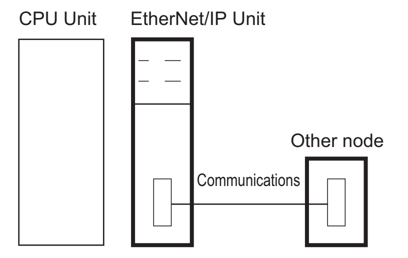
1. Execute a socket service request command to the EtherNet/IP unit using the SendCmd instruction.



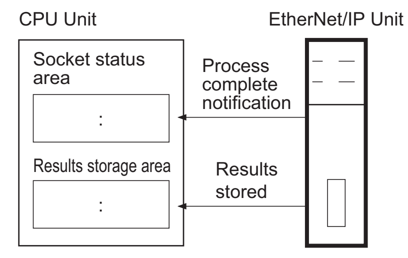
2. SendCmd instruction ends normally when the socket service request command is received and a response is returned.



3. The EtherNet/IP Unit starts the processing requested by the parameters in the socket service request.



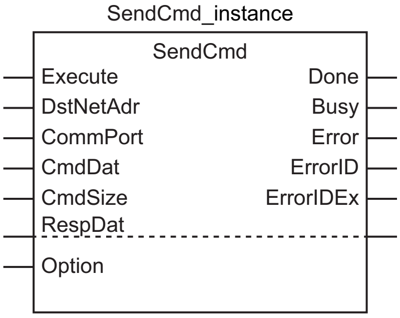
4. When the process has been completed, the result is stored in the results storage area defined in the socket service request and the socket status will indicate completion of processing.



## 13-7-2 Socket Service Request

The values to be set commonly for the input variables of the SendCmd instruction when a socket service is requested are shown in the table below. The settings of variables shown as “Depends on service” are described for each socket service request later in this section.

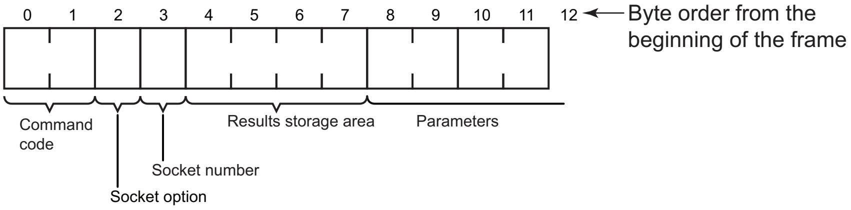
Refer to the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502) for details on each parameter.



| Variable | Meaning | Setting | Remarks |
|---|---|---|---|
| DstNetAdr | Destination network address | --- | |
| NetNo | Network address | 0 | This can be used as the default setting of the structure _sDNET_ADR. |
| NodeNo | Node address | 0 | This can be used as the default setting of the structure _sDNET_ADR. |
| UnitNo | Unit number | Specify the UNIT No. rotary switch setting + 16#10. | Example: When the UNIT No. switch is set to *A*, specify 16#1A. |
| CommPort | Destination serial port | _NONE | |
| ComdDat[] array | Command to send | Depends on service | |
| CmdSize | Command data size | Depends on service | |
| RespDat[] array | Response storage array | Depends on service | |
| Option | Response | --- | |
| isNonResp | No response | FALSE | This can be used as the default setting of the structure _sRESPONSE. |
| TimeOut | Timeout time | 20 | This can be used as the default setting of the structure _sRESPONSE. |
| Retry | Retry count | 0 | This can be used as the default setting of the structure _sRESPONSE. |

## Basic Command Format

The basic format of the command is shown in the following diagram.



### ● Command Code

Specifies the code for the requested processing.

### ● Socket Option

For the TCP OPEN REQUEST (ACTIVE or PASSIVE) command, specifies whether or not the keep-alive or linger function is to be used. For all other commands it is disabled. (Set to 0).

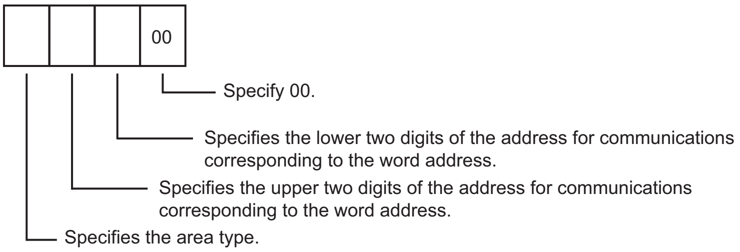
### ● Socket Number

Specifies the socket number for which the processing is requested, from 1 to 8.

### ● Results Storage Area

Specifies the area to store the results of the requested processing.

The four bytes of the results storage area are composed as follows.


00
Specify 00.
Specifies the lower two digits of the address for communications corresponding to the word address.
Specifies the upper two digits of the address for communications corresponding to the word address.
Specifies the area type.

For the area type and the address for communications corresponding to the word address, the specification method is the same as described in *Send/Receive Data Address* (on page 13-22).

### ● Parameters

Specifies the parameters defined for the command code. Refer to *13-7 Using Socket Services with SendCmd Instruction* for details.

## UDP OPEN REQUEST

Requests processing to open a UDP socket.

### ● Input Variables

The values to be set for the input variables of the SendCmd instruction are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| CmdDat[0] | Command code | The command code for a UDP socket open request is 16#2701. | BYTE#16#27 |
| CmdDat[1] | | | BYTE#16#01 |
| CmdDat[2] | Socket option | The socket option specified as 1 byte. The setting is not valid for this command. Set to 0. | BYTE#16#00 |
| CmdDat[3] | UDP socket number | The UDP socket number to be opened is specified as 1 byte between 1 and 8. | |
| CmdDat[4] | Results storage area | The area to store the results of service processing in the EtherNet/IP Unit is specified.<br>The contents stored in the Result Storage Area are described later in this section. | Refer to *Basic Command Format* (on page 13-35) for the specification method. |
| CmdDat[5] | | | |
| CmdDat[6] | | | |
| CmdDat[7] | | | |
| CmdDat[8] | Local UDP port number | The UDP port number for communications with the socket is specified as 2 bytes (0 cannot be specified). | |
| CmdDat[9] | | | |
| CmdSize | Command size | The valid data length of *CmdDat* is specified. | UINT#10 |

### ● In-Out Variables

There is no need to set values to in-out variables during execution of the SendCmd instruction.

The values to be set in in-out variables after completion of SendCmd instruction execution are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| RespDat[0] | Command code | The same value as that of CmdDat[0] will be set. | BYTE#16#27 |
| RespDat[1] | | The same value as that of CmdDat[1] will be set. | BYTE#16#01 |
| RespDat[2] | Response code | The response codes of the SendCmd instruction will be stored. | |
| RespDat[3] | | | |

The meanings of the response codes are shown in the following table.

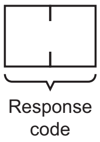
| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0105 | Local IP address setting error. |
| 0302 | CPU Unit error; cannot execute. |
| 1001 | Command too large |
| 1002 | Command too small |
| 1100 | UDP socket number is out of range.<br>Local UDP port number is 0. |
| 1101 | The variable type for the results storage area is out of range. |

| Response code | Meaning |
|---|---|
| 1103 | The bit number in the results storage area is not 00. |
| 220F | Specified socket is already open or already processing an close request. |
| 2211 | High traffic at Unit; cannot execute service |

### ● Results Storage Area

The results of service processing in the EtherNet/IP Unit are stored in the specified results storage area.

The data format is shown below.



The results storage area stores a response code.

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 003E | Internal buffer cannot be obtained due to high reception traffic (ENOBUFS). |
| 0049 | Duplicate UDP port number (EADDRINUSE). |

## UDP RECEIVE REQUEST

Requests processing to receive data via a UDP socket.

### ● Input Variables

The values to be set for the input variables of the SendCmd instruction are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| CmdDat[0] | Command code | The command code for a UDP socket receive request is 16#2702. | BYTE#16#27 |
| CmdDat[1] | | | BYTE#16#02 |
| CmdDat[2] | Socket option | The socket option specified as 1 byte. The setting is not valid for this command. Set to 0. | BYTE#16#00 |
| CmdDat[3] | UDP socket number | The UDP socket number to receive data is specified as 1 byte between 1 and 8. | |
| CmdDat[4] | Results storage area | The area to store the results of service processing in the EtherNet/IP Unit is specified.<br>The contents stored in the Result Storage Area are described later in this section. | Refer to *Basic Command Format* (on page 13-35) for the specification method. |
| CmdDat[5] | | | |
| CmdDat[6] | | | |
| CmdDat[7] | | | |
| CmdDat[8] | Number of reception bytes | The number of bytes of data to be received is specified. Up to 1,984 bytes can be specified.<br>The number of bytes received will be stored in the results storage area. | |
| CmdDat[9] | | | |

| Variable | Item | Description | Value |
|---|---|---|---|
| CmdDat[10] | Timeout time | The maximum control time between receiving the receive request and storing the result.<br>If this set time limit is exceeded, the code for a timeout error will be set as the results storage response code.<br>The value is set in units of 0.1 s. The timeout time will be unlimited if the value is set to 0. | |
| CmdDat[11] | | | |
| CmdSize | Command size | The valid data length of *CmdDat* is specified. | UINT#12 |

## ● In-Out Variables

There is no need to set values to in-out variables during execution of the SendCmd instruction.

The values to be set in in-out variables after completion of SendCmd instruction execution are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| RespDat[0] | Command code | The same value as that of CmdDat[0] will be set. | BYTE#16#27 |
| RespDat[1] | | The same value as that of CmdDat[1] will be set. | BYTE#16#02 |
| RespDat[2] | Response code | The response codes of the SendCmd instruction will be stored. | |
| RespDat[3] | | | |

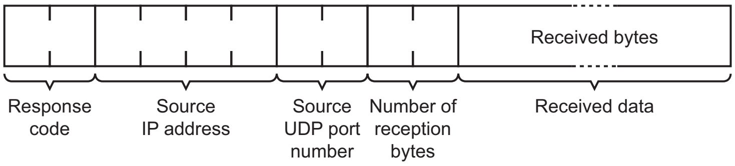
The meanings of the response codes are shown in the following table.

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0105 | Local IP address setting error. |
| 0302 | CPU Unit error; cannot execute. |
| 1001 | Command too large |
| 1002 | Command too small |
| 1100 | UDP socket number or number of reception bytes is out of range. |
| 1101 | The variable type for the results storage area is out of range. |
| 1103 | The bit number in the results storage area is not 00. |
| 220F | Specified socket is already processing a receive request. |
| 2210 | Specified socket is not open. |
| 2211 | High traffic at Unit; cannot execute service |
| 2240 | Mode is incorrect; cannot execute service. (The high-speed socket service option was enabled and a socket service was used with a SendCmd instruction.)<br>Or, the socket service was executed with the SendCmd instruction when the layout type of the allocated CIO Area words is set to *Default*. |

## ● Results Storage Area

The results of service processing in the EtherNet/IP Unit are stored in the specified results storage area.

The data format is shown below.



The result storage area stores the following information.

- Response code

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 003E | Internal buffer cannot be obtained due to high reception traffic (ENOBUFS). |
| 0066 | Internal memory cannot be obtained; cannot execute service. |
| 0080 | Receive request timed out. |
| 0081 | Specified socket was closed during reception processing. |

- Source IP address
  The source IP address is stored in hexadecimal.

- Source UDP port number
  The UDP port number of the source is stored.

- Number of reception bytes
  The number of bytes received is stored.

- Received data
  The received data is stored.

### Precautions for Correct Use

If a packet is received which contains more bytes than the number specified in *Number of reception bytes* specified in the command, the specified number of bytes will be stored and the remainder of the bytes will be discarded.

## UDP SEND REQUEST

Requests processing to send data via a socket.

### ● Input Variables

The values to be set for the input variables of the SendCmd instruction are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| CmdDat[0] | Command code | The command code for a UDP socket send request is 16#2703. | BYTE#16#27 |
| CmdDat[1] | | | BYTE#16#03 |
| CmdDat[2] | Socket option | The socket option specified as 1 byte. The setting is not valid for this command. Set to 0. | BYTE#16#00 |
| CmdDat[3] | UDP socket number | The UDP socket number to receive data is specified as 1 byte between 1 and 8. | |
| CmdDat[4] | Results storage area | The area to store the results of service processing in the EtherNet/IP Unit is specified.<br>The contents stored in the Result Storage Area are described later in this section. | Refer to *Basic Command Format* (on page 13-35) for the specification method. |
| CmdDat[5] | | | |
| CmdDat[6] | | | |
| CmdDat[7] | | | |
| CmdDat[8] | Destination IP address | The destination IP address is specified in hexadecimal. | |
| CmdDat[9] | | | |
| CmdDat[10] | | | |
| CmdDat[11] | | | |
| CmdDat[12] | Destination UDP port number | The destination UDP port number is specified. | |
| CmdDat[13] | | | |
| CmdDat[14] | Number of bytes to send | The number of bytes to send is specified. Up to 1,984 bytes can be specified, or up to 1,472 bytes can be specified if the broadcast address is specified as the send destination IP address.<br>The results storage area stores the number of bytes actually sent. | |
| CmdDat[15] | | | |
| CmdDat[16] to | Data to send | The data to send to the remote node is specified. | |
| CmdSize | Command size | The valid data length of *CmdDat* is specified. | UINT# (CmdDat array size) |

### ● In-Out Variables

There is no need to set values to in-out variables during execution of the SendCmd instruction.

The values to be set in in-out variables after completion of SendCmd instruction execution are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| RespDat[0] | Command code | The same value as that of CmdDat[0] will be set. | BYTE#16#27 |
| RespDat[1] | | The same value as that of CmdDat[1] will be set. | BYTE#16#03 |
| RespDat[2] | Response code | The response codes of the SendCmd instruction will be stored. | |
| RespDat[3] | | | |

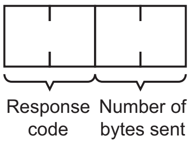
The meanings of the response codes are shown in the following table.

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0105 | Local IP address setting error. |
| 0302 | CPU Unit error; cannot execute. |
| 1001 | Command too large |
| 1002 | Command too small |
| 1003 | The number of bytes sent does not match the sent data length. |
| 1100 | UDP socket number or number of reception bytes is out of range.<br>The destination IP address is 0.<br>The destination UDP port number is 0. |
| 1101 | The variable type for the results storage area is out of range. |
| 1103 | The bit number in the results storage area is not 00. |
| 220F | Specified socket is already processing a send request. |
| 2210 | Specified socket is not open. |
| 2211 | High traffic at Unit; cannot execute service |
| 2240 | Mode is incorrect; cannot execute service. (The high-speed socket service option was enabled and a socket service was used with a SendCmd instruction.)<br>Or, the socket service was executed with the SendCmd instruction when the layout type of the allocated CIO Area words is set to *Default*. |

## ● Results Storage Area

The results of service processing in the EtherNet/IP Unit are stored in the specified results storage area.

The data format is shown below.



The result storage area stores the following information.

- Response code

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 003E | Internal buffer cannot be obtained due to high reception traffic (ENOBUFS). |
| 0042 | Destination IP address is a broadcast address and the number of bytes sent is greater than 1,472 bytes (EMSGSIZE). |
| 004C | Network ID is incorrect or the destination IP address is incorrect (EADDRNOTAVAIL). |
| 004E | Network ID is not in the IP router table, router settings are incorrect, or the destination IP address is incorrect (ENETUNREACH). |
| 0051 | Router settings are incorrect or the destination IP address is incorrect (EHOSTUNREACH). |

- Number of bytes sent
  The number of bytes actually sent is stored.

## UDP CLOSE REQUEST

Requests processing to close a UDP socket.

### ● Input Variables

The values to be set for the input variables of the SendCmd instruction are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| CmdDat[0] | Command code | The command code for a UDP socket close request is 16#2704. | BYTE#16#27 |
| CmdDat[1] | | | BYTE#16#04 |
| CmdDat[2] | Socket option | The socket option specified as 1 byte. The setting is not valid for this command. Set to 0. | BYTE#16#00 |
| CmdDat[3] | UDP socket number | The UDP socket number to receive data is specified as 1 byte between 1 and 8. | |
| CmdDat[4] | Results storage area | The area to store the results of service processing in the EtherNet/IP Unit is specified.<br>The contents stored in the Result Storage Area are described later in this section. | Refer to *Basic Command Format* (on page 13-35) for the specification method. |
| CmdDat[5] | | | |
| CmdDat[6] | | | |
| CmdDat[7] | | | |
| CmdSize | Command size | The valid data length of *CmdDat* is specified. | UINT#8 |

### ● In-Out Variables

There is no need to set values to in-out variables during execution of the SendCmd instruction.

The values to be set in in-out variables after completion of SendCmd instruction execution are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| RespDat[0] | Command code | The same value as that of CmdDat[0] will be set. | BYTE#16#27 |
| RespDat[1] | | The same value as that of CmdDat[1] will be set. | BYTE#16#04 |
| RespDat[2] | Response code | The response codes of the SendCmd instruction will be stored. | |
| RespDat[3] | | | |

The meanings of the response codes are shown in the following table.

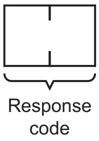
| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0105 | Local IP address setting error. |
| 0302 | CPU Unit error; cannot execute. |
| 1001 | Command too large |
| 1002 | Command too small |
| 1100 | UDP socket number is out of range. |
| 1101 | The variable type for the results storage area is out of range. |
| 1103 | The bit number in the results storage area is not 00. |

| Response code | Meaning |
|---|---|
| 2210 | Specified socket is not open. |
| 2211 | High traffic at Unit; cannot execute service |
| 2240 | Mode is incorrect; cannot execute service. (The high-speed socket service option was enabled and a socket service was used with a SendCmd instruction.)<br>Or, the socket service was executed with the SendCmd instruction when the layout type of the allocated CIO Area words is set to *Default*. |

### ● Results Storage Area

The results of service processing in the EtherNet/IP Unit are stored in the specified results storage area.

The data format is shown below.



The results storage area stores a response code.

| Response code | Meaning |
|---|---|
| 0000 | Normal end |

## PASSIVE TCP OPEN REQUEST

Requests processing to open a TCP socket. The socket will wait to be connected to another node.

### ● Input Variables

The values to be set for the input variables of the SendCmd instruction are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| CmdDat[0] | Command code | The command code for a TCP socket passive open request is 16#2710. | BYTE#16#27 |
| CmdDat[1] | | | BYTE#16#10 |
| CmdDat[2] | Socket option | The socket option specified as 1 byte.<br>The specification details are described later in this section. | |
| CmdDat[3] | UDP socket number | The UDP socket number to receive data is specified as 1 byte between 1 and 8. | |
| CmdDat[4] | Results storage area | The area to store the results of service processing in the EtherNet/IP Unit is specified.<br>The contents stored in the Result Storage Area are described later in this section. | Refer to *Basic Command Format* (on page 13-35) for the specification method. |
| CmdDat[5] | | | |
| CmdDat[6] | | | |
| CmdDat[7] | | | |
| CmdDat[8] | Local TCP port number | The TCP port number for communications with the socket is specified as 2 bytes (0 cannot be specified). TCP port No. 21 for FTP server cannot be specified. | |
| CmdDat[9] | | | |

| Variable | Item | Description | Value |
|---|---|---|---|
| CmdDat[10] | Timeout time | The maximum control time between receiving the open request and storing the result.<br>If this set time limit is exceeded, the code for a timeout error will be set as the results storage response code.<br>The value is set in units of 0.1 s. The timeout time will be unlimited if the value is set to 0. | |
| CmdDat[11] | | | |
| CmdDat[12] | Remote IP address | The IP address of the remote node is specified in hexadecimal.<br>• If all zeros are set, no remote node is specified and connection is awaited from any node.<br>• If any other value is set, connection is awaited from the specified remote node. | |
| CmdDat[13] | | | |
| CmdDat[14] | | | |
| CmdDat[15] | | | |
| CmdDat[16] | Remote TCP port number | The TCP port number of the remote node is specified.<br>• If all zeros are set, no remote node TCP port number is specified.<br>• If any other value is set, the TCP port number of the remote node is specified. | |
| CmdDat[17] | | | |
| CmdSize | Command size | The valid data length of *CmdDat* is specified. | UINT#18 |

## ● In-Out Variables

There is no need to set values to in-out variables during execution of the SendCmd instruction.

The values to be set in in-out variables after completion of SendCmd instruction execution are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| RespDat[0] | Command code | The same value as that of CmdDat[0] will be set. | BYTE#16#27 |
| RespDat[1] | | The same value as that of CmdDat[1] will be set. | BYTE#16#10 |
| RespDat[2] | Response code | The response codes of the SendCmd instruction will be stored. | |
| RespDat[3] | | | |

The meanings of the response codes are shown in the following table.

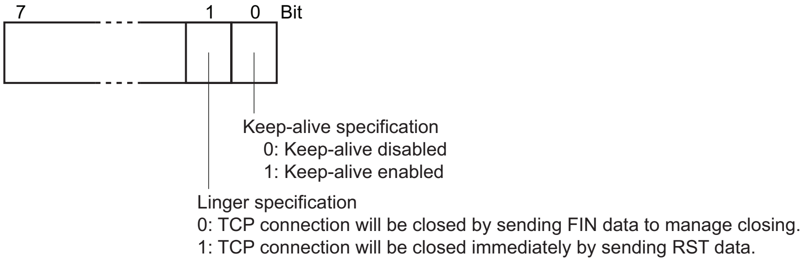
| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0105 | Local IP address setting error. |
| 0302 | CPU Unit error; cannot execute. |
| 1001 | Command too large |
| 1002 | Command too small |
| 1100 | TCP socket number is out of range.<br>Local TCP port number is 0. |
| 1101 | The variable type for the results storage area is out of range. |
| 1103 | The bit number in the results storage area is not 00. |

| Response code | Meaning |
|---|---|
| 220F | The specified socket (connection) is already open or is currently being opened or closed. |
| 2211 | High traffic at Unit; cannot execute service |
| 2240 | Mode is incorrect; cannot execute service. (The high-speed socket service option was enabled and a socket service was used with a SendCmd instruction.)<br>Or, the socket service was executed with the SendCmd instruction when the layout type of the allocated CIO Area words is set to *Default*. |

## ● Socket Option

The socket option specified as 1 byte.

The data format is shown below.



7
1
0
Bit
Keep-alive specification
0: Keep-alive disabled
1: Keep-alive enabled
Linger specification
0: TCP connection will be closed by sending FIN data to manage closing.
1: TCP connection will be closed immediately by sending RST data.

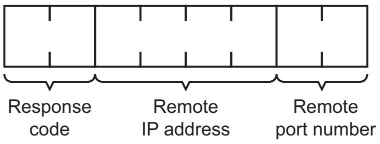
### Additional Information

- If the linger option is not specified and a TCP connection is closed, FIN data will be sent and then approximately 1 minute will be used to confirm the transmission and perform other closing management with the remote node. Therefore, it may not be possible to immediately use TCP sockets with the same port number.
- In contrast, when the linger option is specified, RST data will be sent when TCP is closed, and closing will be performed immediately. This enables immediately opening TCP sockets with the same port number. Data that was sent immediately before closing, however, is not checked for transmission to the remote node. If the linger option is specified, ensure the arrival of the send data in the application.

## ● Results Storage Area

The results of service processing in the EtherNet/IP Unit are stored in the specified results storage area.

The data format is shown below.



The result storage area stores the following information.

- Response code

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 003E | Internal buffer cannot be obtained due to high reception traffic (ENOBUFS). |
| 0042 [*1] | Error occurred. (EMSGSIZE). |
| 0045 | Error in communications with remote node (ECONNABORTED). |
| 0049 | Duplicate TCP port number (EADDRINUSE). |
| 004A [*1] | Error occurred. (ECONNREFUSED). |
| 004B [*1] | Error in communications with remote node (ECONNRESET). |
| 004E [*1] | Remote IP address parameter error (ENETUNREACH). |
| 0051 [*1] | Remote IP address parameter error (EHOSTUNREACH). |
| 0053 | Error in communications with remote node (ETIMEDOUT) or remote node does not exist. |
| 0066 | Internal memory cannot be obtained; cannot execute service. |
| 0080 | Open request timed out. |
| 0081 | Specified socket was closed during open processing. |
| 0082 | Connection could not be established with specified remote node. |

[*1] These response codes will be returned only on large, multilevel networks.

- Remote IP address
  The IP address of the connected remote node is stored.

- Remote TCP port number
  The TCP port number of the connected remote node is stored.

### Precautions for Correct Use

Processing varies change as shown in the table below according to the specified combination of remote IP address and remote TCP port number.

| Remote IP address | Remote TCP port number | Processing |
|---|---|---|
| 0 | 0 | All connection requests accepted. |
| 0 | Not 0 | Connection requests accepted only for the same port number. |
| Not 0 | 0 | Connection requests accepted only for the same IP address. |
| Not 0 | Not 0 | Connection requests accepted only for the same port number and IP address. |

## ACTIVE TCP OPEN REQUEST

Requests processing to open a TCP socket. The socket will be connected to another node.

### ● Input Variables

The values to be set for the input variables of the SendCmd instruction are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| CmdDat[0] | Command code | The command code for a TCP socket active open request is 16#2711. | BYTE#16#27 |
| CmdDat[1] | | | BYTE#16#11 |
| CmdDat[2] | Socket option | The socket option specified as 1 byte.<br>The specification details are described later in this section. | |
| CmdDat[3] | UDP socket number | The UDP socket number to receive data is specified as 1 byte between 1 and 8. | |
| CmdDat[4] | Results storage area | The area to store the results of service processing in the EtherNet/IP Unit is specified.<br>The contents stored in the Result Storage Area are described later in this section. | Refer to *Basic Command Format* (on page 13-35) for the specification method. |
| CmdDat[5] | | | |
| CmdDat[6] | | | |
| CmdDat[7] | | | |
| CmdDat[8] | Local TCP port number | The TCP port number for communications with the socket is specified as 2 bytes. TCP port No. 21 for FTP server cannot be specified. An available TCP port number is automatically assigned if 0 is specified. | |
| CmdDat[9] | | | |
| CmdDat[10] | Remote IP address | The IP address of the remote node is specified (must be non-zero) in hexadecimal. | |
| CmdDat[11] | | | |
| CmdDat[12] | | | |
| CmdDat[13] | | | |
| CmdDat[14] | Remote TCP port number | The TCP port number of the remote node is specified (must be non-zero). | |
| CmdDat[15] | | | |
| CmdSize | Command size | The valid data length of *CmdDat* is specified. | UINT#16 |

### ● In-Out Variables

There is no need to set values to in-out variables during execution of the SendCmd instruction.

The values to be set in in-out variables after completion of SendCmd instruction execution are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| RespDat[0] | Command code | The same value as that of CmdDat[0] will be set. | BYTE#16#27 |
| RespDat[1] | | The same value as that of CmdDat[1] will be set. | BYTE#16#11 |
| RespDat[2] | Response code | The response codes of the SendCmd instruction will be stored. | |
| RespDat[3] | | | |

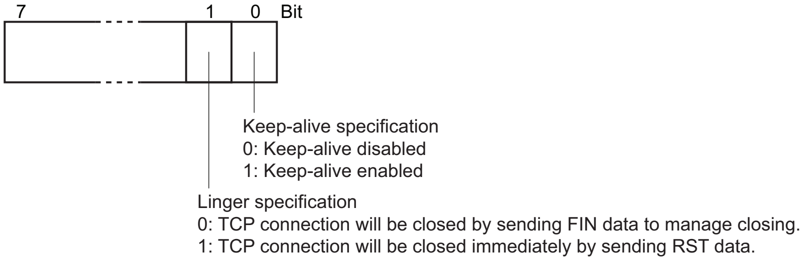
The meanings of the response codes are shown in the following table.

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0105 | Local IP address setting error. |
| 0302 | CPU Unit error; cannot execute. |
| 1001 | Command too large |
| 1002 | Command too small |
| 1100 | TCP socket number is out of range.<br>Remote IP address or the remote TCP port number is 0. |
| 1101 | The variable type for the results storage area is out of range. |
| 1103 | The bit number in the results storage area is not 00. |
| 220F | The specified socket (connection) is already open or is currently being opened or closed. |
| 2211 | High traffic at Unit; cannot execute service |
| 2240 | Mode is incorrect; cannot execute service. (The high-speed socket service option was enabled and a socket service was used with a SendCmd instruction.)<br>Or, the socket service was executed with the SendCmd instruction when the layout type of the allocated CIO Area words is set to *Default*. |

## ● Socket Option

The socket option specified as 1 byte.

The data format is shown below.


7
1
0
Bit
Keep-alive specification
0: Keep-alive disabled
1: Keep-alive enabled
Linger specification
0: TCP connection will be closed by sending FIN data to manage closing.
1: TCP connection will be closed immediately by sending RST data.

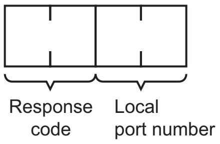
### Additional Information

- If the linger option is not specified and a TCP connection is closed, FIN data will be sent and then approximately 1 minute will be used to confirm the transmission and perform other closing management with the remote node. Therefore, it may not be possible to immediately use TCP sockets with the same port number.
- In contrast, when the linger option is specified, RST data will be sent when TCP is closed, and closing will be performed immediately. This enables immediately opening TCP sockets with the same port number. Data that was sent immediately before closing, however, is not checked for transmission to the remote node. If the linger option is specified, ensure the arrival of the send data in the application.

### ● Results Storage Area

The results of service processing in the EtherNet/IP Unit are stored in the specified results storage area.

The data format is shown below.



The result storage area stores the following information.

- Response code

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 000D | Remote IP address parameter error (EACCES). |
| 003E | Internal buffer cannot be obtained due to high reception traffic (ENOBUFS). |
| 0042 *1 | Error occurred. (EMSGSIZE). |
| 0044 | ICMP data received (ENOPROTOOPT). |
| 0045 | Local socket was closed (ECONNABORTED). |
| 0049 | Duplicate TCP port number (EADDRINUSE). |
| 004A | Error (ECONNREFUSED) or the remote node has not been opened as passive socket. |
| 004B *1 | Error in communications with remote node (ECONNRESET). |
| 004C | Remote IP address parameter error (EADDRNOTAVAIL).<br>Wrong parameter designation.<br>Attempt was made to set the local TCP port of the local node to Active Open. |
| 004E | Remote IP address parameter error (ENETUNREACH).<br>The network ID is not in the IP router table or router settings are incorrect. |
| 0051 | Remote IP address parameter error (EHOSTUNREACH).<br>Router settings are incorrect. |
| 0053 | Error in communications with remote node (ETIMEDOUT) or remote node does not exist. |
| 0081 | Specified socket was closed during open processing. |

*1 These response codes will be returned only on large, multilevel networks.

- Local TCP port number
  The TCP port numbers allocated to the open socket are stored in the results storage area.

## TCP RECEIVE REQUEST

Requests processing to receive data via a TCP socket.

### ● Input Variables

The values to be set for the input variables of the SendCmd instruction are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| CmdDat[0] | Command code | The command code for a TCP socket receive request is 16#2712. | BYTE#16#27 |
| CmdDat[1] | | | BYTE#16#12 |
| CmdDat[2] | Socket option | The socket option specified as 1 byte. The setting is not valid for this command. Set to 0. | BYTE#16#00 |
| CmdDat[3] | UDP socket number | The UDP socket number to receive data is specified as 1 byte between 1 and 8. | |
| CmdDat[4] | Results storage area | The area to store the results of service processing in the EtherNet/IP Unit is specified.<br>The contents stored in the Result Storage Area are described later in this section. | Refer to *Basic Command Format* (on page 13-35) for the specification method. |
| CmdDat[5] | | | |
| CmdDat[6] | | | |
| CmdDat[7] | | | |
| CmdDat[8] | Number of reception bytes | The number of bytes of data to be received is specified. Up to 1,984 bytes can be specified.<br>The number of bytes received will be stored in the results storage area. | |
| CmdDat[9] | | | |
| CmdDat[10] | Timeout time | The maximum control time between receiving the receive request and storing the result.<br>If this set time limit is exceeded, the code for a timeout error will be set as the response code in the results storage area. The value is set in units of 0.1 s. The timeout time will be unlimited if the value is set to 0. | |
| CmdDat[11] | | | |
| CmdSize | Command size | The valid data length of *CmdDat* is specified. | UINT#12 |

### ● In-Out Variables

There is no need to set values to in-out variables during execution of the SendCmd instruction.

The values to be set in in-out variables after completion of SendCmd instruction execution are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| RespDat[0] | Command code | The same value as that of CmdDat[0] will be set. | BYTE#16#27 |
| RespDat[1] | | The same value as that of CmdDat[1] will be set. | BYTE#16#12 |
| RespDat[2] | Response code | The response codes of the SendCmd instruction will be stored. | |
| RespDat[3] | | | |

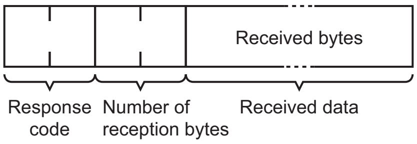
The meanings of the response codes are shown in the following table.

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0105 | Local IP address setting error. |
| 0302 | CPU Unit error; cannot execute. |
| 1001 | Command too large |
| 1002 | Command too small |
| 1100 | TCP socket number or number of reception bytes is out of range. |
| 1101 | The variable type for the results storage area is out of range. |
| 1103 | The bit number in the results storage area is not 00. |
| 220F | Specified socket is already processing a receive request. |
| 2210 | Specified socket has not been connected. |
| 2211 | High traffic at Unit; cannot execute service |
| 2240 | Mode is incorrect; cannot execute service. (The high-speed socket service option was enabled and a socket service was used with a SendCmd instruction.)<br>Or, the socket service was executed with the SendCmd instruction when the layout type of the allocated CIO Area words is set to *Default*. |

## ● Results Storage Area

The results of service processing in the EtherNet/IP Unit are stored in the specified results storage area.

The data format is shown below.



The result storage area stores the following information.

- Response code

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 003E | Internal buffer cannot be obtained due to high reception traffic (ENOBUFS). |
| 0042 [*1] | ICMP data received (EMSGSIZE). |
| 0044 [*1] | ICMP data received (ENOPROTOOPT). |
| 0045 [*1] | Error in communications with remote node (ECONNABORTED). |
| 004B | Error in communications with remote node (ECONNRESET). |
| 004E [*1] | Remote IP address parameter error (ENETUNREACH).<br>Network ID is not in the IP router table or router settings are incorrect. |
| 004F [*1] | ICMP data received (EHOSTDOWN). |
| 0051 [*1] | ICMP data received (EHOSTUNREACH).<br>Router settings are incorrect. |
| 0053 | Error in communications with remote node (ETIMEDOUT). |
| 0066 | Internal memory cannot be obtained; cannot execute service. |
| 0080 | Receive request timed out. |
| 0081 | Specified socket was closed during reception processing. |

[*1] These response codes will be returned only on large, multilevel networks.

- Number of reception bytes
  The number of bytes received is stored.

- Received data
  The received data is stored.

## TCP SEND REQUEST

Requests processing to send data via a TCP socket.

### ● Input Variables

The values to be set for the input variables of the SendCmd instruction are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| CmdDat[0] | Command code | The command code for a TCP socket receive request is 16#2713. | BYTE#16#27 |
| CmdDat[1] | | | BYTE#16#13 |
| CmdDat[2] | Socket option | The socket option specified as 1 byte. The setting is not valid for this command. Set to 0. | BYTE#16#00 |
| CmdDat[3] | UDP socket number | The UDP socket number to receive data is specified as 1 byte between 1 and 8. | |
| CmdDat[4] | Results storage area | The area to store the results of service processing in the EtherNet/IP Unit is specified.<br>The contents stored in the Result Storage Area are described later in this section. | Refer to *Basic Command Format* (on page 13-35) for the specification method. |
| CmdDat[5] | | | |
| CmdDat[6] | | | |
| CmdDat[7] | | | |

| Variable | Item | Description | Value |
|---|---|---|---|
| CmdDat[8] | Number of bytes to send | The number of bytes to send is specified. Up to 1,984 bytes can be specified. | |
| CmdDat[9] | | | |
| CmdDat[10] to | Data to send | The data to send is specified. | |
| CmdSize | Command size | The valid data length of *CmdDat* is specified. | UINT# (CmdDat array size) |

### ● In-Out Variables

There is no need to set values to in-out variables during execution of the SendCmd instruction.

The values to be set in in-out variables after completion of SendCmd instruction execution are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| RespDat[0] | Command code | The same value as that of CmdDat[0] will be set. | BYTE#16#27 |
| RespDat[1] | | The same value as that of CmdDat[1] will be set. | BYTE#16#13 |
| RespDat[2] | Response code | The response codes of the SendCmd instruction will be stored. | |
| RespDat[3] | | | |

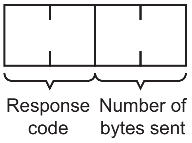
The meanings of the response codes are shown in the following table.

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0105 | Local IP address setting error. |
| 0302 | CPU Unit error; cannot execute. |
| 1001 | Command too large |
| 1002 | Command too small |
| 1003 | The number of bytes sent does not match the amount of data. |
| 1100 | TCP socket number or number of reception bytes is out of range. |
| 1101 | The variable type for the results storage area is out of range. |
| 1103 | The bit number in the results storage area is not 00. |
| 220F | Specified socket is already processing a send request. |
| 2210 | Specified socket has not been connected. |
| 2211 | High traffic at Unit; cannot execute service |
| 2240 | Mode is incorrect; cannot execute service. (The high-speed socket service option was enabled and a socket service was used with a SendCmd instruction.)<br>Or, the socket service was executed with the SendCmd instruction when the layout type of the allocated CIO Area words is set to *Default*. |

### Results Storage Area

The results of service processing in the EtherNet/IP Unit are stored in the specified results storage area.

The data format is shown below.



The result storage area stores the following information.

- Response code

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0020 | Connection with remote socket broken during send (EPIPE). |
| 003E | Internal buffer cannot be obtained due to high reception traffic (ENOBUFS). |
| 0042 [*1] | Error occurred. (EMSGSIZE). |
| 0044 [*1] | ICMP data received (ENOPROTOOPT). |
| 0045 [*1] | Error in communications with remote node (ECONNABORTED). |
| 004B [*1] | Error in communications with remote node (ECONNRESET). |
| 004E [*1] | Remote IP address parameter error (ENETUNREACH). |
| 004F [*1] | ICMP data received (EHOSTDOWN). |
| 0051 [*1] | Remote IP address parameter error (EHOSTUNREACH). |
| 0053 [*1] | Error in communications with remote node (ETIMEDOUT). |
| 0081 | Specified socket was closed during send processing. |

[*1] These response codes will be returned only on large, multilevel networks.

- Number of bytes sent
  The number of bytes received is stored.

## TCP CLOSE REQUEST

Requests processing to close a TDP socket. Other processing being carried out is forcibly ended and a code is recorded in the results storage area.

### Input Variables

The values to be set for the input variables of the SendCmd instruction are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| CmdDat[0] | Command code | The command code for a TCP socket receive request is 16#2714. | BYTE#16#27 |
| CmdDat[1] | | | BYTE#16#14 |
| CmdDat[2] | Socket option | The socket option specified as 1 byte. The setting is not valid for this command. Set to 0. | BYTE#16#00 |
| CmdDat[3] | UDP socket number | The UDP socket number to receive data is specified as 1 byte between 1 and 8. | |

| Variable | Item | Description | Value |
|---|---|---|---|
| CmdDat[4] | Results storage area | The area to store the results of service processing in the EtherNet/IP Unit is specified.<br>The contents stored in the Result Storage Area are described later in this section. | Refer to *Basic Command Format* (on page 13-35) for the specification method. |
| CmdDat[5] | | | |
| CmdDat[6] | | | |
| CmdDat[7] | | | |
| CmdSize | Command size | The valid data length of *CmdDat* is specified. | UINT#8 |

### ● In-Out Variables

There is no need to set values to in-out variables during execution of the SendCmd instruction.

The values to be set in in-out variables after completion of SendCmd instruction execution are shown in the following table.

| Variable | Item | Description | Value |
|---|---|---|---|
| RespDat[0] | Command code | The same value as that of CmdDat[0] will be set. | BYTE#16#27 |
| RespDat[1] | | The same value as that of CmdDat[1] will be set. | BYTE#16#14 |
| RespDat[2] | Response code | The response codes of the SendCmd instruction will be stored. | |
| RespDat[3] | | | |

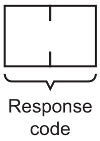
The meanings of the response codes are shown in the following table.

| Response code | Meaning |
|---|---|
| 0000 | Normal end |
| 0105 | Local IP address setting error. |
| 0302 | CPU Unit error; cannot execute. |
| 1001 | Command too large |
| 1002 | Command too small |
| 1100 | TCP socket number is out of range. |
| 1101 | The variable type for the results storage area is out of range. |
| 1103 | The bit number in the results storage area is not 00. |
| 220F | Specified socket is already processing a send request. |
| 2210 | Specified socket has not been connected. |
| 2211 | High traffic at Unit; cannot execute service |
| 2240 | Mode is incorrect; cannot execute service. (The high-speed socket service option was enabled and a socket service was used with a SendCmd instruction.)<br>Or, the socket service was executed with the SendCmd instruction when the layout type of the allocated CIO Area words is set to *Default*. |

### Results Storage Area

The results of service processing in the EtherNet/IP Unit are stored in the specified results storage area.

The data format is shown below.



The results storage area stores a response code.

- Response code

| Response code | Meaning |
|---|---|
| 0000 | Normal end |

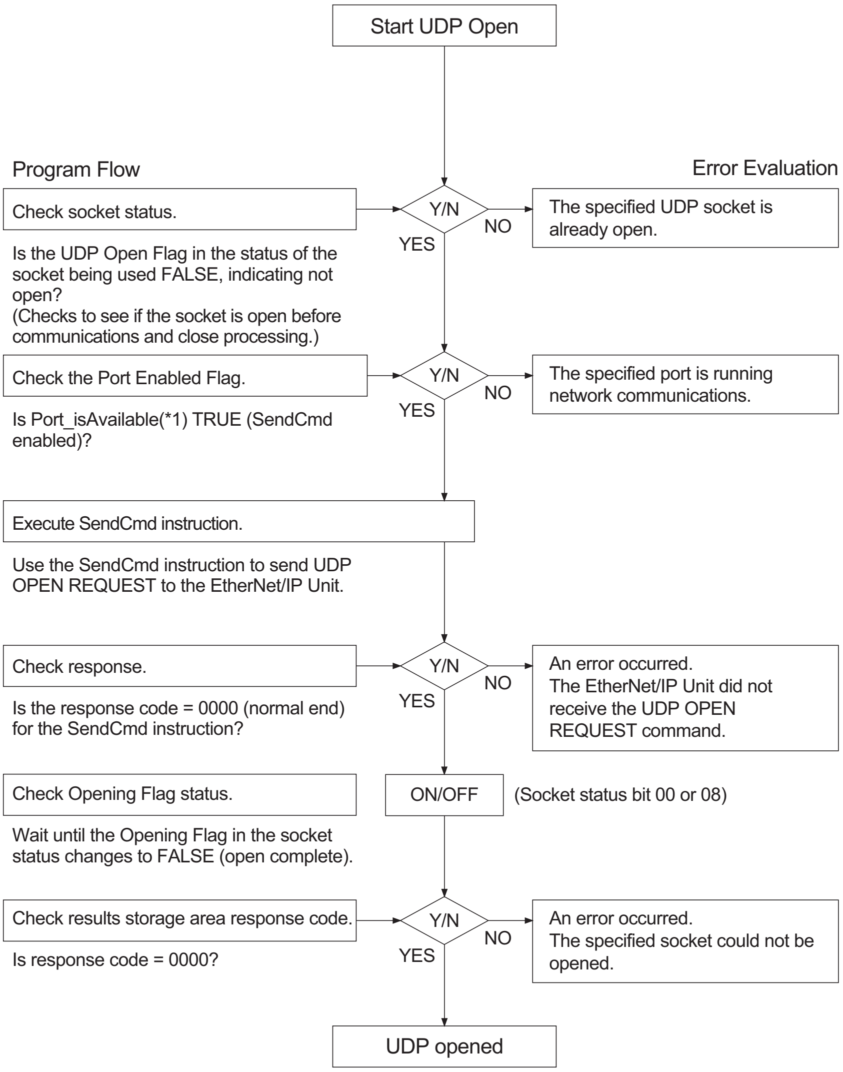
**Precautions for Correct Use**

Any other processing, such as sending or receiving data, being carried out when this close command is executed will be forcibly ended and a code will be stored in the results storage area to indicate that this processing was forcibly ended.

## 13-7-3 Using Socket Services and Socket Status

When using socket services, it is important to consider the timing of the status changes in the socket status. The diagram below shows a flowchart for opening UDP.

The flow is similar for other socket services. Replace the names of the appropriate flags in the flowchart to adapt it to other socket services.


Start UDP Open
Program Flow
Error Evaluation
Check socket status.
Y/N
NO
YES
The specified UDP socket is already open.
Is the UDP Open Flag in the status of the socket being used FALSE, indicating not open?
(Checks to see if the socket is open before communications and close processing.)
Check the Port Enabled Flag.
Y/N
NO
YES
The specified port is running network communications.
Is Port_isAvailable(*1) TRUE (SendCmd enabled)?
Execute SendCmd instruction.
Use the SendCmd instruction to send UDP OPEN REQUEST to the EtherNet/IP Unit.
Check response.
Y/N
NO
YES
An error occurred.
The EtherNet/IP Unit did not receive the UDP OPEN REQUEST command.
Is the response code = 0000 (normal end) for the SendCmd instruction?
Check Opening Flag status.
ON/OFF
(Socket status bit 00 or 08)
Wait until the Opening Flag in the socket status changes to FALSE (open complete).
Check results storage area response code.
Y/N
NO
YES
An error occurred.
The specified socket could not be opened.
Is response code = 0000?
UDP opened

(*1) _Port_isAvailable is a system-defined variable for the Controller to determine whether or not SendCmd is enabled.
Refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) for details.

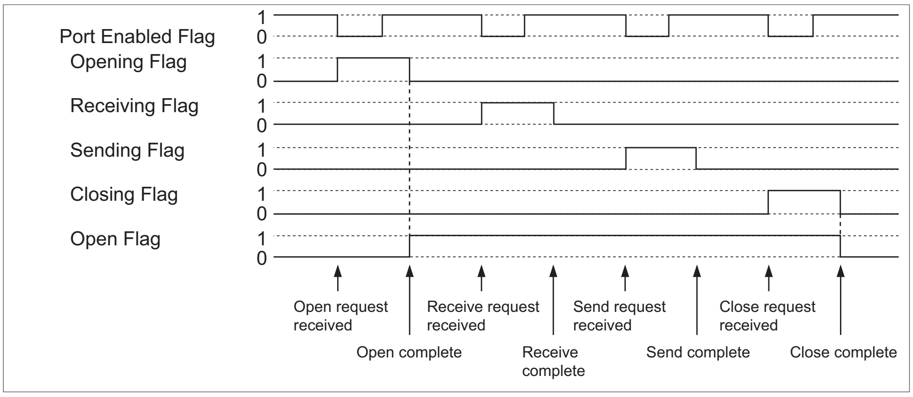
## 13-7-4 Response Codes

The execution result of the SendCmd instruction is stored in an output variable (Done, Busy, Error, ErrorID, or ErrorIDEx) for the SendCmd instruction. Refer to the *NJ/NX-series Instructions Reference Manual* (Cat. No. W502) for details.

When the SendCmd instruction ends normally, the EtherNet/IP unit processes the socket service request and stores the processing result as a response code in the results storage area. Refer to *13-7-2 Socket Service Request* for details.

## 13-7-5 Communications Timing Chart

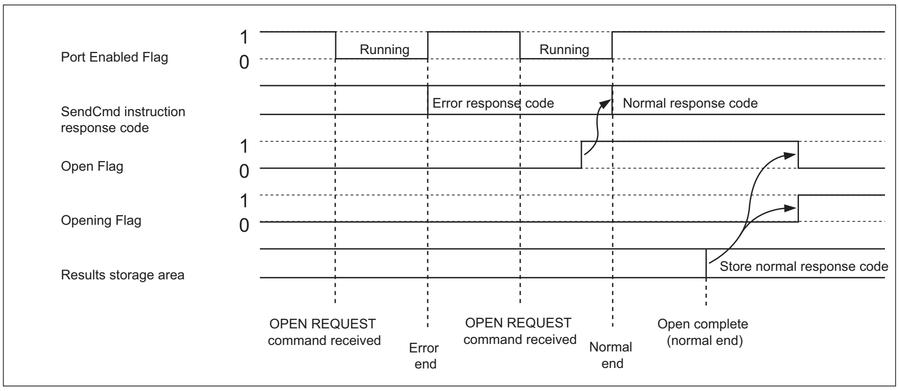
The timing of the status changes of the bits in the socket status and the Port Enabled Flag is shown in the following diagram.



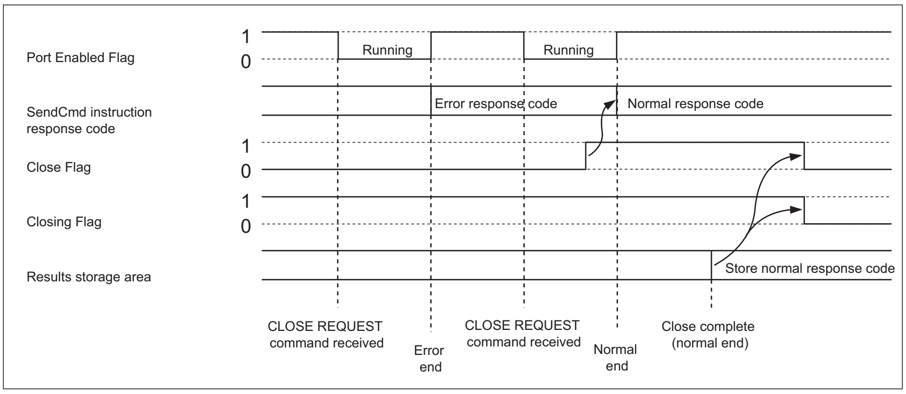
## 13-7-6 Socket Service Timing Chart

The timing of the socket service open, send, receive, and close request instructions are shown in the following diagrams.

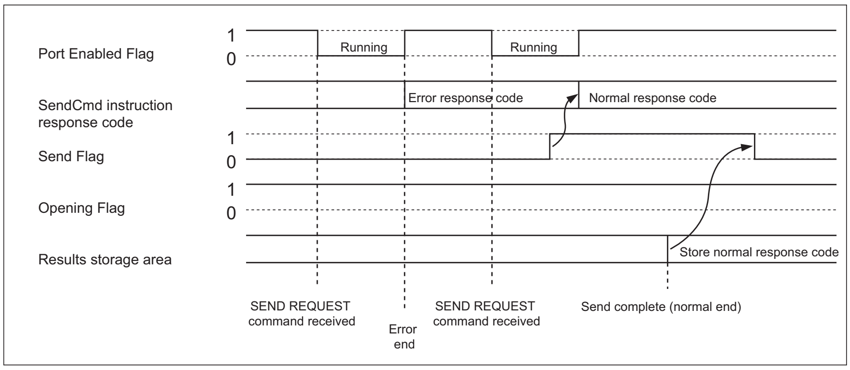
### ● OPEN REQUEST


Port Enabled Flag
SendCmd instruction response code
Open Flag
Opening Flag
Results storage area
1
0
Running
Running
Error response code
Normal response code
Store normal response code
OPEN REQUEST command received
Error end
OPEN REQUEST command received
Normal end
Open complete (normal end)

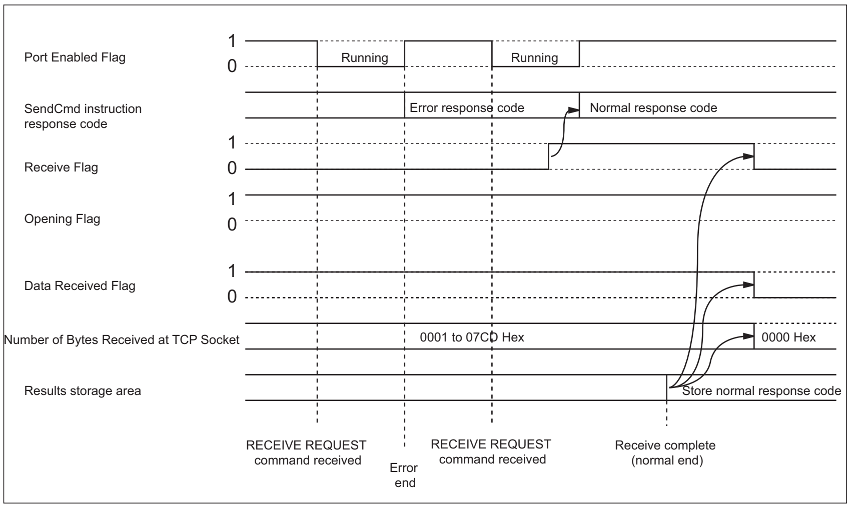
### ● CLOSE REQUEST


Port Enabled Flag
SendCmd instruction response code
Close Flag
Closing Flag
Results storage area
1
0
Running
Running
Error response code
Normal response code
Store normal response code
CLOSE REQUEST command received
Error end
CLOSE REQUEST command received
Normal end
Close complete (normal end)

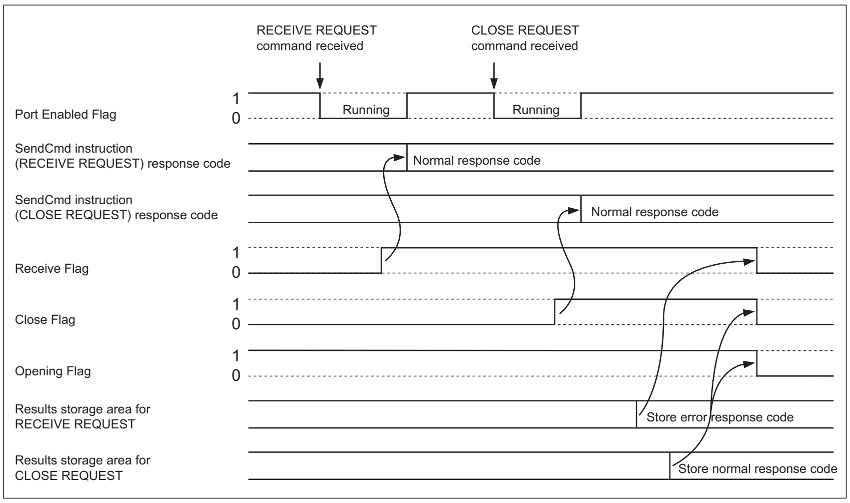
## ● SEND REQUEST


Port Enabled Flag
1
0
Running
Running
SendCmd instruction response code
Error response code
Normal response code
Send Flag
1
0
Opening Flag
1
0
Results storage area
Store normal response code
SEND REQUEST command received
Error end
SEND REQUEST command received
Send complete (normal end)

## ● RECEIVE REQUEST


Port Enabled Flag
1
0
Running
Running
SendCmd instruction response code
Error response code
Normal response code
Receive Flag
1
0
Opening Flag
1
0
Data Received Flag
1
0
Number of Bytes Received at TCP Socket
0001 to 07CD Hex
0000 Hex
Results storage area
Store normal response code
RECEIVE REQUEST command received
Error end
RECEIVE REQUEST command received
Receive complete (normal end)

## ● CLOSE REQUEST during RECEIVE REQUEST


RECEIVE REQUEST command received
CLOSE REQUEST command received
Port Enabled Flag
1
0
Running
Running
SendCmd instruction (RECEIVE REQUEST) response code
Normal response code
SendCmd instruction (CLOSE REQUEST) response code
Normal response code
Receive Flag
1
0
Close Flag
1
0
Opening Flag
1
0
Results storage area for RECEIVE REQUEST
Store error response code
Results storage area for CLOSE REQUEST
Store normal response code

### Additional Information

- The timing shown in the above diagram occurs if a CLOSE REQUEST instruction is executed during SEND REQUEST instruction execution.
- The timing shown in the diagram also applies if a CLOSE REQUEST instruction is executed during OPEN REQUEST instruction execution, with the exception of the status of the Opening Flag.

## 13-7-7 Sample Programming

Refer to *A-9 Sample Programming for Socket Services* for sample programming for socket services by SendCmd instruction.

# 13-8 Considerations in Using Socket Services

## 13-8-1 Considerations Common to UDP/TCP Socket Services

- If a short response monitor time is specified in SendCmd instruction control data and the EtherNet/IP Unit is operating under a high load, a result may be stored even if the response code indicates a time-out.
  If this occurs, increase the monitor time specified with SendCmd instruction.
- The socket status area in the CIO Area is zeroed when the Controller's operating mode is changed (e.g., from PROGRAM to RUN).
  The actual EtherNet/IP Unit socket status, however, will remain unchanged after the socket status area is zeroed.
  To avoid this problem, use the IOM Hold setting in the Controller Setup. Refer to the Controller's operation manuals for details on settings.
- The Results Storage Error Flag will change to TRUE in the socket status to indicate that the specified Results Storage Area does not exist in the Controller. Correct the user program.
- Communications time may increase if multiple EtherNet/IP Unit functions are used simultaneously or due to the contents of the user program.
- Communications efficiency may decrease due to high communications loads on the network.
- All data is flushed from the socket's communications buffer when a socket is closed with the CLOSE REQUEST command.
  In some cases, the transmit data for the SEND REQUEST command issued just before the socket was closed may not be sent.
- When sockets are open, the EtherNet/IP Unit provides a 4,096-byte buffer for each TCP socket and 9,016-byte buffer for each UDP socket to allow data to be received at any time.
  Receive data will be discarded for a socket if the buffer becomes full. The user application must therefore issue RECEIVE REQUEST commands frequently enough to prevent the internal buffers from becoming full.

## 13-8-2 Considerations for UDP Socket Service Only

- The UDP socket sets a broadcast address for the destination node address to broadcast data to all nodes of the network simultaneously.
  The maximum length of broadcast data is 1,472 bytes.
  Data in multiple fragments (over 1,473 bytes for a UDP socket) cannot be broadcast.
- The UDP socket does not check the transmitted data to ensure communications reliability. To increase communication reliability, communications checks and retries must be included in the user application program.

## 13-8-3 Considerations for TCP Socket Service Only

- If the TCP socket of the remote node closes (the connection is broken) during communications, the TCP socket at the local node must also be closed.
  The communications Results Storage Area can be used to check if the connection has been broken. Close the local socket immediately after detecting that the remote TCP socket has closed.

| | |
|---|---|
| TCP Receive Results Storage Area: | Response code = 004B (error at remote node) |
| TCP Send Results Storage Area: | Response code = 0020 (connection broken with remote socket during transmission) |

- Data can remain in a buffer at the local node if the remote TCP socket closes during communications.
  Any data remaining in the buffer will be discarded when the TCP socket is closed.
  To avoid problems of this nature, steps will have to be taken in the application program, such as sending data to enable closing, and then only closing once reception of this data has been confirmed.
- When closing a connection for a TCP socket, the first port to be closed cannot be reopened for at least 60 seconds after the other port closes.
  However, this restriction does not apply for a port opened using the TCP ACTIVE OPEN REQUEST command with a local TCP port number of 0 (port number automatically assigned) which is closed from the side that actively opened the socket.
- A connection is established for a passively opened socket by actively opening it from another socket.
  A connection will not be established by a different socket attempting to actively open the socket that is already actively opening a socket.
  Similarly, a connection will not be established if a different socket attempts to passively open a socket that is already being passively opened by another socket.
  You cannot actively open multiple connections to a socket passively opened at the EtherNet/IP Unit.
- The EtherNet/IP Unit TCP sockets have keep-alive function to check that the connection is normal if communications do not occur for a set time period through a communications line for which a connection has been established.
  Checks made by the remote node, however, are received as responses, so that it is not necessary for the user program to consider the keep-alive function.

## 13-8-4 Considerations for Manipulating Device Variables

- Send and reception processing cannot be performed at the same time when Socket Service Request Switches are used for socket services because there is only one Socket Service Parameter Area for each socket.
- For example, if the Send Request Switch is changed to TRUE when data is being received, the response code will be 110C hex, indicating that a Request Switch changed to TRUE during communications processing.
  (The response code for the reception will overwrite this code when processing has been completed.)
- If more than one Request Switch is changed to TRUE simultaneously, the response code will be 110C hex and all requested processing will end in an error.
- Close processing can be performed, however, even during open, send, or receive processing. This enables emergency close processing. Also, the only parameter required for close processing is the socket number, so a socket can be closed even when parameters are set for another process.

## 13-8-5 Times Required for Sending and Receiving for Socket Services

The transmission delays for socket service is calculated as the sum of the communications processing times for both nodes.

$$\text{Transmission delay} = \left(\begin{array}{c}\text{Remote node send}\\ \text{processing time}\end{array}\right) + \left(\begin{array}{c}\text{Local node receive}\\ \text{processing time}\end{array}\right) + \left(\begin{array}{c}\text{Local node send}\\ \text{processing time}\end{array}\right) + \left(\begin{array}{c}\text{Remote node receive}\\ \text{processing time}\end{array}\right)$$

Calculate the maximum EtherNet/IP Unit transmission delays for sending and receiving using the following formulas.

The delays found using the following formulas, however, are approximate values when one socket service is used. If multiple socket services are used, the delays will increase depending on the operating conditions. Also, the transmission delay on the network relative to the processing time is so small that it can be ignored, and so it is omitted.

### Requesting UDP Socket Services by Manipulating Device Variables

- High-speed Socket Services Enabled and Task Period of Less Than 1.85 ms
  Transmission processing time = reception processing time =
  task period × 5 + number of send/receive bytes × 0.0002 + 1.45 (ms)
- High-speed Socket Services Enabled and Task Period of 1.85 ms or Greater
  Transmission processing time = reception processing time =
  Task period × 6 (ms)
- High-speed Socket Services Disabled
  Transmission processing time = reception processing time =
  Task period × 7 (ms)

### Requesting TCP Socket Services by Manipulating Device Variables

- High-speed Socket Services Enabled and Task Period of Less Than 1.85 ms
  Transmission processing time = reception processing time =
  task period × 5 + number of send/receive bytes × 0.0002 + 1.45 (ms)
- High-speed Socket Services Enabled and Task Period of 1.85 ms or Greater
  Transmission processing time = reception processing time =
  Task period × 6 (ms)
- High-speed Socket Services Disabled
  Transmission processing time = reception processing time =
  Task period × 7 (ms)

### Requesting UDP Socket Services by Executing SendCmd Instruction

- Task Period of Less Than 20 ms
  Transmission processing time = reception processing time =
  Task period × 6 + 70.0 (ms)
- Task Period of 20 ms or Greater
  Transmission processing time = reception processing time =
  Task period × 6 (ms)

## Requesting TCP Socket Services by Executing SendCmd Instruction

- Task Period of Less Than 20 ms
  Transmission processing time = reception processing time =
  Task period × 6 + 70.0 (ms)
- Task Period of 20 ms or Greater
  Transmission processing time = reception processing time =
  Task period × 6 (ms)

**Additional Information**

- The values obtained from the above equations are guidelines for the transmission delay time when one socket in the EtherNet/IP Unit is used only.
  The execution time required for the user program is not included.
- The communications time for the remote nodes depends on the device being used.
  For remote nodes that are not EtherNet/IP Units, calculate the communications time according to the device's operation manual.
- The actual operating environment can cause transmission delays larger than those calculated with the methods given here.
  Among the causes of longer delays are the following: traffic on the network, window sizes at network nodes, traffic through the EtherNet/IP Unit (e.g., simultaneous servicing of multiple sockets and socket service communications, etc.), and the system configuration.
- The above values are guidelines when the default (4%) for the uniform peripheral servicing time in the Controller Setup is used.
- By increasing the value of the uniform peripheral servicing time, the maximum transmission delay time for socket services can be shorter.
- Processing cannot be faster than the send and receive processing performance of the EtherNet/IP Unit if the send request is processed periodically using a ladder program timer or if receive request processing is performed for continuous data from a remote node. In particular, the data buffer on the receiving side may be exhausted. In such a case, adjust the send timing (i.e., send timing from remote node) or receive frequency so that the actual load is approximately 1.5 times slower than the processing performance.

Example: When using TCP socket services between two Controllers by manipulating device variables (high-speed socket service enabled) to send/receive 512 bytes in both directions, the guideline for the maximum transmission delay time can be calculated according to the following conditions as shown in the table below.

Task period (local node) = 1 ms

Task period (remote node ) = 4 ms

| Item | Calculation |
|---|---|
| Reception processing time (local node) | 1 × 5 + 512 × 0.0002 + 1.45 = 6.552 ms ≈ 6.6 ms |
| Transmission processing time (local node) | 1 × 5 + 512 × 0.0002 + 1.45 = 6.552 ms ≈ 6.6 ms |
| Transmission processing time (remote node) | 4 × 6 = 24.0 ms |
| Reception processing time (remote node) | 4 × 6 = 24.0 ms |
| Maximum transmission delay | 6.6 + 6.6 + 24.0 + 24.0 = 61.2 ms |

# 14

# Communications Performance and Communications Load

# 14-1 Communications System

## 14-1-1 Tag Data Link Communications Method

### Requested Packet Interval (RPI) Settings

In tag data links for the EtherNet/IP Unit, 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:**

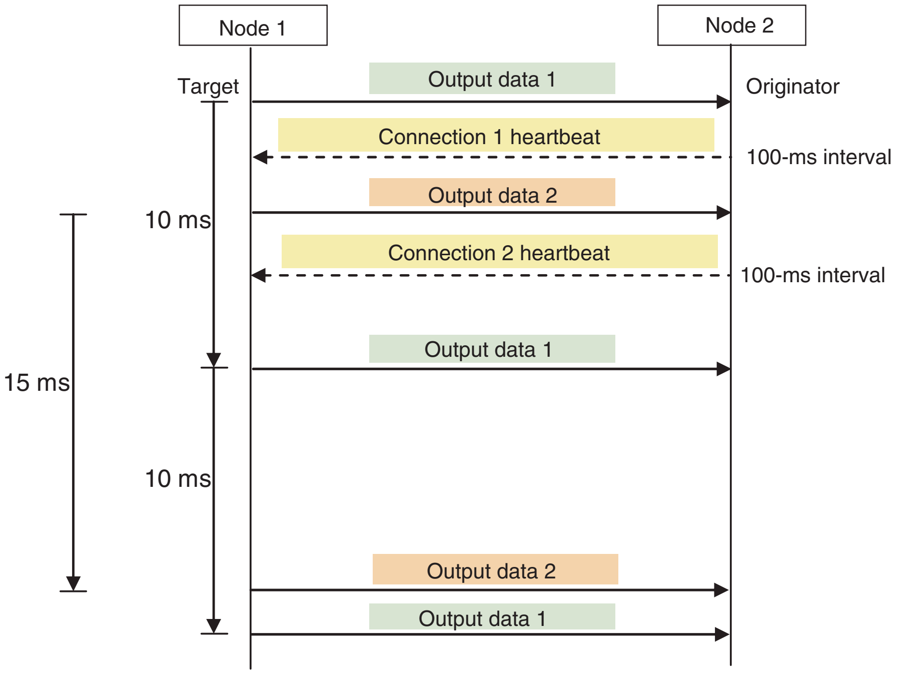
In this example, 2 tag data link connections are set for node 2 (the originator) and node 1 (the target).

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

The weighted number of packets transferred each second is called the bandwidth usage. "N" in this section represents a weighing factor according to the packet data size.

The bandwidth usage is calculated from the RPI, heartbeat, and the factor of N as follows for each connection:

Bandwidth used in a connection = (1,000 ÷ RPI (ms) × N) + (1,000 ÷ Heartbeat transmission period (ms))

N = Tag data link's allowable bandwidth ÷ (Tag data link's allowable bandwidth + Coefficient × Data size per connection)

| Unit version | Allowed tag data link communications bandwidth | Coefficient | N |
|---|---|---|---|
| 2.1 or earlier | 6,000 | 0 | 1 |
| 3.0 | 12,000 | -4.155 | 1 to 2 |

Use the following equation to calculate the total bandwidth used by each Unit (refers to as an EtherNet/IP Unit in the following examples).

Total bandwidth used by Unit = Total bandwidth used by originator connections + Total bandwidth used by target connections

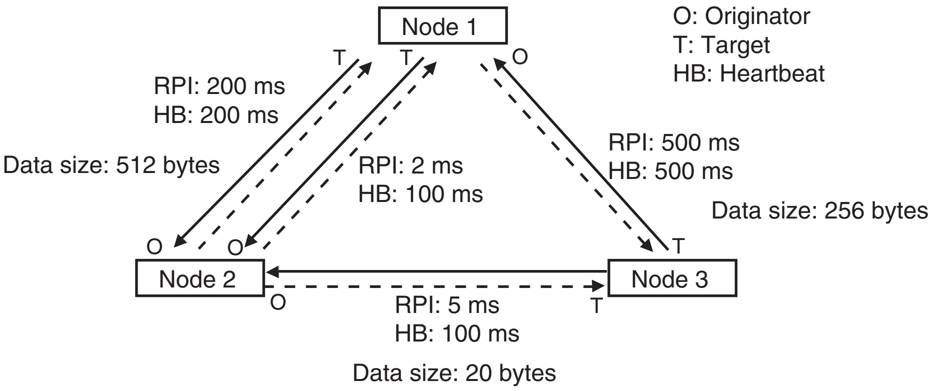
**Note** Connections set as target connections must also be added to the total bandwidth used by target connections.

Make the connection settings so that the Unit's total bandwidth used does not exceed its upper value.

| Unit version | Total bandwidth used |
|---|---|
| 2.1 or earlier | 6,000 |
| 3.0 | 12,000 |

Example

Node 1 has both originator and target connections, and sends 512 bytes of data at an RPI of 200 ms and 10 bytes of data at an RPI of 2 ms, and receives 256 bytes of data at an RPI of 500 ms.
Node 2 has originator connections only, and receives 512 bytes of data at an RPI of 200 ms, 10 bytes of data at an RPI of 2 ms, and 20 bytes of data at an RPI of 5 ms.
Node 3 has target connections only, and sends 20 bytes of data at an RPI of 5 ms, and 256 bytes of data at an RPI of 500 ms.



Each node's total bandwidth used is calculated as follows:

### ● Communication using Units with Unit version 2.1 or earlier

- Total bandwidth used for node 1 Unit
  = 1,000 / 200 ms × 1 + 1,000 / 2 ms × 1 + 1,000 / 500 ms × 1 (for data)
  + 1,000 / 200 ms + 1,000 / 100 ms + 1,000 / 500 ms (for heartbeat)
  = 524
- Total bandwidth used for node 2 Unit
  = 1,000 / 200 ms × 1 + 1,000 / 2 ms × 1 + 1,000 / 5 ms × 1 (for data)
  + 1,000 / 200 ms + 1,000 / 100 ms + 1,000 / 100 ms (for heartbeat)
  = 730
- Total bandwidth used for node 3 Unit
  = 1,000 / 5 ms × 1 + 1,000 / 500 ms × 1 (for data)
  + 1,000 / 100 ms + 1,000 / 500 ms (for heartbeat)
  = 214

All of the Units are within the upper value of the total bandwidth used of 6,000 pps, so they can transfer data.

### ● Communication using Units with Unit version 3.0

| Data size (bytes) | Factor N |
|---|---|
| 10 | 12,000 / (12,000 - 4.155 × 10) = 1.003 |
| 20 | 12,000 / (12,000 - 4.155 × 20) = 1.007 |
| 256 | 12,000 / (12,000 - 4.155 × 256) = 1.097 |
| 512 | 12,000 / (12,000 - 4.155 × 512) = 1.215 |

- Total bandwidth used for node 1 Unit
  = 1,000 / 200 ms × 1.215 + 1,000 / 2 ms × 1.003 + 1,000 / 500 ms × 1.097 (for data)
  + 1,000 / 200 ms + 1,000 / 100 ms + 1,000 / 500 ms (for heartbeat)
  = 527
- Total bandwidth used for node 2 Unit
  = 1,000 / 200 ms × 1.215 + 1,000 / 2 ms × 1.003 + 1,000 / 5 ms × 1.007 (for data)
  + 1,000 / 200 ms + 1,000 / 100 ms + 1,000 / 100 ms (for heartbeat)
  = 734
- Total bandwidth used for node 3 Unit
  = 1,000 / 5 ms × 1.007 + 1,000 / 500 ms × 1.097 (for data)
  + 1,000 / 100 ms + 1,000 / 500 ms (for heartbeat)
  = 216

All of the Units are within the tag data link's allowable bandwidth of 12,000 pps, so they can transfer data.

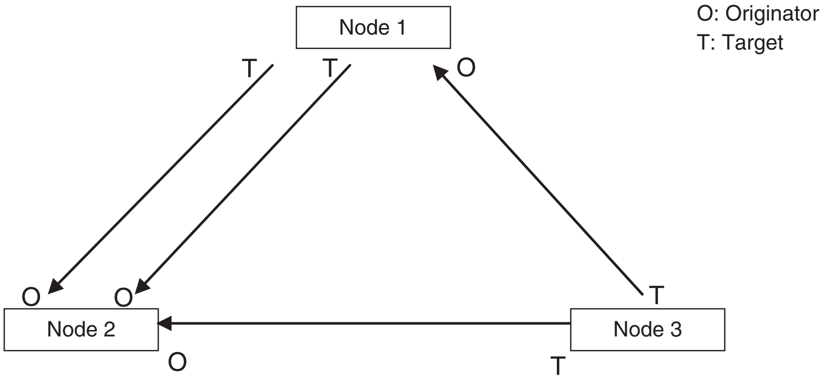
## 14-1-2 Calculating the Number of Connections

The maximum number of connections for the EtherNet/IP Unit is 256.

The number of connections must be set to 256 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 3. Therefore, the total is three 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 EtherNet/IP Unit is 256 max.

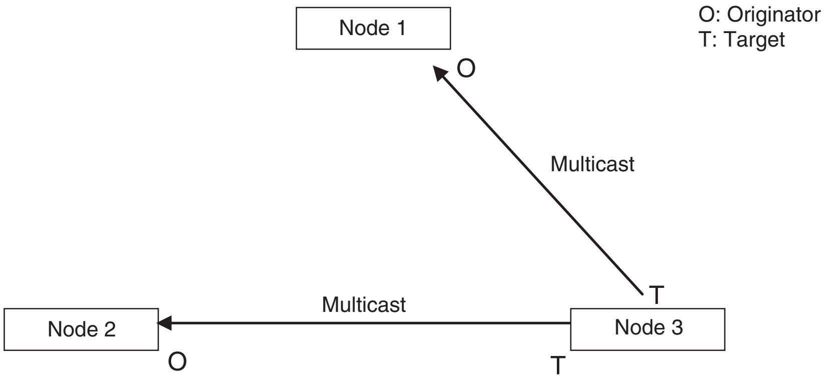


Also, if multicast is set, one packet will be sent, but the number of connections will be consumed.

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.

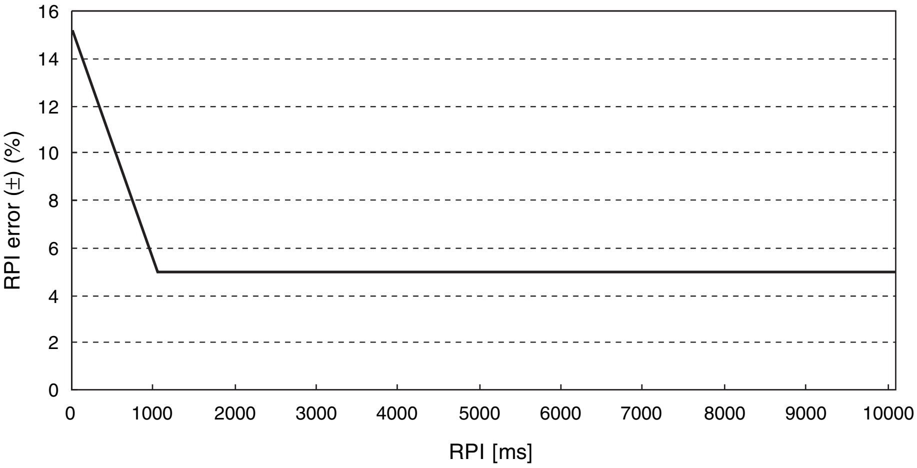
Caution is required because the number of connections consumed is the same as for unicast connections even when multicast connections are set.



## 14-1-3 Packet Interval (RPI) Accuracy

The send processing delay occurs in the EtherNet/IP Unit 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 (±) (%) |
|---|---|
| 0.5 to 1,000 ms | 15 − (RPI [ms]/100) |
| 1,000 to 10,000 ms | 5% of the RPI |



# 14-2 Adjusting the Communications Load

14-2 Adjusting the Communications Load
14

In an Ethernet network using an Ethernet switch, the network bandwidth is not shared by all of the nodes; independent transmission paths are established between individual nodes through the Ethernet switch.

A dedicated communications buffer is established in the Ethernet switch for communications between the nodes and full-duplex communications (simultaneous transmission and reception) are performed asynchronously with other transmission paths. The communications load in other transmission paths does not affect communications, so packet collisions do not occur and stable, high-speed communications can be performed.

The Ethernet switch functions shown in the following table determine the performance of tag data links.

| Item | Description |
|---|---|
| Buffer capacity | This is the amount of data that can be buffered when packets accumulate at the Ethernet switch. |
| Multicast filtering | This function transfers multicast packets to specific nodes only. |
| QoS function | This function performs priority control on packet transfers. |

The following table shows the setting ranges of the tag data link settings that can be made for an EtherNet/IP Unit.

| 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) | 6,000 to 1,2000 pps |
| Connection resources | Number of connections that can be established | 256 max. |
| Packet interval (RPI: Requested Packet Interval) | Refresh period for tag data | 0.5 to 10,000 ms in 0.5-ms increments |

When the tag data link settings exceed the capabilities of the Ethernet switch being used, increase the packet interval (RPI) value. Particularly when using an Ethernet switch that does not support multicast filtering, the settings must be made considering that multicast packets will be sent even to nodes without connection settings.

**Additional Information**

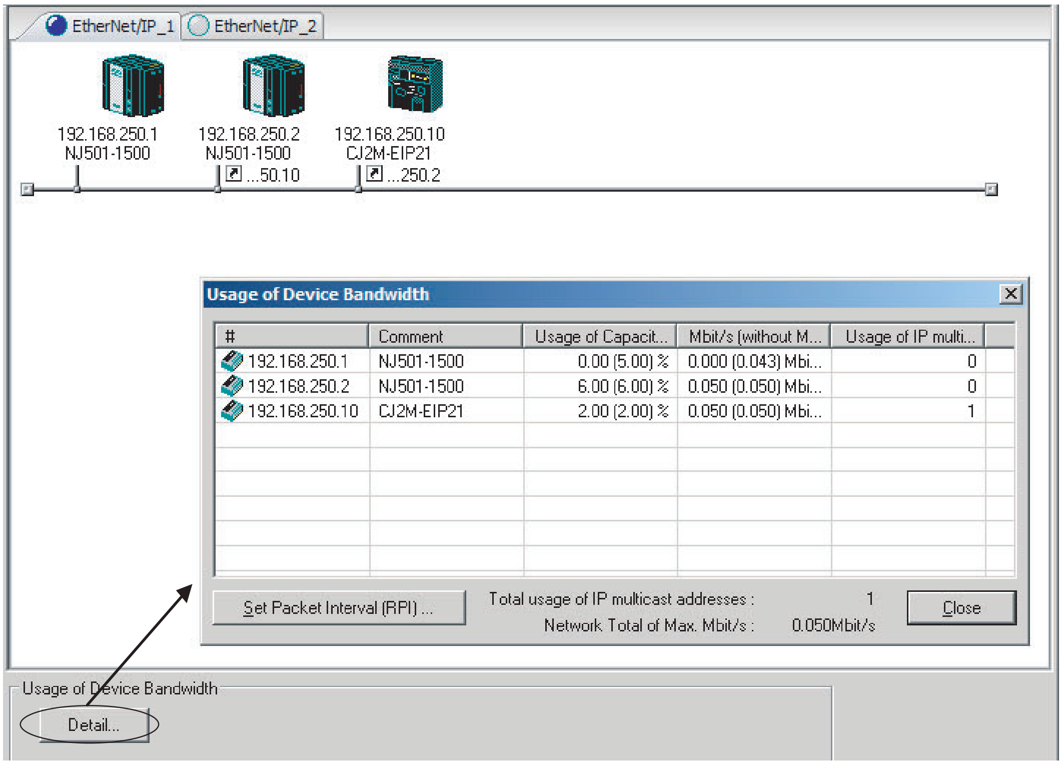
If the Network Configurator is used to set the connection type in the connection settings to a multicast connection, multicast packets will be used. If the connection type is set to a point-to-point connection, multicast packets are not used.

In addition, if the required tag data link performance cannot be achieved with the Ethernet switch's capabilities, re-evaluate the overall network configuration and correct it by taking steps such as selecting a different Ethernet switch or splitting the network.

The following sections show how to check the device bandwidth being used by the tag data links in the designed network, and how to set the appropriate values.

## 14-2-1 Checking Bandwidth Usage for Tag Data Links

The Network Configurator can display the bandwidth actually used for tag data links at each EtherNet/IP Unit, 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.



| # | Comment | Usage of Capacit... | Mbit/s (without M... | Usage of IP multi... |
|---|---|---|---|---|
| 192.168.250.1 | NJ501-1500 | 0.00 (5.00) % | 0.000 (0.043) Mbi... | 0 |
| 192.168.250.2 | NJ501-1500 | 6.00 (6.00) % | 0.050 (0.050) Mbi... | 0 |
| 192.168.250.10 | CJ2M-EIP21 | 2.00 (2.00) % | 0.050 (0.050) Mbi... | 1 |

Total usage of IP multicast addresses : 1

Network Total of Max. Mbit/s : 0.050Mbit/s

| 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<br>The values outside parentheses are for when multicast filtering is used.<br>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.<br>The values outside parentheses are for when multicast filtering is used.<br>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 EtherNet/IP Unit is displayed as the *Usage of Capacity* and the bandwidth used for tag data link communications in the entire network is displayed as the *Mbit/s*. The usage of capacity and used network bandwidth that are displayed in parentheses are for an Ethernet switch that does not use multicast filtering. In this case, multicast packets will be sent to even the nodes without connection settings, so the displayed values will include these packets as well. These values can be adjusted according to instructions in *14-2-4 Changing the RPI*.

### ● Checking the Total Number of Multicast IP Addresses in the Network

When using an Ethernet switch that provides multicast filtering, there must be enough multicast filters for the network being used. The number of multicast IP address used in the entire network that is displayed by the Network Configurator is based on connection settings. Make sure that the number of multicast IP addresses used in the entire network does not exceed the number of multicast filters supported by the Ethernet switch. If necessary, change to an Ethernet switch with enough multicast filters, or adjust the usage of capacity and network bandwidth for tag data links (*Mbit/s*) values given for an Ethernet switch without multicast filtering (i.e., the values in parentheses). These values can be adjusted according to instructions in *14-2-4 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 *14-2-4 Changing the RPI*.

## 14-2-2 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.

## 14-2-3 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., an Ethernet switch or media converter) may be exceeded as the result of how the network was wired (e.g., cascade connections of Ethernet switches), causing a tag data link to operate abnormally. Check the bandwidth of the transmission path for all cascade connections. If the bandwidth is exceeded, rewire the network or increase the bandwidth between Ethernet switches (e.g., to 1 Gbps). If these countermeasures are not possible, change the connection settings, e.g., the RPI settings, and adjust the bandwidth for all cascade connections until the total network bandwidth is not exceeded.

### ● Ethernet Switches with Multicast Filtering (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., an Ethernet switch or media converter) may be exceeded as the result of how the network was wired (e.g., cascade connections of Ethernet switches), causing a tag data link to operate abnormally. Check the bandwidth of the transmission path for all cascade connections. If the bandwidth is exceeded, rewire the network or increase the bandwidth between Ethernet switches (e.g., to 1 Gbps). If these countermeasures are not possible, change the connection settings, e.g., the RPI settings, and adjust the bandwidth for all cascade connections until the total network bandwidth is not exceeded.
- Is the network bandwidth usage without multicast filtering under 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.

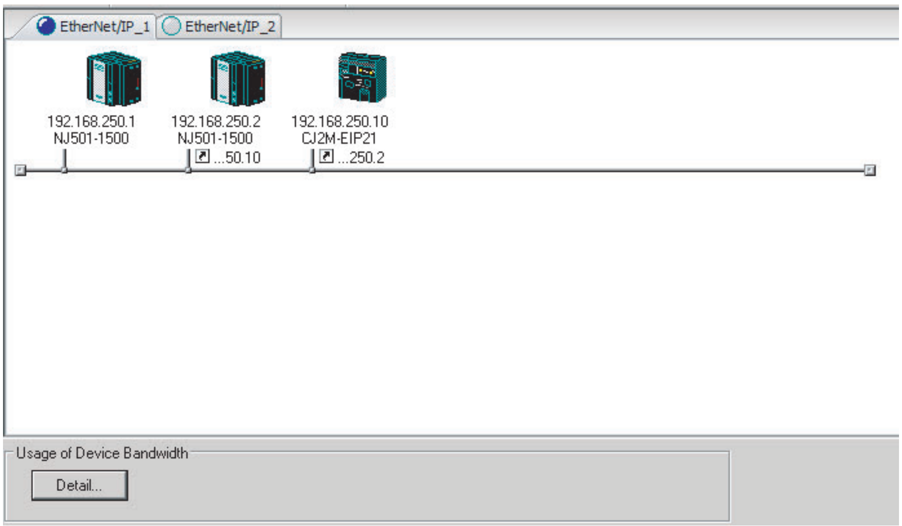
## 14-2-4 Changing the RPI

You can check the usage of capacity offline without multicast filtering against the tag data link's allowable bandwidth by following the procedures in *14-2-1 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).

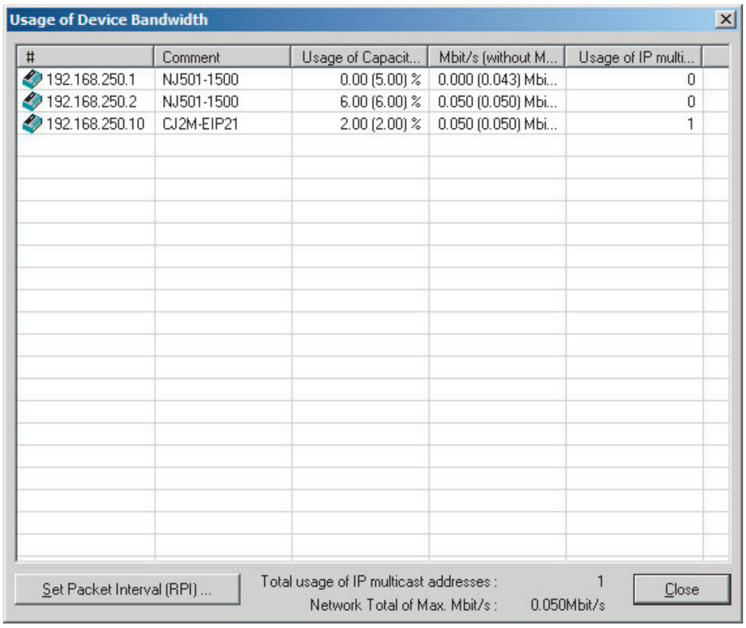
If the required communications performance cannot be achieved by changing the settings, re-evaluate the network starting with the network configuration.

**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.



| # | Comment | Usage of Capacit... | Mbit/s (without M... | Usage of IP multi... |
|---|---|---|---|---|
| 192.168.250.1 | NJ501-1500 | 0.00 (5.00) % | 0.000 (0.043) Mbi... | 0 |
| 192.168.250.2 | NJ501-1500 | 6.00 (6.00) % | 0.050 (0.050) Mbi... | 0 |
| 192.168.250.10 | CJ2M-EIP21 | 2.00 (2.00) % | 0.050 (0.050) Mbi... | 1 |

Total usage of IP multicast addresses : 1
Network Total of Max. Mbit/s : 0.050Mbit/s

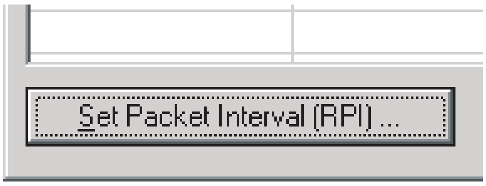
The *Usage of Capacity* (without multicast filter) column will show the percentage of the allowed tag data link bandwidth being used, and the *Mbit/s* (without multicast filter) column will show the network bandwidth being used.

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

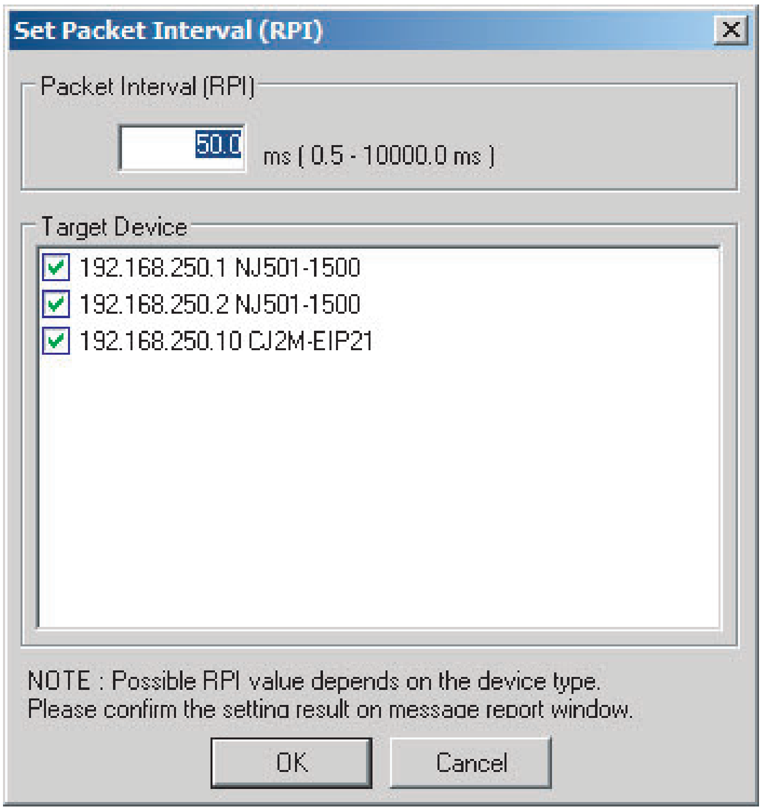
The RPI settings can be changed with the following three methods.

Method 1:
Change All Connections to the Same RPI
The usage of capacity without a multicast filter can be adjusted for all devices by changing the packet intervals (RPI) settings for all of the device’s connections to the same RPI at the same time.

**(1) Click the Set Packet Interval (RPI) Button at the bottom of the Usage of Device Bandwidth Dialog Box.**



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


Set Packet Interval (RPI)

Packet Interval (RPI)

50.0 ms ( 0.5 - 10000.0 ms )

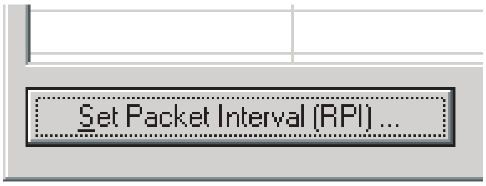
Target Device

☑ 192.168.250.1 NJ501-1500
☑ 192.168.250.2 NJ501-1500
☑ 192.168.250.10 CJ2M-EIP21

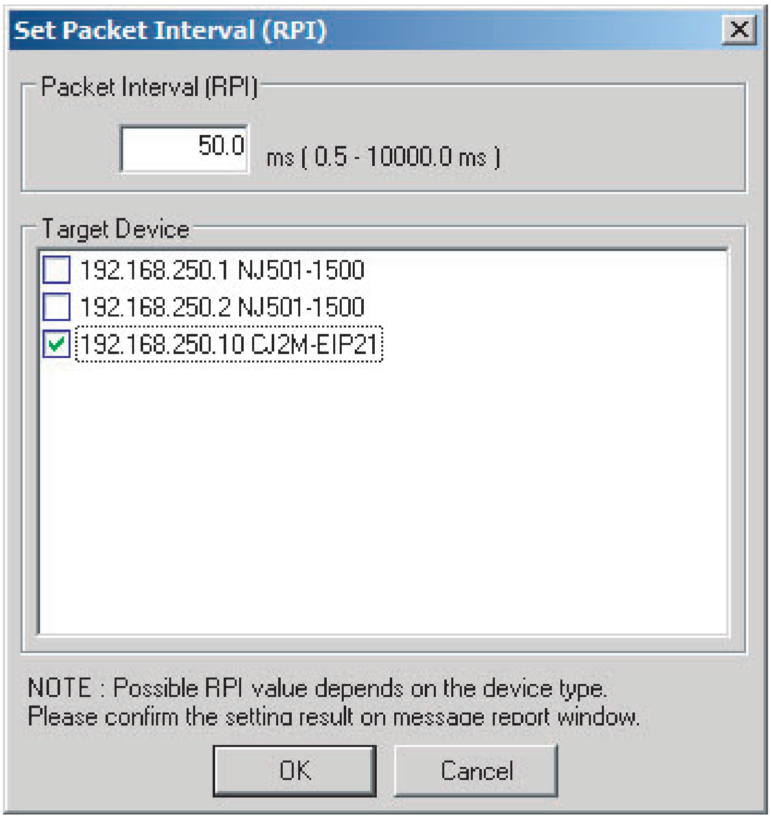
NOTE : Possible RPI value depends on the device type.
Please confirm the setting result on message report window.

Method 2:
Change a Particular Device’s Packet Interval (RPI) Setting:
The usage of capacity without multicast filtering can be adjusted for only a particular device against the tag data link’s allowable bandwidth by changing the packet intervals (RPI) settings for all of the device’s connections together. In this case, the usage of capacity will also change for the target devices of the connection for which the packet interval is changed.

**(1) Click the Set Packet Interval (RPI) Button at the bottom of the Usage of Device Bandwidth Dialog Box.**



**(2) The Set Packet Interval (RPI) Dialog Box will be displayed. In the *Target Device* Area, deselect the target devices that are not being adjusted by removing the check marks.**

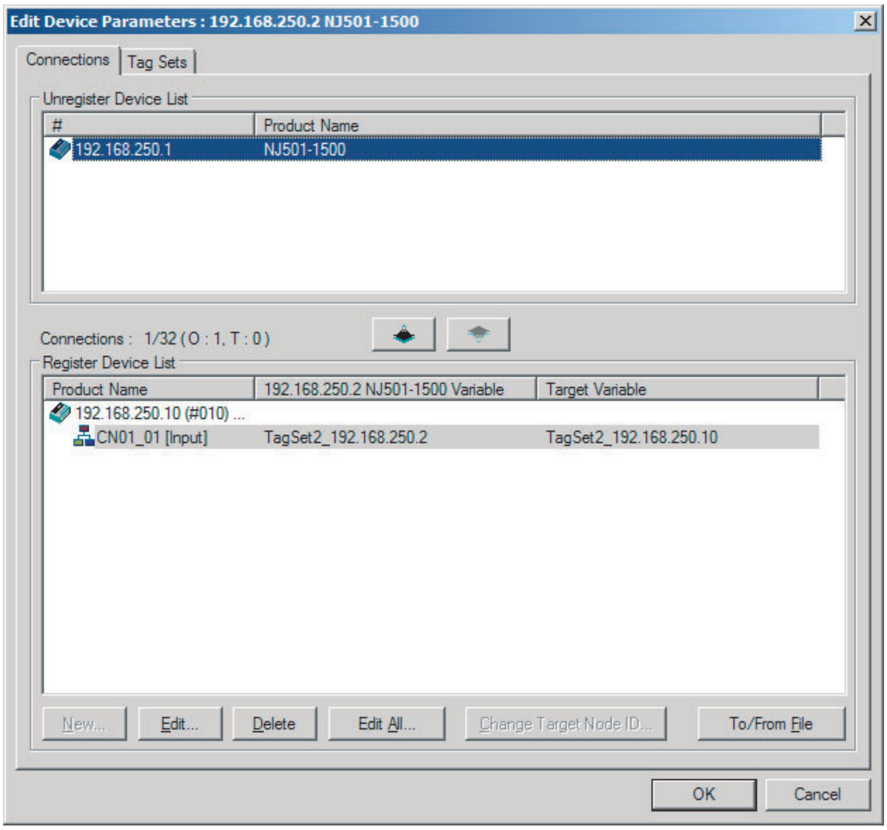

Set Packet Interval (RPI)
Packet Interval (RPI)
50.0
ms ( 0.5 - 10000.0 ms )
Target Device
□ 192.168.250.1 NJ501-1500
□ 192.168.250.2 NJ501-1500
☑ 192.168.250.10 CJ2M-EIP21
NOTE : Possible RPI value depends on the device type.
Please confirm the setting result on message report window.
OK
Cancel

**(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.**


Edit Device Parameters : 192.168.250.2 NJ501-1500

Unregister Device List

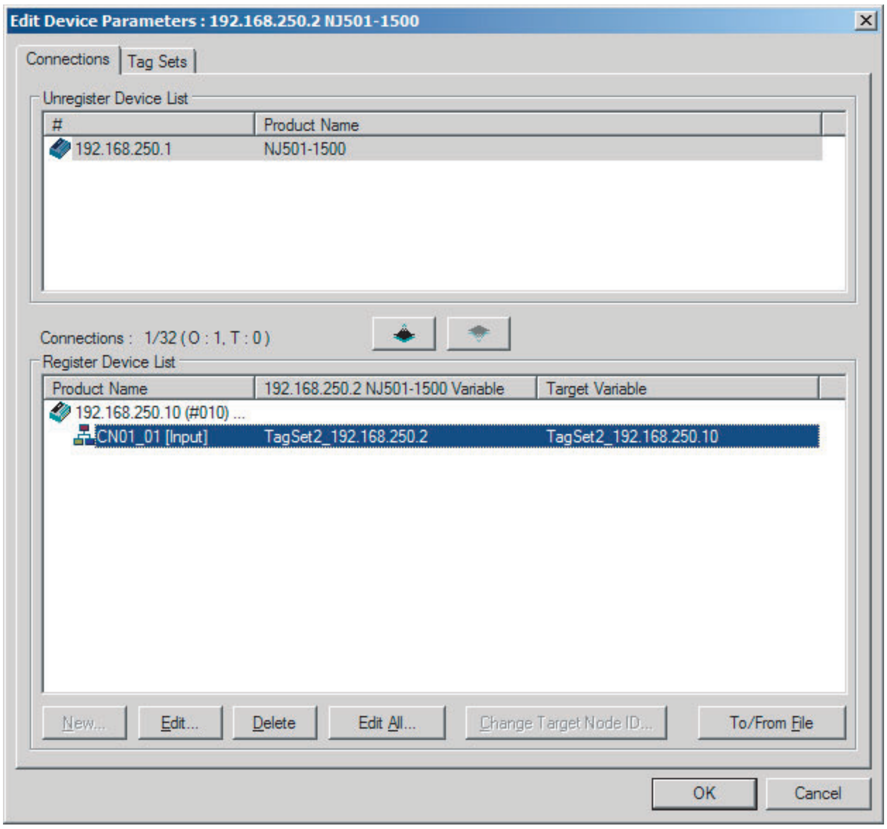
| # | Product Name |
|---|---|
| 192.168.250.1 | NJ501-1500 |

Connections : 1/32 ( O : 1, T : 0 )

Register Device List

| Product Name | 192.168.250.2 NJ501-1500 Variable | Target Variable |
|---|---|---|
| 192.168.250.10 (#010) ... | | |
| CN01_01 [Input] | TagSet2_192.168.250.2 | TagSet2_192.168.250.10 |

(3) **In the Register Device List, select the connection for which you want to change the RPI, and click the Edit Button.**



Edit Device Parameters : 192.168.250.2 NJ501-1500

Connections | Tag Sets

Unregister Device List

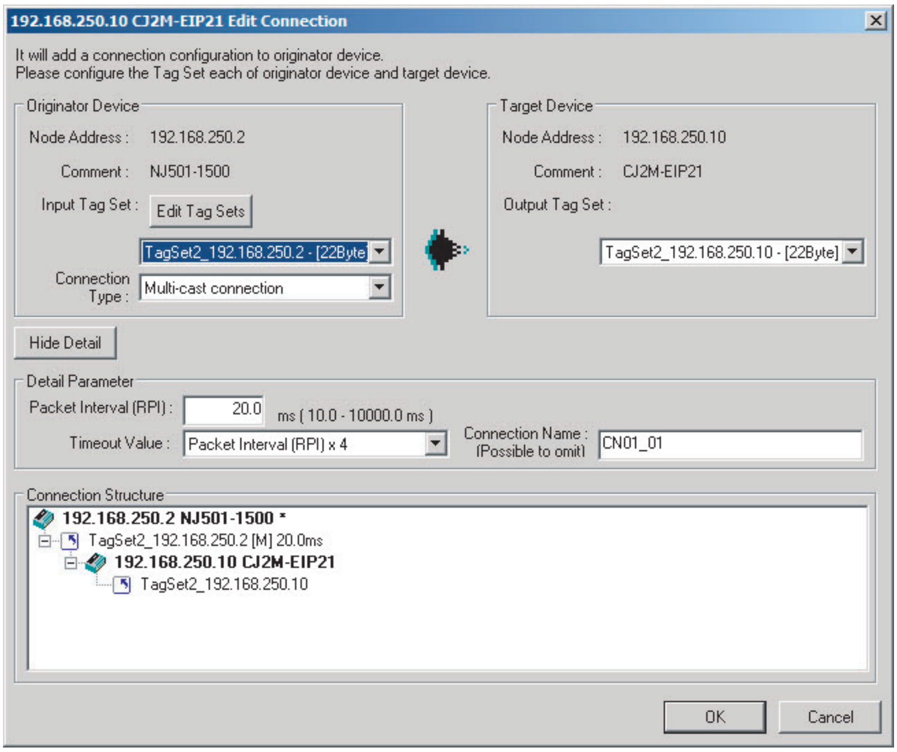
| # | Product Name |
|---|---|
| 192.168.250.1 | NJ501-1500 |

Connections : 1/32 (O : 1, T : 0)

Register Device List

| Product Name | 192.168.250.2 NJ501-1500 Variable | Target Variable |
|---|---|---|
| 192.168.250.10 (#010) ... | | |
| CN01_01 [Input] | TagSet2_192.168.250.2 | TagSet2_192.168.250.10 |

(4) **The device’s Edit Connection Dialog Box will be displayed. Input a new packet interval (RPI) value, and click the OK Button.**



192.168.250.10 CJ2M-EIP21 Edit Connection

It will add a connection configuration to originator device.
Please configure the Tag Set each of originator device and target device.

Originator Device

Node Address : 192.168.250.2

Comment : NJ501-1500

Input Tag Set : TagSet2_192.168.250.2 - [22Byte

Connection Type : Multi-cast connection

Target Device

Node Address : 192.168.250.10

Comment : CJ2M-EIP21

Output Tag Set : TagSet2_192.168.250.10 - [22Byte]

Detail Parameter

Packet Interval (RPI) : 20.0 ms ( 10.0 - 10000.0 ms )

Timeout Value : Packet Interval (RPI) x 4

Connection Name : (Possible to omit) CN01_01

Connection Structure

```
192.168.250.2 NJ501-1500 *
  TagSet2_192.168.250.2 [M] 20.0ms
    192.168.250.10 CJ2M-EIP21
      TagSet2_192.168.250.10
```

**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 *14-2-3 Adjusting Device Bandwidth Usage*.

- Reduce the number of nodes and number of connections.
- Split the network.

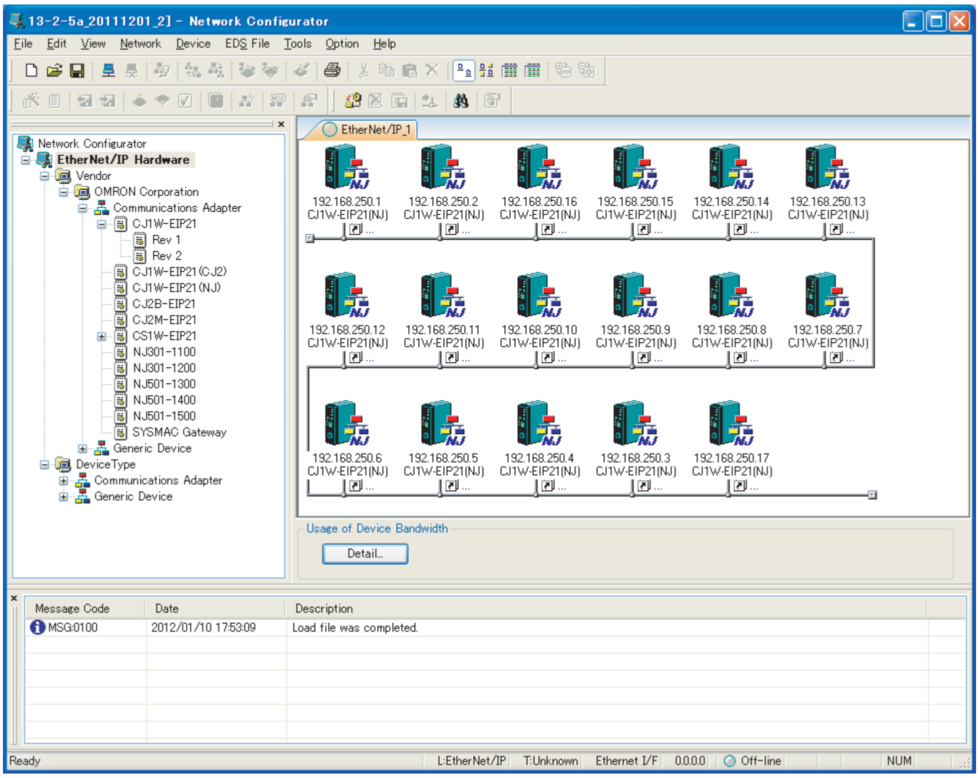
**5** Check the bandwidth usage again.
If you have changed the connection settings, click the **Detail** Button in the Usage of Device Bandwidth Area at the bottom of the Network Configuration Window and check bandwidth usage according to the instructions in *14-2-1 Checking Bandwidth Usage for Tag Data 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.

## 14-2-5 RPI Setting Examples

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

### Conditions



13-2-5a_20111201_2] - Network Configurator

File Edit View Network Device EDS File Tools Option Help

```
Network Configurator
  EtherNet/IP Hardware
    Vendor
      OMRON Corporation
        Communications Adapter
          CJ1W-EIP21
            Rev 1
            Rev 2
          CJ1W-EIP21(CJ2)
          CJ1W-EIP21(NJ)
          CJ2B-EIP21
          CJ2M-EIP21
          CS1W-EIP21
          NJ301-1100
          NJ301-1200
          NJ501-1300
          NJ501-1400
          NJ501-1500
          SYSMAC Gateway
        Generic Device
    DeviceType
      Communications Adapter
      Generic Device
```

EtherNet/IP_1

192.168.250.1 CJ1W-EIP21(NJ), 192.168.250.2 CJ1W-EIP21(NJ), 192.168.250.16 CJ1W-EIP21(NJ), 192.168.250.15 CJ1W-EIP21(NJ), 192.168.250.14 CJ1W-EIP21(NJ), 192.168.250.13 CJ1W-EIP21(NJ)

192.168.250.12 CJ1W-EIP21(NJ), 192.168.250.11 CJ1W-EIP21(NJ), 192.168.250.10 CJ1W-EIP21(NJ), 192.168.250.9 CJ1W-EIP21(NJ), 192.168.250.8 CJ1W-EIP21(NJ), 192.168.250.7 CJ1W-EIP21(NJ)

192.168.250.6 CJ1W-EIP21(NJ), 192.168.250.5 CJ1W-EIP21(NJ), 192.168.250.4 CJ1W-EIP21(NJ), 192.168.250.3 CJ1W-EIP21(NJ), 192.168.250.17 CJ1W-EIP21(NJ)

Usage of Device Bandwidth

Detail...

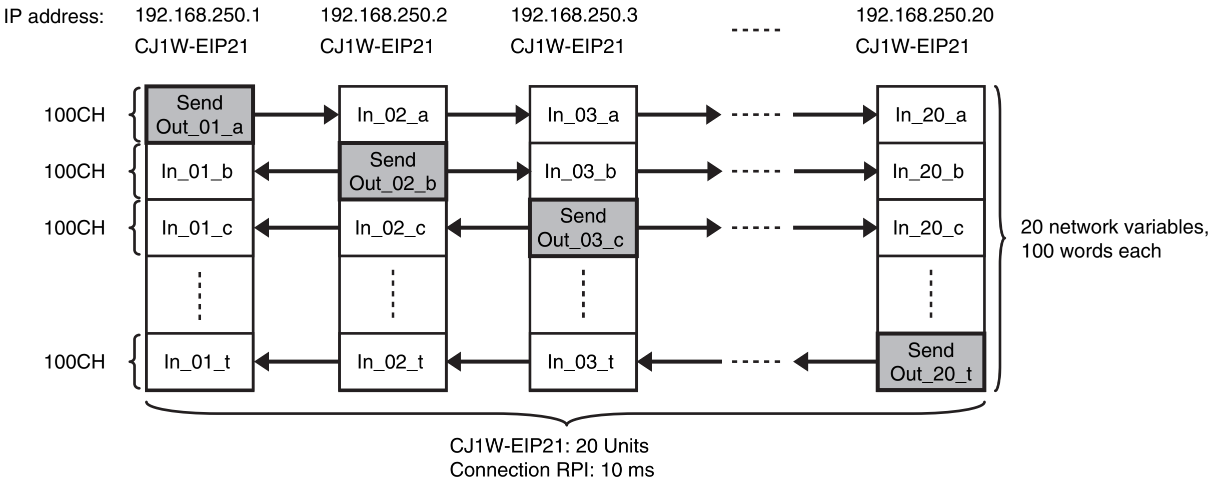
| Message Code | Date | Description |
|---|---|---|
| MSG:0100 | 2012/01/10 17:53:09 | Load file was completed. |

Ready | L:EtherNet/IP | T:Unknown | Ethernet I/F | 0.0.0.0 | Off-line | NUM

Usages of capacity shown in the following examples are calculated based on when CJ1W-EIP21 EtherNet/IP Units with unit version 2.1 or earlier are used.

- Connections:

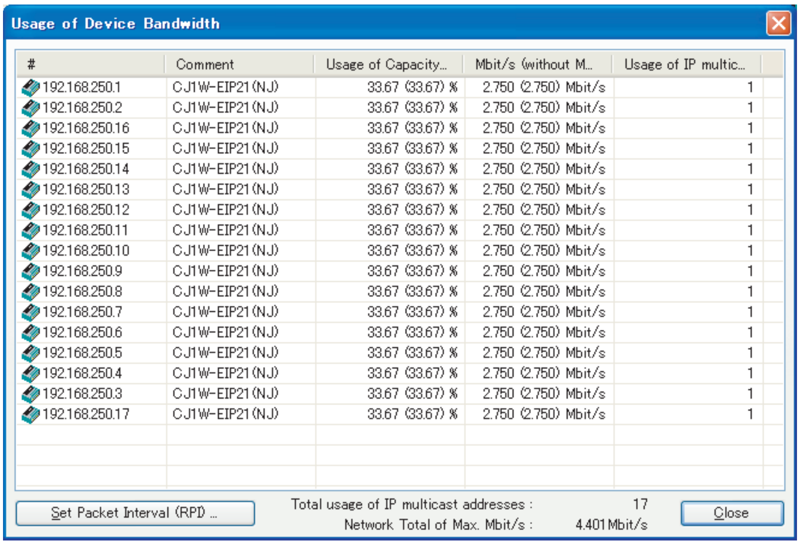
Example:Twenty CJ1W-EIP21 Units are connected to the network.
Each device has one 100-word tag for transmission and nineteen 100-word tags for reception, so that the Units exchange data mutually. By default, the packet intervals (RPIs) are set to 10 ms for all of the connections. The devices' IP addresses range from 192.168.250.1 to 192.168.250.20.



CJ1W-EIP21: 20 Units
Connection RPI: 10 ms

## 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 33.67%, as shown in the following dialog box.



| # | Comment | Usage of Capacity... | Mbit/s (without M... | Usage of IP multic... |
|---|---|---|---|---|
| 192.168.250.1 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.2 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.16 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.15 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.14 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.13 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.12 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.11 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.10 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.9 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.8 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.7 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.6 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.5 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.4 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.3 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |
| 192.168.250.17 | CJ1W-EIP21 (NJ) | 33.67 (33.67) % | 2.750 (2.750) Mbit/s | 1 |

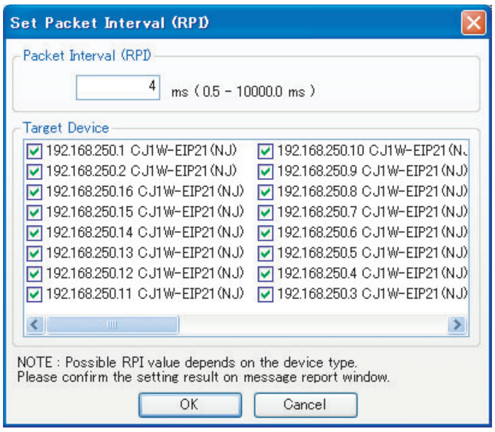
Total usage of IP multicast addresses : 17
Network Total of Max. Mbit/s : 4.401 Mbit/s

## 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 33.67% with the RPI set to 10 ms for all of the connections, so the RPI will be set to 4 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 4 ms as the new RPI value, then click the **OK** Button.

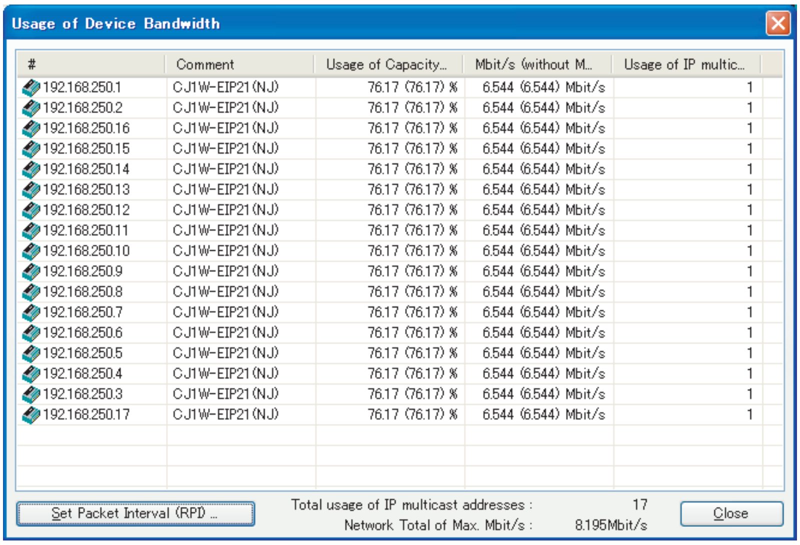


Packet Interval (RPI)
4 ms ( 0.5 - 10000.0 ms )

Target Device
☑ 192.168.250.1 CJ1W-EIP21 (NJ)
☑ 192.168.250.2 CJ1W-EIP21 (NJ)
☑ 192.168.250.16 CJ1W-EIP21 (NJ)
☑ 192.168.250.15 CJ1W-EIP21 (NJ)
☑ 192.168.250.14 CJ1W-EIP21 (NJ)
☑ 192.168.250.13 CJ1W-EIP21 (NJ)
☑ 192.168.250.12 CJ1W-EIP21 (NJ)
☑ 192.168.250.11 CJ1W-EIP21 (NJ)
☑ 192.168.250.10 CJ1W-EIP21 (N.
☑ 192.168.250.9 CJ1W-EIP21 (NJ)
☑ 192.168.250.8 CJ1W-EIP21 (NJ)
☑ 192.168.250.7 CJ1W-EIP21 (NJ)
☑ 192.168.250.6 CJ1W-EIP21 (NJ)
☑ 192.168.250.5 CJ1W-EIP21 (NJ)
☑ 192.168.250.4 CJ1W-EIP21 (NJ)
☑ 192.168.250.3 CJ1W-EIP21 (NJ)

NOTE : Possible RPI value depends on the device type.
Please confirm the setting result on message report window.

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 76.17% and the fastest set value is 4 ms.



Usage of Device Bandwidth

| # | Comment | Usage of Capacity... | Mbit/s (without M... | Usage of IP multic... |
|---|---|---|---|---|
| 192.168.250.1 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.2 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.16 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.15 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.14 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.13 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.12 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.11 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.10 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.9 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.8 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.7 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.6 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.5 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.4 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.3 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |
| 192.168.250.17 | CJ1W-EIP21 (NJ) | 76.17 (76.17) % | 6.544 (6.544) Mbit/s | 1 |

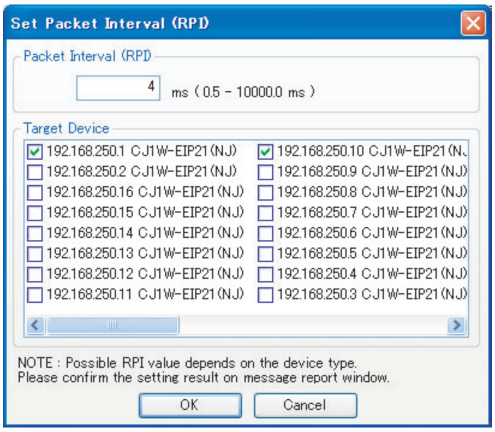
Total usage of IP multicast addresses : 17

Network Total of Max. Mbit/s : 8.195Mbit/s

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

In this example, we want faster tag data links for devices 192.168.250.1 and 192.168.250.10 only. To do this, click the **Set Packet Interval (RPI)** Button at the bottom of the Usage of Device Bandwidth Dialog Box. The Set Packet Interval (RPI) Dialog Box is displayed.

In the Target Device Area, clear the selections of all devices other than 192.168.250.1 and 192.168.250.10. Input 4 ms as the new RPI value, then click the **OK** Button.



Set Packet Interval (RPI)

Packet Interval (RPI)

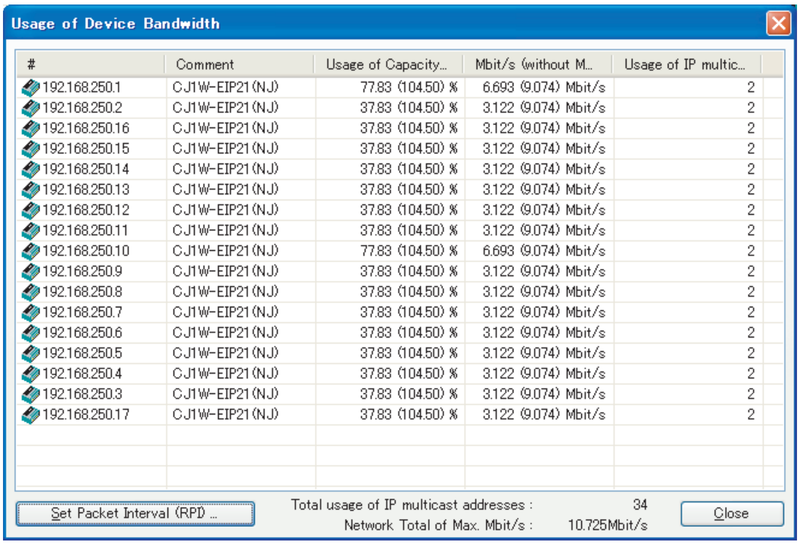
4 ms ( 0.5 - 10000.0 ms )

Target Device

☑ 192.168.250.1 CJ1W-EIP21 (NJ) ☑ 192.168.250.10 CJ1W-EIP21 (N...
☐ 192.168.250.2 CJ1W-EIP21 (NJ) ☐ 192.168.250.9 CJ1W-EIP21 (NJ)
☐ 192.168.250.16 CJ1W-EIP21 (NJ) ☐ 192.168.250.8 CJ1W-EIP21 (NJ)
☐ 192.168.250.15 CJ1W-EIP21 (NJ) ☐ 192.168.250.7 CJ1W-EIP21 (NJ)
☐ 192.168.250.14 CJ1W-EIP21 (NJ) ☐ 192.168.250.6 CJ1W-EIP21 (NJ)
☐ 192.168.250.13 CJ1W-EIP21 (NJ) ☐ 192.168.250.5 CJ1W-EIP21 (NJ)
☐ 192.168.250.12 CJ1W-EIP21 (NJ) ☐ 192.168.250.4 CJ1W-EIP21 (NJ)
☐ 192.168.250.11 CJ1W-EIP21 (NJ) ☐ 192.168.250.3 CJ1W-EIP21 (NJ)

NOTE : Possible RPI value depends on the device type.
Please confirm the setting result on message report window.

The percentage of the allowed tag data link bandwidth being used (Usage of Capacity) increases to 77.83% 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 33.64% to 37.83%) for all of the other devices, which connect with devices 192.168.250.1 and 192.168.250.10.



Usage of Device Bandwidth

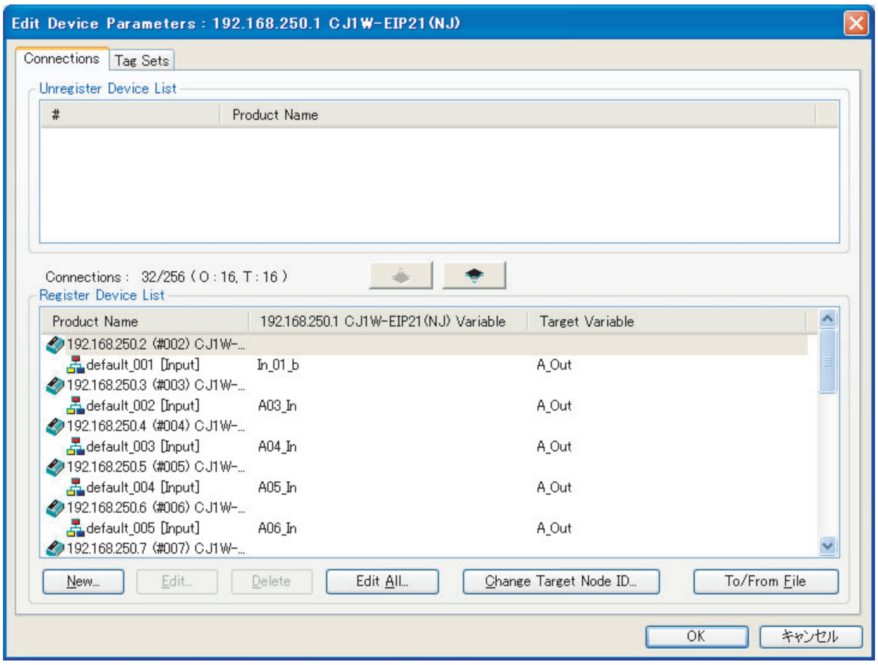
| # | Comment | Usage of Capacity... | Mbit/s (without M... | Usage of IP multic... |
|---|---|---|---|---|
| 192.168.250.1 | CJ1W-EIP21(NJ) | 77.83 (104.50) % | 6.693 (9.074) Mbit/s | 2 |
| 192.168.250.2 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |
| 192.168.250.16 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |
| 192.168.250.15 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |
| 192.168.250.14 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |
| 192.168.250.13 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |
| 192.168.250.12 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |
| 192.168.250.11 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |
| 192.168.250.10 | CJ1W-EIP21(NJ) | 77.83 (104.50) % | 6.693 (9.074) Mbit/s | 2 |
| 192.168.250.9 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |
| 192.168.250.8 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |
| 192.168.250.7 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |
| 192.168.250.6 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |
| 192.168.250.5 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |
| 192.168.250.4 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |
| 192.168.250.3 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |
| 192.168.250.17 | CJ1W-EIP21(NJ) | 37.83 (104.50) % | 3.122 (9.074) Mbit/s | 2 |

Total usage of IP multicast addresses : 34
Network Total of Max. Mbit/s : 10.725Mbit/s

In this case, if there is no multicast filter, the value becomes 104.50%. If there is no multicast filter for an Ethernet switch, communications errors may occur depending on the communications load of the EtherNet/IP Unit.

### 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.



Edit Device Parameters : 192.168.250.1 CJ1W-EIP21(NJ)

Unregister Device List

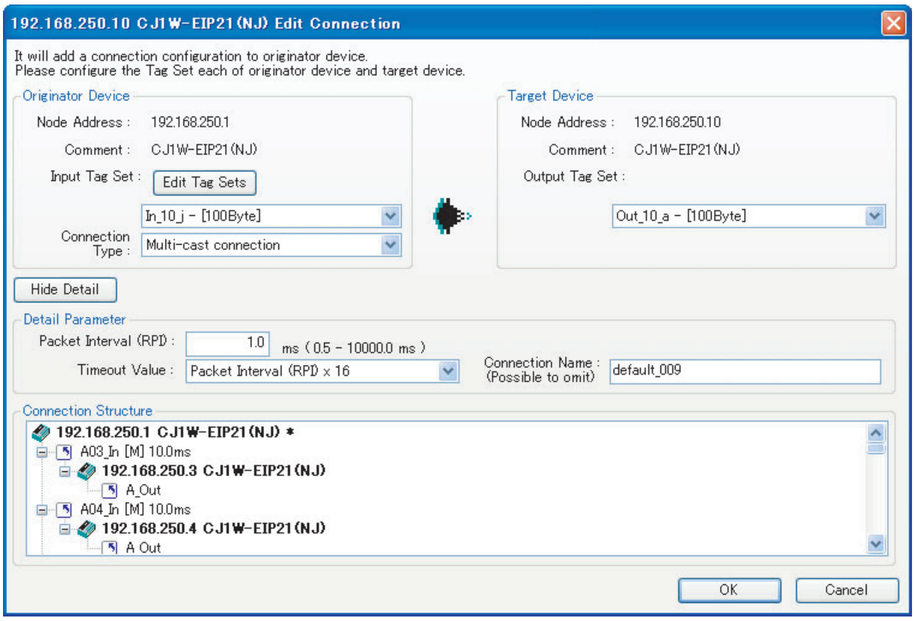
| # | Product Name |
|---|---|

Connections : 32/256 ( O : 16, T : 16 )

Register Device List

| Product Name | 192.168.250.1 CJ1W-EIP21(NJ) Variable | Target Variable |
|---|---|---|
| 192.168.250.2 (#002) CJ1W-... | | |
| default_001 [Input] | In_01_b | A_Out |
| 192.168.250.3 (#003) CJ1W-... | | |
| default_002 [Input] | A03_In | A_Out |
| 192.168.250.4 (#004) CJ1W-... | | |
| default_003 [Input] | A04_In | A_Out |
| 192.168.250.5 (#005) CJ1W-... | | |
| default_004 [Input] | A05_In | A_Out |
| 192.168.250.6 (#006) CJ1W-... | | |
| default_005 [Input] | A06_In | A_Out |
| 192.168.250.7 (#007) CJ1W-... | | |

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.



192.168.250.10 CJ1W-EIP21(NJ) Edit Connection

It will add a connection configuration to originator device.
Please configure the Tag Set each of originator device and target device.

Originator Device

Node Address : 192.168.250.1
Comment : CJ1W-EIP21(NJ)
Input Tag Set : In_10_j - [100Byte]
Connection Type : Multi-cast connection

Target Device

Node Address : 192.168.250.10
Comment : CJ1W-EIP21(NJ)
Output Tag Set : Out_10_a - [100Byte]

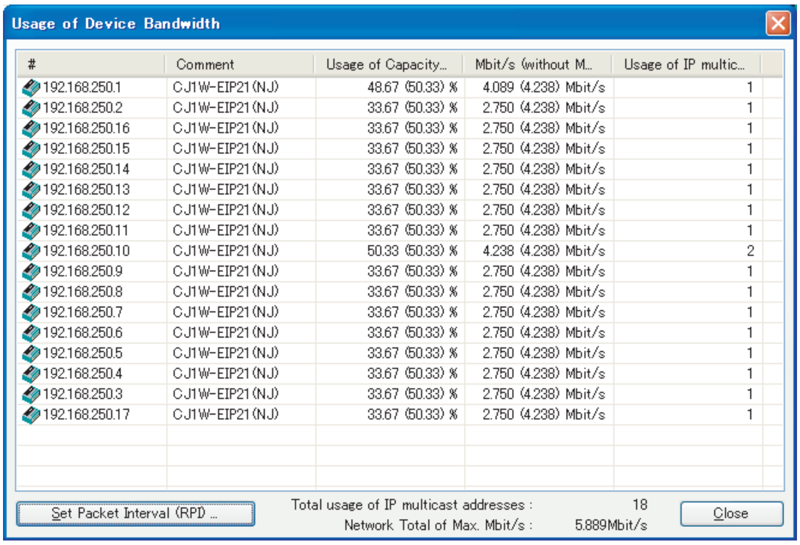
Detail Parameter

Packet Interval (RPI) : 1.0 ms ( 0.5 - 10000.0 ms )
Timeout Value : Packet Interval (RPI) x 16
Connection Name : (Possible to omit) default_009

Connection Structure

```
192.168.250.1 CJ1W-EIP21(NJ) *
  A03_In [M] 10.0ms
    192.168.250.3 CJ1W-EIP21(NJ)
      A_Out
  A04_In [M] 10.0ms
    192.168.250.4 CJ1W-EIP21(NJ)
      A_Out
```

In the Edit Connection Dialog Box, input 1 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 48.67%, which indicates that a RPI is set to a higher speed for this device.



Usage of Device Bandwidth

| # | Comment | Usage of Capacity... | Mbit/s (without M... | Usage of IP multic... |
|---|---|---|---|---|
| 192.168.250.1 | CJ1W-EIP21(NJ) | 48.67 (50.33) % | 4.089 (4.238) Mbit/s | 1 |
| 192.168.250.2 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |
| 192.168.250.16 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |
| 192.168.250.15 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |
| 192.168.250.14 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |
| 192.168.250.13 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |
| 192.168.250.12 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |
| 192.168.250.11 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |
| 192.168.250.10 | CJ1W-EIP21(NJ) | 50.33 (50.33) % | 4.238 (4.238) Mbit/s | 2 |
| 192.168.250.9 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |
| 192.168.250.8 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |
| 192.168.250.7 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |
| 192.168.250.6 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |
| 192.168.250.5 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |
| 192.168.250.4 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |
| 192.168.250.3 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |
| 192.168.250.17 | CJ1W-EIP21(NJ) | 33.67 (50.33) % | 2.750 (4.238) Mbit/s | 1 |

Total usage of IP multicast addresses : 18
Network Total of Max. Mbit/s : 5.889Mbit/s

In this case, the tag data link bandwidth that is used by device 192.168.250.10 (Usage of Capacity) also increases (from 33.67% to 50.33%).

# 14-3 I/O Response Time in Tag Data Links

**Additional Information**

This section provides information on the EtherNet/IP Unit. The data processing times for the built-in EtherNet/IP ports on the NJ501-□□□□, NJ301-□□□□, or NJ101-□□□□ NJ-series CPU Units, CJ2HCPU6□- EIP CPU Units, and CJ2M-CPU3□ CPU Units are different. For details, refer to *7-4 Tag Data Links with Models Other than NJ-Series CPU Units*.

As explained in *7-1-7 Concurrency of Tag Data Link Data*, the tag (network variable) with a refreshing task is refreshed when the refreshing task is executed in the user program. By setting the refreshing task, you can calculate the I/O response time that is not affected by the system service.

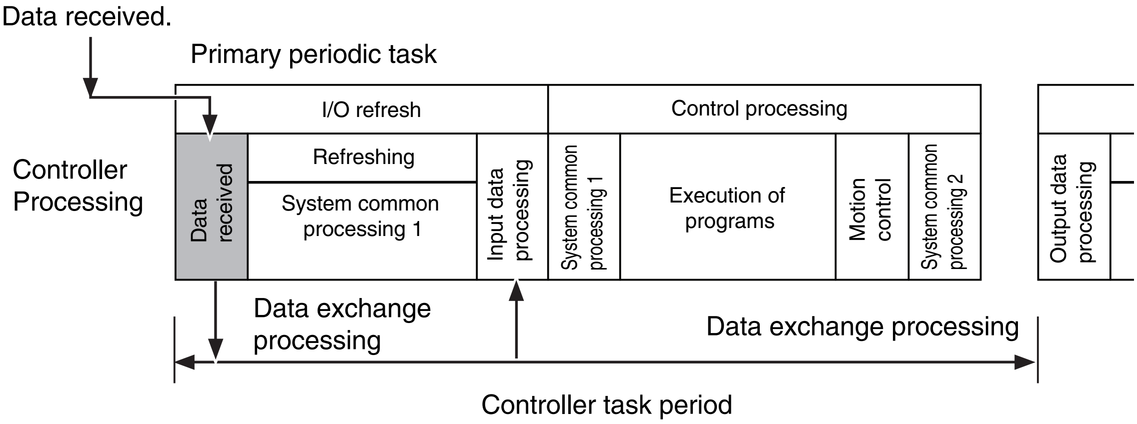
**Precautions for Correct Use**

The refreshing task must be set to all tags (network variables). If both tags (network variable) with a refreshing task and without it exist in a configuration, system service may affect the operation and I/O response time described in this section may not be maintained.

This section describes the I/O response time when refreshing tasks are set properly.

## 14-3-1 Timing of Data Transmissions

The following diagram shows the timing of tag data link transfers between the EtherNet/IP Unit and the CPU Unit. Data is transferred during I/O refresh processing for the task that is 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.

## 14-3-2 EtherNet/IP Unit Data Processing Time

This section describes the data processing time required to transfer data between the EtherNet/IP Unit and the CPU Unit.

### Data Processing Time Overview

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

#### 1. Variable Access Time

First, calculate the time required to transfer tag data (or the time required to access variables). This calculation is performed for each task. Therefore, if the same refreshing task is set for multiple tag sets, calculate the total time required for all tags in the tag sets.

Use the following equation to calculate the variable access time.

Variable access time [μs] = Total size of variables [bytes] × a + Number of variables × b
+ Number of accesses × c + d

Number of accesses: Number of tag sets

a to d: Constant values as given below

| CPU Unit model | Refreshing direction | Constant value [μs] | | | |
|---|---|---|---|---|---|
| | | a | b | c | d |
| NJ501-□□□□ | CPU Unit to EtherNet/IP Unit | 0.005 | 1.51*1 | 1.41 | 6.68 |
| | EtherNet/IP Unit to CPU Unit | 0.009 | 2.95*2 | | |
| NJ301-□□□□ | CPU Unit to EtherNet/IP Unit | 0.008 | 1.91*3 | 2.15 | 7.52 |
| | EtherNet/IP Unit to CPU Unit | 0.010 | 3.52*4 | | |
| NJ101-□□□□ | CPU Unit to EtherNet/IP Unit | 0.013 | 4.41 | 3.83 | 10.29 |
| | EtherNet/IP Unit to CPU Unit | 0.016 | 5.48 | | |

*1 The value is 1.58 for a CPU Unit with unit version 1.02 or earlier.

*2 The value is 3.18 for a CPU Unit with unit version 1.02 or earlier.

*3 The value is 2.14 for a CPU Unit with unit version 1.02 or earlier.

*4 The value is 4.08 for a CPU Unit with unit version 1.02 or earlier.

#### 2. Number of Data Transfers

Tag data is transferred as a part of task processing.
If the time required to process the data transfer is greater than the "variable access time" *2, the data cannot be sent entirely in one task period and is sent separately instead.

Number of data transfers = ("Time required to send the data entirely" *1 ÷ "Variable access time" *2)
+ 1*3

*1 This is the variable access time as calculated in step 1 above.

*2 The "variable access time" refers to the maximum processing time for accessing variables. Set the time for each task on the Task Setup Display, which is displayed by selecting ***Configurations and Setup*** − ***Task Setup*** in the Sysmac Studio.

*3 Delay in Data Transfer between the CPU Unit and an EtherNet/IP Unit

**Precautions for Correct Use**

The maximum number of words that can be transferred through the EtherNet/IP Unit 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. I/O Refreshing Time

You must calculate the processing time for I/O refreshing between the EtherNet/IP Unit and the CPU Unit. (I/O refreshing is when the data is actually transferred.)

The following two elements are relevant.

- Number of Fragments
  Even if the data size of the tag data links is equal to or less than the maximum link data size that can be sent and received for each EtherNet/IP Unit (369,664 bytes), the data transfer between the CPU Unit and the EtherNet/IP Unit must be fragmented if the data transfer size that can be processed by the CPU Unit in one transfer is exceeded. The maximum data transfer sizes are given below.

  Data Transfer Sizes for Each Data Transfer with the CPU Unit

  Output/send data: Approx. 14,810 bytes max.

  Input/receive data: Approx. 14,810 bytes max.

- I/O Refresh Processing Time
  The actual I/O refresh processing time depends on the following conditions.
  (Number of mounted CPU Bus Units × 1 ms) + I/O refresh time* + 1 task period

*1.I/O Refreshing Time Guidelines

| Total link data size<br>I/O refresh processing type | 20 bytes | 1,444 bytes | 2,888 bytes | 8,664 bytes | 12,864 bytes or more |
|---|---|---|---|---|---|
| Input refreshing | 0.3 ms | 0.5 ms | 0.7 ms | 1.6 ms | 2.1 ms |
| Output refreshing | 0.1 ms | 0.3 ms | 0.6 ms | 1.2 ms | 2.3 ms |

### 4. Actual Time Required for Data Transfer

The actual time that is required for data transfer is calculated as follows based on the values found for (2) and (3) above.

Task period × (Number of data transfers (2) + Number of fragments (3) – 1) + I/O refresh processing time (3)

## 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.

- CPU Unit with Tag Data Links:
  NJ501-□□□□
- Connection Direction
  The local EtherNet/IP Unit is set as the originator node.
- 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.

[(8 + 4 + 10) variables × 2.95 µs] + [(600 + 200 + 1,000) bytes × 0.009 µs] + 3 × 1.41 µs + 6.68 µs = 92.01 µs

**2** Calculate the number of data transfers.

| | |
|---|---|
| Time required for the data transfer: | “Variable access time” in step 1 = 92.01 µs |
| Variable access time set for the task: | 500 µs × 0.03 = 15 µs |
| Number of data transfers: | 92.01 µs ÷ 15 µs + 1 = 7.13 times |

Thus, approximately 8 data transfers are required.

**3** Calculate the I/O refresh processing time.

1 unit × 1 ms + 0.5 ms + 0.5 ms = 2 ms (2,000 µs)

**4** Calculate the actual time required for the data transfer.

500 µs × (8 times + 1 – 1) + 2,000 µs = 6,000 µs

## 14-3-3 Effect of Tag Data Links on Task Periods

The tag data is transferred during task processing. Therefore, if you need to complete transfer processing for task data within one task period, adjust the variable access time and task period settings in the Task Setup to change the task period.

**1** Calculate the time required for the data transfer and set the result as the “variable access time”. Refer to *1. Variable Access Time* for the formula to calculate the variable access time.

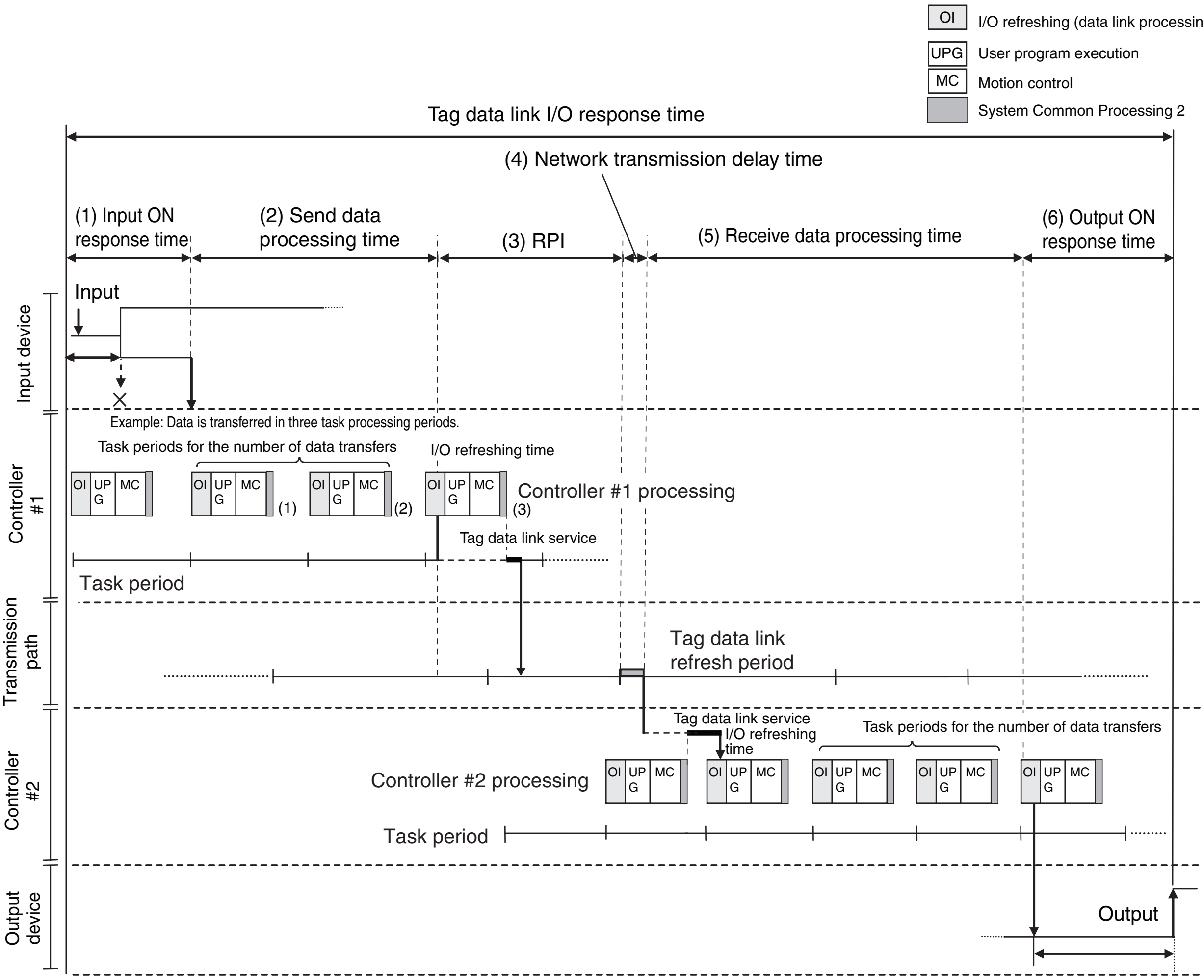
**Note** If the same refreshing task is set for multiple tag sets, the total of all tag values in the tag sets is used.

**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/NX-series CPU Unit Software User’s Manual* (Cat. No. W501).

## 14-3-4 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.


OI I/O refreshing (data link processing)
UPG User program execution
MC Motion control
System Common Processing 2
Tag data link I/O response time
(4) Network transmission delay time
(1) Input ON response time
(2) Send data processing time
(3) RPI
(5) Receive data processing time
(6) Output ON response time
Input device
Input
Example: Data is transferred in three task processing periods.
Task periods for the number of data transfers
I/O refreshing time
Controller #1
OI UPG MC
(1)
(2)
(3)
Controller #1 processing
Tag data link service
Task period
Transmission path
Tag data link refresh period
Tag data link service
I/O refreshing time
Task periods for the number of data transfers
Controller #2
Controller #2 processing
Task period
Output device
Output

- With unit version 1.03 or later, processing for tag data links is executed in the tag data link service.
- With unit version 1.01 to 1.02, processing for tag data links is executed in the system services. 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 EtherNet/IP Unit. 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 *14-3-2 EtherNet/IP Unit 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 EtherNet/IP Unit 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 *14-3-2 EtherNet/IP Unit 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

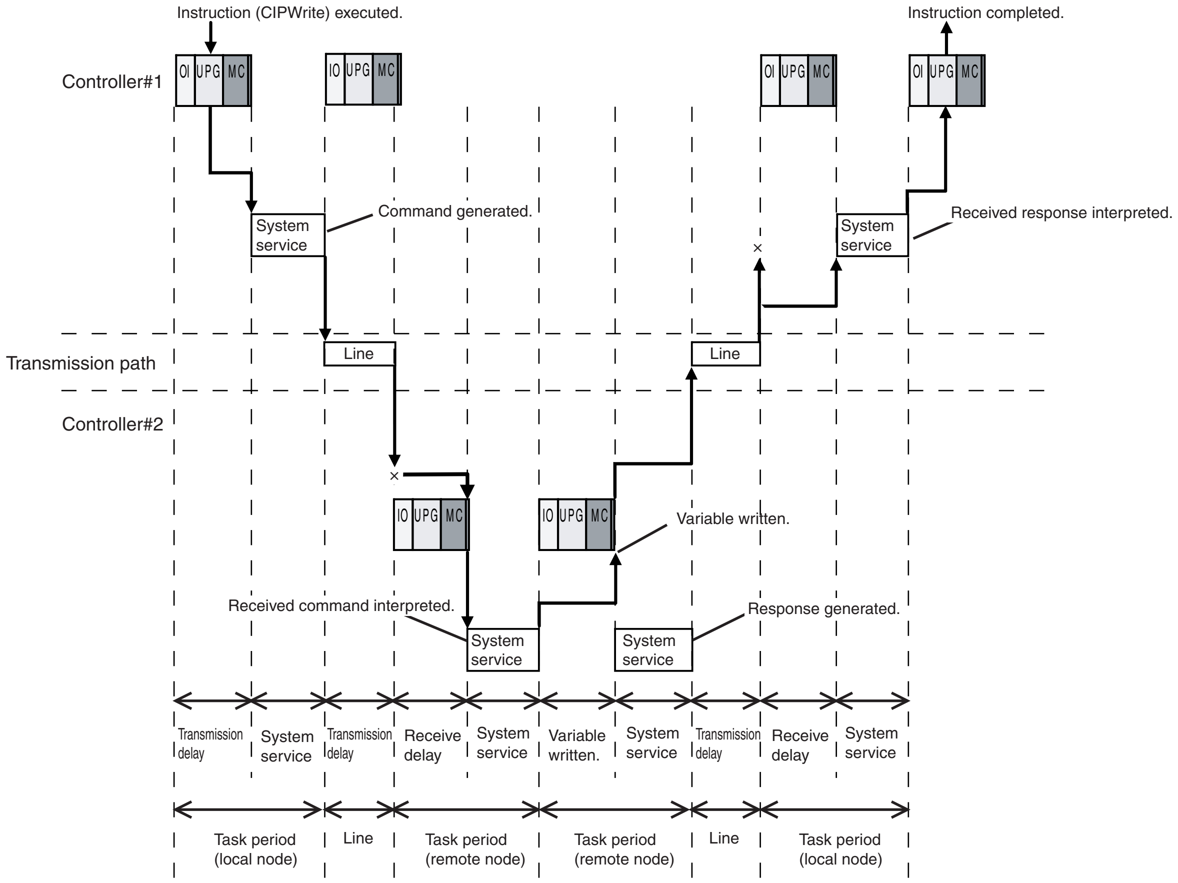
This is the delay time for the external output device from when the Controller specified turning ON the output until the output is actually turned ON.
Output ON response time = Output device delay time + Task period

**Additional Information**

The I/O response time may be longer due to noise, or other events.

# 14-4 Message Service Transmission Delay

This section describes the delay time that occurs in CIP communications instruction (CIPWrite) service processing.



$$\text{Maximum transmission delay time} = \underbrace{\text{Send delay} + \text{System service execution time}}_{\text{(Local node task period)}} + \text{Transmission delay} + \underbrace{\text{Receive delay} + \text{System service execution time}}_{\text{(Remote node task period)}}$$

$$+ \underbrace{\text{Variable write time} + \text{System service execution time}}_{\text{(Remote node task period)}} + \text{Transmission delay} + \underbrace{\text{Receive delay} + \text{System service execution time}}_{\text{(Local node task period)}}$$

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 EtherNet/IP Unit 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.

15

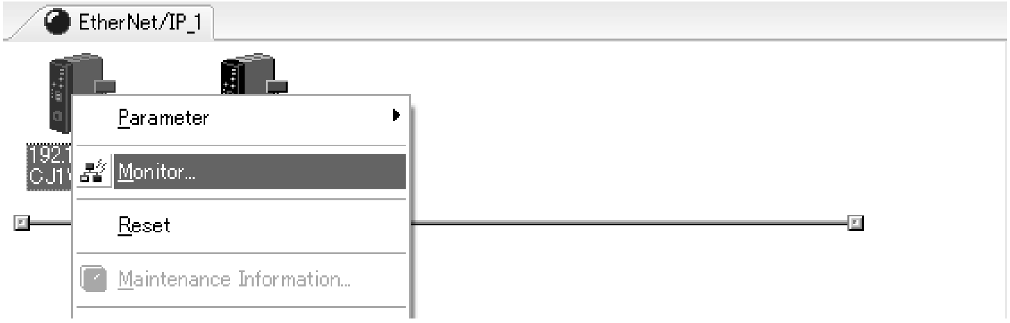
# Troubleshooting

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.

# 15-1 Checking Status with the Network Configurator

## 15-1-1 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***.



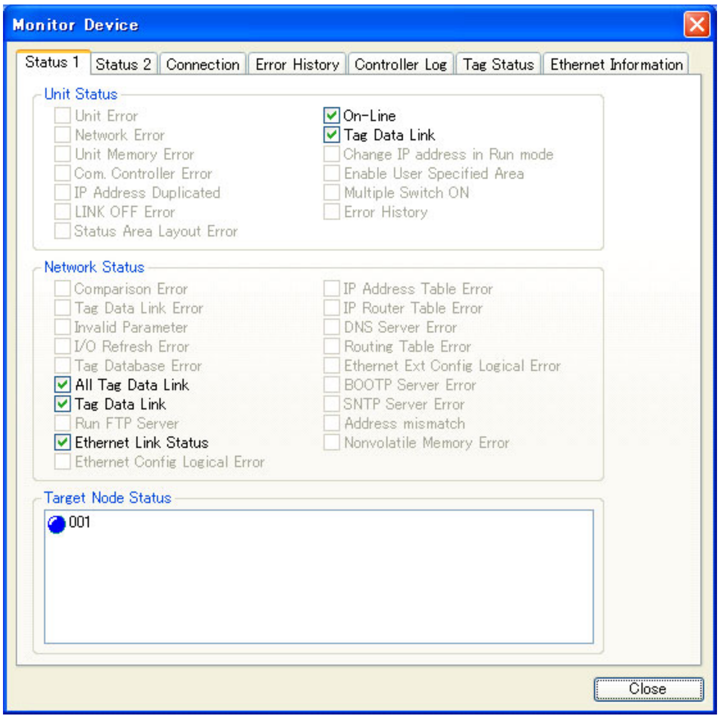
The Monitor Device Dialog Box will be displayed.

**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 *Status 1* Tab Page shows the status that you can monitor with device variables for the CJ-series Unit (i.e., Unit Status 1, Unit Status 2, Communications Status 1, Communications Status 2, and Communications Status 3). The given status is TRUE if the check box for it is selected. In addition, the *Target Node Status* Area shows the connection status of the target nodes that are connected to the EtherNet/IP Unit as the tag data link originator. 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, this information is displayed in red.



Unit Status

- □ Unit Error
- □ Network Error
- □ Unit Memory Error
- □ Com. Controller Error
- □ IP Address Duplicated
- □ LINK OFF Error
- □ Status Area Layout Error
- ☑ On-Line
- ☑ Tag Data Link
- □ Change IP address in Run mode
- □ Enable User Specified Area
- □ Multiple Switch ON
- □ Error History

Network Status

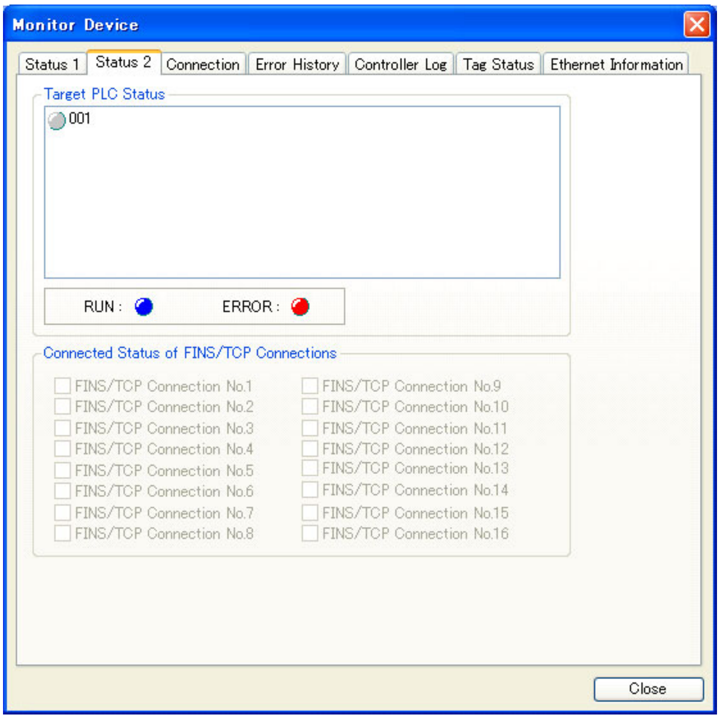
- □ Comparison Error
- □ Tag Data Link Error
- □ Invalid Parameter
- □ I/O Refresh Error
- □ Tag Database Error
- ☑ All Tag Data Link
- ☑ Tag Data Link
- □ Run FTP Server
- ☑ Ethernet Link Status
- □ Ethernet Config Logical Error
- □ IP Address Table Error
- □ IP Router Table Error
- □ DNS Server Error
- □ Routing Table Error
- □ Ethernet Ext Config Logical Error
- □ BOOTP Server Error
- □ SNTP Server Error
- □ Address mismatch
- □ Nonvolatile Memory Error

Target Node Status

● 001

### ● Status 2 Tab Page

The *Target Controller Status* Area on the Status 2 Tab Page shows the Target Node PLC Operating Flags and Target Node PLC Error Flags from the Normal Target Node Table of the target table information that you can monitor with the device variables for the CJ-series Unit for status for the nodes for which the EtherNet/IP Unit is set as the originator of the tag data links. The node is displayed in blue if the connection is normal, or red if there is an error. The *Connected Status of FINS/TCP Connections* Area shows the status of FINS/TCP connections. The check box is selected if the corresponding connection is established. The *Connected Status of FINS/TCP Connections* Field shows the status of FINS/TCP connections. There will be a check mark in the box when the corresponding connection is established (connected).



Target PLC Status

○ 001

RUN : ●  ERROR : ●

Connected Status of FINS/TCP Connections

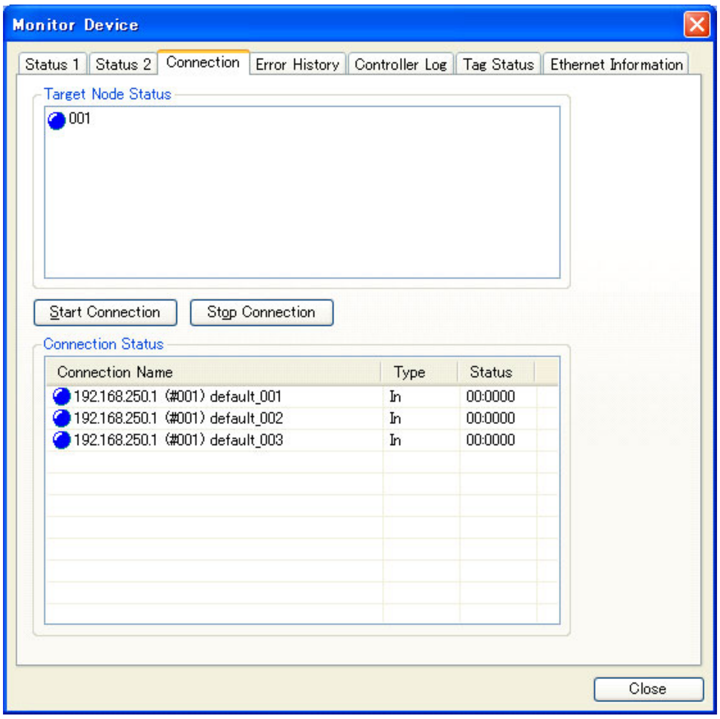
□ FINS/TCP Connection No.1  □ FINS/TCP Connection No.9
□ FINS/TCP Connection No.2  □ FINS/TCP Connection No.10
□ FINS/TCP Connection No.3  □ FINS/TCP Connection No.11
□ FINS/TCP Connection No.4  □ FINS/TCP Connection No.12
□ FINS/TCP Connection No.5  □ FINS/TCP Connection No.13
□ FINS/TCP Connection No.6  □ FINS/TCP Connection No.14
□ FINS/TCP Connection No.7  □ FINS/TCP Connection No.15
□ FINS/TCP Connection No.8  □ FINS/TCP Connection No.16

#### 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 nodes that act 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 the information is displayed in red. However, this information is displayed in gray if the connection to a node is stopped. In addition, the *Connection Status* Area shows the current status each connection that is set as the originator. This information can be used to identify the cause of tag data link errors. For details on the connection status, refer to *15-3 Connection Status Codes and Error Processing*.



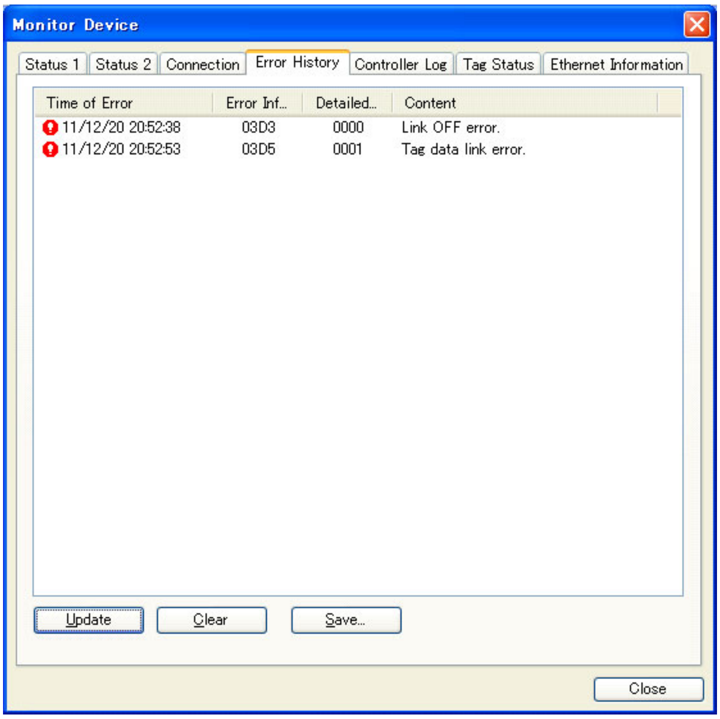
● 001

| Connection Name | Type | Status |
|---|---|---|
| ● 192.168.250.1 (#001) default_001 | In | 00:0000 |
| ● 192.168.250.1 (#001) default_002 | In | 00:0000 |
| ● 192.168.250.1 (#001) default_003 | In | 00:0000 |

## Error History Tab Page

The *Error History* Tab Page displays the error log stored in the EtherNet/IP Unit. Errors that occurred in the past are recorded, and can be saved in a computer file as required.

In some cases, error records are cleared when the power is turned OFF, and in other cases the records are retained. For details on the error log, refer to *15-4 Error Log*.

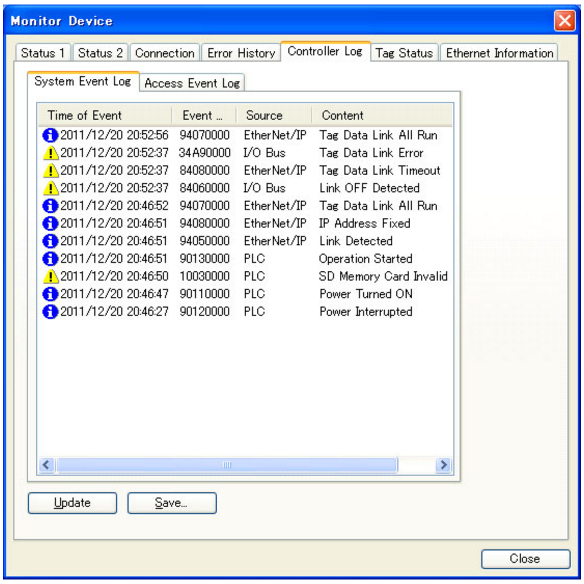


| Time of Error | Error Inf... | Detailed... | Content |
|---|---|---|---|
| 11/12/20 20:52:38 | 03D3 | 0000 | Link OFF error. |
| 11/12/20 20:52:53 | 03D5 | 0001 | Tag data link error. |

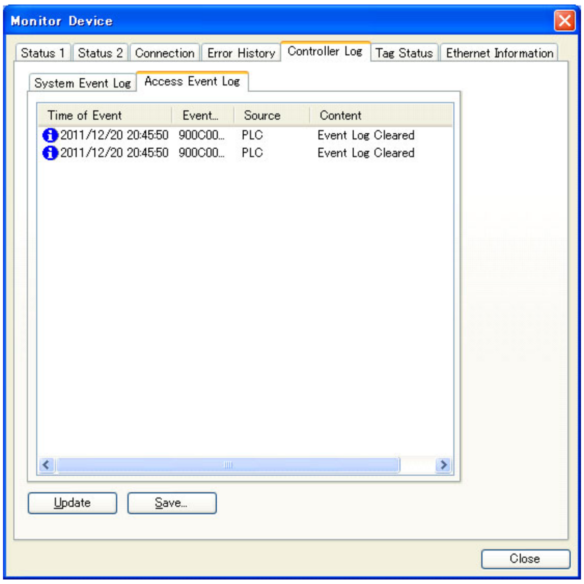
## ● Controller Log Tab Page

This tab page displays the Controller event log that is stored in the CPU Unit of the Controller where the EtherNet/IP is connected. The error log 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.



| Time of Event | Event ... | Source | Content |
|---|---|---|---|
| ⓘ 2011/12/20 20:52:56 | 94070000 | EtherNet/IP | Tag Data Link All Run |
| ⚠ 2011/12/20 20:52:37 | 34A90000 | I/O Bus | Tag Data Link Error |
| ⚠ 2011/12/20 20:52:37 | 84080000 | EtherNet/IP | Tag Data Link Timeout |
| ⚠ 2011/12/20 20:52:37 | 84060000 | I/O Bus | Link OFF Detected |
| ⓘ 2011/12/20 20:46:52 | 94070000 | EtherNet/IP | Tag Data Link All Run |
| ⓘ 2011/12/20 20:46:51 | 94080000 | EtherNet/IP | IP Address Fixed |
| ⓘ 2011/12/20 20:46:51 | 94050000 | EtherNet/IP | Link Detected |
| ⓘ 2011/12/20 20:46:51 | 90130000 | PLC | Operation Started |
| ⚠ 2011/12/20 20:46:50 | 10030000 | PLC | SD Memory Card Invalid |
| ⓘ 2011/12/20 20:46:47 | 90110000 | PLC | Power Turned ON |
| ⓘ 2011/12/20 20:46:27 | 90120000 | PLC | Power Interrupted |



| Time of Event | Event... | Source | Content |
|---|---|---|---|
| ⓘ 2011/12/20 20:45:50 | 900C00... | PLC | Event Log Cleared |
| ⓘ 2011/12/20 20:45:50 | 900C00... | PLC | Event Log Cleared |

## ● Tag Status Tab Page

This tab page shows 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 to show the status of the settings.

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.

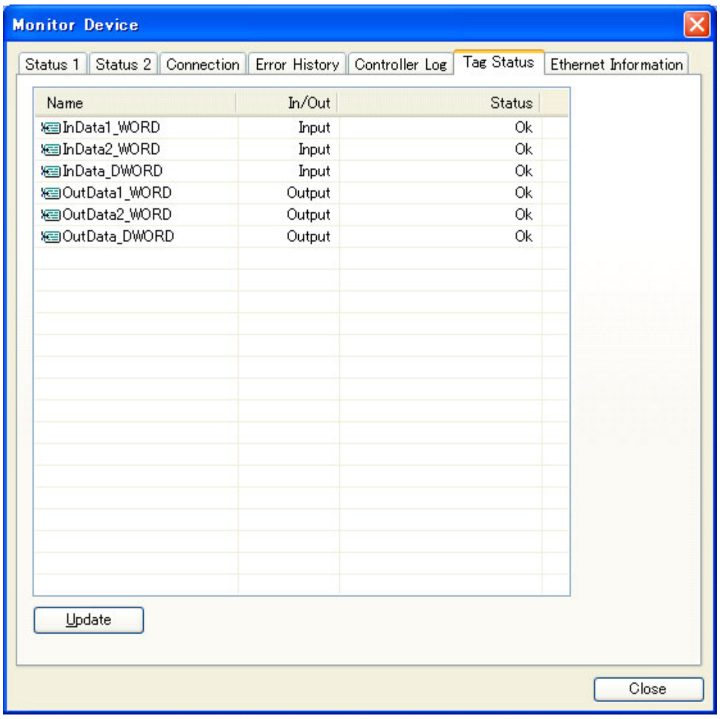
Size does not match error: Different sizes are set for the network variables and the tag settings. A connection will not be established for a tag for which this error occurs.

No tag: A network variable is not set in the variable table in the CPU Unit for the specified tag setting. Or, instead of a member of union variable, unions are specified.
A connection will not be established for a tag for which this error occurs.

Controller I/F error: The system service execution time at startup of the CPU unit is insufficient. Review the “Start delay time at startup” setting. Refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) for details of the “Start delay time at startup” setting.
If the error still persists after reviewing the “Start delay time at startup” settings, there is a problem in the bus interface with the CPU Unit. Determine the cause based on the indicators and the error log.

Attribute error: Writing is not possible for Read Only and Constant attributes.

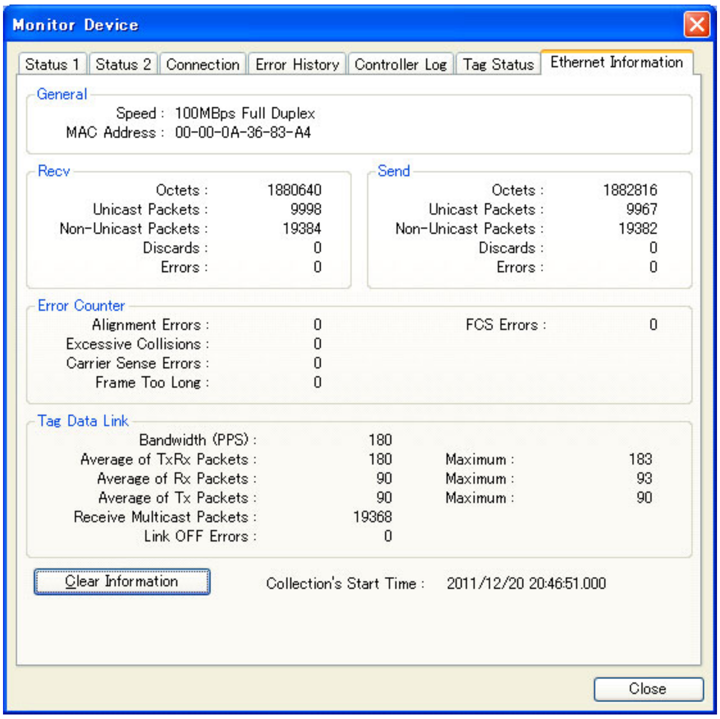
If the status is not “Normal resolution completed,” check the tag data link settings or the network variable settings in the variable table in the NJ-series CPU Unit.



| Name | In/Out | Status |
|---|---|---|
| InData1_WORD | Input | Ok |
| InData2_WORD | Input | Ok |
| InData_DWORD | Input | Ok |
| OutData1_WORD | Output | Ok |
| OutData2_WORD | Output | Ok |
| OutData_DWORD | Output | Ok |

## ● Ethernet Information Tab Page

This tab page displays the communications status at the communications driver level of the EtherNet/IP Unit. 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).



General
Speed : 100MBps Full Duplex
MAC Address : 00-00-0A-36-83-A4

| Recv | | Send | |
|---|---|---|---|
| Octets : | 1880640 | Octets : | 1882816 |
| Unicast Packets : | 9998 | Unicast Packets : | 9967 |
| Non-Unicast Packets : | 19384 | Non-Unicast Packets : | 19382 |
| Discards : | 0 | Discards : | 0 |
| Errors : | 0 | Errors : | 0 |

Error Counter

| | | | |
|---|---|---|---|
| Alignment Errors : | 0 | FCS Errors : | 0 |
| Excessive Collisions : | 0 | | |
| Carrier Sense Errors : | 0 | | |
| Frame Too Long : | 0 | | |

Tag Data Link

| | | | |
|---|---|---|---|
| Bandwidth (PPS) : | 180 | | |
| Average of TxRx Packets : | 180 | Maximum : | 183 |
| Average of Rx Packets : | 90 | Maximum : | 93 |
| Average of Tx Packets : | 90 | Maximum : | 90 |
| Receive Multicast Packets : | 19368 | | |
| Link OFF Errors : | 0 | | |

Collection's Start Time : 2011/12/20 20:46:51.000

# 15-2 Using the LED Indicators and Display for Troubleshooting

## 15-2-1 Errors Occurring at the EtherNet/IP Unit

| Indicator | | | Error | Cause | Unit operation | Error log (hex) | Countermeasure |
|---|---|---|---|---|---|---|---|
| MS | NS | 7-segment* | | | | | |
| **Errors Related to CPU Unit Data Exchange** | | | | | | | |
| Flashing red | Not lit | H1 | Duplicate unit number | The same unit number is set on another Unit. | Operation stops. | --- | Set the unit numbers correctly and restart the EtherNet/IP Unit. |
| Flashing red | Not lit | H2 | CPU Unit faulty | --- | Operation stops. | --- | Replace the CPU Unit if the error recurs when the CPU Unit is restarted. |
| Lit red | Not lit | H3 | EtherNet/IP Unit faulty | --- | Operation stops. | --- | Replace the EtherNet/IP Unit if the error recurs when the Unit is restarted. |
| Flashing red | Not lit | H4 | Node address setting error | The node address set on the switches is invalid (00 or FF.) | Operation stops. | --- | Set the node address correctly and restart the EtherNet/IP Unit. |
| Flashing red | Not lit | H6 | CPU Unit faulty | --- | Records the error in the error log (time/date all zeroes).<br>Operation stops. | 000F | Replace the CPU Unit if the error recurs when the CPU Unit is restarted. |
| Flashing red | Not lit | H7 | I/O table not registered | The CPU Unit's I/O table is not registered. | Operation stops. | 0006 | Create the I/O table. |
| Flashing red | --- | H8 | Restoring SD Memory Card backup or Sysmac Studio Controller backup data failed. | Restoring backup data failed for some reason. | All settings for the EtherNet/IP Unit will be cleared except for the following errors: File missing, memory not mounted, or inconsistent models. | --- | If the problem persists even when you repeat the simple backup procedure, replace the SD Memory Card, the EtherNet/IP Unit, or, if the problem is with the built-in port, the CPU Unit. |
| Flashing red | --- | H9 | I/O bus error | An error occurred while exchanging data with the CPU Unit. | • If the Unit is the originator of the tag data link connection, it stops communications.<br>• Communications will continue for target connection. If the Unit is the target of the tag data link connection and the Controller status is included in the communications data, the corresponding element in the Target Node PLC Error Flags will change to TRUE. | 000E | Check and correct the CPU Unit's operating environment. |

* The 7-segment display alternates between the node address and error code.

| Indicator | | | Error | Cause | Unit operation | Error log (hex) | Countermeasure |
|---|---|---|---|---|---|---|---|
| MS | NS | 7-segment* | | | | | |
| **Errors Related to CPU Unit Data Exchange** | | | | | | | |
| Flashing red | --- | HA | CPU Unit memory error | A parity error occurred during an operation such as reading the routing tables. | Records the error in the error log. If the routing tables were being read, the routing tables are treated as missing. | 0012 | Register the routing tables in the CPU Unit again and restart the CPU Unit.<br>Replace the CPU Unit if the error recurs. |
| | | | | A memory error has occurred for the tag database in the CPU Unit | • If a variable (tag name) is specified in the tag data link or Unit Status Area, refreshing the user-specified status area is stopped and tag data links will operate as follows:<br>• Tag data link communications will be stopped for originator connections.<br>• Communications will continue for target connection. If the Unit is the target of the tag data link connection and the Controller status is included in the communications data, the corresponding element in the Target Node PLC Error Flags will change to TRUE.<br>**Note** Recovery is possible from this error. If recovery is achieved, the tag data links will be restarted to return to normal status. | 0017 | Download the tag data to the CPU Unit again.<br>Replace the CPU Unit if the error recurs. |
| Flashing red | Not lit | Hb | CPU Unit event servicing timeout | A timeout occurred during an operation such as reading the routing tables to the CPU Unit. | Operation stops. | 0011 | Replace the EtherNet/IP Unit or the CPU Unit if the error recurs when the Unit is restarted. |
| Flashing red | --- | HC | Routing table error | There is a logic error in the routing table settings. | The Unit continues operating without the routing tables. | 021A | Create the routing tables again. |

* The 7-segment display alternates between the node address and error code.

15-2 Using the LED Indicators and Display for Troubleshooting

15

15-2-1 Errors Occurring at the EtherNet/IP Unit

| Indicator | | | Error | Cause | Unit operation | Error log (hex) | Countermeasure |
|---|---|---|---|---|---|---|---|
| MS | NS | 7-segment* | | | | | |
| **Errors Related to CPU Unit Data Exchange** | | | | | | | |
| Flashing red | --- | HE | CPU Unit service monitoring error | Servicing from the CPU Unit was not completed within the fixed interval. The monitoring time is normally 11 s. | • If the Unit is the originator of the tag data link connection, it stops communications.<br>• Communications will continue for target connection. If the Unit is the target of the tag data link connection and the Controller status is included in the communications data, the corresponding element in the Target Node PLC Error Flags will change to TRUE.<br>**Note** Recovery is possible for this error. When operation is restored, tag data link startup processing will be performed and operations will return to normal. | 0002 | Check and correct the CPU Unit's operating environment. |
| Flashing red | --- | HF | CPU Unit watchdog timer error | An error occurred in the CPU Unit. | • If the Unit is the originator of the tag data link connection, it stops communications.<br>• Communications will continue for target connection. If the Unit is the target of the tag data link connection and the Controller status is included in the communications data, the Target Node PLC Error Flags will change to TRUE. | 0001 | Replace the CPU Unit. |

* The 7-segment display alternates between the node address and error code.

| Indicator | | | Error | Cause | Unit operation | Error log (hex) | Countermeasure |
|---|---|---|---|---|---|---|---|
| MS | NS | 7-segment* | | | | | |
| **Errors Related to CPU Unit Data Exchange** | | | | | | | |
| Flashing red | --- | HH | CPU Unit Fatal Error | A fatal error occurred in the CPU Unit. | • If the Unit is the originator of the tag data link connection, it stops communications.<br>• Communications will continue for target connection. If the Unit is the target of the tag data link connection and the Controller status is included in the communications data, the corresponding element in the Target Node PLC Error Flags will change to TRUE.<br>• The tag data link's send data will be cleared to 0 in accordance with the Output OFF settings, and data transfer will continue with that data. | 0015 | Eliminate the cause of the error in the CPU Unit.<br>The tag data link will restart automatically when the cause of the error is eliminated. |
| **Errors Related to the Software Switches** | | | | | | | |
| --- | --- | C6 | Multiple Switches ON | Two or more software switches were ON simultaneously, or a second software switch was turned ON before a prior operation was completed. | • The error code will be displayed on the 7-segment display for 30 seconds.<br>• The error display will be cleared the next time that a settings operation is completed normally. **_MultiSwOnErr* (Multiple Switches ON Error) will change to TRUE. | --- | Execute software switch operations one at a time. |
| --- | --- | d5 | Verification Error (target non-existent) | The target registered in the device parameters does not exist. | The Unit will periodically attempt to reconnect to the target.<br>**_TDLOpnErr* (Verification Error), **_UnitErr* (Unit Error Occurred), and **_NetErr* (Network Error Occurred) will change to TRUE. | --- | Check the following items:<br>• Is the registered node's power supply ON?<br>• Is the cable connected?<br>• Is the cable damaged or loose?<br>• Are CIP message communications stopped at the target node or originator?<br>• Are CIP message communications permitted for packet filtering for the target node and devices in the communications path?<br>• Is there excessive noise? |

* The 7-segment display alternates between the node address and error code.

| Indicator | | | Error | Cause | Unit operation | Error log (hex) | Countermeasure |
|---|---|---|---|---|---|---|---|
| MS | NS | 7-segment* | | | | | |
| **Errors Related to the Tag Data Links** | | | | | | | |
| --- | --- | d6 | Connection Failed | The connection could not be established because device parameters (such as the variable name and size) did not match in the originator and target, or connection resources are insufficient. | The Unit will periodically attempt to reconnect to the target.<br>*_*TDLOpnErr* (Verification Error), *_*UnitErr* (Unit Error Occurred), and *_*NetErr* (Network Error Occurred) will change to TRUE. | 03D4 | Correct the device parameter settings, and download the device parameters again from the Network Configurator. |
| --- | --- | d9 | Tag Data Link Error | A timeout occurred in the tag data link. (Tag data was not received from the target within the specified timeout time.) | The Unit will periodically attempt to reconnect to the target where the error occurred.<br>*_*TDLErr* (Tag Data Link Error), *_*UnitErr* (Unit Error Occurred), and *_*NetErr* (Network Error Occurred) will change to TRUE. | 03D5 | Check the following items:<br>• Is the registered node's power supply ON?<br>• Is the cable connected?<br>• Is the cable damaged or loose?<br>• Are CIP message communications stopped at the target node?<br>• Are CIP message communications permitted for packet filtering for the originator and devices in the communications path?<br>• Is there excessive noise? |

* The 7-segment display alternates between the node address and error code.

| Indicator | | | Error | Cause | Unit operation | Error log (hex) | Countermeasure |
|---|---|---|---|---|---|---|---|
| MS | NS | 7-segment* | | | | | |
| **Errors Related to Memory Access** | | | | | | | |
| Flashing red | --- | E9 | Memory Access Error | An error occurred in the Unit's non-volatile memory itself. This error will occur in the following cases.<br>1. An error occurred while writing the error log.<br>2. An error occurred while writing the device parameters.<br>**Note** This error does not indicate checksum errors detected when reading data. | Case 1:<br>The error record remains in RAM only. Subsequent writes to non-volatile memory are all ignored. Other than that, normal operation continues. (Error records continue to be written to RAM.)<br>Case 2:<br>Tag data links and message communications will continue operating.<br>*_*UnitErr* (Unit Error Occurred), *_*UnitMemErr* (Unit Memory Error), and *_*MemErr* (Nonvolatile Memory Error) will change to TRUE. | 0602 | Download the Unit Setup from the tab pages of the Edit Parameters Dialog Box of the Sysmac Studio and download the device parameters from the Network Configurator. If the error recurs, replace the EtherNet/IP Unit or the CPU Unit. |
| Flashing red | --- | E8 | Device Parameters Error | The I/O Area set in the device parameters does not exist in the CPU Unit.<br>A checksum error or logic error was detected in the parameters.<br>The Unit was mounted to a different PLC (e.g., from NJ to CJ2) after the Unit settings were made. | There is an error in the parameter settings stored in the Unit's non-volatile memory. (An error can occur when power is interrupted while data is being written to non-volatile memory.)<br>*_*UnitErr* (Unit Error Occurred) and *_*CommParamErr* (Invalid Communications Parameter) will change to TRUE. | 021A | Download the Unit Setup from the tab pages of the Edit Parameters Dialog Box of the Sysmac Studio and download the device parameters from the Network Configurator. If the error recurs, replace the EtherNet/IP Unit or the CPU Unit. |

* The 7-segment display alternates between the node address and error code.

| Indicator | | | Error | Cause | Unit operation | Error log (hex) | Countermeasure |
|---|---|---|---|---|---|---|---|
| MS | NS | 7-segment* | | | | | |
| **Errors Related to Memory Access** | | | | | | | |
| Flashing red | --- | EA | IP Advanced Settings Error | | There is an error in the parameter settings stored in the Unit's non-volatile memory. (An error can occur when power is interrupted while data is being written to non-volatile memory.)<br>The Unit Error Occurred Flag (n+10, bit 00) and Invalid Communications Parameter Flag (n+12, bit 04) will go ON. | 03D1 | Identify the error log data, correct the settings, and then download the Unit Setup from the tab pages of the Edit Parameters Dialog Box of the Sysmac Studio. |
| Flashing red | --- | F2 | Ethernet Basic Settings Error | | --- | 03D0 | Download the settings from the TCP/IP or Ethernet Tab Pages of the Edit Parameters Dialog Box of the Sysmac Studio or download the TCP/IP settings from the Network Configurator. |
| --- | --- | E1 | Ethernet Link Not Detected | The link with the Ethernet switch could not be detected.<br>**Note** This error will not occur when data links are not set. | • The Unit will be offline and unable to communicate. Errors will be returned to all communications requests.<br>• Data exchanges (refreshing) will continue with the CPU Unit.<br>**_UnitErr* (Unit Error Occurred), **_NetErr* (Network Error Occurred), and **_LkOffErr* (Link OFF Error) will change to TRUE. **_LkSta* (Link Status) will change to FALSE. | 03D3 | Check the following items:<br>• Is the cable connected?<br>• Is the cable damaged or loose?<br>• Is there excessive noise? |

* The 7-segment display alternates between the node address and error code.

| Indicator | | | Error | Cause | Unit operation | Error log (hex) | Countermeasure |
|---|---|---|---|---|---|---|---|
| MS | NS | 7-segment* | | | | | |
| **Errors Related to Memory Access** | | | | | | | |
| --- | --- | E3 | Server Connection Error | An error occurred in communications with the DNS server. | *_*DNSSvrErr* (DNS Server Error) will change to TRUE. | 03C4<br>Details: 00xx | Perform one of the following:<br>• Correct the DNS server settings.<br>• Check the communications path (EtherNet/IP Unit, cable connections, hubs, routers, and servers) and correct any problems. |
| **Errors Related to the Network** | | | | | | | |
| --- | --- | E3 | Server Connection Error | An error occurred with the BOOTP server.<br>1. There was no response from the BOOTP server.<br>2. The BOOTP server attempted to set an invalid IP address in the EtherNet/IP Unit. | Case 1:<br>The Unit will continue sending requests to the BOOTP server until there is a response. In the meantime, the Unit will be offline and unable to communicate. Errors will be returned to all communications requests. Data exchanges (refreshing) will continue with the CPU Unit.<br>Case 2:<br>The Unit will operate with the default IP address (192.168.250.1.node_ address).<br>*_*UnitErr* (Unit Error Occurred), *_*NetErr* (Network Error Occurred), and *_*BootpSvrErr* (BOOTP Server Error) will change to TRUE. | 03C4<br>Details: 06xx | Perform one of the following:<br>• Correct the BOOTP server settings.<br>• Check the communications path (EtherNet/IP Unit, cable connections, hubs, routers, and servers) and correct any problems. |

* The 7-segment display alternates between the node address and error code.

| Indicator | | | Error | Cause | Unit operation | Error log (hex) | Countermeasure |
|---|---|---|---|---|---|---|---|
| MS | NS | 7-segment* | | | | | |
| **Errors Related to the Network** | | | | | | | |
| --- | --- | E3 | Server Connection Error | An error occurred in communications with the STNP server. | **_SNTPSvrErr* (SNTP Server Error) will change to TRUE. | 03C4<br>Details: 03xx | Perform one of the following:<br>• Correct the SNTP server settings.<br>• Check the communications path (EtherNet/IP Unit, cable connections, hubs, routers, and servers) and correct any problems. |
| | | | | An error occurred in transmission to the SNMP trap. | --- | 03C4<br>Details: 07xx | Perform one of the following:<br>• Correct the SNMP trap settings.<br>• Check the communications path (EtherNet/IP Unit, cable connections, hubs, routers, and servers) and correct any problems. |
| --- | Lit red | F0 | IP Address Duplication | The IP address of the EtherNet/IP Unit is the same as the IP address set for another node. | • The Unit will be offline and unable to communicate. Errors will be returned to all communications requests.<br>• Data exchanges (refreshing) will continue with the CPU Unit.<br>**_UnitErr* (Unit Error Occurred), **_NetErr* (Network Error Occurred), and **_IPAdrDupErr* (IP Address Duplication Error) will change to TRUE. | 0211 | Check the IP addresses set on other nodes. Restart the EtherNet/IP Unit after correcting the IP address settings to eliminate duplications. |

* The 7-segment display alternates between the node address and error code.

| Indicator | | | Error | Cause | Unit operation | Error log (hex) | Countermeasure |
|---|---|---|---|---|---|---|---|
| MS | NS | 7-segment* | | | | | |
| **Errors Related to the Network** | | | | | | | |
| Flashing red | --- | F3 | Not lit | The target IP address conversion method is set to *Automatic generation*, but the last byte of the local IP address does not match the value set on the Node Address Setting Switch. | • Operation will continue with the set IP address as the local IP address. **_AdrMismatchErr* (Address Mismatch) will change to TRUE. | --- | Check the IP address and the Node Address Setting Switch setting. |
| Flashing red | Not lit | F4 | Communications Controller Error | An error occurred in the Communications Controller in the EtherNet/IP Unit. | • The Unit will be offline and unable to communicate. Errors will be returned to all communications requests.<br>• Data exchanges (refreshing) will continue with the CPU Unit.<br>**_UnitErr* (Unit Error Occurred), **_NetErr* (Network Error Occurred), and **_LANHwErr* (Communications Controller Error) will change to TRUE. | 020F | Replace the EtherNet/IP Unit or the CPU Unit if the error recurs when the Unit is restarted. |
| Flashing red | --- | C8 | Node Address Setting Changed During Operation | The Node Address Setting Switch was changed during operation. | Operation will continue.<br>**_IPAdrChgErr* (Operating IP Address Change) will change to TRUE. | --- | Restart the EtherNet/IP Unit after setting the correct node address. |
| **Errors Related to the Unit** | | | | | | | |
| Lit red | Not lit | --- | Special Unit Error | An error occurred in a Special I/O Unit or CPU Bus Unit. | Records the error in the error log.<br>Operation stops. | 0601 | Restart the CPU Unit.<br>Replace the EtherNet/IP Unit or the CPU Unit if the error recurs. |

* The 7-segment display alternates between the node address and error code.

# 15-3 Connection Status Codes and Error Processing

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. For details, refer to *15-1 Checking Status with the Network Configurator*.

The following table shows the possible originator/target configurations.

| | Originator | Target |
|---|---|---|
| Configuration 1 | CS1W-EIP21/EIP21S, CJ1W-EIP21/EIP21S, CJ2H-CPU□□-EIP, CJ2M-CPU3□, NJ501-□□□□, NJ301-□□□□, and NJ101-□□□□ | CS1W-EIP21/EIP21S, CJ1W-EIP21/EIP21S, CJ2H-CPU□□-EIP, CJ2M-CPU3□, NJ501-□□□□, NJ301-□□□□, and NJ101-□□□□ |
| Configuration 2 | CS1W-EIP21/EIP21S, CJ1W-EIP21/EIP21S, CJ2H-CPU□□-EIP, CJ2M-CPU3□, NJ501-□□□□, NJ301-□□□□, and NJ101-□□□□ | Other company's device |
| Configuration 3 | Other company's device | CS1W-EIP21/EIP21S, CJ1W-EIP21/EIP21S, CJ2H-CPU□□-EIP, CJ2M-CPU3□, NJ501-□□□□, NJ301-□□□□, and NJ101-□□□□ |

The following table shows the likely causes of the errors causes for each configuration and connection status (code).

| Connection status | | Source of error | Handling | | |
|---|---|---|---|---|---|
| General Status (hex) | Additional Status (hex) | | Configuration 1 | Configuration 2 | Configuration 2 |
| 00 | 0000 | Normal status code:<br>The connection has been opened and the tag data link is communicating normally. | --- | --- | --- |
| 01 | 0100 | Error code returned from target:<br>Attempted to open multiple connections at the same connection. | This error does not occur. | Depends on the target's specifications. (Contact the target device's manufacturer for details on preventing the error from occurring in the future.) | Depends on the originator's specifications. (Contact the originator device's manufacturer for details on preventing the error from occurring in the future.) |
| 01 | 0103 | Error code returned from target:<br>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:<br>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.<br>(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. |

| Connection status | | Source of error | Handling | | |
|---|---|---|---|---|---|
| General Status (hex) | Additional Status (hex) | | Configuration 1 | Configuration 2 | Configuration 2 |
| 01 | 0107 | Error code returned from target:<br>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:<br>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.)<br>Only multicast and point-to-point can be set. | Check which connection types can be used by the originator.<br>(An error will occur if a connection other than multicast or point-to-point is set.) |
| 01 | 0109 | Error code returned from target:<br>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:<br>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 originator. (Restart the tag data link communications with the software switch.) |
| 01 | 0111 | Error code returned from target:<br>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:<br>Attempted to open more connections than allowed by the specifications (CJ2M-EIP21: 32, other CPU Units: 256). | Check the connection settings (number of connections) at the originator and target. | Check the connection settings (number of connections) at the originator and target.<br>Check the connection specifications for another company's devices. | Check the connection settings (number of connections) at the originator and target.<br>Check the connection specifications for another company's devices. |
| 01 | 0114 | Error code returned from target:<br>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.)<br>Confirm 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 2 |
| 01 | 0115 | Error code returned from target:<br>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.)<br>Confirm that the target device's EDS file is correct. | Check the originator's connection settings. |
| 01 | 0116 | Error code returned from target:<br>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 EDS file and set it again. | Depends on the target's specifications. (Contact the target device's manufacturer.)<br>Confirm that the target device's EDS file is correct. | Check the originator's connection settings. |
| 01 | 0117 | Error code returned from target:<br>The tag set specified in the connection's target variables does not exist. | Check whether the originator and target tag sets and tags are set correctly.<br>Check the settings of the variables in the CPU Unit. | Depends on the target's specifications.<br>(Contact the target device's manufacturer.) | Check the originator's connection settings. Check whether the target's tag sets and tags are set correctly.<br>Check the settings of the variables in the CPU Unit. |
| 01 | 011A | Error code returned from originator:<br>Connection could not be established because the buffer was full due to high traffic. | An unexpected network load 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. | An unexpected network load 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. | Follow the operating specifications for the originator. (Consult the originator manufacturer.) |

| Connection status | | Source of error | Handling | | |
|---|---|---|---|---|---|
| General Status (hex) | Additional Status (hex) | | Configuration 1 | Configuration 2 | Configuration 2 |
| 01 | 011B | Error code returned from target:<br>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 0.5 ms or greater. |
| 01 | 0203 | Error code returned from target:<br>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 load, change the performance settings. For example, increase the timeout time or RPI setting.<br>Also, check whether CIP message communications of the target are stopped or CIP communications are permitted by the originator and Packet Filter function of the device on the route. | | |
| 01 | 0204 | Error code returned from target:<br>The connection-opening 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.<br>Also, check whether the CIP message communications of the target or originator are stopped and whether the CIP communications are permitted by Packet Filter function of the target device or the device on the route. | | |
| 01 | 0205 | Error code returned from target:<br>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 occurred at originator or error code returned from target:<br>The tag data link's allowable bandwidth (pps) was exceeded. | Check the originator and target connection settings (number of connections and RPI). | Check the target's connection settings (number of connections and RPI). | Check the originator and target connection settings (number of connections and RPI). |
| 01 | 0311 | Error code returned from target:<br>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:<br>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.) |

| Connection status | | Source of error | Handling | | |
|---|---|---|---|---|---|
| General Status (hex) | Additional Status (hex) | | Configuration 1 | Configuration 2 | Configuration 2 |
| 01 | 0315 | Error code returned from target:<br>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:<br>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 in originator:<br>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 a connection-opening request. | Depends on the originator's specifications. (Contact the originator device's manufacturer.) |
| 08 | --- | Error code returned from target:<br>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 in originator:<br>The connection operation is stopped. | The connection was stopped because the Tag Data Link Stop Bit was changed to TRUE, or the settings data is being downloaded.<br>Either change the Tag Data Link Start Bit to TRUE, or wait until the settings data has been downloaded.<br>Includes Controller stop errors, Unit failure. To handle these errors, refer to *15-2-1 Errors Occurring at the EtherNet/IP 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.) | 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 2 |
| D0 | 0002 | Error code generated in originator:<br>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.) |
| **Unique OMRON Error Codes** | | | | | |
| 01 | 0810 | Error code returned from target:<br>New data could not be obtained from the CPU Unit when opening connection. (The Unit will automatically retry, 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 PLC caused the PLC to stop.<br>If the task period was too long, the problem will be resolved automatically. If the PLC has stopped, identify and correct the error.<br>If the PLC system is stopped, identify the cause of the error from the CPU Unit error data. | 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 in originator:<br>New data could not be obtained from the CPU Unit when opening connection. (The Unit will automatically retry, and attempt to open the connection again.) | This error may occur if the CPU Unit's task period was long when opening the connection.<br>If the task period was too long, the problem will be resolved 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.) |

# 15-4 Error Log

Errors detected by the EtherNet/IP Unit are stored in the error log along with the date and time of their occurrence. The error log can be read from the Network Configurator and cleared from the Sysmac Studio.

## 15-4-1 Error Log Data

### Error Log Data Configuration

- **Error Descriptions**

  A short description of the error is given.

- **Detailed Error Code**

  The detailed error code provides further troubleshooting information on the error.

  Refer to *15-4-2 Error Log Error Codes* for the error descriptions and detailed error codes.

- **Clock Information**

  The year, month, day, hour, minutes, and sections that the error occurred are recorded.

**Additional Information**

The EtherNet/IP Unit reads and uses the clock information from the CPU Unit.

If the EtherNet/IP Unit cannot read the clock information from the CPU Unit, the clock time in the error log will be all zeros.

The correct time is not recorded if the internal clock is not set correctly. If such an error record is read, the clock information will be invalid.

### Storage Location of Error Log Data

- **Error Log Data in RAM**

  When an error occurs, one record is stored in the error log table in RAM inside the EtherNet/IP Unit. Up to 64 records are recorded.

- **Error Log Data in Non-volatile Memory**

  When a high-priority error occurs, a record is stored both in the error log table in RAM and in the error log table in non-volatile memory in the EtherNet/IP Unit. The contents of the error log table in non-volatile memory is retained even if the power supply to the Controller is turned OFF or the Controller is reset. (When the Controller is reset, the error log table in the non-volatile memory is automatically loaded to the error log table in RAM.) One record is stored in the non-volatile memory for each error. Up to 32 records are stored. When the maximum number of records is exceeded, the oldest records are deleted and only the newest records are retained. (The maximum number of records is 64 for RAM and 32 for the non-volatile memory.)

- **Clearing Error Log Data**

  You can use the Controller event log functions of the Sysmac Studio to clear the error log data from the RAM or non-volatile memory in the EtherNet/IP Unit.

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for the functions and operating procedures for the Controller event logs.

## 15-4-2 Error Log Error Codes

| Error code (hex) | Error | Detail code | | Saved in non-volatile memory |
|---|---|---|---|---|
| | | First byte | Second byte | |
| 0001 | CPU Unit watchdog timer error | 00 hex | 00 hex | Yes |
| 0002 | CPU Unit service monitoring error | Monitoring time (ms) | | Yes |
| 0006 | Other CPU error | Bit D11: Unit not in Registered I/O Tables<br>(Other bits are reserved for system use.) | | Yes |
| 000E | I/O bus error | 00 hex | 00 hex | Yes |
| 000F | CPU Unit initialization error | 00 hex | 00 hex | Yes |
| 0011 | Event timed out | MRC (main command) | SRC (subcommand) | Yes |
| 0012 | CPU Unit memory error | 01 hex: Read error<br>02 hex: Write error | 03 hex: Routing tables<br>05 hex: CPU Bus Unit Area (CIO or DM) | Yes |
| 0015 | CPU Unit fatal error | 00 hex | 00 hex | Yes |
| 0017 | Tag database error | 00 hex | 00 hex | Yes |
| 0103 | Resend count exceeded (send failed) | CIP Frame:<br>FFFF | | No |
| 0105 | Node address setting error (send failed) | | | No |
| 0107 | Remote node not in network (send failed) | | | No |
| 0108 | No Unit with specified unit address (send failed) | | | No |
| 010B | CPU Unit error (send failed) | | | No |
| 010D | Destination address not in routing tables (send failed) | | | No |
| 010E | Not registered in routing tables (send failed) | | | No |
| 010F | Routing table error (send failed) | | | No |
| 0110 | Too many relay points (send failed) | | | No |
| 0111 | Command too long (send failed) | | | No |
| 0112 | Header error (send failed) | | | No |
| 0117 | Internal buffers full; packet discarded | | | No |
| 0118 | Illegal packet discarded | | | No |
| 0119 | Local node busy (send failed) | | | No |
| 0120 | Unexpected routing error | | | No |
| 0122 | Service not supported in current mode; packet discarded | | | No |
| 0123 | Internal send buffer full; packet discarded | | | No |
| 0124 | Maximum frame size exceeded; routing failed | | | No |
| 0125 | Response timeout; packet discarded | | | No |
| 020F | Communications controller error | 00 hex | 01 hex | Yes |
| 0211 | IP address duplication | Port number (always 02) | Lower eight bits of IP address | Yes |

| Error code (hex) | Error | Detail code | | Saved in non-volatile memory |
|---|---|---|---|---|
| | | First byte | Second byte | |
| 021A | Logic error in setting table | 00 hex | 02 hex: Network parameters<br>03 hex: Routing tables<br>04 hex: Unit Setup<br>0E hex: Unit name<br>12 hex: Status area layout setting error<br>13 hex: Status area layout setting verification error<br>15 hex: Installation in a PLC of another series (e.g., from NJ to CJ2) after setting the Unit. | Yes |
| 0300 | Parameter error; packet discarded | CIP Frame:<br>FFFF | | No |
| 0347 | I/O refreshing error | 00 hex | 00 hex | Yes |
| 03C0 | FINS/TCP setting error | 01 to 10 hex:<br>Connection number | 01: Automatically allocated FINS node address duplication<br>02: Destination IP address error<br>03: Destination port number error | No |
| 03C1 | Server settings error | 00 hex: DNS<br>03 hex: SNTP<br>04 hex: FTP<br>06 hex: BOOTP<br>07 hex: SNMP<br>08 hex: SNMP Trap<br>09 hex: FINS/UDP<br>0A hex: FINS/TCP | 01: IP address<br>02: Host name<br>03: Port number<br>04: Other parameter | No |
| 03C2 | FINS/TCP packet discarded | 01 to 10 hex:<br>Connection number | 02 hex: Reopening because remote node closed<br>03 hex: Reopening because of reception error<br>04 hex: Reopening because of transmission error<br>05 hex: Reopening because RST received from remote node<br>06 hex: Reopening because of no keep-alive response<br>07 hex: Illegal FINS/TCP procedure<br>08 hex: Insufficient memory during server processing<br>09 hex: Insufficient memory during client processing<br>0A hex: Insufficient memory during node switching | No |

| Error code (hex) | Error | Detail code |  | Saved in non-volatile memory |
|---|---|---|---|---|
|  |  | First byte | Second byte |  |
| 03C3 | FINS/UDP packet discarded | 00 hex | 01 to FE hex:<br>Source node address | No |
| 03C4 | Server connection error | 00 hex: DNS<br>03 hex: SNTP<br>04 hex: FTP<br>06 hex: BOOTP<br>07 hex: SNMP<br>08 hex: SNMP Trap | 01 hex: Specified host does not exist<br>02 hex: No such service at specified host<br>03 hex: Timeout<br>06 hex: Host name resolution error<br>07 hex: Transmission error<br>08 hex: Reception error<br>09 hex: Other error<br>0A hex: Obtaining IP address error | No |
| 03C6 | Clock write error | 0001: The clock time could not be updated because a error occurred in the CPU Unit. | Clear the error from the CPU Unit. | No |
|  |  | 0002: The clock time could not be updated because the CPU Unit or operating mode does not support this function. | Refer to Section *14 Communications Performance and Communications Load* and check the application conditions. |  |
| 03D0 | Ethernet basic setting error | 01 hex: Ethernet setting error | 01 hex: Checksum error<br>11 hex: Inconsistent settings<br>12 hex: Specified baud rate is not supported. | Yes |
|  |  | 02 hex: TCP/IP basic setting error | 01 hex: Checksum error<br>11 hex: Invalid IP address<br>12 hex: Invalid subnet mask<br>13 hex: Invalid default gateway address<br>14 hex: Invalid primary name server<br>15 hex: Invalid secondary name server<br>16 hex: Invalid domain name<br>17 hex: Invalid host name |  |

| Error code (hex) | Error | Detail code |  | Saved in non-volatile memory |
|---|---|---|---|---|
|  |  | First byte | Second byte |  |
| 03D1 | Ethernet advanced setting error | 02 hex: FINS setting error | 01 hex: Checksum error<br>10 hex: Invalid IP router table<br>11 hex: Invalid FINS/UDP setting<br>12 hex: Invalid FINS/TCP setting<br>13 hex: Invalid FTP setting<br>14 hex: Invalid SNTP setting<br>15 hex: Invalid SNMP setting<br>16 hex: Invalid SNMP trap setting | Yes |
| 03D2 | Packet discarded. | 01 hex | 00 hex | No |
| 03D3 | Link OFF error | 00 hex | 00 hex | No |
| 03D4 | Verification error (Tag data link only)<br>**Note** For details on identifying the cause of the verification error, refer to *15-3 Connection Status Codes and Error Processing*.<br>This error will not be stored when a target node is missing. | Connection instance number (1 to 255) | Lower eight bits of IP address | No |
| 03D5 | Tag data link error | 00 hex | Lower eight bits of IP address | No |
| 0601 | CPU Bus Unit error | Variable |  | Yes |
| 0602 | CPU Bus Unit memory error | 01: Read error<br>02: Write error | 02 hex: Network parameter<br>06 hex: Error log<br>09 hex: Identity data<br>0E hex: Unit name<br>0F hex: Ethernet basic setting<br>10 hex: Ethernet advanced setting<br>11 hex: MAC address<br>12 hex: Status area layout setting<br>14 hex: Term Tag address resolution memory write error | Yes * |

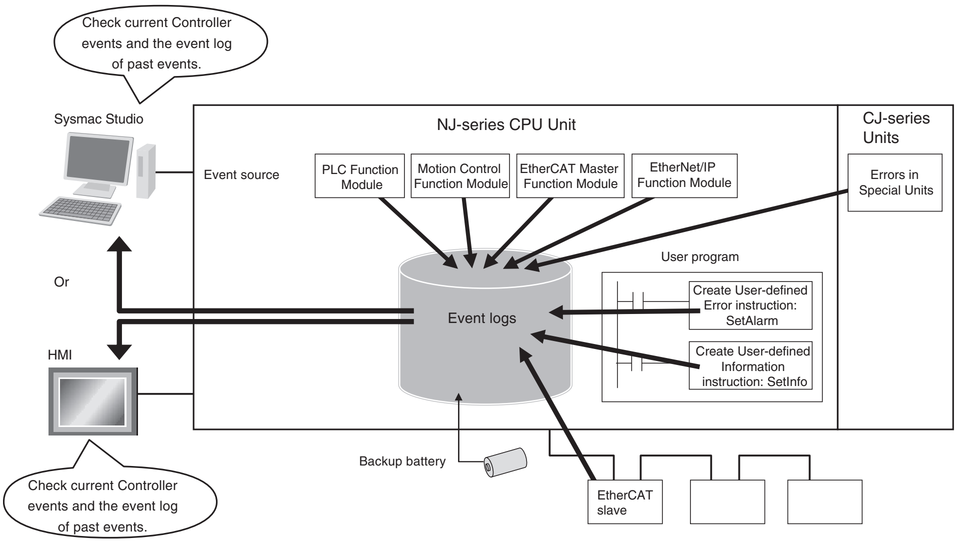
* If a memory error occurs in the error log area of the non-volatile memory, the record will not be stored in the non-volatile memory.

# 15-5 Event Logs

## 15-5-1 Overview of the Event Logs

You use the same methods to manage all of the events that occur on the NJ-series Controller. (The events include errors and information.) You can use the Sysmac Studio or an HMI to confirm current Controller events and the logs of events that have occurred. These logs are called event logs. Controller errors that occur for this Unit are also reported as events in the NJ-series CPU Unit.

Refer to the *NJ/NX-series CPU Unit Software User's Manual* (Cat. No. W501) for details on the event logs in an NJ-series CPU Unit. Refer to the *NJ/NX-series Troubleshooting Manual* (Cat. No. W503) for details on Controller errors, confirmation methods, and corrections.


Check current Controller events and the event log of past events.
Sysmac Studio
Or
HMI
Check current Controller events and the event log of past events.
NJ-series CPU Unit
CJ-series Units
Event source
PLC Function Module
Motion Control Function Module
EtherCAT Master Function Module
EtherNet/IP Function Module
Errors in Special Units
User program
Event logs
Create User-defined Error instruction: SetAlarm
Create User-defined Information instruction: SetInfo
Backup battery
EtherCAT slave

To use an HMI to check events, connect the HMI to the built-in EtherNet/IP port on the CPU Unit.

## 15-5-2 Error Table

The errors that may occur for this Unit are listed below.

The following abbreviations and symbols are used in the event level column.

| Abbreviation | Name |
|---|---|
| Maj | Major fault level |
| Prt | Partial fault level |
| Min | Minor fault level |
| Obs | Observation |
| Info | Information |

| Symbol | Meaning |
|---|---|
| S | Event levels that are defined by the system. |
| U | Event levels that can be changed by the user.* |

* This symbol appears only for events for which the user can change the event level.

A version in parentheses in the *Event code* column is the unit version of the CPU Unit when the event was added.

Refer to the *NJ/NX-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 | |
| 047A0000 hex | Unit Memory Error (Device Error) | An error occurred when writing to the error history or device parameters in non-volatile memory in the EtherNet/IP Unit. | • There is a source of noise nearby.<br>• Non-volatile memory failure | | | S | | | page 15-36 |
| 047B0000 hex | Non-volatile Memory Error | An error occurred in non-volatile memory. | • There is a source of noise nearby.<br>• Non-volatile memory failure | | | S | | | page 15-37 |
| 047C0000 hex | Communications Controller Error | An error occurred in the communications controller. | • Noise<br>• Communications Controller hardware error | | | S | | | page 15-37 |
| 14840000 hex | Invalid Communications Parameter | An error was found in the validation check of the parameters for tag data links that are saved in non-volatile memory. | • The power was interrupted during a download.<br>• A communications error occurred during a download.<br>• Non-volatile memory failure | | | S | | | page 15-38 |
| 14850000 hex | Tag Database Error | A tag database error occurred in the CPU Unit when using variables for tag data links, status layout, etc. | • The power was interrupted during a download.<br>• A communications error occurred during a download. | | | S | | | page 15-38 |
| 34A80000 hex | Verification Error | The information registered for a target node in the tag data link parameters is different from the actual node information. | • The specified target does not exist.<br>• Variable names do not match.<br>• The connection size is incorrect.<br>• Insufficient connection resources | | | S | | | page 15-39 |

| Event code | Event name | Meaning | Assumed cause | Level |  |  |  |  | Reference |
|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  | Maj | Prt | Min | Obs | Info |  |
| 34A90000 hex | Tag Data Link Error | There were two or more errors in a connection as an originator. The following are excluded. | • The power supply to the target node is OFF.<br>• Communications with the target node stop.<br>• The Ethernet cable for EtherNet/IP is disconnected.<br>• The Ethernet cable for EtherNet/IP is disconnected.<br>• Noise |  |  | S |  |  | page 15-40 |
| 34AA0000 hex | Tag Refresh Error | An unsupported data area or address range is specified for the tag data links. | • An unsupported data area or address range was specified for the tag data links. |  |  | S |  |  | page 15-40 |
| 34AB0000 hex | Basic Ethernet Setting Error | There is an illegal TCP/IP setting. | • The power was interrupted during a download.<br>• A communications error occurred during a download. |  |  | S |  |  | page 15-41 |
| 34AC0000 hex | IP Address Table Error | The IP address table information is incorrect. | • The power was interrupted during a download.<br>• A communications error occurred during a download. |  |  | S |  |  | page 15-41 |
| 34AD0000 hex | IP Router Table Error | The IP router table information is incorrect. | • The power was interrupted during a download.<br>• A communications error occurred during a download. |  |  | S |  |  | page 15-42 |
| 34AE0000 hex | Routing Table Error | The routing table information is incorrect. | • The power was interrupted during a download.<br>• A communications error occurred during a download. |  |  | S |  |  | page 15-42 |
| 34AF0000 hex | Ethernet Advanced Setting Error | There is an illegal FINS setting. | • The power was interrupted during a download.<br>• A communications error occurred during a download. |  |  | S |  |  | page 15-43 |
| 34B00000 hex | Address Mismatch | The host ID of the local IP address is inconsistent with the FINS node address. Or, the last segment of the local IP address is inconsistent with the setting on the node address switches. | • The IP address conversion method is set to automatic generation, but the host ID of the local IP address is inconsistent with the FINS node address or the last segment of the local IP address is inconsistent with the setting on the node address switch. |  |  | S |  |  | page 15-43 |
| 381C0000 hex | Status Area Layout Setting Error | An error occurred in the layout setting of the EtherNet/IP Unit. | • There is an error in the layout settings of the EtherNet/IP Unit. |  |  | S |  |  | page 15-44 |
| 54AE0000 hex | Multiple Switches ON Error | More than one software switch changed to TRUE at the same time. | • More than one software switch changed to TRUE at the same time. Or, another software switch changed to TRUE before processing was completed for a previous software switch. |  |  | S | U |  | page 15-44 |
| 84E00000 hex | IP Address Duplication Error | The same IP address is used more than once. | • The IP address of the EtherNet/IP port is also used as the IP address of another node. |  |  | S |  |  | page 15-45 |

| Event code | Event name | Meaning | Assumed cause | Level | | | | | Reference |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Maj | Prt | Min | Obs | Info | |
| 84E10000 hex | BOOTP Server Error | Connection with the BOOTP server failed. | • Server setting error (The acquired IP address is illegal.)<br>• Server is down.<br>• An error occurred in the communications path. | | | S | | | page 15-46 |
| 54AF0000 hex | Access Detected Outside Range of Variable | Accessing a value that is out of range was detected for a tag variable that is used in a tag data link. | • An out-of-range value was written by an EtherNet/IP tag data link for a variable with a specified range. A value that does not specify an enumerator was written by an EtherNet/IP tag data link for an enumeration variable. | | | | S | | page 15-46 |
| 84E20000 hex | Link OFF Error | The Ethernet link status turned OFF. | • The Ethernet cable is disconnected.<br>• An Ethernet cable is disconnected or loose.<br>• The switching hub power supply is turned OFF.<br>• Baud rate mismatch.<br>• Noise | | | U | S | | page 15-47 |

## 15-5-3 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.

<table>
<tr><td>Event name</td><td colspan="3">Gives the name of the error.</td><td>Event code</td><td colspan="2">Gives the code of the error.</td></tr>
<tr><td>Meaning</td><td colspan="6">Gives a short description of the error.</td></tr>
<tr><td>Source</td><td colspan="2">Gives the source of the error.</td><td>Source details</td><td>Gives details on the source of the error.</td><td>Detection timing</td><td>Tells when the error is detected.</td></tr>
<tr><td>Error attributes</td><td>Level</td><td>Tells the level of influence on control.*1</td><td>Recovery</td><td>Gives the recovery method.*2</td><td>Log category</td><td>Tells which log the error is saved in.*3</td></tr>
<tr><td>Effects</td><td>User program</td><td>Tells what will happen to execution of the user program.*4</td><td>Operation</td><td colspan="3">Provides special information on the operation that results from the error (event).</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="6">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.</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="6">Lists the possible causes, corrections, and preventive measures for the error.</td></tr>
<tr><td>Attached information</td><td colspan="6">This is the attached information that is displayed by the Sysmac Studio or an HMI</td></tr>
<tr><td>Precautions/ Remarks</td><td colspan="6">Provides precautions, restrictions, and supplemental information. If the user can set the event level, the event levels that can be set, the recovery method, operational information, and other information are also provided.</td></tr>
</table>

*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.

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Unit Memory Error (Device Error) | | | Event code | 047A0000 hex | |
| Meaning | An error occurred when writing to the error history or device parameters in non-volatile memory in the EtherNet/IP Unit. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | At power ON, Controller reset, or Unit restart |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | The following operations will be performed depending on the type of error in the non-volatile memory. (You can check the error history in the EtherNet/IP Unit and the CIO Area status.)<br>02 hex (network parameters): Tag data links stop.<br>06 hex (error history): The error history is not saved in non-volatile memory.<br>09 hex (identity information): EtherNet/IP communications stop.<br>0E hex (Unit name): No affect on other communications.<br>0F hex (basic Ethernet settings): Communications are not possible.<br>10 hex (advanced Ethernet settings): The relevant function stops.<br>11 hex (MAC address): Ethernet communications stop.<br>12 hex (status area layout settings): Operation is performed using default layout settings.<br>14 hex (Tag address resolve memory write error): Tag data link communications stops. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | There is a source of noise nearby. | | If device parameter writing fails, download the device parameters or connection settings using the Network Configurator or the Sysmac Studio. Then, if the error persists even after cycling the power supply or restarting the EtherNet/IP Unit, check for ambient noise. If there is a noise source, remove it. | | Use the Unit away from sources of noise. Implement noise countermeasures. | |
| | Non-volatile memory failure | | If the error occurs even after performing the actions described above, replace the EtherNet/IP Unit. | | None | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Non-volatile Memory Error | | | Event code | 047B0000 hex | |
| Meaning | An error occurred in non-volatile memory. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | Continuously |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | The Unit will stop if this error is detected in self-diagnosis when the power supply is turned ON. Operation continues if the error occurs during operation. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | There is a source of noise nearby. | | Remove any nearby sources of noise. | | Use the Unit away from sources of noise. Implement noise countermeasures. | |
| | Non-volatile memory failure | | If the above cause does not apply, replace the Unit. | | None | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Communications Controller Error | | | Event code | 047C0000 hex | |
| Meaning | An error occurred in the communications controller. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | After link is established |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | The EtherNet/IP Unit is offline. Communications are not possible. Error responses are returned to all communications requests. Data refreshing with the CPU Unit continues. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Noise | | Cycle the power supply to the Controller. Implement noise countermeasures. | | Use the Unit away from sources of noise. Implement noise countermeasures. | |
| | Communications Controller hardware error | | If the error occurs even after performing the actions described above, replace the EtherNet/IP Unit. | | None | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Invalid Communications Parameter | | | Event code | 1484 0000 hex | |
| Meaning | An error was found in the validation check of the parameters for tag data links that are saved in non-volatile memory. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | At power ON, Controller reset, or Unit restart |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | Tag link function of the EtherNet/IP Unit cannot be used. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The power was interrupted during a download. | | Download the device parameters or connection settings from the Network Configurator or the Sysmac Studio again. | | Make sure that the power supply is not interrupted during a download. | |
| | A communications error occurred during a download. | | Check for ambient noise. If there is a noise source, remove it. | | Use the Unit away from sources of noise. Implement noise countermeasures. | |
| | Non-volatile memory failure | | If the error occurs even after performing the actions described above, replace the EtherNet/IP Unit. | | None | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Tag Database Error | | | Event code | 1485 0000 hex | |
| Meaning | A tag database error occurred in the CPU Unit when using variables for tag data links, status layout, etc. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | At power ON, Controller reset, or Unit restart |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | Tag data link function of the EtherNet/IP Unit cannot be used. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The power was interrupted during a download. | | Download the settings from the Sysmac Studio again. | | Make sure that the power supply is not interrupted during a download. | |
| | A communications error occurred during a download. | | Implement noise countermeasures if there is excessive noise. | | Implement noise countermeasures. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Verification Error | | | Event code | 34A80000 hex | |
| Meaning | The information registered for a target node in the tag data link parameters is different from the actual node information. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | After link is established |
| Error attributes | Level | Minor fault | Recovery | Automatic recovery | Log category | System |
| Effects | User program | Continues. | Operation | Reconnection processing is periodically repeated for the target. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The specified target does not exist. | | If the power supply for nodes with registered connections is not ON, turn ON the power supply. Correct the cable if it is disconnected or loose. Implement noise countermeasures if there is excessive noise. | | Check that the power supply for connection nodes is turned ON. Connect the cable securely. Implement noise countermeasures if there is excessive noise. | |
| | Variable names do not match. | | Correct the data link parameters or change the connection nodes so that the data link parameters match the actual node information. | | Make the settings so that the data link parameters match the actual node information. | |
| | The connection size is incorrect. | | | | | |
| | Insufficient connection resources | | Reduce the number of class-3 messages that are being used. | | Reduce the number of data links and class-3 messages that are used. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | Identify the target node in the detailed information of the error history. | | | | | |

15-5 Event Logs

15

15-5-3 Error Descriptions

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Tag Data Link Error | | | Event code | 34A90000 hex | |
| Meaning | There were two or more errors in a connection as an originator. The following are excluded.<br>• Connections as a target<br>• Connection timeouts due to a Link OFF Error with the switching hub | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | After link is established |
| 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. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| 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. | | Make sure that the target node starts normally. | |
| | 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 disconnected. | | Replace the Ethernet cable. | | None | |
| | Noise | | Implement noise countermeasures if there is excessive noise. | | Implement noise countermeasures if there is excessive noise. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Tag Refresh Error | | | Event code | 34AA0000 hex | |
| Meaning | An unsupported data area or address range is specified for the tag data links. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | Continuously |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | Tag data link function of the EtherNet/IP Unit cannot be used. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | An unsupported data area or address range was specified for the tag data links. | | Correct the tag data links so that the area types and address ranges are within the specified ranges. | | Set the tag data links so that the area types and address ranges are within the specified ranges. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Basic Ethernet Setting Error | | | Event code | 34AB0000 hex | |
| Meaning | There is an illegal TCP/IP setting. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | At power ON, Controller reset, or Unit restart |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | Ethernet communications for the EtherNet/IP Unit are not possible. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The power was interrupted during a download. | | Download the settings from the Sysmac Studio again. | | Make sure that the power supply is not interrupted during a download. | |
| | A communications error occurred during a download. | | Implement noise countermeasures if there is excessive noise. | | Implement noise countermeasures. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | IP Address Table Error | | | Event code | 34AC0000 hex | |
| Meaning | The IP address table information is incorrect. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | At power ON, Controller reset, or Unit restart |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | Ethernet communications for the EtherNet/IP Unit are not possible. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The power was interrupted during a download. | | Download the settings from the Sysmac Studio again. | | Make sure that the power supply is not interrupted during a download. | |
| | A communications error occurred during a download. | | Implement noise countermeasures if there is excessive noise. | | Implement noise countermeasures. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | IP Router Table Error | | | Event code | 34AD0000 hex | |
| Meaning | The IP router table information is incorrect. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | At power ON, Controller reset, or Unit restart |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | Ethernet communications for the EtherNet/IP Unit are not possible. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The power was interrupted during a download. | | Download the settings from the Sysmac Studio again. | | Make sure that the power supply is not interrupted during a download. | |
| | A communications error occurred during a download. | | Implement noise countermeasures if there is excessive noise. | | Implement noise countermeasures. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Routing Table Error | | | Event code | 34AE0000 hex | |
| Meaning | The routing table information is incorrect. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | At power ON, Controller reset, or Unit restart |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | Ethernet communications for the EtherNet/IP Unit are not possible. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The power was interrupted during a download. | | Download the settings from the Sysmac Studio again. | | Make sure that the power supply is not interrupted during a download. | |
| | A communications error occurred during a download. | | Implement noise countermeasures if there is excessive noise. | | Implement noise countermeasures. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Ethernet Advanced Setting Error | | | Event code | 34AF0000 hex | |
| Meaning | There is an illegal FINS setting. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | At power ON, Controller reset, or Unit restart |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | The relevant function stops. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The power was interrupted during a download. | | Identify the cause of the error from the detailed information in the error history of EtherNet/IP Unit, and then correct the settings. | | Make sure that the power supply is not interrupted during a download. | |
| | A communications error occurred during a download. | | Implement noise countermeasures if there is excessive noise. | | Implement noise countermeasures. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Address Mismatch | | | Event code | 34B00000 hex | |
| Meaning | The host ID of the local IP address is inconsistent with the FINS node address. Or, the last segment of the local IP address is inconsistent with the setting on the node address switches. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | At power ON, Controller reset, or Unit restart |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | FINS communications are not possible. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The IP address conversion method is set to automatic generation, but the host ID of the local IP address is inconsistent with the FINS node address or the last segment of the local IP address is inconsistent with the setting on the node address switch. | | Set the IP address or node address to match. | | None | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Status Area Layout Setting Error | | | Event code | 381C0000 hex | |
| Meaning | An error occurred in the layout setting of the EtherNet/IP Unit. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | At power ON, Controller reset, or Unit restart |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | Operation will be performed with the default layout settings while this error exists. In the following case, however, operation is performed with the user-set layout settings.<br>• A non-existent area is specified. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | There is an error in the layout settings of the EtherNet/IP Unit. | | Correct the layout settings of the EtherNet/IP Unit. | | Check that there are no errors in the layout settings of the EtherNet/IP Unit. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Multiple Switches ON Error | | | Event code | 54AE0000 hex | |
| Meaning | More than one software switch changed to TRUE at the same time. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | At software switch operation |
| Error attributes | Level | Minor fault | Recovery | Automatic recovery | Log category | System |
| Effects | User program | Continues. | Operation | Not affected. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | More than one software switch changed to TRUE at the same time. Or, another software switch changed to TRUE before processing was completed for a previous software switch. | | Do not allow more than one software switch to be TRUE at the same time. | | Do not allow more than one software switch to be TRUE at the same time. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | You can change the event level to the observation level. If you change the level to the observation level, recovery procedures are not required. | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | IP Address Duplication Error | | | Event code | 84E00000 hex | |
| Meaning | The same IP address is used more than once. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | After determining IP addresses |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | EtherNet/IP communications are not possible. Error responses are returned to all communications requests. Data refreshing with the PLC Function Module will continue. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The IP address of the EtherNet/IP port is also used as the IP address of another node. | | Perform either of the following and then cycle the power supply to the Controller or reset the Controller.<br>• 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.<br>• Remove the node that has the duplicate IP address from the network. | | Perform allocations so that IP addresses of nodes on the network are used for only one node. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | BOOTP Server Error | | | Event code | 84E10000 hex | |
| Meaning | Connection with the BOOTP server failed. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | After link is established |
| Error attributes | Level | Minor fault | Recovery | Error reset | Log category | System |
| Effects | User program | Continues. | Operation | Requests to the BOOTP server will continue until there is a response from the BOOTP server. During that time, communications using BOOTP are not possible. Error responses are returned to all communications requests. Data refreshing with the PLC Function Module will continue. The IP address of the EtherNet/IP port that was specified by the BOOTP server was an illegal value. Operation is performed with the illegal default IP address (192.168.250.*node_address*). | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | Server setting error (The acquired IP address is illegal.) | | If an illegal value is set for the EtherNet/IP IP address in the server, correct it. | | Set the EtherNet/IP addresses in the server so that there are no illegal values. | |
| | 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 | None | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Access Detected Outside Range of Variable | | | Event code | 54AF0000 hex | |
| Meaning | Accessing a value that is out of range was detected for a tag variable that is used in a tag data link. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | When variable is written |
| 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 with a specified range. A value that does not specify an enumerator was written by an EtherNet/IP tag data link for an enumeration variable. | | Correct the value that is written to the variable with a specified range so that the value is in the range. Correct the value that is written to the enumeration variable so that the value specifies an enumerator. | | Write values that are in range for variables with specified ranges. Write values that specify enumerators to enumeration variables. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | • Write operations for out-of-range values or values that do not specify enumerators do not end normally.<br>• Write operations for in-range values or values that specify enumerators end normally. | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| Event name | Link OFF Error | | | Event code | 84E20000 hex | |
| Meaning | The Ethernet link status turned OFF. | | | | | |
| Source | PLC Function Module | | Source details | CJ-series Unit | Detection timing | After link is established |
| Error attributes | Level | Observation | Recovery | --- | Log category | System |
| Effects | User program | Continues. | Operation | The EtherNet/IP Unit is offline. Communications are not possible. Error responses are returned to all communications requests. Data refreshing with the CPU Unit continues. | | |
| System-defined variables | Variable | | Data type | | Name | |
| | None | | --- | | --- | |
| Cause and correction | Assumed cause | | Correction | | Prevention | |
| | The Ethernet cable is disconnected. | | Connect the Ethernet cable. | | Connect the Ethernet cable securely. Check the cable to make sure that it is not disconnected. | |
| | An Ethernet cable is disconnected or loose. | | Connect the Ethernet cable securely. If the cable is broken, replace it. | | | |
| | The switching hub power supply is turned OFF. | | Turn ON the power supply to the switching hub. Replace the switching hub if it fails. | | Do not turn OFF the power supply to the switching hub. | |
| | 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. | |
| Attached information | None | | | | | |
| Precautions/ Remarks | You can change the event level to the minor fault level. If you change the level to the minor fault level, the Recovery column above will be changed to “Automatic recovery.” | | | | | |

# 15-6 Troubleshooting

## 15-6-1 CPU Unit's ERR Lit or Flashing

Use the following table to troubleshoot the system when the CPU Unit's ERR indicator is lit or flashing when the EtherNet/IP Unit is mounted.

| | |
|---|---|
| An I/O verification error occurred. | • Confirm that the Unit is connected properly.<br>• Compare the Unit Configuration with the actual configuration of Units and correct the Unit Configuration. |
| A CPU Bus Unit setting error occurred. | • A Unit model in the Unit Configuration is different from the Unit model that is mounted in the Controller. Compare the Unit Configuration with the actual configuration of Units and correct the Unit Configuration. |
| A CPU Bus error occurred. | • Confirm that the Unit is connected properly.<br>• Restart the Unit. Replace the Unit if it doesn't restart. |
| An I/O Bus error occurred. | • Confirm that the Unit is connected properly.<br>• Restart the Unit. Replace the Unit if it doesn't restart. |

For details, refer to the CPU Unit's Operation Manual.

## 15-6-2 General Ethernet Problems

| | |
|---|---|
| The 100M and 10M Indicators on the EtherNet/IP Unit are both OFF. | • Confirm that the cable being used has the correct ratings.<br>• Confirm that the cable is properly connected to the Ethernet switch, and the Ethernet switch's power supply is ON. (The 7-segment display will indicate error E1.)<br>• If the Ethernet switch's settings can be changed, confirm that the Ethernet link settings are the same as the settings for the EtherNet/IP Unit. (For details, refer to *2-4 Network Installation*.) |
| The NS Indicator on the EtherNet/IP Unit is lit red. | • Check whether the same IP address is set on another node. (The 7-segment display will indicate error F0.) |

## 15-6-3 Tag Data Links Fail to Start*

* Use the following table to troubleshoot tag data links when the Tag Data Links Operating Information in Communications Status 1 does not change to TRUE.

| | |
|---|---|
| The indicators on the EtherNet/IP Unit are all OFF. | • Check whether power is being supplied to the Controller.<br>• Check whether the EtherNet/IP Unit is mounted in the CPU Rack correctly.<br>• If a watchdog timer (WDT) error has occurred in the CPU Unit, follow the procedures described in the CPU Unit's Operation Manual to correct the problem.<br>• All of the indicators for the EtherNet/IP Unit will be OFF if a CPU Bus Unit error has occurred. Check for a CPU Bus Unit error.<br>• Restart the Unit. Replace the Unit if it doesn't restart. |
| The MS indicator on the EtherNet/IP Unit is lit green, but the NS indicator remains OFF. | • If the EtherNet/IP Unit's 7-segment display is displaying an error code, refer to the tables in *15-2 Using the LED Indicators and Display for Troubleshooting*.<br>• Confirm that the cables are properly connected to the Ethernet switch and the power supply to the Ethernet switch is ON. |
| The MS indicator on the EtherNet/IP Unit is lit green, but the NS indicator continues to flash green. | • If the EtherNet/IP Unit's 7-segment display is displaying an error code, refer to the tables in *15-2 Using the LED Indicators and Display for Troubleshooting*.<br>• The NS indicator will continue to flash green if the tag data link settings have not been set in the Unit. Use the Network Configurator to set the tag data link settings in the Unit, and then restart the Unit. |
| The MS indicators is lit green on the EtherNet/IP Unit, but the NS indicator continues to flash red. | • Identify the error code shown on the 7-segment display based on the tables in *15-2 Using the LED Indicators and Display for Troubleshooting*, and eliminate the cause of the error. |

## 15-6-4 Tag Data Link Problems

| | |
|---|---|
| Tag data is not concurrent. | Check the following items and correct the user program.<br>• 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.<br>• Refer to the product manuals for products from other manufacturers. |
| At startup, some of the receive data is FALSE when it should be TRUE. | • When received data is used in the ladder program, use the All Tag Data Links Operating Information in Communications Status 1, or the corresponding element of the Target Node PLC Operating Flags as a condition. If the Target Node PLC Operating Flags is used, the Controller status must be included in tag sets of both the sending and receiving nodes.<br>• If the Output OFF function (Output Inhibit) is enabled in the output (produce) tag settings, all of the output data will be OFF if a fatal error occurs in the CPU Unit. Check the status of the output (producer) PLC. |
| The tag data links start and stop communicating intermittently. | • Check whether the baud rate is set to 10 Mbps, or a 10M or 100M repeater hub is being used. The performance of the tag data links assumes that an Ethernet switch is used to achieve a 12000-pps*[1] bandwidth for full-duplex, 100-Mbps auto-negotiation communications.<br>• Refer to *15-1 Checking Status with the Network Configurator* for details on checking the error counters on the Monitor Device Window's *Ethernet Information* Tab Page. The error and discarded packet counters indicate problems such as noise in the communications path, the use of substandard cables, damaged cables/connectors, loose connectors, abnormally high communications load, or incorrect wiring (loops) in the Ethernet switch wiring.<br>• Contact the Ethernet switch manufacturer to determine whether there are any problems with the transfer capacity of the Ethernet switches in the communications path. If Ethernet switches are arranged in a cascade connection, there may be a heavy load concentrated at a mid-level Ethernet switch. In the EtherNet/IP Unit itself, processing is performed with a higher priority than message communications, so specifications provide for a 12,000 pps bandwidth for other CPU Units in tag data link performance only.<br>• Refer to *15-1 Checking Status with the Network Configurator* for details on checking the connection status on the Monitor Device Window's *Connection* Tab Page. Eliminate any errors, which can be identified in the tables in *15-3 Connection Status Codes and Error Processing*. |

*1 For the Units with unit version 2.1 or earlier, this is 6,000 pps.

## 15-6-5 Message Timeout Problems

| | |
|---|---|
| Timeout errors occur frequently in message services (CIP UCMM, CIP Class 3, or FINS). | • When there is a high load in the tag data link, and the CPU Unit’s task period is relatively long or there are messages coming in from many nodes, the message service response time may be delayed and messages may be discarded occasionally.<br>• In this case, the communications load must be reduced by increasing (slowing) the tag data link’s RPI, reducing the message load, or increasing the timeout value.<br>• The tag data link’s bandwidth usage can be checked on the Monitor Device Window’s *Ethernet Information* Tab Page. Refer to *15-1 Checking Status with the Network Configurator* for details.<br>• The error log error codes that indicate discarded messages (insufficient memory) due to heavy communications loads are 0117, 0119, 0123, 0125, 03C2 (detail code □□08, □□09, or □□0A), 03C3, and 03D2. Refer to *15-1 Checking Status with the Network Configurator* for details on reading the error codes on the *Error History* Tab Page. |

# 15-7 Cleaning and Maintenance

This section describes daily maintenance and the cleaning and inspection methods.

## 15-7-1 Cleaning

Clean the EtherNet/IP Unit regularly as described below in order to keep it in optimal operating condition.

- Wipe the Unit daily with a dry, soft cloth.
- When a spot cannot be removed with a dry cloth, dampen the cloth with a neutral cleanser (2% solution), wring out the cloth, and wipe the Unit.
- A smudge may remain on the Unit from gum, vinyl, or tape that was left on for a long time. Remove the smudge when cleaning.

**Precautions for Correct Use**

Never use volatile solvents, such as paint thinner, benzene, or chemical wipes.

## 15-7-2 Inspection

Inspect the Unit regularly to keep it in its optimal operating condition.

In general, inspect the Unit once every 6 to 12 months. However, inspect it more frequently if the Unit is used at a high temperature or humidity or under dirty or dusty conditions.

### Equipment Required for Inspections

Prepare the following equipment before inspecting the Unit.

**● Equipment Required Daily**

- Phillips screwdriver and flat-blade screwdriver
- Voltage tester or digital voltmeter
- Industrial alcohol and clean cotton cloth

**● Equipment Required Occasionally**

- Synchronoscope
- Oscilloscope with pen recorder
- Thermometer and hygrometer (humidity meter)

## Daily Inspections

Check the items in the following table and correct any items that are below the standard. Improve the operating environment or adjust the EtherNet/IP Unit to correct the situation if any of the standards in the following table are not met.

| Daily inspection | Inspection | Criteria | Inspection method |
|---|---|---|---|
| Environmental conditions | Ambient and in-panel temperature | 0 to 55°C | Thermometer |
| | Ambient and in-panel humidity | 10% to 90% (with no condensation or icing) | Hygrometer |
| | Dust and dirt accumulation | No dust or dirt | Visual |
| Installation conditions | Is the Unit installed securely? | No looseness | |
| | Make sure that Ethernet cable connectors are fully inserted and locked. | No looseness | Phillips screwdriver |

# 15-8 Precautions on Equipment Replacement

If the EtherNet/IP Unit becomes faulty, it may affect the operation of the other nodes with which it communicates. Take immediate steps to recover operation as soon as possible.

We recommend that you keep a spare EtherNet/IP Unit available to restore network operation as quickly as possible.

## 15-8-1 Precautions When Replacing the EtherNet/IP Unit

Observe the following precautions when you replace the EtherNet/IP Unit.

- Turn OFF the power supply to the Controller before you replace the EtherNet/IP Unit.
- Check the new Unit to make sure that there are no errors.
- When you return an EtherNet/IP Unit for repair, attach a sheet of paper detailing the problem and return the Unit to your OMRON representative.

If there is a faulty contact, try wiping the contact with a clean cotton cloth dampened with alcohol. Remove any lint before you mount the Unit.

**Precautions for Correct Use**

To prevent electric shock when replacing the EtherNet/IP Unit, always stop communications in the network and turn OFF the power supply to all of the nodes before removing the Unit.

## 15-8-2 Settings Required after Unit Replacement

After you replace the EtherNet/IP Unit, make sure that the wiring and the following settings are the same as before the Unit was replaced.

- Node Address and Unit Number
  Set the rotary switches on the front of the Unit to same value as the Unit that was replaced.
- Configuration Data (Parameter Settings)
  The configuration data is saved in non-volatile memory in the EtherNet/IP Unit. Therefore, you must either transfer the configuration data from the previous Unit to the computer or reset the configuration data. Use the applicable Support Software.
  - Unit Settings (e.g., Local IP Address)
    Use the Sysmac Studio to transfer the data or to set the data again. Refer to *Section 5 Sysmac Studio Settings for the EtherNet/IP Unit*.
  - Parameter Settings (e.g., Parameters for Tag Data Links)
    Use the Network Configurator to transfer the data or to set the data again. Refer to *Section 7 Tag Data Link Functions*.

**Precautions for Correct Use**

When you replace the CPU Unit, start operation only after you transfer the data that was used before the replacement. Use the Sysmac Studio. Unexpected accidents may occur depending on the relation between the user program and the status of variables with a Retain attribute.

## 15-8-3 EtherNet/IP Unit Replacement Procedure

**1** Turn OFF the power supply to the Controller where the EtherNet/IP Unit to replace is mounted and to all external devices that are connected to the network.

**2** Remove the Ethernet cable from the EtherNet/IP Unit to replace and remove the EtherNet/IP Unit.

**3** Set the following hardware switches to the same values as the previous EtherNet/IP Unit.

- Node address and unit number switches

**4** Turn ON the power supply to the Controller where you replaced the EtherNet/IP Unit and to all of the external devices that are connected to the network.

**5** Check the indicators and status indications on the EtherNet/IP Unit, and make sure that the system is operating correctly.

App

# Appendices

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

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

| Item | EtherNet/IP Unit CS1W/CJ1W-EIP21S | EtherNet/IP Unit other than CS1W/CJ1W-EIP21S (built-in port on CJ2 CPU Unit) | | | Built-in Ether-Net/IP port on NJ-series CPU Unit | CJ-series Ethernet Unit |
|---|---|---|---|---|---|---|
| | Unit version 1.0 Lot number: 241001□ or later | Unit version 1.0 | Unit version 2.0 | Unit version 2.1 or later | | |
| Tag data link communications service | OK | OK | OK | OK | OK | --- |
| CIP message communications service | OK | OK | OK | OK | OK | --- |
| Socket service | OK | --- | --- | --- | OK | OK |
| File transfer (FTP) | OK | --- | OK | OK | OK | OK |
| Mail send/receive | --- | --- | --- | --- | --- | OK |
| Web functions | --- | --- | --- | --- | --- | OK |
| Automatic adjustment of PLC/Controller’s internal clock | OK | --- | OK | OK | OK | OK |
| Error history | OK | OK | OK | OK | OK[*1] | OK |
| Response to PING command | OK | OK | OK | OK | OK | OK |
| SNMP/SNMP trap | OK | --- | OK | OK | OK | --- |
| CIDR function for IP addresses | OK | --- | OK | OK | OK | --- |
| Online connection via EtherNet/IP using CX-One/Sysmac Studio | OK | --- | OK | OK | --- | --- |
| Online connection via EtherNet/IP using Network Configurator | OK | OK | OK | OK | OK | --- |
| Mounting in a Controller with an NJ-series CPU Unit | OK[*2 *3] | --- | --- | OK[*2 *3] | --- | --- |

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

*2 Only the CJ1W-EIP21/EIP21S can be mounted on NJ-series CPU Units.

*3 You cannot use the following functions if you connect to the CPU Unit through an EtherNet/IP Unit.

- Going online with a CPU Unit from the Sysmac Studio. (However, you can go online from the Network Configurator.)
- Troubleshooting from an HMI

# A-2 Use the Sysmac Studio to Set the Tag Data Links (EtherNet/IP Connections)

## A-2-1 Overview of the Tag Data Links (EtherNet/IP Connections) Settings with the Sysmac Studio

You can use the Sysmac Studio to set the settings required for creating tag data links (EtherNet/IP connections)[*1] between NJ-series Controllers.

*1 The tag data links and EtherNet/IP connections enable cyclic tag data exchanges on an EtherNet/IP network between Controllers or between Controllers and other devices. Here, "EtherNet/IP connection" refers to both the tag data links and the EtherNet/IP connections.

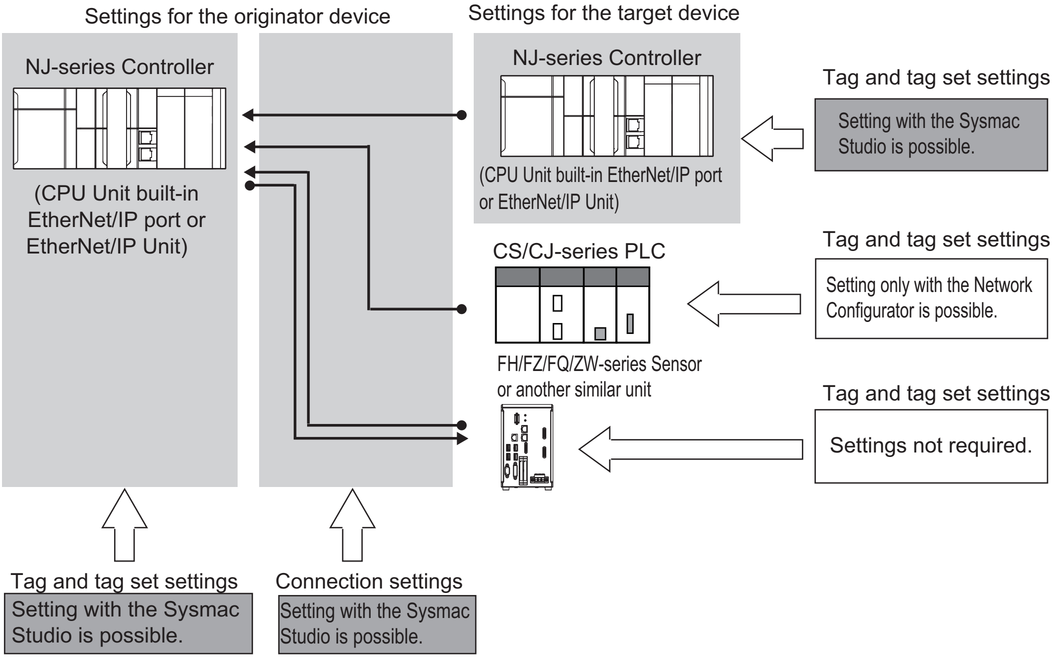
**Version Information**

Sysmac Studio version 1.10 or higher is required to use the Tag Data Link (EtherNet/IP Connection) Settings.

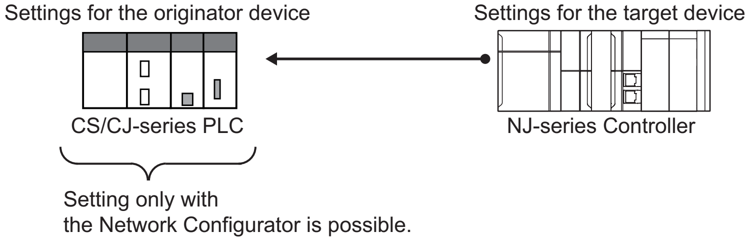
### Acceptable System Configuration Conditions for Setting the EtherNet/IP Connection Settings on the Sysmac Studio

If an NJ-series Controller operates as the originator device, you can use the Sysmac Studio to set the originator device settings for the EtherNet/IP connections.

Similarly, if an NJ-series Controller operates as the target device, you can use the Sysmac Studio to set the tags and tag sets of the target device.


Settings for the originator device
Settings for the target device
NJ-series Controller
(CPU Unit built-in EtherNet/IP port or EtherNet/IP Unit)
NJ-series Controller
(CPU Unit built-in EtherNet/IP port or EtherNet/IP Unit)
Tag and tag set settings
Setting with the Sysmac Studio is possible.
CS/CJ-series PLC
Tag and tag set settings
Setting only with the Network Configurator is possible.
FH/FZ/FQ/ZW-series Sensor or another similar unit
Tag and tag set settings
Settings not required.
Tag and tag set settings
Setting with the Sysmac Studio is possible.
Connection settings
Setting with the Sysmac Studio is possible.

Use the Network Configurator if a CS/CJ-series PLC operates as the originator device.


Settings for the originator device
Settings for the target device
CS/CJ-series PLC
NJ-series Controller
Setting only with
the Network Configurator is possible.

## A-2-2 Procedure to Make the EtherNet/IP Connection Settings with the Sysmac Studio

| | | Setup Window |
|---|---|---|
| **1** Registering devices<br>Register devices with which the EtherNet/IP connections are established to the project. | | • Main Window |
| **2** Creating network variables(*1) | ... Refer to *Registering the Network Variable for the Originator Device* on page A-12. | Global Variable Table on the Sysmac Studio |
| **3** Registering tags and tag sets<br>Register the network variables that are set in step 2 as tags and tag sets. | ... Refer to *Registering the Tag and Tag Set* on page A-14. | EtherNet/IP Connection Settings (Tag Set Display) |
| **4** Setting Connections<br>Specify devices (i.e., target devices and originator devices) and tag sets to communicate with using the EtherNet/IP connections. | ... Refer to *Setting Connections for the Originator Device* on page A-17. | EtherNet/IP Connection Settings (Connections Display) |
| **5** Going online from the Sysmac Studio | ... Refer to *Transferring the Connection Settings Data* on page A-29. | • Main Window |
| **6** Downloading EtherNet/IP connection settings<br>**Note** Connections automatically start after the download. | ... Refer to *Transferring the Connection Settings Data* on page A-29. | • Synchronization Window / Transfer to Controller Dialog Box<br>• EtherNet/IP Connection Settings |
| **7** Checking operation<br>Stopping and starting connections | ... Refer to *A-2-5 Checking Communications Status with the Sysmac Studio and Troubleshooting* on page A-34. | EtherNet/IP Connection Monitor Tab Page |

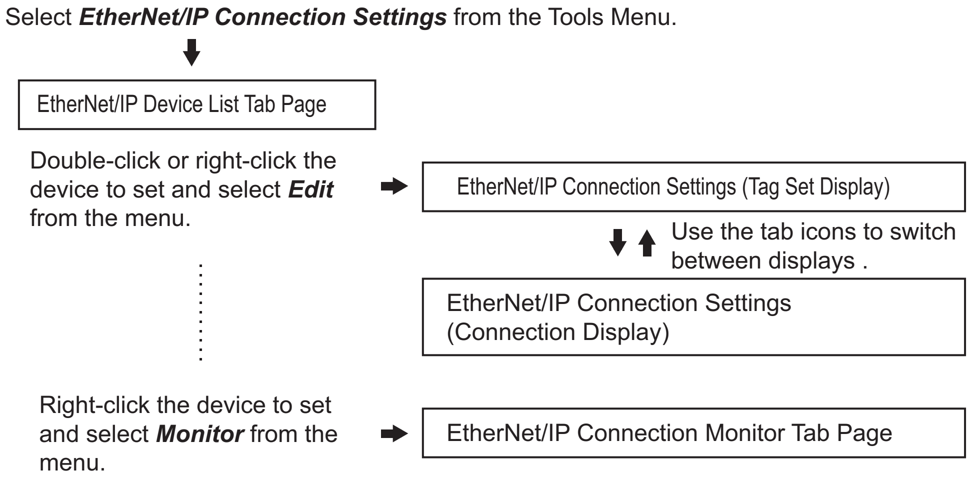
*1 Variables with its Network Publish attribute set to ***Output*** or ***Input*** in the Global Variable Table are called network variables.

## A-2-3 EtherNet/IP Connection Settings

This section describes the screen configuration for EtherNet/IP connection settings.

### ● Screen Transitions in the EtherNet/IP Connection Settings

- Connection Settings



Select ***EtherNet/IP Connection Settings*** from the Tools Menu.

EtherNet/IP Device List Tab Page

Double-click or right-click the device to set and select ***Edit*** from the menu.

EtherNet/IP Connection Settings (Tag Set Display)

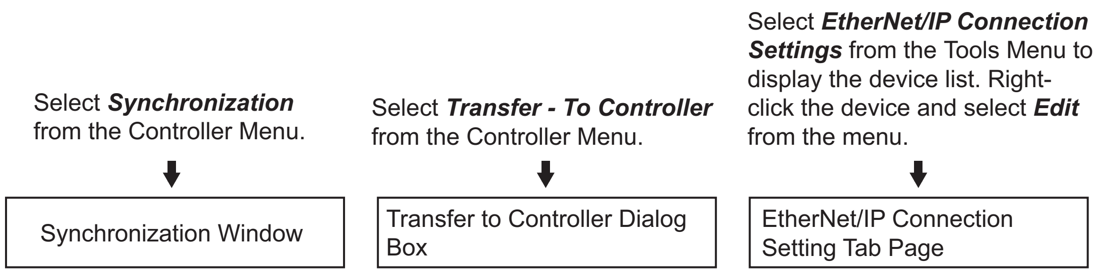
Use the tab icons to switch between displays .

EtherNet/IP Connection Settings (Connection Display)

Right-click the device to set and select ***Monitor*** from the menu.

EtherNet/IP Connection Monitor Tab Page

- Transferring connection settings to the Controller from the computer



Select ***Synchronization*** from the Controller Menu.

Synchronization Window

Select ***Transfer - To Controller*** from the Controller Menu.

Transfer to Controller Dialog Box

Select ***EtherNet/IP Connection Settings*** from the Tools Menu to display the device list. Right-click the device and select ***Edit*** from the menu.

EtherNet/IP Connection Setting Tab Page

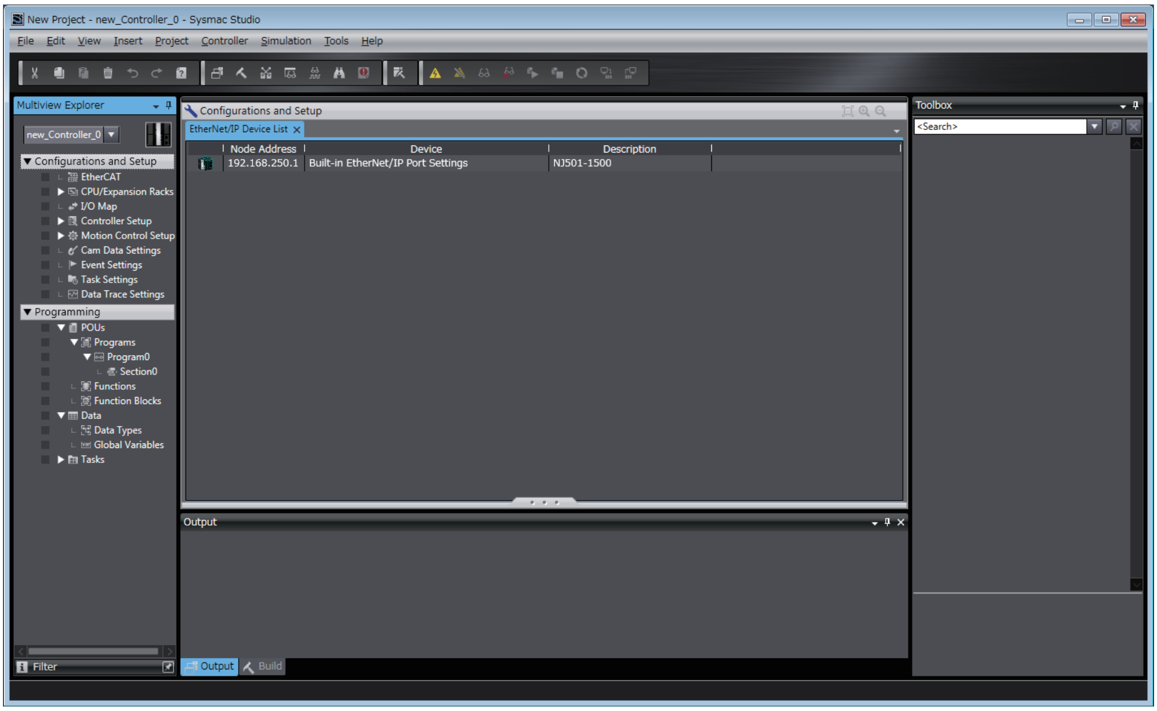
**Precautions for Correct Use**

Execute Transfer from the EtherNet/IP Connection Setting Tab Page to transfer only the connection settings.

Transferring part of a project with the Transfer to Controller Dialog Box or the Transfer from Controller Dialog Box is not possible. Therefore, only clearing the *Do not transfer the connection settings* Check Box will not transfer the connection settings if data in both the computer and the Controller are the same.

### ● EtherNet/IP Device List Tab Page (Refer to *Registering the Tag and Tag Set* on page A-14.)

The list indicates the devices to which EtherNet/IP connections can be set.



| Node Address | Device | Description |
|---|---|---|
| 192.168.250.1 | Built-in EtherNet/IP Port Settings | NJ501-1500 |

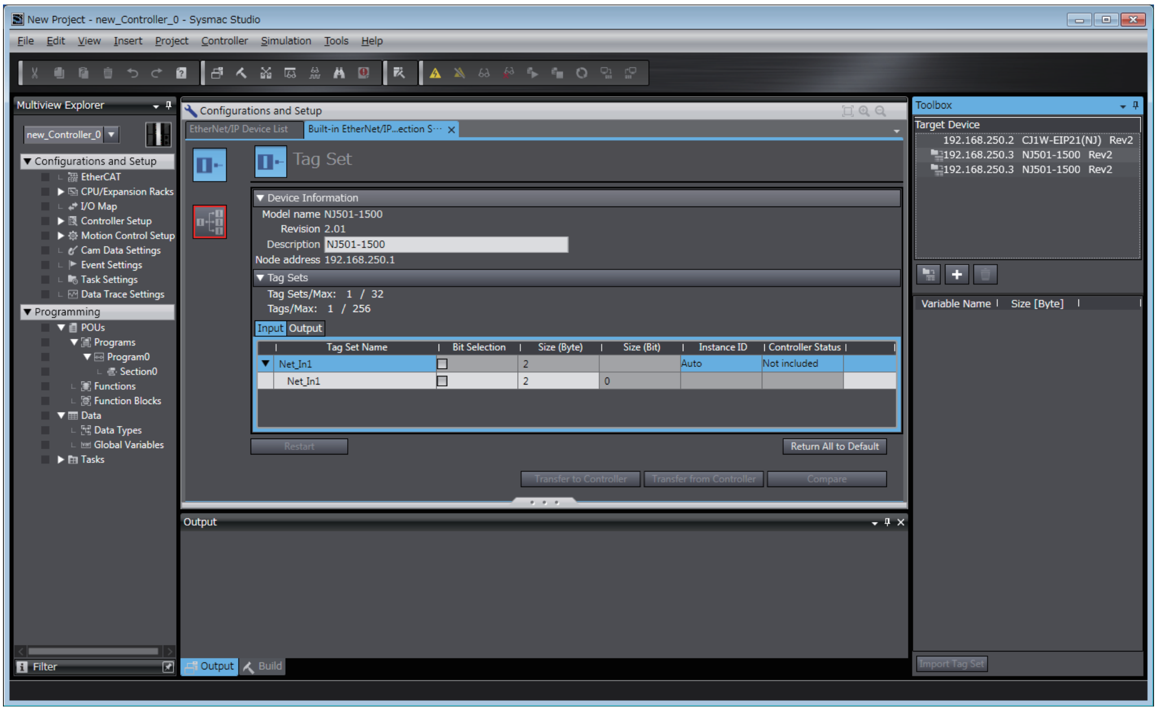
## ● EtherNet/IP Connection Settings (Tag Set Display) (Refer to *Registering the Tag and Tag Set* on page A-14.)

Register tag sets required to create connections. Each tag set represents the data that is sent and received through a connection. You can register up to eight tags in one tag set.

The name and size of the tag must be the same as those of the network variable[*1].

Set whether to include the Controller status information in tags for the tag sets. You can also set the data output operation at a fatal error occurrence for output tags.

[*1] Variables with its Network Publish attribute set to ***Output*** or ***Input*** in the Global Variable Table are called network variables.



Target Device:
192.168.250.2 CJ1W-EIP21(NJ) Rev2
192.168.250.3 NJ501-1500 Rev2
192.168.250.3 NJ501-1500 Rev2

Device Information
Model name NJ501-1500
Revision 2.01
Description NJ501-1500
Node address 192.168.250.1

Tag Sets
Tag Sets/Max: 1 / 32
Tags/Max: 1 / 256

Input Output

| Tag Set Name | Bit Selection | Size (Byte) | Size (Bit) | Instance ID | Controller Status |
|---|---|---|---|---|---|
| ▼ Net_In1 | □ | 2 | | Auto | Not included |
| Net_In1 | □ | 2 | 0 | | |

## ● EtherNet/IP Connection Settings (Connection Display) (Refer to *Setting Connections for the Originator Device* on page A-17.)

Specify the target devices and set their connections.

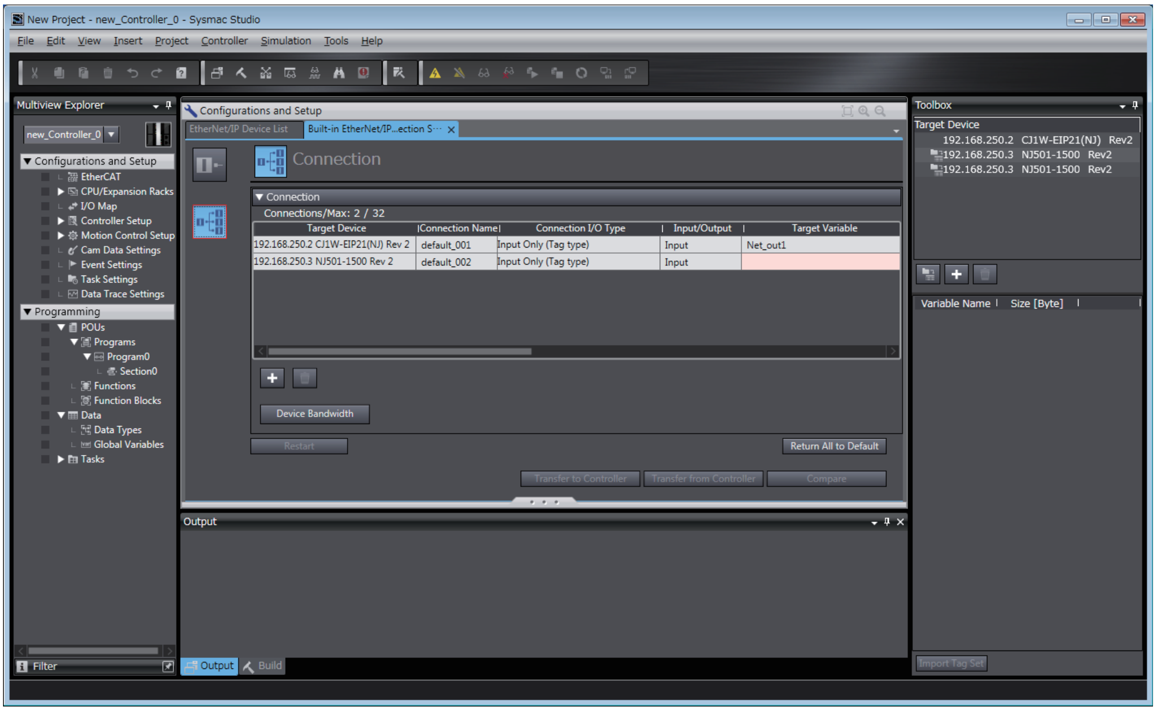
For each connection, set the following information: Connection Name, Connection I/O Type, I/O, target device tag set (target variable), originator device tag set (originator variable), Packet Interval (RPI), and Timeout Value.

### Precautions for Correct Use

If you changed the IP address, model or revision of the target device after making the connection settings, perform the following.

With the Sysmac Studio version 1.11 or higher, change the connection settings entirely.

With the Sysmac Studio version 1.10 or lower, create the connections again.



Connection

Connections/Max: 2 / 32

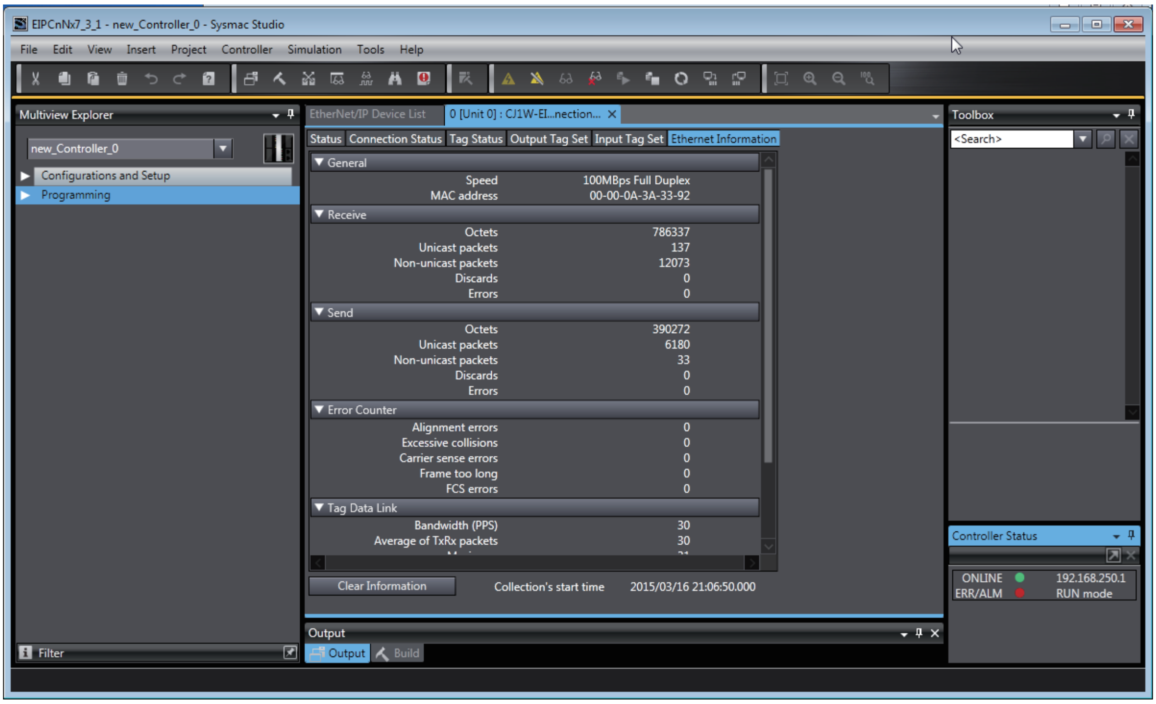
| Target Device | Connection Name | Connection I/O Type | Input/Output | Target Variable |
|---|---|---|---|---|
| 192.168.250.2 CJ1W-EIP21(NJ) Rev 2 | default_001 | Input Only (Tag type) | Input | Net_out1 |
| 192.168.250.3 NJ501-1500 Rev 2 | default_002 | Input Only (Tag type) | Input | |

Target Device

| | | |
|---|---|---|
| 192.168.250.2 | CJ1W-EIP21(NJ) | Rev2 |
| 192.168.250.3 | NJ501-1500 | Rev2 |
| 192.168.250.3 | NJ501-1500 | Rev2 |

- **EtherNet/IP Connection Monitor Tab Page (Refer to *A-2-5 Checking Communications Status with the Sysmac Studio and Troubleshooting* on page A-34.)**

You can check the EtherNet/IP connection setting status offline and communications status online. When online, you can start and stop connections.



▼ General

| | |
|---|---|
| Speed | 100MBps Full Duplex |
| MAC address | 00-00-0A-3A-33-92 |

▼ Receive

| | |
|---|---|
| Octets | 786337 |
| Unicast packets | 137 |
| Non-unicast packets | 12073 |
| Discards | 0 |
| Errors | 0 |

▼ Send

| | |
|---|---|
| Octets | 390272 |
| Unicast packets | 6180 |
| Non-unicast packets | 33 |
| Discards | 0 |
| Errors | 0 |

▼ Error Counter

| | |
|---|---|
| Alignment errors | 0 |
| Excessive collisions | 0 |
| Carrier sense errors | 0 |
| Frame too long | 0 |
| FCS errors | 0 |

▼ Tag Data Link

| | |
|---|---|
| Bandwidth (PPS) | 30 |
| Average of TxRx packets | 30 |

Collection's start time 2015/03/16 21:06:50.000

ONLINE ● 192.168.250.1
ERR/ALM ● RUN mode

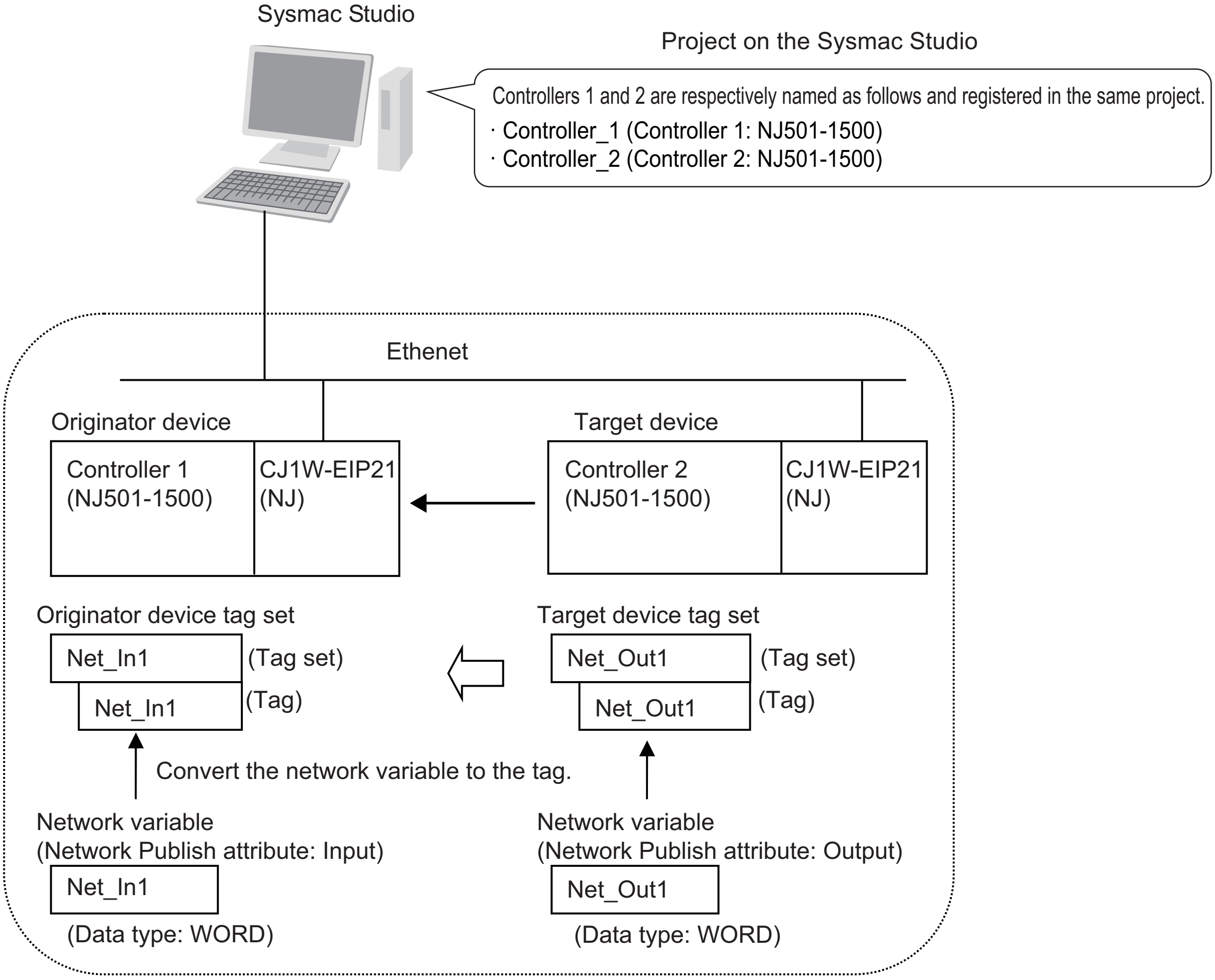
## A-2-4 Making the EtherNet/IP Connection Settings with the Sysmac Studio

This section describes the procedure to make the EtherNet/IP connection settings with the Sysmac Studio.

Here, we take the following system configuration as an example to describe how to set the EtherNet/IP connection settings.

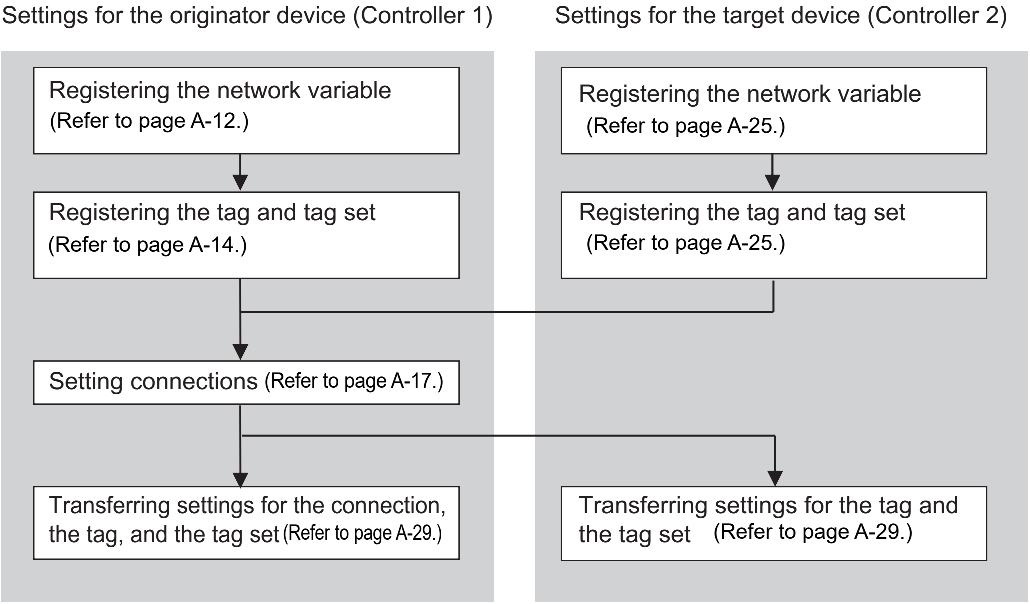
Example: System that connects the EtherNet/IP port on Controller 1 and the built-in EtherNet/IP port on Controller 2 via Ethernet

- Set the settings so that values in the network variable *Net_Out1* allocated for Controller 2 are sent to the network variable *Net_In1* allocated for Controller 1 at the set RPI of 50 ms cycle.
- This example assumes the programs for both Controllers 1 and 2 are registered in the same project.


Sysmac Studio
Project on the Sysmac Studio
Controllers 1 and 2 are respectively named as follows and registered in the same project.
· Controller_1 (Controller 1: NJ501-1500)
· Controller_2 (Controller 2: NJ501-1500)
Ethenet
Originator device
Controller 1
(NJ501-1500)
CJ1W-EIP21
(NJ)
Target device
Controller 2
(NJ501-1500)
CJ1W-EIP21
(NJ)
Originator device tag set
Net_In1
(Tag set)
Net_In1
(Tag)
Target device tag set
Net_Out1
(Tag set)
Net_Out1
(Tag)
Convert the network variable to the tag.
Network variable
(Network Publish attribute: Input)
Net_In1
(Data type: WORD)
Network variable
(Network Publish attribute: Output)
Net_Out1
(Data type: WORD)

Follow the flow below to set the settings to Controllers 1 and 2 for which to establish EtherNet/IP connections.

The required settings for the originator device and the target device are shown below.



Settings for the originator device (Controller 1)

Registering the network variable (Refer to page A-12.)

Registering the tag and tag set (Refer to page A-14.)

Setting connections (Refer to page A-17.)

Transferring settings for the connection, the tag, and the tag set (Refer to page A-29.)

Settings for the target device (Controller 2)

Registering the network variable (Refer to page A-25.)

Registering the tag and tag set (Refer to page A-25.)

Transferring settings for the tag and the tag set (Refer to page A-29.)

## Registering the Network Variable for the Originator Device

Register the network variable that is sent and received using the EtherNet/IP connections.

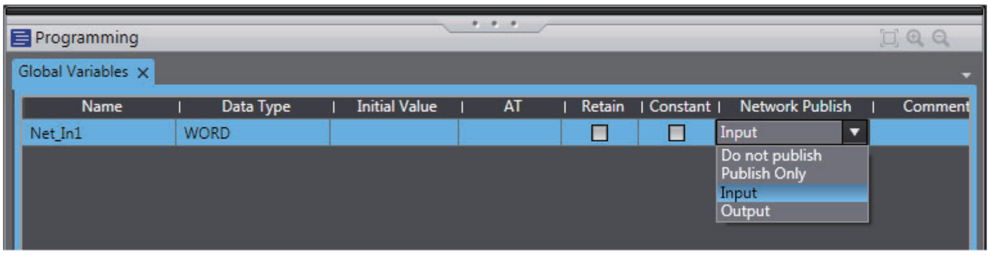
Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for the operations for registering variables.

**1** Assign the network variable to the tag used for the EtherNet/IP connection for Controller 1 (originator device).

This network variable receives data from Controller 2 (target device).

Set the Network Publish attribute to ***Input*** or ***Output*** in the Global Variable Table for the variable so that the variable serves as a network variable, i.e., the variable can be used for the EtherNet/IP connections.

In this example, set the network variable for Controller 1 as shown below.



| Name | Data Type | Initial Value | AT | Retain | Constant | Network Publish | Comment |
|---|---|---|---|---|---|---|---|
| Net_In1 | WORD | | | □ | □ | Input | |

- Variable name: Net_In1
- Data type: WORD
- Network Publish attribute: Input

## ● Network Variables Used for the EtherNet/IP Connections

- Network variable name

  You cannot specify an I/O memory address for a tag name in the EtherNet/IP connection settings. Thus, do not specify an I/O memory address for the network variable name that is to be assigned to a tag.

  The following text strings are recognized as the I/O memory address names.

  **(1) Variable names that contain only single-byte numerals from 0000 to 6143**

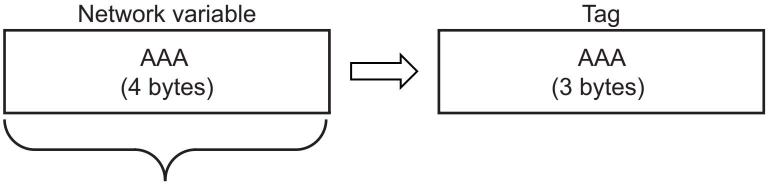
  **(2) Variable names with the following single-byte letters (uppercase or lowercase) followed by single-byte numerals**

  - H (H000 to H511)
  - W (W000 to W511)
  - D (D00000 to D32767)
  - E0_ to E18_ (E0_00000 to E0_32767, to E18_00000 to E18_32767)

  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.

- Size of variables

  To use an EtherNet/IP Unit as an EtherNet/IP device, set an even number of bytes for the size of the network variable used for the EtherNet/IP connections regardless of an odd number of bytes for the tag size.



The CPU Unit memory is consumed in units of two bytes. To assign tags of odd numbers of bytes
to network variables, specify even byte numbers (i.e., sizes of the tags + 1) to the network variables.

- Data concurrency

  To maintain concurrency in the values of network variables that are assigned to tags, you must set refreshing tasks.

  Refer to *7-1-7 Concurrency of Tag Data Link Data* for details.

## Registering the Tag and Tag Set

Register the required tag and tag set for the EtherNet/IP connections.

You can register tags and tag sets in the EtherNet/IP Connection Setting Tab Page.

### Precautions for Correct Use

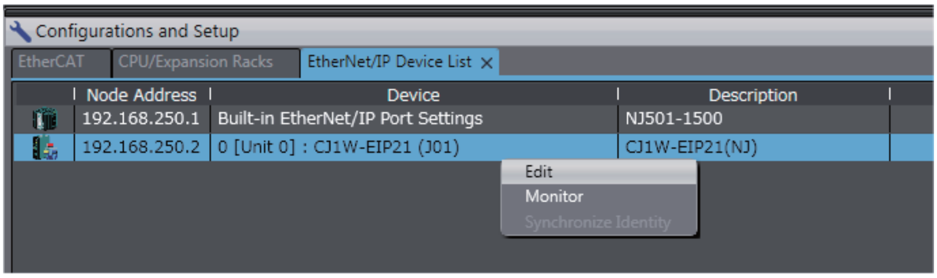
Make the following settings to refresh all of the tag data in the same tag set at the same time.

- Use the Sysmac Studio, in advance, to specify the same refreshing task for all of the variables that are assigned to tags in the tag set.
- 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.

**1** Select ***EtherNet/IP Connection Settings*** from the Tools Menu.

The EtherNet/IP Device List Tab Page is displayed.

**2** In this example, right click *CJ1W-EIP21* (the EtherNet/IP originator device controller 1) and select ***Edit*** from the menu to open the EtherNet/IP Connection Setting Tab Page.



| Node Address | Device | Description |
|---|---|---|
| 192.168.250.1 | Built-in EtherNet/IP Port Settings | NJ501-1500 |
| 192.168.250.2 | 0 [Unit 0] : CJ1W-EIP21 (J01) | CJ1W-EIP21(NJ) |

**3** Click the ▮▪ (Show Tag Set Display) icon in the EtherNet/IP Connection Setting Tab Page.

**4** Click the **Input** tab to switch to the Input Tab Page. Register the tag set and the tag.

Use one of the following methods to register the tag set and the tag.

- Independent registration : Manually registers network variables in the Controller as tags.
- Batch registration : Registers all network variables in the Controller as tags at the same time.

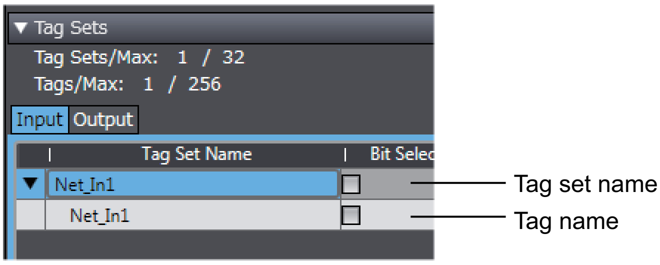
**5** Register tags and tag sets independently.

**(1) Right-click anywhere in the Input Tab Page of the EtherNet/IP Connection Setting Tab Page and select *Create New Tag Set* from the menu.**

**(2) Enter tag set name *Net_In1* directly into the list in the Input Tab Page.**

**(3) Right-click anywhere in the Input Tab Page and select *Create New Tag* from the menu.**

**(4) Enter tag name *Net_In1*.**



### Precautions for Correct Use

Any name can be specified for the tag set if the name matches one of the registered network variable names in the Controller.

As you enter characters (or immediately after you press the Ctrl + Space Keys), the Sysmac Studio Entry Assistance provides a list of variable names registered in the Controller. Select the variable name from the list.

### Additional Information

You can register up to 8 tags in a tag set.

Set as shown below to register multiple tags.

Example:

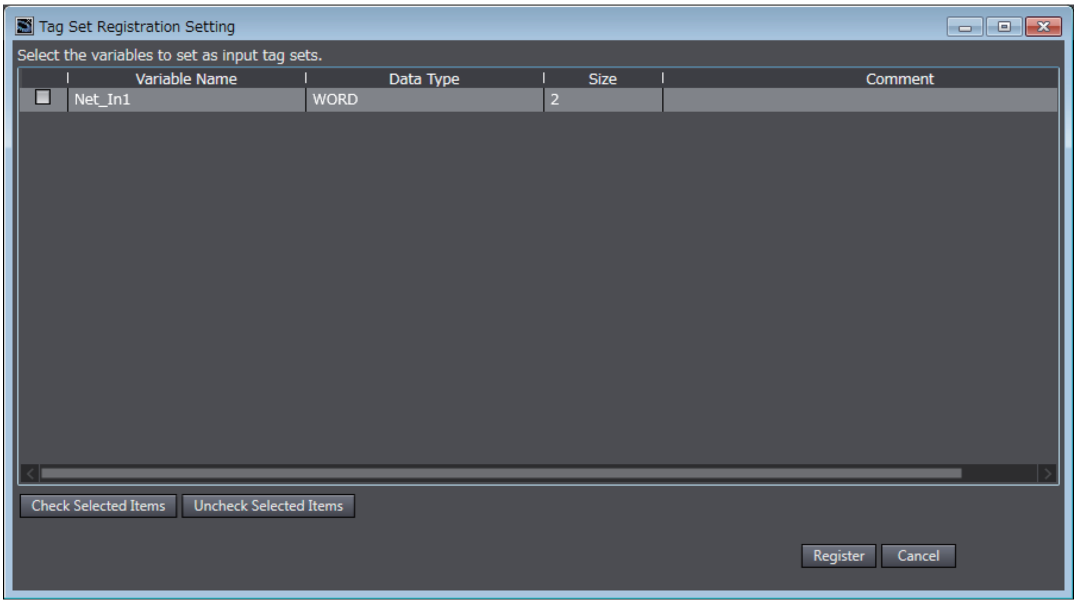
| | Tag set name | |
|---|---|---|
| ▼ | Network_Input_Value | .... (Tag set name) |
| | Net_In1 | .... (Tag name) |
| | Net_In2 | .... (Tag name) |

**6** Register all tags and tag sets at the same time.

**(1) Right-click anywhere in the Input Tab Page of the EtherNet/IP Connection Setting Tab Page and select *Register All Tag Sets* to display the Tag Set Registration Setting Dialog Box.**

This dialog box lists the following variables that are registered in the Global Variable Table.

- When registering all tag sets in the Input Tab Page:
  lists all variables that are published to the network as inputs.
- When registering all tag sets in the Output Tab Page:
  lists all variables that are published to the network as outputs.



Tag Set Registration Setting

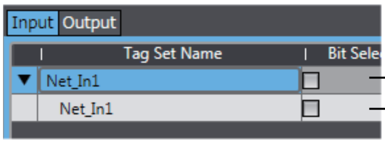
Select the variables to set as input tag sets.

| | Variable Name | Data Type | Size | Comment |
|---|---|---|---|---|
| □ | Net_In1 | WORD | 2 | |

**(2) Select the variable to register as a tag, and then click the Register Button.**

**(3) The automatically registered tag is added to the list in the EtherNet/IP Connection Setting Tab Page.**

With automatic registration, the tag is registered under a tag set having the same name as the tag, i.e., a single tag is registered in a single tag set.

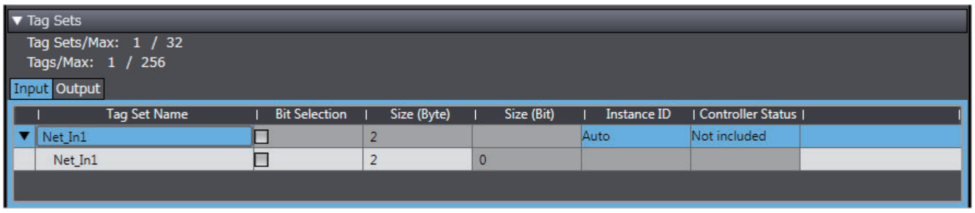

Input Output

| | Tag Set Name | Bit Sele |
|---|---|---|
| ▼ | Net_In1 | □ |
| | Net_In1 | □ |

Tag set name

Tag name

**7** Set the following settings for the registered tag and tag set.


▼ Tag Sets

Tag Sets/Max: 1 / 32

Tags/Max: 1 / 256

Input Output

| | Tag Set Name | Bit Selection | Size (Byte) | Size (Bit) | Instance ID | Controller Status |
|---|---|---|---|---|---|---|
| ▼ | Net_In1 | □ | 2 | | Auto | Not included |
| | Net_In1 | □ | 2 | 0 | | |

- Setting for Tag Sets

| Name | Item |
|---|---|
| Tag Set Name | Enter the tag set name.<br>You can change the names as required. |
| Size (Byte) | Gives the total size of the tag in bytes. |
| Instance ID | Gives the instance ID.<br>• Auto<br>• IN_{min}...IN_{max}<br>{min} represents the minimum number of Produced Assembly identification numbers recorded in the EDS files for the relevant devices.<br>{max} represents the maximum number of Produced Assembly identification numbers recorded in the EDS files for the relevant devices. |
| Controller Status | Specify whether to include the Controller status in the tag set. |

- Setting for Tags

| Name | Item |
|---|---|
| Tag Name | Enter the tag name.<br>Specify the tag name that matches one of the registered network variable names in the Controller. |
| Bit Selection | Specify whether to set the tag data size in bits.<br>Selected: Set the size in bits.<br>Not selected: Set the size in bytes. |
| Size (Byte) | Gives the size of the tag in bytes. |
| Size (Bit) | Gives the size of the tag in bits. |
| Output at Fatal Error | Specify whether to clear the output data or continue to send it when a major fault level Controller error occurs in the Controller.<br>• Retained<br>• Cleared |

## Setting Connections for the Originator Device

After the tag set registration, set the connection settings for transferring data using the EtherNet/IP connections.

Make the connection settings in the originator device (i.e., Controller 1 in this example) only.

Register the tag and tag set for Controller 2 (originator device) before setting the connection settings as described in this example.

Refer to *Registering the Tag and Tag Set for the Target Device* on page A-25 for the operations for registering tags and tag sets.

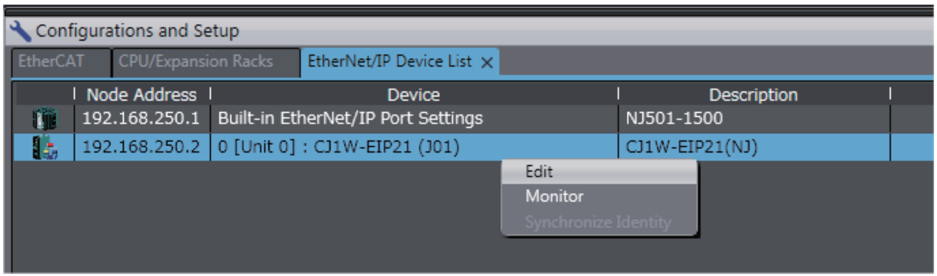
### Precautions for Correct Use

If you change the IP address, model, or revision of the target device after making the connection settings, you must also change the target device settings that are included in the connection settings. For information on how to change the target device settings in the connection settings, refer to *Changing the Target Device Settings after Making Connection Settings*.

**1** Select ***EtherNet/IP Connection Settings*** from the Tools Menu to display the EtherNet/IP Device List Tab Page.

**2** In this example, right click *CJ1W-EIP21* (the EtherNet/IP originator device controller 1) and select ***Edit*** from the menu to open the EtherNet/IP Connection Setting Tab Page.

The EtherNet/IP Connection Setting Tab Page is displayed.



| Node Address | Device | Description |
|---|---|---|
| 192.168.250.1 | Built-in EtherNet/IP Port Settings | NJ501-1500 |
| 192.168.250.2 | 0 [Unit 0] : CJ1W-EIP21 (J01) | CJ1W-EIP21(NJ) |

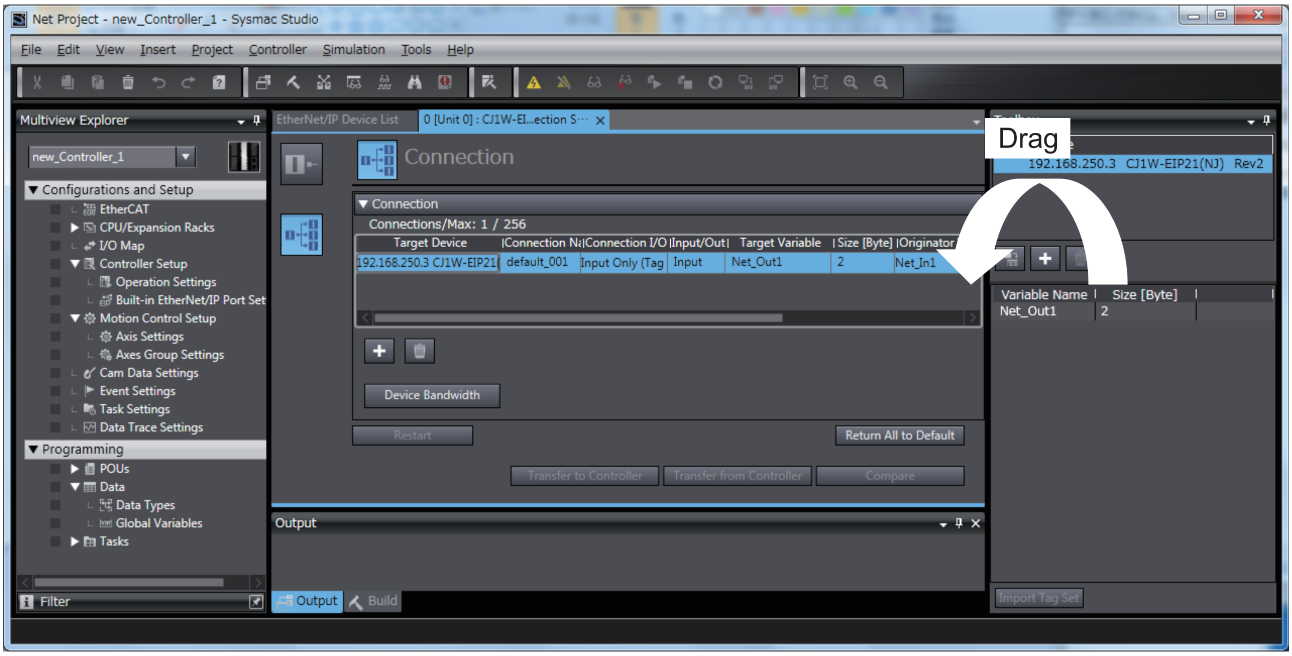
**3** Click the (Show Connection Display) icon in the EtherNet/IP Connection Setting Tab Page.

**4** Select **CJ1W-EIP21(NJ)** from **Target Device** in the Toolbox on the right of the tab page.

This operation displays the target device tag set *Net_Out1* that is set for Controller 2 in the **Variable Name** column.

**5** Drag the target device tag set *Net_Out1* in the **Variable Name** column of the Toolbox to the connection list.

As you enter characters (or immediately after you press the Ctrl + Space Keys), a list of target device variables that can be set for the connection is provided. Select the value from the list.



Connection

Connections/Max: 1 / 256

| Target Device | Connection Na | Connection I/O | Input/Out | Target Variable | Size [Byte] | Originator |
|---|---|---|---|---|---|---|
| 192.168.250.3 CJ1W-EIP21( | default_001 | Input Only (Tag | Input | Net_Out1 | 2 | Net_In1 |

192.168.250.3 CJ1W-EIP21(NJ) Rev2

| Variable Name | Size [Byte] |
|---|---|
| Net_Out1 | 2 |

**6** Specify **Originator Variable** and its **Size [Byte]** for the tag set *Net_Out1* added in step 5.

Here, specify *Net_In1* for **Originator Variable** and *2* for its **Size [Byte]**.

Change the other settings as required.

You can set the following items in the connection settings.

| Name | Setting Methods |
|---|---|
| Target Device | Select the target device. |
| Connection Name | Any name can be given to the connection (32 single-byte characters max.). |
| Connection I/O Type | *Input Only (Tag type)* is selected if EtherNet/IP connections are used with a CS1W-EIP21, CS1W-EIP21S, CJ1W-EIP21, CJ2B-EIP21, CJ2M-EIP21, CJ1W-EIP21(CJ2), CJ1W-EIP21(NJ), CJ1W-EIP21S, CJ1W-EIP21S(CJ2), CJ1W-EIP21S(NJ), NX701-□□□□, NJ501-□□□□, NJ301-□□□□, or NJ101-□□□□ CPU Unit.<br>When you create EtherNet/IP connections for other target devices, select the connection I/O type specified in that device's EDS file.<br>Use the **Input Only (ID type)** setting when another company's node is the originator and does not support connection settings with a Tag type setting. |
| Input/Output | The connection's input/output is automatically displayed based on the selected connection.<br>Input Only: Just **Input** is displayed. |
| Target Variable | Select the target node's tag set to assign it.<br>• **Input** is specified for **Input/Output**: Select the target's output (produce) tag set.<br>• **Output** is specified for **Input/Output**: Select the target's input (consume) tag set. |
| Size [Byte] | The data sizes of the target variables are displayed. |

| Name | Setting Methods |
|---|---|
| Originator Variable | Select the originator node's tag set to assign it.<br>• **Input** is specified for **Input/Output**: Select the originator's input (consume) tag set.<br>• **Output** is specified for **Input/Output**: Select the originator's output (produce) tag set. |
| Size [Byte] | Enter the data sizes of the originator variables. |
| Connection Type | Select whether the data is sent in multi-cast or unicast (point-to-point) form. The default setting is multi-cast.<br>• Multi-cast connection: Select when the same data is shared by multiple nodes. This setting is usually used.<br>• Point-to-point connection: Select when the same data is not shared by multiple nodes.<br>In a unicast transmission, other nodes are not burdened with an unnecessary load.<br>**Note** Refer to *7-1-4 Overview of Operation* for details on using multicast and unicast connections, and counting the number of connections. |
| RPI [ms] | Set the data update cycle (i.e., the packet interval) of each connection between the originator and target.<br>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.<br>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.<br>The default setting is RPI x 4.<br>The timeout value must be at least 10 ms. |

**7** The Toolbox displays the target devices if the devices are registered in the same Sysmac Studio project as where the originator devices are registered.

You can use one of the following methods to add unregistered devices in the same Sysmac Studio project as where the originator devices are registered to the Target Device List.

- Importing devices that are registered in another project

  You can import NJ-series Controllers registered in another project data and add them to the Device List.
- Registering devices using user-specified settings

  You can manually add target devices to the device list.

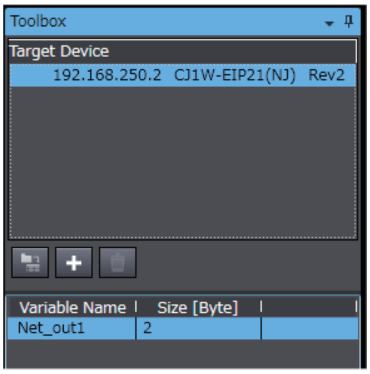
**Additional Information**

You can add target devices to the Device List by installing EDS files that include connection information for the devices in the Sysmac Studio and register the devices to the project.

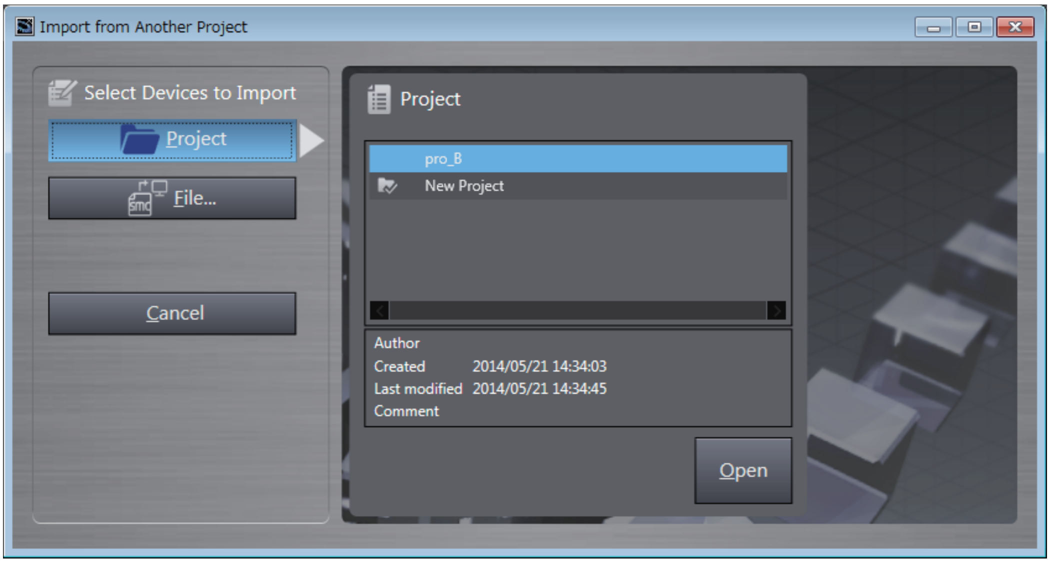
Refer to *Adding EDS Files* on page A-22 for details.

**8** Import devices that are registered in another project.

**(1) Click the (Import a device from another project) Button in the Toolbox on the right of the EtherNet/IP Connection Setting Tab Page.**



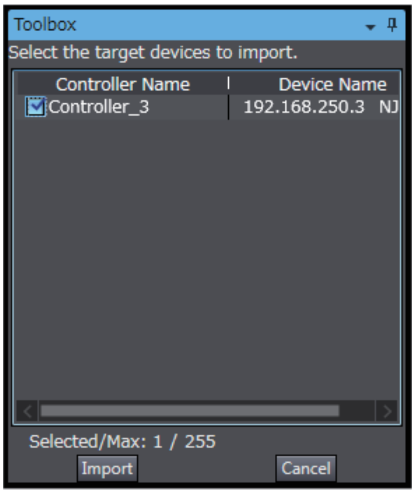
**(2) The Import from Another Project Dialog Box is displayed. Click the Project Button, select a project to import and click the Open Button.**


Import from Another Project
Select Devices to Import
Project
File...
Cancel
Project
pro_B
New Project
Author
Created 2014/05/21 14:34:03
Last modified 2014/05/21 14:34:45
Comment
Open

**(3) The list of EtherNet/IP devices registered in the selected project will be displayed. Select the target devices to import, and click the Import Button.**

**Note** Only the project for which the EtherNet/IP connection settings are set will be displayed.

The imported EtherNet/IP devices are added to the Target Device List in the Toolbox.

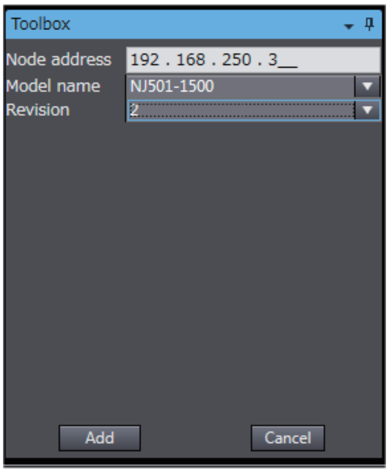


**9** Register devices as required.

**(1) Click the + Button under the Target Device List in the Toolbox.**

The Add Target Device Pane is displayed.

**(2) Enter relevant items for the target devices to add.**



Node address: 192 . 168 . 250 . 3__

Model name: NJ501-1500

Revision: 2

| Menu | Description |
|---|---|
| Node address | Enter the target device IP address. |
| Model name | Select the target device model. |
| Revision | Select the revision of the target device. |

**(3) Here, set the following items for Controller 3 and click the Add Button.**

The target device is added to the Target Device List in the Toolbox.

Node address: 192.168.250.3

Model name: NJ501-1500

Revision: 2

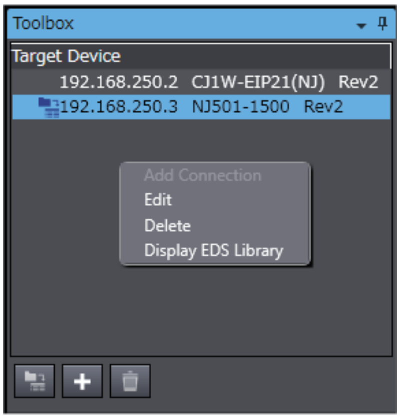
**(4) You can click the Import Tag Set Button to import the tag sets that are set in the Network Configurator to the target devices.**

Select ***Export to File*** from the **To/From File** Button in the Tag Sets Tab Page of the Edit Device Parameters Dialog Box to generate CSV files to import.

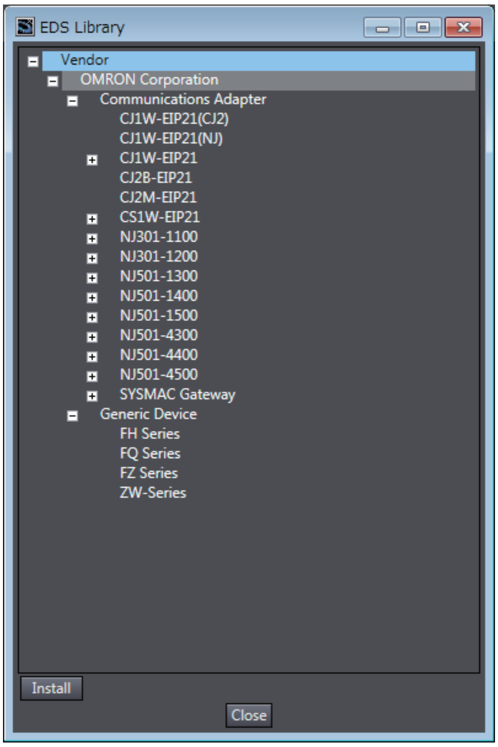
### ● Adding EDS Files

**Note** The Modular EDS device is supported by the Sysmac Studio version 1.11 or higher.

**1** Right-click anywhere in the Target Device List in the Toolbox of the EtherNet/IP Connection Setting Tab Page and select ***Display EDS Library*** from the menu.



**2** The EDS Library Dialog Box is displayed. Click the Install Button.



**3** Select the EDS file to add, and then click the **Open** Button.

The EDS file is added.

**4** The EtherNet/IP device with the EDS file installed is added to the EDS Library.

Devices listed in the EDS Library are used as a candidate device list when adding devices to the Target Device List in the Toolbox of the EtherNet/IP Connection Setting Tab Page.

### ● Changing the Target Device Settings after Making Connection Settings

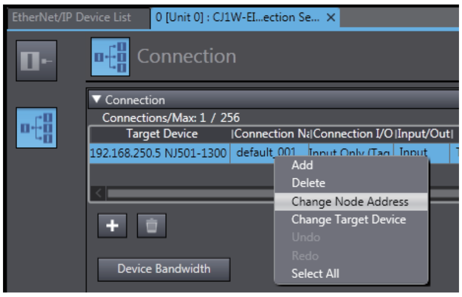
If you change the IP address, model, or revision of the target device after making the connection settings, you must also change the target device settings that are included in the connection settings. You can change the target device settings entirely.

**Precautions for Correct Use**

When you use the Sysmac Studio version 1.10 or lower, create the connections again if you changed the target device after configuring the connection settings.

## Changing the IP Addresses for All Target Devices

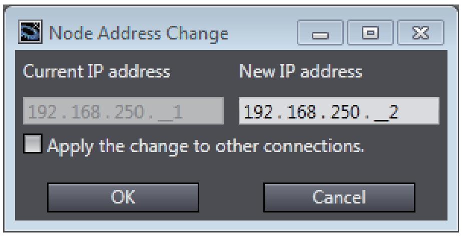
**1** Right-click one of the connection lines and select ***Change Node Address*** from the menu.



Connections/Max: 1 / 256

| Target Device | Connection Na | Connection I/O | Input/Out |
|---|---|---|---|
| 192.168.250.5 NJ501-1300 | default_001 | Input Only (Tag | Input |

**2** The Node Address Change Dialog Box is displayed. Enter a new IP address in *New IP address*.



Current IP address: 192 . 168 . 250 . _1

New IP address: 192 . 168 . 250 . _2

□ Apply the change to other connections.

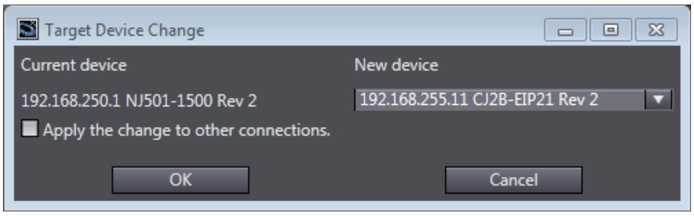
**3** To apply the same change to other connections, select the *Apply the change to other connections* Check Box.

**4** Click the **OK** Button.

## Changing All Target Device Information including Model Names and Revisions

**1** Right-click one of the connection lines and select ***Change Node Address*** from the menu.

**2** The Target Device Change Dialog Box is displayed. Select a target device from *New device*.



Current device: 192.168.250.1 NJ501-1500 Rev 2

New device: 192.168.255.11 CJ2B-EIP21 Rev 2 ▼

□ Apply the change to other connections.

**Precautions for Correct Use**

- Changeable target devices are limited to ones that have "OMRON" in the Vendor ID and is an EDS device of the Communications Adapter in the Device Type.
- To display a device in the list of selectable new target devices, the device must be registered as the target device in the Toolbox.

**3** To apply the same change to other connections, select the *Apply the change to other connections* Check Box.

**4** Click the **OK** Button.

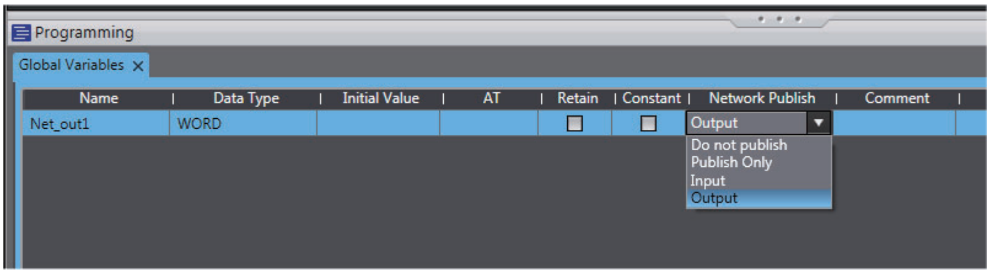
## Registering the Network Variable for the Target Device

**1** Assign the network variable to the tag used for the EtherNet/IP connection for Controller 2 (target device).

This network variable stores data to send to Controller 1 (originator device).

Set the Network Publish attribute to ***Input*** or ***Output*** in the Global Variable Table for the variable so that the variable serves as a network variable, i.e., the variable can be used for the EtherNet/IP connections.

In this example, set the network variable for Controller 1 as shown below.



| Name | Data Type | Initial Value | AT | Retain | Constant | Network Publish | Comment |
|---|---|---|---|---|---|---|---|
| Net_out1 | WORD | | | □ | □ | Output | |

- Name: Net_Out1
- Data type: WORD
- Network Publish attribute: Output

## Registering the Tag and Tag Set for the Target Device

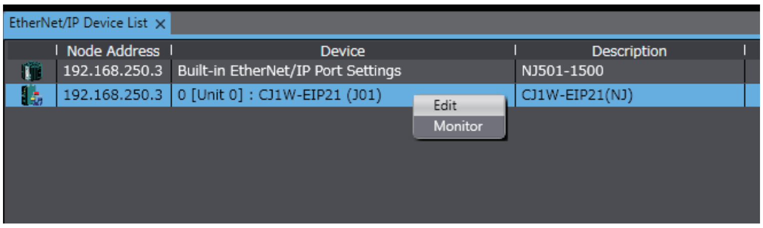
Set the tag and tag set for the target device.

**1** Select ***EtherNet/IP Connection Settings*** from the Tools Menu.

The EtherNet/IP Device List Tab Page is displayed.

**2** Right-click CJ1W-EIP21, the EtherNet/IP Unit connected to the Controller 2 (originator device in this example), and select ***Edit*** from the menu.

The EtherNet/IP Connection Setting Tab Page is displayed.

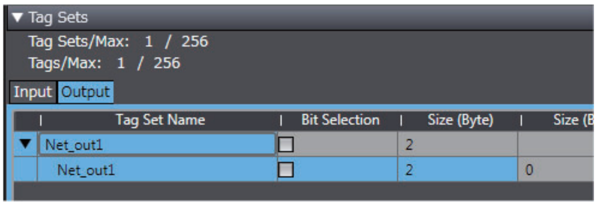


| Node Address | Device | Description |
|---|---|---|
| 192.168.250.3 | Built-in EtherNet/IP Port Settings | NJ501-1500 |
| 192.168.250.3 | 0 [Unit 0] : CJ1W-EIP21 (J01) | CJ1W-EIP21(NJ) |

**3** Click the (Show Tag Set Display) icon in the EtherNet/IP Connection Setting Tab Page.

**4** Click the **Output** tab to switch to the Output Tab Page. Register the following tag and tag set.

The tag and tag set can be registered in the same way as for the target device. (Refer to *Registering the Tag and Tag Set* on page A-14.)



Tag Sets/Max: 1 / 256
Tags/Max: 1 / 256

| Tag Set Name | Bit Selection | Size (Byte) | Size (B |
|---|---|---|---|
| ▼ Net_out1 | □ | 2 | |
| Net_out1 | □ | 2 | 0 |

## Checking the Device Bandwidth Usage

The PPS for the device can be displayed from the EtherNet/IP Connection Setting Tab Page.

This value is for when multicast filtering is used.

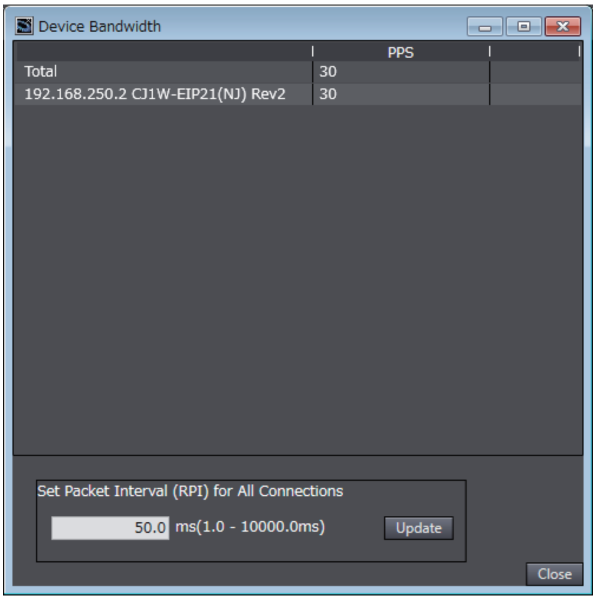
### Precautions for Correct Use

In the Device Bandwidth Dialog Box, you can only check the PPS being used for the EtherNet/IP connections from one originator device to its target devices.

The actual PPS used for the EtherNet/IP network must be calculated by taking into account of all PPS used on the EtherNet/IP network (i.e., PPS used for connections for the other devices in the EtherNet/IP network than the one given on the dialog box must be included into the calculation).

### ● Procedure

Click the **Device Bandwidth** Button in the EtherNet/IP Connection Setting Tab Page for the target device.



Device Bandwidth

| | PPS |
|---|---|
| Total | 30 |
| 192.168.250.2 CJ1W-EIP21(NJ) Rev2 | 30 |

Set Packet Interval (RPI) for All Connections

50.0 ms(1.0 - 10000.0ms)

| Menu | Description |
|---|---|
| PPS | Gives the PPS for each target device and total PPS used for all target devices. |
| Set Packet Interval (RPI) for All Connections | Changes all Packet Interval (RPI) values for all target devices. |

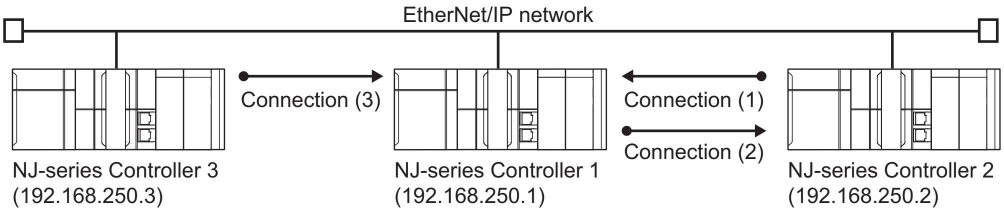
### Additional Information

You can specify a value in **Set Packet Interval (RPI) for All Connections** and click the **Update** Button to change packet interval (RPI) values set in the connection settings for all target devices to the specified value.

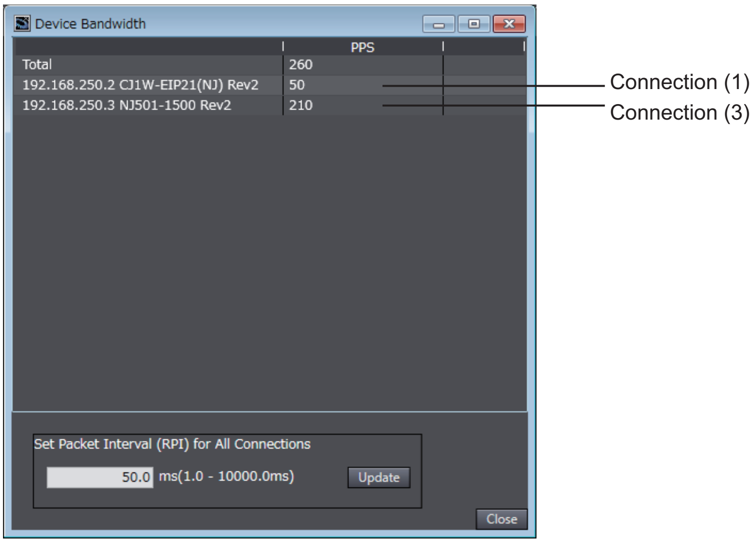
## ● Calculation Example for PPS for Each Device by the EtherNet/IP Connections

Establishing following three EtherNet/IP connections between Controllers (1) to (3) in the EtherNet/IP network

| Connection type | Relevant devices in the EtherNet/IP connections | Device PPS |
|---|---|---|
| Connection (1) | NJ-series Controller 2 (target device)<br>to NJ-series Controller 1 (originator device) | 50 pps |
| Connection (2) | NJ-series Controller 1 (target device)<br>to NJ-series Controller 2 (originator device) | 10 pps |
| Connection (3) | NJ-series Controller 3 (target device)<br>to NJ-series Controller 1 (originator device) | 210 pps |



PPS for each EtherNet/IP device is as given below.

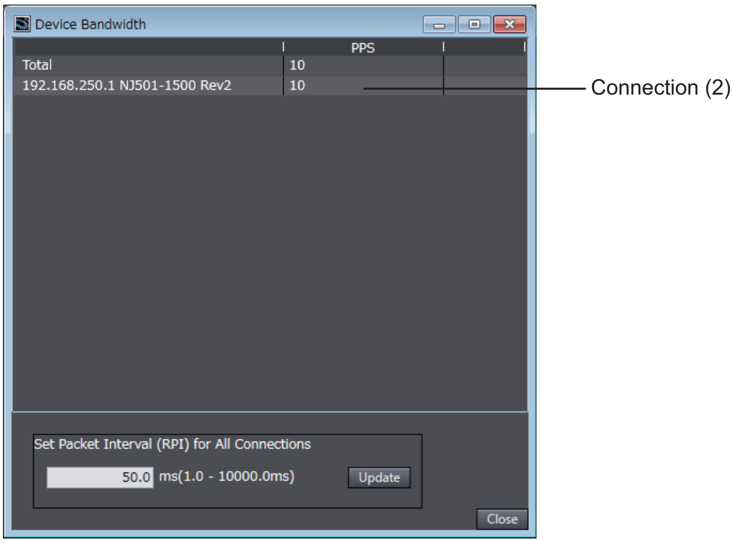


| | PPS |
|---|---|
| Total | 260 |
| 192.168.250.2 CJ1W-EIP21(NJ) Rev2 | 50 |
| 192.168.250.3 NJ501-1500 Rev2 | 210 |

Set Packet Interval (RPI) for All Connections

50.0 ms(1.0 - 10000.0ms)

**EtherNet/IP connection settings for Controller 1**



Device Bandwidth

| | PPS |
|---|---|
| Total | 10 |
| 192.168.250.1 NJ501-1500 Rev2 | 10 |

Set Packet Interval (RPI) for All Connections

50.0 ms(1.0 - 10000.0ms) Update

Close

Connection (2)

**EtherNet/IP connection settings for Controller 2**

In this example, the PPS for Connection (1) is 50 pps, the PPS for Connection (2) is 10 pps, and the PPS for Connection (3) is 210 pps. Therefore, PPS for each EtherNet/IP device is as given below.

192.168.250.1: 270 pps = 50 pps (for Connection (1)) + 10 pps (for Connection (2)) + 210 pps (for Connection (3))

192.168.250.2: 60 pps = 50 pps (for Connection (1)) + 10 pps (for Connection (2))

192.168.250.3: 210 pps = 210 pps (for Connection (3))

## ● Adjusting Method

If the calculation result value exceeds the values in the specifications of the devices used in the EtherNet/IP connections, re-evaluate the overall network configuration and correct it by taking steps such as selecting a different Ethernet switch or splitting the network.

If the RPI is made longer, the PPS for the EtherNet/IP connections will decrease.

You can change the RPI value in the connection settings for all target devices by specifying a value in Set Packet Interval (RPI) for All Connections in this dialog box.

Refer to *14-2-2 Tag Data Link Bandwidth Usage and RPI* on page 14-9 for the relationship between the PPS for the device and the RPI.

## Transferring the Connection Settings Data

**Precautions for Correct Use**

- If the node addresses (IP addresses) are not set correctly, you may connect to the wrong Controller and set incorrect device parameters. Download data only after you confirm that you are connected to the correct Controller.
- If incorrect connection settings are set, it may cause equipment to operate unpredictably. Even when the correct connection settings are set, make sure that there will be no effect on equipment before you transfer the data.
- A connection error will result if the network variables that are used in the tag settings are not set in the Controller. Before downloading the connection settings, check to confirm that the network variables used in the tag settings are set in the Controller.
- If a communications error occurs, the output status depends on the specifications of the device being used. When a communications error occurs for a device that is used along with output devices, check the operating specifications and implement safety countermeasures.
- The built-in EtherNet/IP port and the port on the EtherNet/IP Unit are 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.
- The EtherNet/IP connections between relevant nodes is stopped during a download. Before you download data in RUN mode, make sure that it will not affect the controlled system.

  Also implement interlocks on data processing in ladder programming that uses EtherNet/IP connections when the connections are stopped or a connection error occurs.
- In the EtherNet/IP network, if the device PPS exceeds the unit's allowable bandwidth (PPS), the EtherNet/IP connection operations may not agree with the settings.

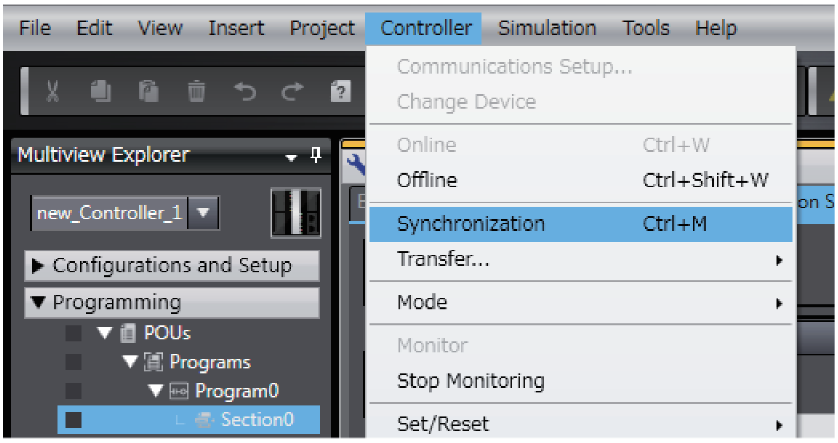
  If you increase the RPI value in such a case, there are cases when the problem can be resolved (i.e., the operations agree the settings).

## ● Synchronizing/Transferring a Whole Project

You can synchronize and transfer the EtherNet/IP connection settings along with the program data. You can also transfer all the EtherNet/IP connection settings along with the program data.

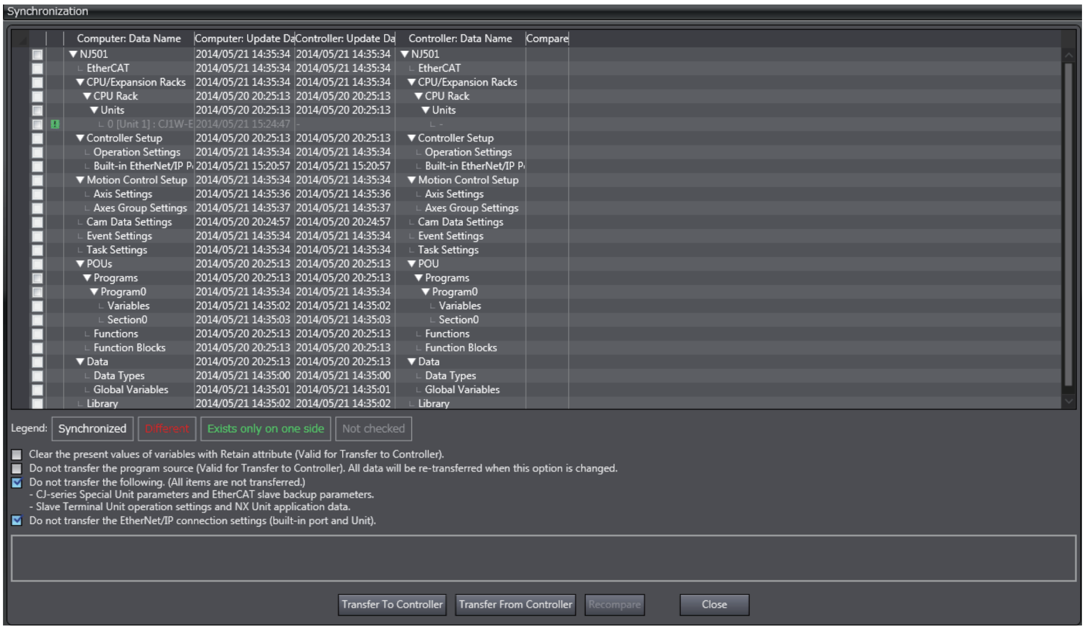
- Synchronizing and transferring the data

**1** Establish an online connection between the computer and the Controller and then select ***Synchronization*** from the Controller Menu. (Or, click the ⟳ Button on the Toolbar.)



The Synchronization Window is displayed, and comparison of the user program and parameter settings between the Sysmac Studio and the Controller is started.

**2** The following Uploading and Downloading Data Window is displayed after the automatic comparison.



| | | Computer: Data Name | Computer: Update Da | Controller: Update Da | Controller: Data Name | Compare |
|---|---|---|---|---|---|---|
| □ | | ▼NJ501 | 2014/05/21 14:35:34 | 2014/05/21 14:35:34 | ▼NJ501 | |
| □ | | EtherCAT | 2014/05/21 14:35:34 | 2014/05/21 14:35:34 | EtherCAT | |
| □ | | ▼CPU/Expansion Racks | 2014/05/21 14:35:34 | 2014/05/21 14:35:34 | ▼CPU/Expansion Racks | |
| □ | | ▼CPU Rack | 2014/05/20 20:25:13 | 2014/05/20 20:25:13 | ▼CPU Rack | |
| □ | | ▼Units | 2014/05/20 20:25:13 | 2014/05/20 20:25:13 | ▼Units | |
| □ | ! | 0 [Unit 1] : CJ1W-E | 2014/05/21 15:24:47 | - | - | |
| □ | | ▼Controller Setup | 2014/05/20 20:25:13 | 2014/05/20 20:25:13 | ▼Controller Setup | |
| □ | | Operation Settings | 2014/05/21 14:35:34 | 2014/05/21 14:35:34 | Operation Settings | |
| □ | | Built-in EtherNet/IP P | 2014/05/21 15:20:57 | 2014/05/21 15:20:57 | Built-in EtherNet/IP P | |
| □ | | ▼Motion Control Setup | 2014/05/21 14:35:34 | 2014/05/21 14:35:34 | ▼Motion Control Setup | |
| □ | | Axis Settings | 2014/05/21 14:35:36 | 2014/05/21 14:35:36 | Axis Settings | |
| □ | | Axes Group Settings | 2014/05/21 14:35:37 | 2014/05/21 14:35:37 | Axes Group Settings | |
| □ | | Cam Data Settings | 2014/05/20 20:24:57 | 2014/05/20 20:24:57 | Cam Data Settings | |
| □ | | Event Settings | 2014/05/21 14:35:34 | 2014/05/21 14:35:34 | Event Settings | |
| □ | | Task Settings | 2014/05/21 14:35:34 | 2014/05/21 14:35:34 | Task Settings | |
| □ | | ▼POUs | 2014/05/20 20:25:13 | 2014/05/20 20:25:13 | ▼POU | |
| □ | | ▼Programs | 2014/05/20 20:25:13 | 2014/05/20 20:25:13 | ▼Programs | |
| □ | | ▼Program0 | 2014/05/21 14:35:34 | 2014/05/21 14:35:34 | ▼Program0 | |
| □ | | Variables | 2014/05/21 14:35:02 | 2014/05/21 14:35:02 | Variables | |
| □ | | Section0 | 2014/05/21 14:35:03 | 2014/05/21 14:35:03 | Section0 | |
| □ | | Functions | 2014/05/20 20:25:13 | 2014/05/20 20:25:13 | Functions | |
| □ | | Function Blocks | 2014/05/20 20:25:13 | 2014/05/20 20:25:13 | Function Blocks | |
| □ | | ▼Data | 2014/05/20 20:25:13 | 2014/05/20 20:25:13 | ▼Data | |
| □ | | Data Types | 2014/05/21 14:35:00 | 2014/05/21 14:35:00 | Data Types | |
| □ | | Global Variables | 2014/05/21 14:35:01 | 2014/05/21 14:35:01 | Global Variables | |
| □ | | Library | 2014/05/21 14:35:02 | 2014/05/21 14:35:02 | Library | |

□ Clear the present values of variables with Retain attribute (Valid for Transfer to Controller).
□ Do not transfer the program source (Valid for Transfer to Controller). All data will be re-transferred when this option is changed.
☑ Do not transfer the following. (All items are not transferred.)
- CJ-series Special Unit parameters and EtherCAT slave backup parameters.
- Slave Terminal Unit operation settings and NX Unit application data.
☑ Do not transfer the EtherNet/IP connection settings (built-in port and Unit).

**3** Clear the *Do not transfer the EtherNet/IP connection settings (built-in port and Unit)* Check Box and click the **Transfer to Controller** Button.
Then the EtherNet/IP connection settings are transferred along with the not-synchronized data. If no EtherNet/IP connection settings are set in the Sysmac Studio, no data will be sent.

- Transferring all data

**1** Establish an online connection between the computer and the Controller and then select ***Transfer* - *To Controller*** from the Controller Menu. (Or, click the Button on the Toolbar.)

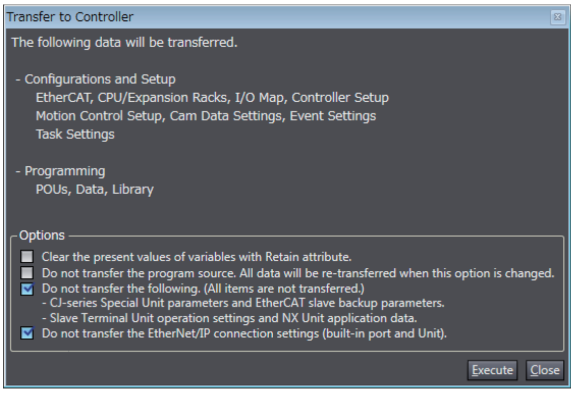
**2** The **Transfer to Controller** Dialog Box is displayed.

Clear the *Do not transfer the EtherNet/IP connection settings (built-in port and Unit)* Check Box and click the **Execute** Button.

### Precautions for Correct Use

To transfer only the connection settings, execute Transfer from the EtherNet/IP Connection Setting Tab Page.

The connection settings are not transferred from the Synchronization Window, the Transfer to Controller Dialog Box, and the Transfer from Controller Dialog Box, even if you clear the *Do not transfer the connection setting* Check Box, as long as the data in the computer and in the Controller is the same.



Transfer to Controller

The following data will be transferred.

- Configurations and Setup
  EtherCAT, CPU/Expansion Racks, I/O Map, Controller Setup
  Motion Control Setup, Cam Data Settings, Event Settings
  Task Settings

- Programming
  POUs, Data, Library

Options

☐ Clear the present values of variables with Retain attribute.
☐ Do not transfer the program source. All data will be re-transferred when this option is changed.
☑ Do not transfer the following. (All items are not transferred.)
- CJ-series Special Unit parameters and EtherCAT slave backup parameters.
- Slave Terminal Unit operation settings and NX Unit application data.
☑ Do not transfer the EtherNet/IP connection settings (built-in port and Unit).

## ● Transferring Only the EtherNet/IP Connection Settings

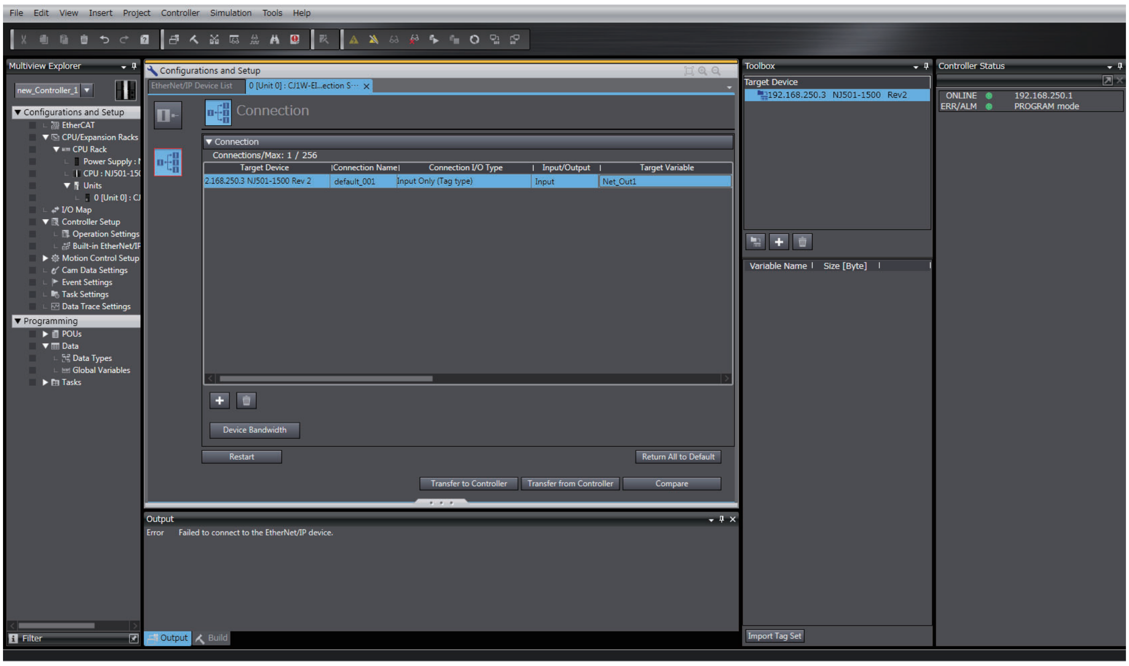
You can transfer tag sets and connections to the EtherNet/IP devices.

**1** Establish an online connection with the Controller.

**2** Click the **Transfer to Controller** or **Transfer from Controller** Button in the EtherNet/IP Connection Setting Tab Page.

The tag settings and connection settings set at that time are transferred to the Controller connected online.

**3** If the Controller connected online is in RUN mode, the dialog box to confirm whether to switch to PROGRAM mode before transferring the settings is displayed.



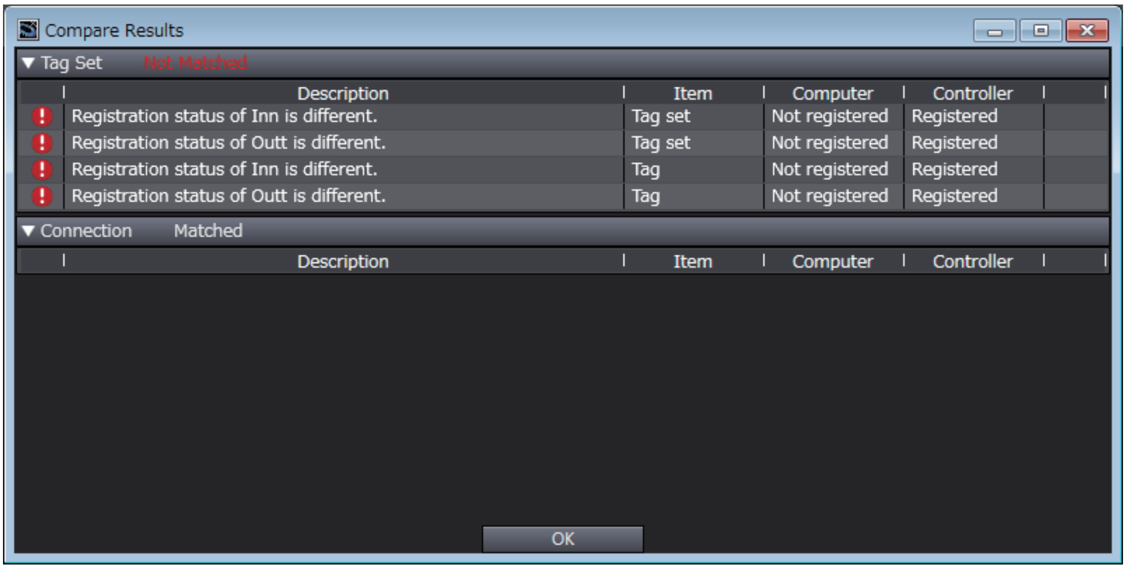
| Target Device | Connection Name | Connection I/O Type | Input/Output | Target Variable |
|---|---|---|---|---|
| 2.168.250.3 NJ501-1500 Rev 2 | default_001 | Input Only (Tag type) | Input | Net_Out1 |

```
Error    Failed to connect to the EtherNet/IP device.
```

## ● Comparison

The differences in the tag set and connection settings between the project and the EtherNet/IP devices can be displayed.

**1** Click the **Compare** Button in the EtherNet/IP Connection Setting Tab Page.



Compare Results

▼ Tag Set    Not Matched

| | Description | Item | Computer | Controller |
|---|---|---|---|---|
| ! | Registration status of Inn is different. | Tag set | Not registered | Registered |
| ! | Registration status of Outt is different. | Tag set | Not registered | Registered |
| ! | Registration status of Inn is different. | Tag | Not registered | Registered |
| ! | Registration status of Outt is different. | Tag | Not registered | Registered |

▼ Connection    Matched

| | Description | Item | Computer | Controller |
|---|---|---|---|---|

## Starting and Stopping EtherNet/IP Connections

### ● Automatically Starting EtherNet/IP Connections

The EtherNet/IP device is automatically restarted and EtherNet/IP connections are automatically started immediately after the connection settings are downloaded from the Sysmac Studio.

**Precautions for Correct Use**

Connections are adversely cut off if any of the following errors occurs in the CPU Unit that is the originator while EtherNet/IP connections are active.

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

### ● Starting and Stopping the EtherNet/IP Connections for the Entire Network

You can start and stop EtherNet/IP connections from the user program or from the Sysmac Studio.

**Precautions for Correct Use**

Use the same method (i.e., either the user program or the tool software) to both start and stop EtherNet/IP connections.

For example, if you use the *_EIP_TDLinkStopCmd* (Tag Data Link Communications Stop Switch) system-defined variable to stop EtherNet/IP connections, you cannot start them from the Sysmac Studio and the Network Configurator.

## A-2-5 Checking Communications Status with the Sysmac Studio and Troubleshooting

You can monitor the communications status of the EtherNet/IP connections after their settings are set. You can also check errors.

**Precautions for Correct Use**

Make sure that the connection settings in both the Sysmac Studio and the Controller are consistent before using the monitor functions. You can use the *Comparison* on page A-32 to see if they are the same.

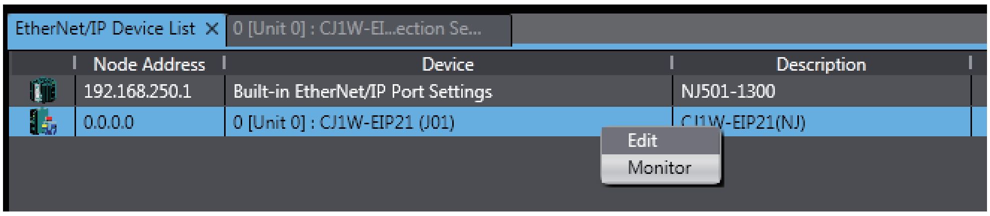
### Checking Communications Status with the Sysmac Studio

You can check the communications status on the EtherNet/IP connections in the EtherNet/IP Connection Monitor Tab Page.

**1** Select ***EtherNet/IP Connection Settings*** from the Tools Menu to display the EtherNet/IP Device List Tab Page.

**2** Right-click the Controller you want to check the communications status and select ***Monitor*** from the menu.

The pane to monitor the EtherNet/IP connection is displayed. This pane has six tabs for each communications status.



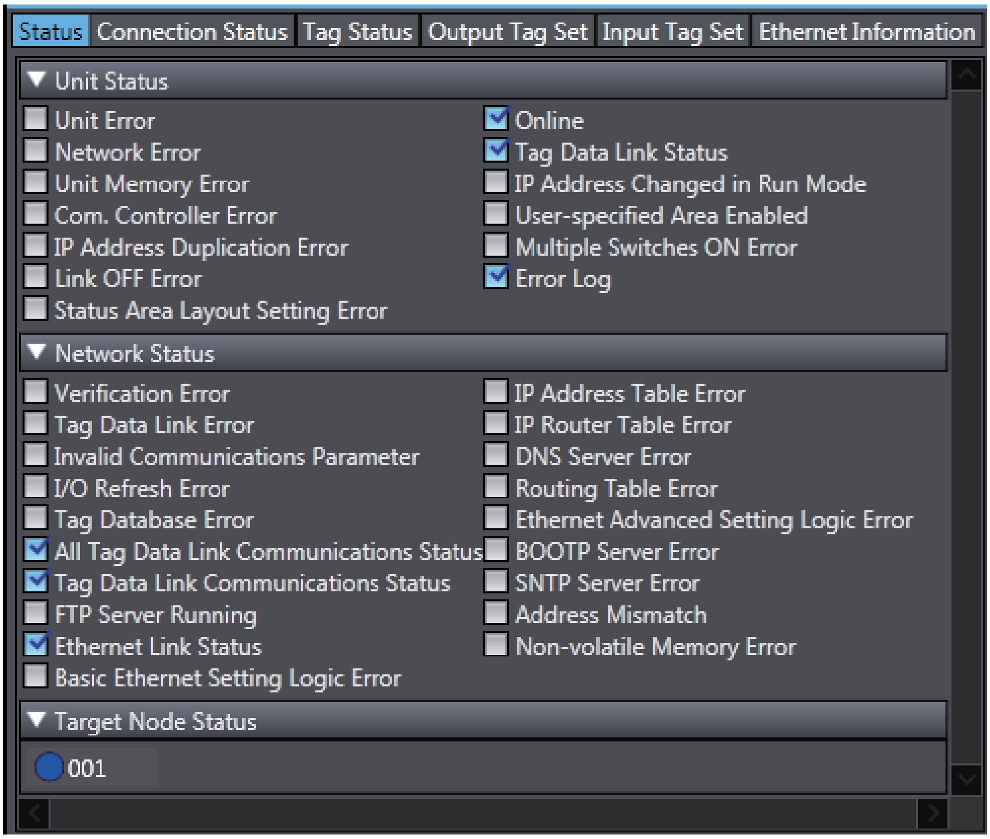
EtherNet/IP Device List ×
0 [Unit 0] : CJ1W-EI...ection Se...
Edit
Monitor

| Node Address | Device | Description |
|---|---|---|
| 192.168.250.1 | Built-in EtherNet/IP Port Settings | NJ501-1300 |
| 0.0.0.0 | 0 [Unit 0] : CJ1W-EIP21 (J01) | CJ1W-EIP21(NJ) |

**3** Select one of the six tabs for which you want to confirm the communications status.

- Status Tab Page

This tab page gives the TRUE/FALSE status of the system-defined variables that monitors the tag data link errors and communication status. If any of the variables are TRUE, the checkbox in front of the variable will be selected. Refer to *15-1-1 The Network Configurator's Device Monitor Function* for details on each status item.



▼ Unit Status

□ Unit Error
□ Network Error
□ Unit Memory Error
□ Com. Controller Error
□ IP Address Duplication Error
□ Link OFF Error
□ Status Area Layout Setting Error
☑ Online
☑ Tag Data Link Status
□ IP Address Changed in Run Mode
□ User-specified Area Enabled
□ Multiple Switches ON Error
☑ Error Log

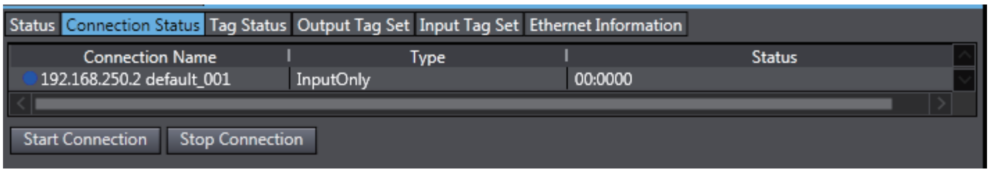
▼ Network Status

□ Verification Error
□ Tag Data Link Error
□ Invalid Communications Parameter
□ I/O Refresh Error
□ Tag Database Error
☑ All Tag Data Link Communications Status
☑ Tag Data Link Communications Status
□ FTP Server Running
☑ Ethernet Link Status
□ Basic Ethernet Setting Logic Error
□ IP Address Table Error
□ IP Router Table Error
□ DNS Server Error
□ Routing Table Error
□ Ethernet Advanced Setting Logic Error
□ BOOTP Server Error
□ SNTP Server Error
□ Address Mismatch
□ Non-volatile Memory Error

▼ Target Node Status

● 001

- Connection Status Tab Page
  Current status of each connection is given.

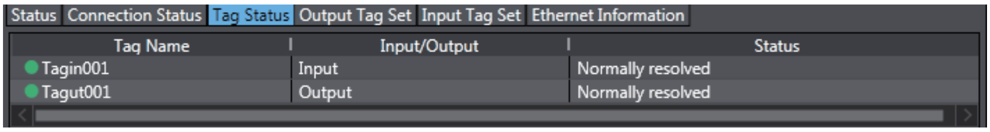


| Connection Name | Type | Status |
|---|---|---|
| 192.168.250.2 default_001 | InputOnly | 00:0000 |

| Name | Description |
|---|---|
| Connection Name | Gives the current status of each connection with the following text colors.<br>Green: Normal<br>Red: There is at least one connection that has not been established.<br>Gray: There are no connections or the connection operation is stopped. |
| Type | Gives the connection type. |
| Status | Gives the current status on each connection with codes.<br>• Normal operation: 00:0000<br>• Abnormal operation: Gives an error code. This information can be used to identify the cause of EtherNet/IP connection errors. Refer to *15-3 Connection Status Codes and Error Processing* on page 15-20 for details on the connection status. |

- Tag Status Tab Page

  This tab page gives if the tag settings for each tag for EtherNet/IP connections are set so that data can be exchanged with target devices.



| Tag Name | Input/Output | Status |
|---|---|---|
| Tagin001 | Input | Normally resolved |
| Tagut001 | Output | Normally resolved |

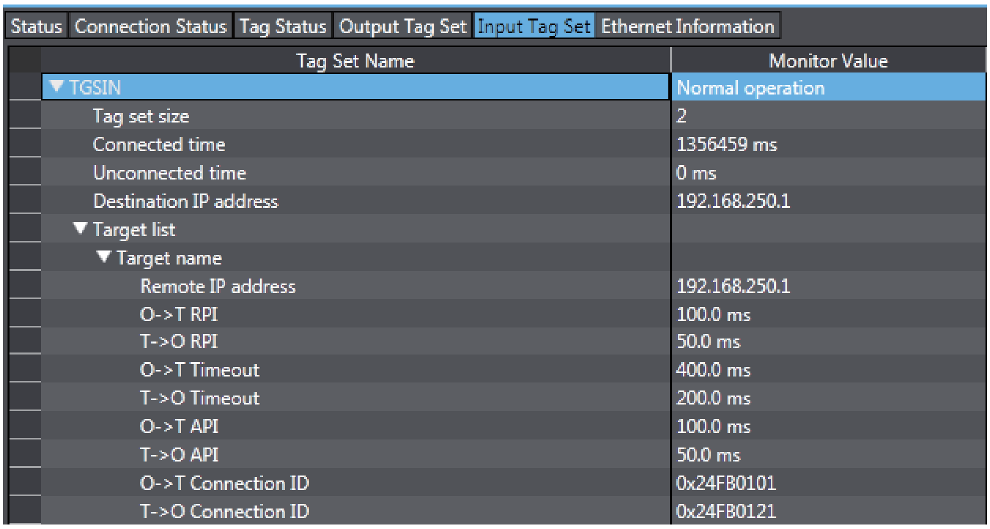
| Name | Description |
|---|---|
| Tag Name | Gives the tag name. |
| Input/Output | Gives the type of the tag. |
| Status | The following status is displayed depending on the status that is set.<br>• Normally resolved: Normal data exchange is possible.<br>• Different sizes: Different sizes are set for the network variables and the tag settings.<br>A connection will not be established for a tag for which this error occurs.<br>• No tag: A network variable is not set in the variable table in the CPU Unit for the specified tag setting. Or, instead of a member of union variable, unions are specified.<br>A connection will not be established for a tag for which this error occurs.<br>• Attribute error: Writing is not possible for Read Only and Constant attributes. |

- Output Tag Set and Input Tag Set Tab Pages

  You can monitor the status of each input/output tag set that is used for the EtherNet/IP connections.

  **Note** The tag set status monitor is not available for a built-in EtherNet/IP port on NJ-series Controller version 1.08 or earlier.

  Click ▼ of each tag to display its detailed information.



| Tag Set Name | Monitor Value |
|---|---|
| ▼ TGSIN | Normal operation |
| Tag set size | 2 |
| Connected time | 1356459 ms |
| Unconnected time | 0 ms |
| Destination IP address | 192.168.250.1 |
| ▼ Target list | |
| ▼ Target name | |
| Remote IP address | 192.168.250.1 |
| O->T RPI | 100.0 ms |
| T->O RPI | 50.0 ms |
| O->T Timeout | 400.0 ms |
| T->O Timeout | 200.0 ms |
| O->T API | 100.0 ms |
| T->O API | 50.0 ms |
| O->T Connection ID | 0x24FB0101 |
| T->O Connection ID | 0x24FB0121 |

| Name | Description |
|---|---|
| Tag Set Name | Gives the connection status.<br>If there is a connection error, "Not connected or error" is given. |
| Tag set size | Gives the size of the tag set in bytes. |
| Connected time | Gives the total connection duration in milliseconds. |
| Unconnected time | Gives the total disconnection duration in milliseconds. |
| Number of connections (in the Output Tag Set Tab Page) | Gives the number of connections. |
| Number of connected originators (in the Output Tag Set Tab Page) | Gives the number of the connected originator devices. |
| Originator list (in the Output Tag Set Tab Page), Target list (in the Input Tag Set Tab Page) | Gives the detailed information of the connected originators. |
| Originator name (in the Output Tag Set Tab Page), Produced tag name (in the Input Tag Set Tab Page) | Gives no information. |
| IP address (in the Output Tag Set Tab Page), Remote IP address (in the Input Tag Set Tab Page) | Gives the IP addresses allocated for the originators. |
| Connected time (in the Output Tag Set Tab Page) | Gives the total duration of connection with the originator in milliseconds. |
| Unconnected time (in the Output Tag Set Tab Page) | Gives the total duration of disconnection with the originator in milliseconds. |

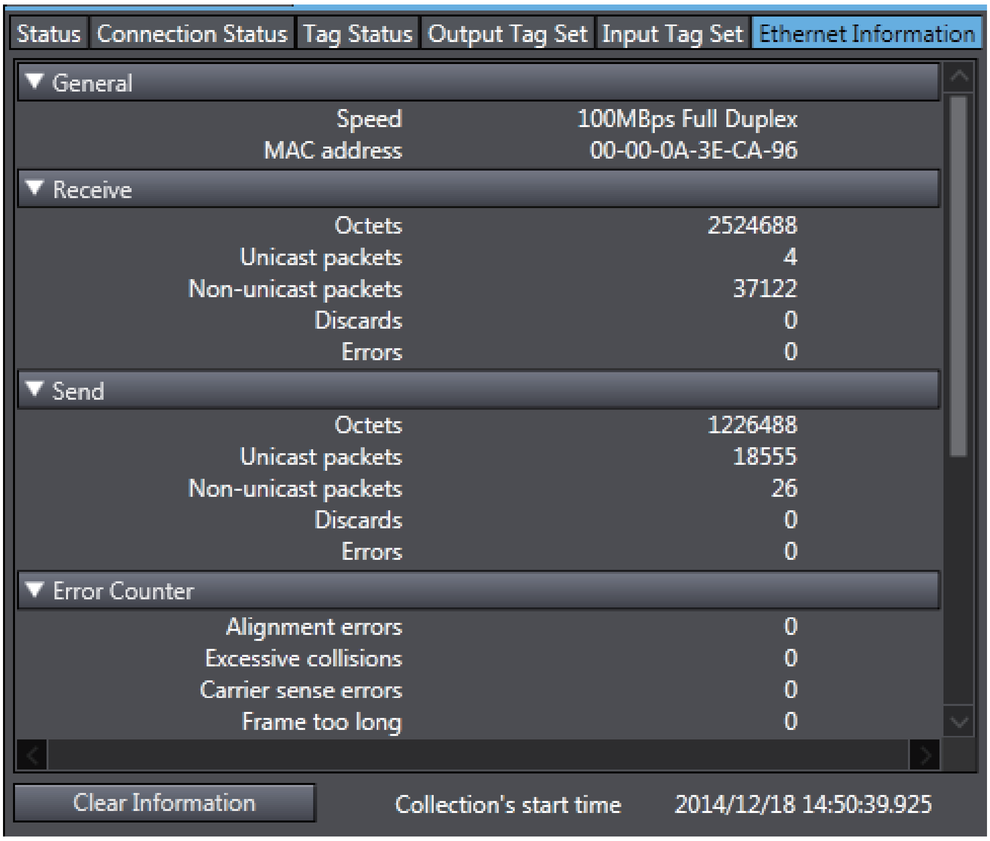
| Name | Description |
|---|---|
| Destination IP address (in the Output Tag Set Tab Page) | Gives the destination IP addresses. If the multi-cast connections are used, its own multi-cast address is displayed. |
| O->T RPI | Gives the RPI of connection from the originator to the target in milliseconds. |
| T->O RPI | Gives the RPI of connection from the target to the originator in milliseconds. |
| O->T Timeout | Gives the timeout time for the connections from the originator to the target in milliseconds. |
| T->O Timeout | Gives the timeout time for the connections from the target to the originator in milliseconds. |
| O->T API | Gives the API of connection from the originator to the target in milliseconds. |
| T->O API | Gives the API of connection from the target to the originator in milliseconds. |
| O->T Connection ID | Gives the connection identification for the connections from the originator to the target in hexadecimal. |
| T->O Connection ID | Gives the connection identification for the connections from the target to the originator in hexadecimal. |

- Ethernet Information Tab Page

This tab page displays the communications status at the communications driver level of the EtherNet/IP Unit (built-in port).

The error counter information can be used to confirm whether communications problems have occurred.

Under the Tag Data Link, you can confirm characteristics such as the bandwidth usage (PPS).



▼ General

| | |
|---|---|
| Speed | 100MBps Full Duplex |
| MAC address | 00-00-0A-3E-CA-96 |

▼ Receive

| | |
|---|---|
| Octets | 2524688 |
| Unicast packets | 4 |
| Non-unicast packets | 37122 |
| Discards | 0 |
| Errors | 0 |

▼ Send

| | |
|---|---|
| Octets | 1226488 |
| Unicast packets | 18555 |
| Non-unicast packets | 26 |
| Discards | 0 |
| Errors | 0 |

▼ Error Counter

| | |
|---|---|
| Alignment errors | 0 |
| Excessive collisions | 0 |
| Carrier sense errors | 0 |
| Frame too long | 0 |

Collection's start time 2014/12/18 14:50:39.925

## A-2-6 Troubleshooting

In the case that there is a setting error or a communications error in the EtherNet/IP networks, the Sysmac Studio displays the error in the Troubleshooting Dialog Box.

Refer to *15-6 Troubleshooting* on page 15-48 for the confirmation methods for errors and information on errors.

### Troubleshooting When Transferring and Monitoring the EtherNet/IP Connection Settings Fail with Sysmac Studio Version 1.10 or Higher

The first time you establish an online connection between the Controller and the computer that runs the Sysmac Studio version 1.10 or higher with Windows Firewall on the computer enabled, the dialog box to confirm the connection may be displayed. If that occurs, make the following selection in the dialog box.

- Unblock (on Windows XP/Vista)
- Allow access (on Windows 7)

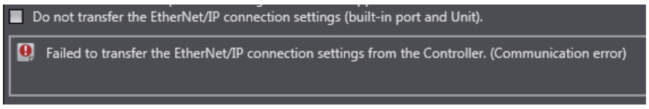
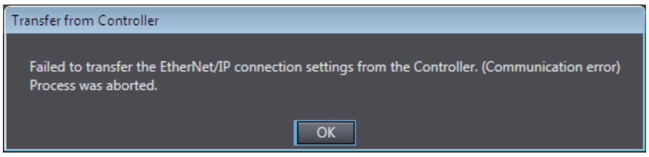
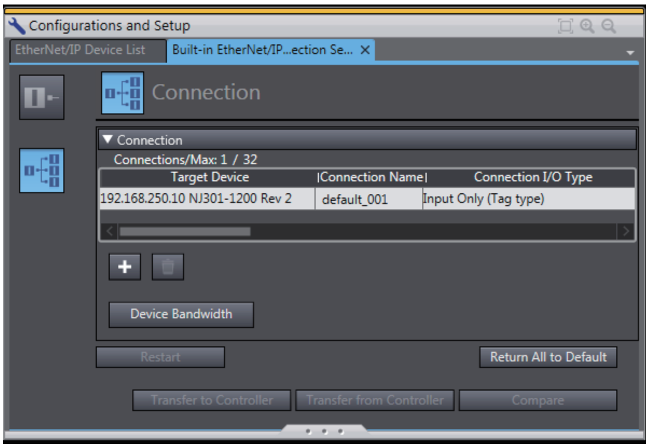
If you make other selections than above, there are cases when transferring and monitoring the EtherNet/IP connection settings cannot properly be performed even if the online connection is successfully established.

In such cases, follow the procedure described in Method 1 or Method 2 below.

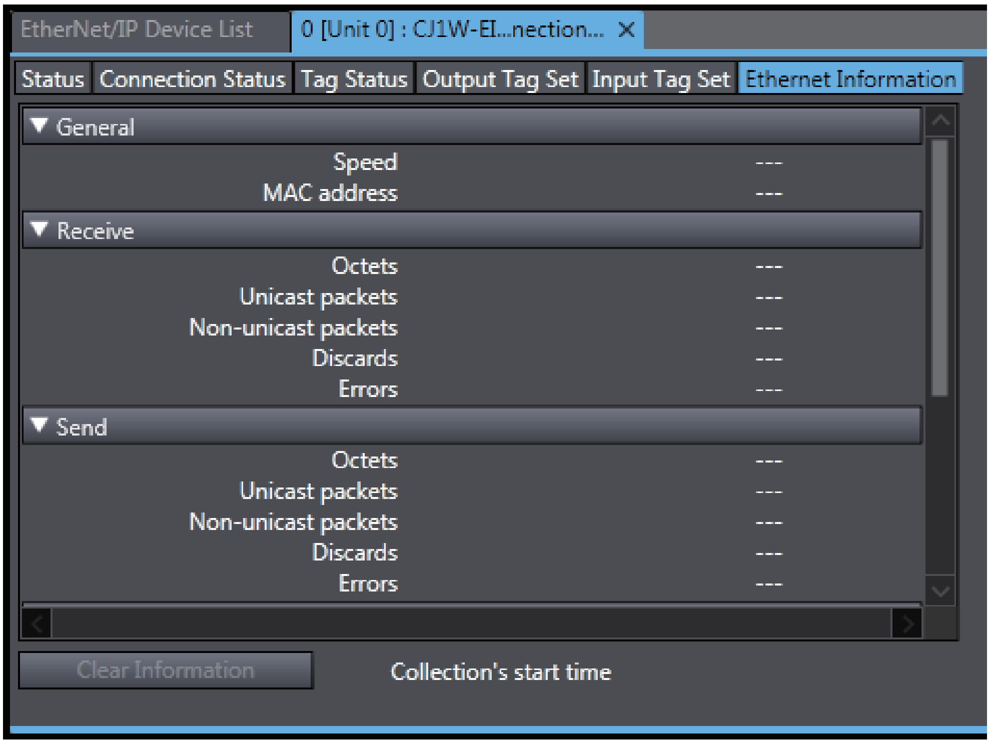
- *Method 1: Disabling Windows Firewall Settings* (on page A-41)
- *Method 2: Selecting the Use Option for the CIP Message Server* (on page A-44)

## Problems

- The connection setting data cannot be transferred

| Data Transmission Screen | Problem |
|---|---|
| Synchronization Window | The Sysmac Studio displays the following error message and the data will not be transferred.<br><br>□ Do not transfer the EtherNet/IP connection settings (built-in port and Unit).<br>Failed to transfer the EtherNet/IP connection settings from the Controller. (Communication error) |
| Transfer to Controller Dialog Box | The Sysmac Studio displays the following error dialog box and the data will not be transferred.<br><br>Transfer from Controller<br>Failed to transfer the EtherNet/IP connection settings from the Controller. (Communication error)<br>Process was aborted.<br> |
| EtherNet/IP Connection Setting Tab Page | The **Transfer to Controller** and **Transfer from Controller** Buttons are grayed out and the data cannot be transferred/compared.<br><br> Configurations and Setup<br>EtherNet/IP Device List<br>Built-in EtherNet/IP...ection Se...<br>Connection<br>Connection<br>Device Bandwidth<br>Restart<br>Return All to Default<br>Transfer to Controller<br>Transfer from Controller<br>Compare<br>Connections/Max: 1 / 32<br>Target Device: 192.168.250.10 NJ301-1200 Rev 2<br>Connection Name: default_001<br>Connection I/O Type: Input Only (Tag type) |

- Monitoring the settings cannot be performed
  Monitor data items in the EtherNet/IP Connection Monitor Tab Page remain "---".



| General | |
|---|---|
| Speed | --- |
| MAC address | --- |
| **Receive** | |
| Octets | --- |
| Unicast packets | --- |
| Non-unicast packets | --- |
| Discards | --- |
| Errors | --- |
| **Send** | |
| Octets | --- |
| Unicast packets | --- |
| Non-unicast packets | --- |
| Discards | --- |
| Errors | --- |

## ● Method 1: Disabling Windows Firewall Settings

### Precautions for Correct Use

The main function of the firewall is to prevent unwanted access from external sources (e.g., the Internet).

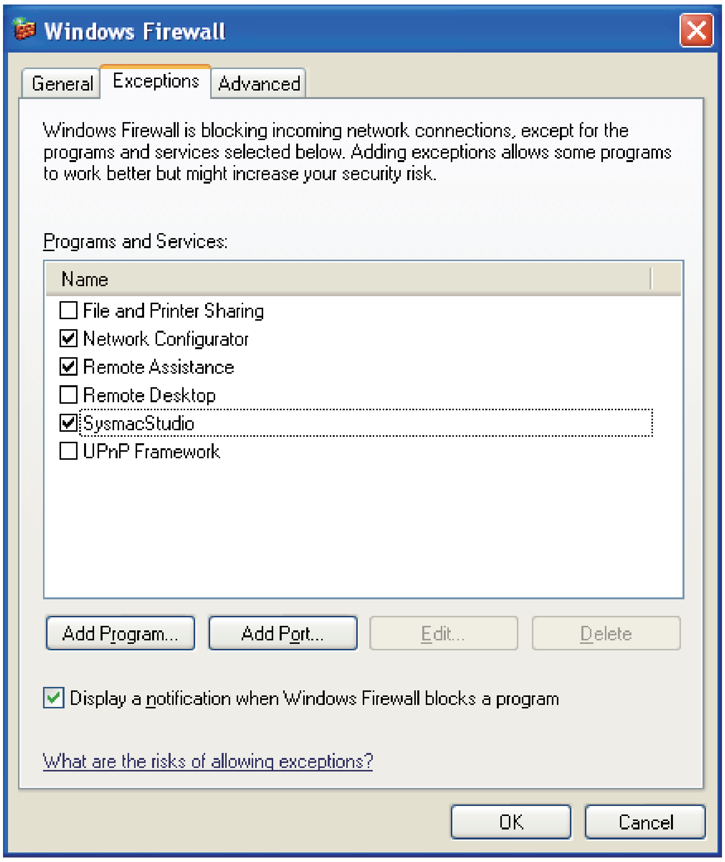
The changes that are made with the following procedures are to allow the Sysmac Studio and the NJ-series Controller to connect. If your computer is on an inhouse network, make sure that security will not be jeopardized before you change the settings.

- Windows XP

**1** Open the **Control Panel** from the **Windows Start Menu** and then select **Windows Firewall** icon.

The **Windows Firewall** Dialog Box is displayed.

**2** Click on the **Exceptions** tab and select **Sysmac Studio** in the **Programs and Services** list.



Windows Firewall is blocking incoming network connections, except for the programs and services selected below. Adding exceptions allows some programs to work better but might increase your security risk.

Programs and Services:

| Name |
|---|
| □ File and Printer Sharing |
| ☑ Network Configurator |
| ☑ Remote Assistance |
| □ Remote Desktop |
| ☑ SysmacStudio |
| □ UPnP Framework |

☑ Display a notification when Windows Firewall blocks a program

- Windows Vista, Windows 7, or later version

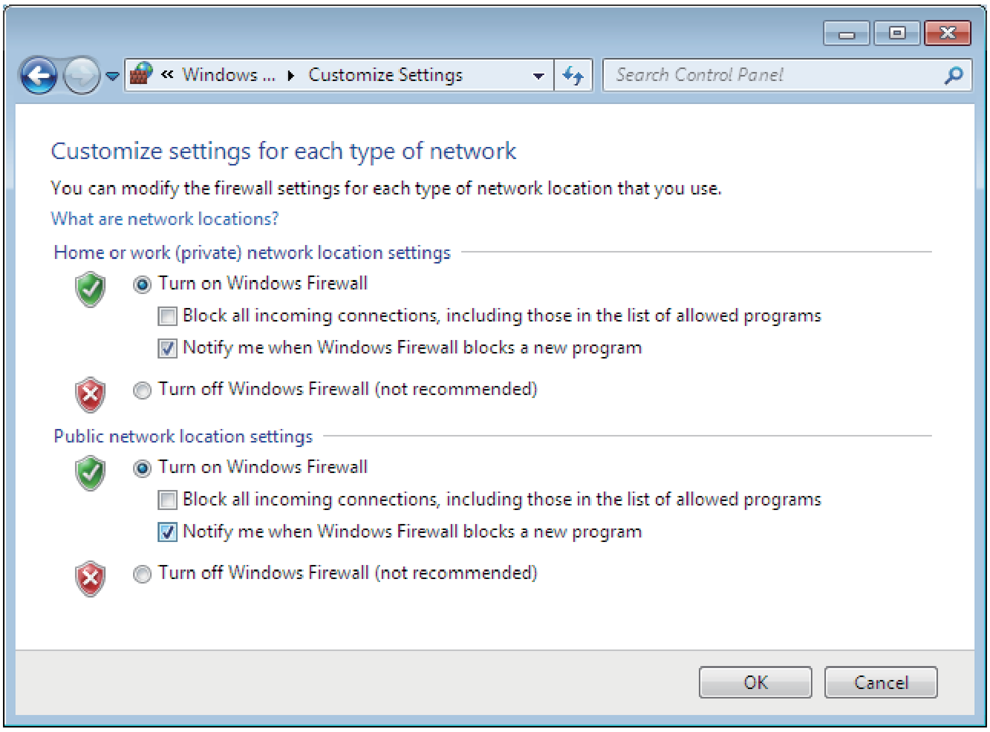
**1** Open the **Control Panel** from the **Windows Start Menu** and then select **Windows Firewall** icon.

The **Windows Firewall** Dialog Box is displayed.

**2** Select Turn Windows Firewall on or off.

The **Customize Settings** Dialog box is displayed.

**3** Clear the **Block all incoming connections, including those in the list of allowed programs** Check Box and click the OK Button.



Customize settings for each type of network

You can modify the firewall settings for each type of network location that you use.

What are network locations?

Home or work (private) network location settings

● Turn on Windows Firewall

□ Block all incoming connections, including those in the list of allowed programs

☑ Notify me when Windows Firewall blocks a new program

○ Turn off Windows Firewall (not recommended)

Public network location settings

● Turn on Windows Firewall

□ Block all incoming connections, including those in the list of allowed programs

☑ Notify me when Windows Firewall blocks a new program

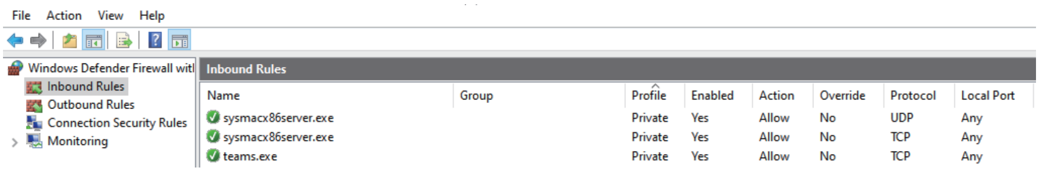
○ Turn off Windows Firewall (not recommended)

**4** Select Advanced settings in the Windows Firewall Dialog Box.

The **Windows Firewall with Advanced Security** Dialog Box is displayed.

**5** Click ***Inbound Rules*** in the left pane. Then, for Sysmac Studio version 1.31 or higher, double-click **sysmacx86server** in the **Inbound Rules** list. For Sysmac Studio version lower than 1.31, double-click **SysmacStudio**.

If you double-click **sysmacx86server**, the **sysmacx86server Properties** Dialog Box is displayed. If you double-click **SysmacStudio**, the **SysmacStudio Properties** Dialog Box is displayed.



Inbound Rules

| Name | Group | Profile | Enabled | Action | Override | Protocol | Local Port |
|---|---|---|---|---|---|---|---|
| sysmacx86server.exe | | Private | Yes | Allow | No | UDP | Any |
| sysmacx86server.exe | | Private | Yes | Allow | No | TCP | Any |
| teams.exe | | Private | Yes | Allow | No | TCP | Any |

**6** For Sysmac Studio version 1.31 or higher, make the following settings in the **sysmacx86server Properties** Dialog Box. For Sysmac Studio version lower than 1.31, make the following settings in the **SysmacStudio Properties** Dialog Box.

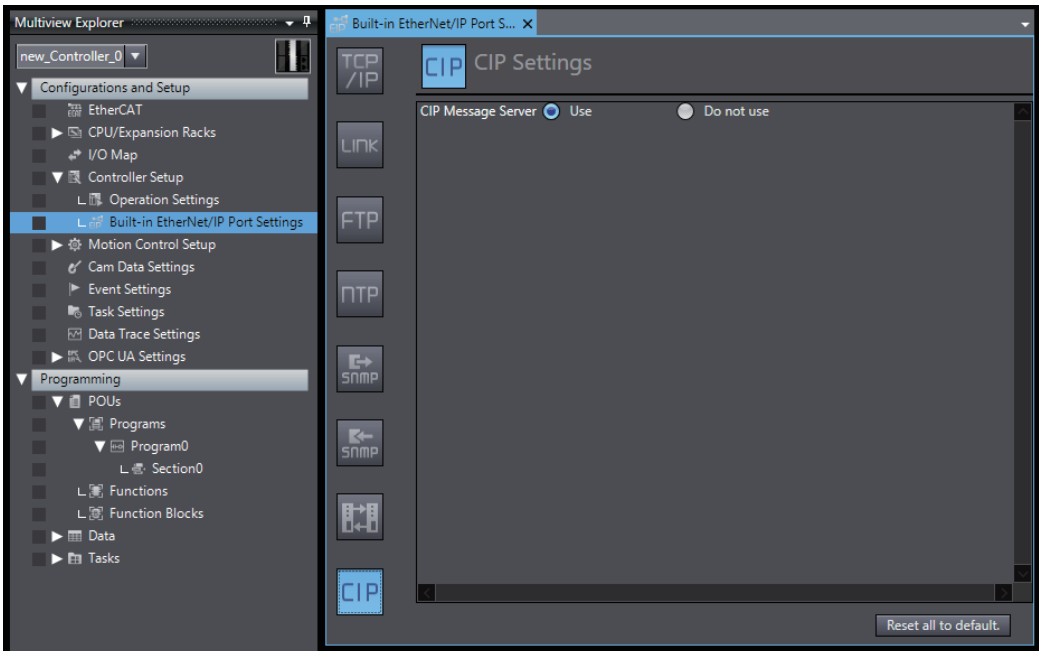
- **Profiles** on the **Advanced** Tab Page: Select the **Public** Check Box, if not selected.
- **General** on the **General** Tab Page: Select the **Enabled** Check Box, if not selected.
- **Action** on the **General** Tab Page: Select the **Allow the connection** Option.

### ● Method 2: Selecting the Use Option for the CIP Message Server

**1** Connect the Sysmac Studio to the Controller.

**2** Select **Configurations and Setup** – **Controller Setup** – **Built-in EtherNet/IP Port Settings** – **CIP Settings**.

**3** Change the setting to select the **Use** Option for **CIP Message Server**.



CIP Message Server ● Use ○ Do not use

### ● Method 3: Configuring Packet Filter Settings to Allow Packets Used by Sysmac Studio's EtherNet/IP Connection Settings

**1** Connect the Sysmac Studio to the Controller.

**2** Select **Configurations and Setup** – **Controller Setup** – **Built-in EtherNet/IP Port Settings** – **TCP/IP Settings**.

**3** Enter the settings for **Packet Filter** to allow packets used by Sysmac Studio's EtherNet/IP connection settings. Refer to *Packet Filter* in *TCP/IP Settings Display* in the *NJ/NX-series CPU Unit Built-in EtherNet/IP Port User's Manual* (Cat. No. W506) for detailed settings.

### ● Method 4: Cycle the Power Supply to the Controller

Cycle the power supply to the NJ-series Controller and transfer/monitor the EtherNet/IP connections settings again.

**Note** You may need to cycle the power supply when reflecting the changes in the IP address of the built-in EtherNet/IP port or executing Transfer to Controller.

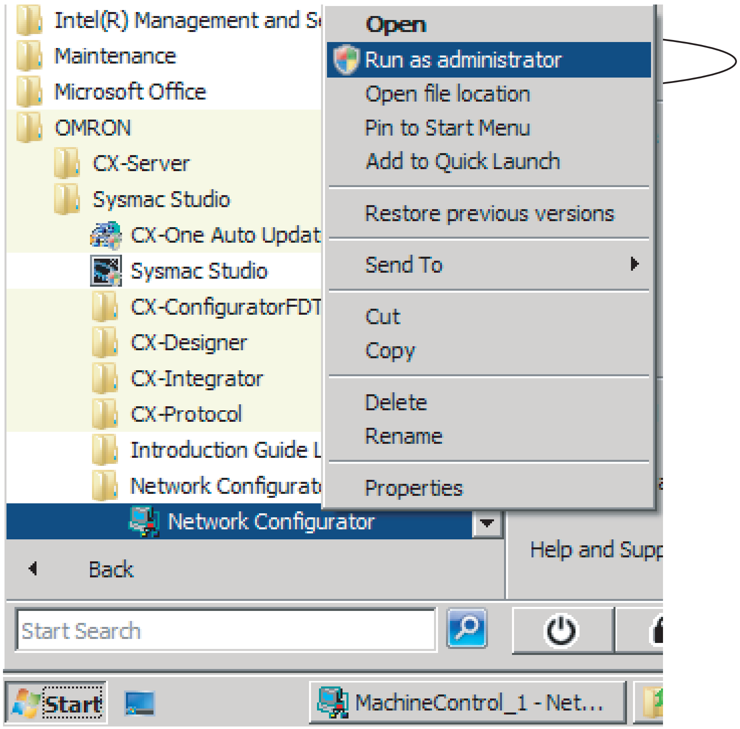
# A-3 EDS File Management

This section describes the EDS file management functions used in the Network Configurator.

### Precautions for Correct Use

For Windows Vista or Windows 7, we recommend that you start the Network Configurator as the administrator. Otherwise, the following condition will result due to user management for Windows security functions. The results of the following operations are not applied for logins with other user accounts and must be repeated: installing, creating, and deleting EDS files, and creating EDS index files. You can run the Network Configurator as the administrator with the following procedure.

1. Select the **Network Configurator** from the **Start** Menu, and then right-click.
2. Select ***Run as administrator*** from the pop-up menu that is displayed.



## A-3-1 Installing EDS Files

### EDS File – Install

The Network Configurator can support new devices if the proper EDS files are installed.

To install the EDS file, use the following procedure.

1. Select ***EDS File*** – ***Install***.
   The Install EDS File Dialog Box is displayed.
2. Select the EDS file to install and click the **Open** Button. Next, select the icon file (*.ico). The EDS file is added to the Hardware List as a new device. If the hardware already exists, the new Hardware List will overwrite the previous one. If the hardware versions are different, a hardware device is added to the Hardware List for each version.

## A-3-2 Creating EDS Files

### EDS File – Create

The EDS files are required by the Network Configurator to create a network configuration. To create an EDS file, use the following procedure.

1. Select ***EDS File*** – ***Create.***
2. Set the device information. You can obtain the device information from the device on the network if the network is online.
3. The device is added to the Hardware List as a new device, just like when you install an EDS file.

**Additional Information**

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

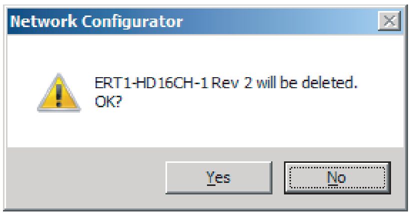
## A-3-3 Deleting EDS Files

### EDS File – Delete

To delete an EDS file, use the following procedure.

**1** Select the device from the Hardware List.

**2** Select ***EDS File – Delete.***
The following confirmation dialog box is displayed.



Network Configurator

ERT1-HD16CH-1 Rev 2 will be deleted.
OK?

**3** Click the **Yes** Button.
The selected device is deleted from the Hardware List together with the EDS file.

## A-3-4 Saving EDS Files

### EDS File – Save

To save the EDS file, use the following procedure.

**1** Select the target hardware device in the Hardware List, and then select ***EDS File – Save***.

A Save EDS File Dialog Box is displayed.

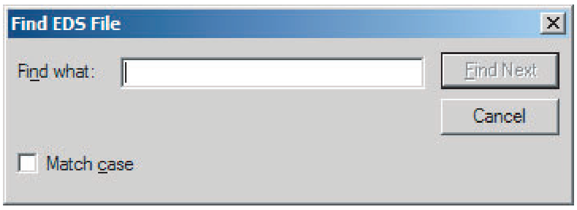
**2** Input the folder and file names and click the **Save** Button.
The EDS file is saved.

## A-3-5 Searching EDS Files

### EDS File – Find

To search the devices (EDS files) displayed in the Hardware List, use the following procedure.

**1** Select ***EDS file – Find***.
The following dialog box is displayed.



Find EDS File

Find what:

**2** Input the character string to search for and click the **Find Next** Button.

**3** When a matching device is found, the cursor moves to that position.

**4** To quit the search operation, click the **Cancel** Button.

**Additional Information**

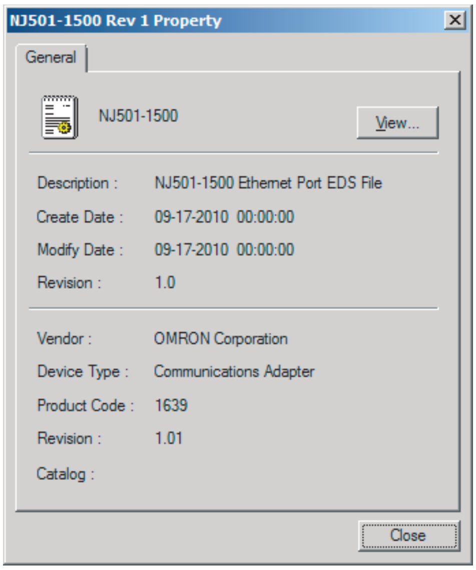
- The device is found only if it is located below the present cursor position in the Hardware List.
- To search all the devices, select *Hardware* in the Hardware List before you perform the search procedure.

## A-3-6 Displaying EDS File Properties

### EDS File – Property

To display the properties of the EDS file, use the following procedure.

1. Select the desired hardware (device) from the Hardware List.
2. Select ***EDS File*** – ***Property***.
   The following dialog box is displayed.



NJ501-1500 Rev 1 Property

General

NJ501-1500

Description : NJ501-1500 Ethernet Port EDS File
Create Date : 09-17-2010 00:00:00
Modify Date : 09-17-2010 00:00:00
Revision : 1.0

Vendor : OMRON Corporation
Device Type : Communications Adapter
Product Code : 1639
Revision : 1.01
Catalog :

The time and date when the EDS file was created is displayed, along with the device information.

## A-3-7 Creating EDS Index Files

### EDS File – Create EDS Index File

To manually add an EDS file or if a device is not displayed correctly in the hardware list, use the following procedure to recreate the EDS index file. (This applies to Network Configurator version 3.30 or higher.)

1. Select ***EDS File*** – ***Create EDS Index File***.
2. Restart the Network Configurator.

# A-4 Precautions for Using the Network Configurator on Windows XP, Windows Vista, or Windows 7

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/NX Series Ethernet Direct I/F***
- If you select ***Option*** − ***Select Interface*** − ***NJ/NX 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-4-1 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.

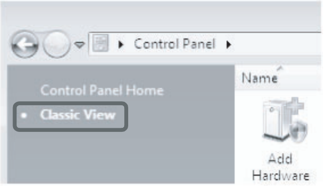
2 Click the **Unblock** Button.

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.

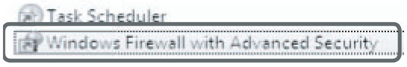
## Windows Vista or Windows 7

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.

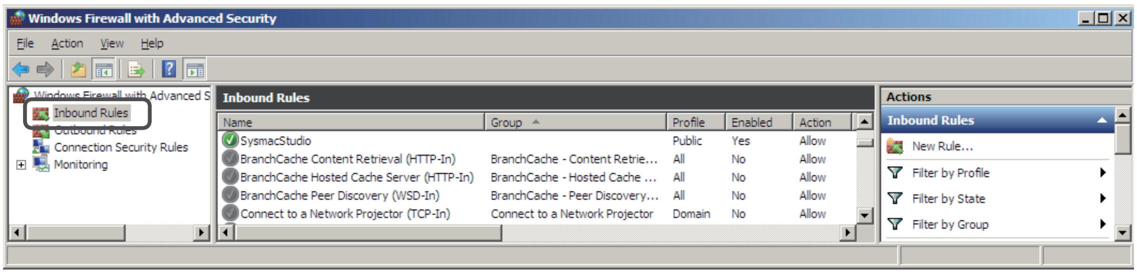
**1** Select *Control Panel* from the Windows Start Menu and change the display to Classic View.



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



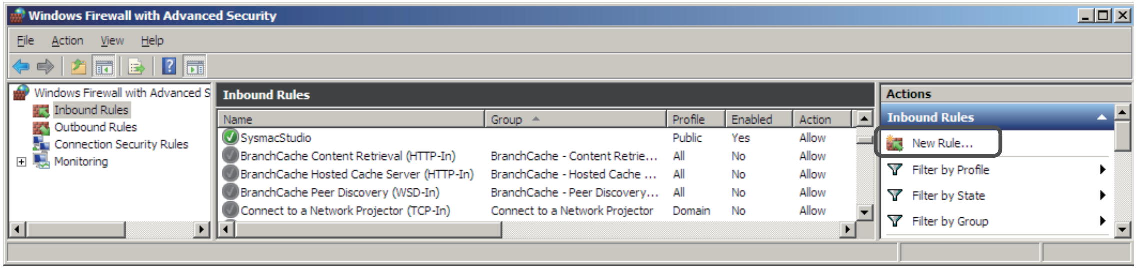
**3** Select *Inbound Rules* under *Windows Firewall with Advanced Security on Local Computer* on the left side of the Windows Firewall with Advanced Security Dialog Box.



Inbound Rules

| Name | Group | Profile | Enabled | Action |
|---|---|---|---|---|
| SysmacStudio |  | Public | Yes | Allow |
| BranchCache Content Retrieval (HTTP-In) | BranchCache - Content Retrie... | All | No | Allow |
| BranchCache Hosted Cache Server (HTTP-In) | BranchCache - Hosted Cache ... | All | No | Allow |
| BranchCache Peer Discovery (WSD-In) | BranchCache - Peer Discovery... | All | No | Allow |
| Connect to a Network Projector (TCP-In) | Connect to a Network Projector | Domain | No | Allow |

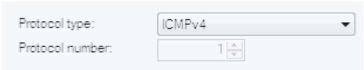
**4** Select *New Rule* under *Inbound Rules* in the Actions Area on the right side of the dialog box.



Inbound Rules

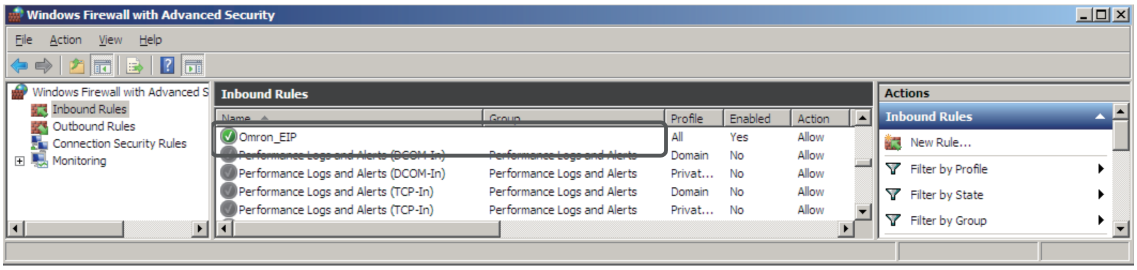
| Name | Group | Profile | Enabled | Action |
|---|---|---|---|---|
| SysmacStudio |  | Public | Yes | Allow |
| BranchCache Content Retrieval (HTTP-In) | BranchCache - Content Retrie... | All | No | Allow |
| BranchCache Hosted Cache Server (HTTP-In) | BranchCache - Hosted Cache ... | All | No | Allow |
| BranchCache Peer Discovery (WSD-In) | BranchCache - Peer Discovery... | All | No | Allow |
| Connect to a Network Projector (TCP-In) | Connect to a Network Projector | Domain | No | Allow |

**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.<br><br>Protocol type: ICMPv4<br>Protocol number: 1 |
| 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.

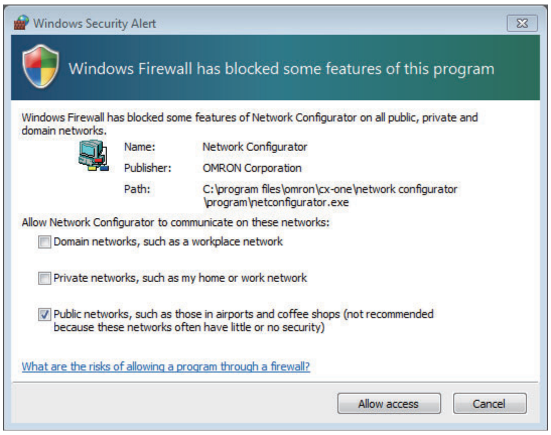


Windows Firewall with Advanced Security

| Name | Group | Profile | Enabled | Action |
|---|---|---|---|---|
| Omron_EIP | | All | Yes | Allow |
| Performance Logs and Alerts (DCOM-In) | Performance Logs and Alerts | Domain | No | Allow |
| Performance Logs and Alerts (DCOM-In) | Performance Logs and Alerts | Privat... | No | Allow |
| Performance Logs and Alerts (TCP-In) | Performance Logs and Alerts | Domain | No | Allow |
| Performance Logs and Alerts (TCP-In) | Performance Logs and Alerts | Privat... | No | Allow |

**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 Security Alert

Windows Firewall has blocked some features of this program

Windows Firewall has blocked some features of Network Configurator on all public, private and domain networks.

Name: Network Configurator

Publisher: OMRON Corporation

Path: C:\program files\omron\cx-one\network configurator \program\netconfigurator.exe

Allow Network Configurator to communicate on these networks:

□ Domain networks, such as a workplace network

□ Private networks, such as my home or work network

☑ Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

What are the risks of allowing a program through a firewall?

(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-5 Variable Memory Allocation Methods

You must be aware of the way in which memory is allocated to variables to align the memory locations of the members of structure or union variables with variables in other devices. Adjustments are necessary mainly when structure variables are used in the following type of communications with other devices.

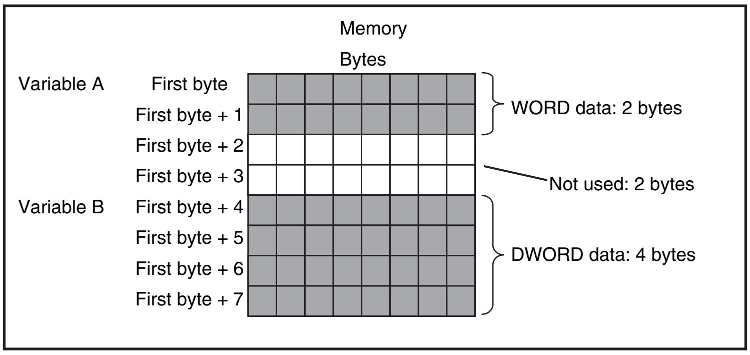
- When using EtherNet/IP tag data links or CIP messages to access variables between NJ-series CPU Units and other CPU Units
- When using structure variables to exchange data with devices other than CPU Units, such as ID Tags

## A-5-1 Variable Memory Allocation Rules

The amount of memory and the memory locations that are allocated for a variable depend on the data type of the variable. The amount of memory and the memory locations that are allocated for array elements, structure members, and union members depend on the data types, but also on the declarations that are made for the arrays, structures, and unions.

### Data Type Alignment and Memory Allocation Amounts

The data size is determined for each data type. The data size is the minimum amount of memory that is required to store the value or values of that data type. On the other hand, memory for variables is automatically structured by the Controller for the most efficient access. Therefore, the total amount of memory that is required for variables is not necessarily the total of the data sizes of the variables. For example, if WORD and DWORD variables are declared, the total of the data sizes is six bytes, but eight bytes are allocated in memory, as shown in the following figure.



Variable Table

| Name | Data type |
|---|---|
| A | WORD |
| B | DWORD |

This information for determining the location of a variable in memory is called the alignment. The alignment is determined for each data type. The amount of memory and the memory locations for the variables are given below.

| Item | Specification |
|---|---|
| Amount of memory that is allocated | An integral multiple of the alignment. However, the minimum amount of memory is the data size. |
| Locations in memory | At an integral multiple of the alignment starting from the start of the variable in memory. |

The alignments and the amounts of memory that are allocated for the basic data types and enumerations are given below.

| Data type | Alignment [bytes] | Amount of memory that is allocated [bytes] |
|---|---|---|
| **BOOL** | 2 | 2 |
| **BYTE, USINT, or SINT** | 1 | 1 |
| **WORD, UINT, or INT** | 2 | 2 |
| **DWORD, UDINT, or DINT** | 4 | 4 |
| **LWORD, ULINT, or LINT** | 8 | 8 |
| **REAL** | 4 | 4 |
| **LREAL** | 8 | 8 |
| **TIME, DATE, TIME_OF_DAY, or DATE_AND_TIME** | 8 | 8 |
| **STRING[N+1]**[*1] | 1 | N+1 |
| **Enumerations** | 4 | 4 |

[*1] N is the maximum number of characters handled. For example, if a maximum of 10 single-byte characters are handled, the NULL character is added, so memory for 11 characters must be reserved.

The elements of arrays and the members of structures and unions are located in memory for the most efficient access. The alignments and the amounts of memory that are allocated for arrays, structures, and unions are determined by the variable declarations, as described below.

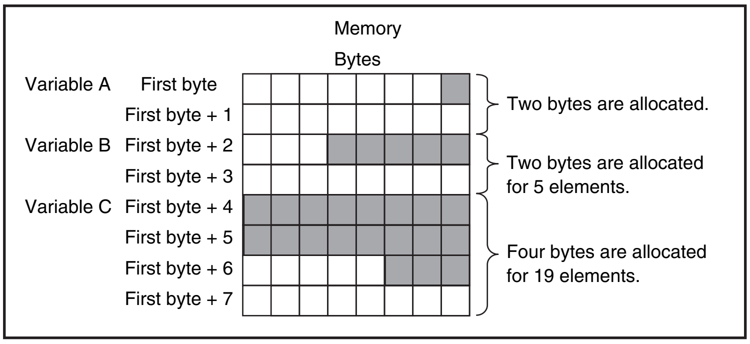
| Data type | Alignment | Amount of memory that is allocated |
|---|---|---|
| **Array** | Same as alignment of the data type of the elements | (Amount of memory that is allocated for the data type of the elements) × Number of elements[*] |
| **Structure** | The largest alignment of all of the members | The integral multiple of the alignment that is larger than the total amount of memory that is allocated when the members are arranged in order at integral multiples of the alignment of the data types of the members |
| **Union** | The largest alignment of all of the members | The largest amount of memory that is allocated for any of the members |

[*] BOOL arrays are an exception. Refer to *Precautions for Correct Use*, below, for the amount of memory that is allocated for BOOL arrays.

### Precautions for Correct Use

**Amount of Memory That Is Allocated for BOOL Arrays**

Two bytes are allocated in memory for individual BOOL variables, BOOL structure members, and BOOL union variables. However, for a BOOL array, two bytes of memory are not allocated for each element. One bit is allocated in order for each element. For the entire array, a multiple of two bytes of memory is allocated (including unused bits).



Variable Table

| Name | Data type |
|---|---|
| A | BOOL |
| B | ARRAY[1..5]OF BOOL |
| C | ARRAY[0..18]OF BOOL |

Therefore, the following formula gives the amount of memory that is allocated for a BOOL array. For 1 to 16 elements, 2 bytes are allocated. For 17 to 32 elements, 4 bytes are allocated.

$$\text{Amount of memory} = 2\left\lfloor \frac{\text{Number of elements} - 1}{16} \right\rfloor + 2$$

Truncate the decimal portion of the result of the calculation in brackets.

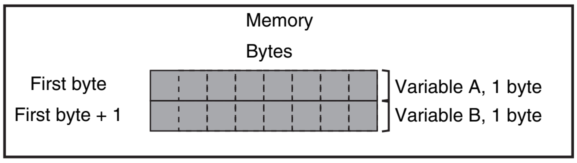
Specific examples of the rules for memory allocation for variables of each data type are given below.

## Basic Data Types

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

One byte of memory is allocated for the one-byte alignment.

Example: Two consecutive BYTE variables



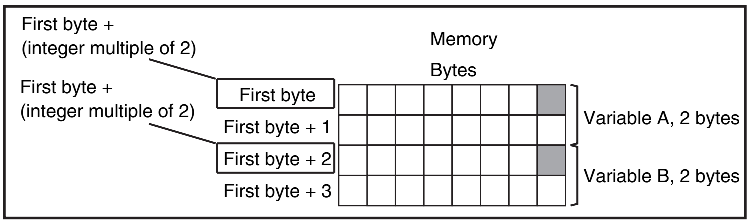
Variable Table

| Name | Data type |
|---|---|
| A | BYTE |
| B | BYTE |

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

Two bytes of memory are allocated for the two-byte alignment.

Example: Two consecutive BOOL variables



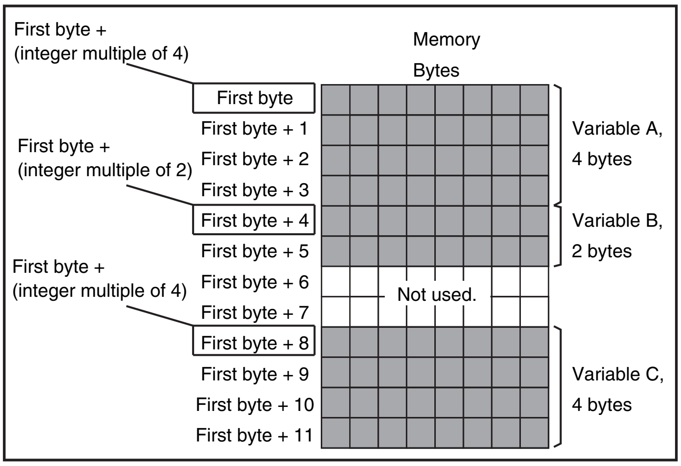
Variable Table

| Name | Data type |
|---|---|
| A | BOOL |
| B | BOOL |

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

Four bytes of memory are allocated for the four-byte alignment.
The location of the first byte of data in memory is an integer multiple of four bytes. Therefore, if a variable with a two-byte alignment, such as WORD data, is inserted, two bytes of unused memory will remain.

Example: Consecutive variables in the following order: DWORD, WORD, and DWORD



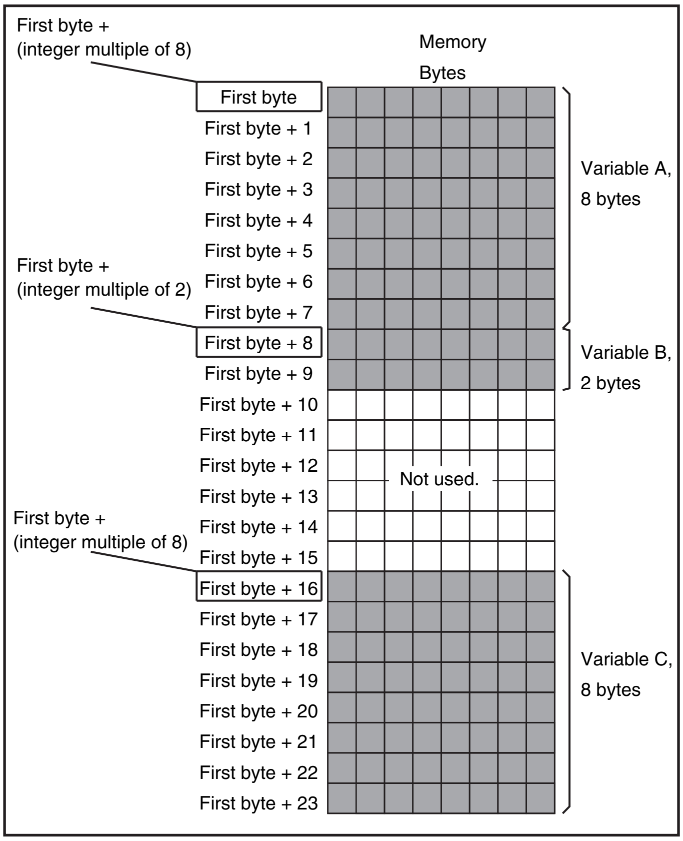
Variable Table

| Name | Data type |
|---|---|
| A | DWORD |
| B | WORD |
| C | DWORD |

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

Eight bytes of memory are allocated for the eight-byte alignment.
The location of the first byte of data in memory is an integer multiple of eight bytes. Therefore, if a variable with a two-byte alignment, such as WORD data, is inserted, six bytes of unused memory will remain. If a variable with a four-byte alignment, such as DWORD data, is inserted, four bytes of unused memory will remain.

Example: Consecutive variables in the following order: LWORD, WORD, and LWORD



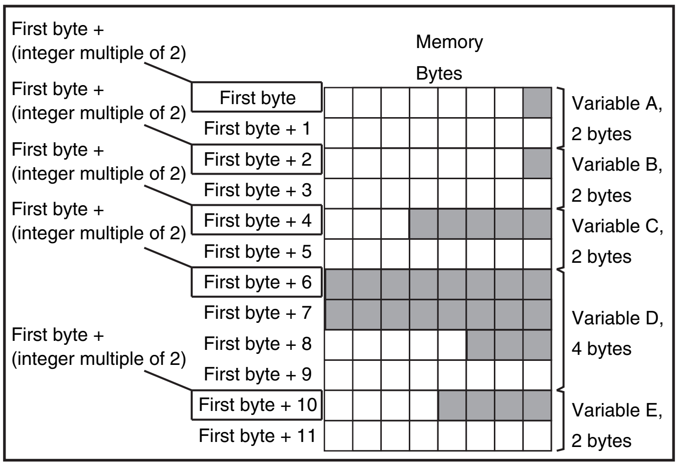
Variable Table

| Name | Data type |
|---|---|
| A | LWORD |
| B | WORD |
| C | LWORD |

## Arrays

A continuous section of memory is allocated for the elements of the array based on the data size of the data type of the array variable. The alignment of an array is the same as alignment of the data type of the elements.

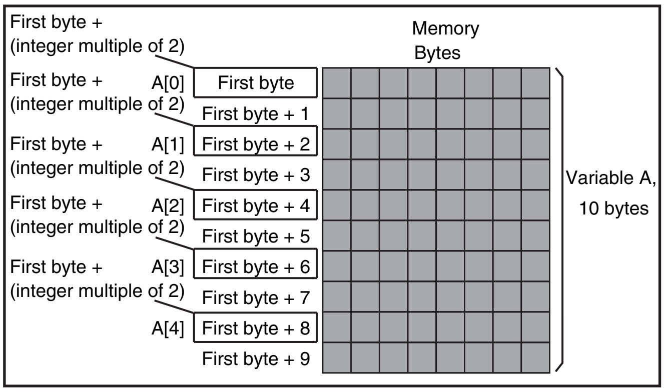
Example: Continuous variables in the following order: two BOOL variable, one BOOL array with five elements, one BOOL array with 19 elements, and one BOOL array with four elements



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 |

Example: INT array with five elements



Variable Table

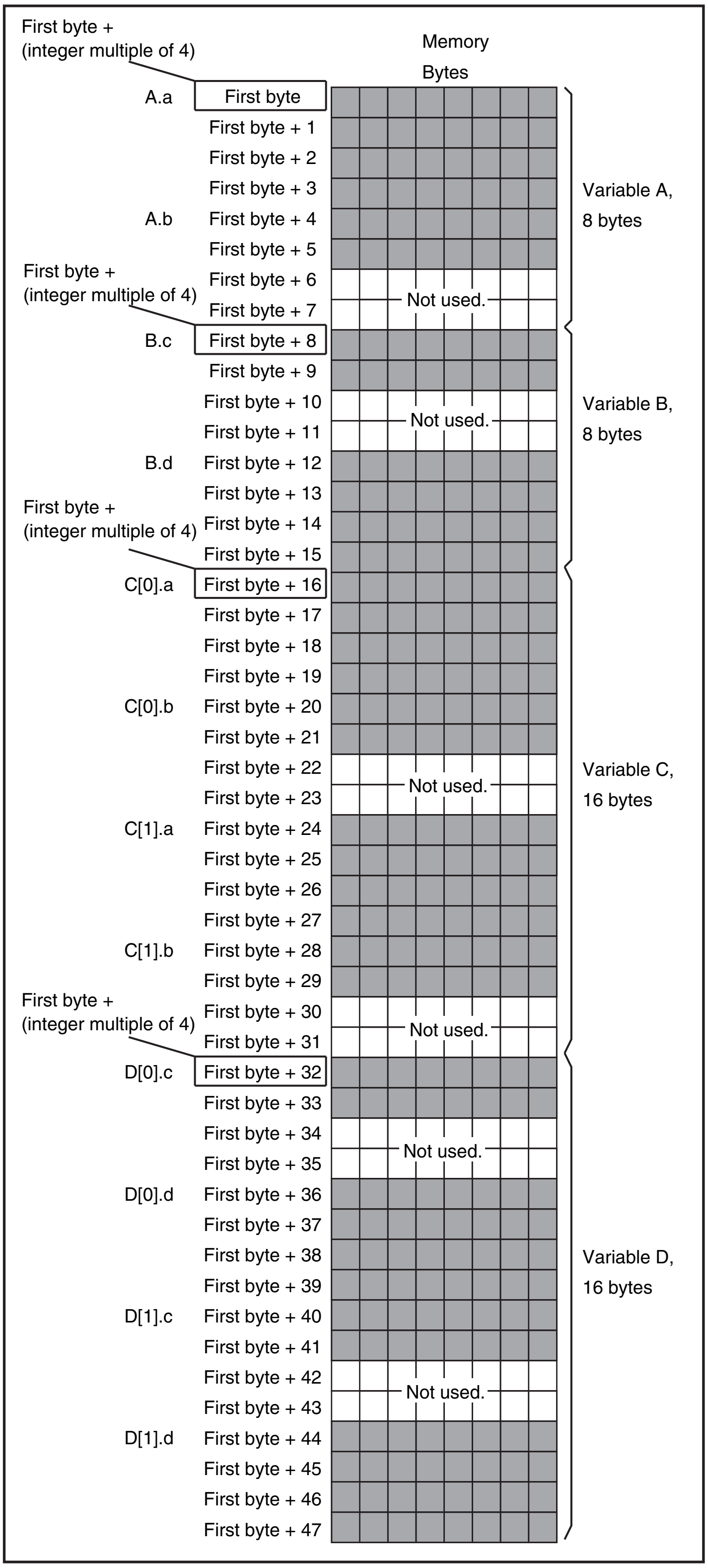
| Name | Data type |
|---|---|
| Variable A | ARRAY[0..4] OF INT |

## Structures

For a structure variable, the members are located in memory in the order that they are declared. Each member is located at an integer multiple of the alignment of the data type of the member. Therefore, there can be unused memory between members or at the end of members. The alignment of a structure is the largest alignment of all of the members. The amount of memory that is allocated is the integral multiple of the alignment that is larger than the total amount of memory that is allocated when the members are arranged in order at integral multiples of the alignment of the data types of the members.

Example: The alignments and the amounts of memory that are allocated for the four variable declarations given in the following figure are given in the following table.

| Variable | Alignment [bytes] | Amount of memory that is allocated [bytes] |
|---|---|---|
| A | 4 | 8 |
| B | 4 | 8 |
| C | 4 | 16 |
| D | 4 | 16 |



Data Type Definitions

| Name | Data type |
|---|---|
| Structure *STR_A* | STRUCT |
| a | DINT |
| b | INT |

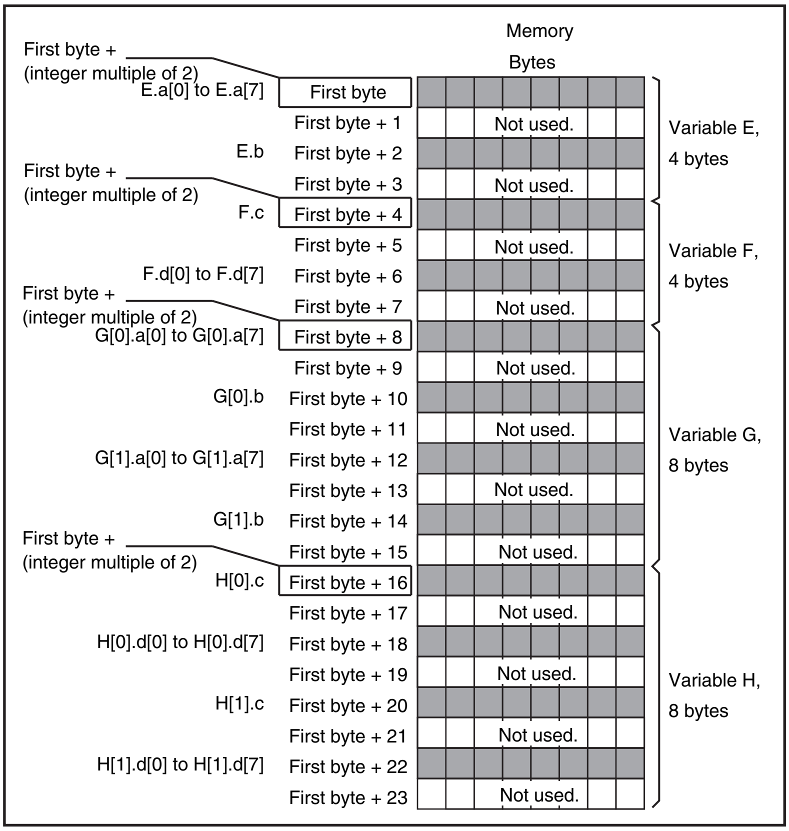
| Name | Data type |
|---|---|
| Structure *STR_B* | STRUCT |
| c | INT |
| d | DINT |

Variable Table

| Name | Data type |
|---|---|
| Variable A | Structure *STR_A* |
| Variable B | Structure *STR_B* |
| Variable C | ARRAY[0..1] OF STR_A |
| Variable D | ARRAY[0..1] OF STR_B |

Example: The alignments and the amounts of memory that are allocated for the four variable declarations given in the following figure are given in the following table.

| Variable | Alignment [bytes] | Amount of memory that is allocated [bytes] |
|---|---|---|
| E | 2 | 4 |
| F | 2 | 4 |
| G | 2 | 8 |
| H | 2 | 8 |



Data Type Definitions

| Name | Data type |
|---|---|
| Structure *STR_C* | STRUCT |
| a | ARRAY[0..7] OF BOOL |
| b | BYTE |

| Name | Data type |
|---|---|
| Structure *STR_D* | STRUCT |
| c | BYTE |
| d | ARRAY[0..7] OF BOOL |

Variable Table

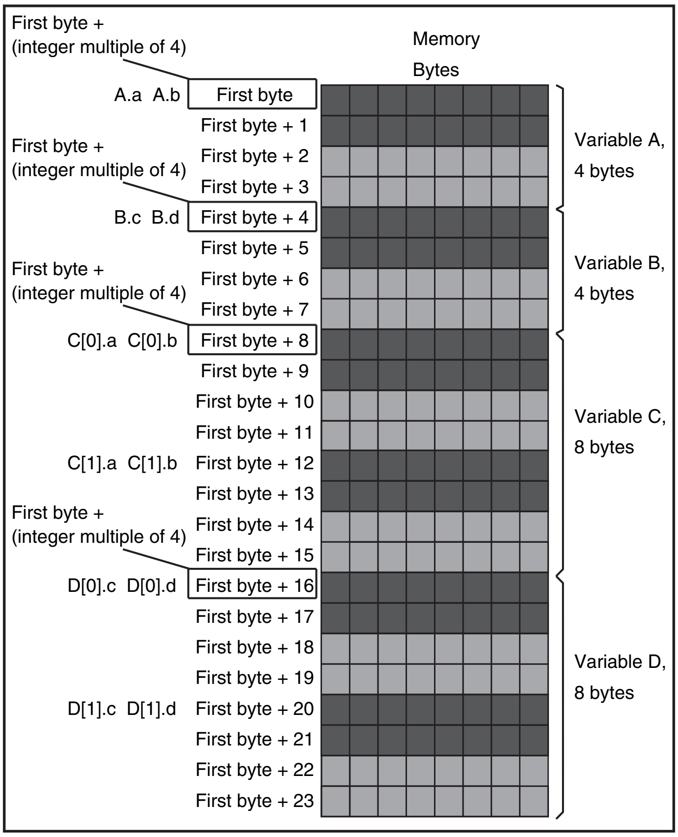
| Name | Data type |
|---|---|
| Variable E | Structure *STR_C* |
| Variable F | Structure *STR_D* |
| Variable G | ARRAY[0..1] OF STR_C |
| Variable H | ARRAY[0..1] OF STR_D |

## Unions

For a union variable, the members overlap in the same memory locations. The alignment of a union is largest alignment of all of the members. The amount of memory that is allocated is the largest amount of memory that is allocated for any of the members.

Example: The alignments and the amounts of memory that are allocated for the four variable declarations given in the following figure are given in the following table.

| Variable | Alignment [bytes] | Amount of memory that is allocated [bytes] |
|---|---|---|
| A | 4 | 4 |
| B | 4 | 4 |
| C | 4 | 8 |
| D | 4 | 8 |



Data Type Definitions

| Name | Data type |
|---|---|
| Union *UNI_A* | UNION |
| a | DWORD |
| b | WORD |

| Name | Data type |
|---|---|
| Union *UNI_B* | UNION |
| c | WORD |
| d | DWORD |

Variable Table

| Name | Data type |
|---|---|
| Variable A | Union *UNI_A* |
| Variable B | Union *UNI_B* |
| Variable C | ARRAY[0..1] OF UNI_A |
| Variable D | ARRAY[0..1] OF UNI_B |

## A-5-2 Important Case Examples

When you exchange structure variable data between an NJ-series CPU Unit and a remote device, you must align the memory configuration of the structure variable members with those of the remote device. This section describes what to do in either the NJ-series CPU Unit or in the remote device.

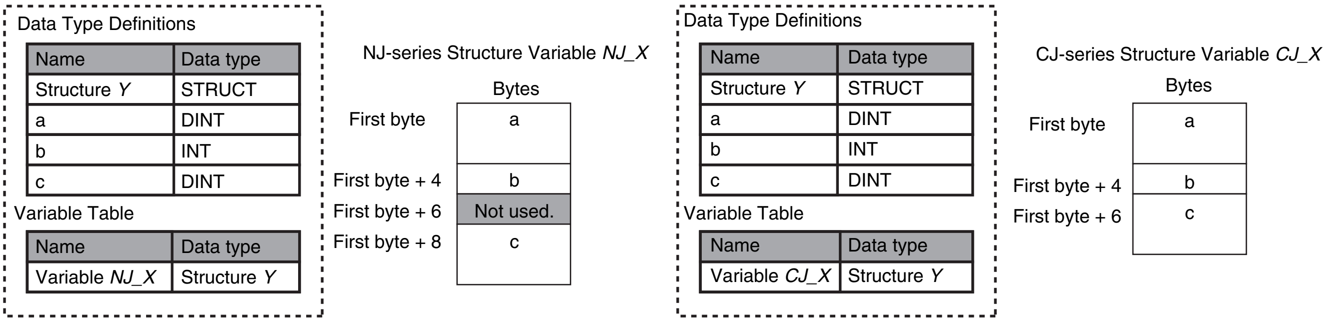
**Additional Information**

This is not necessary when you exchange data between NJ-series CPU Units.

### Aligning the Memory Configuration with a Remote Device

There are two methods that you can use to align the memory configuration with a remote device. For example, the differences in the memory configuration for structure variables between an NJ-series CPU Unit and a CJ-series CPU Unit are shown below.

This section describes how to align the memory configuration for these Units.



Data Type Definitions

| Name | Data type |
|---|---|
| Structure *Y* | STRUCT |
| a | DINT |
| b | INT |
| c | DINT |

Variable Table

| Name | Data type |
|---|---|
| Variable *NJ_X* | Structure *Y* |

NJ-series Structure Variable *NJ_X*

|  | Bytes |
|---|---|
| First byte | a |
| First byte + 4 | b |
| First byte + 6 | Not used. |
| First byte + 8 | c |

Data Type Definitions

| Name | Data type |
|---|---|
| Structure *Y* | STRUCT |
| a | DINT |
| b | INT |
| c | DINT |

Variable Table

| Name | Data type |
|---|---|
| Variable *CJ_X* | Structure *Y* |

CJ-series Structure Variable *CJ_X*

|  | Bytes |
|---|---|
| First byte | a |
| First byte + 4 | b |
| First byte + 6 | c |

#### ● Method 1: Changing the Memory Configuration of the Structure Variable in the NJ-series CPU Unit

With an NJ-series CPU Unit, you can specify member offsets to change the memory configuration of the members of a structure variable. You can change the memory configuration of the members of a structure variable in the NJ-series CPU Unit so that it is the same as the memory configuration in a remote device that the CPU Unit will communicate with. Specify the member offsets for a structure variable when you register the structure data type.

To communicate with a CJ-series CPU Unit, you can set the offset type to *CJ* to automatically use the CJ-series memory structure. You can set the offset type to *User* to freely set your own offsets.

### Version Information

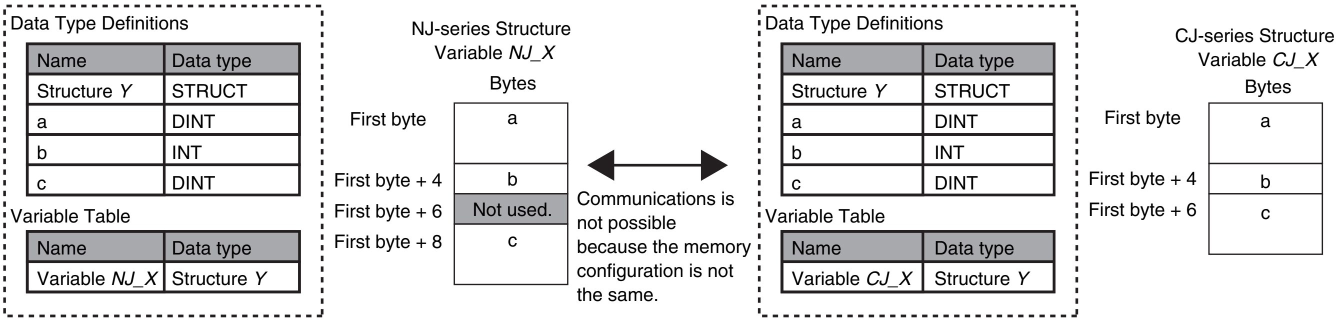
The following table gives the unit version of the CPU Units and the Sysmac Studio version that are required to specify member offsets.

| Unit version of CPU Unit | Sysmac Studio version | | |
|---|---|---|---|
| | 1.01 or lower | 1.02 | 1.03 or higher |
| 1.01 or later | Not possible. | Possible.* | Possible. |
| 1.00 | Not possible. | Not possible. | Not possible. |

* You cannot select the memory offset type. You can set member offsets.

If you change the memory configuration of a structure variable by setting offsets, you must make the same changes for the same structure variable in other NJ-series CPU Units on the network.
Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No W504-E1-03 or higher) for the procedure to change the memory configuration of a structure variable.

Example: The following example shows how the memory configuration of the structure variable members in the NJ-series CPU Unit is changed to match the memory configuration of the structure variable members in the CJ-series CPU Unit.



Data Type Definitions

| Name | Data type |
|---|---|
| Structure *Y* | STRUCT |
| a | DINT |
| b | INT |
| c | DINT |

Variable Table

| Name | Data type |
|---|---|
| Variable *NJ_X* | Structure *Y* |

Communications is not possible because the memory configuration is not the same.

Data Type Definitions

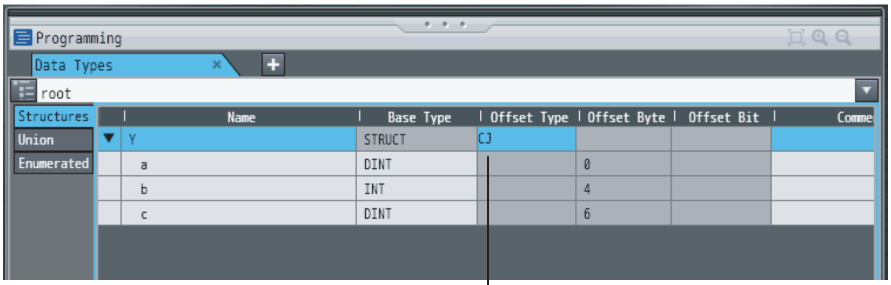
| Name | Data type |
|---|---|
| Structure *Y* | STRUCT |
| a | DINT |
| b | INT |
| c | DINT |

Variable Table

| Name | Data type |
|---|---|
| Variable *CJ_X* | Structure *Y* |

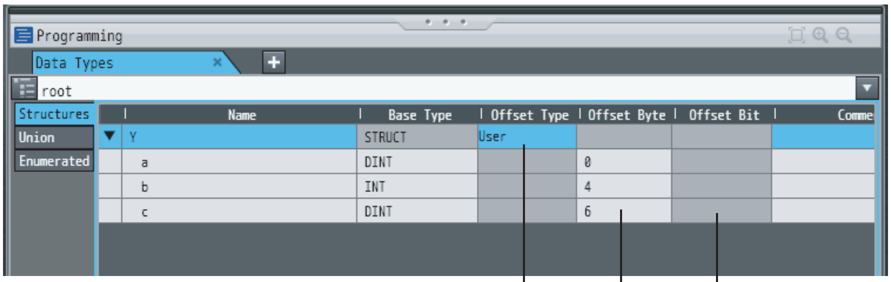
To align the memory configurations in the NJ-series and CJ-series CPU Units, offsets are set in the Sysmac Studio.

Here, the following offsets are set for member c of data type Y of the structure variable NJ_X.



| Name | Base Type | Offset Type | Offset Byte | Offset Bit | Comme |
|---|---|---|---|---|---|
| ▼ Y | STRUCT | CJ | | | |
| a | DINT | | 0 | | |
| b | INT | | 4 | | |
| c | DINT | | 6 | | |

(1) Offset type is set to *CJ*.

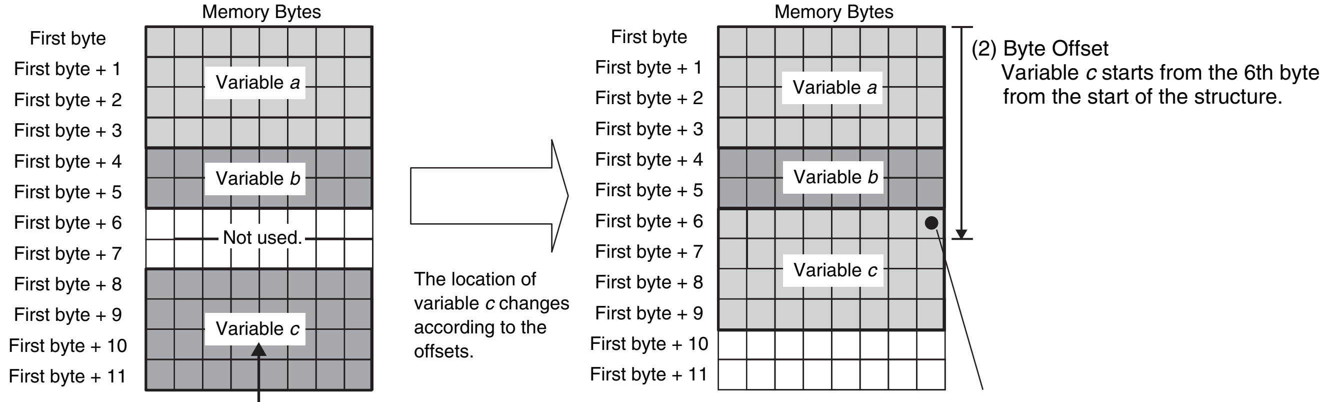


| Name | Base Type | Offset Type | Offset Byte | Offset Bit | Comme |
|---|---|---|---|---|---|
| ▼ Y | STRUCT | User | | | |
| a | DINT | | 0 | | |
| b | INT | | 4 | | |
| c | DINT | | 6 | | |

(3) Bit Offset
Set the location of the first bit of the member variable.

(2) Byte Offset
Set the location of the first byte of the member from the beginning of the structure variable.

(1) Offset Type
Specify *User.*



The location of variable *c* changes according to the offsets.

(2) Byte Offset
Variable *c* starts from the 6th byte from the start of the structure.

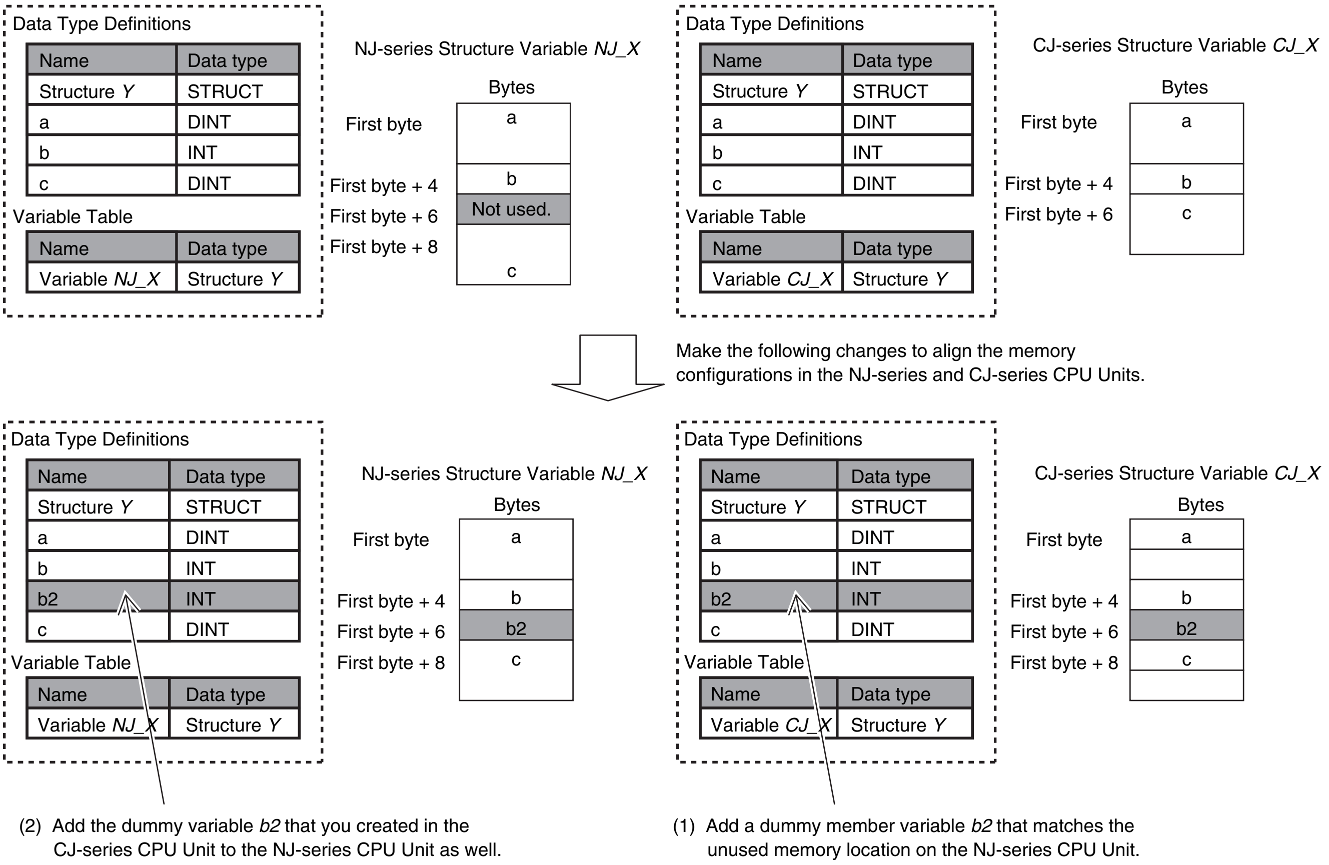
Set a byte offset of 6 and a bit offset of 0 (no offset) for variable *c*.

(3) Bit Offset
Variable *c* starts from the 0th bit from the start of the byte.

### Method 2: Changing the Memory Configuration of the Structure Variable in the Remote Device

You can insert a member into the structure variable of the remote device to change it to match the memory configuration of the structure variable in the NJ-series CPU Unit. Both the memory configuration and the data types must be the same between the two structure variables. You therefore need to create the same members in both the remote device and the NJ-series CPU Unit.

Example: The following example shows how the memory configuration of the structure variable in the CJ-series CPU Unit is changed to match the memory configuration of the structure variable in the NJ-series CPU Unit.



Data Type Definitions

| Name | Data type |
|---|---|
| Structure *Y* | STRUCT |
| a | DINT |
| b | INT |
| c | DINT |

Variable Table

| Name | Data type |
|---|---|
| Variable *NJ_X* | Structure *Y* |

Data Type Definitions

| Name | Data type |
|---|---|
| Structure *Y* | STRUCT |
| a | DINT |
| b | INT |
| c | DINT |

Variable Table

| Name | Data type |
|---|---|
| Variable *CJ_X* | Structure *Y* |

Make the following changes to align the memory configurations in the NJ-series and CJ-series CPU Units.

Data Type Definitions

| Name | Data type |
|---|---|
| Structure *Y* | STRUCT |
| a | DINT |
| b | INT |
| b2 | INT |
| c | DINT |

Variable Table

| Name | Data type |
|---|---|
| Variable *NJ_X* | Structure *Y* |

Data Type Definitions

| Name | Data type |
|---|---|
| Structure *Y* | STRUCT |
| a | DINT |
| b | INT |
| b2 | INT |
| c | DINT |

Variable Table

| Name | Data type |
|---|---|
| Variable *CJ_X* | Structure *Y* |

(2) Add the dummy variable *b2* that you created in the CJ-series CPU Unit to the NJ-series CPU Unit as well.

(1) Add a dummy member variable *b2* that matches the unused memory location on the NJ-series CPU Unit.

# A-6 Precautions When Accessing External Outputs in 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 will be overwritten by the execution results of the user program. The value that is written from the tag data link or communications instruction will therefore not be output to the external device.
  The following types of variable are assigned to the external outputs.
  - The devices variables (or global variables) that are assigned to an I/O port of a CJ-series Basic Output Unit
  - The devices variables (or global variables) that are assigned to an I/O port of an EtherCAT output slave
  - The 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.

# A-7 Differences in Available Functions Depending on the CPU Unit (NJ or CJ Series)

Some of the specifications when an EtherNet/IP Unit is connected to a CJ-series CPU Unit are different from the specifications when the EtherNet/IP Unit is connected to an NJ-series CPU Unit. A list of these differences is provided below.

## A-7-1 Functional Differences

There are differences in function support, function names, and specifications between when an EtherNet/IP Unit is connected to a CJ-series CPU Unit and when the EtherNet/IP Unit is connected to an NJ-series CPU Unit.

### Unsupported Functions

The following functions cannot be used when the EtherNet/IP Unit is connected to an NJ-series CPU Unit.

- EtherNet/IP Datalink Tool in the Network Configurator
- Going online with the CPU Unit from the Sysmac Studio through the EtherNet/IP Unit (You can go online from the Network Configurator.)

### Functions with Different Names

The names of the following functions are different between when an EtherNet/IP Unit is connected to a CJ-series CPU Unit and when the EtherNet/IP Unit is connected to an NJ-series CPU Unit. The corresponding names are listed in the following table.

| Item | EtherNet/IP Unit connected to a CJ-series CPU Unit | EtherNet/IP Unit connected to an NJ-series CPU Unit |
|---|---|---|
| Backup functions | Simple backup function | SD Memory Card backup function* |
| | Backup with PLC backup tool | Sysmac Studio Controller backups* |

* An NJ-series CPU Unit with unit version 1.03 or later and Sysmac Studio version 1.04 or higher are required.

**Precautions for Correct Use**

For NJ-series CPU Units, use the following combination of Ethernet/IP Units and the Sysmac Studio.

| EtherNet/IP Unit | NJ-series CPU Unit | Sysmac Studio |
|---|---|---|
| CJ1W-EIP21 with unit version 2.1 or later | Unit version 1.01 or later | Version 1.02 or higher |
| CJ1W-EIP21S with unit version.1.0<br>Lot. number: 241001□ or later | Unit version 1.67 or later | Version 1.60 or higher |

## A-7-2 Differences in Access Methods from the User Program

When the EtherNet/IP Unit is used with an NJ-series CPU Unit, device variables for CJ-series Units with AT specifications in memory for CJ-series Units are used in the user program to access the functions of the EtherNet/IP Unit. The word addresses, bit numbers, and device variables for the CJ-series Unit in I/O memory in the CJ-series CPU Unit and in the memory for CJ-series Units in the NJ-series CPU Unit are given in the following tables.

First word allocated in the CIO Area: n = 1,500 + Unit number × 25 (The unit number is from 0 to 15.)

First word allocated in the DM Area: m = D30000 + Unit number × 100 (The unit number is from 0 to 15.)

### CIO Area Words Allocated to CPU Bus Units

- **CIO n: Software Switches**

The device variable that corresponds to all of the bits in CIO n is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n | 00 to 15 | *_SoftSwCmd | Software Switches<br>(The functions of bits 00 to 15 in CIO n correspond to the functions of the device variable given on the left.) |

The device variables that correspond to bits 00 to 15 in CIO n are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n | 00 and 01 | --- | Not used. |
| | 2 | *_TDLStartCmd | Tag Data Link Start Bit |
| | 3 | --- | Not used. |
| | 4 | *_TDLStopCmd | Tag Data Link Stop Bit |
| | 5 | *_AdjTmCmd | Adjust Clock Bit |
| | 06 to 09 | --- | Not used. |
| | 10 | *_SktForceCloseCmd | Force Socket Close Switch [*1] |
| | 11 to 15 | --- | Not used. |

*1 This switch is provided only for the CJ1W-EIP21S.

- **CIO n+2 to n+5: Target Node PLC Operating Flags**

The device variable that corresponds to all of the bits in CIO n+2 to n+5 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+2 to n+5 | 00 to 15 in each word | *_TargetPLCMdSta | Target Node PLC Operating Flags<br>(The functions of bits 00 to 15 in these words and the functions of the device variable given on the left correspond as given below.)<br>• Bits 00 to 15 of CIO n+2 correspond to bits 00 to 15 of the device variable given on the left.<br>• Bits 00 to 15 of CIO n+3 correspond to bits 16 to 31 of the device variable given on the left.<br>• Bits 00 to 15 of CIO n+4 correspond to bits 32 to 47 of the device variable given on the left.<br>• Bits 00 to 15 of CIO n+5 correspond to bits 48 to 63 of the device variable given on the left. |

The device variables that correspond to bits 00 to 15 in CIO n+2 to n+5 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| CIO n+2 | 0 | *_TargetPLCMdSta[0] | Target Node PLC Operating Flag for Node Address 0 |
| | 1 | *_TargetPLCMdSta[1] | Target Node PLC Operating Flag for Node Address 1 |
| | 2 | *_TargetPLCMdSta[2] | Target Node PLC Operating Flag for Node Address 2 |
| | 3 | *_TargetPLCMdSta[3] | Target Node PLC Operating Flag for Node Address 3 |
| | 4 | *_TargetPLCMdSta[4] | Target Node PLC Operating Flag for Node Address 4 |
| | 5 | *_TargetPLCMdSta[5] | Target Node PLC Operating Flag for Node Address 5 |
| | 6 | *_TargetPLCMdSta[6] | Target Node PLC Operating Flag for Node Address 6 |
| | 7 | *_TargetPLCMdSta[7] | Target Node PLC Operating Flag for Node Address 7 |
| CIO n+2 | 8 | *_TargetPLCMdSta[8] | Target Node PLC Operating Flag for Node Address 8 |
| | 9 | *_TargetPLCMdSta[9] | Target Node PLC Operating Flag for Node Address 9 |
| | 10 | *_TargetPLCMdSta[10] | Target Node PLC Operating Flag for Node Address 10 |
| | 11 | *_TargetPLCMdSta[11] | Target Node PLC Operating Flag for Node Address 11 |
| | 12 | *_TargetPLCMdSta[12] | Target Node PLC Operating Flag for Node Address 12 |
| | 13 | *_TargetPLCMdSta[13] | Target Node PLC Operating Flag for Node Address 13 |
| | 14 | *_TargetPLCMdSta[14] | Target Node PLC Operating Flag for Node Address 14 |
| | 15 | *_TargetPLCMdSta[15] | Target Node PLC Operating Flag for Node Address 15 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| CIO n+3 | 0 | *_TargetPLCMdSta[16] | Target Node PLC Operating Flag for Node Address 16 |
| | 1 | *_TargetPLCMdSta[17] | Target Node PLC Operating Flag for Node Address 17 |
| | 2 | *_TargetPLCMdSta[18] | Target Node PLC Operating Flag for Node Address 18 |
| | 3 | *_TargetPLCMdSta[19] | Target Node PLC Operating Flag for Node Address 19 |
| | 4 | *_TargetPLCMdSta[20] | Target Node PLC Operating Flag for Node Address 20 |
| | 5 | *_TargetPLCMdSta[21] | Target Node PLC Operating Flag for Node Address 21 |
| | 6 | *_TargetPLCMdSta[22] | Target Node PLC Operating Flag for Node Address 22 |
| | 7 | *_TargetPLCMdSta[23] | Target Node PLC Operating Flag for Node Address 23 |
| | 8 | *_TargetPLCMdSta[24] | Target Node PLC Operating Flag for Node Address 24 |
| | 9 | *_TargetPLCMdSta[25] | Target Node PLC Operating Flag for Node Address 25 |
| | 10 | *_TargetPLCMdSta[26] | Target Node PLC Operating Flag for Node Address 26 |
| | 11 | *_TargetPLCMdSta[27] | Target Node PLC Operating Flag for Node Address 27 |
| | 12 | *_TargetPLCMdSta[28] | Target Node PLC Operating Flag for Node Address 28 |
| | 13 | *_TargetPLCMdSta[29] | Target Node PLC Operating Flag for Node Address 29 |
| | 14 | *_TargetPLCMdSta[30] | Target Node PLC Operating Flag for Node Address 30 |
| | 15 | *_TargetPLCMdSta[31] | Target Node PLC Operating Flag for Node Address 31 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| CIO n+4 | 0 | *_TargetPLCMdSta[32] | Target Node PLC Operating Flag for Node Address 32 |
| | 1 | *_TargetPLCMdSta[33] | Target Node PLC Operating Flag for Node Address 33 |
| | 2 | *_TargetPLCMdSta[34] | Target Node PLC Operating Flag for Node Address 34 |
| | 3 | *_TargetPLCMdSta[35] | Target Node PLC Operating Flag for Node Address 35 |
| | 4 | *_TargetPLCMdSta[36] | Target Node PLC Operating Flag for Node Address 36 |
| | 5 | *_TargetPLCMdSta[37] | Target Node PLC Operating Flag for Node Address 37 |
| | 6 | *_TargetPLCMdSta[38] | Target Node PLC Operating Flag for Node Address 38 |
| | 7 | *_TargetPLCMdSta[39] | Target Node PLC Operating Flag for Node Address 39 |
| | 8 | *_TargetPLCMdSta[40] | Target Node PLC Operating Flag for Node Address 40 |
| | 9 | *_TargetPLCMdSta[41] | Target Node PLC Operating Flag for Node Address 41 |
| | 10 | *_TargetPLCMdSta[42] | Target Node PLC Operating Flag for Node Address 42 |
| | 11 | *_TargetPLCMdSta[43] | Target Node PLC Operating Flag for Node Address 43 |
| | 12 | *_TargetPLCMdSta[44] | Target Node PLC Operating Flag for Node Address 44 |
| | 13 | *_TargetPLCMdSta[45] | Target Node PLC Operating Flag for Node Address 45 |
| | 14 | *_TargetPLCMdSta[46] | Target Node PLC Operating Flag for Node Address 46 |
| | 15 | *_TargetPLCMdSta[47] | Target Node PLC Operating Flag for Node Address 47 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| CIO n+5 | 0 | *_TargetPLCMdSta[48] | Target Node PLC Operating Flag for Node Address 48 |
| | 1 | *_TargetPLCMdSta[49] | Target Node PLC Operating Flag for Node Address 49 |
| | 2 | *_TargetPLCMdSta[50] | Target Node PLC Operating Flag for Node Address 50 |
| | 3 | *_TargetPLCMdSta[51] | Target Node PLC Operating Flag for Node Address 51 |
| | 4 | *_TargetPLCMdSta[52] | Target Node PLC Operating Flag for Node Address 52 |
| | 5 | *_TargetPLCMdSta[53] | Target Node PLC Operating Flag for Node Address 53 |
| | 6 | *_TargetPLCMdSta[54] | Target Node PLC Operating Flag for Node Address 54 |
| | 7 | *_TargetPLCMdSta[55] | Target Node PLC Operating Flag for Node Address 55 |
| | 8 | *_TargetPLCMdSta[56] | Target Node PLC Operating Flag for Node Address 56 |
| | 9 | *_TargetPLCMdSta[57] | Target Node PLC Operating Flag for Node Address 57 |
| | 10 | *_TargetPLCMdSta[58] | Target Node PLC Operating Flag for Node Address 58 |
| | 11 | *_TargetPLCMdSta[59] | Target Node PLC Operating Flag for Node Address 59 |
| | 12 | *_TargetPLCMdSta[60] | Target Node PLC Operating Flag for Node Address 60 |
| | 13 | *_TargetPLCMdSta[61] | Target Node PLC Operating Flag for Node Address 61 |
| | 14 | *_TargetPLCMdSta[62] | Target Node PLC Operating Flag for Node Address 62 |
| | 15 | *_TargetPLCMdSta[63] | Target Node PLC Operating Flag for Node Address 63 |

### ● CIO n+6 to n+9: Target Node PLC Error Flags

The device variable that corresponds to all of the bits in CIO n+6 to n+9 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+6 to n+9 | 00 to 15 in each word | *_TargetPLCErrSta | Target Node PLC Error Flags<br>(The functions of bits 00 to 15 in these words and the functions of the device variable given on the left correspond as given below.)<br>• Bits 00 to 15 of CIO n+6 correspond to bits 00 to 15 of the device variable given on the left.<br>• Bits 00 to 15 of CIO n+7 correspond to bits 16 to 31 of the device variable given on the left.<br>• Bits 00 to 15 of CIO n+8 correspond to bits 32 to 47 of the device variable given on the left.<br>• Bits 00 to 15 of CIO n+9 correspond to bits 48 to 63 of the device variable given on the left. |

The device variables that correspond to bits 00 to 15 in CIO n+6 to n+9 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| CIO n+6 | 0 | *_TargetPLCErrSta[0] | Target Node PLC Error Flag for Node Address 0 |
| | 1 | *_TargetPLCErrSta[1] | Target Node PLC Error Flag for Node Address 1 |
| | 2 | *_TargetPLCErrSta[2] | Target Node PLC Error Flag for Node Address 2 |
| | 3 | *_TargetPLCErrSta[3] | Target Node PLC Error Flag for Node Address 3 |
| | 4 | *_TargetPLCErrSta[4] | Target Node PLC Error Flag for Node Address 4 |
| | 5 | *_TargetPLCErrSta[5] | Target Node PLC Error Flag for Node Address 5 |
| | 6 | *_TargetPLCErrSta[6] | Target Node PLC Error Flag for Node Address 6 |
| | 7 | *_TargetPLCErrSta[7] | Target Node PLC Error Flag for Node Address 7 |
| | 8 | *_TargetPLCErrSta[8] | Target Node PLC Error Flag for Node Address 8 |
| | 9 | *_TargetPLCErrSta[9] | Target Node PLC Error Flag for Node Address 9 |
| | 10 | *_TargetPLCErrSta[10] | Target Node PLC Error Flag for Node Address 10 |
| | 11 | *_TargetPLCErrSta[11] | Target Node PLC Error Flag for Node Address 11 |
| | 12 | *_TargetPLCErrSta[12] | Target Node PLC Error Flag for Node Address 12 |
| | 13 | *_TargetPLCErrSta[13] | Target Node PLC Error Flag for Node Address 13 |
| | 14 | *_TargetPLCErrSta[14] | Target Node PLC Error Flag for Node Address 14 |
| | 15 | *_TargetPLCErrSta[15] | Target Node PLC Error Flag for Node Address 15 |
| CIO n+7 | 0 | *_TargetPLCErrSta[16] | Target Node PLC Error Flag for Node Address 16 |
| | 1 | *_TargetPLCErrSta[17] | Target Node PLC Error Flag for Node Address 17 |
| | 2 | *_TargetPLCErrSta[18] | Target Node PLC Error Flag for Node Address 18 |
| | 3 | *_TargetPLCErrSta[19] | Target Node PLC Error Flag for Node Address 19 |
| | 4 | *_TargetPLCErrSta[20] | Target Node PLC Error Flag for Node Address 20 |
| | 5 | *_TargetPLCErrSta[21] | Target Node PLC Error Flag for Node Address 21 |
| | 6 | *_TargetPLCErrSta[22] | Target Node PLC Error Flag for Node Address 22 |
| | 7 | *_TargetPLCErrSta[23] | Target Node PLC Error Flag for Node Address 23 |
| | 8 | *_TargetPLCErrSta[24] | Target Node PLC Error Flag for Node Address 24 |
| | 9 | *_TargetPLCErrSta[25] | Target Node PLC Error Flag for Node Address 25 |
| | 10 | *_TargetPLCErrSta[26] | Target Node PLC Error Flag for Node Address 26 |
| | 11 | *_TargetPLCErrSta[27] | Target Node PLC Error Flag for Node Address 27 |
| | 12 | *_TargetPLCErrSta[28] | Target Node PLC Error Flag for Node Address 28 |
| | 13 | *_TargetPLCErrSta[29] | Target Node PLC Error Flag for Node Address 29 |
| | 14 | *_TargetPLCErrSta[30] | Target Node PLC Error Flag for Node Address 30 |
| | 15 | *_TargetPLCErrSta[31] | Target Node PLC Error Flag for Node Address 31 |
| CIO n+8 | 0 | *_TargetPLCErrSta[32] | Target Node PLC Error Flag for Node Address 32 |
| | 1 | *_TargetPLCErrSta[33] | Target Node PLC Error Flag for Node Address 33 |
| | 2 | *_TargetPLCErrSta[34] | Target Node PLC Error Flag for Node Address 34 |
| | 3 | *_TargetPLCErrSta[35] | Target Node PLC Error Flag for Node Address 35 |
| | 4 | *_TargetPLCErrSta[36] | Target Node PLC Error Flag for Node Address 36 |
| | 5 | *_TargetPLCErrSta[37] | Target Node PLC Error Flag for Node Address 37 |
| | 6 | *_TargetPLCErrSta[38] | Target Node PLC Error Flag for Node Address 38 |
| | 7 | *_TargetPLCErrSta[39] | Target Node PLC Error Flag for Node Address 39 |
| | 8 | *_TargetPLCErrSta[40] | Target Node PLC Error Flag for Node Address 40 |
| | 9 | *_TargetPLCErrSta[41] | Target Node PLC Error Flag for Node Address 41 |
| | 10 | *_TargetPLCErrSta[42] | Target Node PLC Error Flag for Node Address 42 |
| | 11 | *_TargetPLCErrSta[43] | Target Node PLC Error Flag for Node Address 43 |
| | 12 | *_TargetPLCErrSta[44] | Target Node PLC Error Flag for Node Address 44 |
| | 13 | *_TargetPLCErrSta[45] | Target Node PLC Error Flag for Node Address 45 |
| | 14 | *_TargetPLCErrSta[46] | Target Node PLC Error Flag for Node Address 46 |
| | 15 | *_TargetPLCErrSta[47] | Target Node PLC Error Flag for Node Address 47 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| CIO n+9 | 0 | *_TargetPLCErrSta[48] | Target Node PLC Error Flag for Node Address 48 |
| | 1 | *_TargetPLCErrSta[49] | Target Node PLC Error Flag for Node Address 49 |
| | 2 | *_TargetPLCErrSta[50] | Target Node PLC Error Flag for Node Address 50 |
| | 3 | *_TargetPLCErrSta[51] | Target Node PLC Error Flag for Node Address 51 |
| | 4 | *_TargetPLCErrSta[52] | Target Node PLC Error Flag for Node Address 52 |
| | 5 | *_TargetPLCErrSta[53] | Target Node PLC Error Flag for Node Address 53 |
| | 6 | *_TargetPLCErrSta[54] | Target Node PLC Error Flag for Node Address 54 |
| | 7 | *_TargetPLCErrSta[55] | Target Node PLC Error Flag for Node Address 55 |
| | 8 | *_TargetPLCErrSta[56] | Target Node PLC Error Flag for Node Address 56 |
| | 9 | *_TargetPLCErrSta[57] | Target Node PLC Error Flag for Node Address 57 |
| | 10 | *_TargetPLCErrSta[58] | Target Node PLC Error Flag for Node Address 58 |
| | 11 | *_TargetPLCErrSta[59] | Target Node PLC Error Flag for Node Address 59 |
| | 12 | *_TargetPLCErrSta[60] | Target Node PLC Error Flag for Node Address 60 |
| | 13 | *_TargetPLCErrSta[61] | Target Node PLC Error Flag for Node Address 61 |
| | 14 | *_TargetPLCErrSta[62] | Target Node PLC Error Flag for Node Address 62 |
| | 15 | *_TargetPLCErrSta[63] | Target Node PLC Error Flag for Node Address 63 |

### ● CIO n+10: Unit Status 1

The device variable that corresponds to all of the bits in CIO n+10 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+10 | 00 to 15 | *_UnitSta1 | Unit Status 1<br>(The functions of bits 00 to 15 in CIO n+10 correspond to the functions of bits 00 to 15 of the device variable given on the left.) |

The device variables that correspond to bits 00 to 15 in CIO n+10 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+10 | 0 | *_UnitErr | Unit Error Occurred |
| | 1 | *_NetErr | Network Error Occurred |
| | 02 and 03 | --- | Not used. |
| | 4 | *_UnitMemErr | Unit Memory Error |
| | 5 | *_LANHwErr | Communications Controller Error |
| | 6 | *_IPAdrDupErr | IP Address Duplication Error |
| | 07 and 08 | --- | Not used. |
| | 9 | *_LkOffErr | Link OFF Error |
| | 10 to 13 | --- | Not used. |
| | 14 | *_UserStaAreaCfgErr | Status Area Layout Setting Error |
| | 15 | --- | Not used. |

- **CIO n+11: Unit Status 2**

The device variable that corresponds to all of the bits in CIO n+11 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| **Word address** | **Bit numbers** | **Variable name** | **Description** |
| CIO n+11 | 00 to 15 | *_UnitSta2 | Unit Status 2<br>(The functions of bits 00 to 15 in CIO n+11 correspond to the functions of bits 00 to 15 of the device variable given on the left.) |

The device variables that correspond to bits 00 to 15 in CIO n+11 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| **Word address** | **Bit numbers** | **Variable name** | **Description** |
| CIO n+11 | 0 | *_ETNOnlineSta | Online |
| | 1 | *_TDLOprSta | Tag Data Link Operating |
| | 2 | *_IPAdrChgErr | Operating IP Address Change |
| | 03 to 10 | --- | Not used. |
| | 11 | *_UserStaAreaEnblSta | User Settings Area Enabled |
| | 12 and 13 | --- | Not used. |
| | 14 | *_MultiSwOnErr | Multiple Switches ON Error |
| | 15 | *_ErrLogStoreSta | Error Log Stored |

- **CIO n+12: Communications Status 1**

The device variable that corresponds to all of the bits in CIO n+12 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| **Word address** | **Bit number** | **Variable name** | **Description** |
| CIO n+12 | 00 to 15 | Communications Status 1 | Communications Status 1<br>(The functions of bits 00 to 15 in CIO n+12 correspond to the functions of bits 00 to 15 of the device variable given on the left.) |

The device variables that correspond to bits 00 to 15 in CIO n+12 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| **Word address** | **Bit number** | **Variable name** | **Description** |
| CIO n+12 | 0 | *_TDLOpnErr | Verification Error |
| | 1 | --- | Not used. |
| | 2 | *_TDLErr | Tag Data Link Error |
| | 3 | --- | Not used. |
| | 4 | *_CommParamErr | Invalid Communications Parameter |
| | 5 | *_TagRefreshErr | Tag Refresh Error |
| | 6 | *_TagDbErr | Tag Database Error |
| | 07 to 13 | --- | Not used. |
| | 14 | *_TDLAllRunSta | All Tag Data Links Operating |
| | 15 | *_TDLRunSta | Tag Data Links Operating |

### CIO n+13: Communications Status 2

The device variable that corresponds to all of the bits in CIO n+13 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+13 | 00 to 15 | *_CommSta2 | Communications Status 2<br>(The functions of bits 00 to 15 in CIO n+13 correspond to the functions of bits 00 to 15 of the device variable given on the left.) |

The device variables that correspond to bits 00 to 15 in CIO n+13 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+13 | 0 | *_FTPSta | FTP Status |
| | 01 to 13 | --- | Not used. |
| | 14 | *_LkSta | Link Status |
| | 15 | --- | Not used. |

### CIO n+14: Communications Status 3

The device variable that corresponds to all of the bits in CIO n+14 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+14 | 00 to 15 | *_CommSta3 | Communications Status 3<br>(The functions of bits 00 to 15 in CIO n+14 correspond to the functions of bits 0 to 15 of the device variable given on the left.) |

The device variables that correspond to bits 00 to 15 in CIO n+14 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+14 | 00 and 01 | --- | Not used. |
| | 2 | *_ETNBaseSetErr | Basic Ethernet Settings Error |
| | 3 | *_IPAdrTblErr | IP Address Table Error |
| | 4 | *_IPRouterTblErr | IP Router Table Error |
| | 5 | *_DNSSvrErr | DNS Server Error |
| | 6 | *_RTblErr | Routing Table Error |
| | 07 and 08 | --- | Not used. |
| | 9 | *_ETNAdvSetErr | Ethernet Advanced Settings Error |
| | 10 | *_BootpSvrErr | BOOTP Server Error |
| | 11 | *_SNTPSvrErr | SNTP Server Error |
| | 12 and 13 | --- | Not used. |
| | 14 | *_AdrMismatchErr | Address Mismatch |
| | 15 | *_MemErr | Non-volatile Memory Error |

## CIO n+16 to n+19: Registered Target Node Table

The device variable that corresponds to all of the bits in CIO n+16 to n+19 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+16 to n+19 | 00 to 15 in each word | *_RegTargetSta | Registered Target Node Table<br>(The functions of bits 00 to 15 in these words and the functions of the device variable given on the left correspond as given below.)<br>• Bits 00 to 15 of CIO n+16 correspond to bits 00 to 15 of the device variable given on the left.<br>• Bits 00 to 15 of CIO n+17 correspond to bits 16 to 31 of the device variable given on the left.<br>• Bits 00 to 15 of CIO n+18 correspond to bits 32 to 47 of the device variable given on the left.<br>• Bits 00 to 15 of CIO n+19 correspond to bits 48 to 63 of the device variable given on the left. |

The device variables that correspond to bits 00 to 15 in CIO n+16 to n+19 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+16 | 0 | *_RegTargetSta[0] | Registered Target Node Table Bit for Node Address 0 |
| | 1 | *_RegTargetSta[1] | Registered Target Node Table Bit for Node Address 1 |
| | 2 | *_RegTargetSta[2] | Registered Target Node Table Bit for Node Address 2 |
| | 3 | *_RegTargetSta[3] | Registered Target Node Table Bit for Node Address 3 |
| | 4 | *_RegTargetSta[4] | Registered Target Node Table Bit for Node Address 4 |
| | 5 | *_RegTargetSta[5] | Registered Target Node Table Bit for Node Address 5 |
| | 6 | *_RegTargetSta[6] | Registered Target Node Table Bit for Node Address 6 |
| | 7 | *_RegTargetSta[7] | Registered Target Node Table Bit for Node Address 7 |
| | 8 | *_RegTargetSta[8] | Registered Target Node Table Bit for Node Address 8 |
| | 9 | *_RegTargetSta[9] | Registered Target Node Table Bit for Node Address 9 |
| | 10 | *_RegTargetSta[10] | Registered Target Node Table Bit for Node Address 10 |
| | 11 | *_RegTargetSta[11] | Registered Target Node Table Bit for Node Address 11 |
| | 12 | *_RegTargetSta[12] | Registered Target Node Table Bit for Node Address 12 |
| | 13 | *_RegTargetSta[13] | Registered Target Node Table Bit for Node Address 13 |
| | 14 | *_RegTargetSta[14] | Registered Target Node Table Bit for Node Address 14 |
| | 15 | *_RegTargetSta[15] | Registered Target Node Table Bit for Node Address 15 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+17 | 0 | *_RegTargetSta[16] | Registered Target Node Table Bit for Node Address 16 |
| | 1 | *_RegTargetSta[17] | Registered Target Node Table Bit for Node Address 17 |
| | 2 | *_RegTargetSta[18] | Registered Target Node Table Bit for Node Address 18 |
| | 3 | *_RegTargetSta[19] | Registered Target Node Table Bit for Node Address 19 |
| | 4 | *_RegTargetSta[20] | Registered Target Node Table Bit for Node Address 20 |
| | 5 | *_RegTargetSta[21] | Registered Target Node Table Bit for Node Address 21 |
| | 6 | *_RegTargetSta[22] | Registered Target Node Table Bit for Node Address 22 |
| | 7 | *_RegTargetSta[23] | Registered Target Node Table Bit for Node Address 23 |
| | 8 | *_RegTargetSta[24] | Registered Target Node Table Bit for Node Address 24 |
| | 9 | *_RegTargetSta[25] | Registered Target Node Table Bit for Node Address 25 |
| | 10 | *_RegTargetSta[26] | Registered Target Node Table Bit for Node Address 26 |
| | 11 | *_RegTargetSta[27] | Registered Target Node Table Bit for Node Address 27 |
| | 12 | *_RegTargetSta[28] | Registered Target Node Table Bit for Node Address 28 |
| | 13 | *_RegTargetSta[29] | Registered Target Node Table Bit for Node Address 29 |
| | 14 | *_RegTargetSta[30] | Registered Target Node Table Bit for Node Address 30 |
| | 15 | *_RegTargetSta[31] | Registered Target Node Table Bit for Node Address 31 |
| CIO n+18 | 0 | *_RegTargetSta[32] | Registered Target Node Table Bit for Node Address 32 |
| | 1 | *_RegTargetSta[33] | Registered Target Node Table Bit for Node Address 33 |
| | 2 | *_RegTargetSta[34] | Registered Target Node Table Bit for Node Address 34 |
| | 3 | *_RegTargetSta[35] | Registered Target Node Table Bit for Node Address 35 |
| | 4 | *_RegTargetSta[36] | Registered Target Node Table Bit for Node Address 36 |
| | 5 | *_RegTargetSta[37] | Registered Target Node Table Bit for Node Address 37 |
| | 6 | *_RegTargetSta[38] | Registered Target Node Table Bit for Node Address 38 |
| | 7 | *_RegTargetSta[39] | Registered Target Node Table Bit for Node Address 39 |
| | 8 | *_RegTargetSta[40] | Registered Target Node Table Bit for Node Address 40 |
| | 9 | *_RegTargetSta[41] | Registered Target Node Table Bit for Node Address 41 |
| | 10 | *_RegTargetSta[42] | Registered Target Node Table Bit for Node Address 42 |
| | 11 | *_RegTargetSta[43] | Registered Target Node Table Bit for Node Address 43 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+18 | 12 | *_RegTargetSta[44] | Registered Target Node Table Bit for Node Address 44 |
| | 13 | *_RegTargetSta[45] | Registered Target Node Table Bit for Node Address 45 |
| | 14 | *_RegTargetSta[46] | Registered Target Node Table Bit for Node Address 46 |
| | 15 | *_RegTargetSta[47] | Registered Target Node Table Bit for Node Address 47 |
| CIO n+19 | 0 | *_RegTargetSta[48] | Registered Target Node Table Bit for Node Address 48 |
| | 1 | *_RegTargetSta[49] | Registered Target Node Table Bit for Node Address 49 |
| | 2 | *_RegTargetSta[50] | Registered Target Node Table Bit for Node Address 50 |
| | 3 | *_RegTargetSta[51] | Registered Target Node Table Bit for Node Address 51 |
| | 4 | *_RegTargetSta[52] | Registered Target Node Table Bit for Node Address 52 |
| | 5 | *_RegTargetSta[53] | Registered Target Node Table Bit for Node Address 53 |
| | 6 | *_RegTargetSta[54] | Registered Target Node Table Bit for Node Address 54 |
| | 7 | *_RegTargetSta[55] | Registered Target Node Table Bit for Node Address 55 |
| | 8 | *_RegTargetSta[56] | Registered Target Node Table Bit for Node Address 56 |
| | 9 | *_RegTargetSta[57] | Registered Target Node Table Bit for Node Address 57 |
| | 10 | *_RegTargetSta[58] | Registered Target Node Table Bit for Node Address 58 |
| | 11 | *_RegTargetSta[59] | Registered Target Node Table Bit for Node Address 59 |
| | 12 | *_RegTargetSta[60] | Registered Target Node Table Bit for Node Address 60 |
| | 13 | *_RegTargetSta[61] | Registered Target Node Table Bit for Node Address 61 |
| | 14 | *_RegTargetSta[62] | Registered Target Node Table Bit for Node Address 62 |
| | 15 | *_RegTargetSta[63] | Registered Target Node Table Bit for Node Address 63 |

### CIO n+20 to n+23: Normal Target Node Table

The device variable that corresponds to all of the bits in CIO n+20 to n+23 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+20 to n+23 | 00 to 15 in each word | *_EstbRegTargetSta | Normal Target Node Table<br>(The functions of bits 00 to 15 in these words and the functions of the device variable given on the left correspond as given below.)<br>• Bits 00 to 15 of CIO n+20 correspond to bits 00 to 15 of the device variable given on the left.<br>• Bits 00 to 15 of CIO n+21 correspond to bits 16 to 31 of the device variable given on the left.<br>• Bits 00 to 15 of CIO n+22 correspond to bits 32 to 47 of the device variable given on the left.<br>• Bits 00 to 15 of CIO n+23 correspond to bits 48 to 63 of the device variable given on the left. |

The device variables that correspond to bits 00 to 15 in CIO n+20 to n+23 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+20 | 0 | *_EstbRegTargetSta[0] | Normal Target Node Table Bit for Node Address 0 |
| | 1 | *_EstbRegTargetSta[1] | Normal Target Node Table Bit for Node Address 1 |
| | 2 | *_EstbRegTargetSta[2] | Normal Target Node Table Bit for Node Address 2 |
| | 3 | *_EstbRegTargetSta[3] | Normal Target Node Table Bit for Node Address 3 |
| | 4 | *_EstbRegTargetSta[4] | Normal Target Node Table Bit for Node Address 4 |
| | 5 | *_EstbRegTargetSta[5] | Normal Target Node Table Bit for Node Address 5 |
| | 6 | *_EstbRegTargetSta[6] | Normal Target Node Table Bit for Node Address 6 |
| | 7 | *_EstbRegTargetSta[7] | Normal Target Node Table Bit for Node Address 7 |
| | 8 | *_EstbRegTargetSta[8] | Normal Target Node Table Bit for Node Address 8 |
| | 9 | *_EstbRegTargetSta[9] | Normal Target Node Table Bit for Node Address 9 |
| | 10 | *_EstbRegTargetSta[10] | Normal Target Node Table Bit for Node Address 10 |
| | 11 | *_EstbRegTargetSta[11] | Normal Target Node Table Bit for Node Address 11 |
| | 12 | *_EstbRegTargetSta[12] | Normal Target Node Table Bit for Node Address 12 |
| | 13 | *_EstbRegTargetSta[13] | Normal Target Node Table Bit for Node Address 13 |
| | 14 | *_EstbRegTargetSta[14] | Normal Target Node Table Bit for Node Address 14 |
| | 15 | *_EstbRegTargetSta[15] | Normal Target Node Table Bit for Node Address 15 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+21 | 0 | *_EstbRegTargetSta[16] | Normal Target Node Table Bit for Node Address 16 |
| | 1 | *_EstbRegTargetSta[17] | Normal Target Node Table Bit for Node Address 17 |
| | 2 | *_EstbRegTargetSta[18] | Normal Target Node Table Bit for Node Address 18 |
| | 3 | *_EstbRegTargetSta[19] | Normal Target Node Table Bit for Node Address 19 |
| | 4 | *_EstbRegTargetSta[20] | Normal Target Node Table Bit for Node Address 20 |
| | 5 | *_EstbRegTargetSta[21] | Normal Target Node Table Bit for Node Address 21 |
| | 6 | *_EstbRegTargetSta[22] | Normal Target Node Table Bit for Node Address 22 |
| | 7 | *_EstbRegTargetSta[23] | Normal Target Node Table Bit for Node Address 23 |
| | 8 | *_EstbRegTargetSta[24] | Normal Target Node Table Bit for Node Address 24 |
| | 9 | *_EstbRegTargetSta[25] | Normal Target Node Table Bit for Node Address 25 |
| | 10 | *_EstbRegTargetSta[26] | Normal Target Node Table Bit for Node Address 26 |
| | 11 | *_EstbRegTargetSta[27] | Normal Target Node Table Bit for Node Address 27 |
| | 12 | *_EstbRegTargetSta[28] | Normal Target Node Table Bit for Node Address 28 |
| | 13 | *_EstbRegTargetSta[29] | Normal Target Node Table Bit for Node Address 29 |
| | 14 | *_EstbRegTargetSta[30] | Normal Target Node Table Bit for Node Address 30 |
| | 15 | *_EstbRegTargetSta[31] | Normal Target Node Table Bit for Node Address 31 |
| CIO n+22 | 0 | *_EstbRegTargetSta[32] | Normal Target Node Table Bit for Node Address 32 |
| | 1 | *_EstbRegTargetSta[33] | Normal Target Node Table Bit for Node Address 33 |
| | 2 | *_EstbRegTargetSta[34] | Normal Target Node Table Bit for Node Address 34 |
| | 3 | *_EstbRegTargetSta[35] | Normal Target Node Table Bit for Node Address 35 |
| | 4 | *_EstbRegTargetSta[36] | Normal Target Node Table Bit for Node Address 36 |
| | 5 | *_EstbRegTargetSta[37] | Normal Target Node Table Bit for Node Address 37 |
| | 6 | *_EstbRegTargetSta[38] | Normal Target Node Table Bit for Node Address 38 |
| | 7 | *_EstbRegTargetSta[39] | Normal Target Node Table Bit for Node Address 39 |
| | 8 | *_EstbRegTargetSta[40] | Normal Target Node Table Bit for Node Address 40 |
| | 9 | *_EstbRegTargetSta[41] | Normal Target Node Table Bit for Node Address 41 |
| | 10 | *_EstbRegTargetSta[42] | Normal Target Node Table Bit for Node Address 42 |
| | 11 | *_EstbRegTargetSta[43] | Normal Target Node Table Bit for Node Address 43 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+22 | 12 | *_EstbRegTargetSta[44] | Normal Target Node Table Bit for Node Address 44 |
| | 13 | *_EstbRegTargetSta[45] | Normal Target Node Table Bit for Node Address 45 |
| | 14 | *_EstbRegTargetSta[46] | Normal Target Node Table Bit for Node Address 46 |
| | 15 | *_EstbRegTargetSta[47] | Normal Target Node Table Bit for Node Address 47 |
| CIO n+23 | 0 | *_EstbRegTargetSta[48] | Normal Target Node Table Bit for Node Address 48 |
| | 1 | *_EstbRegTargetSta[49] | Normal Target Node Table Bit for Node Address 49 |
| | 2 | *_EstbRegTargetSta[50] | Normal Target Node Table Bit for Node Address 50 |
| | 3 | *_EstbRegTargetSta[51] | Normal Target Node Table Bit for Node Address 51 |
| | 4 | *_EstbRegTargetSta[52] | Normal Target Node Table Bit for Node Address 52 |
| | 5 | *_EstbRegTargetSta[53] | Normal Target Node Table Bit for Node Address 53 |
| | 6 | *_EstbRegTargetSta[54] | Normal Target Node Table Bit for Node Address 54 |
| | 7 | *_EstbRegTargetSta[55] | Normal Target Node Table Bit for Node Address 55 |
| | 8 | *_EstbRegTargetSta[56] | Normal Target Node Table Bit for Node Address 56 |
| | 9 | *_EstbRegTargetSta[57] | Normal Target Node Table Bit for Node Address 57 |
| | 10 | *_EstbRegTargetSta[58] | Normal Target Node Table Bit for Node Address 58 |
| | 11 | *_EstbRegTargetSta[59] | Normal Target Node Table Bit for Node Address 59 |
| | 12 | *_EstbRegTargetSta[60] | Normal Target Node Table Bit for Node Address 60 |
| | 13 | *_EstbRegTargetSta[61] | Normal Target Node Table Bit for Node Address 61 |
| | 14 | *_EstbRegTargetSta[62] | Normal Target Node Table Bit for Node Address 62 |
| | 15 | *_EstbRegTargetSta[63] | Normal Target Node Table Bit for Node Address 63 |

### ● CIO n+24: FINS/TCP Connection Status

The device variable that corresponds to all of the bits in CIO n+24 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+24 | 00 to 15 | *_FINSTCPConnSta | FINS/TCP Connection Status<br>(The functions of bits 00 to 15 in CIO n+24 correspond to the functions of bits 0 to 15 of the device variable given on the left.) |

The device variables that correspond to bits 00 to 15 in CIO n+24 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| CIO n+24 | 0 | *_FINSTCPConnSta1 | FINS/TCP Connection Status for Connection 1 |
| | 1 | *_FINSTCPConnSta2 | FINS/TCP Connection Status for Connection 2 |
| | 2 | *_FINSTCPConnSta3 | FINS/TCP Connection Status for Connection 3 |
| | 3 | *_FINSTCPConnSta4 | FINS/TCP Connection Status for Connection 4 |
| | 4 | *_FINSTCPConnSta5 | FINS/TCP Connection Status for Connection 5 |
| | 5 | *_FINSTCPConnSta6 | FINS/TCP Connection Status for Connection 6 |
| | 6 | *_FINSTCPConnSta7 | FINS/TCP Connection Status for Connection 7 |
| | 7 | *_FINSTCPConnSta8 | FINS/TCP Connection Status for Connection 8 |
| | 8 | *_FINSTCPConnSta9 | FINS/TCP Connection Status for Connection 9 |
| | 9 | *_FINSTCPConnSta10 | FINS/TCP Connection Status for Connection 10 |
| | 10 | *_FINSTCPConnSta11 | FINS/TCP Connection Status for Connection 11 |
| | 11 | *_FINSTCPConnSta12 | FINS/TCP Connection Status for Connection 12 |
| | 12 | *_FINSTCPConnSta13 | FINS/TCP Connection Status for Connection 13 |
| | 13 | *_FINSTCPConnSta14 | FINS/TCP Connection Status for Connection 14 |
| | 14 | *_FINSTCPConnSta15 | FINS/TCP Connection Status for Connection 15 |
| | 15 | *_FINSTCPConnSta16 | FINS/TCP Connection Status for Connection 16 |

- **CIO n+1 to n+4: Socket Service Request Switches (CJ1W-EIP21S Only)**

When the status area layout in the Status Area Settings Display is set to *User setting*.

The device variable that corresponds to all of the bits in CIO n+1 to n+4 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+1 | 0/8 | *_Skt1UdpOpenCmd/<br>*_Skt2UdpOpenCmd | UDP Open Request Switch (Socket No.1)/<br>UDP Open Request Switch (Socket No.2) |
| | 1/9 | *_Skt1TcpPassiveCmd/<br>*_Skt2TcpPassiveCmd | TCP Passive Open Request Switch (Socket No.1)/<br>TCP Passive Open Request Switch (Socket No.2) |
| | 2/10 | *_Skt1TcpActiveCmd/<br>*_Skt2TcpActiveCmd | TCP Active Open Request Switch (Socket No.1)/<br>TCP Active Open Request Switch (Socket No.2) |
| | 3/11 | *_Skt1SendCmd/<br>*_Skt2SendCmd | Send Request Switch (Socket No.1)/<br>Send Request Switch (Socket No.2) |
| | 4/12 | *_Skt1RcvCmd/<br>*_Skt2RcvCmd | Receive Request Switch (Socket No.1)/<br>Receive Request Switch (Socket No.2) |
| | 5/13 | *_Skt1CloseCmd/<br>*_Skt2CloseCmd | Close Request Switch (Socket No.1)/<br>Close Request Switch (Socket No.2) |
| | 6/14 | --- | --- |
| | 7/15 | --- | --- |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+2 | 0/8 | *_Skt3UdpOpenCmd/<br>*_Skt4UdpOpenCmd | UDP Open Request Switch (Socket No.3)/<br>UDP Open Request Switch (SocketNo.4) |
| | 1/9 | *_Skt3TcpPassiveCmd/<br>*_Skt4TcpPassiveCmd | TCP Passive Open Request Switch (Socket No.3)/<br>TCP Passive Open Request Switch (Socket No.4) |
| | 2/10 | *_Skt3TcpActiveCmd/<br>*_Skt4TcpActiveCmd | TCP Active Open Request Switch (Socket No.3)/<br>TCP Active Open Request Switch (Socket No.4) |
| | 3/11 | *_Skt3SendCmd/<br>*_Skt4SendCmd | Send Request Switch (Socket No.3)/<br>Send Request Switch (Socket No.4) |
| | 4/12 | *_Skt3RcvCmd/<br>*_Skt4RcvCmd | Receive Request Switch (Socket No.3)/<br>Receive Request Switch (Socket No.4) |
| | 5/13 | *_Skt3CloseCmd/<br>*_Skt4CloseCmd | Close Request Switch (Socket No.3)/<br>Close Request Switch (Socket No.4) |
| | 6/14 | --- | --- |
| | 7/15 | --- | --- |
| CIO n+3 | 0/8 | *_Skt5UdpOpenCmd/<br>*_Skt6UdpOpenCmd | UDP Open Request Switch (Socket No.5)/<br>UDP Open Request Switch (Socket No.6) |
| | 1/9 | *_Skt5TcpPassiveCmd/<br>*_Skt6TcpPassiveCmd | TCP Passive Open Request Switch (Socket No.5)/<br>TCP Passive Open Request Switch (Socket No.6) |
| | 2/10 | *_Skt5TcpActiveCmd/<br>*_Skt6TcpActiveCmd | TCP Active Open Request Switch (Socket No.5)/<br>TCP Active Open Request Switch (Socket No.6) |
| | 3/11 | *_Skt5SendCmd/<br>*_Skt6SendCmd | Send Request Switch (Socket No.5)/<br>Send Request Switch (Socket No.6) |
| | 4/12 | *_Skt5RcvCmd/<br>*_Skt6RcvCmd | Receive Request Switch (Socket No.5)/<br>Receive Request Switch (Socket No.6) |
| | 5/13 | *_Skt5CloseCmd/<br>*_Skt6CloseCmd | Close Request Switch (Socket No.5)/<br>Close Request Switch (Socket No.6) |
| | 6/14 | --- | --- |
| | 7/15 | --- | --- |
| CIO n+4 | 0/8 | *_Skt7UdpOpenCmd/<br>*_Skt8UdpOpenCmd | UDP Open Request Switch (Socket No.7)/<br>UDP Open Request Switch (Socket No.8) |
| | 1/9 | *_Skt7TcpPassiveCmd/<br>*_Skt8TcpPassiveCmd | TCP Passive Open Request Switch (Socket No.7)/<br>TCP Passive Open Request Switch (Socket No.8) |
| | 2/10 | *_Skt7TcpActiveCmd/<br>*_Skt8TcpActiveCmd | TCP Active Open Request Switch (Socket No.7)/<br>TCP Active Open Request Switch (Socket No.8) |
| | 3/11 | *_Skt7SendCmd/<br>*_Skt8SendCmd | Send Request Switch (Socket No.7)/<br>Send Request Switch (Socket No.8) |
| | 4/12 | *_Skt7RcvCmd/<br>*_Skt8RcvCmd | Receive Request Switch (Socket No.7)/<br>Receive Request Switch (Socket No.8) |
| | 5/13 | *_Skt7CloseCmd/<br>*_Skt8CloseCmd | Close Request Switch (Socket No.7)/<br>Close Request Switch (Socket No.8) |
| | 6/14 | --- | --- |
| | 7/15 | --- | --- |

### CIO n+16 to n+23: UDP/TCP Socket No. 1 to No. 8 Status (CJ1W-EIP21S Only)

When the status area layout in the Status Area Settings Display is set to *User setting*.

The device variable that corresponds to all of the bits in CIO n+16 to n+23 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| **Word address** | **Bit numbers** | **Variable name** | **Description** |
| CIO n+16 | 0/8 | *_Skt1UdpOpening/<br>*_Skt2UdpOpening | Opening Flag (Socket No.1)/<br>Opening Flag (Socket No.2) |
| | 1/9 | *_Skt1UdpRcving/<br>*_Skt2UdpRcving | Receiving Flag (Socket No.1)/<br>Receiving Flag (Socket No.2) |
| | 2/10 | *_Skt1UdpSending/<br>*_Skt2UdpSending | Sending Flag (Socket No.1)/<br>Sending Flag (Socket No.2) |
| | 3/11 | *_Skt1UdpClosing/<br>*_Skt2UdpClosing | Closing Flag (Socket No.1)/<br>Closing Flag (Socket No.2) |
| | 4/12 | --- | --- |
| | 5/13 | *_Skt1UdpAreaErr/<br>*_Skt2UdpAreaErr | Results Storage Error Flag (Socket No.1)/<br>Results Storage Error Flag (Socket No.2) |
| | 6/14 | *_Skt1UdpOpened/<br>*_Skt2UdpOpened | UDP Open Flag (Socket No.1)/<br>UDP Open Flag (Socket No.2) |
| | 7/15 | --- | --- |
| CIO n+17 | 0/8 | *_Skt3UdpOpening/<br>*_Skt4UdpOpening | Opening Flag (Socket No.3)/<br>Opening Flag (Socket No.4) |
| | 1/9 | *_Skt3UdpRcving/<br>*_Skt4UdpRcving | Receiving Flag (Socket No.3)/<br>Receiving Flag (Socket No.4) |
| | 2/10 | *_Skt3UdpSending/<br>*_Skt4UdpSending | Sending Flag (Socket No.3)/<br>Sending Flag (Socket No.4) |
| | 3/11 | *_Skt3UdpClosing/<br>*_Skt4UdpClosing | Closing Flag (Socket No.3)/<br>Closing Flag (Socket No.4) |
| | 4/12 | --- | --- |
| | 5/13 | *_Skt3UdpAreaErr/<br>*_Skt4UdpAreaErr | Results Storage Error Flag (Socket No.3)/<br>Results Storage Error Flag (Socket No.4) |
| | 6/14 | *_Skt3UdpOpened/<br>*_Skt4UdpOpened | UDP Open Flag (Socket No.3)/<br>UDP Open Flag (Socket No.4) |
| | 7/15 | --- | --- |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+18 | 0/8 | *_Skt5UdpOpening/<br>*_Skt6UdpOpening | Opening Flag (Socket No.5)/<br>Opening Flag (Socket No.6) |
| | 1/9 | *_Skt5UdpRcving/<br>*_Skt6UdpRcving | Receiving Flag (Socket No.5)/<br>Receiving Flag (Socket No.6) |
| | 2/10 | *_Skt5UdpSending/<br>*_Skt6UdpSending | Sending Flag (Socket No.5)/<br>Sending Flag (Socket No.6) |
| | 3/11 | *_Skt5UdpClosing/<br>*_Skt6UdpClosing | Closing Flag (Socket No.5)/<br>Closing Flag (Socket No.6) |
| | 4/12 | --- | --- |
| | 5/13 | *_Skt5UdpAreaErr/<br>*_Skt6UdpAreaErr | Results Storage Error Flag (Socket No.5)/<br>Results Storage Error Flag (Socket No.6) |
| | 6/14 | *_Skt5UdpOpened/<br>*_Skt6UdpOpened | UDP Open Flag (Socket No.5)/<br>UDP Open Flag (Socket No.6) |
| | 7/15 | --- | --- |
| CIO n+19 | 0/8 | *_Skt7UdpOpening/<br>*_Skt8UdpOpening | Opening Flag (Socket No.7)/<br>Opening Flag (Socket No.8) |
| | 1/9 | *_Skt7UdpRcving/<br>*_Skt8UdpRcving | Receiving Flag (Socket No.7)/<br>Receiving Flag (Socket No.8) |
| | 2/10 | *_Skt7UdpSending/<br>*_Skt8UdpSending | Sending Flag (Socket No.7)/<br>Sending Flag (Socket No.8) |
| | 3/11 | *_Skt7UdpClosing/<br>*_Skt8UdpClosing | Closing Flag (Socket No.7)/<br>Closing Flag (Socket No.8) |
| | 4/12 | --- | --- |
| | 5/13 | *_Skt7UdpAreaErr/<br>*_Skt8UdpAreaErr | Results Storage Error Flag (Socket No.7)/<br>Results Storage Error Flag (Socket No.8) |
| | 6/14 | *_Skt7UdpOpened/<br>*_Skt8UdpOpened | UDP Open Flag (Socket No.7)/<br>UDP Open Flag (Socket No.8) |
| | 7/15 | --- | --- |
| CIO n+20 | 0/8 | *_Skt1TcpOpening/<br>*_Skt2TcpOpening | Opening Flag (Socket No.1)/<br>Opening Flag (Socket No.2) |
| | 1/9 | *_Skt1TcpRcving/<br>*_Skt2TcpRcving | Receiving Flag (Socket No.1)/<br>Receiving Flag (Socket No.2) |
| | 2/10 | *_Skt1TcpSending/<br>*_Skt2TcpSending | Sending Flag (Socket No.1)/<br>Sending Flag (Socket No.2) |
| | 3/11 | *_Skt1TcpClosing/<br>*_Skt2TcpClosing | Closing Flag (Socket No.1)/<br>Closing Flag (Socket No.2) |
| | 4/12 | *_Skt1TcpDatRcved/<br>*_Skt2TcpDatRcved | Data Received Flag (Socket No.1)/<br>Data Received Flag (Socket No.2) |
| | 5/13 | *_Skt1TcpAreaErr/<br>*_Skt2TcpAreaErr | Results Storage Error Flag (Socket No.1)/<br>Results Storage Error Flag (Socket No.2) |
| | 6/14 | *_Skt1TcpOpened/<br>*_Skt2TcpOpened | TCP Open Flag (Socket No.1)/<br>TCP Open Flag (Socket No.2) |
| | 7/15 | --- | --- |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO n+21 | 0/8 | *_Skt3TcpOpening/<br>*_Skt4TcpOpening | Opening Flag (Socket No.3)/<br>Opening Flag (Socket No.4) |
| | 1/9 | *_Skt3TcpRcving/<br>*_Skt4TcpRcving | Receiving Flag (Socket No.3)/<br>Receiving Flag (Socket No.4) |
| | 2/10 | *_Skt3TcpSending/<br>*_Skt4TcpSending | Sending Flag (Socket No.3)/<br>Sending Flag (Socket No.4) |
| | 3/11 | *_Skt3TcpClosing/<br>*_Skt4TcpClosing | Closing Flag (Socket No.3)/<br>Closing Flag (Socket No.4) |
| | 4/12 | *_Skt3TcpDatRcved/<br>*_Skt4TcpDatRcved | Data Received Flag (Socket No.3)/<br>Data Received Flag (Socket No.4) |
| | 5/13 | *_Skt3TcpAreaErr/<br>*_Skt4TcpAreaErr | Results Storage Error Flag (Socket No.3)/<br>Results Storage Error Flag (Socket No.4) |
| | 6/14 | *_Skt3TcpOpened/<br>*_Skt4TcpOpened | TCP Open Flag (Socket No.3)/<br>TCP Open Flag (Socket No.4) |
| | 7/15 | --- | --- |
| CIO n+22 | 0/8 | *_Skt5TcpOpening/<br>*_Skt6TcpOpening | Opening Flag (Socket No.5)/<br>Opening Flag (Socket No.6) |
| | 1/9 | *_Skt5TcpRcving/<br>*_Skt6TcpRcving | Receiving Flag (Socket No.5)/<br>Receiving Flag (Socket No.6) |
| | 2/10 | *_Skt5TcpSending/<br>*_Skt6TcpSending | Sending Flag (Socket No.5)/<br>Sending Flag (Socket No.6) |
| | 3/11 | *_Skt5TcpClosing/<br>*_Skt6TcpClosing | Closing Flag (Socket No.5)/<br>Closing Flag (Socket No.6) |
| | 4/12 | *_Skt5TcpDatRcved/<br>*_Skt6TcpDatRcved | Data Received Flag (Socket No.5)/<br>Data Received Flag (Socket No.6) |
| | 5/13 | *_Skt5TcpAreaErr/<br>*_Skt6TcpAreaErr | Results Storage Error Flag (Socket No.5)/<br>Results Storage Error Flag (Socket No.6) |
| | 6/14 | *_Skt5TcpOpened/<br>*_Skt6TcpOpened | TCP Open Flag (Socket No.5)/<br>TCP Open Flag (Socket No.6) |
| | 7/15 | --- | --- |
| CIO n+23 | 0/8 | *_Skt7TcpOpening/<br>*_Skt8TcpOpening | Opening Flag (Socket No.7)/<br>Opening Flag (Socket No.8) |
| | 1/9 | *_Skt7TcpRcving/<br>*_Skt8TcpRcving | Receiving Flag (Socket No.7)/<br>Receiving Flag (Socket No.8) |
| | 2/10 | *_Skt7TcpSending/<br>*_Skt8TcpSending | Sending Flag (Socket No.7)/<br>Sending Flag (Socket No.8) |
| | 3/11 | *_Skt7TcpClosing/<br>*_Skt8TcpClosing | Closing Flag (Socket No.7)/<br>Closing Flag (Socket No.8) |
| | 4/12 | *_Skt7TcpDatRcved/<br>*_Skt8TcpDatRcved | Data Received Flag (Socket No.7)/<br>Data Received Flag (Socket No.8) |
| | 5/13 | *_Skt7TcpAreaErr/<br>*_Skt8TcpAreaErr | Results Storage Error Flag (Socket No.7)/<br>Results Storage Error Flag (Socket No.8) |
| | 6/14 | *_Skt7TcpOpened/<br>*_Skt8TcpOpened | TCP Open Flag (Socket No.7)/<br>TCP Open Flag (Socket No.8) |
| | 7/15 | --- | --- |

## DM Area Words Allocated to CPU Bus Units

### Words m+98 and m+99: IP Address Display/Setting Area

The device variable that corresponds to all of the bits in words m+98 and m+99 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| Words m+98 and m+99 | 00 to 15 | *_IPAdrCfg | IP Address Display/Setting Area |

The device variables that correspond to bits 00 to 15 in words m+98 and m+99 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| Word m+98 | 00 to 03 | *_IPAdr2Cfg | IP Address 2 (This is field 2 shown below.) |
| | 04 to 07 | | |
| | 08 to 11 | *_IPAdr1Cfg | IP Address 1 (This is field 1 shown below.) |
| | 12 to 15 | | |
| Word m+99 | 00 to 03 | *_IPAdr4Cfg | IP Address 4 (This is field 4 shown below.) |
| | 04 to 07 | | |
| | 08 to 11 | *_IPAdr3Cfg | IP Address 3 (This is field 3 shown below.) |
| | 12 to 15 | | |

IP address fields: 1.2.3.4 (hex)

### CIO m+1 to m+8: Number of Bytes Received at TCP Socket No. 1 to 8 (CJ1W-EIP21S Only)

The device variable that corresponds to all of the bits in CIO m+1 to m+8 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO m+1 | 0 to 15 | *_Skt1TcpRcvDatByte | TCP Socket No.1 Number of Bytes Received (0000 to 07C0 hex) |
| CIO m+2 | 0 to 15 | *_Skt2TcpRcvDatByte | TCP Socket No.2 Number of Bytes Received (0000 to 07C0 hex) |
| CIO m+3 | 0 to 15 | *_Skt3TcpRcvDatByte | TCP Socket No.3 Number of Bytes Received (0000 to 07C0 hex) |
| CIO m+4 | 0 to 15 | *_Skt4TcpRcvDatByte | TCP Socket No.4 Number of Bytes Received (0000 to 07C0 hex) |
| CIO m+5 | 0 to 15 | *_Skt5TcpRcvDatByte | TCP Socket No.5 Number of Bytes Received (0000 to 07C0 hex) |
| CIO m+6 | 0 to 15 | *_Skt6TcpRcvDatByte | TCP Socket No.6 Number of Bytes Received (0000 to 07C0 hex) |
| CIO m+7 | 0 to 15 | *_Skt7TcpRcvDatByte | TCP Socket No.7 Number of Bytes Received (0000 to 07C0 hex) |
| CIO m+8 | 0 to 15 | *_Skt8TcpRcvDatByte | TCP Socket No.8 Number of Bytes Received (0000 to 07C0 hex) |

- **CIO m+9 to m+16: TCP Socket No. 1 to 8 Connection Status (CJ1W-EIP21S Only)**

The device variable that corresponds to all of the bits in CIO m+9 to m+16 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO m+9 | 0 to 3 | *_Skt1TcpConSta | TCP Socket No.1 Connection Status |
| CIO m+10 | 0 to 3 | *_Skt2TcpConSta | TCP Socket No.2 Connection Status |
| CIO m+11 | 0 to 3 | *_Skt3TcpConSta | TCP Socket No.3 Connection Status |
| CIO m+12 | 0 to 3 | *_Skt4TcpConSta | TCP Socket No.4 Connection Status |
| CIO m+13 | 0 to 3 | *_Skt5TcpConSta | TCP Socket No.5 Connection Status |
| CIO m+14 | 0 to 3 | *_Skt6TcpConSta | TCP Socket No.6 Connection Status |
| CIO m+15 | 0 to 3 | *_Skt7TcpConSta | TCP Socket No.7 Connection Status |
| CIO m+16 | 0 to 3 | *_Skt8TcpConSta | TCP Socket No.8 Connection Status |

- **CIO m+17 to m+96: Socket Service Parameter Area 1 to 8 (CJ1W-EIP21S Only)**

The device variable that corresponds to all of the bits in CIO m+17 to m+96 is given in the following table.

Socket No. 1

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO m+17 | 0 to 7 | *_Skt1No | UDP/TCP Socket Number |
| | 8 | *_Skt1KeepAlive | Keep-alive Function Used |
| | 9 | *_Skt1Linger | Linger Function Used |
| | 10 to 15 | --- | --- |
| CIO m+18 | 0 to 15 | *_Skt1LocalPortNo | Local UDP/TCP Port Number (0000 to FFFF hex) |
| CIO m+19 | 0 to 7 | *_Skt1RemoteIPAdr2 | 2nd Byte of Remote IP Address (00 to FF hex) |
| | 8 to 15 | *_Skt1RemoteIPAdr1 | 1st Byte of Remote IP Address (00 to FF hex) |
| CIO m+20 | 0 to 7 | *_Skt1RemoteIPAdr4 | 4th Byte of Remote IP Address (00 to FF hex) |
| | 8 to 15 | *_Skt1RemoteIPAdr3 | 3rd Byte of Remote IP Address (00 to FF hex) |
| CIO m+21 | 0 to 15 | *_Skt1RemotePortNo | Remote UDP/TCP Port Number (0000 to FFFF hex) |
| CIO m+22 | 0 to 15 | *_Skt1SendRcvByte | Number Of Send/Receive Bytes (0000 to 07C0 hex (1984)) |
| CIO m+23 | 0 to 7 | *_Skt1SendRcvAdrCh1 | Send/Receive Data Address (Channel Address(Upper 2 digits)) |
| | 8 to 15 | *_Skt1SendRcvAdrType | Send/Receive Data Address (Area Type) |
| CIO m+24 | 0 to 7 | --- | --- |
| | 8 to 15 | *_Skt1SendRcvAdrCh2 | Send/Receive Data Address (Channel Address(Lower 2 digits)) |
| CIO m+25 | 0 to 15 | *_Skt1Timeout | Timeout Value (0000 to FFFF hex) |
| CIO m+26 | 0 to 15 | *_Skt1RespCode | Response Code |

Subsequent sockets, Socket No. 2 to No. 8, are assigned similar variables. The beginning word address is sequentially incremented by +10.

(This table is omitted for Socket No. 2 to No. 7.)

Socket No. 8

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| CIO m+87 | 0 to 7 | *_Skt8No | UDP/TCP Socket Number |
| | 8 | *_Skt8KeepAlive | Keep-alive Function Used |
| | 9 | *_Skt8Linger | Linger Function Used |
| | 10 to 15 | --- | --- |
| CIO m+88 | 0 to 15 | *_Skt8LocalPortNo | Local UDP/TCP Port Number (0000 to FFFF hex) |
| CIO m+89 | 0 to 7 | *_Skt8RemoteIPAdr2 | 2nd Byte of Remote IP Address (00 to FF hex) |
| | 8 to 15 | *_Skt8RemoteIPAdr1 | 1st Byte of Remote IP Address (00 to FF hex) |
| CIO m+90 | 0 to 7 | *_Skt8RemoteIPAdr4 | 4th Byte of Remote IP Address (00 to FF hex) |
| | 8 to 15 | *_Skt8RemoteIPAdr3 | 3rd Byte of Remote IP Address (00 to FF hex) |
| CIO m+91 | 0 to 15 | *_Skt8RemotePortNo | Remote UDP/TCP Port Number (0000 to FFFF hex) |
| CIO m+92 | 0 to 15 | *_Skt8SendRcvByte | Number Of Send/Receive Bytes (0000 to 07C0 hex (1984)) |
| CIO m+93 | 0 to 7 | *_Skt8SendRcvAdrCh1 | Send/Receive Data Address (Channel Address(Upper 2 digits)) |
| | 8 to 15 | *_Skt8SendRcvAdrType | Send/Receive Data Address (Area Type) |
| CIO m+94 | 0 to 7 | --- | --- |
| | 8 to 15 | *_Skt8SendRcvAdrCh2 | Send/Receive Data Address (Channel Address(Lower 2 digits)) |
| CIO m+95 | 0 to 15 | *_Skt8Timeout | Timeout Value (0000 to FFFF hex) |
| CIO m+96 | 0 to 15 | *_Skt8RespCode | Response Code |

## User Settings Area

x: First I/O memory area address that is specified in the allocated CIO Area words.

### ● Words x to x+15: Registered Target Node Table

The device variable that corresponds to all of the bits in words x to x+15 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| **Word address** | **Bit numbers** | **Variable name** | **Description** |
| Words x to x+15 | 00 to 15 in each word | *_x.RegTargetSta.TargetStaWd | Registered Target Node Table<br>(The functions of bits 00 to 15 in these words and the functions of the device variable given on the left correspond as given below.)<br>• Bits 00 to 15 of word x correspond to bits 00 to 15 of the device variable given on the left.<br>• Bits 00 to 15 of word x +1 correspond to bits 16 to 31 of the device variable given on the left.<br>• Bits 00 to 15 of word x +2 correspond to bits 32 to 47 of the device variable given on the left.<br>• Bits 00 to 15 of word x+3 correspond to bits 48 to 63 of the device variable given on the left.<br>• Bits 00 to 15 of word x+4 correspond to bits 64 to 79 of the device variable given on the left.<br>• Bits 00 to 15 of word x+5 correspond to bits 80 to 95 of the device variable given on the left.<br>• Bits 00 to 15 of word x+6 correspond to bits 96 to 111 of the device variable given on the left.<br>• Bits 00 to 15 of word x+7 correspond to bits 112 to 127 of the device variable given on the left.<br>• Bits 00 to 15 of word x+8 correspond to bits 128 to 143 of the device variable given on the left.<br>• Bits 00 to 15 of word x+9 correspond to bits 144 to 159 of the device variable given on the left.<br>• Bits 00 to 15 of word x+10 correspond to bits 160 to 175 of the device variable given on the left.<br>• Bits 00 to 15 of word x+11 correspond to bits 176 to 191 of the device variable given on the left.<br>• Bits 00 to 15 of word x+12 correspond to bits 192 to 207 of the device variable given on the left.<br>• Bits 00 to 15 of word x+13 correspond to bits 208 to 223 of the device variable given on the left.<br>• Bits 00 to 15 of word x+14 correspond to bits 224 to 239 of the device variable given on the left.<br>• Bits 00 to 15 of word x+15 correspond to bits 240 to 255 of the device variable given on the left. |

The device variables that correspond to bits 00 to 15 in words x to x+15 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| Word x | 0 | *_x.RegTargetSta.TargetSta[0] | Registered Target Node Table Bit for Node Address 0 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[15] | Registered Target Node Table Bit for Node Address 15 |
| Word x+1 | 0 | *_x.RegTargetSta.TargetSta[16] | Registered Target Node Table Bit for Node Address 16 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[31] | Registered Target Node Table Bit for Node Address 31 |
| Word x+2 | 0 | *_x.RegTargetSta.TargetSta[32] | Registered Target Node Table Bit for Node Address 32 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[47] | Registered Target Node Table Bit for Node Address 47 |
| Word x+3 | 0 | *_x.RegTargetSta.TargetSta[48] | Registered Target Node Table Bit for Node Address 48 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[63] | Registered Target Node Table Bit for Node Address 63 |
| Word x+4 | 0 | *_x.RegTargetSta.TargetSta[64] | Registered Target Node Table Bit for Node Address 64 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[79] | Registered Target Node Table Bit for Node Address 79 |
| Word x+5 | 0 | *_x.RegTargetSta.TargetSta[80] | Registered Target Node Table Bit for Node Address 80 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[95] | Registered Target Node Table Bit for Node Address 95 |
| Word x+6 | 0 | *_x.RegTargetSta.TargetSta[96] | Registered Target Node Table Bit for Node Address 96 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[111] | Registered Target Node Table Bit for Node Address 111 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| Word x+7 | 0 | *_x.RegTargetSta.TargetSta[112] | Registered Target Node Table Bit for Node Address 112 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[127] | Registered Target Node Table Bit for Node Address 127 |
| Word x+8 | 0 | *_x.RegTargetSta.TargetSta[128] | Registered Target Node Table Bit for Node Address 128 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[143] | Registered Target Node Table Bit for Node Address 143 |
| Word x+9 | 0 | *_x.RegTargetSta.TargetSta[144] | Registered Target Node Table Bit for Node Address 144 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[159] | Registered Target Node Table Bit for Node Address 159 |
| Word x+10 | 0 | *_x.RegTargetSta.TargetSta[160] | Registered Target Node Table Bit for Node Address 160 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[175] | Registered Target Node Table Bit for Node Address 175 |
| Word x+11 | 0 | *_x.RegTargetSta.TargetSta[176] | Registered Target Node Table Bit for Node Address 176 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[191] | Registered Target Node Table Bit for Node Address 191 |
| Word x+12 | 0 | *_x.RegTargetSta.TargetSta[192] | Registered Target Node Table Bit for Node Address 192 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[207] | Registered Target Node Table Bit for Node Address 207 |
| Word x+13 | 0 | *_x.RegTargetSta.TargetSta[208] | Registered Target Node Table Bit for Node Address 208 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[223] | Registered Target Node Table Bit for Node Address 223 |
| Word x+14 | 0 | *_x.RegTargetSta.TargetSta[224] | Registered Target Node Table Bit for Node Address 224 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[239] | Registered Target Node Table Bit for Node Address 239 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| Word x+15 | 0 | *_x.RegTargetSta.TargetSta[240] | Registered Target Node Table Bit for Node Address 240 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.RegTargetSta.TargetSta[255] | Registered Target Node Table Bit for Node Address 255 |

### ● Words x+16 to x+31: Normal Target Node Table

The device variable that corresponds to all of the bits in words x+16 to x+31 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| Words x+16 to x+31 | 00 to 15 in each word | *_x.EstbTargetSta.TargetStaWd | Normal Target Node Table<br>(The functions of bits 00 to 15 in these words and the functions of the device variable given on the left correspond as given below.)<br>• Bits 00 to 15 of word x+16 correspond to bits 00 to 15 of the device variable given on the left.<br>• Bits 00 to 15 of word x+17 correspond to bits 16 to 31 of the device variable given on the left.<br>• Bits 00 to 15 of word x+18 correspond to bits 32 to 47 of the device variable given on the left.<br>• Bits 00 to 15 of word x+19 correspond to bits 48 to 63 of the device variable given on the left.<br>• Bits 00 to 15 of word x+20 correspond to bits 64 to 79 of the device variable given on the left.<br>• Bits 00 to 15 of word x+21 correspond to bits 80 to 95 of the device variable given on the left.<br>• Bits 00 to 15 of word x+22 correspond to bits 96 to 111 of the device variable given on the left.<br>• Bits 00 to 15 of word x+23 correspond to bits 112 to 127 of the device variable given on the left.<br>• Bits 00 to 15 of word x+24 correspond to bits 128 to 143 of the device variable given on the left.<br>• Bits 00 to 15 of word x+25 correspond to bits 144 to 159 of the device variable given on the left.<br>• Bits 00 to 15 of word x+26 correspond to bits 160 to 175 of the device variable given on the left.<br>• Bits 00 to 15 of word x+27 correspond to bits 176 to 191 of the device variable given on the left.<br>• Bits 00 to 15 of word x+28 correspond to bits 192 to 207 of the device variable given on the left.<br>• Bits 00 to 15 of word x+29 correspond to bits 208 to 223 of the device variable given on the left.<br>• Bits 00 to 15 of word x+30 correspond to bits 224 to 239 of the device variable given on the left.<br>• Bits 00 to 15 of word x+31 correspond to bits 240 to 255 of the device variable given on the left. |

The device variables that correspond to bits 00 to 15 in words x+16 to x+31 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| Word x+16 | 0 | *_x.EstbTargetSta.TargetSta[0] | Normal Target Node Table Bit for Node Address 0 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTargetSta.TargetSta[15] | Normal Target Node Table Bit for Node Address 15 |
| Word x+17 | 0 | *_x.EstbTargetSta.TargetSta[16] | Normal Target Node Table Bit for Node Address 16 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTargetSta.TargetSta[31] | Normal Target Node Table Bit for Node Address 31 |
| Word x+18 | 0 | *_x.EstbTargetSta.TargetSta[32] | Normal Target Node Table Bit for Node Address 32 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTargetSta.TargetSta[47] | Normal Target Node Table Bit for Node Address 47 |
| Word x+19 | 0 | *_x.EstbTargetSta.TargetSta[48] | Normal Target Node Table Bit for Node Address 48 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTargetSta.TargetSta[63] | Normal Target Node Table Bit for Node Address 63 |
| Word x+20 | 0 | *_x.EstbTargetSta.TargetSta[64] | Normal Target Node Table Bit for Node Address 64 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTargetSta.TargetSta[79] | Normal Target Node Table Bit for Node Address 79 |
| Word x+21 | 0 | *_x.EstbTargetSta.TargetSta[80] | Normal Target Node Table Bit for Node Address 80 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTargetSta.TargetSta[95] | Normal Target Node Table Bit for Node Address 95 |
| Word x+22 | 0 | *_x.EstbTargetSta.TargetSta[96] | Normal Target Node Table Bit for Node Address 96 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTargetSta.TargetSta[111] | Normal Target Node Table Bit for Node Address 111 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| Word x+23 | 0 | *_x.EstbTarget-Sta.TargetSta[112] | Normal Target Node Table Bit for Node Address 112 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTarget-Sta.TargetSta[127] | Normal Target Node Table Bit for Node Address 127 |
| Word x+24 | 0 | *_x.EstbTarget-Sta.TargetSta[128] | Normal Target Node Table Bit for Node Address 128 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTarget-Sta.TargetSta[143] | Normal Target Node Table Bit for Node Address 143 |
| Word x+25 | 0 | *_x.EstbTarget-Sta.TargetSta[144] | Normal Target Node Table Bit for Node Address 144 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTarget-Sta.TargetSta[159] | Normal Target Node Table Bit for Node Address 159 |
| Word x+26 | 0 | *_x.EstbTarget-Sta.TargetSta[160] | Normal Target Node Table Bit for Node Address 160 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTarget-Sta.TargetSta[175] | Normal Target Node Table Bit for Node Address 175 |
| Word x+27 | 0 | *_x.EstbTarget-Sta.TargetSta[176] | Normal Target Node Table Bit for Node Address 176 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTarget-Sta.TargetSta[191] | Normal Target Node Table Bit for Node Address 191 |
| Word x+28 | 0 | *_x.EstbTarget-Sta.TargetSta[192] | Normal Target Node Table Bit for Node Address 192 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTarget-Sta.TargetSta[207] | Normal Target Node Table Bit for Node Address 207 |
| Word x+29 | 0 | *_x.EstbTarget-Sta.TargetSta[208] | Normal Target Node Table Bit for Node Address 208 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTarget-Sta.TargetSta[223] | Normal Target Node Table Bit for Node Address 223 |
| Word x+30 | 0 | *_x.EstbTarget-Sta.TargetSta[224] | Normal Target Node Table Bit for Node Address 224 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTarget-Sta.TargetSta[239] | Normal Target Node Table Bit for Node Address 239 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| **Word address** | **Bit number** | **Variable name** | **Description** |
| Word x+31 | 0 | *_x.EstbTarget-Sta.TargetSta[240] | Normal Target Node Table Bit for Node Address 240 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.EstbTarget-Sta.TargetSta[255] | Normal Target Node Table Bit for Node Address 255 |

### Words x+32 to x+47: Target Nod4e PLC Operating Flags

The device variable that corresponds to all of the bits in words x+32 to x+47 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| Words x+32 to x+47 | 00 to 15 in each word | *_x.TargetPLCMdSta.TargetStaWd | Target Node PLC Operating Flags<br>(The functions of bits 00 to 15 in these words and the functions of the device variable given on the left correspond as given below.)<br>• Bits 00 to 15 of word x+32 correspond to bits 00 to 15 of the device variable given on the left.<br>• Bits 00 to 15 of word x+33 correspond to bits 16 to 31 of the device variable given on the left.<br>• Bits 00 to 15 of word x+34 correspond to bits 32 to 47 of the device variable given on the left.<br>• Bits 00 to 15 of word x+35 correspond to bits 48 to 63 of the device variable given on the left.<br>• Bits 00 to 15 of word x+36 correspond to bits 64 to 79 of the device variable given on the left.<br>• Bits 00 to 15 of word x+37 correspond to bits 80 to 95 of the device variable given on the left.<br>• Bits 00 to 15 of word x+38 correspond to bits 96 to 111 of the device variable given on the left.<br>• Bits 00 to 15 of word x+39 correspond to bits 112 to 127 of the device variable given on the left.<br>• Bits 00 to 15 of word x+40 correspond to bits 128 to 143 of the device variable given on the left.<br>• Bits 00 to 15 of word x+41 correspond to bits 144 to 159 of the device variable given on the left.<br>• Bits 00 to 15 of word x+42 correspond to bits 160 to 175 of the device variable given on the left.<br>• Bits 00 to 15 of word x+43 correspond to bits 176 to 191 of the device variable given on the left.<br>• Bits 00 to 15 of word x+44 correspond to bits 192 to 207 of the device variable given on the left.<br>• Bits 00 to 15 of word x+45 correspond to bits 208 to 223 of the device variable given on the left.<br>• Bits 00 to 15 of word x+46 correspond to bits 224 to 239 of the device variable given on the left.<br>• Bits 00 to 15 of word x+47 correspond to bits 240 to 255 of the device variable given on the left. |

The device variables that correspond to bits 00 to 15 in words x+32 to x+47 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| Word x+32 | 0 | *_x.TargetPLCMdSta. TargetSta[0] | Target Node PLC Operating Flag for Node Address 0 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta. TargetSta[15] | Target Node PLC Operating Flag for Node Address 15 |
| Word x+33 | 0 | *_x.TargetPLCMdSta. TargetSta[16] | Target Node PLC Operating Flag for Node Address 16 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta. TargetSta[31] | Target Node PLC Operating Flag for Node Address 31 |
| Word x+34 | 0 | *_x.TargetPLCMdSta. TargetSta[32] | Target Node PLC Operating Flag for Node Address 32 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta. TargetSta[47] | Target Node PLC Operating Flag for Node Address 47 |
| Word x+35 | 0 | *_x.TargetPLCMdSta. TargetSta[48] | Target Node PLC Operating Flag for Node Address 48 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta. TargetSta[63] | Target Node PLC Operating Flag for Node Address 63 |
| Word x+36 | 0 | *_x.TargetPLCMdSta. TargetSta[64] | Target Node PLC Operating Flag for Node Address 64 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta. TargetSta[79] | Target Node PLC Operating Flag for Node Address 79 |
| Word x+37 | 0 | *_x.TargetPLCMdSta. TargetSta[80] | Target Node PLC Operating Flag for Node Address 80 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta. TargetSta[95] | Target Node PLC Operating Flag for Node Address 95 |
| Word x+38 | 0 | *_x.TargetPLCMdSta. TargetSta[96] | Target Node PLC Operating Flag for Node Address 96 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta. TargetSta[111] | Target Node PLC Operating Flag for Node Address 111 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| Word x+39 | 0 | *_x.TargetPLCMdSta.TargetSta[112] | Target Node PLC Operating Flag for Node Address 112 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta.TargetSta[127] | Target Node PLC Operating Flag for Node Address 127 |
| Word x+40 | 0 | *_x.TargetPLCMdSta.TargetSta[128] | Target Node PLC Operating Flag for Node Address 128 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta.TargetSta[143] | Target Node PLC Operating Flag for Node Address 143 |
| Word x+41 | 0 | *_x.TargetPLCMdSta.TargetSta[144] | Target Node PLC Operating Flag for Node Address 144 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta.TargetSta[159] | Target Node PLC Operating Flag for Node Address 159 |
| Word x+42 | 0 | *_x.TargetPLCMdSta.TargetSta[160] | Target Node PLC Operating Flag for Node Address 160 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta.TargetSta[175] | Target Node PLC Operating Flag for Node Address 175 |
| Word x+43 | 0 | *_x.TargetPLCMdSta.TargetSta[176] | Target Node PLC Operating Flag for Node Address 176 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta.TargetSta[191] | Target Node PLC Operating Flag for Node Address 191 |
| Word x+44 | 0 | *_x.TargetPLCMdSta.TargetSta[192] | Target Node PLC Operating Flag for Node Address 192 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta.TargetSta[207] | Target Node PLC Operating Flag for Node Address 207 |
| Word x+45 | 0 | *_x.TargetPLCMdSta.TargetSta[208] | Target Node PLC Operating Flag for Node Address 208 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta.TargetSta[223] | Target Node PLC Operating Flag for Node Address 223 |
| Word x+46 | 0 | *_x.TargetPLCMdSta.TargetSta[224] | Target Node PLC Operating Flag for Node Address 224 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta.TargetSta[239] | Target Node PLC Operating Flag for Node Address 239 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| **Word address** | **Bit number** | **Variable name** | **Description** |
| Word x+47 | 0 | *_x.TargetPLCMdSta.TargetSta[240] | Target Node PLC Operating Flag for Node Address 240 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCMdSta.TargetSta[255] | Target Node PLC Operating Flag for Node Address 255 |

### ● Words x+48 to x+63: Target Node PLC Error Flags

The device variable that corresponds to all of the bits in words x+48 to x+63 is given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit numbers | Variable name | Description |
| Words x+48 to x+63 | 00 to 15 in each word | *_x.TargetPLCErrSta.TargetStaWd | Target Node PLC Error Flags<br>(The functions of bits 00 to 15 in these words and the functions of the device variable given on the left correspond as given below.)<br>• Bits 00 to 15 of word x+48 correspond to bits 00 to 15 of the device variable given on the left.<br>• Bits 00 to 15 of word x+49 correspond to bits 16 to 31 of the device variable given on the left.<br>• Bits 00 to 15 of word x+50 correspond to bits 32 to 47 of the device variable given on the left.<br>• Bits 00 to 15 of word x+51 correspond to bits 48 to 63 of the device variable given on the left.<br>• Bits 00 to 15 of word x+52 correspond to bits 64 to 79 of the device variable given on the left.<br>• Bits 00 to 15 of word x+53 correspond to bits 80 to 95 of the device variable given on the left.<br>• Bits 00 to 15 of word x+54 correspond to bits 96 to 111 of the device variable given on the left.<br>• Bits 00 to 15 of word x+55 correspond to bits 112 to 127 of the device variable given on the left.<br>• Bits 00 to 15 of word x+56 correspond to bits 128 to 143 of the device variable given on the left.<br>• Bits 00 to 15 of word x+57 correspond to bits 144 to 159 of the device variable given on the left.<br>• Bits 00 to 15 of word x+58 correspond to bits 160 to 175 of the device variable given on the left.<br>• Bits 00 to 15 of word x+59 correspond to bits 176 to 191 of the device variable given on the left.<br>• Bits 00 to 15 of word x+60 correspond to bits 192 to 207 of the device variable given on the left.<br>• Bits 00 to 15 of word x+61 correspond to bits 208 to 223 of the device variable given on the left.<br>• Bits 00 to 15 of word x+62 correspond to bits 224 to 239 of the device variable given on the left.<br>• Bits 00 to 15 of word x+63 correspond to bits 240 to 255 of the device variable given on the left. |

The device variables that correspond to bits 00 to 15 in words x+48 to x+63 are given in the following table.

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| Word x+48 | 0 | *_x.TargetPLCErrSta. TargetSta[0] | Target Node PLC Error Flag for Node Address 0 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta. TargetSta[15] | Target Node PLC Error Flag for Node Address 15 |
| Word x+49 | 0 | *_x.TargetPLCErrSta. TargetSta[16] | Target Node PLC Error Flag for Node Address 16 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta. TargetSta[31] | Target Node PLC Error Flag for Node Address 31 |
| Word x+50 | 0 | *_x.TargetPLCErrSta. TargetSta[32] | Target Node PLC Error Flag for Node Address 32 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta. TargetSta[47] | Target Node PLC Error Flag for Node Address 47 |
| Word x+51 | 0 | *_x.TargetPLCErrSta. TargetSta[48] | Target Node PLC Error Flag for Node Address 48 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta. TargetSta[63] | Target Node PLC Error Flag for Node Address 63 |
| Word x+52 | 0 | *_x.TargetPLCErrSta. TargetSta[64] | Target Node PLC Error Flag for Node Address 64 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta. TargetSta[79] | Target Node PLC Error Flag for Node Address 79 |
| Word x+53 | 0 | *_x.TargetPLCErrSta. TargetSta[80] | Target Node PLC Error Flag for Node Address 80 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta. TargetSta[95] | Target Node PLC Error Flag for Node Address 95 |
| Word x+54 | 0 | *_x.TargetPLCErrSta. TargetSta[96] | Target Node PLC Error Flag for Node Address 96 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta. TargetSta[111] | Target Node PLC Error Flag for Node Address 111 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| Word x+55 | 0 | *_x.TargetPLCErrSta.TargetSta[112] | Target Node PLC Error Flag for Node Address 112 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta.TargetSta[127] | Target Node PLC Error Flag for Node Address 127 |
| Word x+56 | 0 | *_x.TargetPLCErrSta.TargetSta[128] | Target Node PLC Error Flag for Node Address 128 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta.TargetSta[143] | Target Node PLC Error Flag for Node Address 143 |
| Word x+57 | 0 | *_x.TargetPLCErrSta.TargetSta[144] | Target Node PLC Error Flag for Node Address 144 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta.TargetSta[159] | Target Node PLC Error Flag for Node Address 159 |
| Word x+58 | 0 | *_x.TargetPLCErrSta.TargetSta[160] | Target Node PLC Error Flag for Node Address 160 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta.TargetSta[175] | Target Node PLC Error Flag for Node Address 175 |
| Word x+59 | 0 | *_x.TargetPLCErrSta.TargetSta[176] | Target Node PLC Error Flag for Node Address 176 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta.TargetSta[191] | Target Node PLC Error Flag for Node Address 191 |
| Word x+60 | 0 | *_x.TargetPLCErrSta.TargetSta[192] | Target Node PLC Error Flag for Node Address 192 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta.TargetSta[207] | Target Node PLC Error Flag for Node Address 207 |
| Word x+61 | 0 | *_x.TargetPLCErrSta.TargetSta[208] | Target Node PLC Error Flag for Node Address 208 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta.TargetSta[223] | Target Node PLC Error Flag for Node Address 223 |
| Word x+62 | 0 | *_x.TargetPLCErrSta.TargetSta[224] | Target Node PLC Error Flag for Node Address 224 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta.TargetSta[239] | Target Node PLC Error Flag for Node Address 239 |

| I/O memory location in CJ-series CPU Unit | | Device variable for the CJ-series Unit in NJ-series CPU Unit | |
|---|---|---|---|
| Word address | Bit number | Variable name | Description |
| Word x+63 | 0 | *_x.TargetPLCErrSta. TargetSta[240] | Target Node PLC Error Flag for Node Address 240 |
| | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| | 15 | *_x.TargetPLCErrSta. TargetSta[255] | Target Node PLC Error Flag for Node Address 255 |

# A-8 Replacing a System Using the CJ1W-EIP21 with a System Using the CJ1W-EIP21S

## A-8-1 Differences in Specifications, Functions, Etc.

This section describes the differences in specifications, functions, etc. and the restrictions that apply when you replace the CJ1W-EIP21 with the CJ1W-EIP21S. Check the differences in specifications, functions, etc. between the CJ1W-EIP21 and the CJ1W-EIP21S, and the restrictions that apply, to determine the effects of replacement on the hardware, user program, and applications in the source replacement system.

The following items are described.

| Item | Reference |
|---|---|
| Differences in Unit Functions and Specifications | *Differences in Unit Functions and Specifications* on page A-105 |
| Differences in Memory Allocation | *Differences in Memory Allocation* on page A-108 |
| Restrictions | *Restrictions* on page A-109 |

**Additional Information**

CJ1W-EIP21S EtherNet/IP Units with lot number 241001□ or later can be mounted on NJ-series CPU Units. These Units are printed with "+NJ" at the lower right of the front panel.

### Differences in Unit Functions and Specifications

The tables below show functions and specifications that differ and the effects of the replacement.

| Item | | Source replacement model | Target replacement model | Effects of replacement |
|---|---|---|---|---|
| | | CJ1W-EIP21 | CJ1W-EIP21S<br>Lot. number: 241001□ or later | |
| Weight | | 94 g max. | 91 g max. | The weight decreases 3 g. There is no effect. |
| Current consumption | | 0.41 A max. at 5 VDC | 0.65 A max. at 5 VDC | The current consumption increases 0.24 A. If the power supply is insufficient, you need to change the system configuration to use an Expansion Rack. |
| Applicable CPU Units | | NJ-series CPU Unit: Version 1.01 or later | NJ-series CPU Unit: Version 1.67 or later | There is no effect. |
| CPU Unit words used | Allocated CIO Area words (CPU Bus Unit words) | Socket service functions not available | Socket service functions available | There is no difference in the words used. However, socket service functions are added to reserved and unused bits for the source replacement model. For details on allocation and effects, refer to *Differences in Memory Allocation* on page A-108 |
| | Allocated DM Area words (CPU Bus Unit words) | | | |

| Item | | Source replacement model | Target replacement model | Effects of replacement |
|---|---|---|---|---|
| | | CJ1W-EIP21 | CJ1W-EIP21S<br>Lot. number: 241001□<br>or later | |
| CIP communications service: Tag data links (cyclic communications) | Allowed communications bandwidth per Unit | • 12,000 pps (Unit version 3.0 or later)<br>• 6,000 pps (Unit version 2.1 or earlier) | 12,000 pps | There is no effect. |
| | Maximum number of tags that can be refreshed per CPU Unit task period | Output/Transmission (CPU to EtherNet/IP): 19<br>Input/Reception (EtherNet/IP to CPU): 20<br>(19 for input/reception when User defined is set for the layout of the allocated CIO Area words) | Output/Transmission (CPU to EtherNet/IP): 19<br>Input/Reception (EtherNet/IP to CPU): 20<br>(18 for both of input/reception and output/transmission when User defined is set for the layout of the allocated CIO Area words) | The maximum number of tags in input/reception data will decrease when User defined is set for the layout of the allocated CIO Area words. If the maximum data size is exceeded, refreshing the data in the CPU Unit will span multiple periods. Check the effects on communications performance. |
| | Data that can be refreshed per CPU Unit task period | Output/Transmission (CPU to EtherNet/IP): 7,469 words<br>Input/Reception (EtherNet/IP to CPU): 7,469 words<br>(7,405 words for both input and output when User defined is set for the layout of the allocated CIO Area words) | Output/Transmission (CPU to EtherNet/IP): 7,469 words<br>Input/Reception (EtherNet/IP to CPU): 7,469 words<br>(7,321 words for output and 7,385 words for input when User defined is set for the layout of the allocated CIO Area words) | The data size will decrease when User defined is set for the layout of the allocated CIO Area words. If the maximum data size is exceeded, refreshing the data in the CPU Unit will span multiple periods. Check the effects on communications performance. |
| FINS communications service | Setting and default for using or not using FINS/UDP | • Setting function: Not available<br>• Default: Use | • Setting function: Available<br>Default: Do not use | To use the FINS function, you need to set **FINS/UDP** to *Use* in the FINS Settings Display. |
| | Setting and default for using or not using FINS/TCP | • Setting function: Not available<br>• Default: Use | • Setting function: Available<br>Default: Do not use | To use the FINS function, you need to set **FINS/TCP** to *Use* in the FINS Settings Display. |
| | Keep-alive setting | Fixed to 5 min when the keep-alive function is in effect. | Can be set in the TCP/IP Tab Page. The default is 5 min. | There is no effect. |
| FTP server | | Available (Unit version 2.0 or later) | Available | There is no effect. |
| | Default for using or not using FTP server | Default: Use | Default: Do not use | To use the FTP function, you need to set **FTP server** to *Use* in the FTP Settings Display. |
| | User name/Password | • User name<br>1 to 12 alphanumeric characters<br>Default: CONFIDENTIAL<br>• Password<br>1 to 8 alphanumeric characters<br>Default: - (Not set)<br>Entered characters displayed | • User name<br>1 to 16 alphanumeric characters<br>Default: - (Not set)<br>• Password<br>8 to 16 alphanumeric characters<br>Default: - (Not set)<br>Entered characters replaced by asterisks | To use the FTP function, you need to set the user name (login name) and password again. |

| Item | | Source replacement model | Target replacement model | Effects of replacement |
|---|---|---|---|---|
| | | CJ1W-EIP21 | CJ1W-EIP21S<br>Lot. number: 241001□<br>or later | |
| Automatic clock adjustment | | Available (Unit version 2.0 or later) | Available | There is no effect. |
| BOOTP client | Get the IP address from the BOOTP server at each power ON or restart | Not available | Available | There is no effect. |
| Backup function | Backup data capacity | 459 KB<br>(Simple backup function) | 364 KB<br>(SD Memory Card backup functions and Sysmac Studio Controller Backups) | There is no effect. |
| SNMP/SNMP trap | | Available (Unit version 2.0 or later)<br>Community name: Entered characters displayed | Available<br>Community name: Entered characters replaced by asterisks | There is no effect. |
| CIDR function for IP addresses | | Available (Unit version 2.0 or later) | Available | There is no effect. |
| Setting for using or not using CIP message server | | • Setting function: Not available<br>• Default: Use | • Setting function: Available<br>Default: Use | There is no effect. |
| Security | Packet filter | Not available | Available | After replacement, determine whether or not to use this function to further reduce security risks according to your system operation. |
| | Opening and closing the port [*1] | Without opening and closing the port for the following services.<br>• FINS/UDP service<br>• FINS/TCP service<br>• CIP message server | With opening and closing the port for the following services.<br>• FINS/UDP service<br>• FINS/TCP service<br>• CIP message server | Before replacement, you need to configure the port for the service in use. |
| Socket services | | Not available | Available | There is no effect. |

[*1] Both the CJ1W-EIP21 and the CJ1W-EIP21S support opening and closing the port for the following services.
FTP server and SNMP

## Differences in Device Variables

| Item | Source replacement model | Target replacement model | Effects of replacement |
|---|---|---|---|
| | CJ1W-EIP21 | CJ1W-EIP21S<br>Lot. number: 241001□<br>or later | |
| Device variables for socket services | None | Available | There is no effect. |
| Variables that indicate the status of tag data links when the status area layout is set to *User setting* as listed below<br>• TargetPLCMdSta (Target Node PLC Operating Information)<br>• TargetPLCErrSta (Target Node PLC Error Information)<br>• RegTargetSta (Registered Target Node)<br>• EstbTargetSta (Normal Target Node) | (Reserved) | Undefined | To use socket services with the CJ1W-EIP21S, set the status area layout to *User setting*.<br>At this time, instead of the default variables, the user-defined variable set in **Mapped area** in the Status Area Settings Display should be referenced.<br>• TargetPLCMdSta (Target Node PLC Operating Information)<br>• TargetPLCErrSta (Target Node PLC Error Information)<br>• RegTargetSta (Registered Target Node)<br>EstbTargetSta (Normal Target Node) |

## Differences in Memory Allocation

The table below shows functions that differ in memory allocation between the CJ1W-EIP21 and the CJ1W-EIP21S. Refer to the *CS/CJ-series EtherNet/IP Units Operation Manual* (Cat. No. W465) for details on functions that differ.

| Item | | Source replacement model | Target replacement model | Effects of replacement |
|---|---|---|---|---|
| | | CJ1W-EIP21 | CJ1W-EIP21S<br>Lot number 241001□<br>or later | |
| Allocated CIO Area words (CPU Bus Unit words) | | | | |
| User settings area | n+1 to n+4 | Reserved | Socket Service Request Switches (CPU Unit to EtherNet/IP Unit) | When User defined is set for the layout of the allocated CIO Area words, socket service functions are added to the reserved area words of the source replacement model. Make sure that the reserved area words are not manipulated by the user program. |
| | n+16 to n+23 | Reserved | UDP Socket No. □ Status and TCP Socket No. □ Status (EtherNet/IP Unit to CPU Unit) | |
| Software Switches (CPU Unit to EtherNet/IP Unit) (n) | Bit 10 | Not used | Force Socket Close Switch (CPU Unit to EtherNet/IP Unit) | Socket service functions are added to the target replacement models. Make sure that the reserved bits are not manipulated by the user program. |

| Item | | Source replacement model | Target replacement model | Effects of replacement |
|---|---|---|---|---|
| | | CJ1W-EIP21 | CJ1W-EIP21S<br>Lot number 241001□<br>or later | |
| Unit Status 2 (EtherNet/IP Unit to CPU Unit) (n+11) | Bit 10 | Not used | Reserved | |
| | Bit 12 and Bit 13 | Not used | Reserved | |
| Allocated DM Area words (CPU Bus Unit words) | | | | |
| m+1 to m+8 | | Not used | Number of Bytes Received at TCP Socket No. □ (EtherNet/IP Unit to CPU Unit) | Socket service functions are added to the target replacement models. If the area words are used by the user program, the meaning of the data will change. Consider changing the data that was used before replacement to other available DM Area words. |
| m+9 to m+16 | | Not used | TCP Socket No. □ Connection Status (EtherNet/IP Unit to CPU Unit) | |
| m+17 to m+96 | | Not used | Socket Service Parameter Area □ (EtherNet/IP Unit to CPU Unit or CPU Unit to EtherNet/IP Unit) | |

## Restrictions

The table below shows restrictions on replacement.

| Item | Description | Effects of replacement |
|---|---|---|
| Backup/restoration | The CJ1W-EIP21S has more data to back up than the CJ1W-EIP21 due to the addition of security functions. It also has the same settings but different default values from those of the CJ1W-EIP21. | You cannot restore the backup data of the CJ1W-EIP21 to the CJ1W-EIP21S if it was backed up with the Sysmac Studio. It must be reconfigured in the Sysmac Studio as in the case of the CJ1W-EIP21. Refer to *Setting Up the CJ1W-EIP21S* on page A-111 for the setting procedure. |
| Unit startup time | The startup time of the CJ1W-EIP21S is longer than that of the CJ1W-EIP21. For this reason, the CPU Unit startup time will be delayed by a few seconds compared with the system before replacement. | This means that you need to check the effects on the startup operation of the system. |

## A-8-2 Replacement Flow

This section describes the flow for replacing a system using the CJ1W-EIP21 with a system using the CJ1W-EIP21S. It consists of the flow of preparation and the flow of replacement.

### Flow of Preparation

The table below shows the contents of preparation that you should make before replacement. Use this flow to plan a replacement and prepare data that matches the Unit configuration after replacement.

Make the target replacement Unit available before replacement.

| Step | Item | Description | Reference |
|---|---|---|---|
| 1 | Checking the Differences in Specifications, Functions, Etc. | Check the differences in specifications, functions, etc. between the source and target replacement models, and the restrictions that apply, to determine the effects of replacement on the hardware, user program, and applications in the source replacement system. | *Checking the Differences in Specifications, Functions, Etc.* on page A-111 |
| 2 | Getting the Support Software | Get the Support Software that you will use for preparation and replacement. | *Getting the Support Software* on page A-111 |
| 3 | Reading and Saving Data | Read and save the following data from the source replacement system.<br>You will use the data read in this step, and edit or set it in subsequent steps to match the target replacement system.<br>• Controller project data<br>• Tag data link parameters | *Reading and Saving Data* on page A-111 |
| 4 | Setting Up the CJ1W-EIP21S | Set up the CJ1W-EIP21S in the Unit configuration after replacement. | *Setting Up the CJ1W-EIP21S* on page A-111 |
| 5 | Setting Device Variables | Set device variables for the CJ1W-EIP21S in the Unit configuration after replacement. | *Setting Device Variables* on page A-113 |
| 6 | Editing the User Program | Edit the user program to deal with the differences in specifications due to the replacement and the restrictions that apply. | *Correcting the User Program* on page A-118 |
| 7 | Setting Tag Data Links | Configure tag data links to match the Unit configuration after replacement. | *Setting Tag Data Links* on page A-118 |

### Flow of Replacement

The table below shows the actual replacement steps that you should follow on completion of the preparation steps. You can implement security measures as needed after replacement.

| Step | Item | Description | Reference |
|---|---|---|---|
| 8 | Replacing and Wiring Units | Replace Units according to the Unit configuration after replacement. Then, wire the replaced Units. | *Replacing and Wiring Units* on page A-118 |
| 9 | Turning ON the Power Supply | Turn ON the power supply to the Controller. | *Turning ON the Power Supply* on page A-118 |
| 10 | Transferring Data | From the Support Software connected directly to the CPU Unit, transfer the following data that you prepared to the actual Controller after replacement.<br>• Controller project data<br>• Tag data link parameters | *Transferring Data* on page A-118 |

| Step | Item | Description | Reference |
|---|---|---|---|
| 11 | Checking the Operation | Confirm that the system operates as intended. | *Checking the Operation* on page A-119 |
| 12 | Implementing Security Measures *1 | The CJ1W-EIP21S as well as the Sysmac Studio provide security functions to reduce security risks. Use the security functions in conjunction with system operation to reduce the security risks. | *Implementing Security Measures* on page A-120 |
| 13 | Checking the Operation after Implementing Security Measures | After you implement security measures, check to be sure that the system operates according to the security settings. | *Checking the Operation after Implementing Security Measures* on page A-120 |

*1 Do this as needed.

## A-8-3 Preparation

This section describes the preparation steps that you should follow before replacement.

### Checking the Differences in Specifications, Functions, Etc.

Check the differences in specifications, functions, etc. between the CJ1W-EIP21 and the CJ1W-EIP21S, and the restrictions that apply, to determine the effects of replacement on the hardware, user program, and applications in the source replacement system.

For details, refer to *A-8-1 Differences in Specifications, Functions, Etc.* on page 105.

### Getting the Support Software

Get the Support Software that you will use for preparation and replacement.

Get a version of the Support Software shown in *Unit Versions and Programming Device Versions* on page 37.

### Reading and Saving Data

Read the following data from the source replacement system and save it in any folder on the computer or other storage device.

You will use the data read in this step, and edit or set it in subsequent steps to match the target replacement system.

- Controller project data
- Tag data link parameters

Perform this operation so that you can restore the data from the source replacement system. Even when you have the data in the source replacement system, there is a risk of trouble if it is different from the data in the actual Controller. We recommend that you read and save it.

### Setting Up the CJ1W-EIP21S

Set up the CJ1W-EIP21S in the Unit configuration after replacement.

Using the Sysmac Studio, set the parameters for the CJ1W-EIP21S. Configure the settings to match those of the source replacement model CJ1W-EIP21.

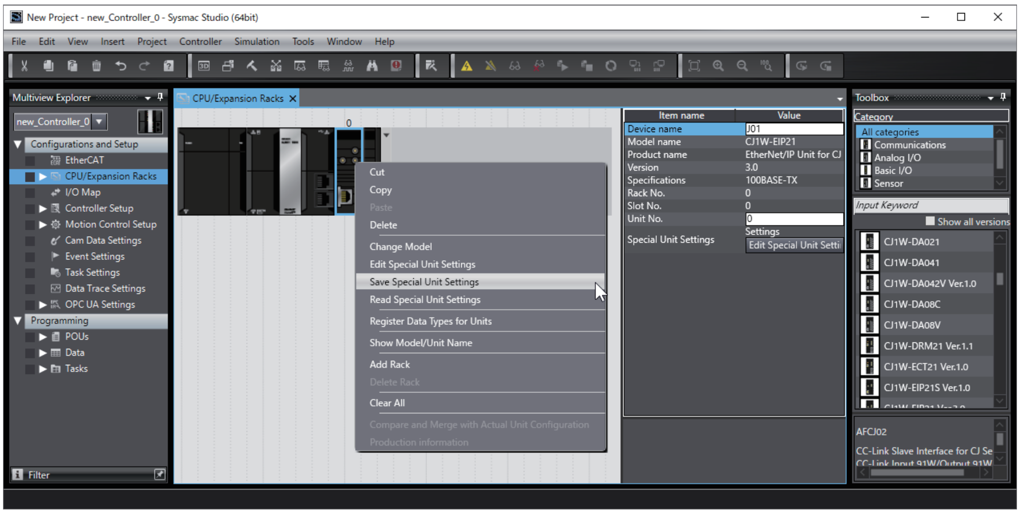
You can use the unit setting porting assistant in the Sysmac Studio to easily transfer the settings of the CJ1W-EIP21 to the CJ1W-EIP21S. In this case, the default values of the CJ1W-EIP21S will be used for settings that are not provided for the CJ1W-EIP21, but are available only with the CJ1W-EIP21S.

The following is the setting procedure using the unit setting porting assistant.

1. In the Sysmac Studio, open the project data from the source replacement model and double-click **CPU/Expansion Racks** under **Configurations and Setup** in the Multiview Explorer.

   The Unit Editor is displayed in the Edit Pane.

2. Right-click the CJ1W-EIP21 and select **Save Special Unit Settings** from the menu.



| Item name | Value |
|---|---|
| Device name | J01 |
| Model name | CJ1W-EIP21 |
| Product name | EtherNet/IP Unit for CJ |
| Version | 3.0 |
| Specifications | 100BASE-TX |
| Rack No. | 0 |
| Slot No. | 0 |
| Unit No. | 0 |
| Special Unit Settings | Settings<br>Edit Special Unit Setti |

   The Save File Dialog Box is displayed.

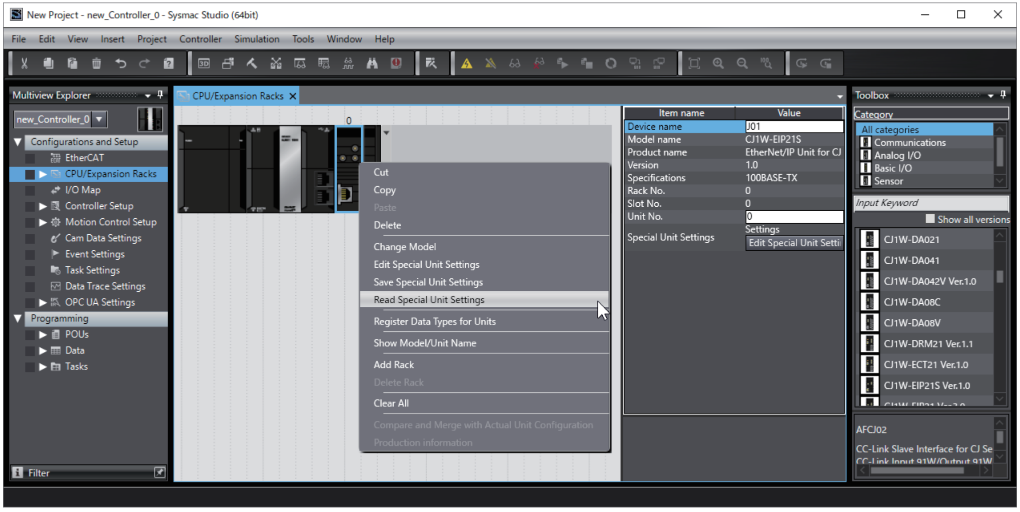
3. In the Save File Dialog Box, enter any file name, and then click the Save Button.

   The settings are saved into a file. The file extension is upf or upf2.

4. In the Sysmac Studio, open the project data in the target replacement model that contains the CJ1W-EIP21S in its configuration, and double-click **CPU/Expansion Racks** under **Configurations and Setup** in the Multiview Explorer.

5. Right-click the CJ1W-EIP21S and select **Read Special Unit Settings** from the menu.

   The Open File Dialog Box is displayed.



| Item name | Value |
|---|---|
| Device name | J01 |
| Model name | CJ1W-EIP21S |
| Product name | EtherNet/IP Unit for CJ |
| Version | 1.0 |
| Specifications | 100BASE-TX |
| Rack No. | 0 |
| Slot No. | 0 |
| Unit No. | 0 |
| Special Unit Settings | Settings<br>Edit Special Unit Setti |

6. Select the file that you saved in step 3, and then click the **Open** Button.

   The settings saved in the file are reflected in the CJ1W-EIP21S settings.

7. For items that are not available with the CJ1W-EIP21, but are available with the CJ1W-EIP21S, the default values are set. Change them as needed.

**Precautions for Correct Use**

- The FTP password setting is not reflected in the settings for the CJ1W-EIP21S due to the difference in the allowable setting range of the password length between the CJ1W-EIP21 and the CJ1W-EIP21S. Set a new password for the CJ1W-EIP21S. The allowable setting range of the FTP password length is as follows.
  CJ1W-EIP21: 1 to 8 single-byte alphanumeric characters
  CJ1W-EIP21S: 8 to 16 single-byte alphanumeric characters
- If the FTP port number specified on the CJ1W-EIP21 is 22 or 443, an error will occur in the Unit Setup for the CJ1W-EIP21S. Change the FTP port number for the CJ1W-EIP21S to other than 22 and 443.

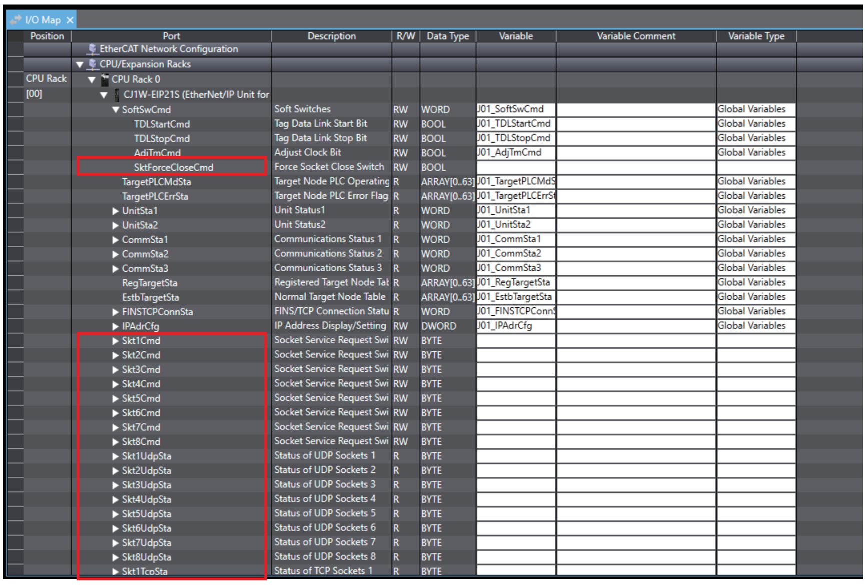
## Setting Device Variables

Set device variables for the CJ1W-EIP21S in the Unit configuration after replacement.

Using the Sysmac Studio, set device variables to match the device variables set on the source replacement model CJ1W-EIP21.

Create device variables for the I/O ports. The differences between the I/O ports on the CJ1W-EIP21 and those on the CJ1W-EIP21S are as follows.

- For the CJ1W-EIP21S, *SktForceCloseCmd* is added below *AdjTmCmd*.
- For the CJ1W-EIP21S, socket-related I/O ports are added below *IPAdrCfg*.

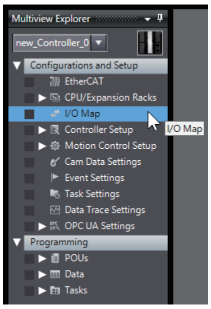


| Position | Port | Description | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|---|
| | EtherCAT Network Configuration | | | | | | |
| | ▼ CPU/Expansion Racks | | | | | | |
| CPU Rack | ▼ CPU Rack 0 | | | | | | |
| [00] | ▼ CJ1W-EIP21S (EtherNet/IP Unit for | | | | | | |
| | ▼ SoftSwCmd | Soft Switches | RW | WORD | J01_SoftSwCmd | | Global Variables |
| | TDLStartCmd | Tag Data Link Start Bit | RW | BOOL | J01_TDLStartCmd | | Global Variables |
| | TDLStopCmd | Tag Data Link Stop Bit | RW | BOOL | J01_TDLStopCmd | | Global Variables |
| | AdjTmCmd | Adjust Clock Bit | RW | BOOL | J01_AdjTmCmd | | Global Variables |
| | SktForceCloseCmd | Force Socket Close Switch | RW | BOOL | | | |
| | TargetPLCMdSta | Target Node PLC Operating | R | ARRAY[0..63] | J01_TargetPLCMdS | | Global Variables |
| | TargetPLCErrSta | Target Node PLC Error Flag | R | ARRAY[0..63] | J01_TargetPLCErrSt | | Global Variables |
| | ▶ UnitSta1 | Unit Status1 | R | WORD | J01_UnitSta1 | | Global Variables |
| | ▶ UnitSta2 | Unit Status2 | R | WORD | J01_UnitSta2 | | Global Variables |
| | ▶ CommSta1 | Communications Status 1 | R | WORD | J01_CommSta1 | | Global Variables |
| | ▶ CommSta2 | Communications Status 2 | R | WORD | J01_CommSta2 | | Global Variables |
| | ▶ CommSta3 | Communications Status 3 | R | WORD | J01_CommSta3 | | Global Variables |
| | RegTargetSta | Registered Target Node Tab | R | ARRAY[0..63] | J01_RegTargetSta | | Global Variables |
| | EstbTargetSta | Normal Target Node Table | R | ARRAY[0..63] | J01_EstbTargetSta | | Global Variables |
| | ▶ FINSTCPConnSta | FINS/TCP Connection Statu | R | WORD | J01_FINSTCPConnS | | Global Variables |
| | ▶ IPAdrCfg | IP Address Display/Setting | RW | DWORD | J01_IPAdrCfg | | Global Variables |
| | ▶ Skt1Cmd | Socket Service Request Swi | RW | BYTE | | | |
| | ▶ Skt2Cmd | Socket Service Request Swi | RW | BYTE | | | |
| | ▶ Skt3Cmd | Socket Service Request Swi | RW | BYTE | | | |
| | ▶ Skt4Cmd | Socket Service Request Swi | RW | BYTE | | | |
| | ▶ Skt5Cmd | Socket Service Request Swi | RW | BYTE | | | |
| | ▶ Skt6Cmd | Socket Service Request Swi | RW | BYTE | | | |
| | ▶ Skt7Cmd | Socket Service Request Swi | RW | BYTE | | | |
| | ▶ Skt8Cmd | Socket Service Request Swi | RW | BYTE | | | |
| | ▶ Skt1UdpSta | Status of UDP Sockets 1 | R | BYTE | | | |
| | ▶ Skt2UdpSta | Status of UDP Sockets 2 | R | BYTE | | | |
| | ▶ Skt3UdpSta | Status of UDP Sockets 3 | R | BYTE | | | |
| | ▶ Skt4UdpSta | Status of UDP Sockets 4 | R | BYTE | | | |
| | ▶ Skt5UdpSta | Status of UDP Sockets 5 | R | BYTE | | | |
| | ▶ Skt6UdpSta | Status of UDP Sockets 6 | R | BYTE | | | |
| | ▶ Skt7UdpSta | Status of UDP Sockets 7 | R | BYTE | | | |
| | ▶ Skt8UdpSta | Status of UDP Sockets 8 | R | BYTE | | | |
| | ▶ Skt1TcpSta | Status of TCP Sockets 1 | R | BYTE | | | |

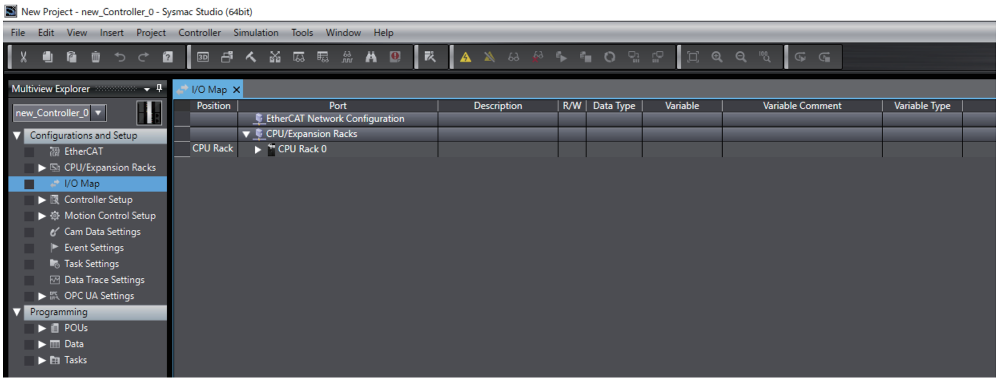
Use the following procedure to transfer the device variables on the CJ1W-EIP21 to the CJ1W-EIP21S.

Perform the procedure for all pairs of source and target replacement models, i.e., CJ1W-EIP21 and CJ1W-EIP21S.

**1** In the Sysmac Studio, open the project data from the source replacement model and double-click **I/O Map** under **Configurations and Setup** in the Multiview Explorer.

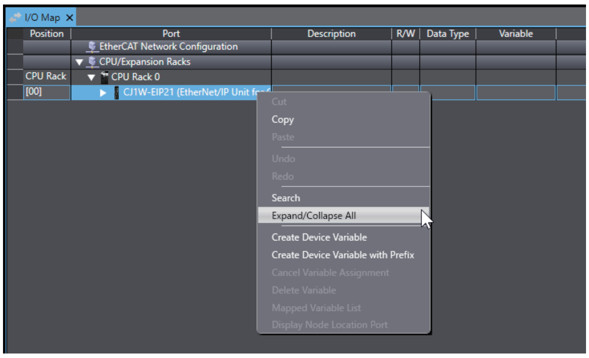


The I/O Map Tab Page is displayed in the Edit Pane.



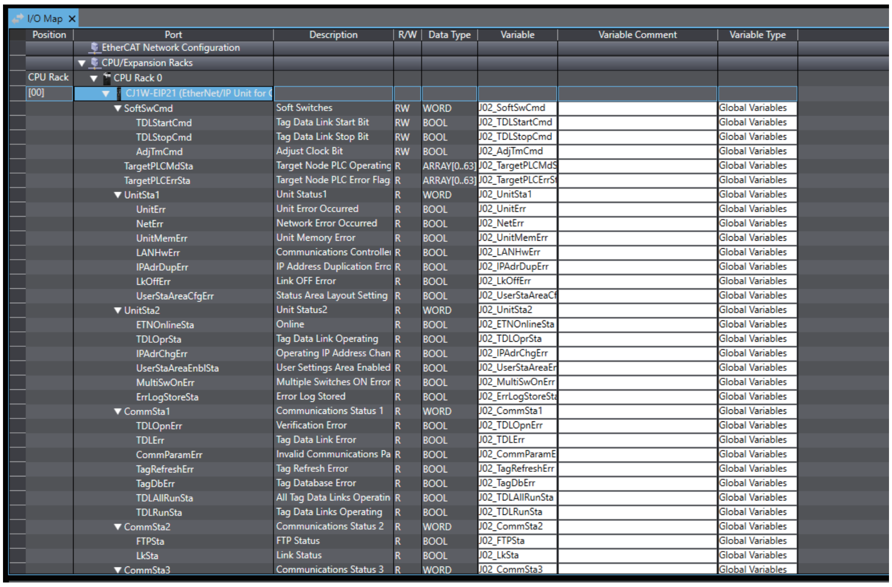
| Position | Port | Description | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|---|
| | EtherCAT Network Configuration | | | | | | |
| | CPU/Expansion Racks | | | | | | |
| CPU Rack | CPU Rack 0 | | | | | | |

**2** In the I/O Map Tab Page, right-click the source replacement model CJ1W-EIP21 and select **Expand/Collapse All**.



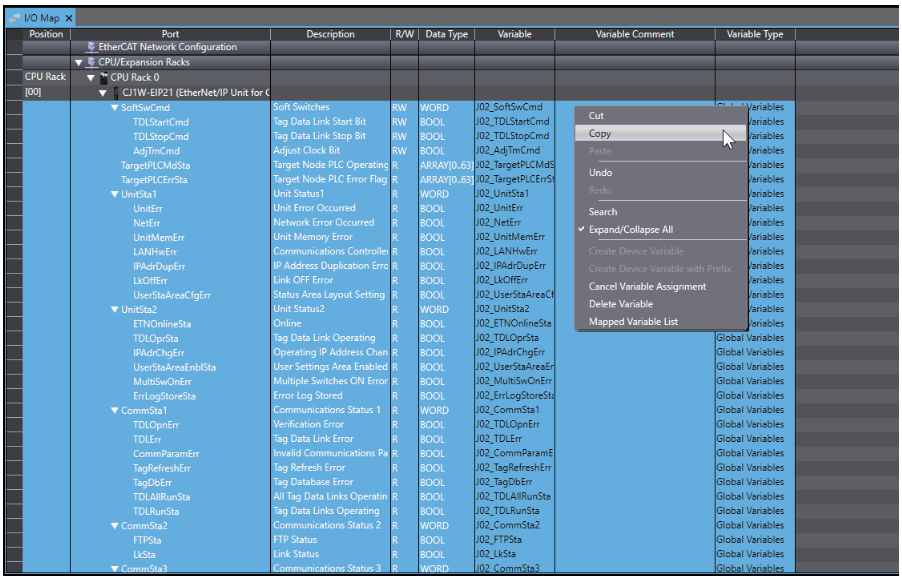
| Position | Port | Description | R/W | Data Type | Variable |
|---|---|---|---|---|---|
| | EtherCAT Network Configuration | | | | |
| | CPU/Expansion Racks | | | | |
| CPU Rack | CPU Rack 0 | | | | |
| [00] | CJ1W-EIP21 (EtherNet/IP Unit fo | | | | |

The I/O ports in the source replacement model CJ1W-EIP21 are expanded.



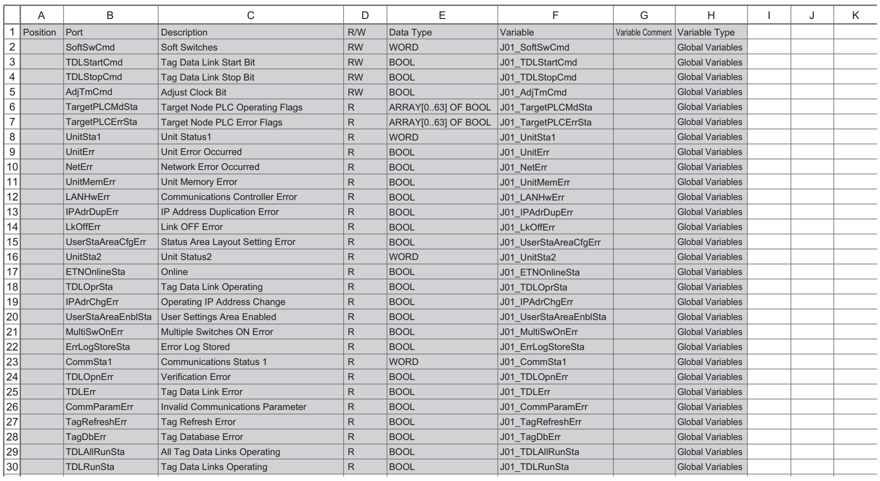
| Position | Port | Description | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|---|
| | EtherCAT Network Configuration | | | | | | |
| | ▼ CPU/Expansion Racks | | | | | | |
| CPU Rack | ▼ CPU Rack 0 | | | | | | |
| [00] | ▼ CJ1W-EIP21 (EtherNet/IP Unit for C | | | | | | |
| | ▼ SoftSwCmd | Soft Switches | RW | WORD | J02_SoftSwCmd | | Global Variables |
| | TDLStartCmd | Tag Data Link Start Bit | RW | BOOL | J02_TDLStartCmd | | Global Variables |
| | TDLStopCmd | Tag Data Link Stop Bit | RW | BOOL | J02_TDLStopCmd | | Global Variables |
| | AdjTmCmd | Adjust Clock Bit | RW | BOOL | J02_AdjTmCmd | | Global Variables |
| | TargetPLCMdSta | Target Node PLC Operating | R | ARRAY[0..63] | J02_TargetPLCMdS | | Global Variables |
| | TargetPLCErrSta | Target Node PLC Error Flag | R | ARRAY[0..63] | J02_TargetPLCErrSt | | Global Variables |
| | ▼ UnitSta1 | Unit Status1 | R | WORD | J02_UnitSta1 | | Global Variables |
| | UnitErr | Unit Error Occurred | R | BOOL | J02_UnitErr | | Global Variables |
| | NetErr | Network Error Occurred | R | BOOL | J02_NetErr | | Global Variables |
| | UnitMemErr | Unit Memory Error | R | BOOL | J02_UnitMemErr | | Global Variables |
| | LANHwErr | Communications Controlle | R | BOOL | J02_LANHwErr | | Global Variables |
| | IPAdrDupErr | IP Address Duplication Erro | R | BOOL | J02_IPAdrDupErr | | Global Variables |
| | LkOffErr | Link OFF Error | R | BOOL | J02_LkOffErr | | Global Variables |
| | UserStaAreaCfgErr | Status Area Layout Setting | R | BOOL | J02_UserStaAreaCf | | Global Variables |
| | ▼ UnitSta2 | Unit Status2 | R | WORD | J02_UnitSta2 | | Global Variables |
| | ETNOnlineSta | Online | R | BOOL | J02_ETNOnlineSta | | Global Variables |
| | TDLOprSta | Tag Data Link Operating | R | BOOL | J02_TDLOprSta | | Global Variables |
| | IPAdrChgErr | Operating IP Address Chan | R | BOOL | J02_IPAdrChgErr | | Global Variables |
| | UserStaAreaEnblSta | User Settings Area Enabled | R | BOOL | J02_UserStaAreaEr | | Global Variables |
| | MultiSwOnErr | Multiple Switches ON Error | R | BOOL | J02_MultiSwOnErr | | Global Variables |
| | ErrLogStoreSta | Error Log Stored | R | BOOL | J02_ErrLogStoreSta | | Global Variables |
| | ▼ CommSta1 | Communications Status 1 | R | WORD | J02_CommSta1 | | Global Variables |
| | TDLOpnErr | Verification Error | R | BOOL | J02_TDLOpnErr | | Global Variables |
| | TDLErr | Tag Data Link Error | R | BOOL | J02_TDLErr | | Global Variables |
| | CommParamErr | Invalid Communications Pa | R | BOOL | J02_CommParamE | | Global Variables |
| | TagRefreshErr | Tag Refresh Error | R | BOOL | J02_TagRefreshErr | | Global Variables |
| | TagDbErr | Tag Database Error | R | BOOL | J02_TagDbErr | | Global Variables |
| | TDLAllRunSta | All Tag Data Links Operatin | R | BOOL | J02_TDLAllRunSta | | Global Variables |
| | TDLRunSta | Tag Data Links Operating | R | BOOL | J02_TDLRunSta | | Global Variables |
| | ▼ CommSta2 | Communications Status 2 | R | WORD | J02_CommSta2 | | Global Variables |
| | FTPSta | FTP Status | R | BOOL | J02_FTPSta | | Global Variables |
| | LkSta | Link Status | R | BOOL | J02_LkSta | | Global Variables |
| | ▼ CommSta3 | Communications Status 3 | R | WORD | J02_CommSta3 | | Global Variables |

**3** Select all I/O ports listed under the source replacement model CJ1W-EIP21 and copy them.



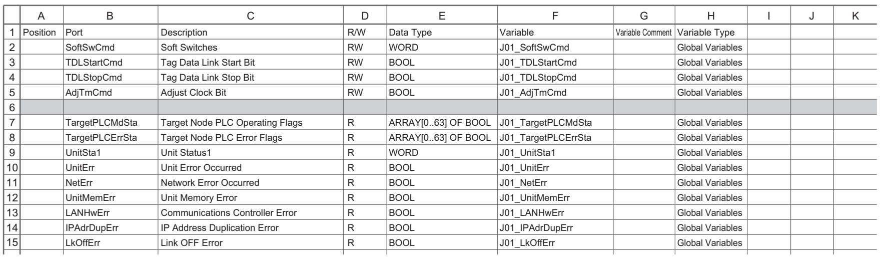
| Position | Port | Description | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|---|
| | EtherCAT Network Configuration | | | | | | |
| | ▼ CPU/Expansion Racks | | | | | | |
| CPU Rack | ▼ CPU Rack 0 | | | | | | |
| [00] | ▼ CJ1W-EIP21 (EtherNet/IP Unit for C | | | | | | |
| | ▼ SoftSwCmd | Soft Switches | RW | WORD | J02_SoftSwCmd | | Variables |
| | TDLStartCmd | Tag Data Link Start Bit | RW | BOOL | J02_TDLStartCmd | | Variables |
| | TDLStopCmd | Tag Data Link Stop Bit | RW | BOOL | J02_TDLStopCmd | | Variables |
| | AdjTmCmd | Adjust Clock Bit | RW | BOOL | J02_AdjTmCmd | | Variables |
| | TargetPLCMdSta | Target Node PLC Operating | R | ARRAY[0..63] | J02_TargetPLCMdS | | Variables |
| | TargetPLCErrSta | Target Node PLC Error Flag | R | ARRAY[0..63] | J02_TargetPLCErrSt | | Variables |
| | ▼ UnitSta1 | Unit Status1 | R | WORD | J02_UnitSta1 | | Variables |
| | UnitErr | Unit Error Occurred | R | BOOL | J02_UnitErr | | Variables |
| | NetErr | Network Error Occurred | R | BOOL | J02_NetErr | | Variables |
| | UnitMemErr | Unit Memory Error | R | BOOL | J02_UnitMemErr | | Variables |
| | LANHwErr | Communications Controlle | R | BOOL | J02_LANHwErr | | Variables |
| | IPAdrDupErr | IP Address Duplication Erro | R | BOOL | J02_IPAdrDupErr | | Variables |
| | LkOffErr | Link OFF Error | R | BOOL | J02_LkOffErr | | Variables |
| | UserStaAreaCfgErr | Status Area Layout Setting | R | BOOL | J02_UserStaAreaCf | | Variables |
| | ▼ UnitSta2 | Unit Status2 | R | WORD | J02_UnitSta2 | | Variables |
| | ETNOnlineSta | Online | R | BOOL | J02_ETNOnlineSta | | Variables |
| | TDLOprSta | Tag Data Link Operating | R | BOOL | J02_TDLOprSta | | Global Variables |
| | IPAdrChgErr | Operating IP Address Chan | R | BOOL | J02_IPAdrChgErr | | Global Variables |
| | UserStaAreaEnblSta | User Settings Area Enabled | R | BOOL | J02_UserStaAreaEr | | Global Variables |
| | MultiSwOnErr | Multiple Switches ON Error | R | BOOL | J02_MultiSwOnErr | | Global Variables |
| | ErrLogStoreSta | Error Log Stored | R | BOOL | J02_ErrLogStoreSta | | Global Variables |
| | ▼ CommSta1 | Communications Status 1 | R | WORD | J02_CommSta1 | | Global Variables |
| | TDLOpnErr | Verification Error | R | BOOL | J02_TDLOpnErr | | Global Variables |
| | TDLErr | Tag Data Link Error | R | BOOL | J02_TDLErr | | Global Variables |
| | CommParamErr | Invalid Communications Pa | R | BOOL | J02_CommParamE | | Global Variables |
| | TagRefreshErr | Tag Refresh Error | R | BOOL | J02_TagRefreshErr | | Global Variables |
| | TagDbErr | Tag Database Error | R | BOOL | J02_TagDbErr | | Global Variables |
| | TDLAllRunSta | All Tag Data Links Operatin | R | BOOL | J02_TDLAllRunSta | | Global Variables |
| | TDLRunSta | Tag Data Links Operating | R | BOOL | J02_TDLRunSta | | Global Variables |
| | ▼ CommSta2 | Communications Status 2 | R | WORD | J02_CommSta2 | | Global Variables |
| | FTPSta | FTP Status | R | BOOL | J02_FTPSta | | Global Variables |
| | LkSta | Link Status | R | BOOL | J02_LkSta | | Global Variables |
| | ▼ CommSta3 | Communications Status 3 | R | WORD | J02_CommSta3 | | Global Variables |

**4** Paste the copied data into a spreadsheet software.



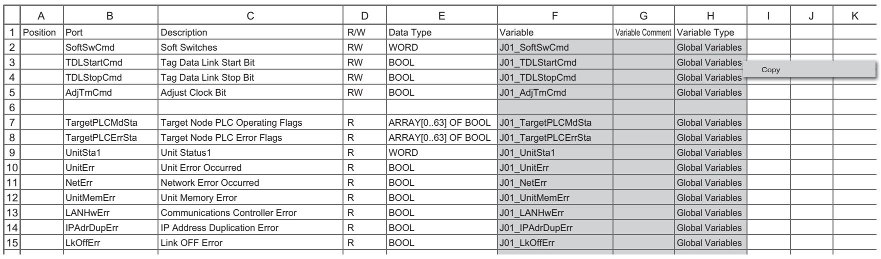
| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Position | Port | Description | R/W | Data Type | Variable | Variable Comment | Variable Type | | | |
| 2 | | SoftSwCmd | Soft Switches | RW | WORD | J01_SoftSwCmd | | Global Variables | | | |
| 3 | | TDLStartCmd | Tag Data Link Start Bit | RW | BOOL | J01_TDLStartCmd | | Global Variables | | | |
| 4 | | TDLStopCmd | Tag Data Link Stop Bit | RW | BOOL | J01_TDLStopCmd | | Global Variables | | | |
| 5 | | AdjTmCmd | Adjust Clock Bit | RW | BOOL | J01_AdjTmCmd | | Global Variables | | | |
| 6 | | TargetPLCMdSta | Target Node PLC Operating Flags | R | ARRAY[0..63] OF BOOL | J01_TargetPLCMdSta | | Global Variables | | | |
| 7 | | TargetPLCErrSta | Target Node PLC Error Flags | R | ARRAY[0..63] OF BOOL | J01_TargetPLCErrSta | | Global Variables | | | |
| 8 | | UnitSta1 | Unit Status1 | R | WORD | J01_UnitSta1 | | Global Variables | | | |
| 9 | | UnitErr | Unit Error Occurred | R | BOOL | J01_UnitErr | | Global Variables | | | |
| 10 | | NetErr | Network Error Occurred | R | BOOL | J01_NetErr | | Global Variables | | | |
| 11 | | UnitMemErr | Unit Memory Error | R | BOOL | J01_UnitMemErr | | Global Variables | | | |
| 12 | | LANHwErr | Communications Controller Error | R | BOOL | J01_LANHwErr | | Global Variables | | | |
| 13 | | IPAdrDupErr | IP Address Duplication Error | R | BOOL | J01_IPAdrDupErr | | Global Variables | | | |
| 14 | | LkOffErr | Link OFF Error | R | BOOL | J01_LkOffErr | | Global Variables | | | |
| 15 | | UserStaAreaCfgErr | Status Area Layout Setting Error | R | BOOL | J01_UserStaAreaCfgErr | | Global Variables | | | |
| 16 | | UnitSta2 | Unit Status2 | R | WORD | J01_UnitSta2 | | Global Variables | | | |
| 17 | | ETNOnlineSta | Online | R | BOOL | J01_ETNOnlineSta | | Global Variables | | | |
| 18 | | TDLOprSta | Tag Data Link Operating | R | BOOL | J01_TDLOprSta | | Global Variables | | | |
| 19 | | IPAdrChgErr | Operating IP Address Change | R | BOOL | J01_IPAdrChgErr | | Global Variables | | | |
| 20 | | UserStaAreaEnblSta | User Settings Area Enabled | R | BOOL | J01_UserStaAreaEnblSta | | Global Variables | | | |
| 21 | | MultiSwOnErr | Multiple Switches ON Error | R | BOOL | J01_MultiSwOnErr | | Global Variables | | | |
| 22 | | ErrLogStoreSta | Error Log Stored | R | BOOL | J01_ErrLogStoreSta | | Global Variables | | | |
| 23 | | CommSta1 | Communications Status 1 | R | WORD | J01_CommSta1 | | Global Variables | | | |
| 24 | | TDLOpnErr | Verification Error | R | BOOL | J01_TDLOpnErr | | Global Variables | | | |
| 25 | | TDLErr | Tag Data Link Error | R | BOOL | J01_TDLErr | | Global Variables | | | |
| 26 | | CommParamErr | Invalid Communications Parameter | R | BOOL | J01_CommParamErr | | Global Variables | | | |
| 27 | | TagRefreshErr | Tag Refresh Error | R | BOOL | J01_TagRefreshErr | | Global Variables | | | |
| 28 | | TagDbErr | Tag Database Error | R | BOOL | J01_TagDbErr | | Global Variables | | | |
| 29 | | TDLAllRunSta | All Tag Data Links Operating | R | BOOL | J01_TDLAllRunSta | | Global Variables | | | |
| 30 | | TDLRunSta | Tag Data Links Operating | R | BOOL | J01_TDLRunSta | | Global Variables | | | |

**5** In the spreadsheet software, add a blank line between *AdjTmCmd* and *TargetPLCMdSta*.



| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Position | Port | Description | R/W | Data Type | Variable | Variable Comment | Variable Type | | | |
| 2 | | SoftSwCmd | Soft Switches | RW | WORD | J01_SoftSwCmd | | Global Variables | | | |
| 3 | | TDLStartCmd | Tag Data Link Start Bit | RW | BOOL | J01_TDLStartCmd | | Global Variables | | | |
| 4 | | TDLStopCmd | Tag Data Link Stop Bit | RW | BOOL | J01_TDLStopCmd | | Global Variables | | | |
| 5 | | AdjTmCmd | Adjust Clock Bit | RW | BOOL | J01_AdjTmCmd | | Global Variables | | | |
| 6 | | | | | | | | | | | |
| 7 | | TargetPLCMdSta | Target Node PLC Operating Flags | R | ARRAY[0..63] OF BOOL | J01_TargetPLCMdSta | | Global Variables | | | |
| 8 | | TargetPLCErrSta | Target Node PLC Error Flags | R | ARRAY[0..63] OF BOOL | J01_TargetPLCErrSta | | Global Variables | | | |
| 9 | | UnitSta1 | Unit Status1 | R | WORD | J01_UnitSta1 | | Global Variables | | | |
| 10 | | UnitErr | Unit Error Occurred | R | BOOL | J01_UnitErr | | Global Variables | | | |
| 11 | | NetErr | Network Error Occurred | R | BOOL | J01_NetErr | | Global Variables | | | |
| 12 | | UnitMemErr | Unit Memory Error | R | BOOL | J01_UnitMemErr | | Global Variables | | | |
| 13 | | LANHwErr | Communications Controller Error | R | BOOL | J01_LANHwErr | | Global Variables | | | |
| 14 | | IPAdrDupErr | IP Address Duplication Error | R | BOOL | J01_IPAdrDupErr | | Global Variables | | | |
| 15 | | LkOffErr | Link OFF Error | R | BOOL | J01_LkOffErr | | Global Variables | | | |

**6** In the spreadsheet software, copy all the data in the Variable, Variable Comment, and Variable Type columns.

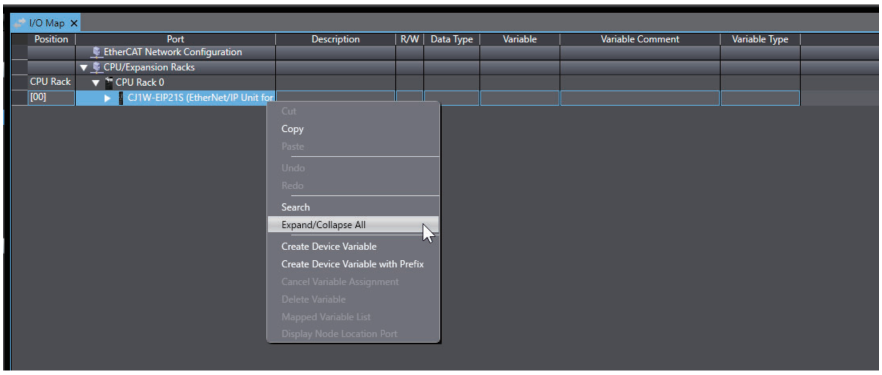


| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Position | Port | Description | R/W | Data Type | Variable | Variable Comment | Variable Type | | | |
| 2 | | SoftSwCmd | Soft Switches | RW | WORD | J01_SoftSwCmd | | Global Variables | | | |
| 3 | | TDLStartCmd | Tag Data Link Start Bit | RW | BOOL | J01_TDLStartCmd | | Global Variables | | | |
| 4 | | TDLStopCmd | Tag Data Link Stop Bit | RW | BOOL | J01_TDLStopCmd | | Global Variables | | | |
| 5 | | AdjTmCmd | Adjust Clock Bit | RW | BOOL | J01_AdjTmCmd | | Global Variables | | | |
| 6 | | | | | | | | | | | |
| 7 | | TargetPLCMdSta | Target Node PLC Operating Flags | R | ARRAY[0..63] OF BOOL | J01_TargetPLCMdSta | | Global Variables | | | |
| 8 | | TargetPLCErrSta | Target Node PLC Error Flags | R | ARRAY[0..63] OF BOOL | J01_TargetPLCErrSta | | Global Variables | | | |
| 9 | | UnitSta1 | Unit Status1 | R | WORD | J01_UnitSta1 | | Global Variables | | | |
| 10 | | UnitErr | Unit Error Occurred | R | BOOL | J01_UnitErr | | Global Variables | | | |
| 11 | | NetErr | Network Error Occurred | R | BOOL | J01_NetErr | | Global Variables | | | |
| 12 | | UnitMemErr | Unit Memory Error | R | BOOL | J01_UnitMemErr | | Global Variables | | | |
| 13 | | LANHwErr | Communications Controller Error | R | BOOL | J01_LANHwErr | | Global Variables | | | |
| 14 | | IPAdrDupErr | IP Address Duplication Error | R | BOOL | J01_IPAdrDupErr | | Global Variables | | | |
| 15 | | LkOffErr | Link OFF Error | R | BOOL | J01_LkOffErr | | Global Variables | | | |

**7** In the Sysmac Studio, open the project data in the target replacement model and double-click **I/O Map** under **Configurations and Setup** in the Multiview Explorer.

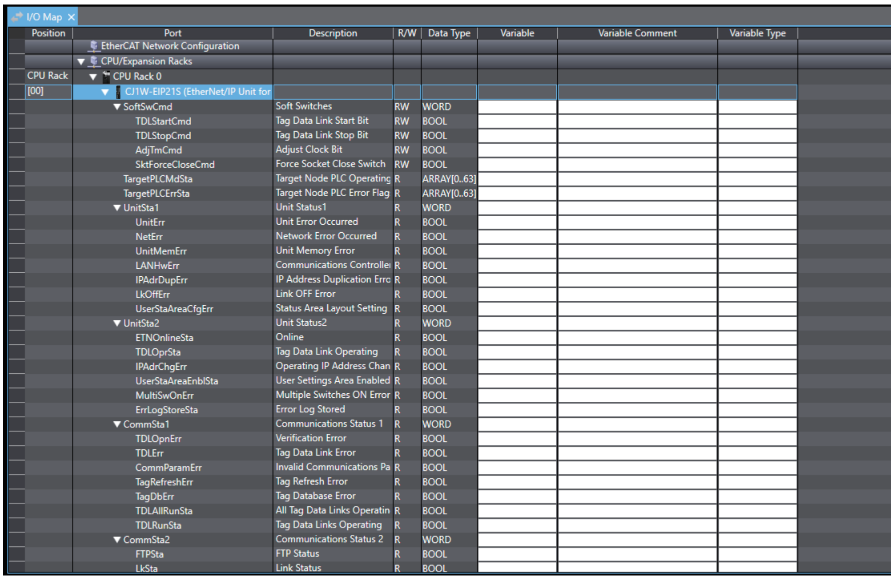
The I/O Map Tab Page is displayed in the Edit Pane.

**8** In the I/O Map Tab Page, right-click the target replacement model CJ1W-EIP21S and select **Expand/Collapse All**.



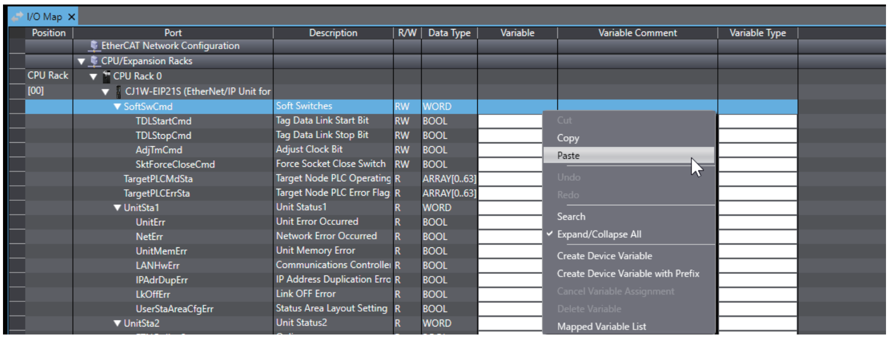
| Position | Port | Description | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|---|
| | EtherCAT Network Configuration | | | | | | |
| | CPU/Expansion Racks | | | | | | |
| CPU Rack | CPU Rack 0 | | | | | | |
| [00] | CJ1W-EIP21S (EtherNet/IP Unit for | | | | | | |

The I/O ports in the target replacement model CJ1W-EIP21S are expanded.



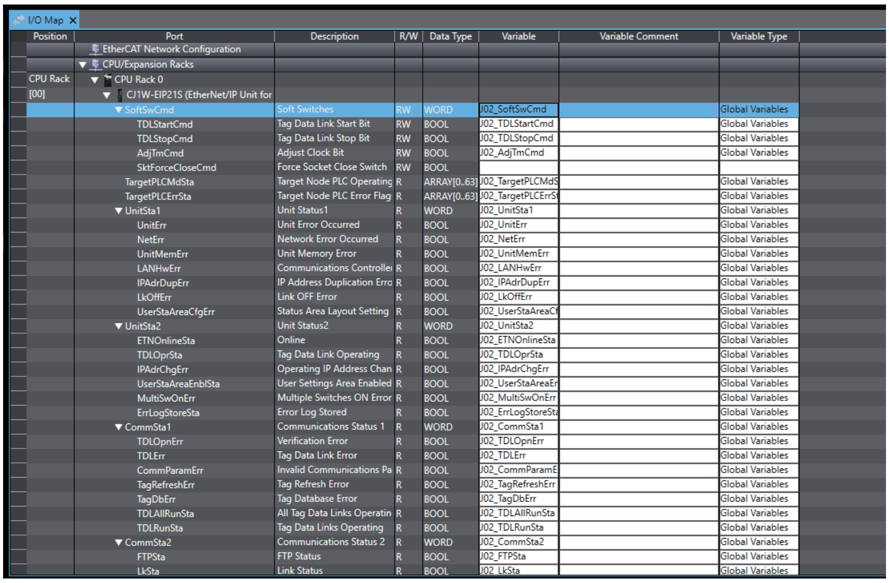
| Position | Port | Description | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|---|
| | EtherCAT Network Configuration | | | | | | |
| | ▼ CPU/Expansion Racks | | | | | | |
| CPU Rack | ▼ CPU Rack 0 | | | | | | |
| [00] | ▼ CJ1W-EIP21S (EtherNet/IP Unit for | | | | | | |
| | ▼ SoftSwCmd | Soft Switches | RW | WORD | | | |
| | TDLStartCmd | Tag Data Link Start Bit | RW | BOOL | | | |
| | TDLStopCmd | Tag Data Link Stop Bit | RW | BOOL | | | |
| | AdjTmCmd | Adjust Clock Bit | RW | BOOL | | | |
| | SktForceCloseCmd | Force Socket Close Switch | RW | BOOL | | | |
| | TargetPLCMdSta | Target Node PLC Operating | R | ARRAY[0..63] | | | |
| | TargetPLCErrSta | Target Node PLC Error Flag | R | ARRAY[0..63] | | | |
| | ▼ UnitSta1 | Unit Status1 | R | WORD | | | |
| | UnitErr | Unit Error Occurred | R | BOOL | | | |
| | NetErr | Network Error Occurred | R | BOOL | | | |
| | UnitMemErr | Unit Memory Error | R | BOOL | | | |
| | LANHwErr | Communications Controlle | R | BOOL | | | |
| | IPAdrDupErr | IP Address Duplication Erro | R | BOOL | | | |
| | LkOffErr | Link OFF Error | R | BOOL | | | |
| | UserStaAreaCfgErr | Status Area Layout Setting | R | BOOL | | | |
| | ▼ UnitSta2 | Unit Status2 | R | WORD | | | |
| | ETNOnlineSta | Online | R | BOOL | | | |
| | TDLOprSta | Tag Data Link Operating | R | BOOL | | | |
| | IPAdrChgErr | Operating IP Address Chan | R | BOOL | | | |
| | UserStaAreaEnblSta | User Settings Area Enabled | R | BOOL | | | |
| | MultiSwOnErr | Multiple Switches ON Error | R | BOOL | | | |
| | ErrLogStoreSta | Error Log Stored | R | BOOL | | | |
| | ▼ CommSta1 | Communications Status 1 | R | WORD | | | |
| | TDLOpnErr | Verification Error | R | BOOL | | | |
| | TDLErr | Tag Data Link Error | R | BOOL | | | |
| | CommParamErr | Invalid Communications Pa | R | BOOL | | | |
| | TagRefreshErr | Tag Refresh Error | R | BOOL | | | |
| | TagDbErr | Tag Database Error | R | BOOL | | | |
| | TDLAllRunSta | All Tag Data Links Operatin | R | BOOL | | | |
| | TDLRunSta | Tag Data Links Operating | R | BOOL | | | |
| | ▼ CommSta2 | Communications Status 2 | R | WORD | | | |
| | FTPSta | FTP Status | R | BOOL | | | |
| | LkSta | Link Status | R | BOOL | | | |

**9** Select the first row listed under the target replacement model CJ1W-EIP21S I/O port and paste the data that you copied from the spreadsheet software.



| Position | Port | Description | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|---|
| | EtherCAT Network Configuration | | | | | | |
| | ▼ CPU/Expansion Racks | | | | | | |
| CPU Rack | ▼ CPU Rack 0 | | | | | | |
| [00] | ▼ CJ1W-EIP21S (EtherNet/IP Unit for | | | | | | |
| | ▼ SoftSwCmd | Soft Switches | RW | WORD | | | |
| | TDLStartCmd | Tag Data Link Start Bit | RW | BOOL | | | |
| | TDLStopCmd | Tag Data Link Stop Bit | RW | BOOL | | | |
| | AdjTmCmd | Adjust Clock Bit | RW | BOOL | | | |
| | SktForceCloseCmd | Force Socket Close Switch | RW | BOOL | | | |
| | TargetPLCMdSta | Target Node PLC Operating | R | ARRAY[0..63] | | | |
| | TargetPLCErrSta | Target Node PLC Error Flag | R | ARRAY[0..63] | | | |
| | ▼ UnitSta1 | Unit Status1 | R | WORD | | | |
| | UnitErr | Unit Error Occurred | R | BOOL | | | |
| | NetErr | Network Error Occurred | R | BOOL | | | |
| | UnitMemErr | Unit Memory Error | R | BOOL | | | |
| | LANHwErr | Communications Controlle | R | BOOL | | | |
| | IPAdrDupErr | IP Address Duplication Erro | R | BOOL | | | |
| | LkOffErr | Link OFF Error | R | BOOL | | | |
| | UserStaAreaCfgErr | Status Area Layout Setting | R | BOOL | | | |
| | ▼ UnitSta2 | Unit Status2 | R | WORD | | | |

The variables, variable comments, and variable types are set for the target replacement model CJ1W-EIP21S.



| Position | Port | Description | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|---|
| | EtherCAT Network Configuration | | | | | | |
| | ▼ CPU/Expansion Racks | | | | | | |
| CPU Rack | ▼ CPU Rack 0 | | | | | | |
| [00] | ▼ CJ1W-EIP21S (EtherNet/IP Unit for | | | | | | |
| | ▼ SoftSwCmd | Soft Switches | RW | WORD | J02_SoftSwCmd | | Global Variables |
| | TDLStartCmd | Tag Data Link Start Bit | RW | BOOL | J02_TDLStartCmd | | Global Variables |
| | TDLStopCmd | Tag Data Link Stop Bit | RW | BOOL | J02_TDLStopCmd | | Global Variables |
| | AdjTmCmd | Adjust Clock Bit | RW | BOOL | J02_AdjTmCmd | | Global Variables |
| | SktForceCloseCmd | Force Socket Close Switch | RW | BOOL | | | |
| | TargetPLCMdSta | Target Node PLC Operating | R | ARRAY[0..63] | J02_TargetPLCMdS | | Global Variables |
| | TargetPLCErrSta | Target Node PLC Error Flag | R | ARRAY[0..63] | J02_TargetPLCErrS | | Global Variables |
| | ▼ UnitSta1 | Unit Status1 | R | WORD | J02_UnitSta1 | | Global Variables |
| | UnitErr | Unit Error Occurred | R | BOOL | J02_UnitErr | | Global Variables |
| | NetErr | Network Error Occurred | R | BOOL | J02_NetErr | | Global Variables |
| | UnitMemErr | Unit Memory Error | R | BOOL | J02_UnitMemErr | | Global Variables |
| | LANHwErr | Communications Controlle | R | BOOL | J02_LANHwErr | | Global Variables |
| | IPAdrDupErr | IP Address Duplication Erro | R | BOOL | J02_IPAdrDupErr | | Global Variables |
| | LkOffErr | Link OFF Error | R | BOOL | J02_LkOffErr | | Global Variables |
| | UserStaAreaCfgErr | Status Area Layout Setting | R | BOOL | J02_UserStaAreaCf | | Global Variables |
| | ▼ UnitSta2 | Unit Status2 | R | WORD | J02_UnitSta2 | | Global Variables |
| | ETNOnlineSta | Online | R | BOOL | J02_ETNOnlineSta | | Global Variables |
| | TDLOprSta | Tag Data Link Operating | R | BOOL | J02_TDLOprSta | | Global Variables |
| | IPAdrChgErr | Operating IP Address Chan | R | BOOL | J02_IPAdrChgErr | | Global Variables |
| | UserStaAreaEnblSta | User Settings Area Enabled | R | BOOL | J02_UserStaAreaEr | | Global Variables |
| | MultiSwOnErr | Multiple Switches ON Error | R | BOOL | J02_MultiSwOnErr | | Global Variables |
| | ErrLogStoreSta | Error Log Stored | R | BOOL | J02_ErrLogStoreSta | | Global Variables |
| | ▼ CommSta1 | Communications Status 1 | R | WORD | J02_CommSta1 | | Global Variables |
| | TDLOpnErr | Verification Error | R | BOOL | J02_TDLOpnErr | | Global Variables |
| | TDLErr | Tag Data Link Error | R | BOOL | J02_TDLErr | | Global Variables |
| | CommParamErr | Invalid Communications Pa | R | BOOL | J02_CommParamE | | Global Variables |
| | TagRefreshErr | Tag Refresh Error | R | BOOL | J02_TagRefreshErr | | Global Variables |
| | TagDbErr | Tag Database Error | R | BOOL | J02_TagDbErr | | Global Variables |
| | TDLAllRunSta | All Tag Data Links Operatin | R | BOOL | J02_TDLAllRunSta | | Global Variables |
| | TDLRunSta | Tag Data Links Operating | R | BOOL | J02_TDLRunSta | | Global Variables |
| | ▼ CommSta2 | Communications Status 2 | R | WORD | J02_CommSta2 | | Global Variables |
| | FTPSta | FTP Status | R | BOOL | J02_FTPSta | | Global Variables |
| | LkSta | Link Status | R | BOOL | J02_LkSta | | Global Variables |

A-8 Replacing a System Using the CJ1W-EIP21 with a System Using the CJ1W-EIP21S

App

A-8-3 Preparation

### Correcting the User Program

Correct the user program to deal with the differences in specifications due to the replacement and the restrictions that apply. Refer to *A-8-1 Differences in Specifications, Functions, Etc.* on page 105 for differences that affect the user program among those from the source replacement model.

After you correct the user program, save the Controller project data.

### Setting Tag Data Links

Configure tag data links to match the Unit configuration after replacement. There is no need to reconfigure them from scratch. Use the Network Configurator and execute the Change Device Type function on the network configuration file for the source replacement model CJ1W-EIP21 to change the source replacement model to the CJ1W-EIP21S.

## A-8-4 Replacement

This section describes the actual replacement steps.

### Replacing and Wiring Units

Replace and wire Units. Refer to the user's manual for the target replacement Unit for the hardware setup, installation, and wiring of each Unit.

Taking into account the risk that you fail in replacement, back up the data.

#### ● Replacing Units

Replace each Unit with the power supply turned OFF.

Configure the hardware settings of the CJ1W-EIP21S, i.e., the unit number and node address (default IP address), to match those of the source replacement model CJ1W-EIP21.

After this, replace the Unit.

If you have Units to replace other than the CJ1W-EIP21S, replace them.

#### ● Wiring the Units

After replacement, wire the Units.

However, leave the Ethernet cable to the CJ1W-EIP21S disconnected at this point.

### Turning ON the Power Supply

Turn ON the power supply to the Controller after replacement.

**Precautions for Correct Use**

The CJ1W-EIP21S requires a Unit startup time. For this reason, the CPU Unit startup time will be delayed by a few seconds compared with the system before replacement.

### Transferring Data

From the Support Software connected directly to the CPU Unit, transfer the following data that you prepared to the actual Controller after replacement.

- Controller project data
- Tag data link parameters

**Precautions for Correct Use**

You cannot place the Sysmac Studio online via the EtherNet/IP port of the CJ1W-EIP21S.

The table below shows the flow of data transfer to the actual Controller. Use this flow to transfer data.

| Step | Item | Description | Reference |
|---|---|---|---|
| 1 | Downloading the Controller Project Data | Transfer project data to the CPU Unit for each Controller after replacement. | *Downloading the Controller Project Data* on page 119 |
| 2 | Connecting Communications Cables | Turn OFF the power supply to the Controller. Then, connect communications cables to the target replacement model CJ1W-EIP21S. | --- |
| 3 | Turning ON the Power Supply to the Controller | Check the safety of the equipment and turn ON the power supply to each Controller. | --- |
| 4 | Downloading Tag Data Link Parameters | Download all tag data link parameters for each EtherNet/IP network. | *Downloading Tag Data Link Parameters* on page 119 |

### ● Downloading the Controller Project Data

Transfer the Controller project data that you created to match the Unit configuration after replacement during preparation to the actual Controller. Connect the Sysmac Studio directly to the CPU Unit, place it online, and transfer the data to the Controller.

**Precautions for Correct Use**

Note the following when you transfer the data from an EtherNet/IP Unit.

In a transfer using **Controller – Transfer** from the menu, data in Special Units is not transferred by default. Clear the **Do not transfer the following. (All items are not transferred.)** Check Box in the transfer options before you transfer the data.

Or, from the CPU and Expansion Racks Tab Page, open the tab page for the EtherNet/IP Unit and use the **Transfer to Controller** Button to transfer the data.

Refer to the *Sysmac Studio Version 1 Operation Manual* (Cat. No. W504) for details of how to transfer the data.

### ● Downloading Tag Data Link Parameters

This data is available when you use tag data links in the source replacement model. Using the Network Configurator, download the tag data link settings that you created to match the Unit configuration after replacement during preparation to the actual Controller.

## Checking the Operation

After you transfer data, use the Sysmac Studio to check that the system is operating as intended. Check also that no error occurs according to the specified error check procedure for each Unit.

Refer to the user’s manual for your Unit and the *NJ/NX-series Troubleshooting Manual* (Cat. No. W503) for the operation check and troubleshooting procedures for each Unit.

## Implementing Security Measures

If necessary, implement measures to reduce the security risks from external attacks before the start of system operation after replacement.

The CJ1W-EIP21S as well as the Sysmac Studio provide security functions to reduce security risks. Use the security functions in conjunction with system operation to reduce the security risks.

For details, refer to *Section 12 Security Functions* on page 12-1.

## Checking the Operation after Implementing Security Measures

After you implement security measures, check to be sure that the system operates according to the security settings.

# A-9 Sample Programming for Socket Services

You can use socket services with the CJ1W-EIP21S. This section describes sample programming in four cases.

For information common to all cases, refer to *A-9-1 System Configuration* and *A-9-2 Required Settings for Sample Programing*.

For sample programming for each of the four cases, refer to the following information.

*A-9-3 Sample Programing for UDP Communications by Manipulating Device Variables*

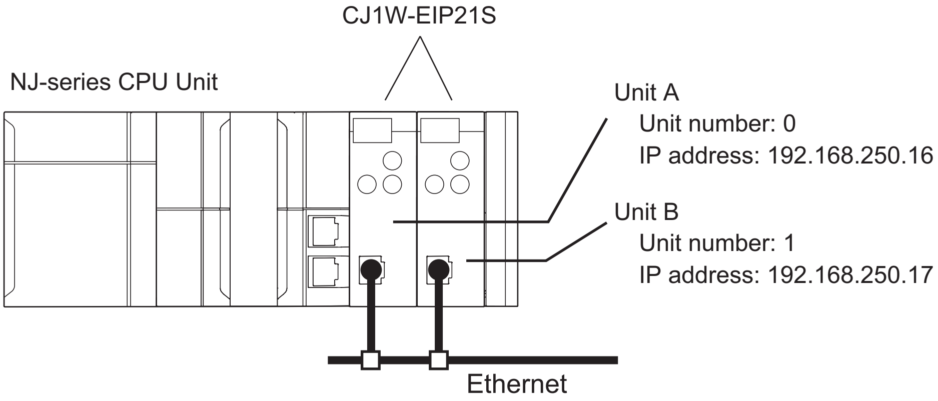
*A-9-4 Sample Programing for TCP Communications by Manipulating Device Variables*

*A-9-5 Sample Programing for UDP Communications by SendCmd Instruction*

*A-9-6 Sample Programing for TCP Communications by SendCmd Instruction*

## A-9-1 System Configuration

The programming examples assume the following system configuration.



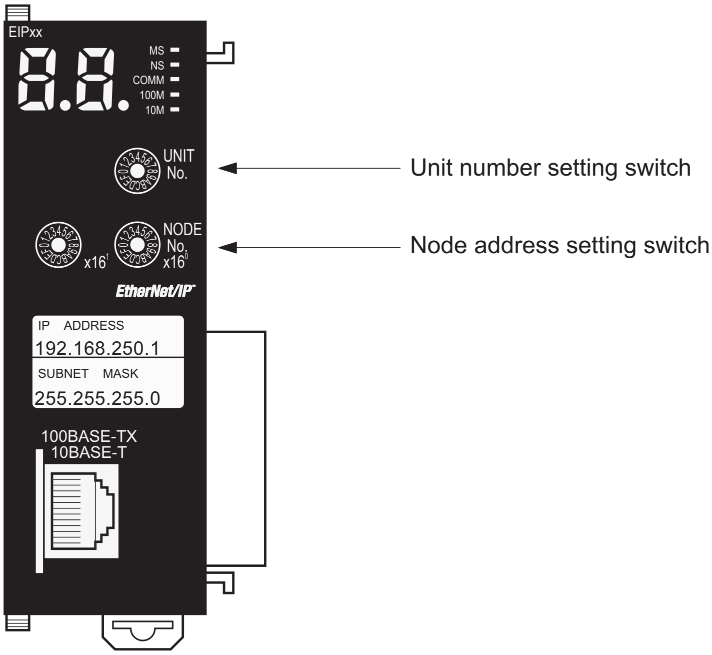
- Mount two CJ1W-EIP21S Units (hereafter referred to as EtherNet/IP Units) on one NJ-series CPU Unit.
- Connect two EtherNet/IP Units with an Ethernet cable.
- Unit A serves as the UDP/TCP client and Unit B serves as the UDP/TCP server.

## A-9-2 Required Settings for Sample Programing

This section describes the settings required to run the sample programs.

### Rotary Switch Setting

Set the rotary switches as follows.

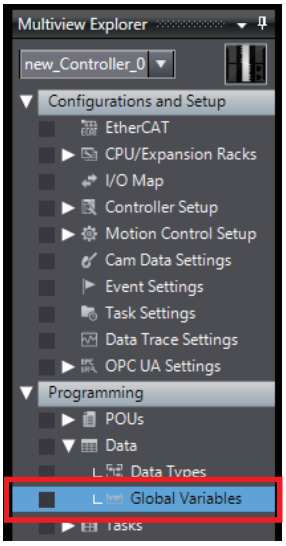


| Rotary switch | Unit A | Unit B |
|---|---|---|
| Unit number setting switch | 0 | 1 |
| Node address setting switch (×16$^1$) | 1 | 1 |
| Node address setting switch (×16$^0$) | 0 | 1 |

## Setup with the Sysmac Studio

### ● Creating Global Variables

Create global variables in the Global Variables Tab Page.



The table below shows the global variables required for all the sample programs.

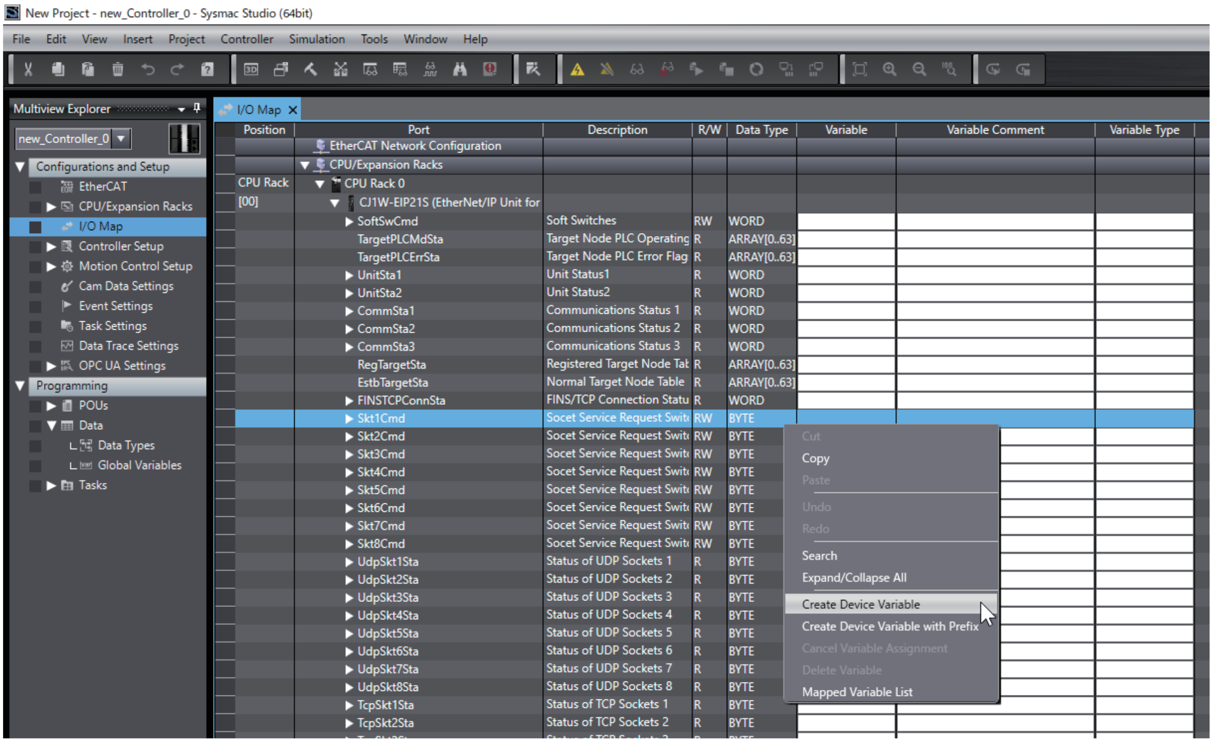
| Name | Data type | AT | Retain | Network Publish | Comment |
|---|---|---|---|---|---|
| J01_StaArea | _sCJEIP21_User_StaArea | | FALSE | Publish Only | For status area layout change |
| J02_StaArea | _sCJEIP21_User_StaArea | | FALSE | Publish Only | For status area layout change |

To use the data type _sCJEIP21_User_StaArea, you must register it in advance. Double-click **CPU/Expansion Racks** under **Configuration and Setup**, right-click the CJ1W-EIP21S, and select ***Register Data Types for Units*** from the menu to register it.

Refer to the description of each sample program for the global variables required for each sample program.

## ● Creating Device Variables

Create device variables for the ports to use in the I/O Map Tab Page.



| Position | Port | Description | R/W | Data Type | Variable | Variable Comment | Variable Type |
|---|---|---|---|---|---|---|---|
| | EtherCAT Network Configuration | | | | | | |
| | CPU/Expansion Racks | | | | | | |
| CPU Rack | CPU Rack 0 | | | | | | |
| [00] | CJ1W-EIP21S (EtherNet/IP Unit for | | | | | | |
| | SoftSwCmd | Soft Switches | RW | WORD | | | |
| | TargetPLCMdSta | Target Node PLC Operating | R | ARRAY[0..63] | | | |
| | TargetPLCErrSta | Target Node PLC Error Flag | R | ARRAY[0..63] | | | |
| | UnitSta1 | Unit Status1 | R | WORD | | | |
| | UnitSta2 | Unit Status2 | R | WORD | | | |
| | CommSta1 | Communications Status 1 | R | WORD | | | |
| | CommSta2 | Communications Status 2 | R | WORD | | | |
| | CommSta3 | Communications Status 3 | R | WORD | | | |
| | RegTargetSta | Registered Target Node Tal | R | ARRAY[0..63] | | | |
| | EstbTargetSta | Normal Target Node Table | R | ARRAY[0..63] | | | |
| | FINSTCPConnSta | FINS/TCP Connection Statu | R | WORD | | | |
| | Skt1Cmd | Socet Service Request Swit | RW | BYTE | | | |
| | Skt2Cmd | Socet Service Request Swit | RW | BYTE | | | |
| | Skt3Cmd | Socet Service Request Swit | RW | BYTE | | | |
| | Skt4Cmd | Socet Service Request Swit | RW | BYTE | | | |
| | Skt5Cmd | Socet Service Request Swit | RW | BYTE | | | |
| | Skt6Cmd | Socet Service Request Swit | RW | BYTE | | | |
| | Skt7Cmd | Socet Service Request Swit | RW | BYTE | | | |
| | Skt8Cmd | Socet Service Request Swit | RW | BYTE | | | |
| | UdpSkt1Sta | Status of UDP Sockets 1 | R | BYTE | | | |
| | UdpSkt2Sta | Status of UDP Sockets 2 | R | BYTE | | | |
| | UdpSkt3Sta | Status of UDP Sockets 3 | R | BYTE | | | |
| | UdpSkt4Sta | Status of UDP Sockets 4 | R | BYTE | | | |
| | UdpSkt5Sta | Status of UDP Sockets 5 | R | BYTE | | | |
| | UdpSkt6Sta | Status of UDP Sockets 6 | R | BYTE | | | |
| | UdpSkt7Sta | Status of UDP Sockets 7 | R | BYTE | | | |
| | UdpSkt8Sta | Status of UDP Sockets 8 | R | BYTE | | | |
| | TcpSkt1Sta | Status of TCP Sockets 1 | R | BYTE | | | |
| | TcpSkt2Sta | Status of TCP Sockets 2 | R | BYTE | | | |

Refer to the description of each sample program for the required device variables.

## ● Setting Up the Unit

From the CPU and Expansion Racks Tab Page, open the Edit Special Unit settings Tab Page for the EtherNet/IP Unit and set the following items. Refer to *13-4 Settings Required for Socket Service Function* for details on the settings.

| Setting | Unit A | Unit B |
|---|---|---|
| IP Address | 192.168.250.16 | 192.168.250.17 |
| Subnet mask | 255.255.255.0 | 255.255.255.0 |
| Layout Type | User setting | User setting |
| Allocation Area | J01_StaArea | J02_StaArea |

## A-9-3 Sample Programing for UDP Communications by Manipulating Device Variables

Use UDP communications and send 100 bytes (50 words) of data from Unit A to Unit B.

This sample program uses UDP socket No. 1 and UDP port No. 4096.

### Description of Operation

**(1) When the internal variable *EnableServer* in the program for Unit B (UDP server) is changed to TRUE, Unit B waits to receive UDP data.**

**(2) If the internal variable *EnableClient* in the program for Unit A (UDP client) is changed to TRUE, Unit A waits to start sending UDP data.**

**(3) When the internal variable *SendTrigger* in the program for Unit A is changed to TRUE, Unit A sends 100 bytes (50 words) of data that is stored in *J01_Skt1UdpSendDat* to Unit B.**

**(4) Unit B receives and stores the data in J02_Skt1UdpRcvDat.**

### User Program for Unit A

#### ● Global Variables

| Name | Data type | AT | Retain | Network Publish | Comment |
|---|---|---|---|---|---|
| J01_StaArea | _sCJEIP21_User_StaArea | | FALSE | Publish Only | For status area layout change |
| J01_Skt1UdpSendDat | ARRAY[0..49] OF WORD | %D0 | TRUE | Do not publish | For storage of send data |

#### ● Device Variables

| Port | Name | Network Publish | Comment |
|---|---|---|---|
| Skt1UdpOpenCmd | J01_Skt1UdpOpenCmd | Do not publish | |
| Skt1SendCmd | J01_Skt1SendCmd | Do not publish | |
| Skt1CloseCmd | J01_Skt1CloseCmd | Do not publish | |
| Skt1UdpOpened | J01_Skt1UdpOpened | Do not publish | |
| Skt1No | J01_Skt1No | Do not publish | |
| Skt1LocalPortNo | J01_Skt1LocalPortNo | Do not publish | |
| Skt1RemoteIPAdr1 | J01_Skt1RemoteIPAdr1 | Do not publish | |
| Skt1RemoteIPAdr2 | J01_Skt1RemoteIPAdr2 | Do not publish | |
| Skt1RemoteIPAdr3 | J01_Skt1RemoteIPAdr3 | Do not publish | |
| Skt1RemoteIPAdr4 | J01_Skt1RemoteIPAdr4 | Do not publish | |
| Skt1RemotePortNo | J01_Skt1RemotePortNo | Do not publish | |
| Skt1SendRcvByte | J01_Skt1SendRcvByte | Do not publish | |
| Skt1SendRcvAdrType | J01_Skt1SendRcvAdrType | Do not publish | Area type of the address to which the variable to store send data is allocated |
| Skt1SendRcvAdrCh1 | J01_Skt1SendRcvAdrCh1 | Do not publish | Upper two digits of the address to which the variable to store send data is allocated |
| Skt1SendRcvAdrCh2 | J01_Skt1SendRcvAdrCh2 | Do not publish | Lower two digits of the address to which the variable to store send data is allocated |
| Skt1RespCode | J01_Skt1RespCode | Do not publish | |

## Internal Variables

| Name | Data type | Comment |
|---|---|---|
| EnableClient | BOOL | |
| ClientState | INT | |
| SendTrigger | BOOL | |

## Program Code

```
IF (EnableClient=TRUE) THEN
    IF (ClientState=0) THEN
        ClientState:=1; // Next state is "Open".
    END_IF;
ELSE
    IF ((J01_Skt1UdpOpened=TRUE) AND (ClientState<5)) THEN
        ClientState:=5;  // Close opened socket.
    END_IF;
END_IF;

CASE ClientState OF
    0:  // Idle
        // Nothing to do.

    1: // Open
        J01_Skt1No:=USINT#1;
        J01_Skt1LocalPortNo:=UINT#4096;
        J01_Skt1UdpOpenCmd:=TRUE; // Open request
        ClientState:=2; // Next state is "Opening".

    2: // Opening
        IF (J01_Skt1UdpOpenCmd=FALSE) THEN // Open request completed
            IF (J01_Skt1RespCode=WORD#0000) THEN // Success
                ClientState:=3;  // Next state is "Opened".
            ELSE // Error
                EnableClient:=FALSE;
                ClientState:=0;  // Next state is "Idle".
            END_IF;
        END_IF;

    3: // Opened
        IF (SendTrigger=TRUE) THEN
            SendTrigger:=FALSE;
            J01_Skt1No              :=USINT#1;
            J01_Skt1RemoteIPAdr1    :=USINT#192;
            J01_Skt1RemoteIPAdr2    :=USINT#168;
            J01_Skt1RemoteIPAdr3    :=USINT#250;
            J01_Skt1RemoteIPAdr4    :=USINT#17;
            J01_Skt1RemotePortNo    :=UINT#4096;
            J01_Skt1SendRcvByte     :=UINT#100;
            J01_Skt1SendRcvAdrType  :=BYTE#16#82;
            J01_Skt1SendRcvAdrCh1   :=BYTE#16#00;
            J01_Skt1SendRcvAdrCh2   :=BYTE#16#00;
            J01_Skt1SendCmd:=TRUE; // Send request
            ClientState:=4; // Next state is "Sending".
        END_IF;

    4: // Sending
        IF (J01_Skt1SendCmd=FALSE) THEN // Send request completed
            IF (J01_Skt1RespCode=WORD#0000) THEN // Success
```

```
                ClientState:=3; // Next state is "Opened".
            ELSE // Error
                ClientState:=5; // Next state is "Close".
            END_IF;
        END_IF;

    5: // Close
        IF (J01_Skt1UdpOpened=TRUE) THEN
            J01_Skt1No:=USINT#1;
            J01_Skt1CloseCmd:=TRUE; // Close request
            ClientState:=6; // Next state is "Closing".
        ELSE
            EnableClient:=FALSE;
            ClientState:=0; // Next state is "Idle".
        END_IF;

    6: // Closing
        IF (J01_Skt1CloseCmd=FALSE) THEN // Close request completed
            EnableClient:=FALSE;
            ClientState:=0; // Next state is "Idle".
        END_IF;
END_CASE;
```

## User Program for Unit B

### ● Global Variables

| Name | Data type | AT | Retain | Network Publish | Comment |
|---|---|---|---|---|---|
| J02_StaArea | _sCJEIP21_User_StaArea | | FALSE | Publish Only | For status area layout change |
| J02_Skt1UdpRcvDat | ARRAY[0..49] OF WORD | %D1000 | TRUE | Do not publish | For storage of receive data |

### ● Device Variables

| Port | Name | Network Publish | Comment |
|---|---|---|---|
| Skt1UdpOpenCmd | J02_Skt1UdpOpenCmd | Do not publish | |
| Skt1RcvCmd | J02_Skt1RcvCmd | Do not publish | |
| Skt1CloseCmd | J02_Skt1CloseCmd | Do not publish | |
| Skt1UdpOpened | J02_Skt1UdpOpened | Do not publish | |
| Skt1No | J02_Skt1No | Do not publish | |
| Skt1LocalPortNo | J02_Skt1LocalPortNo | Do not publish | |
| Skt1SendRcvByte | J02_Skt1SendRcvByte | Do not publish | |
| Skt1SendRcvAdrType | J02_Skt1SendRcvAdrType | Do not publish | Area type of the address to which the variable to store receive data is allocated |
| Skt1SendRcvAdrCh1 | J02_Skt1SendRcvAdrCh1 | Do not publish | Upper two digits of the address to which the variable to store receive data is allocated |
| Skt1SendRcvAdrCh2 | J02_Skt1SendRcvAdrCh2 | Do not publish | Lower two digits of the address to which the variable to store receive data is allocated |
| Skt1Timeout | J02_Skt1Timeout | Do not publish | |
| Skt1RespCode | J02_Skt1RespCode | Do not publish | |

## ● Internal Variables

| Name | Data type | Comment |
|---|---|---|
| EnableServer | BOOL | |
| ServerState | INT | |

## ● Program Code

```
IF (EnableServer=TRUE) THEN
    IF (ServerState=0) THEN
        ServerState:=1; // Next state is "Open".
    END_IF;
ELSE
    IF ((J02_Skt1UdpOpened=TRUE) AND (ServerState<5)) THEN
        ServerState:=5;  // Close opened socket.
    END_IF;
END_IF;

CASE ServerState OF
    0: // Idle
        // Nothing to do.

    1: // Open
        J02_Skt1No          :=USINT#1;
        J02_Skt1LocalPortNo :=UINT#4096;
        J02_Skt1UdpOpenCmd  :=TRUE; // Open request
        ServerState:=2; // Next state is "Opening".

    2: // Opening
        IF (J02_Skt1UdpOpenCmd=FALSE) THEN // Open request completed
            IF (J02_Skt1RespCode=WORD#0000) THEN // Success
                ServerState:=3;  // Next state is "Opened".
            ELSE    // Error
                EnableServer:=FALSE;
                ServerState:=0;  // Next state is "Idle".
            END_IF;
        END_IF;

    3: // Opened
        J02_Skt1No              :=USINT#1;
        J02_Skt1SendRcvByte     :=UINT#100;
        J02_Skt1SendRcvAdrType  :=BYTE#16#82;
        J02_Skt1SendRcvAdrCh1   :=BYTE#16#03;
        J02_Skt1SendRcvAdrCh2   :=BYTE#16#E8;
        J02_Skt1Timeout         :=0;
        J02_Skt1RcvCmd          :=TRUE; // Receive request

        ServerState:=4; // Next state is "Receiving".

    4: // Receiving
        IF (J02_Skt1RcvCmd=FALSE) THEN // Receive request completed
            IF (J02_Skt1RespCode=WORD#0000) THEN // Success
                ServerState:=3; // Next state is "Opened".
            ELSE // Error
                ServerState:=5; // Next state is "Close".
            END_IF;
        END_IF;

    5: // Close
```

```
        IF (J02_Skt1UdpOpened=TRUE) THEN
            J02_Skt1No      :=USINT#1;
            J02_Skt1CloseCmd:=TRUE; // Close request
            ServerState:=6; // Next state is "Closing".
        ELSE
            EnableServer:=FALSE;
            ServerState:=0; // Next state is "Idle".
        END_IF;

    6: // Closing
        IF (J02_Skt1CloseCmd=FALSE) THEN // Close request completed
            EnableServer:=FALSE;
            ServerState:=0; // Next state is "Idle".
        END_IF;
END_CASE;
```

## A-9-4 Sample Programing for TCP Communications by Manipulating Device Variables

Use TCP communications and send 100 bytes (50 words) of data from Unit A to Unit B.

This sample program uses TCP socket No. 2 and TCP port No. 4097.

### Description of Operation

**(1) When the internal variable *EnableServer* in the program for Unit B (TCP server) is changed to TRUE, Unit B waits for a TCP connection.**

**(2) When the internal variable *EnableClient* in the program for Unit A (TCP client) is changed to TRUE, Unit A opens and establishes a TCP connection with Unit B.**

**(3) After the connection is established, Unit B waits to receive TCP data. Unit A waits to start sending TCP data.**

**(4) When the internal variable *SendTrigger* in the program for Unit A is changed to TRUE, Unit A sends 100 bytes (50 words) of data that is stored in *J01_Skt2TcpSendDat* to Unit B.**

**(5) Unit B receives and stores the data in *J02_Skt2TcpRcvDat*.**

### User Program for Unit A

- **Global Variables**

| Name | Data type | AT | Retain | Network Publish | Comment |
|---|---|---|---|---|---|
| J01_StaArea | _sCJEIP21_User_StaArea | | FALSE | Publish Only | For status area layout change |
| J01_Skt2TcpSendDat | ARRAY[0..49] OF WORD | %D100 | TRUE | Do not publish | For storage of send data |

- **Device Variables**

| Port | Name | Network Publish | Comment |
|---|---|---|---|
| Skt2TcpActiveCmd | J01_Skt2TcpActiveCmd | Do not publish | |
| Skt2SendCmd | J01_Skt2SendCmd | Do not publish | |
| Skt2CloseCmd | J01_Skt2CloseCmd | Do not publish | |
| Skt2TcpOpened | J01_Skt2TcpOpened | Do not publish | |
| Skt2No | J01_Skt2No | Do not publish | |
| Skt2RemoteIPAdr1 | J01_Skt2RemoteIPAdr1 | Do not publish | |
| Skt2RemoteIPAdr2 | J01_Skt2RemoteIPAdr2 | Do not publish | |
| Skt2RemoteIPAdr3 | J01_Skt2RemoteIPAdr3 | Do not publish | |
| Skt2RemoteIPAdr4 | J01_Skt2RemoteIPAdr4 | Do not publish | |
| Skt2RemotePortNo | J01_Skt2RemotePortNo | Do not publish | |
| Skt2SendRcvByte | J01_Skt2SendRcvByte | Do not publish | |
| Skt2SendRcvAdrType | J01_Skt2SendRcvAdrType | Do not publish | Area type of the address to which the variable to store send data is allocated |
| Skt2SendRcvAdrCh1 | J01_Skt2SendRcvAdrCh1 | Do not publish | Upper two digits of the address to which the variable to store send data is allocated |
| Skt2SendRcvAdrCh2 | J01_Skt2SendRcvAdrCh2 | Do not publish | Lower two digits of the address to which the variable to store send data is allocated |
| Skt2Timeout | J01_Skt2Timeout | Do not publish | |
| Skt2RespCode | J01_Skt2RespCode | Do not publish | |

## Internal Variables

| Name | Data type | Comment |
|---|---|---|
| EnableClient | BOOL | |
| ClientState | INT | |
| SendTrigger | BOOL | |

## Program Code

```
IF (EnableClient=TRUE) THEN
    IF (ClientState=0) THEN
        ClientState:=1; // Next state is "Open".
    END_IF;
ELSE
    IF ((J01_Skt2TcpOpened=TRUE) AND (ClientState<5)) THEN
            ClientState:=5;  // Close opened socket.
    END_IF;
END_IF;

CASE ClientState OF
    0:  // Idle
        // Nothing to do.

    1: // Open
            J01_Skt2No              :=USINT#2;
            J01_Skt2RemoteIPAdr1    :=USINT#192;
            J01_Skt2RemoteIPAdr2    :=USINT#168;
            J01_Skt2RemoteIPAdr3    :=USINT#250;
            J01_Skt2RemoteIPAdr4    :=USINT#17;
            J01_Skt2RemotePortNo    :=UINT#4097;
            J01_Skt2Timeout         :=0;
            J01_Skt2TcpActiveCmd    :=TRUE; // Active Open request
            ClientState:=2; // Next state is "Opening".

    2: // Opening
        IF (J01_Skt2TcpActiveCmd=FALSE) THEN // Active Open request completed
            IF (J01_Skt2RespCode=WORD#0000) THEN // Success
                ClientState:=3;  // Next state is "Opened".
            ELSE // Error
                EnableClient:=FALSE;
                ClientState:=0;  // Next state is "Idle".
            END_IF;
        END_IF;

    3: // Opened
        IF (SendTrigger=TRUE) THEN
            SendTrigger:=FALSE;
            J01_Skt2No              :=USINT#2;
            J01_Skt2SendRcvByte     :=UINT#100;
            J01_Skt2SendRcvAdrType  :=BYTE#16#82;
            J01_Skt2SendRcvAdrCh1   :=BYTE#16#00;
            J01_Skt2SendRcvAdrCh2   :=BYTE#16#64;
            J01_Skt2SendCmd         :=TRUE; // Send request

            ClientState:=4; // Next state is "Sending".
        END_IF;

    4: // Sending
        IF (J01_Skt2SendCmd=FALSE) THEN // Send request completed
```

```
                IF (J01_Skt2RespCode=WORD#0000) THEN // Success
                    ClientState:=3; // Next state is "Opened".
                ELSE // Error
                    ClientState:=5; // Next state is "Close".
                END_IF;
            END_IF;

        5: // Close
            IF (J01_Skt2TcpOpened=TRUE) THEN
                J01_Skt2No      :=USINT#2;
                J01_Skt2CloseCmd:=TRUE; // Close request
                ClientState:=6; // Next state is "Closing".
            ELSE
                EnableClient:=FALSE;
                ClientState:=0; // Next state is "Idle".
            END_IF;

        6: // Closing
            IF (J01_Skt2CloseCmd=FALSE) THEN // Close request completed
                EnableClient:=FALSE;
                ClientState:=0; // Next state is "Idle".
            END_IF;
    END_CASE;
```

## User Program for Unit B

### ● Global Variables

| Name | Data type | AT | Retain | Network Publish | Comment |
|---|---|---|---|---|---|
| J02_StaArea | _sCJEIP21_User_StaArea | | FALSE | Publish Only | For status area layout change |
| J02_Skt2TcpRcvDat | ARRAY[0..49] OF WORD | %D1100 | TRUE | Do not publish | For storage of receive data |

### ● Device Variables

| Port | Name | Network Publish | Comment |
|---|---|---|---|
| Skt2TcpPassiveCmd | J02_Skt2TcpPassiveCmd | Do not publish | |
| Skt2RcvCmd | J02_Skt2RcvCmd | Do not publish | |
| Skt2CloseCmd | J02_Skt2CloseCmd | Do not publish | |
| Skt2TcpDatRcved | J02_Skt2TcpDatRcved | Do not publish | |
| Skt2TcpOpened | J02_Skt2TcpOpened | Do not publish | |
| Skt2No | J02_Skt2No | Do not publish | |
| Skt2LocalPortNo | J02_Skt2LocalPortNo | Do not publish | |
| Skt2TcpRcvDatByte | J02_Skt2TcpRcvDatByte | Do not publish | |
| Skt2SendRcvByte | J02_Skt2SendRcvByte | Do not publish | |
| Skt2SendRcvAdrType | J02_Skt2SendRcvAdrType | Do not publish | Area type of the address to which the variable to store receive data is allocated |
| Skt2SendRcvAdrCh1 | J02_Skt2SendRcvAdrCh1 | Do not publish | Upper two digits of the address to which the variable to store receive data is allocated |
| Skt2SendRcvAdrCh2 | J02_Skt2SendRcvAdrCh2 | Do not publish | Lower two digits of the address to which the variable to store receive data is allocated |
| Skt2Timeout | J02_Skt2Timeout | Do not publish | |
| Skt2RespCode | J02_Skt2RespCode | Do not publish | |

### ● Internal Variables

| Name | Data type | Comment |
|---|---|---|
| EnableServer | BOOL | |
| ServerState | INT | |

### ● Program Code

```
IF (EnableServer=TRUE) THEN
    IF (ServerState=0) THEN
        ServerState:=1; // Next state is "Open".
    END_IF;
ELSE
    IF ((J02_Skt2TcpOpened=TRUE) AND (ServerState<5)) THEN
            ServerState:=5;  // Close opened socket.
    END_IF;

END_IF;


CASE ServerState OF
    0:  // Idle
        // Nothing to do.

    1: // Open
            J02_Skt2No              :=USINT#2;
            J02_Skt2LocalPortNo     :=UINT#4097;
            J02_Skt2Timeout         :=0;
            J02_Skt2TcpPassiveCmd   :=TRUE; // Passive Open request
            ServerState:=2; // Next state is "Opening".

    2: // Opening
        IF (J02_Skt2TcpPassiveCmd=FALSE) THEN // Passive Open request completed
            IF (J02_Skt2RespCode=WORD#0000) THEN // Success
                ServerState:=3;  // Next state is "Opened".
            ELSE    // Error
                EnableServer:=FALSE;
                ServerState:=0;  // Next state is "Idle".
            END_IF;
        END_IF;

    3: // Opened
        IF ((J02_Skt2TcpDatRcved=TRUE) AND (J02_Skt2TcpRcvDatByte>0)) THEN
            J02_Skt2No              :=USINT#2;
            J02_Skt2SendRcvByte     :=UINT#100;
            J02_Skt2SendRcvAdrType  :=BYTE#16#82;
            J02_Skt2SendRcvAdrCh1   :=BYTE#16#04;
            J02_Skt2SendRcvAdrCh2   :=BYTE#16#4C;
            J02_Skt2Timeout         :=0;
            J02_Skt2RcvCmd          :=TRUE; // Receive request

            ServerState:=4; // Next state is "Receiving".
        END_IF;

    4: // Receiving
        IF (J02_Skt2RcvCmd=FALSE) THEN // Receive request completed
            IF (J02_Skt2RespCode=WORD#0000) THEN // Success
                ServerState:=3; // Next state is "Opened".
            ELSE // Error
```

```
                ServerState:=5; // Next state is "Close".
            END_IF;
        END_IF;

    5: // Close
        IF (J02_Skt2TcpOpened=TRUE) THEN
            J02_Skt2No      :=USINT#2;
            J02_Skt2CloseCmd:=TRUE; // Close request
            ServerState:=6; // Next state is "Closing".
        ELSE
            EnableServer:=FALSE;
            ServerState:=0; // Next state is "Idle".
        END_IF;

    6: // Closing
        IF (J02_Skt2CloseCmd=FALSE) THEN // Close request completed
            EnableServer:=FALSE;
            ServerState:=0; // Next state is "Idle".
        END_IF;
END_CASE;
```

## A-9-5 Sample Programing for UDP Communications by SendCmd Instruction

Use UDP communications and send 100 bytes (50 words) of data from Unit A to Unit B.

This sample program uses UDP socket No. 3 and UDP port No. 4098.

### Description of Operation

**(1) When the internal variable *EnableServer* in the program for Unit B (UDP server) is changed to TRUE, Unit B waits to receive UDP data.**

**(2) If the internal variable *EnableClient* in the program for Unit A (UDP client) is changed to TRUE, Unit A waits to start sending UDP data.**

**(3) When the internal variable *SendTrigger* in the program for Unit A is changed to TRUE, Unit A sends 100 bytes (50 words) of data that is stored in the internal variable *SendDat* for the program to Unit B.**

**(4) Unit B receives and stores the data in the internal variable *RcvDat* in the program.**

### User Program for Unit A

- **Global Variables**

| Name | Data type | AT | Retain | Network Publish | Comment |
|---|---|---|---|---|---|
| J01_StaArea | _sCJEIP21_User_StaArea | | FALSE | Publish Only | For status area layout change |
| J01_Skt3UdpOpenResult | WORD | %D2200 | TRUE | Do not publish | For storage of the socket open request result |
| J01_Skt3UdpSendResult | ARRAY[0..1] OF WORD | %D2220 | TRUE | Do not publish | For storage of the send request result |
| J01_Skt3UdpCloseResult | WORD | %D2210 | TRUE | Do not publish | For storage of socket close request result |

- **Device Variables**

| Port | Name | Network Publish | Comment |
|---|---|---|---|
| Skt3UdpOpening | J01_Skt3UdpOpening | Do not publish | |
| Skt3UdpSending | J01_Skt3UdpSending | Do not publish | |
| Skt3UdpClosing | J01_Skt3UdpClosing | Do not publish | |
| Skt3UdpOpened | J01_Skt3UdpOpened | Do not publish | |

- **Internal Variables**

| Name | Data type | Comment |
|---|---|---|
| EnableClient | BOOL | |
| ClientState | INT | |
| SendTrigger | BOOL | |
| SendCmd_instance | SendCmd | |
| CmdDst | _sDNET_ADR | |
| CmdReq | ARRAY[0..255] OF BYTE | |
| CmdResp | ARRAY[0..3] OF BYTE | |

| Name | Data type | Comment |
|---|---|---|
| CmdOption | _sRESPONSE | |
| PortUsing | BOOL | |
| SendDat | ARRAY[0..49] OF WORD | For storage of send data |

## ● Program Code

```
// State
//  0: Idle
// 10: Prepare Open,        11: Open,       12: Opening,        13: Opened
// 20: Prepare Send,        21: Send,       22: Sending
// 40: Prepare Close,       41: Close,      42: Closing

IF (EnableClient=TRUE) THEN
    IF (ClientState=0) THEN
        ClientState:=10; // Next state is "Prepare Open".
    END_IF;
ELSE
    IF (J01_Skt3UdpOpened=TRUE) AND (ClientState<40) THEN
        ClientState:=40; // Close opened socket. Next state is "Prepare Close".
    END_IF;
END_IF;

CASE ClientState OF
    0:  // Idle
        // Nothing to do.

    10: // Prepare Open
        CmdDst.UnitNo:=BYTE#16#10; // Unit number(=0)
        CmdReq[0]:=BYTE#16#27; // Command
        CmdReq[1]:=BYTE#16#01; //  same as above
        CmdReq[2]:=BYTE#16#00; // Socket option
        CmdReq[3]:=BYTE#16#03; // Socket number(=3)
        CmdReq[4]:=BYTE#16#82; // Result storage area(=D2200/J01_Skt3UpdOpenResult)
        CmdReq[5]:=BYTE#16#08; //  same as above
        CmdReq[6]:=BYTE#16#98; //  same as above
        CmdReq[7]:=BYTE#16#00; //  same as above
        CmdReq[8]:=BYTE#16#10; // Local port number(=4098)
        CmdReq[9]:=BYTE#16#02; //  same as above

        SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
        PortUsing:=FALSE;
        ClientState:=11; // Next state is "Open".

    11: // Open
        IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
            PortUsing:=TRUE;
        END_IF;
        IF (PortUsing=TRUE) THEN
            SendCmd_instance(
                Execute:=TRUE,
                DstNetAdr:=CmdDst,
                CommPort:=_NONE,
                CmdDat:=CmdReq[0],
                CmdSize:=UINT#10,
                RespDat:=CmdResp[0],
                Option:=CmdOption);

            IF (SendCmd_instance.Done=TRUE) THEN
            ClientState:=12; // Next state is "Opening".
        ELSIF (SendCmd_instance.Error=TRUE) THEN
            EnableClient:=FALSE;
            ClientState:=0;  // Next state is "Idle".
        END_IF;
    END_IF;

12: // Opening
    IF (J01_Skt3UdpOpening=FALSE) THEN
         IF (J01_Skt3UdpAreaErr=FALSE) AND (J01_Skt3UdpOpenResult=WORD#16#0000) THEN
            // Successfully opened
            ClientState:=13; // Next state is "Opened".
         ELSE // Fail to open
            EnableClient:=FALSE;
            ClientState:=0;  // Next state is "Idle".
         END_IF;
    END_IF;

13: // Opened
    IF (SendTrigger=TRUE) THEN
        SendTrigger:=FALSE;
        ClientState:=20; // Next state is "Prepare Send"
    END_IF;

20: // Prepare Send
    CmdDst.UnitNo:=BYTE#16#10; // Unit number(=0)
    CmdReq[0] :=BYTE#16#27; // Command
    CmdReq[1] :=BYTE#16#03; //  same as above
    CmdReq[2] :=BYTE#16#00; // Socket option
    CmdReq[3] :=BYTE#16#03; // Socket number(=3)
    CmdReq[4] :=BYTE#16#82; // Result storage area(=D2220/J01_Skt3UdpSendResult)
    CmdReq[5] :=BYTE#16#08; //  same as above
    CmdReq[6] :=BYTE#16#AC; //  same as above
    CmdReq[7] :=BYTE#16#00; //  same as above
    CmdReq[8] :=BYTE#16#C0; // Remote IP address(=192.168.250.17)
    CmdReq[9] :=BYTE#16#A8; //  same as above
    CmdReq[10]:=BYTE#16#FA; //  same as above
    CmdReq[11]:=BYTE#16#11; //  same as above
    CmdReq[12]:=BYTE#16#10; // Remote port number(=4098)
    CmdReq[13]:=BYTE#16#02; //  same as above
    CmdReq[14]:=BYTE#16#00; // Number of bytes sent(=100)
    CmdReq[15]:=BYTE#16#64; //  same as above
    // CmdReqDat[16..115] is data.
    // SendDat(WORD Type) and CmdReq(BYTE Type) have different endiannesses, so swap each bytes.
    ToAryByte(SendDat, _HIGH_LOW ,CmdReq[16]);

    SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
    PortUsing:=FALSE;
    ClientState:=21; // Next state is "Send".

21: // Send
    IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
        PortUsing:=TRUE;
    END_IF;
    IF (PortUsing=TRUE) THEN
            SendCmd_instance(
                Execute:=TRUE,
                DstNetAdr:=CmdDst,
                CommPort:=_NONE,
                CmdDat:=CmdReq[0],
                CmdSize:=UINT#116,
                RespDat:=CmdResp[0],
                Option:=CmdOption);

            IF (SendCmd_instance.Done=TRUE) THEN
                ClientState:=22; // Next state is "Sending".
            ELSIF (SendCmd_instance.Error=TRUE) THEN
                ClientState:=40; // Next state is "Prepare Close".
            END_IF;
        END_IF;

22: // Sending
    IF (J01_Skt3UdpSending=FALSE) THEN
        IF (J01_Skt3UdpAreaErr=FALSE) AND (J01_Skt3UdpSendResult[0]=WORD#16#0000) THEN
            // Successfully sent.
            ClientState:=13; // Next state is "Opened".
        ELSE // Fail to send.
            ClientState:=40; // Next state is "Prepare Close"
        END_IF;
    END_IF;

40: // Prepare Close
    IF (J01_Skt3UdpOpened=TRUE) THEN
        CmdDst.UnitNo:=BYTE#16#10; // Unit number(=0)
        CmdReq[0]:=BYTE#16#27; // Command
        CmdReq[1]:=BYTE#16#04; //  same as above
        CmdReq[2]:=BYTE#16#00; // Socket option
        CmdReq[3]:=BYTE#16#03; // Socket number(=3)
        CmdReq[4]:=BYTE#16#82; // Result storage area(=D2210/J01_Skt3UdpCloseResult)
        CmdReq[5]:=BYTE#16#08; //  same as above
        CmdReq[6]:=BYTE#16#A2; //  same as above
        CmdReq[7]:=BYTE#16#00; //  same as above

        SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
        PortUsing:=FALSE;
        ClientState:=41; // Next state is "Close".
    ELSE
        EnableClient:=FALSE;
        ClientState:=0;  // Next state is "Idle".
    END_IF;

41: // Close
    IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
        PortUsing:=TRUE;
    END_IF;
    IF (PortUsing=TRUE) THEN
        SendCmd_instance(
            Execute:=TRUE,
            DstNetAdr:=CmdDst,
            CommPort:=_NONE,
            CmdDat:=CmdReq[0],
            CmdSize:=UINT#8,
            RespDat:=CmdResp[0],
            Option:=CmdOption);

            IF (SendCmd_instance.Done=TRUE) THEN
                ClientState:=42; // Next state is "Closing".
            ELSIF (SendCmd_instance.Error=TRUE) THEN
```

```
                    EnableClient:=FALSE;
                    ClientState:=0;  // Next state is "Idle".
                END_IF;
        END_IF;

    42:  // Closing
        IF (J01_Skt3UdpClosing=FALSE) THEN
            EnableClient:=FALSE;
            ClientState:=0; // Next state is "Idle".
        END_IF;
END_CASE;
```

## User Program for Unit B

### ● Global Variables

| Name | Data type | AT | Retain | Network Publish | Comment |
|---|---|---|---|---|---|
| J02_StaArea | _sCJEIP21_User_StaArea | | FALSE | Publish Only | For status area layout change |
| J02_Skt3UdpOpenResult | WORD | %D3200 | TRUE | Do not publish | For storage of the socket open request result |
| J02_Skt3UdpRcvResult | ARRAY[0..54] OF WORD | %D3230 | TRUE | Do not publish | For storage of receive request result and receive data |
| J02_Skt3UdpCloseResult | WORD | %D3210 | TRUE | Do not publish | For storage of socket close request result |

### ● Device Variables

| Port | Name | Network Publish | Comment |
|---|---|---|---|
| Skt3UdpOpening | J02_Skt3UdpOpening | Do not publish | |
| Skt3UdpRcving | J02_Skt3UdpRcving | Do not publish | |
| Skt3UdpClosing | J02_Skt3UdpClosing | Do not publish | |
| Skt3UdpOpened | J02_Skt3UdpOpened | Do not publish | |

### ● Internal Variables

| Name | Data type | Comment |
|---|---|---|
| EnableServer | BOOL | |
| ServerState | INT | |
| SendCmd_instance | SendCmd | |
| CmdDst | _sDNET_ADR | |
| CmdReq | ARRAY[0..16] OF BYTE | |
| CmdResp | ARRAY[0..3] OF BYTE | |
| CmdOption | _sRESPONSE | |
| PortUsing | BOOL | |
| RcvDat | ARRAY[0..49] OF WORD | For storage of receive data |

## Program Code

```
// State
//  0: Idle
// 10: Prepare Open,       11: Open,       12: Opening,       13: Opened
// 30: Prepare Receive,    31: Receive,    32: Receiving
// 40: Prepare Close,      41: Close,      42: Closing

IF (EnableServer=TRUE) THEN
    IF (ServerState=0) THEN
        ServerState:=10; // Next state is "Prepare Open".
    END_IF;
ELSE
    IF ((J02_Skt3UdpOpened=TRUE) AND (ServerState<40)) THEN
        ServerState:=40; // Close opened socket. Next state is "Prepare Close".
    END_IF;
END_IF;

CASE ServerState OF
    0:  // Idle
        // Nothing to do.

    10: // Prepare Open
        CmdDst.UnitNo:=BYTE#16#11; // Unit number(=1)
        CmdReq[0]:=BYTE#16#27; // Command
        CmdReq[1]:=BYTE#16#01; //  same as above
        CmdReq[2]:=BYTE#16#00; // Socket option
        CmdReq[3]:=BYTE#16#03; // Socket number(=3)
        CmdReq[4]:=BYTE#16#82; // Result storage area(=D3200/J02_Skt3UpdOpenResult)
        CmdReq[5]:=BYTE#16#0C; //  same as above
        CmdReq[6]:=BYTE#16#80; //  same as above
        CmdReq[7]:=BYTE#16#00; //  same as above
        CmdReq[8]:=BYTE#16#10; //  Local port number(=4098)
        CmdReq[9]:=BYTE#16#02; //  same as above

        SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
        PortUsing:=FALSE;
        ServerState:=11; // Next state is "Open".

    11: // Open
        IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
            PortUsing:=TRUE;
        END_IF;
        IF (PortUsing=TRUE) THEN
            SendCmd_instance(
                Execute:=TRUE,
                DstNetAdr:=CmdDst,
                CommPort:=_NONE,
                CmdDat:=CmdReq[0],
                CmdSize:=UINT#10,
                RespDat:=CmdResp[0],
                Option:=CmdOption);

            IF (SendCmd_instance.Done=TRUE) THEN
                ServerState:=12; // Next state is "Opening".
            ELSIF (SendCmd_instance.Error=TRUE) THEN
                ServerState:=0;  // Next state is "Idle".
            END_IF;
        END_IF;
```

```
12: // Opening
    IF (J02_Skt3UdpOpening=FALSE) THEN
        IF (J02_Skt3UdpAreaErr=FALSE) AND (J02_Skt3UdpOpenResult=WORD#16#0000) THEN
            // Successfully opened
            ServerState:=13; // Next state is "Opened".
        ELSE // Fail to open
            ServerState:=0;  // Next state is "Idle".
        END_IF;

    END_IF;

13: // Opened
     ServerState:=30; // Next state is "Prepare Receive".

30: // Prepare Receive
    CmdDst.UnitNo:=BYTE#16#11; // Unit number(=1)
    CmdReq[0] :=BYTE#16#27; // Command
    CmdReq[1] :=BYTE#16#02; //  same as above
    CmdReq[2] :=BYTE#16#00; // Socket option
    CmdReq[3] :=BYTE#16#03; // Socket number(=3)
    CmdReq[4] :=BYTE#16#82; // Result storage area(=D3230/J02_SKt3UdpRcvResult)
    CmdReq[5] :=BYTE#16#0C; //  same as above
    CmdReq[6] :=BYTE#16#9E; //  same as above
    CmdReq[7] :=BYTE#16#00; //  same as above
    CmdReq[8] :=BYTE#16#00; // Number of reception bytes(=100)
    CmdReq[9] :=BYTE#16#64; //  same as above
    CmdReq[10]:=BYTE#16#00; // Timeout value(=0:unlimited)
    CmdReq[11]:=BYTE#16#00; //  same as above

    SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
    PortUsing:=FALSE;
    ServerState:=31; // Next state is "Receive".

31: // Receive
    IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
        PortUsing:=TRUE;
    END_IF;
    IF (PortUsing=TRUE) THEN
        SendCmd_instance(
            Execute:=TRUE,
            DstNetAdr:=CmdDst,
            CommPort:=_NONE,
            CmdDat:=CmdReq[0],
            CmdSize:=UINT#12,
            RespDat:=CmdResp[0],
            Option:=CmdOption);

        IF (SendCmd_instance.Done=TRUE) THEN
            ServerState:=32; // Next state is "Receiving".
        ELSIF (SendCmd_instance.Error=TRUE) THEN
            ServerState:=40; // Next state is "Prepare Close".
        END_IF;
    END_IF;

32: // Receiving
    IF (J02_Skt3UdpRcving=FALSE) THEN
        IF (J02_Skt3UdpAreaErr=FALSE) AND (J02_Skt3UdpRcvResult[0]=WORD#16#0000) THEN
            // Successfully received. You can process received data here.
            AryMove(J02_Skt3UdpRcvResult[5], RcvDat[0], UINT#50);
```

```
                ServerState:=13; // Next state is "Opened".
            ELSE // Fail to receive.
                ServerState:=40; // Next state is "Prepare Close".
            END_IF;

        END_IF;

    40: // Prepare Close
        IF (J02_Skt3UdpOpened=TRUE) THEN
            CmdDst.UnitNo:=BYTE#16#11; // Unit number(=1)
            CmdReq[0]:=BYTE#16#27; // Command
            CmdReq[1]:=BYTE#16#04; //  same as above
            CmdReq[2]:=BYTE#16#00; // Socket option
            CmdReq[3]:=BYTE#16#03; // Socket number(=3)
            CmdReq[4]:=BYTE#16#82; // Result storage area( =D3210/J02_Skt3UdpCloseResult)
            CmdReq[5]:=BYTE#16#0C; //  same as above
            CmdReq[6]:=BYTE#16#8A; //  same as above
            CmdReq[7]:=BYTE#16#00; //  same as above

            SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
            PortUsing:=FALSE;
            ServerState:=41; // Next state is "Close".
        ELSE
            ServerState:=0;  // Next state is "Idle".
        END_IF;

    41: // Close
        IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
            PortUsing:=TRUE;
        END_IF;
        IF (PortUsing=TRUE) THEN
            SendCmd_instance(
                Execute:=TRUE,
                DstNetAdr:=CmdDst,
                CommPort:=_NONE,
                CmdDat:=CmdReq[0],
                CmdSize:=UINT#8,
                RespDat:=CmdResp[0],
                Option:=CmdOption);

                IF (SendCmd_instance.Done=TRUE) THEN
                    ServerState:=42; // Next state is "Closing".
                ELSIF (SendCmd_instance.Error=TRUE) THEN
                    ServerState:=0;  // Next state is "Idle".
                END_IF;
        END_IF;

    42:  // Closing
        IF (J02_Skt3UdpClosing=FALSE) THEN
             ServerState:=0; // Next state is "Idle".
        END_IF;
END_CASE;
```

## A-9-6 Sample Programing for TCP Communications by SendCmd Instruction

Use TCP communications and send 100 bytes (50 words) of data from Unit A to Unit B.

This sample program uses TCP socket No. 4 and TCP port No. 4099.

### Description of Operation

**(1) When the internal variable *EnableServer* in the program for Unit B (TCP server) is changed to TRUE, Unit B waits for a TCP connection.**

**(2) When the internal variable *EnableClient* in the program for Unit A (TCP client) is changed to TRUE, Unit A opens and establishes a TCP connection with Unit B.**

**(3) After the connection is established, Unit B waits to receive TCP data. Unit A waits to start sending TCP data.**

**(4) When the internal variable *SendTrigger* in the program for Unit A is changed to TRUE, Unit A sends 100 bytes (50 words) of data that is stored in the internal variable *SendDat* for the program to Unit B.**

**(5) Unit B receives and stores the data in the internal variable *RcvDat* in the program.**

### User Program for Unit A

- **Global Variables**

| Name | Data type | AT | Retain | Network Publish | Comment |
|---|---|---|---|---|---|
| J01_StaArea | _sCJEIP21_User_StaArea | | FALSE | Publish Only | For status area layout change |
| J01_Skt4TcpActOpenResult | ARRAY[0..1] OF WORD | %D2300 | TRUE | Do not publish | For storage of the socket open request result |
| J01_Skt4TcpSendResult | ARRAY[0..1] OF WORD | %D2320 | TRUE | Do not publish | For storage of the send request result |
| J01_Skt4TcpCloseResult | WORD | %D2310 | TRUE | Do not publish | For storage of socket close request result |

- **Device Variables**

| Port | Name | Network Publish | Comment |
|---|---|---|---|
| Skt4TcpOpening | J01_Skt4TcpOpening | Do not publish | |
| Skt4TcpSending | J01_Skt4TcpSending | Do not publish | |
| Skt4TcpClosing | J01_Skt4TcpClosing | Do not publish | |
| Skt4TcpOpened | J01_Skt4TcpOpened | Do not publish | |

- **Internal Variables**

| Name | Data type | Comment |
|---|---|---|
| EnableClient | BOOL | |
| ClientState | INT | |
| CmdDst | _sDNET_ADR | |
| CmdReq | ARRAY[0..255] OF BYTE | |
| CmdResp | ARRAY[0..3] OF BYTE | |
| SendCmd_instance | SendCmd | |
| PortUsing | BOOL | |

| Name | Data type | Comment |
|---|---|---|
| CmdOption | _sRESPONSE | |
| SendTrigger | BOOL | |
| SendDat | ARRAY[0..49] OF WORD | |

## ● Program Code

```
// State
//  0: Idle
// 10: Prepare Open,       11: Open,      12: Opening,       13: Opened
// 20: Prepare Send,       21: Send,      22: Sending
// 40: Prepare Close,      41: Close,     42: Closing

IF (EnableClient=TRUE) THEN
    IF (ClientState=0) THEN
        ClientState:=10; // Next state is "Prepare Open".
    END_IF;
ELSE
    IF (J01_Skt4TcpOpened=TRUE) AND (ClientState < 40) THEN
        ClientState:=40; // Close opened socket. Next state is "Prepare Close".
    END_IF;
END_IF;

CASE ClientState OF
    0: //Idle
        //Nothing to do

    10: //Prepare Open
        CmdDst.UnitNo:=BYTE#16#10; // Unit number(=0)
        CmdReq[0] :=BYTE#16#27; // Command(Tcp Socket Open Request)
        CmdReq[1] :=BYTE#16#11; //  same as above
        CmdReq[2] :=BYTE#16#00; // Socket option
        CmdReq[3] :=BYTE#16#04; // Socket number(=4)
        CmdReq[4] :=BYTE#16#82; // Result storage area(=D2300/J01_Skt4TcpActOpenResult)
        CmdReq[5] :=BYTE#16#08; //  same as above
        CmdReq[6] :=BYTE#16#FC; //  same as above
        CmdReq[7] :=BYTE#16#00; //  same as above
        CmdReq[8] :=BYTE#16#10; //  Local port number(=4099)
        CmdReq[9] :=BYTE#16#03; //  same as above
        CmdReq[10]:=BYTE#16#C0; // Remote IP Addr(=192.168.250.17)
        CmdReq[11]:=BYTE#16#A8; // same as above
        CmdReq[12]:=BYTE#16#FA; // same as above
        CmdReq[13]:=BYTE#16#11; // same as above
        CmdReq[14]:=BYTE#16#10; // Remote TCP Port(=4099)
        CmdReq[15]:=BYTE#16#03; // same as above

        SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
        PortUsing:=FALSE;
        ClientState:=11; // Next state is "Open"

    11: // Open
        IF (_Port_isAvailable=TRUE) AND (PortUsing=FALSE) THEN
            PortUsing:=TRUE;
        END_IF;
        IF (PortUsing=TRUE) THEN
            SendCmd_instance(
                Execute:=TRUE,
                DstNetAdr:=CmdDst,
                CommPort:=_NONE,
                CmdDat:=CmdReq[0],
                CmdSize:=UINT#16,
                RespDat:=CmdResp[0],
                Option:=CmdOption);

            IF SendCmd_instance.Done=TRUE THEN
                ClientState:=12; // Next state is "Opening"
            ELSIF SendCmd_instance.Error=TRUE THEN
                EnableClient:=FALSE;
                ClientState:=0;  // Next state is "Idle"
            END_IF;
        END_IF;

    12: // Opening
        IF (J01_Skt4TcpOpening=FALSE) THEN
            IF (J01_Skt4TcpAreaErr=FALSE) AND (J01_Skt4TcpActOpenResult[0]=WORD#16#0000) THEN
                // Successfully opened
                ClientState:=13; // Next state is "Opened"
            ELSE // Fail to open
                EnableClient:=FALSE;
                ClientState:=0;  // Next state is "Idle"
            END_IF;
        END_IF;

    13: // Opened
        IF (SendTrigger=TRUE) THEN
            ClientState:=20; // Next state is "Prepare Send"
            SendTrigger:=FALSE;
        END_IF;

    20: // Prepare Send
        CmdDst.UnitNo:=BYTE#16#10; // Unit number(=0)
        CmdReq[0]:=BYTE#16#27; // Command(TCP Socket Send Request)
        CmdReq[1]:=BYTE#16#13; //  same as above
        CmdReq[2]:=BYTE#16#00; // Socket option
        CmdReq[3]:=BYTE#16#04; // Socket number(=4)
        CmdReq[4]:=BYTE#16#82; // Result storage area(=D2320/J01_Skt4TcpSendResult)
        CmdReq[5]:=BYTE#16#09; //  same as above
        CmdReq[6]:=BYTE#16#10; //  same as above
        CmdReq[7]:=BYTE#16#00; //  same as above
        CmdReq[8]:=BYTE#16#00; //  Number of bytes sent(=100)
        CmdReq[9]:=BYTE#16#64; //  same as above
        // CmdReq[10..109] is data.
        // SendDat(WORD Type) and CmdReq(BYTE Type) have different endiannesses, so swap each bytes.
        ToAryByte(SendDat, _HIGH_LOW ,CmdReq[10]);

        SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
        PortUsing:=FALSE;
        ClientState:=21; // Next state is "Send"

    21: // Send
        IF (_Port_isAvailable=TRUE) AND (PortUsing=FALSE) THEN
            PortUsing:=TRUE;
        END_IF;
        IF (PortUsing=TRUE) THEN
            SendCmd_instance(
                Execute:=TRUE,
                DstNetAdr:=CmdDst,
                CommPort:=_NONE,
            CmdDat:=CmdReq[0],
            CmdSize:=UINT#110,
            RespDat:=CmdResp[0],
            Option:=CmdOption);

        IF SendCmd_instance.Done=TRUE THEN
            ClientState:=22; // Next state is "Sending"
        ELSIF SendCmd_instance.Error=TRUE THEN
            ClientState:=40; // Next state is "Prepare Close"
        END_IF;
    END_IF;

22: // Sending
    IF (J01_Skt4TcpSending=FALSE) THEN
        IF (J01_Skt4TcpAreaErr=FALSE) AND (J01_Skt4TcpSendResult[0]=WORD#16#0000) THEN
            // Successfully sent
            ClientState:=13; // Next state is "Opened"
        ELSE // Fail to send
            ClientState:=40; // Next state is "Prepare Close"
        END_IF;
    END_IF;

40: // Prepare Close
    IF (J01_Skt4TcpOpened=TRUE) THEN
        CmdDst.UnitNo:=BYTE#16#10; // Unit number(=0)
        CmdReq[0]:=BYTE#16#27; // Command(TCP Socket Close Request)
        CmdReq[1]:=BYTE#16#14; //  same as above
        CmdReq[2]:=BYTE#16#00; // Socket option
        CmdReq[3]:=BYTE#16#04; // Socket number(=4)
        CmdReq[4]:=BYTE#16#82; // Result storage area(=D2310/J01_Skt4TcpCloseResult)
        CmdReq[5]:=BYTE#16#09; //  same as above
        CmdReq[6]:=BYTE#16#06; //  same as above
        CmdReq[7]:=BYTE#16#00; //  same as above

        SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
        PortUsing:=FALSE;
        ClientState:=41; // Next state is "Close"
    ELSE
        ClientState:=0;  // Next state is "Idle".
    END_IF;

41: // Close
    IF (_Port_isAvailable=TRUE) AND (PortUsing=FALSE) THEN
        PortUsing:=TRUE;
    END_IF;
    IF (PortUsing=TRUE) THEN
        SendCmd_instance(
            Execute:=TRUE,
            DstNetAdr:=CmdDst,
            CommPort:=_NONE,
            CmdDat:=CmdReq[0],
            CmdSize:=UINT#8,
            RespDat:=CmdResp[0],
            Option:=CmdOption);

        IF SendCmd_instance.Done=TRUE THEN
            ClientState:=42; // Next state is "Closing".
        ELSIF SendCmd_instance.Error=TRUE THEN
            EnableClient:=FALSE;
```

```
                    ClientState:=0;  // Next state is "Idle".
                END_IF;
            END_IF;

        42: // Closing
            IF (J01_Skt4TcpClosing=FALSE) THEN
                EnableClient:=FALSE;
                ClientState:=0; // Next state is "Idle"
            END_IF;
    END_CASE;
```

## User Program for Unit B

### Global Variables

| Name | Data type | AT | Retain | Network Publish | Comment |
|---|---|---|---|---|---|
| J02_StaArea | _sCJEIP21_User_StaArea | | FALSE | Publish Only | For status area layout change |
| J02_Skt4TcpPsvOpenResult | ARRAY[0..3] OF WORD | %D3300 | TRUE | Do not publish | For storage of the socket open request result |
| J02_Skt4TcpRcvResult | ARRAY[0..51] OF WORD | %D3330 | TRUE | Do not publish | For storage of receive request result and receive data |
| J02_Skt4TcpCloseResult | WORD | %D3310 | TRUE | Do not publish | For storage of socket close request result |

### Device Variables

| Port | Name | Network Publish | Comment |
|---|---|---|---|
| Skt4TcpOpening | J02_Skt4TcpOpening | Do not publish | |
| Skt4TcpRcving | J02_Skt4TcpRcving | Do not publish | |
| Skt4TcpClosing | J02_Skt4TcpClosing | Do not publish | |
| Skt4TcpDatRcved | J02_Skt4TcpDatRcved | Do not publish | |
| Skt4TcpOpened | J02_Skt4TcpOpened | Do not publish | |
| Skt4TcpRcvDatByte | J02_Skt4TcpRcvDatByte | Do not publish | |

### Internal Variables

| Name | Data type | Comment |
|---|---|---|
| EnableServer | BOOL | |
| ServerState | INT | |
| CmdDst | _sDNET_ADR | |
| CmdReq | ARRAY[0..255] OF BYTE | |
| CmdResp | ARRAY[0..3] OF BYTE | |
| PortUsing | BOOL | |
| CmdOption | _sRESPONSE | |
| SendCmd_instance | SendCmd | |
| RcvDat | ARRAY[0..49] OF WORD | For storage of receive data |

## Program Code

```
// State
//  0: Idle
// 10: Prepare Open,       11: Open,      12: Opening,       13: Opened
// 30: Prepare Receive,    31: Receive    32: Receiving
// 40: Prepare Close,      41: Close,     42: Closing

IF (EnableServer=TRUE) THEN
    IF (ServerState=0) THEN
        ServerState:=10; // Next state is "Prepare Open".
    END_IF;
ELSE
    IF (J02_Skt4TcpOpened=TRUE) AND (ServerState < 40) THEN
        ServerState:=40; // Close opened socket. Next state is "Prepare Close".
    END_IF;
END_IF;

CASE ServerState OF
    0: // Idle
        // Nothing to do

    10: //Prepare Open
        CmdDst.UnitNo:=BYTE#16#11; // Unit number(=1)
        CmdReq[0] :=BYTE#16#27; // Command(Tcp Socket Passive Open Request)
        CmdReq[1] :=BYTE#16#10; //  same as above
        CmdReq[2] :=BYTE#16#00; // Socket option
        CmdReq[3] :=BYTE#16#04; // Socket number(=4)
        CmdReq[4] :=BYTE#16#82; // Result storage area(=D3300/J02_Skt4TcpPcvOpenResult)
        CmdReq[5] :=BYTE#16#0C; //  same as above
        CmdReq[6] :=BYTE#16#E4; //  same as above
        CmdReq[7] :=BYTE#16#00; //  same as above
        CmdReq[8] :=BYTE#16#10; //  Local port number(=4099)
        CmdReq[9] :=BYTE#16#03; //  same as above
        CmdReq[10]:=BYTE#16#00; // Timeout(= Infinite)
        CmdReq[11]:=BYTE#16#00; // same as above
        CmdReq[12]:=BYTE#16#C0; // Remote IP Addr(=192.168.250.17)
        CmdReq[13]:=BYTE#16#A8; // same as above
        CmdReq[14]:=BYTE#16#FA; // same as above
        CmdReq[15]:=BYTE#16#10; // same as above
        CmdReq[16]:=BYTE#16#10; // Remote TCP Port(=4099)
        CmdReq[17]:=BYTE#16#03; // same as above

        SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
        PortUsing:=FALSE;
        ServerState:=11; // Next state is "Open"

    11: // Open
        IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
            PortUsing:=TRUE;
        END_IF;
        IF (PortUsing=TRUE) THEN
            SendCmd_instance(
                Execute:=TRUE,
                DstNetAdr:=CmdDst,
                CommPort:=_NONE,
                CmdDat:=CmdReq[0],
                CmdSize:=UINT#18,
                RespDat:=CmdResp[0],
                Option:=CmdOption);
        IF SendCmd_instance.Done=TRUE THEN
            ServerState:=12; // Next state is "Opening"
        ELSIF SendCmd_instance.Error=TRUE THEN
            EnableServer:=FALSE;
            ServerState:=0;  // Next state is "Idle"
        END_IF;
    END_IF;

12: // Opening
    IF (J02_Skt4TcpOpening=FALSE) THEN
        IF (J02_Skt4TcpAreaErr=FALSE) AND (J02_Skt4TcpPsvOpenResult[0]=WORD#16#0000) THEN
            // Successfully opened
            ServerState:=13; // Next state is "Opened"
        ELSE // Fail to open
            EnableServer:=FALSE;
            ServerState:=0;  // Next state is "Idle"
        END_IF;
    END_IF;

13: // Opened
    IF (J02_Skt4TcpDatRcved=TRUE) AND (J02_Skt4TcpRcvDatByte > 0) THEN
        ServerState:=30; // Next state is "Prepare Receive"
    END_IF;

30: // Prepare Receive
    CmdDst.UnitNo:=BYTE#16#11; // Unit number(=1)
    CmdReq[0] :=BYTE#16#27; // Command(TCP Socket Receive Request)
    CmdReq[1] :=BYTE#16#12; //  same as above
    CmdReq[2] :=BYTE#16#00; // Socket option
    CmdReq[3] :=BYTE#16#04; // Socket number(=4)
    CmdReq[4] :=BYTE#16#82; // Result storage area(=D3330/J02_Skt4TcpRcvResult)
    CmdReq[5] :=BYTE#16#0D; //  same as above
    CmdReq[6] :=BYTE#16#02; //  same as above
    CmdReq[7] :=BYTE#16#00; //  same as above
    CmdReq[8] :=BYTE#16#00; //  Number of bytes received(=100)
    CmdReq[9] :=BYTE#16#64; //  same as above
    CmdReq[10]:=BYTE#16#00; // Timeout(= Infinity)
    CmdReq[11]:=BYTE#16#00; // same as above

    SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
    PortUsing:=FALSE;
    ServerState:=31; // Next state is "Receive"

31: // Receive
    IF ((_Port_isAvailable=TRUE) AND (PortUsing=FALSE)) THEN
        PortUsing:=TRUE;
    END_IF;
    IF (PortUsing=TRUE) THEN
        SendCmd_instance(
            Execute:=TRUE,
            DstNetAdr:=CmdDst,
            CommPort:=_NONE,
            CmdDat:=CmdReq[0],
            CmdSize:=UINT#12,
            RespDat:=CmdResp[0],
            Option:=CmdOption);

        IF SendCmd_instance.Done=TRUE THEN
```

```
            ServerState:=32; // Next state is "Receiving"
        ELSIF SendCmd_instance.Error=TRUE THEN
            ServerState:=40; // Next state is "Prepare Close"
        END_IF;
    END_IF;

32: // Receiving
    IF (J02_Skt4TcpRcving=FALSE) THEN
        IF (J02_Skt4TcpAreaErr=FALSE) AND (J02_Skt4TcpRcvResult[0]=WORD#16#0000) THEN
            // Successfully received. You can process received data here.
            AryMove(J02_Skt4TcpRcvResult[2], RcvDat[0], UINT#50);
            ServerState:=13; // Next state is "Opened"
        ELSE // Fail to receive
            ServerState:=40; // Next state is "Prepare Close"
        END_IF;
    END_IF;

40: // Prepare Close
    IF (J02_Skt4TcpOpened=TRUE) THEN
        CmdDst.UnitNo:=BYTE#16#11; // Unit number(=1)
        CmdReq[0]:=BYTE#16#27; // Command(TCP Socket Close Request)
        CmdReq[1]:=BYTE#16#14; //  same as above
        CmdReq[2]:=BYTE#16#00; // Socket option
        CmdReq[3]:=BYTE#16#04; // Socket number(=4)
        CmdReq[4]:=BYTE#16#82; // Result storage area(=D3310/J02_Skt4TcpCloseResult)
        CmdReq[5]:=BYTE#16#0C; //  same as above
        CmdReq[6]:=BYTE#16#EE; //  same as above
        CmdReq[7]:=BYTE#16#00; //  same as above

        SendCmd_instance(Execute:=FALSE, CmdDat:=CmdReq[0], RespDat:=CmdResp[0]);
        PortUsing:=FALSE;
        ServerState:=41; // Next state is "Close"
    ELSE
        ServerState:=0;  // Next state is "Idle".
    END_IF;

41: // Close
    IF (_Port_isAvailable=TRUE) AND (PortUsing=FALSE) THEN
        PortUsing:=TRUE;
    END_IF;
    IF (PortUsing=TRUE) THEN
        SendCmd_instance(
            Execute:=TRUE,
            DstNetAdr:=CmdDst,
            CommPort:=_NONE,
            CmdDat:=CmdReq[0],
            CmdSize:=UINT#8,
            RespDat:=CmdResp[0],
            Option:=CmdOption);

        IF SendCmd_instance.Done=TRUE THEN
            ServerState:=42; // Next state is "Closing".
        ELSIF SendCmd_instance.Error=TRUE THEN
            EnableServer:=FALSE;
            ServerState:=0;  // Next state is "Idle".
        END_IF;
    END_IF;

42: // Closing
```

```
        IF (J02_Skt4TcpClosing=FALSE) THEN
            EnableServer:=FALSE;
            ServerState:=0; // Next state is "Idle"
        END_IF;
END_CASE;
```

# A-10 Version Information

This appendix describes the changes in and additions to functional specifications that were made for changes in the unit version of the CPU Units.

### ● Changes in and Additions to Functional Specifications

The following table shows the unit version of the CPU Units and the version of the Sysmac Studio that correspond to changes in or additions to the functional specifications.

| Item | | Change or addition | Reference | Unit version | Sysmac Studio version |
|---|---|---|---|---|---|
| Offsets for structure members | CJ | Addition | A-61 | 1.02 | 1.03 |
| Mounting the CJ1W-EIP21S | | Addition | --- | 1.67 | 1.60 |

I

# Index

# Index

I

**OMRON Corporation** **Industrial Automation Company**

**Kyoto, JAPAN** **Contact : www.ia.omron.com**

***Regional Headquarters***

**OMRON EUROPE B.V.**
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31) 2356-81-300 Fax: (31) 2356-81-388

**OMRON ASIA PACIFIC PTE. LTD.**
438B Alexandra Road, #08-01/02 Alexandra
Technopark, Singapore 119968
Tel: (65) 6835-3011 Fax: (65) 6835-3011

**OMRON ELECTRONICS LLC**
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900 Fax: (1) 847-843-7787

**OMRON (CHINA) CO., LTD.**
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-6023-0333 Fax: (86) 21-5037-2388

**Authorized Distributor:**

**Cat. No. W495-E1-12** 1024