

Programmable Controller

MELSEC iQ-R
series

MELSEC iQ-R OPC UA Server Module User's Manual (Information Model)

-RD81OPC96
-SW1DND-ROPCUA-E (MX OPC UA Module Configurator-R)

SAFETY PRECAUTIONS

(Read these precautions before using this product.)

Before using this product, please read this manual and the relevant manuals carefully and pay full attention to safety to handle the product correctly. If the equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

The precautions given in this manual are concerned with this product only. For the safety precautions of the programmable controller system, refer to the MELSEC iQ-R Module Configuration Manual.

In this manual, the safety precautions are classified into two levels: "⚠ WARNING" and "⚠ CAUTION".

 WARNING	Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.
 CAUTION	Indicates that incorrect handling may cause hazardous conditions, resulting in minor or moderate injury or property damage.

Under some circumstances, failure to observe the precautions given under "⚠ CAUTION" may lead to serious consequences.

Observe the precautions of both levels because they are important for personal and system safety.

Make sure that the end users read this manual and then keep the manual in a safe place for future reference.

[Design Precautions]

WARNING

- Configure safety circuits external to the programmable controller to ensure that the entire system operates safely even when a fault occurs in the external power supply or the programmable controller. Failure to do so may result in an accident due to an incorrect output or malfunction.
 - (1) Emergency stop circuits, protection circuits, and protective interlock circuits for conflicting operations (such as forward/reverse rotations or upper/lower limit positioning) must be configured external to the programmable controller.
 - (2) When the programmable controller detects an abnormal condition, it stops the operation and all outputs are:
 - Turned off if the overcurrent or overvoltage protection of the power supply module is activated.
 - Held or turned off according to the parameter setting if the self-diagnostic function of the CPU module detects an error such as a watchdog timer error.
 - (3) All outputs may be turned on if an error occurs in a part, such as an I/O control part, where the CPU module cannot detect any error. To ensure safety operation in such a case, provide a safety mechanism or a fail-safe circuit external to the programmable controller. For a fail-safe circuit example, refer to the MELSEC iQ-R Module Configuration Manual.
 - (4) Outputs may remain on or off due to a failure of a component such as a relay and transistor in an output circuit. Configure an external circuit for monitoring output signals that could cause a serious accident.
 - In an output circuit, when a load current exceeding the rated current or an overcurrent caused by a load short-circuit flows for a long time, it may cause smoke and fire. To prevent this, configure an external safety circuit, such as a fuse.
 - Configure a circuit so that the programmable controller is turned on first and then the external power supply. If the external power supply is turned on first, an accident may occur due to an incorrect output or malfunction.
 - Configure a circuit so that the external power supply is turned off first and then the programmable controller. If the programmable controller is turned off first, an accident may occur due to an incorrect output or malfunction.
 - For the operating status of each station after a communication failure, refer to manuals for the network used. For the manuals, please consult your local Mitsubishi representative. Incorrect output or malfunction due to a communication failure may result in an accident.
 - When connecting an external device with a CPU module or intelligent function module to modify data of a running programmable controller, configure an interlock circuit in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification, parameter change, forced output, or operating status change) of a running programmable controller, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Improper operation may damage machines or cause accidents. When a Safety CPU is used, data cannot be modified while the Safety CPU is in SAFETY MODE.
 - Especially, when a remote programmable controller is controlled by an external device, immediate action cannot be taken if a problem occurs in the programmable controller due to a communication failure. To prevent this, configure an interlock circuit in the program, and determine corrective actions to be taken between the external device and CPU module in case of a communication failure.
-

[Design Precautions]

WARNING

- Do not write any data to the "system area" and "write-protect area" of the buffer memory in the module. Also, do not use any "use prohibited" signals as an output signal from the CPU module to each module. Doing so may cause malfunction of the programmable controller system. For the "system area", "write-protect area", and the "use prohibited" signals, refer to the user's manual for the module used. For areas used for safety communications, they are protected from being written by users, and thus safety communications failure caused by data writing does not occur.
 - If a communication cable is disconnected, the network may be unstable, resulting in a communication failure of multiple stations. Configure an interlock circuit in the program to ensure that the entire system will always operate safely even if communications fail. Incorrect output or malfunction due to a communication failure may result in an accident. When safety communications are used, an interlock by the safety station interlock function protects the system from an incorrect output or malfunction.
-

[Design Precautions]

CAUTION

- Do not install the control lines or communication cables together with the main circuit lines or power cables. Doing so may result in malfunction due to electromagnetic interference. Keep a distance of 100mm or more between those cables.
 - During control of an inductive load such as a lamp, heater, or solenoid valve, a large current (approximately ten times greater than normal) may flow when the output is turned from off to on. Therefore, use a module that has a sufficient current rating.
 - After the CPU module is powered on or is reset, the time taken to enter the RUN status varies depending on the system configuration, parameter settings, and/or program size. Design circuits so that the entire system will always operate safely, regardless of the time.
 - Do not power off the programmable controller or reset the CPU module while the settings are being written. Doing so will make the data in the flash ROM and SD memory card undefined. The values need to be set in the buffer memory and written to the flash ROM and SD memory card again. Doing so also may cause malfunction or failure of the module.
 - When changing the operating status of the CPU module from external devices (such as the remote RUN/STOP functions), select "Do Not Open by Program" for "Opening Method" of "Module Parameter". If "Open by Program" is selected, an execution of the remote STOP function causes the communication line to close. Consequently, the CPU module cannot reopen the line, and external devices cannot execute the remote RUN function.
-

[Security Precautions]

WARNING

- To maintain the security (confidentiality, integrity, and availability) of the programmable controller and the system against unauthorized access, denial-of-service (DoS) attacks, computer viruses, and other cyberattacks from external devices via the network, take appropriate measures such as firewalls, virtual private networks (VPNs), and antivirus solutions.
-

[Installation Precautions]

WARNING

- Shut off the external power supply (all phases) used in the system before mounting or removing the module. Failure to do so may result in electric shock or cause the module to fail or malfunction.
-

[Installation Precautions]

CAUTION

- Use the programmable controller in an environment that meets the general specifications in the MELSEC iQ-R Module Configuration Manual. Failure to do so may result in electric shock, fire, malfunction, or damage to or deterioration of the product.
 - To mount a module, place the concave part(s) located at the bottom onto the guide(s) of the base unit, and push in the module until the hook(s) located at the top snaps into place. Incorrect interconnection may cause malfunction, failure, or drop of the module.
 - To mount a module with no module fixing hook, place the concave part(s) located at the bottom onto the guide(s) of the base unit, push in the module, and fix it with screw(s). Incorrect interconnection may cause malfunction, failure, or drop of the module.
 - When using the programmable controller in an environment of frequent vibrations, fix the module with a screw.
 - Tighten the screws within the specified torque range. Undertightening can cause drop of the component or wire, short circuit, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction. For the specified torque range, refer to the MELSEC iQ-R Module Configuration Manual.
 - When using an extension cable, connect it to the extension cable connector of the base unit securely. Check the connection for looseness. Poor contact may cause malfunction.
 - When using an SD memory card, fully insert it into the SD memory card slot. Check that it is inserted completely. Poor contact may cause malfunction.
 - Securely insert an extended SRAM cassette or a battery-less option cassette into the cassette connector of the CPU module. After insertion, close the cassette cover and check that the cassette is inserted completely. Poor contact may cause malfunction.
 - Beware that the module could be very hot while power is on and immediately after power-off.
 - Do not directly touch any conductive parts and electronic components of the module, SD memory card, extended SRAM cassette, battery-less option cassette, or connector. Doing so can cause malfunction or failure of the module.
-

[Wiring Precautions]

WARNING

- Shut off the external power supply (all phases) used in the system before installation and wiring. Failure to do so may result in electric shock or cause the module to fail or malfunction.
- After installation and wiring, attach a blank cover module (RG60) to each empty slot before powering on the system for operation. Also, attach an extension connector protective cover^{*1} to each unused extension cable connector as necessary. Directly touching any conductive parts of the connectors while power is on may result in electric shock.

^{*1} For details, please consult your local Mitsubishi Electric representative.

[Wiring Precautions]

CAUTION

- Individually ground the FG and LG terminals of the programmable controller with a ground resistance of 100 ohms or less. Failure to do so may result in electric shock or malfunction.
 - Use applicable solderless terminals and tighten them within the specified torque range. If any spade solderless terminal is used, it may be disconnected when the terminal screw comes loose, resulting in failure.
 - Check the rated voltage and signal layout before wiring to the module, and connect the cables correctly. Connecting a power supply with a different voltage rating or incorrect wiring may cause fire or failure.
 - Connectors for external devices must be crimped or pressed with the tool specified by the manufacturer, or must be correctly soldered. Incomplete connections may cause short circuit, fire, or malfunction.
 - Securely connect the connector to the module. Poor contact may cause malfunction.
 - Do not install the control lines or communication cables together with the main circuit lines or power cables. Doing so may result in malfunction due to noise. Keep a distance of 100mm or more between those cables.
 - Place the cables in a duct or clamp them. If not, dangling cables may swing or inadvertently be pulled, resulting in malfunction or damage to modules or cables.
In addition, the weight of the cables may put stress on modules in an environment of strong vibrations and shocks.
Do not clamp the extension cables with the jacket stripped. Doing so may change the characteristics of the cables, resulting in malfunction.
 - Check the interface type and correctly connect the cable. Incorrect wiring (connecting the cable to an incorrect interface) may cause failure of the module and external device.
 - Tighten the terminal screws or connector screws within the specified torque range. Undertightening can cause drop of the screw, short circuit, fire, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, fire, or malfunction.
 - When disconnecting the cable from the module, do not pull the cable by the cable part. For the cable with connector, hold the connector part of the cable. For the cable connected to the terminal block, loosen the terminal screw. Pulling the cable connected to the module may result in malfunction or damage to the module or cable.
 - When wiring, shield the module using a cover or similar means to prevent foreign matter such as dust or wire chips from entering the module. Such foreign matter can cause a fire, failure, or malfunction. Remove the shield for heat dissipation before operating the system.
 - When a protective film is attached to the top of the module, remove it before system operation. If not, inadequate heat dissipation of the module may cause a fire, failure, or malfunction.
-

[Wiring Precautions]

CAUTION

- Programmable controllers must be installed in control panels. Connect the main power supply to the power supply module in the control panel through a relay terminal block. Wiring and replacement of a power supply module must be performed by qualified maintenance personnel with knowledge of protection against electric shock. For wiring, refer to the MELSEC iQ-R Module Configuration Manual.
 - For Ethernet cables to be used in the system, select the ones that meet the specifications in the user's manual for the module used. If not, normal data transmission is not guaranteed.
-

[Startup and Maintenance Precautions]

WARNING

- Do not touch any terminal while power is on. Doing so will cause electric shock or malfunction.
 - Correctly connect the battery connector. Do not charge, disassemble, heat, short-circuit, solder, or throw the battery into the fire. Also, do not expose it to liquid or strong shock. Doing so will cause the battery to produce heat, explode, ignite, or leak, resulting in injury and fire.
 - Shut off the external power supply (all phases) used in the system before cleaning the module or retightening the terminal screws, connector screws, or module fixing screws. Failure to do so may result in electric shock.
-

[Startup and Maintenance Precautions]

CAUTION

- When connecting an external device with a CPU module or intelligent function module to modify data of a running programmable controller, configure an interlock circuit in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification, parameter change, forced output, or operating status change) of a running programmable controller, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Improper operation may damage machines or cause accidents.
 - Especially, when a remote programmable controller is controlled by an external device, immediate action cannot be taken if a problem occurs in the programmable controller due to a communication failure. To prevent this, configure an interlock circuit in the program, and determine corrective actions to be taken between the external device and CPU module in case of a communication failure.
 - Do not disassemble or modify the modules. Doing so may cause failure, malfunction, injury, or a fire.
 - Use any radio communication device such as a cellular phone or PHS (Personal Handy-phone System) 25cm or more away in all directions from the programmable controller. Failure to do so may cause malfunction.
 - Shut off the external power supply (all phases) used in the system before mounting or removing the module. Failure to do so may cause the module to fail or malfunction.
 - Tighten the screws within the specified torque range. Undertightening can cause drop of the component or wire, short circuit, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
 - After the first use of the product, do not perform each of the following operations more than 50 times (IEC 61131-2/JIS B 3502 compliant).
Exceeding the limit may cause malfunction.
 - Mounting/removing the module to/from the base unit
 - Inserting/removing the extended SRAM cassette or battery-less option cassette to/from the CPU module
 - Mounting/removing the terminal block to/from the module
 - Connecting/disconnecting the extension cable to/from the base unit
 - After the first use of the product, do not insert/remove the SD memory card to/from the CPU module more than 500 times. Exceeding the limit may cause malfunction.
 - Do not touch the metal terminals on the back side of the SD memory card. Doing so may cause malfunction or failure of the module.
 - Do not touch the integrated circuits on the circuit board of an extended SRAM cassette or a battery-less option cassette. Doing so may cause malfunction or failure of the module.
 - Do not drop or apply shock to the battery to be installed in the module. Doing so may damage the battery, causing the battery fluid to leak inside the battery. If the battery is dropped or any shock is applied to it, dispose of it without using.
 - Startup and maintenance of a control panel must be performed by qualified maintenance personnel with knowledge of protection against electric shock. Lock the control panel so that only qualified maintenance personnel can operate it.
-

[Startup and Maintenance Precautions]

CAUTION

- Before handling the module, touch a conducting object such as a grounded metal to discharge the static electricity from the human body. Wearing a grounded antistatic wrist strap is recommended. Failure to discharge the static electricity may cause the module to fail or malfunction.
 - After unpacking, eliminate static electricity from the module to prevent electrostatic discharge from affecting the module. If an electrostatically charged module comes in contact with a grounded metal object, a sudden electrostatic discharge of the module may cause failure.
For details on how to eliminate static electricity from the module, refer to the following.
Safety Precautions For Static Electricity Before Using MELSEC MX Controller MX-R Model and MELSEC iQ-R Series Models (FA-A-0368)
 - Use a clean and dry cloth to wipe off dirt on the module.
-

[Operating Precautions]

CAUTION

- When changing data and operating status, and modifying program of the running programmable controller from an external device such as a personal computer connected to an intelligent function module, read relevant manuals carefully and ensure the safety before operation. Incorrect change or modification may cause system malfunction, damage to the machines, or accidents.
 - Do not power off the programmable controller or reset the CPU module while the setting values in the buffer memory are being written to the flash ROM in the module. Doing so will make the data in the flash ROM and SD memory card undefined. The values need to be set in the buffer memory and written to the flash ROM and SD memory card again. Doing so can cause malfunction or failure of the module.
-

[Disposal Precautions]

CAUTION

- When disposing of this product, treat it as industrial waste.
 - When disposing of batteries, separate them from other wastes according to the local regulations. For details on battery regulations in EU member states, refer to the MELSEC iQ-R Module Configuration Manual.
-

[Transportation Precautions]

CAUTION

- When transporting lithium batteries, follow the transportation regulations. For details on the regulated models, refer to the MELSEC iQ-R Module Configuration Manual.
 - The halogens (such as fluorine, chlorine, bromine, and iodine), which are contained in a fumigant used for disinfection and pest control of wood packaging materials, may cause failure of the product. Prevent the entry of fumigant residues into the product or consider other methods (such as heat treatment) instead of fumigation. The disinfection and pest control measures must be applied to unprocessed raw wood.
-

CONDITIONS OF USE FOR THE PRODUCT

- (1) MELSEC programmable controller ("the PRODUCT") shall be used in conditions;
- i) where any problem, fault or failure occurring in the PRODUCT, if any, shall not lead to any major or serious accident; and
 - ii) where the backup and fail-safe function are systematically or automatically provided outside of the PRODUCT for the case of any problem, fault or failure occurring in the PRODUCT.
- (2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries. MITSUBISHI ELECTRIC SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI ELECTRIC USER'S, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.
- ("Prohibited Application")
- Prohibited Applications include, but not limited to, the use of the PRODUCT in;
- Nuclear Power Plants and any other power plants operated by Power companies, and/or any other cases in which the public could be affected if any problem or fault occurs in the PRODUCT.
 - Railway companies or Public service purposes, and/or any other cases in which establishment of a special quality assurance system is required by the Purchaser or End User.
 - Aircraft or Aerospace, Medical applications, Train equipment, transport equipment such as Elevator and Escalator, Incineration and Fuel devices, Vehicles, Manned transportation, Equipment for Recreation and Amusement, and Safety devices, handling of Nuclear or Hazardous Materials or Chemicals, Mining and Drilling, and/or other applications where there is a significant risk of injury to the public or property.
- Notwithstanding the above restrictions, Mitsubishi Electric may in its sole discretion, authorize use of the PRODUCT in one or more of the Prohibited Applications, provided that the usage of the PRODUCT is limited only for the specific applications agreed to by Mitsubishi Electric and provided further that no special quality assurance or fail-safe, redundant or other safety features which exceed the general specifications of the PRODUCTS are required. For details, please contact the Mitsubishi Electric representative in your region.
- (3) Mitsubishi Electric shall have no responsibility or liability for any problems involving programmable controller trouble and system trouble caused by DoS attacks, unauthorized access, computer viruses, and other cyberattacks.

INTRODUCTION

Thank you for purchasing the Mitsubishi Electric MELSEC iQ-R series programmable controllers.

MX OPC UA Module Configurator-R has the tag mode and information model mode. For the overview of each mode, refer to the following.

 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

This manual describes the contents for using the information model mode.

For those for using the tag mode and common to each mode, read the following manuals.

 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

Before using this product, please read this manual and the relevant manuals carefully and develop familiarity with the functions and performance of the MELSEC iQ-R series programmable controller to handle the product correctly.

Note that the menu names and operating procedures may differ depending on an operating system in use and its version.

When reading this manual, replace the names and procedures with the applicable ones as necessary.

Please make sure that the end users read this manual.

Relevant product

RD81OPC96

COMPLIANCE WITH EMC AND LOW VOLTAGE DIRECTIVES

Method of ensuring compliance

To ensure that Mitsubishi Electric programmable controllers maintain the EMC and Low Voltage Directives or other regulations when incorporated into other machinery or equipment, certain measures may be necessary. Please refer to one of the following manuals.

-  MELSEC iQ-R Module Configuration Manual (SH-081262ENG)
-  Safety Guidelines (IB-0800525)

Certification marks on the side of the programmable controller indicate compliance with the relevant regulations.

Additional measures

To ensure that this product maintains the EMC and Low Voltage Directives or other regulations, please refer to the following.

-  MELSEC iQ-R Module Configuration Manual (SH-081262ENG)
-  Safety Guidelines (IB-0800525)

CONTENTS

SAFETY PRECAUTIONS	1
CONDITIONS OF USE FOR THE PRODUCT	9
INTRODUCTION	10
COMPLIANCE WITH EMC AND LOW VOLTAGE DIRECTIVES	10
RELEVANT MANUALS	14
TERMS	15
GENERIC TERMS AND ABBREVIATIONS	15
CHAPTER 1 PART NAMES	16
CHAPTER 2 SPECIFICATIONS	18
2.1 Performance Specifications	18
Hardware specifications	18
Software specifications	18
2.2 Access Specifications for a CPU Module	22
Accessible CPU modules	22
Accessible routes	22
Accessible devices	22
Access units	22
CHAPTER 3 FUNCTION LIST	23
CHAPTER 4 PROCEDURE FOR OPERATION	25
4.1 Starting an OPC UA Server Module and Configuration Personal Computer	25
Network setting for connection	25
Setting procedure for an OPC UA server module	26
4.2 Configuration Tool	35
4.3 Parameter Setting	40
4.4 SD Memory Card	40
CHAPTER 5 SYSTEM CONFIGURATION	41
5.1 System Configuration	41
Overall system configuration	41
System configuration for the initial setting, maintenance, and inspection	41
System configuration during operation	41
5.2 Connectable Devices	42
5.3 Operating Environment	42
5.4 Considerations for System Configuration	42
5.5 Supported Software Package	42
CHAPTER 6 WIRING	43
CHAPTER 7 INSTALLATION AND UNINSTALLATION	45
CHAPTER 8 FUNCTIONS	47
8.1 Information Model Function	47
Label Node builder	49
Service Set	52

8.2	Data Assignment Function	59
	DataType assignment	60
	Structure assignment	65
	Enumeration assignment	67
	Method assignment	68
	Type conversion check	69
8.3	Data Input/Output Function	75
	Data management	76
	Data	78
	Processing	80
8.4	OPC UA Server Function	92
	Service Set	93
8.5	Operation Function	94
8.6	File Input/Output Function	94
CHAPTER 9 MX OPC UA Module Configurator-R		95
9.1	MX OPC UA Module Configurator-R	95
9.2	Operating Procedure	95
9.3	Project File Handling	96
	Creating a new project	96
	Opening a project	96
	Saving a project	97
9.4	Node Setting	98
	NodeTree	98
	Base Attributes	100
	NodeClass Attributes	101
	Structure Fields	105
	Enumeration Values	105
	References	106
	Importing an information model file	107
	Exporting an information model file	109
9.5	Mapping	110
	Array mapping	113
9.6	Global Label Linking	116
	Importing global labels	116
	Building label Nodes	117
9.7	Namespace	120
	NamespaceArray	120
	RequiredModel	121
9.8	User Setting	122
9.9	Network Setting	125
9.10	Security Setting	126
9.11	Server Certificate Setting	129
9.12	Discovery Server Setting	130
9.13	Optional Operation Setting	131
9.14	Module Control	133
	Connection destination setting	133
	Online data operations	133
	Server operation	133
9.15	Module Diagnostics	134

Information on an OPC UA server module	134
OPC UA server module management	134
SD memory card diagnostics	134
9.16 Electronic Certificate Management	135
User certificate management	135
Application certificate management	135
 CHAPTER 10 TROUBLESHOOTING	 136
10.1 Checking Method for Error Descriptions	136
10.2 Checking the Module Status	136
10.3 Troubleshooting by Symptom	137
Troubleshooting on the configuration tool	137
Troubleshooting on communication with an OPC UA client	138
10.4 Error Code List	139
10.5 Event Code List	139
 APPENDIX	 140
Appendix 1 External Dimensions	140
Appendix 2 Module Labels	141
Appendix 3 Input/Output Signals	142
Appendix 4 Buffer Memory	143
Appendix 5 Data Assignment Availability for Standard DataTypes	144
Appendix 6 Available Characters	147
Available ASCII characters	148
Appendix 7 Processing Time	149
Processing time	149
Startup time	152
Appendix 8 XML File Format	154
XML file specifications	154
Description rules for mapping information	154
Appendix 9 Software Licenses and Copyrights	162
Appendix 10 Use in a Redundant System	164
Appendix 11 Added and Changed Functions	165
 INDEX	 166
REVISIONS	168
WARRANTY	169
INFORMATION AND SERVICES	170
TRADEMARKS	170
COPYRIGHTS	170

RELEVANT MANUALS

The following manuals are relevant to this product.

Manual name [manual number]	Description
MELSEC iQ-R OPC UA Server Module User's Manual (Information Model) [SH-082679ENG] (this manual)	Specifications, procedure for operation, functions, configuration tool, and troubleshooting in the information model mode of an OPC UA server module
MELSEC iQ-R OPC UA Server Module User's Manual (Startup) [SH-081693ENG]	Specifications, procedure for operation, wiring, and operation examples in the tag mode (and common to each mode) of an OPC UA server module
MELSEC iQ-R OPC UA Server Module User's Manual (Application) [SH-081694ENG]	Functions, configuration tool, parameter setting, troubleshooting, input/output signals, and buffer memory in the tag mode (and common to each mode) of an OPC UA server module
GX Works3 Operating Manual [SH-081215ENG]	System configurations, parameter settings, and operation methods for the online function in GX Works3
MELSEC iQ-R Module Configuration Manual [SH-081262ENG]	The combination of the MELSEC iQ-R series modules, common information on the installation/wiring in the system, and specifications of the power supply module, base unit, SD memory card, and battery

This manual does not include detailed information on the following:

- General specifications
- Applicable combinations of CPU modules and the other modules, and the number of mountable modules
- Applicable combinations of remote head modules and the other modules, and the number of mountable modules
- Installation

For details, refer to the following.

 MELSEC iQ-R Module Configuration Manual

TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
Discovery server	A server that manages endpoints of each OPC UA server on a network
Information model mode	A mode that supports information models. Nodes published to an AddressSpace can be imported from an information model file, and data can be set (mapped) for each Node. The Nodes can also be created by importing a GX Works3 project.
Modeling tool	A software application for creating and defining information models for OPC UA
Safety project	A GX Works3 project with user authentication
Tag mode	A mode for setting Nodes (tags or groups) published to an AddressSpace. Tags or groups can be created, and data used for tags can be set.

GENERIC TERMS AND ABBREVIATIONS

Unless otherwise specified, this manual uses the following generic terms and abbreviations.

Generic term/abbreviation	Description
Configuration tool	MX OPC UA Module Configurator-R
OPC UA	OPC Unified Architecture. Platform independent service-oriented architecture that integrates all the functionality of each OPC Classic specification into an extensible framework.
OPC UA server module	A MELSEC iQ-R OPC UA server module

1 PART NAMES

For details on each part name of an OPC UA server module, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

2 SPECIFICATIONS

This chapter explains the specifications of an OPC UA server module.

2.1 Performance Specifications

This section shows the performance specifications of an OPC UA server module.

Hardware specifications

For details, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

Software specifications

OPC UA server specifications

For details on supported OPC UA Profiles, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

Nodes with ns=0 that can be used for this product are published to an AddressSpace of a module.

Before modeling, refer to the AddressSpace of the module from an OPC UA client to check if they can be used.

Note that instances (Objects and Variables) using the following Types are not supported.

Types	NodeId	BrowseName
DataTypes (Structure)	ns=0;i=15618	ConnectionTransportDataType
	ns=0;i=15623	DataSetReaderDataType
	ns=0;i=15629	DataSetReaderMessageDataType
	ns=0;i=15597	DataSetWriterDataType
	ns=0;i=15605	DataSetWriterMessageDataType
	ns=0;i=15334	DataTypeSchemaHeader
	ns=0;i=14524	FieldMetaData
	ns=0;i=15578	PublishedDataSetDataType
	ns=0;i=15530	PubSubConfigurationDataType
	ns=0;i=15617	PubSubConnectionDataType
	ns=0;i=15609	PubSubGroupDataType
	ns=0;i=96	RolePermissionType
	ns=0;i=15630	SubscribedDataSetDataType
	ns=0;i=15616	WriterGroupMessageDataType
VariableType	ns=0;i=2380	ProgramDiagnosticType
ObjectType	ns=0;i=21104	DataSetReaderMessageType
	ns=0;i=15306	DataSetReaderType
	ns=0;i=21096	DataSetWriterMessageType
	ns=0;i=15298	DataSetWriterType
	ns=0;i=14509	PublishedDataSetType
	ns=0;i=17998	WriterGroupMessageType

OPC UA communication specifications

Service	Item	Specification
CreateMonitoredItems ModifyMonitoredItems	SamplingInterval	50 to 60000 [ms] For values that can be specified, refer to the following:  SamplingInterval
	QueueSize	Maximum: 10
	Maximum number of MonitoredItems	Total for all sessions: 15000 ^{*1*2}
CreateSubscription ModifySubscription	PublishingInterval	Minimum: 50 [ms]
	LifetimeCount	Minimum: 10000 ÷ (PublishingInterval value)
	MaxKeepAliveCount	Minimum: 5000 ÷ (PublishingInterval value)
	Maximum number of Subscriptions	Per session: 10 ^{*1} Total for all sessions: 150 ^{*1}
Publish	Maximum number of Notifications per Publish	No limit
CreateSession	Maximum number of sessions	15
	SessionTimeout	10000 to 180000 [ms]
Read	MaxNodesPerRead (maximum number of Nodes per request)	25000 ^{*3}
Write	MaxNodesPerWrite (maximum number of Nodes per request)	3000 ^{*3}
Call	MaxNodesPerMethodCall (maximum number of Methods per request)	1
	Number of InputArguments per request (operation verified)	20
	Number of OutputArguments per request ^{*4} (operation verified)	
Browse	MaxNodesPerBrowse	10000
	MaxNodesPerTranslateBrowsePathsToNodeIds	
Common Request	MaxRequestMessageSize	3 MB (equivalent to the data size of a Read Request for 25000 Nodes if the average is 128 bytes for 1 Node)

*1 May not be used up to the maximum number, because a Subscription is requested and a MonitoredItem is added by default for some OPC UA clients to monitor the connection status with an OPC UA server.

*2 3000 for the firmware version '12'

*3 Even if the upper limit is not exceeded, the StatusCode for Bad_NoCommunication is returned in the following cases:

- The total number of units of mapped data exceeds 25000.
- The total size of mapped data exceeds 2048K words.

*4 The Call of a Method using OutputArguments is not supported.

Information models and mapping specifications

Item		Specification
Information model file	Number of files	10
	File size (total)	20 MB
Namespace	Version information (ns=0)	■ Module Version = 1.04.10, PublicationDate = 2021-09-15T00:00:00Z (UTC) ■ Configuration tool Version = 1.04.11, PublicationDate = 2022-03-29T00:00:00Z (UTC)
	Number of Namespaces	10 ^{*1}
	Number of characters	1 to 134 characters
Node	Number of Nodes	50000
	Number of mappable Variables (total) ^{*2,3}	25000
	Variable data size (total) ^{*3}	2048K words ^{*4}
	Number of Node hierarchies ^{*5}	20
	Number of Structure hierarchies ^{*5}	20
	MaxArrayLength	25000
	Number of mappable array elements	10240 (upper limit of the product of one to three dimensions)
	MaxStringLength	1 KB (number of bytes in UTF-8) ^{*6}
	MaxByteStringLength	10 KB ^{*6}
Mapping	Number of units of mapped data (total)	25000 ^{*7}
	Mapped data size (total)	■ For standard sampling Same as 'Variable data size (total) ^{*8} ■ For high-speed sampling The total number of mapped data points, 32768 points (32K words), is the upper limit. ^{*9,10,11}
	Label name length	800 characters (full path)
	Label name length (total)	2500000 characters ^{*7} (equivalent to an average of 100 characters per label in the maximum number of mappings)

*1 Excluding ns=0 (<http://opcfoundation.org/UA/>) and MELNS (<https://www.mitsubishielectric.com/fa/OPCUA/MELNS/ROPCUA>).

*2 Nodes satisfying the following conditions are targets:

- Under Objects
- Not under Objects > Server
- Not under Objects > Aliases

*3 Excluding InputArguments and OutputArguments of a Method, and Nodes and Handles under a Method.

*4 The size varies depending on the DataType.

- Boolean: 1 word conversion
- String: 256 words conversion
- NodeId: 258 words conversion
- QualifiedName: 257 words conversion
- Others: Conversion of the maximum number of words for the data type of a label that can be assigned (example: 2 words conversion for SByte, 4 words conversion for Float) ( Page 60 Data types of labels)

Note that the Structure follows conversion of the number of words for the DataType of a Structure field.

In addition, the Array follows conversion of the number of words for its DataType, and the product of the number of dimensions and the number of array elements according to the ValueRank and ArrayDimensions. However, when limited assignment is available, the number of dimensions and the number of array elements follow mapping array specification. If not setting mapping array specification, it will be '0.' For details on handling Arrays related to data assignment, refer to the following:

 Page 64 Array handling

*5 Nodes satisfying the following conditions are targets:

- Under Objects
- Not under Objects > Server

*6 If requests exceeding the upper limit are received from an OPC UA client, the error message for Bad_EncodingLimitsExceeded is returned.

*7 Counted as mappings to each member of a structure when mapping data to a NodeId or QualifiedName, as follows:

Data Type	Number of units of mapped data (total)	Label name length (total)	Example of label name length (total)
NodeId	3	Label name length $\times$ 3 + 33 characters	When the label name is label01 (7 characters) $7 \times 3 + 33 = 54$ characters
QualifiedName	2	Label name length $\times$ 2 + 20 characters	When the label name is labelA (6 characters) $6 \times 2 + 20 = 32$ characters

*8 'Variable data size (total)' includes the data size of Variables that cannot be mapped; therefore, the actual mappable data size is smaller than the upper limit of 'Variable data size (total).'

*9 Excluding data mapped to a Variable without the CurrentRead specified for the AccessLevel from the total number.

*10 If the upper limit is exceeded, a number of sequence scan synchronization sampling function points exceeded error (error code: 1941H) occurs in a module. (No error occurs on the configuration tool.)

*11 In high-speed sampling, target data is collected by using the sequence scan synchronization sampling function of a CPU module. When multiple intelligent function modules use this function for the same CPU module, note the total number of points for target data. If it exceeds the number of points that can be collected, an error may occur in registration processing of an intelligent function module registered later. The number of points differs depending on the type of a CPU module. For details, refer to the manual for a CPU module used.

2.2 Access Specifications for a CPU Module

This section shows the access specifications for a CPU module.

Accessible CPU modules

For details, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

Accessible routes

Only the control CPU can be accessed.

Access to other CPUs in the own station or another station is not supported.

Accessible devices

Devices cannot be accessed.

However, if a device is assigned to a global label, it can be accessed.

Devices that can be assigned to global labels are the same as ones accessible in the tag mode.

For details, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

Access units

For details, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

3 FUNCTION LIST

This chapter shows the function list.

For details on the functions, refer to the following:

☞ Page 47 FUNCTIONS

Function	Description
Information model function	To map data to Nodes in order to exchange data between an OPC UA client and OPC UA server module. In addition, Objects and Variables can be created based on label information of global labels imported from a GX Works3 project. An OPC UA server module constructs an AddressSpace according to the setting of the configuration tool, and publishes it to OPC UA clients.
Data assignment function	To assign mapped data to Nodes of an OPC UA server module. A type conversion check is performed as necessary (signed ⇔ unsigned, bit range, character code, etc.). • Structure: Byte string conversion is performed according to the structure of a Field. • Enumeration: Value conversion is performed according to the setting of the configuration tool (value ⇔ value, value ⇔ string).
Data input/output function	To input or output data according to the setting of the configuration tool.
OPC UA server function	To set general items of an OPC UA server and module.
Operation function	To perform various communications between an OPC UA server module and the configuration tool, such as reading, writing, or updating a setting, and managing electronic certificates.
File input/output function	To operate various files related to the information model mode.

MEMO

4 PROCEDURE FOR OPERATION

This chapter shows the procedure for operation for using an OPC UA server module.

4.1 Starting an OPC UA Server Module and Configuration Personal Computer

—: No operation required

Step	OPC UA server module ^{*1}	Configuration personal computer
1	Mount an OPC UA server module on a base unit. ^{*2}	Install the configuration tool. ^{*3}
2	Switch the dot matrix LED display mode switch to the right (SHOW).	—
3	Connect the OPC UA server module to a configuration personal computer on a 1:1 basis. ^{*4}	
4	Turn the power of the system ON. (The ERR LED flashes because an SD memory card is not inserted.)	—
5	Insert an SD memory card. ^{*5}	—
6	—	Start the configuration tool (information model mode).
7	—	Select "Direct Connection" in the "Target Setting" screen. ^{*6}
8	—	Format the SD memory card as necessary.
9	—	Configure the setting for the OPC UA server module.
10	—	Write the setting to the OPC UA server module.
11	—	Update the setting of the OPC UA server module.
12	Connect the OPC UA server module and configuration personal computer to a network according to the network setting configured in the configuration tool. ^{*7}	
13	Start the operation.	

*1 Perform the self-diagnostic test as necessary. For details, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

*2 Do not use an electric screwdriver to attach and remove module fixing screws.

*3 For details on the procedure, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

*4 For details on the system configuration for the initial setting, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

*5 For details on the procedure, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

*6 Specify the following (default user setting) for connection in the "Target Setting" screen of the configuration tool.

- Use the User Authentication: Select the checkbox.

- User Name: RD81OPC96

- Password: MITSUBISHI

*7 For details on the network setting for connection, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

Network setting for connection

For details on the network setting for connecting an OPC UA server module to a configuration personal computer via a hub, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

Setting procedure for an OPC UA server module

The setting for an OPC UA server module can be configured in the configuration tool.

The following shows an operating procedure example for writing a project file set in the configuration tool to an OPC UA server module.

Operating procedure

1. Set Nodes published to an AddressSpace. The setting method differs depending on whether to use an information model. (☞ Page 26 Node setting (when not using an information model), Page 30 Node setting (when using an information model))
2. Set the network for an OPC UA server module. (☞ Page 32 Network setting)
3. Set the security for OPC UA communication. (☞ Page 32 Security setting)
4. Write the setting to the OPC UA server module. (☞ Page 33 Writing the setting)
5. Store an application certificate of an OPC UA client in the OPC UA server module.
Also, store a server certificate of the OPC UA server module in the OPC UA client. (☞ Page 33 Storing an application certificate)

Node setting (when not using an information model)

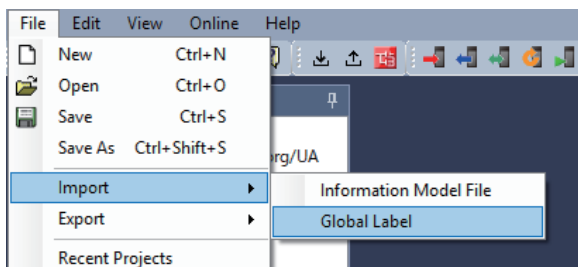
The following shows the procedure for importing global labels to import label information from a GX Works3 project and using the label Node builder function to create Nodes published to an AddressSpace from the imported label information.

For details on the label Node builder function, refer to the following:

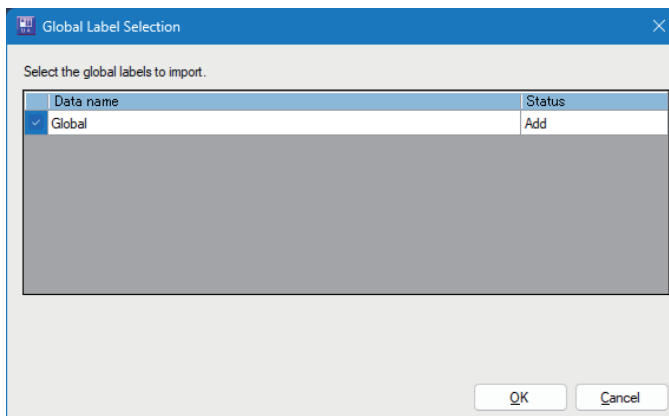
☞ Page 49 Label Node builder

Operating procedure

1. Start the configuration tool, and select [File] ⇒ [Import] ⇒ [Global Label].



2. Select a GX Works3 project to import label information, and click the [Open] button.
3. Select setting data to import, and click the [OK] button.



4. Check the import result, and click the [OK] button.

5. In the [Label] tab in the bottom window, select the checkbox of an item to create a Node from the imported label information.

For details on the method for creating Nodes, refer to the following:

(☞ Page 118 Methods for creating, deleting, and updating label Nodes)

Label			
Data Name		Label Name	Data Type
Global Label		☒ labelA	Word [Unsigned]/Bit String [16-bit]
☒ Global		☐ labelB	FLOAT [Single Precision]
Structure		☒ labelC	Timer
☐ Struct_1		☐ labelD	Pointer
☒ Struct_2		☐ ST_1	Struct_1
☐ Struct_3		☒ ST_2	Struct_2
		☐ ST_3	Struct_3
		☒ Element	Bit(0..1)

6. Select "Information Model" in the navigation tree to open the NodeTree screen, and check the created Node.

Information Model	
NodeTree	Mapping
0:Root	
0:Objects	
Global	
labelA	labelA
ST_2	
Element	
0:Aliases	

7. Create Variables for structure members and Variables for array elements as necessary. (☞ Page 28 Creating Variables for structure members, Page 29 Creating Variables for array elements)

■Creating Variables for structure members

The following shows the procedure for creating Variables for structure members when creating a Node from a structure type global label.

Operating procedure

1. Select "Optional Operation Setting" in the navigation tree.
2. Set "Variables for Structure Fields" to "Create" under "Label Node Setting," and click the [Apply] button.

The screenshot shows the 'Optional Operation Setting' dialog box. It has a table with two columns: 'Item' and 'Setting'. The 'Item' column lists 'Data Sampling Setting', 'Label Node Setting', and 'Variables for Structure Fields'. The 'Setting' column shows 'Standard sampling' for 'Data Sampling Setting', 'Do not create' for 'Label Node Setting', and 'Create' for 'Variables for Structure Fields'. Below the table is a 'Description' section with text explaining the settings. At the bottom right is an 'Apply' button.

Item	Setting
Data Sampling Setting	
Sampling Mode	Standard sampling
Label Node Setting	
Variables for Array Elements	Do not create
Variables for Structure Fields	Create

Description
Set whether to "Create" or "Do not create" Variables for structure fields.
When set to "Create," Variables for the structure fields will be created under the structure label Variable. (*1)
*1: OPC UA clients can individually specify Variables for structure fields. This minimizes the amount of data transmitted during communication but increases the number of Variables and the size of Variable data.

Apply

3. Select "Information Model" in the navigation tree to open the NodeTree screen, and check the created Variables for structure members.

The screenshot shows the 'Information Model' dialog box. It has two main sections: 'NodeTree' and 'Mapping'. The 'NodeTree' section shows a hierarchical tree structure with nodes like '0:Root', '0:Objects', 'Global', 'labelA', 'ST_2', and 'Element'. The 'Mapping' section shows a table with two columns: 'NodeTree' and 'Mapping'. The table contains the following data:

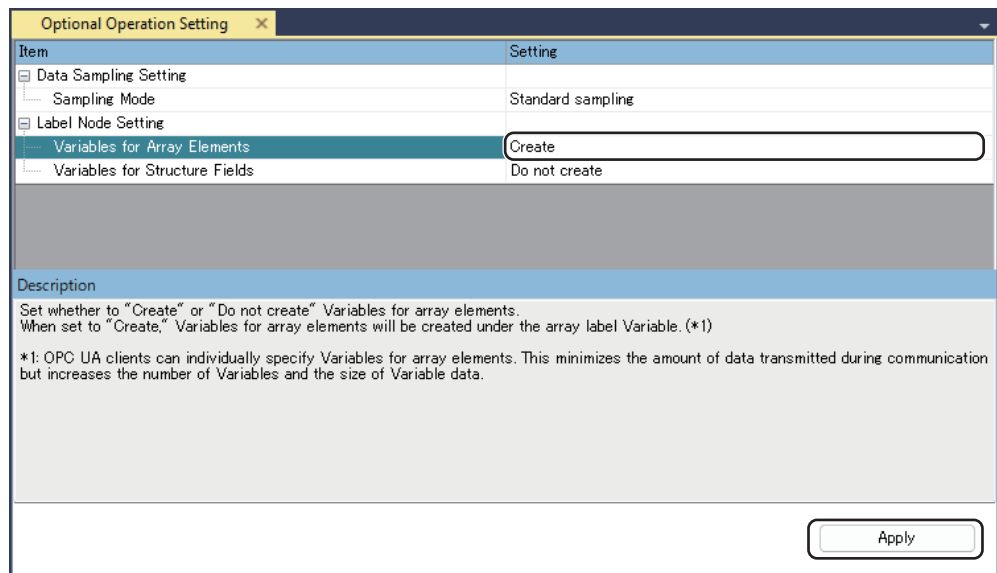
NodeTree	Mapping
0:Root	
0:Objects	
Global	
labelA	labelA
ST_2	
memberA	ST_2.memberA
memberB	ST_2.memberB
memberC	ST_2.memberC
memberA	ST_2.memberA
memberB	ST_2.memberB
memberC	ST_2.memberC
Element	

■Creating Variables for array elements

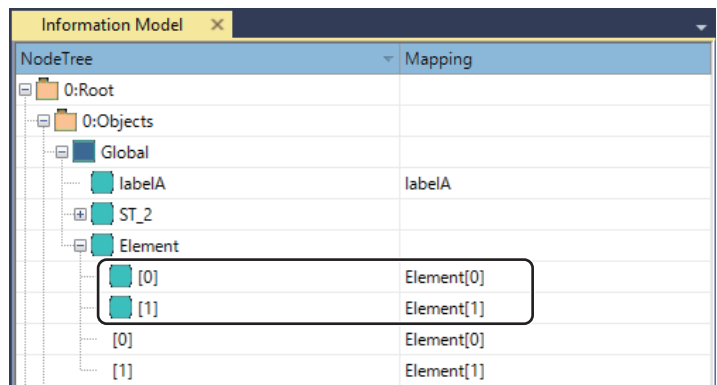
The following shows the procedure for creating Variables for array elements when creating a Node from an array type global label.

Operating procedure

- 1. Select "Optional Operation Setting" in the navigation tree.
- 2. Set "Variables for Array Elements" to "Create" under "Label Node Setting," and click the [Apply] button.



- 3. Select "Information Model" in the navigation tree to open the NodeTree screen, and check the created Variables for array elements.



Node setting (when using an information model)

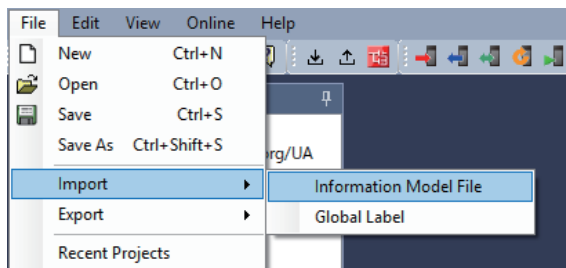
The following shows the procedure for importing an information model file to import Nodes published to an AddressSpace and setting (mapping) data for each Node.

For details on mapping, refer to the following:

☞ Page 110 Mapping

Operating procedure

1. Use a modeling tool to create an information model file.
For the method for using a modeling tool, refer to the manual for the tool used.
2. Start the configuration tool, and select [File] ⇒ [Import] ⇒ [Information Model File].

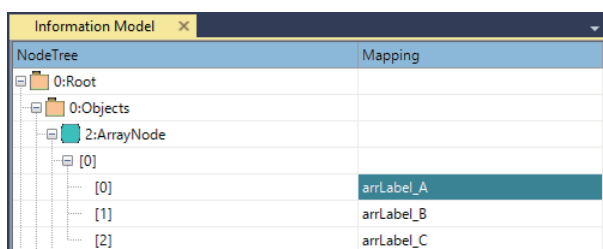


3. Select an information model file to use, and click the [Open] button.
4. Repeat steps 2 and 3 to import all necessary information model files.
(When selecting "Namespaces" in the navigation tree, the imported Namespaces can be checked.)

Point

When using an information model file that references another file, import the referenced information model file (RequiredModel) first. (Example: When using ❷Server_A.xml that references ❶Opc.Ua.PackML.NodeSet2.xml, import ❶ then ❷.)

5. Select "Information Model" in the navigation tree to open the NodeTree screen, and enter the label names of global labels used for a CPU module in the "Mapping" column.



Point

The setting can be configured efficiently by using copy and paste.

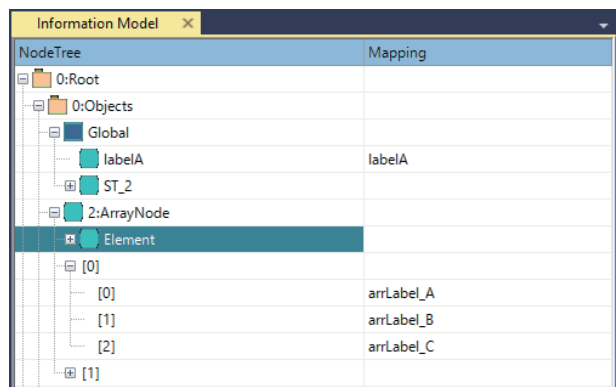
6. Place Nodes created by using the label Node builder function as necessary. (☞ Page 31 Placing Nodes created by using the label Node builder function in an information model)

■Placing Nodes created by using the label Node builder function in an information model

The following shows the procedure for placing Nodes created by using the label Node builder function when importing an information model file.

Operating procedure

- 1. Create Nodes with the same procedure as shown in the following:
(☞ Page 26 Node setting (when not using an information model))
- 2. Select "Information Model" in the navigation tree to open the NodeTree screen, and drag and drop the Nodes created in step 1 onto a move destination Node.



Network setting

Set the network for an OPC UA server module.

Operating procedure

Select "Network" in the navigation tree to open the setting screen, and set the network for an OPC UA server module.

Network

☒ Enable Ethernet port (CH1)

IP Address: 192 . 168 . 3 . 10

Subnet Mask: 255 . 255 . 255 . 0

Port No.: 4840

☐ Enable Ethernet port (CH2)

IP Address:

Subnet Mask:

Default Gateway: Not set

Host Name: RD01OPC96

Apply

End Point URL

Set network of OPC UA server module

Ethernet(CH1)

Ethernet(CH2)

End Point URL: opc.tcp://192.168.3.10:4840



An endpoint URL in the subwindow is used when connecting from an OPC UA client.

Security setting

Set the security for OPC UA communication.

Operating procedure

Select "Security" in the navigation tree to open the setting screen, and set the security for OPC UA communication.

Security

SecureChannel User Authentication Application Authentication

Security Policy

☐ Aes256-Sha256-RsaPss

☐ Aes128-Sha256-RsaOaep

☒ Basic256Sha256

☐ Basic256(Deprecated)

☐ Basic128Rsa15(Deprecated)

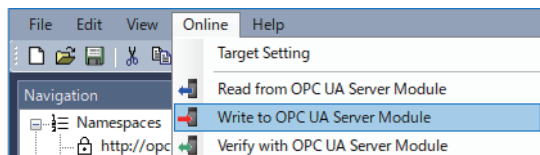
Security Mode: Sign

Writing the setting

Write the setting to the OPC UA server module.

Operating procedure

1. Select [Online] ⇒ [Write to OPC UA Server Module].



Point

Establish connection with the OPC UA server module in the connection destination setting in advance.

4

2. Power OFF and ON or reset the CPU module.

Storing an application certificate

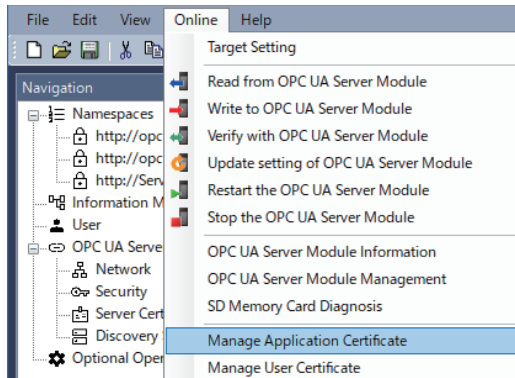
Store an application certificate of an OPC UA client in the OPC UA server module.

Also, store a server certificate of the OPC UA server module in the OPC UA client.

Operating procedure

■ Procedure for storing an application certificate of the OPC UA client in the OPC UA server module

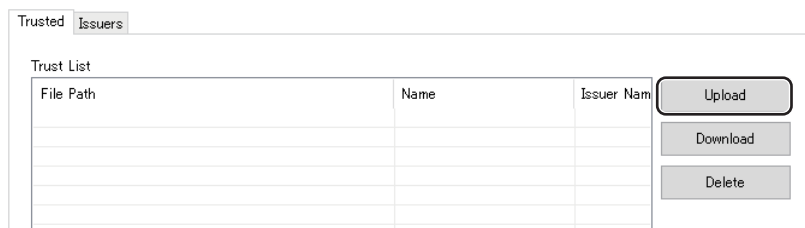
1. Acquire an application certificate of the OPC UA client.
2. Select [Online] ⇒ [Manage Application Certificate].



Point

Establish connection with the OPC UA server module in the connection destination setting in advance.

3. Click the [Upload] button of "Trust List" to store the acquired application certificate in the OPC UA server module.



■ Procedure for storing a server certificate of the OPC UA server module in the OPC UA client

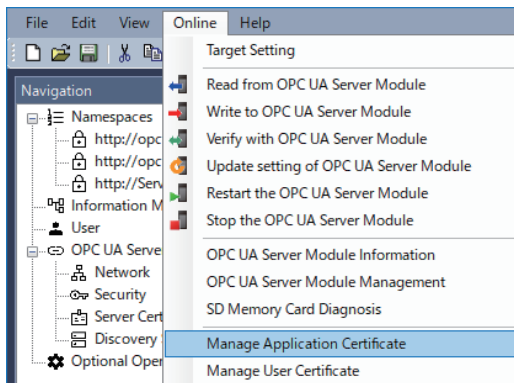
Point

Check if the time in the CPU module is correct.

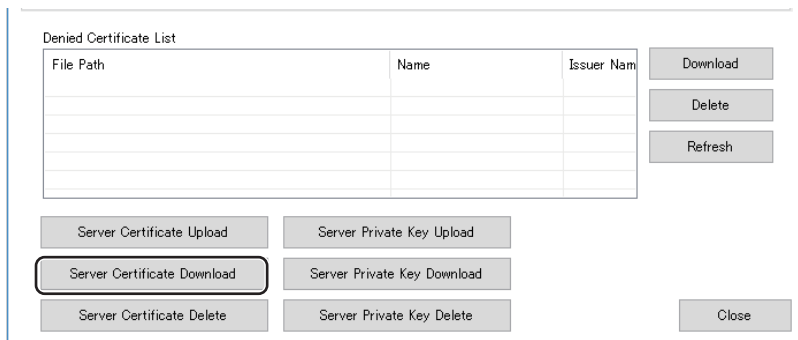
If it is not correct, set it to the correct time then perform the following:

- ❶ Delete a server certificate in the "Manage Application Certificate" screen of the configuration tool.
- ❷ Perform any of the following procedures to stop and restart the operation of the OPC UA server module and regenerate the server certificate.
 - Power-OFF and ON or reset
 - [Online] ⇒ [Update setting of OPC UA Server Module]
 - [Online] ⇒ [Stop the OPC UA Server Module]/[Restart the OPC UA Server Module]

1. Select [Online] ⇒ [Manage Application Certificate].



2. Click the [Server Certificate Download] button to acquire a server certificate of the OPC UA server module.



3. Store the acquired server certificate in the trust list of the OPC UA client.

4.2 Configuration Tool

This section shows the configuration tool (information model mode) for configuring the setting to utilize information models in a programmable controller.

Nodes published to an AddressSpace can be imported from an information model file, and data can be set (mapped) for each Node.

The Nodes can also be created by importing a GX Works3 project.

For details on the configuration tool, refer to the following:

☞ Page 95 MX OPC UA Module Configurator-R

Startup method

Operating procedure

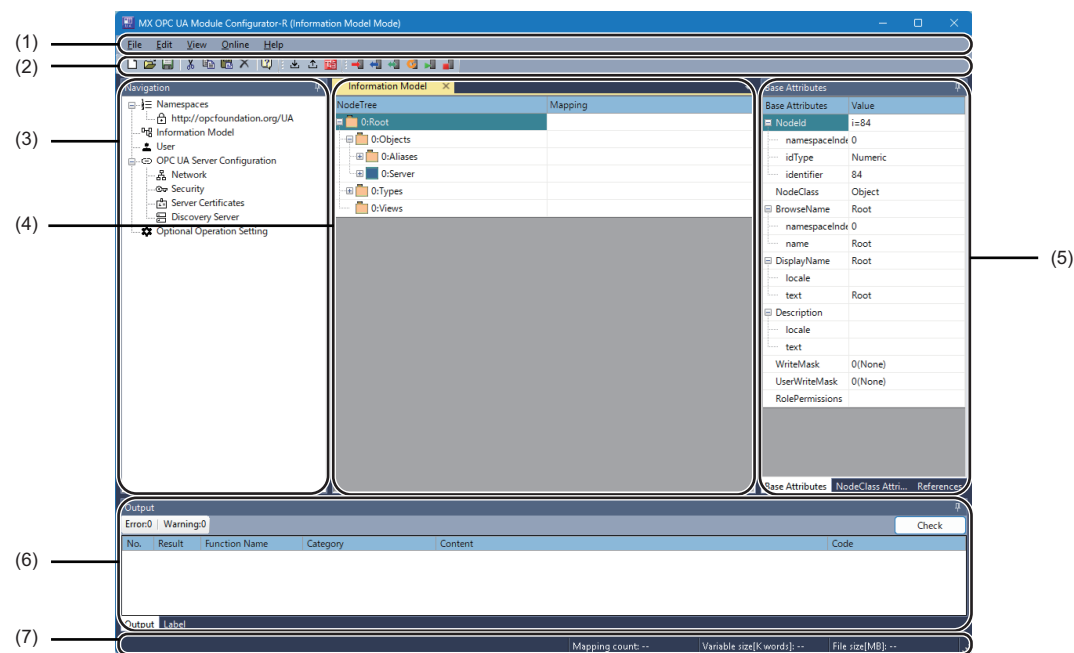
Start MELSEC iQ-R series OPC UA Server Module Configuration Tool (Information Model Mode) from "MELSOFT" in Windows Start.

Display language switching

For details, refer to the following:

☞ MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

Screen configuration



- (1) Menu bar
- (2) Toolbar
- (3) Navigation tree^{*1,2} (☞ Page 37 Navigation tree)
- (4) Main window^{*1,2} (☞ Page 37 Main window/subwindow)
- (5) Subwindow^{*1,2} (☞ Page 37 Main window/subwindow)
- (6) Bottom window^{*1,2} (☞ Page 37 Bottom window - [Output] tab, Page 38 Bottom window - [Label] tab)
- (7) Status bar (☞ Page 39 Status bar)

*1 The window and its tab can be displayed outside the main frame by dragging and dropping.

*2 The display statuses of the windows and table columns, their widths, etc. are retained only while the tool is running.

■Menu configuration

Menu item			Description
File	New		To discard a project being edited and create a new project.
	Open		To open a project file saved in a personal computer.
	Save		To overwrite and save a project being edited to a file.
	Save As		To save a project being edited under a new file name.
	Import	Information Model File	To import a Namespace, Nodes, and mapping information from an XML file.
		Global Label	To import label information used for building label Nodes from a GX Works3 project.
	Export	Information Model File	To output Nodes and mapping information in a specified Namespace in an XML file format.
	Recent Projects	(Recently opened project path)	To select and open a project file used recently in the configuration tool.
	Exit		To end the configuration tool.
Edit	Cut		To cut a selected item and copy it to the clipboard.
	Copy		To copy a selected item to the clipboard.
	Paste		To paste an item on the clipboard.
	Delete		To delete a selected item.
	Select All		To select all items on the list.
View	Toolbar	Standard	To show or hide the buttons for basic functions.
		Information Model	To show or hide the buttons for information models.
		Online	To show or hide the buttons for online operations.
	Status Bar		To show or hide the status bar.
	Docking Window	Navigation	To show or hide the navigation tree.
		Sub Window	To show or hide the subwindow.
		Bottom Window	To show or hide the bottom window.
	Switch Display Language		To switch the display language.
Online	Target Setting		To configure the setting for connecting to an OPC UA server module.
	Read from OPC UA Server Module		To read a setting from an OPC UA server module.
	Write to OPC UA Server Module		To write a setting to an OPC UA server module.
	Verify with OPC UA Server Module		To verify a setting in an OPC UA server module with one in the configuration tool.
	Update setting of OPC UA Server Module		To update the setting of an OPC UA server module.
	Restart the OPC UA Server Module		To restart the operation of an OPC UA server module.
	Stop the OPC UA Server Module		To stop the operation of an OPC UA server module.
	OPC UA Server Module Information		To display the operating status of an OPC UA server module.
	OPC UA Server Module Management		To display information on an OPC UA server module and perform operations to it.
	SD Memory Card Diagnosis		To display the current use status of an SD memory card inserted in an OPC UA server module and format the SD memory card.
	Manage Application Certificate		To manage application certificates in an OPC UA server module.
	Manage User Certificate		To manage user certificates in an OPC UA server module.
Help	Help		To open the user's manuals for an OPC UA server module.
	Version Information		To display the version information of the configuration tool.

■Navigation tree

The functions are displayed in a tree format.

Item		Reference
Namespaces	{Namespace URI}	Page 120 Namespace
Information Model		Page 98 Node Setting
User		Page 122 User Setting
OPC UA Server Configuration	Network	Page 125 Network Setting
	Security	Page 126 Security Setting
	Server Certificates	Page 129 Server Certificate Setting
	Discovery Server	Page 130 Discovery Server Setting
Optional Operation Setting		Page 131 Optional Operation Setting

■Main window/subwindow

The display screen switches according to the selection status in the navigation tree.

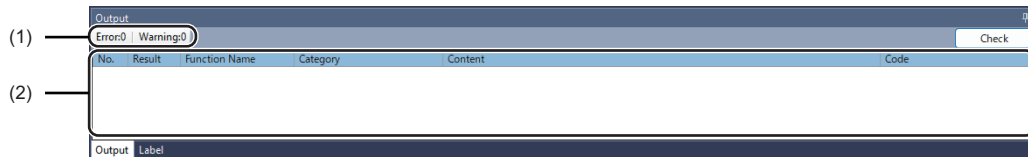
■Bottom window - [Output] tab

Errors and warnings occurred while operating the configuration tool are displayed. Details on their causes and corrective actions can be checked on the tooltip.

Error: When the operation fails

Warning: When an alert is required

Window



Displayed items

Item		Description
(1) Number of occurrences		The numbers of occurrences of errors and warnings are displayed.
[Check] button		Click this to clear the output list and update it to the latest information.
(2) Output list	Result	A result (error or warning) is displayed.
	Function Name	The function name is displayed.
	Category	An operation that causes an error or warning is displayed.
	Content	The description and corrective action are displayed.
	Code	The code is displayed.

■Bottom window - [Label] tab

Label information imported by importing global labels is displayed. ( Page 116 Importing global labels)

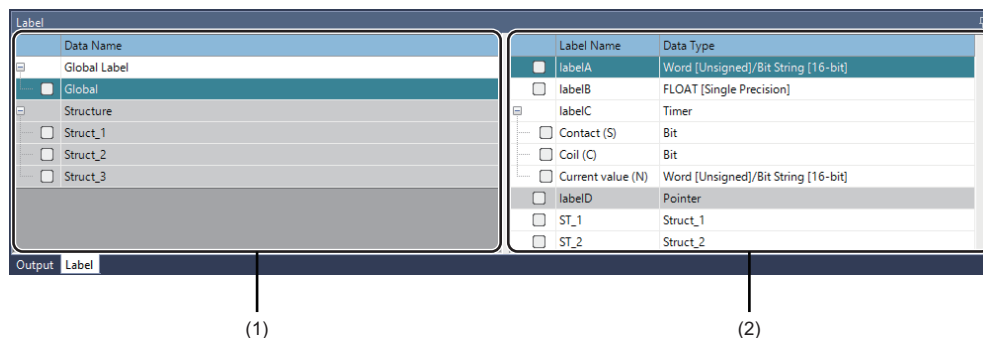
Label Nodes can be built in this screen. For details on the label Node builder function, refer to the following:

 Page 49 Label Node builder

For details on the methods for creating, deleting, and updating label Nodes, refer to the following:

 Page 118 Methods for creating, deleting, and updating label Nodes

Window



Displayed items

Item	Description
(1) Data list	Imported data (global label setting data and structure definitions) is displayed in a tree. A label Node for data with its checkbox selected is created on NodeTree. (The checkbox of a structure definition cannot be selected or unselected manually. *1) By double-clicking a data row in which a label Node is created, the cursor jumps to the corresponding label Node on NodeTree. Global label setting data can be deleted from the data list by pressing the  key. (The corresponding label Nodes (Objects) and their child label Nodes are also deleted.)
(2) Label list	■When global label setting data is selected in the data list Labels of the selected setting data are displayed. *2*3*4*5 A label Node for a label with its checkbox selected is created on NodeTree. *6 By double-clicking a label row in which a label Node is created, the cursor jumps to the corresponding label Node on NodeTree. ■When a structure definition is selected in the data list Members of the selected structure definition are displayed. *2*3*4 Grayed-out structure members are not added to a Field of a Structure.

*1 A label Node (Structure) is automatically created or deleted. For details, refer to the following:

 Page 118 Methods for creating, deleting, and updating label Nodes

*2 For timer or counter labels, their contacts, coils, and current values are displayed as child items in a tree.

*3 Grayed-out labels conflict with the restrictions on global labels.

For details on the restrictions on global labels, refer to the following:

 Page 49 Restrictions on global labels

*4 For structure type global labels, child items of timers and counters, and structure members, "-" is displayed for column items that are not applicable.

*5 The display contents of the column item "Access from External Device" are as follows:

- True: Accessible

- False: Inaccessible

*6 For the methods for creating label Nodes (Fields) and label Nodes (Elements), refer to the following:

 Page 118 Methods for creating, deleting, and updating label Nodes

Point

- Column items not displayed by default (Class, Assign (Device/Label), and Access from External Device) are displayed by right-clicking the column title and selecting the items.
- When right-clicking each row, the operation menus are displayed.

■Status bar

The setting statuses are displayed. Details can be checked on the tooltip.

Item	Description
Mapping count	The number of units of mapped data (total) is displayed. In addition, the label name length (total) can be checked on the tooltip.
Variable size[K words]	The Variable data size (total) is displayed. In addition, the number of mappable Variables can be checked on the tooltip.
File size[MB]	The file size (total) is displayed. The value is updated when saving a project.

If the displayed setting contents exceed their upper limits in the specifications, the character color on the status bar changes.

- Red: When the setting contents displayed on the status bar exceed their upper limits
- Orange: When the setting contents of other items in the tooltip exceed their upper limits

For details on the upper limits, refer to the following:

 Page 20 Information models and mapping specifications

4.3 Parameter Setting

Various operation settings of an OPC UA server module can be configured in the parameter setting of GX Works3.

For details on the procedure for adding an OPC UA server module to GX Works3, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

For details on the module parameter setting, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

4.4 SD Memory Card

For the procedures for inserting and removing an SD memory card, and considerations for using it, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

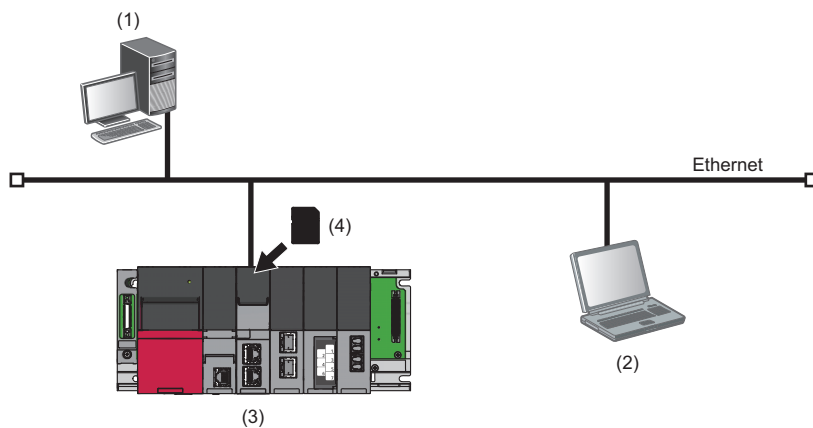
5 SYSTEM CONFIGURATION

This chapter shows the system configuration of an OPC UA server module.

5.1 System Configuration

Overall system configuration

The following figure shows the overall system configuration when using an OPC UA server module.



- (1) OPC UA client
- (2) Configuration personal computer (configuration tool, modeling tool, GX Works3)
- (3) OPC UA server module
- (4) SD memory card

Point

For available CPU modules and the number of mountable ones, refer to the following:

📖 MELSEC iQ-R Module Configuration Manual

System configuration for the initial setting, maintenance, and inspection

For details, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

System configuration during operation

For details, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

5.2 Connectable Devices

For details, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

5.3 Operating Environment

For details, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

5.4 Considerations for System Configuration

Considerations for using a multiple CPU system

When using the information model mode, only the control CPU can be accessed.

Access to other CPUs in the own station is not supported.

Considerations for accessing another station

When using the information model mode, only the control CPU can be accessed.

Access to another station is not supported.

5.5 Supported Software Package

This section shows the software package supported by an OPC UA server module.

Software package	Software version
GX Works3	'1.105K' or later

6 WIRING

For details on wiring an Ethernet cable, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

MEMO

7 INSTALLATION AND UNINSTALLATION

For details on installing and uninstalling MX OPC UA Module Configurator-R, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

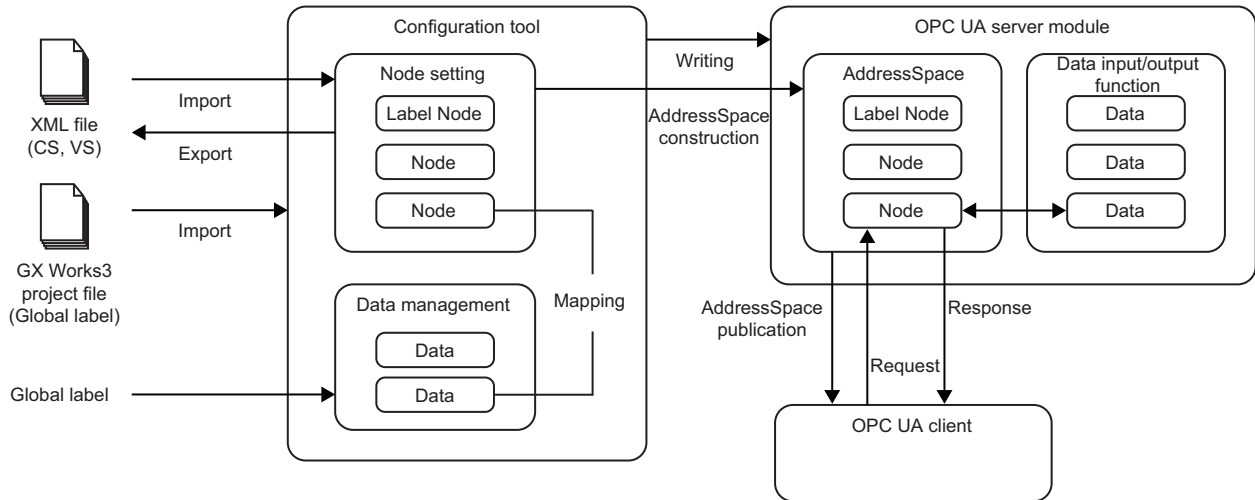
MEMO

8 FUNCTIONS

This chapter explains the details on the functions of an OPC UA server module.

8.1 Information Model Function

The information model function is for exchanging data between an OPC UA client and OPC UA server module.



- To use this function, configure the Node setting in the configuration tool and write it to an OPC UA server module.
- In the Node setting, import an information model file to import Nodes from an XML file in which information models such as CS and VS are described, and map data to the imported Nodes.
- By using the label Node builder function, Nodes can be created from label information imported by importing global labels. These Nodes (label Nodes) are automatically mapped; therefore, manual mapping is not required.
- The OPC UA server module constructs an AddressSpace according to the Node setting and publishes it to an OPC UA client.
- When the OPC UA server module receives a request from the client, it reads and writes data by using the data input/output function and responds to the client.

Function list

The information model function consists of the following items:

Item	Description	Reference
Node setting	To set Nodes published to an AddressSpace.	Page 98 Node Setting
Mapping	To set (map) data for Nodes.	Page 110 Mapping
Global label linking	To create Nodes based on label information of global labels.	Page 49 Label Node builder Page 116 Global Label Linking
Namespace	To display information on a Namespace in a project.	Page 120 Namespace
View Service Set	To provide the functions of the View Service Set of the OPC UA specifications. Nodes in an AddressSpace can be searched for according to a request from an OPC UA client.	Page 52 Service Set
Attribute Service Set	To provide the functions of the Attribute Service Set of the OPC UA specifications. A value can be read from or written to an Attribute of a Node according to a request from an OPC UA client.	
Method Service Set ^{*1}	To provide the functions of the Method Service Set of the OPC UA specifications. A Method can be executed according to a request from an OPC UA client.	
MonitoredItem Service Set	To provide the functions of the MonitoredItem Service Set of the OPC UA specifications. A Node to be monitored for the Subscription Service Set can be registered.	
Subscription Service Set	To provide the functions of the Subscription Service Set of the OPC UA specifications. Changes in Node data can be reported according to a request from an OPC UA client.	

^{*1} Available for an OPC UA server module with firmware version '13' or later and the configuration tool with software version '1.13P' or later.

Label Node builder

This is a function to create Objects and mapped Variables from label information imported by importing global labels. Simply by selecting the checkbox of an item to create a Node in the bottom window ([Label] tab), a Node can be created on NodeTree. ( Page 38 Bottom window - [Label] tab)

- When selecting the checkbox of global label setting data, an Object is created, and when selecting the checkbox of a global label, a Variable is created.
- When creating an Object by selecting the checkbox of global label setting data, then creating a Variable for a global label under the setting data, the Variable is grouped under the corresponding Object.
- A created Object and Variable can be dragged and dropped on NodeTree and moved to a specified position in an information model.
- For a structure type global label, the DataType (Structure) corresponding to the target structure definition is automatically created or deleted according to whether a Variable for the global label is created.
- Whether to create Variables for structure members and Variables for array elements can be set in the optional operation setting. ( Page 131 Optional Operation Setting)

Point

By creating Variables for structure members and Variables for array elements, each Variable can be used individually by OPC UA clients, reducing the amount of data during communication. However, the number of Variables and the Variable data size increase.

For details on the methods for creating, deleting, and updating label Nodes, refer to the following:

 Page 118 Methods for creating, deleting, and updating label Nodes

For an operating procedure example for building label Nodes, refer to the following:

 Page 26 Setting procedure for an OPC UA server module

Restrictions on global labels

The following table shows the restrictions on global labels and the behaviors when a global label conflicts with any restriction.

Item	Restriction	Behavior when a conflict occurs
Class	Only the following can be used: • VAR_GLOBAL • VAR_GLOBAL_RETAIN	The global labels are grayed out in the label list in the bottom window ([Label] tab).
Data type	The following cannot be used: • Pointer type (including a member of a structure) • Function block	The global labels and structure members are grayed out in the label list in the bottom window ([Label] tab). ^{*1*2}
Access from external device	Only the following can be used: • Label with this checkbox selected	The global labels are grayed out in the label list in the bottom window ([Label] tab).
Assign (device/label)	Assignable devices are the same as accessible devices in the tag mode. For details on the accessible devices, refer to the following: ( MELSEC iQ-R OPC UA Server Module User's Manual (Startup))	• Structure members with inaccessible devices assigned^{*3} are not mapped. • Global labels (excluding structure members) with inaccessible devices assigned are grayed out in the label list.
	The following cannot be used: • Structure type label with assignment and structure type members	Structure members with assignments and in the second hierarchy or deeper ^{*3} are not mapped. (Example) • Str1.Member1: Mapped • Str1.Str2.Member2: Not mapped
Full path length	Up to 800 characters	Structure members with full path lengths exceeding the upper limit ^{*3} are not mapped.

^{*1} Grayed-out structure members are not added to a Field of a Structure.

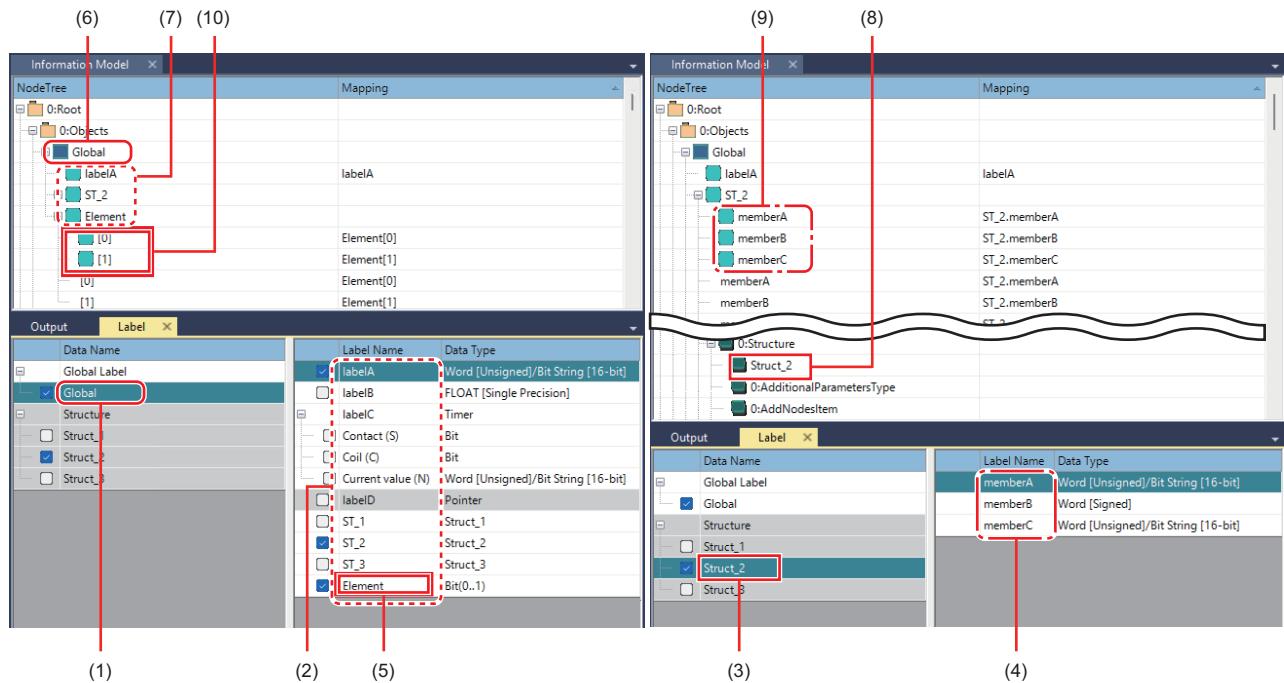
^{*2} If all members of a structure definition conflict with any restriction, the corresponding Structure behaves as a Structure with no Fields.

^{*3} Refers to the applicable Fields under a label Node (Variable) and the applicable label Nodes (Fields) (including Fields in a label Node (Field)).

Label Node types

Nodes created by using this function are collectively referred to as label Nodes. The following table shows the label Node types.

Label Node type	Description
Label Node (Object)	Object corresponding to global label setting data
Label Node (Variable)	Variable corresponding to a global label
Label Node (Structure)	DataType (Structure) corresponding to a structure definition
Label Node (Field)	Variable corresponding to a structure member (Variable corresponding to a Field of a Structure)
Label Node (Element)	Variable corresponding to an array element



- (1) Global label setting data
- (2) Global label
- (3) Structure definition
- (4) Structure member
- (5) Array type global label
- (6) Label Node (Object)
- (7) Label Node (Variable)
- (8) Label Node (Structure)
- (9) Label Node (Field)
- (10) Label Node (Element)

Namespace of a label Node

The Namespace dedicated to building label Nodes is referred to as MELNS. All label Nodes are created in MELNS.

- MELNS is automatically added or deleted according to whether label Nodes exist.
- MELNS can also be deleted by selecting [Delete] from the right-click menu in the NamespaceArray screen. (All label Nodes are deleted.)
- This Namespace is dedicated to building label Nodes; therefore, it cannot be imported even if exported as an information model file.

■MELNS

Item	Description
NamespaceIndex	Value from 2 to 12 • When only Namespace ns=0 is included in a project: 2 • When a Namespace other than ns=0 is included in a project: Maximum value + 1
NamespaceURI	https://www.mitsubishielectric.com/fa/OPCUA/MELNS/ROPCUA
Version	1.0.0
PublicationDate	Date and time when a label Node was added/deleted/updated ^{*1} /moved Format: yyyy-MM-ddTHH:mm:ss.ffffffZ (UTC)
RequiredModel	Information model with label Nodes under an Object or Variable

^{*1} Date and time when importing global labels, regardless of whether the label Node is affected

NodeId/Reference of a label Node

■NodeId

This indicates the NodeId of a label Node.

- NamespaceIndex: Fixed to MELNS
- IdType: Fixed to s (String)
- Identifier: Refer to the following:

Label Node type	Identifier	Example
Label Node (Object)	"CT\$" + Global label setting data name	CT\$global1
Label Node (Variable)	Global label name ^{*1}	label01
Label Node (Structure)	"ST\$" + Structure definition name	ST\$struct1
Label Node (Field)	Structure type global label name + '.' (period) + Structure member name ^{*1}	strLabel01.mem01
Label Node (Element)	Array type global label name + [{Element number}] ^{*1*2}	arrLabel01[1,2,3]

^{*1} For a timer or counter, the appropriate suffix, S (contact), C (coil), or N (current value), is added to the end, delimited by a period (.).
(Example) label01.S, strLabel01.mem01.C, arrLabel01[1,2,3].N

^{*2} An element number is added in ascending order of dimension, delimited by a comma (.).

■Reference

This indicates the Reference of a label Node.

Label Node type	ReferenceType	SourceNode
Label Node (Object)	Organizes	Placement destination Object
Label Node (Variable)	HasComponent	Placement destination Object or Variable
Label Node (Structure)	HasSubtype	Structure (i=22)
Label Node (Field)	HasComponent	Parent label Node (Variable)
Label Node (Element)	HasComponent	Parent label Node (Variable)

Service Set

The following table lists the Service Sets and Services supported by this product and related to the information model function.

Service Set	Service	Reference
View Service Set	Browse	Page 52 View Service Set
	BrowseNext	
	TranslateBrowsePathsToNodeIds	
Attribute Service Set	Read ^{*1}	Page 52 Read
	Write	Page 53 Write
MonitoredItem Service Set	CreateMonitoredItems	Page 54 CreateMonitoredItems
	ModifyMonitoredItems	Page 56 ModifyMonitoredItems
	SetMonitoringMode	—
	SetTriggering	—
	DeleteMonitoredItems	Page 56 DeleteMonitoredItems
Subscription Service Set	CreateSubscription	Page 56 Subscription Service Set
	ModifySubscription	
	SetPublishingMode	
	Publish	
	Republish	
	DeleteSubscription	
Method Service Set	Call	Page 57 Call

^{*1} The Read of the DataTypeDefinition for the DataType Node with ns=0 is not supported.
For details on each Service Set and Service, refer to the OPC UA specifications.
The following explains the details on the specifications unique to this product for each Service.

View Service Set

Services are provided to search for Nodes in an AddressSpace.

Attribute Service Set

Services are provided to access an Attribute that is a part of a Node.

An OPC UA client can read and write data in a CPU module by requesting the Read and Write of Values from an OPC UA server module.

■Read

The Read Service is supported by this product.

In addition, the Read with an IndexRange specified is supported.

- Processing for a Value

When a Read Request is issued from an OPC UA client, an OPC UA server module requests a CPU module to read data mapped to a target Variable, and then returns a response to the OPC UA client based on the data reading response. For details on the processing, refer to the following:

 Page 81 Processing when requesting the Read

If no data is mapped to the Variable, data is not read from the CPU module and a Value value retained in the OPC UA server module is returned as it is.

- ServiceResult

A service execution result (ServiceResult) is included in a response (Response) for the Read.

When the Read succeeds, Good is returned as the execution result.

The following codes are returned if the Read fails due to a cause specific to an OPC UA server module. For other possible codes, refer to the OPC UA specifications.

Code	Cause	Corrective action
Bad_TooManyOperations	The load on an OPC UA server module is high, and a request cannot be processed.	Reduce the frequency of requests to the OPC UA server module.
Bad_Shutdown	An OPC UA server module is stopping, and a request cannot be processed.	Restart the OPC UA server module then resend the request.

- StatusCode

Data reading results (results) for the Attributes of each Node are included in a response (Response) for the Read, and a StatusCode indicating whether the reading succeeds or fails is included in the results.

When the reading succeeds, the StatusCode for Good is returned.

The following codes are returned if the reading fails due to a cause specific to an OPC UA server module. For other possible codes, refer to the OPC UA specifications.

Code	Cause	Corrective action
Bad_NoCommunication	• A mapped global label is not written to a CPU module. • The setting for a global label is incorrect. • The value of a global label is out of the range of the DataTypes of Nodes.	• Review the setting for mapping. • If an error occurs in the OPC UA server module, check the contents described in the corrective action for the error code to take corrective action.
Bad_NotSupported	A request was sent for an unsupported Variable.	Do not send a request for an unsupported Variable.

■Write

The Write Service is supported by this product.

Only the Value is a target Attribute for the Write. Note that the Write is not supported if the following conditions apply:

- An IndexRange is specified.
- The number of elements in each dimension of an array does not match.
- Processing for a Value

When a Write Request is issued from an OPC UA client, an OPC UA server module requests a CPU module to write data to data mapped to a target Variable, and then returns a response to the OPC UA client based on the data writing response. For details on the processing, refer to the following:

 Page 87 Processing when requesting the Write

If no data is mapped to the Variable, data is not written to the CPU module and a Value value retained in the OPC UA server module is rewritten.

- ServiceResult

A service execution result (ServiceResult) is included in a response (Response) for the Write.

When the Write succeeds, Good is returned as the execution result.

The following codes are returned if the Write fails due to a cause specific to an OPC UA server module. For other possible codes, refer to the OPC UA specifications.

Code	Cause	Corrective action
Bad_TooManyOperations	The load on an OPC UA server module is high, and a request cannot be processed.	Reduce the frequency of requests to the OPC UA server module.
Bad_Shutdown	An OPC UA server module is stopping, and a request cannot be processed.	Restart the OPC UA server module then resend the request.

- StatusCode

Data writing results (results) for the Attributes of each Node are included in a response (Response) for the Write, and a StatusCode indicating whether the writing succeeds or fails is included in the results.

When the writing succeeds, the StatusCode for Good is returned.

The following codes are returned if the writing fails due to a cause specific to an OPC UA server module. For other possible codes, refer to the OPC UA specifications.

Code	Cause	Corrective action
Bad_NoCommunication	• A mapped global label is not written to a CPU module. • The setting for a global label is incorrect.	• Review the setting for mapping. • If an error occurs in the OPC UA server module, check the contents described in the corrective action for the error code to take corrective action.
Bad_NotSupported	A request was sent for an unsupported Variable.	Do not send a request for an unsupported Variable.
Bad_OutOfRange	A written value exceeds the range of values for the data type of the mapping destination.	Write a value within the range of values for the data type of the mapping destination.
Bad_WriteNotSupported	The number of elements in an Array does not match the writing destination.	Match the number of elements in the Array before writing.

MonitoredItem Service Set

Services are provided to handle MonitoredItems to be monitored for changes in Node data and Event occurrences. By associating a MonitoredItem with a Subscription and combining it with the Subscription Service Set, changes in data and Event occurrences are reported to an OPC UA client.

Precautions

If repeatedly requesting the CreateMonitoredItems, ModifyMonitoredItems, or DeleteMonitoredItems from an OPC UA server module, the performance of the module may degrade. It is recommended to send requests from an OPC UA client in a batch if possible.

■CreateMonitoredItems

The CreateMonitoredItems Service is supported by this product.

- Periodic data reading

Variables added to a MonitoredItem by using the CreateMonitoredItems from an OPC UA client are targets for periodic data reading. The cycle is determined based on a value (ms units) specified for the itemsToCreate → requestedParameters → samplingInterval included in a Request for the CreateMonitoredItems. For details on the processing, refer to the following:

 Page 82 Processing when performing a Subscription

- ServiceResult

A service execution result (ServiceResult) is included in a response (Response) for the CreateMonitoredItems.

When the CreateMonitoredItems succeeds, Good is returned as the execution result.

For codes that can be returned if the CreateMonitoredItems fails, refer to the OPC UA specifications.

- StatusCode

Results (results) of registering each item in the monitoring target are included in a response (Response) for the CreateMonitoredItems, and a StatusCode indicating whether the registration succeeds or fails is included in the results. When the registration succeeds, the StatusCode for Good is returned.

The following codes are returned if the registration fails due to a cause specific to an OPC UA server module. For other possible codes, refer to the OPC UA specifications.

Code	Cause	Corrective action
Bad_NotSupported	A request was sent for an unsupported Variable.	Do not send a request for an unsupported Variable.
Bad_TooManyMonitoredItems	The total Variable data size in MonitoredItems has reached the upper limit.*1	• Reduce the number of MonitoredItems. • Reduce the Variable data size.

*1 For the upper limit of the Variable data size (total), refer to the following:

☞ Page 20 Information models and mapping specifications

The Variable data sizes for each of the Built-in DataTypes are as follows.

For an array, the value is a data size multiplied by the number of elements.

Built-in DataTypes	Data size [byte]
Boolean	1
SByte	1
Byte	1
Int16	2
UInt16	2
Int32	4
UInt32	4
Int64	8
UInt64	8
Float	4
Double	8
String	1024
DateTime	8
Guid	1
ByteString	10240
XmlElement	1
NodeId	1030
ExpandedNodeId	1
StatusCode	1
QualifiedName	1026
LocalizedText	1024
ExtensionObject	1
DataValue	1
Variant	1
DiagnosticInfo	1

■ModifyMonitoredItems

The ModifyMonitoredItems Service is supported by this product.

- Change of the periodic data reading cycle

When using this service to request a change in the SamplingInterval for a Variable from an OPC UA client, the periodic data reading cycle also changes.

- ServiceResult

A service execution result (ServiceResult) is included in a response (Response) for the ModifyMonitoredItems.

When the ModifyMonitoredItems succeeds, Good is returned as the execution result.

For codes that can be returned if the ModifyMonitoredItems fails, refer to the OPC UA specifications.

- StatusCode

Results (results) of changing each monitored item are included in a response (Response) for the ModifyMonitoredItems, and a StatusCode indicating whether the change succeeds or fails is included in the results.

When the change succeeds, the StatusCode for Good is returned.

For codes that can be returned if the change fails, refer to the OPC UA specifications.

■DeleteMonitoredItems

The DeleteMonitoredItems Service is supported by this product.

- Exclusion from periodic data reading targets

Variables deleted from all MonitoredItems by using the DeleteMonitoredItems from an OPC UA client are excluded from periodic data reading targets.

- ServiceResult

A service execution result (ServiceResult) is included in a response (Response) for the DeleteMonitoredItems.

When the DeleteMonitoredItems succeeds, Good is returned as the execution result.

For codes that can be returned if the DeleteMonitoredItems fails, refer to the OPC UA specifications.

- StatusCode

Results (results) of deleting each monitored item are included in a response (Response) for the DeleteMonitoredItems, and a StatusCode indicating whether the deletion succeeds or fails is included in the results.

When the deletion succeeds, the StatusCode for Good is returned.

For codes that can be returned if the deletion fails, refer to the OPC UA specifications.

Subscription Service Set

Services are provided to report a notification to an OPC UA client.

Method Service Set

A service is provided to call Methods retained in an Object.

■Call

The Call Service is supported by this product.

- Processing for the Call

When a Call Request is issued from an OPC UA client, an OPC UA server module notifies a CPU module via data mapped to a Handle, and then returns a response to the OPC UA client asynchronously with the execution status of a program of the CPU module. For details on the processing, refer to the following:

 Page 90 Processing when requesting the Call

If no data is mapped to the Handle, the CPU module is not notified.

If no data is mapped to InputArguments, the values of locations where the arguments are used in the program of the CPU module are ignored.

Precautions

- Turn OFF the 0th bit of data mapped to the Handle on the program side of the CPU module. The same Method cannot be executed until it turns OFF.
- Whether the Call succeeds or fails is determined by whether data writing (writing to data mapped to the Handle and InputArguments) succeeds or fails, regardless of whether the program of the CPU module succeeds or fails. Therefore, even if the CPU module is in the STOP state, the Call succeeds when the data writing succeeds. Note that the program does not run and the 0th bit of the Handle remains ON while the CPU module is in the STOP state.

- ServiceResult

A service execution result (ServiceResult) is included in a response (Response) for the Call.

When the Call succeeds, Good is returned as the execution result.

The following code is returned if the Call fails due to a cause specific to an OPC UA server module. For other possible codes, refer to the OPC UA specifications.

Code	Cause	Corrective action
Bad_Shutdown	An OPC UA server module is stopping, and a request cannot be processed.	Restart the OPC UA server module then resend the request.

- StatusCode

Results (results) for each Method are included in a response (Response) for the Call, and a StatusCode indicating whether the Method execution succeeds or fails is included in the results.

When the Method execution succeeds, the StatusCode for Good is returned.

The following codes are returned if the Method execution fails due to a cause specific to an OPC UA server module. For other possible codes, refer to the OPC UA specifications.

Code	Cause	Corrective action
Bad_TooManyOperations	• The Method is being executed. • The load on an OPC UA server module is high, and a request cannot be processed.	• Resend the request after the 0th bit of a Handle is turned OFF by a program of a CPU module. If processing to turn the bit OFF is not included in the program, add it. • Reduce the frequency of requests to the OPC UA server module.
Bad_NotImplemented	Data is not mapped to a Handle.	Map data to the Handle.
Bad_NoCommunication	• A mapped global label is not written to a CPU module. • The setting for a global label is incorrect.	• Review the setting for mapping. • If an error occurs in the OPC UA server module, check the contents described in the corrective action for the error code to take corrective action.
Bad_NotSupported	The NodeClass of a Node specified for an objectId is ObjectType. (An OPC UA server module does not support Methods belonging to ObjectType.)*1	Specify a Node with the NodeClass of Object for the objectId.

*1 When executing a Method belonging to the ObjectType with ns=0, a StatusCode not for Bad_NotSupported may be returned.

- InputArgumentResultCode

Determination results (inputArgumentResults) for each InputArgument are included in a result of a Method, and a StatusCode indicating whether the InputArgument value is correct or incorrect is included in the results.

When the InputArgument value is correct, the StatusCode for Good is returned.

The following codes are returned if the InputArgument value is determined to be incorrect due to a cause specific to an OPC UA server module. For other possible codes, refer to the OPC UA specifications.

Code	Cause	Corrective action
Bad_OutOfRange	A written value exceeds the range of values for the data type of the mapping destination.	Write a value within the range of values for the data type of the mapping destination.
Bad_NotSupported	• Data is mapped to an unsupported Argument^{*1}. • The number of elements in an Array of an Argument does not match the writing destination.	• Do not map data to an unsupported Argument. • Match the number of elements in the Array before performing the Call.

^{*1} Structures with optional fields, OptionSets, Unions, and Structures with a Subtype defined for Structures other than ns=0 are applicable.

8.2 Data Assignment Function

The data assignment function is for assigning mapped data to Nodes.

Function list

The data assignment function consists of the following items:

Item	Description	Reference
DataTypes assignment	To assign data to each DataType of Built-in DataTypes.	Page 60 DataType assignment
Structure assignment	To assign data to each Field of a Structure.	Page 65 Structure assignment
Enumeration assignment	To assign data with an enumerator (Value) of an Enumeration as Int32.	Page 67 Enumeration assignment
Method assignment	To assign data to a Handle, and to an Argument under InputArguments.	Page 68 Method assignment
Type conversion check	To check if data can be represented when there is any difference in signs, bit ranges, and character codes between the DataType of a Node and the data type of a data source.	Page 69 Type conversion check

DataType assignment

Data types that can be assigned to Built-in DataTypes

■Data types of labels

The following table shows the data types of labels that can be assigned to Built-in DataTypes.

Some data types can be assigned with type conversion checks. ( Page 69 Type conversion check)

For details on the Built-in DataTypes, refer to the OPC UA specifications.

⊙: Can be assigned, ○: Can be assigned (with a type conversion check), —: Cannot be assigned

Built-in DataTypes	Data type of a label								
	Bit	Word [unsigned]/bit string [16-bit]	Double word [unsigned]/bit string [32-bit]	Word [signed]	Double word [signed]	Float (single precision) *1	Float (double precision) *1	Time	Structure
Boolean	⊙	—	—	—	—	—	—	—	—
SByte	—	○	○	○	○	—	—	—	—
Byte	—	○	○	○	○	—	—	—	—
Int16	—	○	○	⊙	○	—	—	—	—
UInt16	—	⊙	○	○	○	—	—	—	—
Int32	—	○	○	○	⊙	—	—	⊙	—
UInt32	—	○	⊙	○	○	—	—	—	—
Int64	—	○	○	○	○	—	—	—	—
UInt64	—	○	○	○	○	—	—	—	—
Float	—	—	—	—	—	⊙	○*2	—	—
Double	—	—	—	—	—	○*2	⊙	—	—
String	—	—	—	—	—	—	—	—	—
DateTime	—	—	○*3	—	○*3	—	—	—	—
Guid	—	—	—	—	—	—	—	—	—
ByteString	—	—	—	—	—	—	—	—	—
XmlElement	—	—	—	—	—	—	—	—	—
NodeId	—	—	—	—	—	—	—	—	○*4*5*6
ExpandedNodeId	—	—	—	—	—	—	—	—	—
StatusCode	—	—	—	—	—	—	—	—	—
QualifiedName	—	—	—	—	—	—	—	—	○*4*5*6
LocalizedText	—	—	—	—	—	—	—	—	—
ExtensionObject	—	—	—	—	—	—	—	—	—
DataValue	—	—	—	—	—	—	—	—	—
Variant	—	—	—	—	—	—	—	—	—
DiagnosticInfo	—	—	—	—	—	—	—	—	—

Built-in DataTypes	Data type								
	String	String [Unicode]	Timer	Retentive timer	Long timer	Long retentive timer	Counter	Long counter	Pointer
Boolean	—	—	⌚*7	⌚*7	⌚*7	⌚*7	⌚*7	⌚*7	—
SByte	—	—	○*8	○*8	○*8	○*8	○*8	○*8	—
Byte	—	—	○*8	○*8	○*8	○*8	○*8	○*8	—
Int16	—	—	○*8	○*8	○*8	○*8	○*8	○*8	—
UInt16	—	—	⌚*8	⌚*8	○*8	○*8	⌚*8	○*8	—
Int32	—	—	○*8	○*8	○*8	○*8	○*8	○*8	—
UInt32	—	—	○*8	○*8	⌚*8	⌚*8	○*8	⌚*8	—
Int64	—	—	○*8	○*8	○*8	○*8	○*8	○*8	—
UInt64	—	—	○*8	○*8	○*8	○*8	○*8	○*8	—
Float	—	—	—	—	—	—	—	—	—
Double	—	—	—	—	—	—	—	—	—
String	○*4*9*10	○*4*5	—	—	—	—	—	—	—
DateTime	—	—	—	—	○*8	○*8	—	○*8	—
Guid	—	—	—	—	—	—	—	—	—
ByteString	—	—	—	—	—	—	—	—	—
XmlElement	—	—	—	—	—	—	—	—	—
NodeId	—	—	—	—	—	—	—	—	—
ExpandedNodeId	—	—	—	—	—	—	—	—	—
StatusCode	—	—	—	—	—	—	—	—	—
QualifiedName	—	—	—	—	—	—	—	—	—
LocalizedText	○*4*9*10*11	○*4*5*11	—	—	—	—	—	—	—
ExtensionObject	—	—	—	—	—	—	—	—	—
DataValue	—	—	—	—	—	—	—	—	—
Variant	—	—	—	—	—	—	—	—	—
DiagnosticInfo	—	—	—	—	—	—	—	—	—

*1 If the value is -0, a denormalized number, non-numeric value, or $\pm\infty$ in the Write, the StatusCode for Bad_OutOfRange is returned.

*2 May be an approximate value because the precision differs.

*3 The number of seconds is assigned starting from 0:00:00 on January 1st, 2000.

Assignment targets correspond to start devices storing second data for the CPU module instruction (SEC2DATE(P)(_U)) before change. (LUMELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks))

*4 Character strings exceeding the valid number of characters in the Write are truncated and those within the range are applied. In this case, no error is output to an OPC UA client or module.

*5 Character codes are converted between UTF-16 and UTF-8.

*6 For the specifications of the structure, refer to the following:

☞ Structure specifications

*7 The contact and coil can be used.

*8 The current value can be used.

*9 Character codes are converted between SJIS and UTF-8.

*10 If the number of characters specified for a string type label is an odd number, 'the specified number of characters - 1' is the valid number of characters.

*11 For the Text part of LocalizedText only. For the definition of LocalizedText, refer to the OPC UA specifications.

- Structure specifications

The following shows the structure specifications of a label that can be assigned to a NodeId and QualifiedName.

Structure definition names are optional; however, member names and data types must be set according to the specifications. If any member name does not comply with the specifications, a global label unavailable error (error code: 1921H) occurs. In addition, if the member name matches but the data type does not match, a data type unavailable error (error code: 1A00H) occurs.

For details on the errors, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

For the definitions of NodeId and QualifiedName, refer to the OPC UA specifications.

NodeId

Member name	Data type	Value to be stored	Example of NodeId representation
NamespaceIndex	Word [unsigned]/bit string [16-bit]	NamespaceIndex of a NodeId (index of NamespaceURI)	■When the IdType is NUMERIC ns=2;i=5001 ■When the IdType is STRING ns=2;s=bLabel1
IdType	Word [signed]	IdType of a NodeId (representation format of Identifier)*1 • 0: NUMERIC • 1: STRING	
Identifier	■When the IdType is 0 (NUMERIC) • String [Unicode] • Double word [unsigned]/bit string [32-bit] ■When the IdType is 1 (STRING) • String [Unicode]	Identifier of a NodeId (identifier) (Example) • When the IdType is 0 (NUMERIC): 5001 • When the IdType is 1 (STRING): bLabel1	

*1 GUID and OPAQUE are not supported.



Data type for Identifier

- Check the information model used. When the IdType of a NodeId is NUMERIC only (the Identifier is described as 'i='), it is recommended to use the double word [unsigned] or bit string [32-bit].
- For other cases than the above, or if multiple IdTypes are included, use the string [Unicode].

QualifiedName

Member name	Data type	Value to be stored
NamespaceIndex	Word [unsigned]/bit string [16-bit]	NamespaceIndex of a QualifiedName (index of NamespaceURI)
Name	String [Unicode]	Name of a QualifiedName

Additional information

When adding an OPC UA server module in the setting in which module labels are used in GX Works3, structure definitions in the specifications are also added. By using the added structure definitions, labels can be created without creating structure definitions.

Structure definitions corresponding to Built-in DataTypes are as follows:

Built-in DataTypes	Data type for Identifier	Structure definition name
NodeId	Double word [unsigned]/bit string [32-bit]	M+RD81OPC96_UANodeId_N
	String [Unicode]	M+RD81OPC96_UANodeId_S
QualifiedName	—	M+RD81OPC96_UAQualifiedName

Data types that can be assigned to other DataTypes

■Data types of labels

The following table shows the data types of labels that can be assigned to DataTypes other than Built-in DataTypes.

No.	Category	Data type
1	Subtype of Built-in DataTypes	Compliant with the specifications of the parent Built-in DataTypes.*1 (Page 60 Data types that can be assigned to Built-in DataTypes)
2	Subtype of Enumeration	Refer to the specifications of the Enumeration. (Page 67 Enumeration assignment)
3	Subtype of Structure	Refer to the specifications of the Structure. (Page 65 Structure assignment)
4	Other than No.1 to 3	Cannot be assigned.

*1 DataTypes using the OptionSetValues Property are not supported.

Ex.

The following table shows examples of data types of labels that can be assigned.

DataTypes	Category	Data type
Number	—	Other than No.1 to 4 (abstract data type)
	Decimal	Other than No.1 to 4
	Double	—
		Duration
		MyDuration
	Integer	—
		Int16
		MyInt16
Enumeration	—	Other than No.1 to 4 (abstract data type)
	MyEnumeration	Subtype of Enumeration
Structure	—	Other than No.1 to 4 (abstract data type)
	MyStructure	Subtype of Structure

Array handling

■Array assignment method

Assign data to each element of an Array.

The following table shows examples of global label assignment for a one-dimensional Variable Array with three elements.

Variable	Global label	Description
MyVariable[0]	LabelA	When assigning a global label that is basic data and non-array
MyVariable[1]	LabelB[0]	When assigning an element of an array global label
MyVariable[2]	stLabelC.member	When assigning a member of a structure global label

Note that the data assignment availability differs depending on values in the ValueRank and ArrayDimensions of Variables.

○: Can be assigned, ▲: Can be assigned with limitations^{*1}, ×: Cannot be used, { }: Number of elements (n, m, and k are any positive numbers other than 0.)

	Number of dimensions	Not set (null)	1		2			3		
	ArrayDimensions	-	Not set {n}	Set {0}	Not set {n,m}	Partially set {n,0}, etc.	All set {0,0}	Not set {n,m,k}	Partially set {n,m,0}, etc.	All set {0,0,0}
ValueRank	-3 (ScalarOrOneDimension)	▲{1}*2								
	-2 (Any)									
	0 (OneOrMoreDimension)									
	-1 (Scalar)*3	○								
	1	▲{1}*2	○{n}	▲{1}*2	×			×		
	2	▲{1,1}*2	×		○{n,m}	×	▲{1,1}*2			
	3	▲{1,1,1}*2			×			○{n,m,k}	×	▲{1,1,1}*2
	4 or higher*4	×								
	-4 or lower*4									
	Others*3									

*1 The number of dimensions, elements, or both is undefined; therefore, the numbers of elements listed in the table are the initial values for mapping array specification.

*2 '×: Cannot be used' for an Argument under InputArguments of a Method.

*3 If the ValueRank in an information model file is "" or the attribute is omitted, it is treated as -1 (Scalar).

*4 Not supported.

In addition, if a value in the ArrayDimensions exceeds the upper limit value, it cannot be used.

○: Can be assigned, ×: Cannot be used

ArrayDimensions	Assignment availability
10240 or less ^{*1}	○
10241 or more ^{*1}	×

*1 Indicates the product of valid ArrayDimensions.

■Limited assignment

When limited assignment is available, the handling is as follows:

- For limited assignment according to the ValueRank

The Rank for mapping array specification is followed. If not setting mapping array specification, the ValueRank is regarded as 1 (OneDimension) for handling.

For details on mapping array specification, refer to the following:

📖 Page 115 Mapping array specification

- For limited assignment according to the ArrayDimensions

The Dim 1 to 3 for mapping array specification are followed. If not setting mapping array specification, the ArrayDimensions are regarded as 1 for handling. (If the product of valid ArrayDimensions exceeds 10240 after being regarded as 1, it is treated as unavailable.)

For details on mapping array specification, refer to the following:

📖 Page 115 Mapping array specification

Structure assignment



In this section, the Subtype of Structure is also represented as Structure.

Data types that can be assigned to Fields of a Structure

■Data types of labels

A label can be assigned to each Field of a Structure if its data type can be assigned in the following:

📖 Page 60 DataType assignment

Handling when a Field is of Structure

When a Field of a Structure is of Structure, data can also be assigned recursively to its Field.

However, if a Field is in a hierarchy of a Structure exceeding the upper limit, data cannot be assigned. For the upper limit, refer to the following:

📖 Page 20 Information models and mapping specifications

s01 (Structure)	—			1st hierarchy	
	member01			Field in the 1st hierarchy	Can be assigned.
	s02 (Structure)	—			2nd hierarchy
		member02			Field in the 2nd hierarchy
		...			...
		sMAX-1 (Structure)	—		
			(MAX-1) hierarchy		
			memberMAX-1		
		sMAX (Structure)	—		
			MAX hierarchy		
			memberMAX		
		sMAX+1 (Structure)	—		
			(MAX+1) hierarchy		
			memberMAX+1		
			Field in the (MAX+1) hierarchy		
			Cannot be assigned.		
			...		

8

Array handling

■When a Structure is an Array

Data can be assigned to each element of an Array and each Field of a Structure.

The following table shows examples of global label assignment when a Structure with three Fields is a one-dimensional Array with two elements.

Variable	Global label	Description
MyStructure[0].Member1	LabelA	When assigning a global label that is basic data and non-array
MyStructure[0].Member2	LabelB[0]	When assigning an element of an array global label
MyStructure[0].Member3	stLabelC.member	When assigning a member of a structure global label
MyStructure[1].Member1	stLabelD[0].member	
MyStructure[1].Member2	stLabelE.member[0]	
MyStructure[1].Member3	stLabelF[0].member[0]	

■When a Field of a Structure is an Array

Data can be assigned to each element of an Array of a Field.

The following table shows examples of global label assignment when the first Field of a Structure with two Fields is a one-dimensional Array with three elements.

Variable	Global label	Description
MyStructure.MemberA[0]	LabelA	When assigning a global label that is basic data and non-array
MyStructure.MemberA[1]	LabelB[0]	When assigning an element of an array global label
MyStructure.MemberA[2]	stLabelC.member	When assigning a member of a structure global label
MyStructure.MemberB	stLabelD[0].member	

■Restrictions

Data cannot be assigned to a Variable using a Structure with a Field the ValueRank of which is 2 or higher, 0, or -2 or lower. (It cannot be assigned to all Fields included in a target Variable, regardless of the hierarchies in which the Fields are included.) Fields and Elements of the applicable Variables are not displayed in the NodeTree screen.

Subtype handling

A Structure with a Subtype defined cannot be used for a Structure other than ns=0.

The following table shows availability examples of the Subtype.

○: Available, ×: Not available

Structure (ns=0;i=22) configuration			Availability
Structure	(ns=0)	(ns=0)	○
		(ns=0) (ns=other than 0)	○
		(ns=other than 0)	○
		(ns=other than 0) (ns=other than 0)	×
	(ns=other than 0)	(ns=other than 0)	×

Enumeration assignment



In this section, the Subtype of Enumeration is also represented as Enumeration.

Data types that can be assigned to Enumerations

■Data types of labels

The following table shows the data types of labels that can be assigned to Enumerations.

Note that an enumerator (value) of an Enumeration is treated as the Int32 type in the OPC UA specifications. For details, refer to the OPC UA specifications.

◎: Can be assigned, ○: Can be assigned (with a type conversion check), —: Cannot be assigned

Data type of a label	Assignment availability
Bit	—
Word [unsigned]/bit string [16-bit]	○
Double word [unsigned]/bit string [32-bit]	○
Word [signed]	○
Double word [signed]	◎
Float (single precision)	—
Float (double precision)	—
Time	—
String	—
String [Unicode]	—
Timer	—
Retentive timer	—
Long timer	—
Long retentive timer	—
Counter	—
Long counter	—
Pointer	—

Array handling

A label can be assigned to an Array of an Enumeration according to the contents in the following:

📖 Page 64 Array handling

Structure handling

A label can be assigned according to the contents in the following even if an Enumeration is a Field of a Structure.

📖 Page 65 Structure assignment

Restrictions

- Only Subtypes immediately under an Enumeration (ns=0;i=29) can be used.
- Enumerations including a Value of -1 in EnumValues are not supported.

Method assignment

A label can be assigned to a Handle, and to an Argument under InputArguments.

Data types that can be assigned to a Handle

■Data types of labels

A label of word [unsigned] or bit string [16-bit] can be assigned to a Handle.

Data types that can be assigned to an Argument under InputArguments

■Data types of labels

A label can be assigned to each Argument if its data type can be assigned in the following:

☞ Page 60 DataType assignment

Type conversion check

When converting the data type, whether data before conversion can be represented after conversion can be checked.

Conversion type	Reference
Sign conversion	Page 69 Sign conversion
Bit range conversion	Page 69 Bit range conversion
Character code conversion	Page 74 Character code conversion
DateTime conversion	Page 74 DateTime conversion
Enumeration conversion	Page 74 Enumeration conversion

Sign conversion

If data before conversion can be represented, the sign is converted to the same value as the data before conversion.
If the sign of data is reversed, an error is reported to the conversion source and the sign is not converted.

Bit range conversion

■Bit range (wide) → (narrow)

Since the bit range is converted to one with lower precision, the precision of the bit string after conversion will be lower than before, and the original data may not be represented.

If the original data can be represented, the data is represented even after conversion.

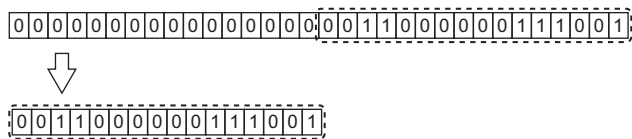
If the original data cannot be represented, an error is reported to the conversion source and the bit range is not converted.

Ex.

If the original data can be represented

12345 of double word [signed] → Int16

The data remains 12345 and the original data can be represented.

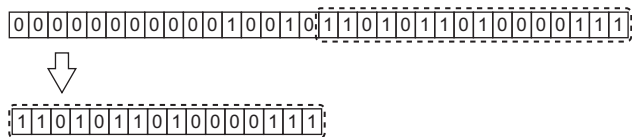


Ex.

If the original data cannot be represented

1234567 of double word [signed] → Int16

The data will be 54919 and the original data cannot be represented.



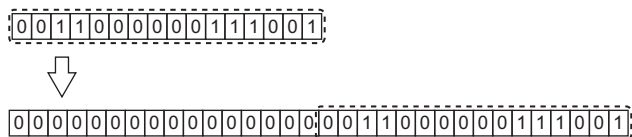
■Bit range (narrow) → (wide)

Since the bit range is converted to one with higher precision, the precision of the bit string before conversion can be retained, and the original data can be represented even after conversion.

Ex.

12345 of Int16 → double word [signed]

The data remains 12345 and the original data can be represented.



■Floating point conversion (data reading)

Data is converted in the following case:

- Data reading from a label 'float (double precision)' to a Variable 'Float'*1

*1 A rounding error occurs.

Read values corresponding to data reading values are as follows:

Data reading value (double-precision real number)		➡	Read value (Float)
1 2 3 3 3 3 4	$+\infty$	➡	$+\infty$
	$+1.79769313486231e+308 \geq$ $+3.402823e+38 <$	➡	
	$+3.402823e+38 \geq$ $+1.175495e-38 \leq$	➡	Data reading value in the left (The precision is changed.)
	$+1.175495e-38 >$ $+2.22507385850721e-308 \leq$	➡	0
	$+2.22507385850721e-308 >$ $0 <$	➡	
	0	➡	
	-0	➡	-0
	-0 > $-2.22507385850721e-308 <$	➡	
	$-2.22507385850721e-308 \geq$ $-1.175495e-38 <$	➡	
	$-1.175495e-38 \geq$ $-3.402823e+38 \leq$	➡	Data reading value in the left (The precision is changed.)
	$-3.402823e+38 >$ $-1.79769313486231e+308 \leq$	➡	$-\infty$
	$-\infty$	➡	
	NaN	➡	NaN

❶ Value range of double-precision real numbers

❷ Value range of single-precision real numbers

❸ Denormalized number

❹ Non-numeric value

- Data reading from a label 'float (single precision)' to a Variable 'Double'

Read values corresponding to data reading values are as follows:

Data reading value (single-precision real number)		➡	Read value (Double)
	$+\infty$	➡	$+\infty$
	$+3.402823\text{e}+38 \geq$ $+1.175495\text{e}-38 \leq$	➡	Data reading value in the left (The precision is changed.)
①	② $+1.175495\text{e}-38 >$ $0 <$	➡	0
	0	➡	
	-0	➡	-0
	② $-0 >$ $-1.175495\text{e}-38 <$	➡	
	$-1.175495\text{e}-38 \geq$ $-3.402823\text{e}+38 \leq$	➡	Data reading value in the left (The precision is changed.)
	$-\infty$	➡	$-\infty$
③	NaN	➡	NaN

- ① Value range of single-precision real numbers
 ② Denormalized number
 ③ Non-numeric value

■Floating point conversion (data writing)

Data is converted in the following case:

- Data writing from a Variable 'Float' to a label 'float (double precision)'

Data writing values corresponding to Write values of a Variable are as follows:

Write value (Float)*1		➡	Data writing value (double-precision real number)
	+∞	➡	+∞ *2
①	+3.402823e+38≥ +1.175495e-38≤	➡	Write value in the left (The precision is changed.)
	② +1.175495e-38> 0<	➡	0
	0	➡	
	-0	➡	-0 *2
	② -0> -1.175495e-38<	➡	
	-1.175495e-38≥ -3.402823e+38≤	➡	Write value in the left (The precision is changed.)
	-∞	➡	-∞ *2
③	NaN	➡	NaN *2

① Value range of single-precision real numbers

② Denormalized number

③ Non-numeric value

*1 The number of significant digits that can be entered may differ depending on an OPC UA client.

*2 Cannot be written according to the following. (The StatusCode for Bad_OutOfRange is returned.)

📖 Page 60 Data types of labels

- Data writing from a Variable 'Double' to a label 'float (single precision)'

Data writing values corresponding to Write values of a Variable are as follows:

Write value (Double)*1		➡	Data writing value (single-precision real number)
1 2 3 3 3 3 4	+∞	➡	+∞ *2
	+1.79769313486231e+308≥ +3.402823e+38<	➡	
	+3.402823e+38≥ +1.175495e-38≤	➡	
	+1.175495e-38> +2.22507385850721e-308≤	➡	Write value in the left (The precision is changed.)
	+2.22507385850721e-308> 0<	➡	
	0	➡	
	-0	➡	-0 *2
	-0> -2.22507385850721e-308<	➡	
	-2.22507385850721e-308≥ -1.175495e-38<	➡	
	-1.175495e-38≥ -3.402823e+38≤	➡	Write value in the left (The precision is changed.)
	-3.402823e+38> -1.79769313486231e+308≤	➡	-∞ *2
	-∞	➡	
	NaN	➡	NaN *2

- ❶ Value range of double-precision real numbers
- ❷ Value range of single-precision real numbers
- ❸ Denormalized number
- ❹ Non-numeric value

*1 The number of significant digits that can be entered may differ depending on an OPC UA client.

*2 Cannot be written according to the following. (The StatusCode for Bad_OutOfRange is returned.)

📖 Page 60 Data types of labels

Character code conversion

A type conversion check is performed according to the following contents:

Built-in DataTypes	Data type of a label	
	String Character code: ASCII, SJIS	String [Unicode] Character code: UTF-16
String Character code: UTF-8	Only the ranges of ASCII and SJIS can be represented. If a character cannot be represented, it is converted to a katakana middle dot (0x8145).	Character codes are converted between UTF-8 and UTF-16.
LocalizedText Character code: UTF-8		

DateTime conversion

A type conversion check is performed according to the data types of labels that can be assigned only when the data direction is from an OPC UA client to an OPC UA server module.

Data type of a label	Description
Double word [unsigned]/bit string [32-bit]	Values in the range of 2000/1/1 0:00:00 to 2099/12/31 23:59:59 can be used. For values out of the range above, an error is reported to an OPC UA client.
Double word [signed]	Values in the range of 2000/1/1 0:00:00 to 2067/12/31 23:59:59 can be used. For values out of the range above, an error is reported to an OPC UA client.

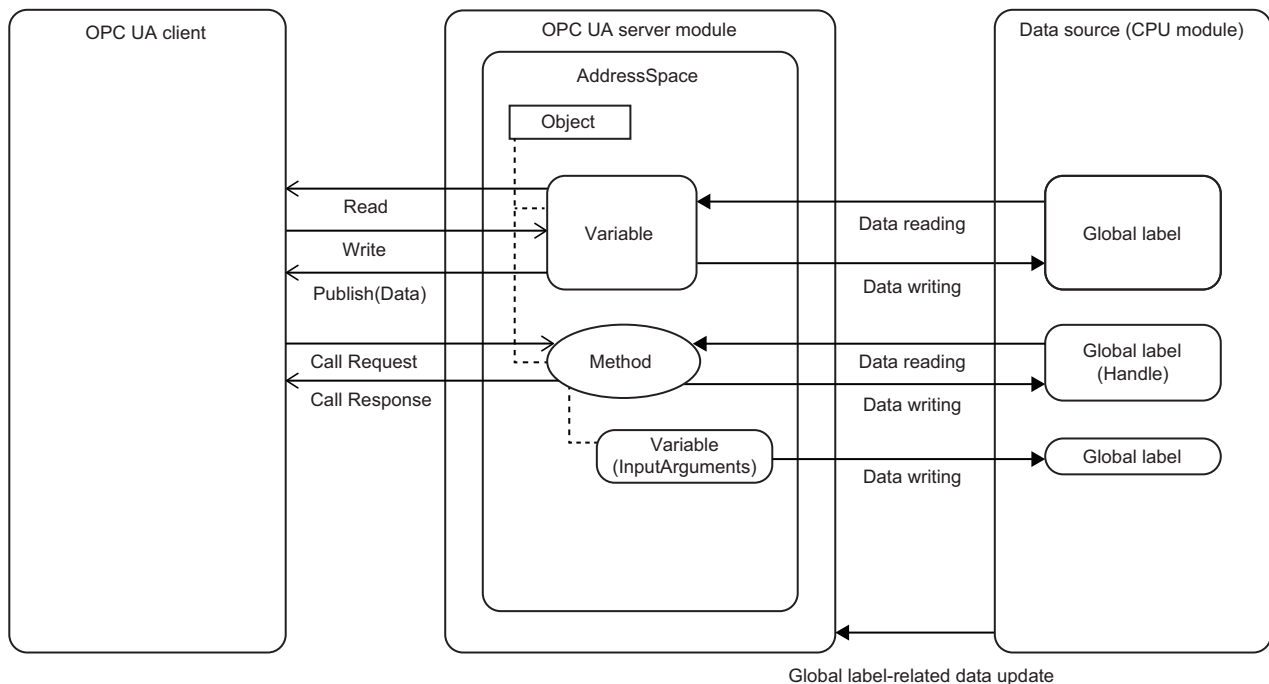
Enumeration conversion

An enumerator (value) of an Enumeration is treated as the Int32 type in the OPC UA specifications; therefore, a type conversion check is performed in the same manner as Int32 type sign conversion and bit range conversion.

8.3 Data Input/Output Function

The data input/output function is for exchanging data between an OPC UA server module and a data source (CPU module).

→ : Data flow
 → : Request/response



- To use this function, configure the Node setting in the configuration tool and write it to an OPC UA server module.
- The OPC UA server module publishes data to an OPC UA client by using an AddressSpace that consists of multiple Nodes, and the OPC UA client requests the Read, Write, Subscription^{*1}, and Call for Nodes.
- In order for the OPC UA client to access data in a CPU module, the OPC UA server module uses this function to update Node data and the data in the CPU module.

*1 The OPC UA server module Publishes Node data in response to a Subscription request from the OPC UA client.

Function list

The data input/output function consists of the following items:

Item	Description	Reference
Data management	To manage data used for mapping. Global labels of a CPU module can be used as data for an OPC UA server module by specifying their names in mapping.	Page 76 Data management
Data sampling setting	To set a sampling mode when an OPC UA server module collects data.	Page 131 Optional Operation Setting
Data	Node data	Data published to an AddressSpace by an OPC UA server module
	Data in a CPU module	Data retained in a CPU module
Processing	Data reading	To acquire data in a CPU module and apply it to Node data. This processing is basically performed when requesting the Read of the Attribute Service Set and for Sampling of the Subscription Service Set. For details on the execution timings, refer to the following: ☞ Page 80 Data reading
	Data writing	To apply Node data to data in a CPU module. This processing is performed when requesting the Write of the Attribute Service Set.
	Global label-related data update	To acquire the addresses of global labels from a CPU module. This processing allows the operation to continue without changing the setting of an OPC UA server module even if the settings such as labels, programs, FBs, and CPU parameters of a CPU module are changed.
	Method execution	To notify a CPU module of the start of a Method. This processing is performed when requesting the Call of the Method Service Set.

Data management

The following shows data used for mapping.

Data source	Data category	Description	Reference
CPU module	—	Control CPU of the own station only	—
	Global label	Global label of a CPU module	Page 76 Global label

Global label

A label name is used for mapping. It is not case-sensitive.

■Availability and specification methods

The following tables show the availability and specification methods for each data type of a global label.

- Without array

○: Available, ×: Not available

Data type	Availability	Specification method	Example
Basic data type (excluding the timer and counter)	○	Specify a label name.	LabelName
Structure type label	○	For a member of a structure type label, specify a member name delimited by a period (.) after the label name.	StructureLabelName.MemberName
	×	A structure type label name cannot be specified.	StructureLabelName
Structure type member	○	For a member of a structure type member, specify a member name delimited by a period (.) after the structure type member name.	StructureLabelName.StructureMemberName.MemberName
	×	A structure type member name cannot be specified.	StructureLabelName.StructureMemberName
Timer/counter	○	For a timer or counter type label or member, specify 'S,' 'C,' or 'N' respectively indicating the contact, coil, or current value after the name.	- Contact: TimerLabelName.S - Current value: CounterLabelName.N
Others	×	Neither bit nor digit specification is available. (If specifying a bit, a global label unavailable error (error code: 1921H) occurs.)	- Bit specification: LabelName.1 - Digit specification: K4LabelName

- One dimension

○: Available, ×: Not available

Data type	Availability	Specification method	Example
Basic data type (excluding the timer and counter)	○	For an array, specify the element number with [{element number}] after the label name.	ArrayLabelName[0]
Structure type label Structure type member	○	- When a member is an array, specify the element number with [{element number}] after the member name. - When a structure type label or member is an array, specify the element number with [{element number}] after the structure type label name or member name, and specify a member name of the structure delimited by a period (.).	- StructureLabelName.ArrayMemberName[0] - StructureArrayLabelName[0].MemberName
Timer/counter	○	For a timer or counter type array, specify the element number with [{element number}] after the name, and specify 'S,' 'C,' or 'N' respectively indicating the contact, coil, or current value after the element number.	Coil: TimerArrayLabelName[0].C
Others	×	Neither bit nor digit specification is available. (If specifying a bit, a global label unavailable error (error code: 1921H) occurs.)	- Bit specification: LabelName[0].1 - Digit specification: K4LabelName[0]

- Multiple dimensions

○: Available, ×: Not available

Data type	Availability	Specification method ^{*1}	Example
Basic data type (excluding the timer and counter)	○	• For a multidimensional (two or three) array, specify the element number in ascending order of the number of dimensions delimited by a comma (,). • When a structure type label or member is a multidimensional array, specify the element number in the same manner as the above. • When a timer or counter is a multidimensional array, specify the element number in the same manner as the above, and specify 'S,' 'C,' or 'N' respectively indicating the contact, coil, or current value after the element number.	• 2DimensionArrayLabelName[0,0] • 3DimensionArrayLabelName[0,0,0] • StructureLabelName.ArrayMemberName[0,0,0] • StructureArrayLabelName[0,0,0].MemberName • Contact: TimerArrayLabelName[0,0,0].S (Element numbers indicate one dimension, two dimensions, and three dimensions from the left.)
Structure type label			
Structure type member			
Timer/counter			
Others	×	Neither bit nor digit specification is available. (If specifying a bit, a global label unavailable error (error code: 1921H) occurs.)	• Bit specification: LabelName[0,0].1 • Digit specification: K4LabelName[0,0]

*1 When specifying elements in the format "[A][B][C]," they are automatically converted to the format "[A,B,C]."

■ Restrictions

The following table shows the restrictions on global labels and the behaviors when a global label conflicts with any restriction.

Item		Restriction	Behavior when a conflict occurs
Label/structure	Class	Only the following can be used: • VAR_GLOBAL • VAR_GLOBAL_RETAIN	An error occurs in an OPC UA server module.
	Data type	The following cannot be used ^{*1} : • Pointer type (including a member of a structure) • Function block	
	Access from external device	Only the following can be used: • Label with this checkbox selected	
	Assign (device/label)	• Assignable devices are the same as accessible devices in the tag mode. For details on the accessible devices, refer to the following: (MELSEC iQ-R OPC UA Server Module User's Manual (Startup)) • Structure type labels with assignments and structure type members cannot be used.	
Character	Full path length ^{*2}	Up to 800 characters	Mapping is unavailable in the configuration tool.
Data type	Category	The data types shown as assignable in the following can be used:  Page 60 Data Type assignment	Mapping is unavailable in the configuration tool.

*1 If used, a global label unavailable error (error code: 1921H) occurs.

*2 Total character string length of a structure type label name and its member name (for a structure type member, its member is also included) (including periods (.), an element number ([0]), and commas (,))

Data

The following shows data used for the data input/output function.

☞ Page 78 Node data

☞ Page 79 Data in a CPU module

Node data

A Node is data retained in an OPC UA server module, and is classified into eight NodeClasses.

Nodes with the NodeClass of Variable and Method are targets for the data input/output function.

NodeClass	Description* ¹	Data input/output function ○: Available, —: Not available	Reference
Variable	Represents a value.	○	Page 78 Variable
VariableType	Provides the type definition of a value.	—	—
Object	Represents a real object.	—	—
ObjectType	Provides the definition of an object.	—	—
ReferenceType	Defines the meaning of the reference relationship between Nodes.	—	—
DataType	Represents the data type of a value.	—	—
Method	Represents a function that can be called.	○* ²	Page 79 Method
View	Represents a part of a Node in an AddressSpace.	—	—

*¹ For details, refer to the OPC UA specifications.

*² Available for an OPC UA server module with firmware version '13' or later and the configuration tool with software version '1.13P' or later.

■Variable

A Variable is a Node representing a value, and mapped data in a CPU module can be input or output.

Variable initialization with the initial value

The processing differs depending on whether the initial value of a Variable (Value attribute value) is set in the information model.

The initial value setting is supported only for Built-in DataType Variables to which global labels can be assigned. (☞ Page 60 Data types of labels)

- If a Value attribute value is set

If an initial value is set for a Variable, a Value attribute value of the Variable is initialized with the set value when constructing an AddressSpace.

If a Value attribute value is invalid, refer to the following:

☞ If a Value attribute value is not set

Note that the following Variable cannot be initialized.

Variable	Initial value handling
Multidimensional array	Treated in the same manner as the following regardless of whether the initial value is set. ☞ If a Value attribute value is not set

- If a Value attribute value is not set

If the initial value of a Variable (Value attribute value) is not set in the information model, the Value may be displayed as Null when accessing from an OPC UA client.

Value update timings

The following table shows the update timings of Variable values.

Update timing	Update target	Remarks	Reference
Starting a module	All Variables	—	—
Updating the setting of a module	All Variables	—	—
Reading data	Variables to be read	If data in a CPU module is not mapped, a Variable value is not updated.	☞ Page 80 Data reading
Writing data	Variables to be written	If data in a CPU module is not mapped, a Variable value is updated but data is not written to the CPU module.	☞ Page 87 Data writing

When starting an OPC UA server module or updating its setting, all Variable values are initialized with their initial values. Then, data reading processing is performed for a Variable to which data in a CPU module is mapped and the Variable value is updated.

■Method

A Method is a Node representing a function that can be called from an OPC UA client, and can be executed via mapped data in a CPU module.

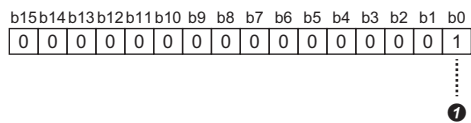
A Method itself does not retain data, but it maps data in a CPU module to a Handle, which notifies the CPU module that the Method has been Called from an OPC UA client, and to InputArguments, which are the input arguments of the Method.

Handle

A Handle is word [unsigned] or bit string [16-bit] data.

A specific function is assigned to each bit, and the availability of the function is determined by whether the bit is ON or OFF.

Bit	Item	Description
b0	Trigger	Turn this data ON to notify a CPU module that a Method has been Called from an OPC UA client. After the Method processing is completed, the CPU module turns this data OFF.
b1 to b15	System area	System area (fixed to '0')



①: Trigger (0: OFF, 1: ON)

Data in a CPU module

This is data retained in a CPU module. For available data, refer to the following:

☞ Page 76 Data management

Processing

The following shows processing performed when using the data input/output function.

- Page 80 Data reading
- Page 87 Data writing
- Page 88 Global label-related data update
- Page 90 Method execution

Data reading

This processing is for acquiring data in a CPU module and applying it to Node data.

The execution timings of data reading and sampling targets are as follows.

A sampling mode can be set in the configuration tool. (Page 131 Optional Operation Setting)

Sampling mode	Execution timing	Sampling target	Reference
Standard sampling	Requesting the Read of the Attribute Service Set	Variables for the Read	Page 81 Processing when requesting the Read
	Performing a Subscription (in a cycle of a RevisedSamplingInterval)	Variables registered in MonitoredItems	Page 82 Processing when performing a Subscription
High-speed sampling (MonitoredItems only)*1*2	Requesting the Read of the Attribute Service Set	Variables for the Read	Page 81 Processing when requesting the Read
	Performing a Subscription (in a cycle for each sequence scan)	Variables registered in MonitoredItems	Page 84 Processing when using high-speed sampling
High-speed sampling (all Variables)*1	Each sequence scan	All Variables	

*1 Available for an OPC UA server module with firmware version '13' or later and the configuration tool with software version '1.13P' or later.

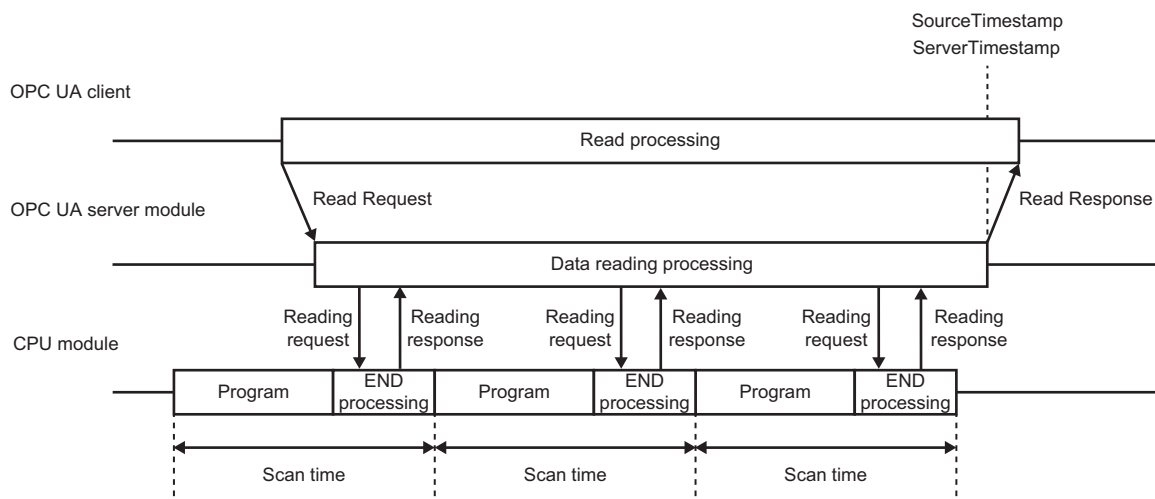
*2 A high-speed sampling target is changed when Creating, Deleting, or Modifying a MonitoredItem. High-speed sampling is stopped until the change is completed.

When the sampling mode is high-speed sampling, data is read for each sequence scan, and a value is set for a Variable.

Therefore, data is not read in a cycle of a RevisedSamplingInterval while performing a Subscription, and the current Variable value is sampled.

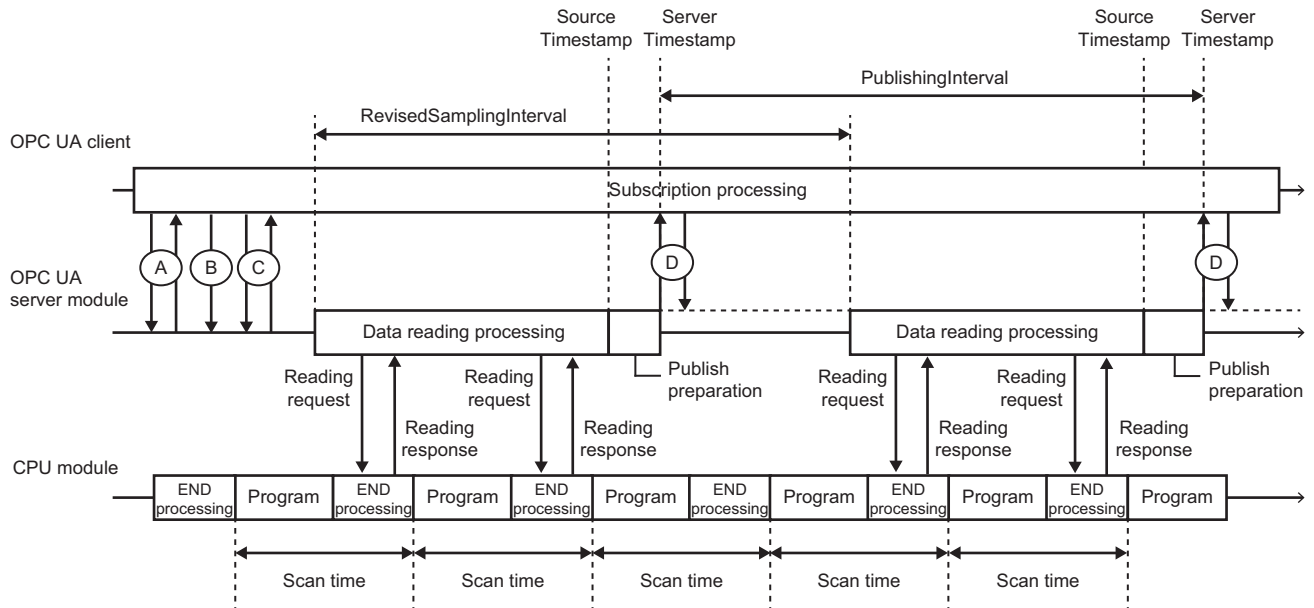
In addition, when the sampling mode is high-speed sampling (all Variables), data is not read when requesting the Read from an OPC UA client, and the current Variable value is returned.

■ Processing when requesting the Read



- When a Read Request is issued from an OPC UA client, an OPC UA server module requests a CPU module to read data mapped to a target Variable, and then returns a response to the OPC UA client based on the data reading response.
- The time when the data reading processing is completed is added to both the SourceTimestamp and ServerTimestamp as a time stamp added to the response to the OPC UA client.
- The OPC UA server module accesses the device memory during the END processing of the CPU module. The number of device points (access unit) that can be accessed in processing at a time is limited. If the limit is exceeded when reading data, the device memory may be accessed over multiple sequence scans. (➡ Page 86 Access unit and data inconsistency)

■ Processing when performing a Subscription



- A: CreateSubscription Request/Response
 B: Publish Request
 C: CreateMonitoredItems Request/Response
 D: Publish Response/Request

- When a Subscription request is received from an OPC UA client, data is periodically read in a cycle specified by the OPC UA client ($\text{SamplingInterval}^{*1}$).
- The data reading targets are Variables specified by using the CreateMonitoredItems, and data mapped to Variables with the same $\text{SamplingInterval}^{*1}$ specified is read in a batch.
- In the same manner as data reading when requesting the Read, the data may be read over multiple scans at a time. (Page 86 Access unit and data inconsistency)
- If a Variable value is changed due to the data reading, the value is reported to the OPC UA client with a Publish response. In this case, the time when the data reading processing is completed is added to the SourceTimestamp and the one when the Publish response is sent is added to the ServerTimestamp as a time stamp.
- Variables deleted from all MonitoredItems by using the DeleteMonitoredItems are excluded from the periodic data reading targets.

*1 Refers to a RevisedSamplingInterval corrected by an OPC UA server module.

SamplingInterval

A SamplingInterval is specified for each Node by an OPC UA client.

In addition, a SamplingInterval value is specified by an OPC UA client when registering a MonitoredItem by using the CreateMonitoredItems or ModifyMonitoredItems. ( Page 54 CreateMonitoredItems, Page 56 ModifyMonitoredItems)

The following values can be specified for a SamplingInterval.

- 50
- 100 to 900 (in 100 units)
- 1000 to 9000 (in 1000 units)
- 10000 to 60000 (in 10000 units)

If an unavailable value is specified, it is corrected to a RevisedSamplingInterval for operation.

No.	SamplingInterval [ms] before correction	RevisedSamplingInterval [ms] after correction
1	Negative number	The same value as a PublishingInterval ^{*1} . However, if the PublishingInterval value applies to No.2, it is corrected again.
2	Value other than the following: • 50 • 100 to 900 (in 100 units) • 1000 to 9000 (in 1000 units) • 10000 to 60000 (in 10000 units)	The closest value among the values listed below. Note that if the difference between two values is the same, the value is corrected to the smaller one. • 50 • 100 to 900 (in 100 units) • 1000 to 9000 (in 1000 units) • 10000 to 60000 (in 10000 units) (Example) • Before correction: 220 → after correction: 200 • Before correction: 250 → after correction: 200 • Before correction: 260 → after correction: 300

^{*1} Can be specified by using the CreateSubscription or ModifySubscription.

If a Node is registered in multiple MonitoredItems, data is read in a cycle with the smallest value among specified RevisedSamplingIntervals.

Note that Subscription processing is of the best effort type, and the operation may not be performed at a RevisedSamplingInterval corresponding to a specified SamplingInterval.

The minimum RevisedSamplingInterval and actual SamplingInterval used for a module according to a SamplingInterval requested from an OPC UA client can be checked in the following buffer memories:

Buffer memory name	Address
RevisedSamplingInterval (minimum) [ms]	Un\G12419 to 12420
Actual SamplingInterval (minimum) [ms]	Un\G12421 to 12422

If data mapped to a Variable is successfully read but the reading interval exceeds the RevisedSamplingInterval + 5%, the StatusCode for Good_Overload is set for the Variable Value.

■ Processing when using high-speed sampling

- Mapped data is read for each sequence scan.

The sequence scan synchronization sampling function of a CPU module^{*1} is used to read data from the control CPU.

For details on the execution timings and sampling targets, refer to the following:

📖 Page 80 Data reading

- The timestamps added to a response to an OPC UA client are as follows: the time when data is read by using the sequence scan synchronization sampling function for SourceTimestamp, and the time when data is sampled in a cycle of a RevisedSamplingInterval for ServerTimestamp.^{*2}
- If data reading succeeds but the high-speed sampling interval^{*3} exceeds the RevisedSamplingInterval, the StatusCode for Good_Overload is set for the Variable Value.
- When performing the Write, high-speed sampling of data mapped to a Write target Variable is temporarily stopped.

*1 Use a CPU module supporting the sequence scan synchronization sampling function (global label sampling). For the availability, refer to the manual for the CPU module used.

Note that the total number of mounted modules using the sequence scan synchronization sampling function for one CPU module must be four or fewer.

*2 For details on processing when requesting the Read for high-speed sampling (MonitoredItems only), refer to the following:

📖 Page 81 Processing when requesting the Read

*3 The high-speed sampling interval [ms] can be checked in the buffer memory. For details on the buffer memory, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

Precautions

- High-speed sampling can be expected to reduce the response time for the Read and Publish, but it is not a setting to guarantee that data is collected without missing any data for each sequence scan.
- In addition, high-speed sampling allows data to be collected faster than standard sampling, but the scan time increases depending on the data size.

Collectible devices

The following table shows the high-speed sampling availability for each device.

If an uncollectible device is assigned to a mapped global label, a sequence scan synchronization sampling function unsupported device error (error code: 1945H) occurs.

○: Available, ×: Not available

Device name (device)		Availability
Function input (FX)		×
Function output (FY)		×
Function register (FD)		×
Special relay (SM)		○
Special register (SD)		○
Input (X)		○
Output (Y)		○
Internal relay (M)		○
Latch relay (L)		○
Annunciator (F)		○
Edge relay (V)		○
Link relay (B)		○
Data register (D)		○
Link register (W)		○
Timer	Contact (TS)	○
	Coil (TC)	○
	Current value (T/TN)	○
Long timer	Contact (LTS)	○
	Coil (LTC)	○
	Current value (LT/LTN)	○
Counter	Contact (CS)	○
	Coil (CC)	○
	Current value (C/CN)	○
Long counter	Contact (LCS)	○
	Coil (LCC)	○
	Current value (LC/LCN)	○
Retentive timer	Contact (STS)	○
	Coil (STC)	○
	Current value (ST/STN)	○
Long retentive timer	Contact (LSTS)	○
	Coil (LSTC)	○
	Current value (LST/LSTN)	○
Link special relay (SB)		○
Link special register (SW)		○
Step relay (S)		×
Index register (Z)		○
Long index register (LZ)		○
File register	(R)	○
	(ZR)	○
Link direct device	Link input (Jn\X)	×
	Link output (Jn\Y)	×
	Link relay (Jn\B)	×
	Link special relay (Jn\SB)	×
	Link register (Jn\W)	×
	Link special register (Jn\SW)	×
Module access device (Un\G)		×
CPU buffer memory access device	(U3En\G)	×
	(U3En\HG)	×

Device name (device)	Availability
Refresh data register (RD)	○
Data-unassigned global label (GV)	○

■ Access unit and data inconsistency

For details on the number of data points (access unit) that can be accessed in processing at a time when accessing data in a CPU module, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

If the number of data points to be read exceeds the access unit, the processing is divided and the data is read over multiple sequence scans, which may cause data inconsistency.

In addition, the processing may be divided due to a cause other than the access unit.

Cause	Description
Data mapped to a Structure	When reading data mapped to a Field with the DataType of Structure, the data may be read over multiple sequence scans even if the number of device points to be read does not exceed the access unit, which may cause data inconsistency.

■ Operation when processing fails

- If data reading fails due to a global label-related data mismatch, the data is read again after updating the data related to the global label. For details on the processing, refer to the following:

📖 Page 89 Processing when reading or writing data

- Multiple units of data^{*1} are read at a time. If even one unit of data fails to be read, the other data is also treated as a failure, and Variable Values are not updated. For the StatusCodes returned to an OPC UA client if data reading fails when requesting the Read, refer to the following:

📖 StatusCode

- *1 For the Read: Nodes included in a Read Request are targets.
For a Subscription: Nodes with the same RevisedSamplingInterval are targets.

- If high-speed sampling fails, the following error codes are output.

Error code (Hexadecimal)	Output condition
1941H	The total number of access points for each module using the sequence scan synchronization sampling function of the same control CPU exceeds the maximum number of points. (The number of access points is calculated by rounding up in 8K points.)
1945H	An unsupported device for the sequence scan synchronization sampling function is used. ^{*2}
1946H	The control CPU of an OPC UA server module does not support the 'sequence scan synchronization sampling function for global labels.'

- *2 This error code is output when a device assigned to a global label is not supported for high-speed sampling.

For the list of high-speed sampling availability for devices, refer to the following:

📖 Page 84 Processing when using high-speed sampling

Note that device-unassigned global labels can be used.

For details on the error codes, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

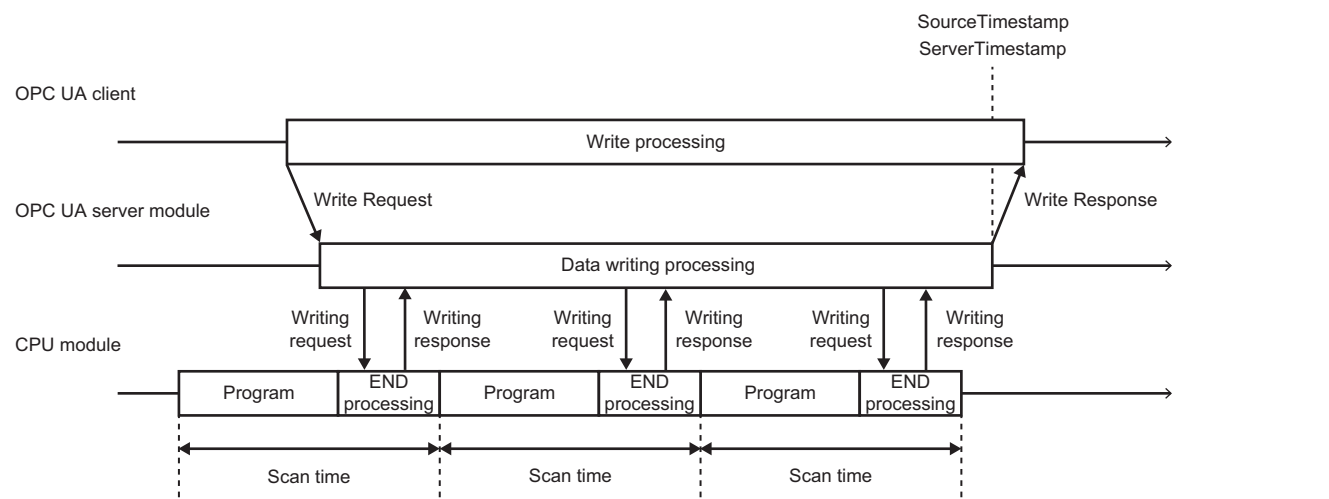
Data writing

This processing is for applying Node data to data in a CPU module.

The execution timing of data writing is as follows:

Execution timing	Target Node	Reference
Requesting the Write of the Attribute Service Set	Variables for the Write	Page 87 Processing when requesting the Write

■Processing when requesting the Write



- When a Write Request is issued from an OPC UA client, an OPC UA server module requests a CPU module to write data to data mapped to a target Variable, and then returns a response to the OPC UA client based on the data writing response.
- When the data writing processing succeeds, the written value is set for the Variable. In this case, the time when the data writing processing is completed is added to both the SourceTimestamp and ServerTimestamp as a time stamp.
- The OPC UA server module accesses data in the CPU module during the END processing of the CPU module. The number of data points (access unit) that can be accessed in processing at a time is limited. If the limit is exceeded when writing data, the data may be accessed over multiple sequence scans. (👉 Page 87 Access unit and data inconsistency)

■ Access unit and data inconsistency

For details on the number of data points (access unit) that can be accessed in processing at a time when accessing data in a CPU module, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

If the number of data points to be written exceeds the access unit, the processing is divided and the data is written over multiple sequence scans, which may cause data inconsistency.

In addition, the processing may be divided due to a cause other than the access unit.

Cause	Description
Data mapped to a Structure	When writing data to data mapped to a Field with the DataType of Structure, the data may be written over multiple sequence scans even if the number of device points to be written does not exceed the access unit, which may cause data inconsistency.

■ Operation when processing fails

- If data writing fails due to a global label-related data mismatch, the data is written again after updating the data related to the global label. For details, refer to the following:

Page 89 Processing when reading or writing data

- Multiple units of data*1 are written at a time. If even one unit of data fails to be written, the other data is also treated as a failure, and Variable Values and data values in a CPU module are not updated. For the StatusCodes returned to an OPC UA client if data writing fails when requesting the Write, refer to the following:

 StatusCode

*1 Nodes included in a Write Request are targets.

Global label-related data update

This processing is for updating the addresses and data types of global labels.

It allows the operation to continue without changing the setting of an OPC UA server module even if the settings such as labels, programs, FBs, and CPU parameters of a CPU module are changed.

To perform this processing, the following information including global labels mapped in the configuration tool must be written to a CPU module in GX Works3 in advance.

- Global label setting
- Global label assignment information

In addition, the following must be set for mapping target global labels in the global label setting of GX Works3.

- Select "Access from External Device."
- Select a class other than "VAR_GLOBAL_CONSTANT."

Whether updating related data has started or completed can be checked with an event (related data updating start (event code: 24000H) or related data updating finish (event code: 24001H)), and whether related data is being updated can be checked in the buffer memory ('Label update running status' (Un\G8617)). For the event code list and details on the buffer memory, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

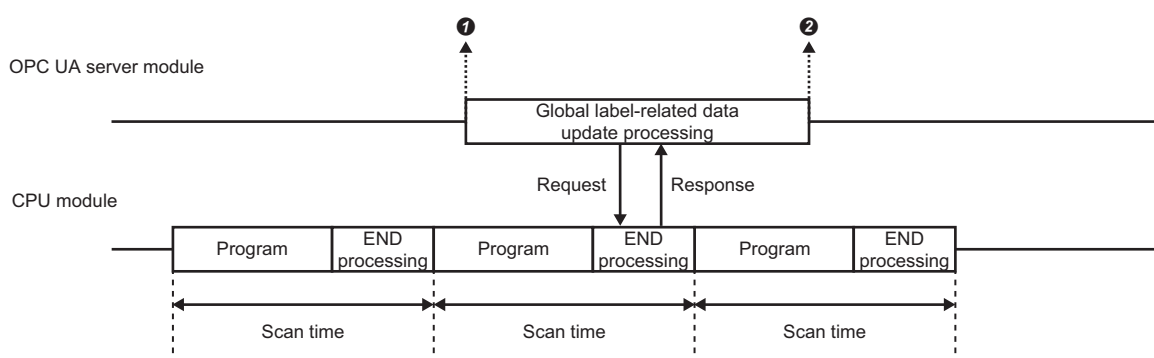
The execution timings of global label-related data updates are as follows:

Execution timing	Target Node	Reference
Starting a module	All Variables	Page 88 Processing when starting, restarting an OPC UA server module, or updating its setting
Updating the setting of a module		
Restarting a module		
Reading data		Page 89 Processing when reading or writing data
Writing data		

Note that if changing a global label name, data before the change remains in the global label assignment information of a CPU module even when performing the online program change; therefore, it is recommended to write the global label assignment information to the CPU module in the "Online Data Operation" screen of GX Works3. Otherwise, a global label unavailable error (error code: 1921H) does not occur even when using the label name before the change, and the label after the name change is accessed.

Precautions for the online program change can be checked by clicking the [Precautions] button in the "Online Program Change" screen of GX Works3.

■ Processing when starting, restarting an OPC UA server module, or updating its setting



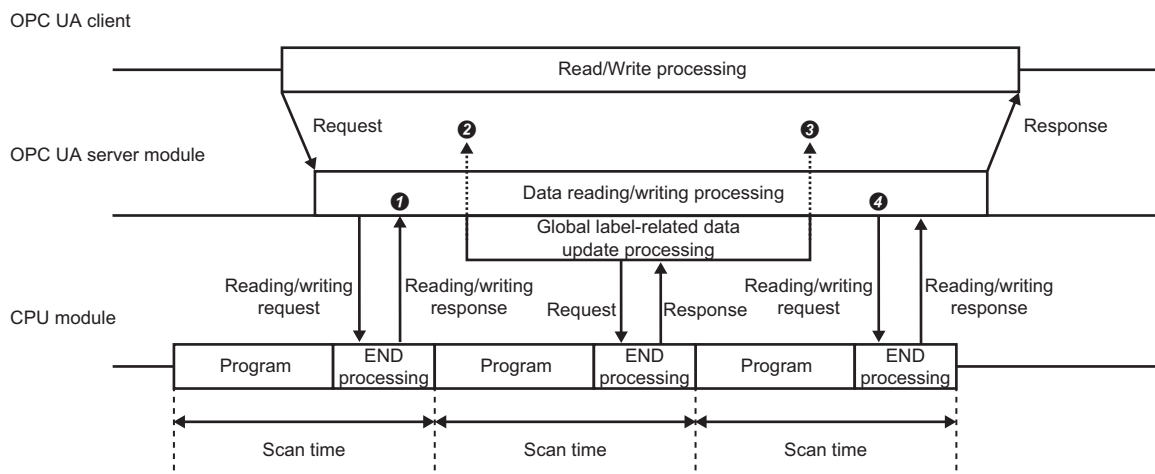
①: Related data updating start (event code: 24000H)

②: Related data updating finish (event code: 24001H)*1

*1 If the processing fails, a related data updating failure (event code: 24003H) is output.

- When starting, restarting an OPC UA server module, or updating its setting, the addresses and data types of global labels are acquired. Global labels linked to all Variables are acquisition targets.
- If not using any global label, this processing is not performed.

■Processing when reading or writing data



- ①: A related data mismatch detected
 ②: Related data updating start (event code: 24000H)
 ③: Related data updating finish (event code: 24001H)*1
 ④: Data reading or writing processing performed again

*1 If the processing fails, a related data updating failure (event code: 24003H) is output.

- If a global label-related data mismatch is detected in data reading or data writing, the related data is updated and the addresses and data types of the global labels are reacquired. Global labels linked to all Variables are acquisition targets.
- When updating the related data is completed normally, the data is read or written again.
- If accessing from an OPC UA client and updating global label-related data while writing to a programmable controller, a related data updating failure (event code: 24003H) may be output. In this case, access from the OPC UA client again after writing to the programmable controller is completed, then values can be read or written normally when a related data updating finish (event code: 24001H) is output.

■Operation when processing fails

If updating data related to a global label fails, the following error codes are output.

For details on the error codes, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

Error code (Hexadecimal)	Output condition
1921H	• A mapped global label is not written to a CPU module. • "Access from External Device" is not selected in the global label setting of GX Works3. • The class is "VAR_GLOBAL_CONSTANT" in the global label setting of GX Works3. • Neither the global label setting nor global label assignment information is written to a CPU module.
1A00H	The data type of a label and the DataType of a Variable are mapped in an unavailable combination.
1A01H	A label value read from a CPU module is out of the range of mapped Variables.

Error codes 1921H and 1A00H are caused by incorrect settings and require setting changes to recover. Therefore, the StatusCode for Bad_NoCommunication is returned in response to the Read, Write, and Publish even for normal combinations of Nodes. However, for a Node with no label assigned, the StatusCode for Good is returned in response to the Publish.

■Differences in operations from the tag mode

The following table shows the differences in global label-related data update processing between the information model mode and the tag mode.

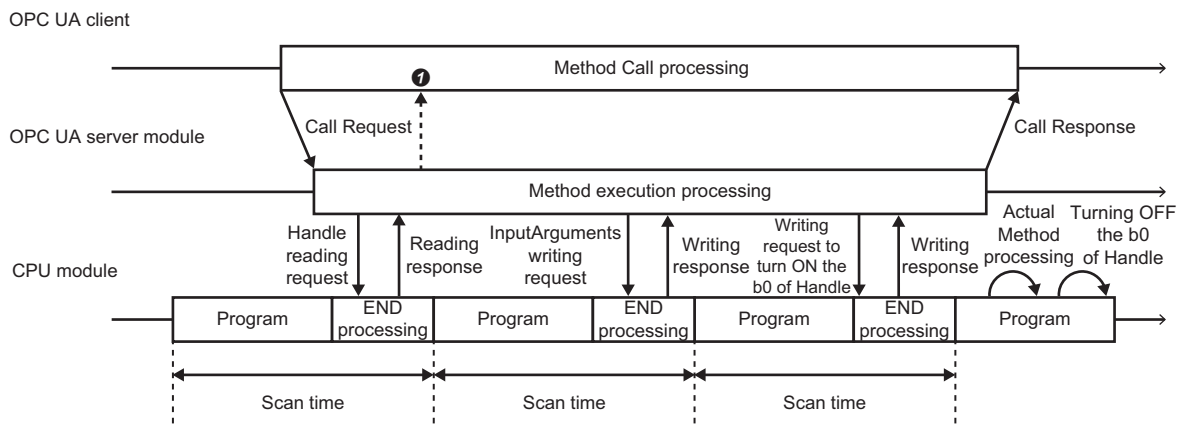
Information model mode	Tag mode
Related data is always updated when starting, restarting an OPC UA server module, or updating its setting.	Related data is updated only if there is a mismatch when comparing the setting when importing global labels in the configuration tool with that of a CPU module.
Information on the data type is acquired from a CPU module. (The operation follows the changed data type acquired from the CPU module.)	Even if the data type of a global label is changed with a related data update, the operation is performed as if it was the data type set in the address space (tag) setting of the configuration tool.

Method execution

This processing is for notifying a CPU module of the start of a Method in response to a Call request from an OPC UA client. The actual processing of a Method is performed in a program of a CPU module prepared by a user. The execution timing of Method execution processing is as follows:

Execution timing	Target Node	Reference
Requesting the Call of the Method Service Set	Methods for the Call	Page 90 Processing when requesting the Call

■Processing when requesting the Call



①: If the 0th bit of a Handle is ON, an error response is returned and the processing ends.

- When a Call Request is issued from an OPC UA client, an OPC UA server module requests a CPU module to read data mapped to the Handle.
- If the 0th bit of the Handle is ON, the same Method is regarded as already being executed, and the StatusCode for `Bad_TooManyOperations` is returned to the OPC UA client.
- If the 0th bit of the Handle is OFF, data mapped to InputArguments is written, then the bit turns ON.
- When the 0th bit of the Handle turns ON, the CPU module uses the values of InputArguments as necessary to perform the actual processing of the Method. Once the processing is completed, the bit turns OFF in the program.

For a program example, refer to the following:

📖 Page 91 Program example

■Operation when processing fails

- If data reading or writing fails due to a global label-related data mismatch, the data is read or written again after updating the data related to the global label. For details, refer to the following:

📖 Page 89 Processing when reading or writing data

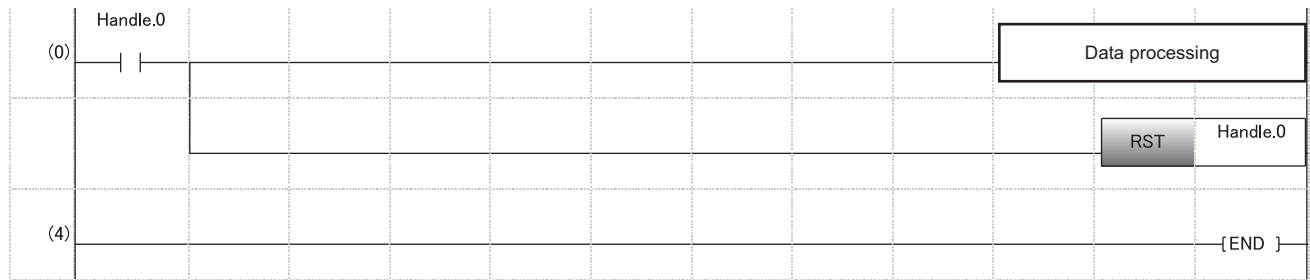
- As mentioned above (📖 Page 90 Processing when requesting the Call), if the 0th bit of the Handle is already ON, the same Method is regarded as being executed, and the StatusCode for `Bad_TooManyOperations` is returned to the OPC UA client.

■Program example

Global label used in a program

Global label name	Data type
Handle	Word [unsigned]/bit string [16-bit]

Program example



- (0) The 0th bit of a Handle turns ON when requesting the Call from an OPC UA client, and data processing is performed. After the data processing is completed, the 0th bit of the Handle turns OFF so that the Call can be requested again for the same Method.

8.4 OPC UA Server Function

The OPC UA server function is for allowing an OPC UA server module to accept connections from an OPC UA client, such as network and authentication (security).

Function list

The OPC UA server function consists of the following items:

Item	Description	Reference
User	To manage users of an OPC UA server module.	Page 122 User Setting
Network	To set the IP address, subnet mask, etc. of an OPC UA server module.	Page 125 Network Setting
Security	To set authentication when accessing an OPC UA server module from the configuration tool or an OPC UA client.	Page 126 Security Setting
Server certificate	To set the contents of a server certificate of an OPC UA server module used for application authentication.	Page 129 Server Certificate Setting
Discovery server	To specify a discovery server to register an OPC UA server module.	Page 130 Discovery Server Setting
Discovery Service Set	To provide the functions of the Discovery Service Set of the OPC UA specifications.	Page 93 Service Set
SecureChannel Service Set	To provide the functions of the SecureChannel Service Set of the OPC UA specifications.	
Session Service Set	To provide the functions of the Session Service Set of the OPC UA specifications.	

Service Set

The following table shows the Service Sets and Services related to the OPC UA server function supported by this product.

Service Set	Service	Reference
Discovery Service Set	FindServers	—
	GetEndpoints	
	RegisterServer* ¹	Page 93 RegisterServer
	RegisterServer2* ¹	Page 93 RegisterServer2
SecureChannel Service Set	OpenSecureChannel	Page 93 SecureChannel Service Set
	CloseSecureChannel	
Session Service Set	CreateSession	Page 93 Session Service Set
	ActivateSession	
	CloseSession	

*¹ Requests from an OPC UA server module to a discovery server are supported. (Ones to an OPC UA server module are not supported.) For details on each Service Set and Service, refer to the OPC UA specifications.

The following explains the details on the specifications unique to this product for each Service.

Discovery Service Set

Services are provided to detect Endpoints implemented by an OPC UA server module and read the security configurations for the Endpoints (security policy, security mode, etc.).

■RegisterServer

A function is supported for an OPC UA server module to issue a RegisterServer Request to a discovery server and register its own information in the discovery server. Specify a discovery server for registration in the discovery server setting of the configuration tool. (📖 Page 130 Discovery Server Setting)

A RegisterServer2 Request is first used for registration in a discovery server. If a Bad_ServiceUnsupported response is returned from the discovery server, the registration is attempted again by using a RegisterServer Request.

Note that RegisterServer Requests to an OPC UA server module are not supported.

■RegisterServer2

A function is supported for an OPC UA server module to issue a RegisterServer2 Request to a discovery server and register its own information in the discovery server. Specify a discovery server for registration in the discovery server setting of the configuration tool. (📖 Page 130 Discovery Server Setting)

If a Bad_ServiceUnsupported response is returned from the discovery server, the registration is attempted again by using a RegisterServer Request.

Note that RegisterServer2 Requests to an OPC UA server module are not supported.

SecureChannel Service Set

Services are provided to open communication channels that ensure the confidentiality and integrity of all messages exchanged with an OPC UA server module.

For supported security policies and security modes, refer to the following:

📖 Page 126 [SecureChannel] tab

Session Service Set

Services are provided to authenticate users and establish sessions.

For supported user authentication methods, refer to the following:

📖 Page 127 [User Authentication] tab

8.5 Operation Function

The operation function is for performing various communications between an OPC UA server module and the configuration tool, such as reading, writing, or updating a setting, and managing electronic certificates.

Function list

The operation function consists of the following items:

Item	Description	Reference
Module control	To perform operations on an OPC UA server module, such as reading, writing, or verifying a project file, and updating a setting.	Page 133 Module Control
Module diagnostics	To provide operations for displaying the operating status of an OPC UA server module, setting the firmware update prohibited state, displaying the SD memory card status, and formatting an SD memory card.	Page 134 Module Diagnostics
Electronic certificate management	To manage user certificates and application certificates used for OPC UA communication.	Page 135 Electronic Certificate Management
Setting file management	To manage a setting file for determining the operation of an OPC UA server module.	—
SD memory card format function	To format an SD memory card inserted in an OPC UA server module.	MELSEC iQ-R OPC UA Server Module User's Manual (Application)
Firmware update function	To update the firmware of an OPC UA server module.	
Initialization function	To initialize a firmware update-prohibited password retained in an OPC UA server module.	
Self-diagnostics function	To perform a self-diagnostic test to check the hardware of an OPC UA server module.	
Online module change function	To replace a module with another without stopping a running system. For the procedure, refer to the manual in the right column.	MELSEC iQ-R Online Module Change Manual

8.6 File Input/Output Function

The file input/output function is for operating various files related to the information model mode.

The following files can be used in the configuration tool.

Item	Description	Reference
Information model mode project file (.mxcfg2)	A project file of the configuration tool. Use one to save and open a project.	Page 96 Project File Handling
Information model file (.xml)	An XML file in which information models are described. Use one to import Nodes for constructing an AddressSpace and to directly describe mapping information of the Nodes.	Page 154 XML File Format
GX Works3 project file (.gx3)	A GX Works3 project file. Use one to import global labels used for global label linking.	Page 116 Global Label Linking

9 MX OPC UA Module Configurator-R

This chapter explains MX OPC UA Module Configurator-R.

9.1 MX OPC UA Module Configurator-R

MX OPC UA Module Configurator-R (information model mode) is a tool for configuring the setting to utilize information models, which are provided by the OPC Foundation or created by users, in a programmable controller.

The following is the main function.

Function	Description
Mapping (mapping)	To link information models to global labels in a programmable controller. Nodes published to an AddressSpace can be imported from an information model file (XML file), and data can be set (mapped) for each Node.

For details on the screen configuration of MX OPC UA Module Configurator-R, refer to the following:

 Page 35 Screen configuration

For using the tag mode and details on common specifications with the tag mode, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

9.2 Operating Procedure

Refer to the following:

 Page 26 Setting procedure for an OPC UA server module

9.3 Project File Handling

This section describes the details on handling project files.

Project file compatibility

The following combinations are not compatible. When reading a project file from an OPC UA server module, start the configuration tool in a mode supporting the project.

- Project file in the tag mode and the configuration tool in the information model mode
- Project file in the information model mode and the configuration tool in the tag mode

Creating a new project

Create a new project.

Operating procedure

Select [File] ⇒ [New].

Opening a project

Read a saved project.

Precautions

Opening a project takes time according to the number of Nodes due to decoding information models and mapping information.

(Example) If the number of Nodes included in an information model file exceeds 10000, it may take one minute or more.

Selecting and opening a project

Operating procedure

1. Select [File] ⇒ [Open].
2. Select a target file, and click the [Open] button.

Opening a file used recently

Operating procedure

Select [File] ⇒ [Recent Projects] to select a file to open.

Dragging and dropping a project to open

Operating procedure

Drag and drop a project file onto the configuration tool.

Saving a project

Save a setting being edited to a project file.

Overwriting and saving a project

Operating procedure

Select [File] ⇒ [Save].

Saving a project under a new name

Operating procedure

1. Select [File] ⇒ [Save As].
2. Specify a save destination and file name, and click the [Save] button.

9.4 Node Setting

This section shows the screen for setting Nodes published to an AddressSpace.

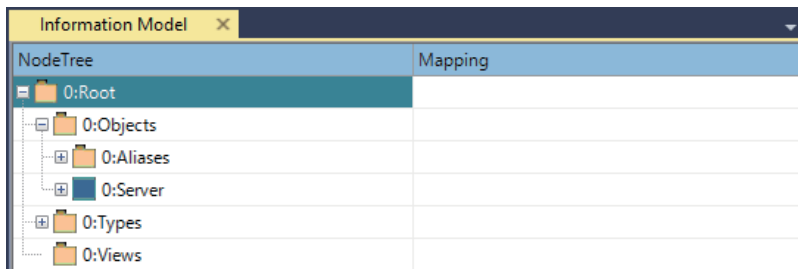
NodeTree

Nodes published to an AddressSpace are displayed in a tree in the main window.

Mapping can be set in this screen. ( Page 110 Mapping)

Window

Select "Information Model" in the navigation tree.



Displayed items

Item	Description
NodeTree	The BrowseNames of Nodes published to an AddressSpace are displayed in a tree.* ¹ When a Node (Variable) is a Structure or Array, its Field* ² or ArrayElements* ³ are displayed in a tree. "Handle" is displayed immediately under a Node of Method. The Names of Arguments are displayed for Arguments under InputArguments and OutputArguments of a Method.* ⁴ Label Nodes (Objects) and label Nodes (Variables) created by using the label Node builder function can be moved and deleted by dragging and dropping. For details, refer to the following:  Page 99 Moving and deleting label Nodes
DisplayName	The DisplayNames of Nodes are displayed.
NodeClass	The NodeClasses of Nodes are displayed.
DataType (Node)	The DataTypes of Nodes (Variables) are displayed. The DataTypes of Arguments are displayed for Arguments under InputArguments and OutputArguments of a Method.
TypeDefinition	The TypeDefinitions of Nodes (Objects and Variables) are displayed.
ValueRank	The ValueRanks of Nodes (Variables) are displayed. The ValueRanks of Arguments are displayed for Arguments under InputArguments and OutputArguments of a Method.
ArrayDimensions	The ArrayDimensions of Nodes (Variables) are displayed. The ArrayDimensions of Arguments are displayed for Arguments under InputArguments and OutputArguments of a Method.
AccessLevel	The AccessLevels of Nodes (Variables) are displayed.
Description	The Descriptions of Nodes are displayed. The Descriptions of Arguments are displayed for Arguments under InputArguments and OutputArguments of a Method.
Mapping	Enter data to be mapped. ( Page 110 Mapping)

- *¹ If importing an invalid XML file (not compliant with the OPC UA specifications) such as the examples below, the NodeTree screen may not be displayed properly.
- The same NodeId is used for different Nodes.
 - The same Name is used for Fields of one Structure.
- *² The Names of Fields are displayed.
- *³ The [{numbers of elements}] for the numbers of dimensions according to the ValueRank are displayed in order of [One dimension], [Two dimensions], and [Three dimensions] from the left. For details, refer to the following:
 Page 113 Array mapping
- *⁴ The rows are displayed in order of the element numbers of Arguments. For example, when ArgB, ArgA, and ArgC are defined in [0], [1], and [2] of InputArguments respectively, they are displayed in the order of ArgB, ArgA, and ArgC from top to bottom.



- An item not displayed by default is displayed by right-clicking the column title and selecting the item.
- When right-clicking each cell, the operation menus are displayed.
 - [Mapping Array Specification] can be selected when the ValueRank and ArrayDimensions can be assigned with limitations. (Page 64 Array assignment method)
 - [Node Delete] can be selected when selecting a label Node (Object) or label Node (Variable) in the NodeTree column.

Moving and deleting label Nodes

Label Nodes (Objects) and label Nodes (Variables) can be moved and deleted.



Nodes created by using the label Node builder function are collectively referred to as label Nodes.

For details on the label Node builder function, refer to the following:

Page 49 Label Node builder

■ Move

Label Nodes (Objects) and label Nodes (Variables) can be moved by dragging and dropping. Nodes under the moved targets are also moved.

○: Movable, —: Not movable

Move destination (The values in the brackets [] indicate NodeIds.)	Move target	
	Label Node (Variable)	Label Node (Object)
Immediately under Root [i=84]	—	—
Immediately under Objects [i=85]	○	○
Under Aliases [i=23470]	—	—
Under Server [i=2253]	—	—
Object under Objects [i=85]	○	○
Variable under Objects [i=85]	○	—
Label Node (Field) under Objects [i=85]	—	—
Label Node (Element) under Objects [i=85]	—	—
Method under Objects [i=85]	—	—
InputArguments and OutputArguments of a Method	—	—
NodeClass under Objects [i=85] other than the above	—	—
Under Types [i=86]	—	—
Under Views [i=87]	—	—

■ Delete

Label Nodes (Objects) and label Nodes (Variables) can be deleted from NodeTree by the following operation. Nodes under the deleted targets are also deleted.

Operating procedure

Select and right-click a target label Node in the NodeTree column (multiple selections allowed), then select [Node Delete] from the shortcut menu. (The label Node can also be deleted by pressing the key.)

Base Attributes

Attributes common to the NodeClass among those of a Node selected in the NodeTree screen are displayed in the [Base Attributes] tab in the subwindow.*1*2

For the specifications of each Attribute, refer to the OPC UA specifications.

- *1 If multiple Nodes exist immediately under a selected Node, it may take time to display the Base Attributes screen.
- *2 If importing an invalid XML file (not compliant with the OPC UA specifications), the Base Attributes screen may not be displayed properly.

Window

Base Attributes	
Base Attributes	Value
NodeId	i=2253
NodeClass	Object
BrowseName	Server
DisplayName	Server
Description	
WriteMask	0(None)
UserWriteMask	0(None)
RolePermissions	

Displayed items

Item	Description
NodeId	Each Attribute value of a Node selected in the NodeTree screen is displayed.
NodeClass	
BrowseName	
DisplayName*3	
Description*3	
WriteMask	
UserWriteMask	
RolePermissions	

*3 The Locale is not supported.

NodeClass Attributes

Attributes specific to the NodeClass among those of a Node selected in the NodeTree screen are displayed in the [NodeClass Attributes] tab in the subwindow.*1*2

For the specifications of each Attribute, refer to the OPC UA specifications.

- *1 If multiple Nodes exist immediately under a selected Node, it may take time to display the NodeClass Attributes screen.
- *2 If importing an invalid XML file (not compliant with the OPC UA specifications), the NodeClass Attributes screen may not be displayed properly.

NodeClass: Object

The following appears when the NodeClass of a Node selected in the NodeTree screen is Object.

Window

NodeClass Attributes	
NodeClass Attributes	Value
EventNotifier	0(None)

Displayed items

Item	Description
EventNotifier	An EventNotifier value is displayed.

NodeClass: Variable

The following appears when the NodeClass of a Node selected in the NodeTree screen is Variable.

Window

NodeClass Attributes	
NodeClass Attributes	Value
Value	
DataType	Boolean
ValueRank	2
ArrayDimensions	2,3
AccessLevel	3(CurrentRead,CurrentWrite)
UserAccessLevel	3(CurrentRead,CurrentWrite)
MinimumSamplingInterval	0
Historizing	False

Displayed items

Item	Description
Value *1*2*3	Each Attribute value is displayed.
DataType	
ValueRank	
ArrayDimensions	
AccessLevel	
UserAccessLevel	
MinimumSamplingInterval	
Historizing	

- *1 May not be displayed correctly if the DataType is other than one of Built-in DataTypes that can be assigned (excluding Enumeration and Structure).
- *2 When the DataType is Float or Double, the display is as follows:
 - Float: The number of significant digits in the mantissa part is 7 digits. If it is exceeded, the 7th digit is displayed as a rounded number.
 - Double: The number of significant digits in the mantissa part is 15 digits. If it is exceeded, the 15th digit is displayed as a rounded number.
- *3 Displayed in a JSON format when the DataType is Structure.

NodeClass: Method

The following appears when the NodeClass of a Node selected in the NodeTree screen is Method.

Window

NodeClass Attributes	
NodeClass Attributes	Value
Executable	True
UserExecutable	True

Displayed items

Item	Description
Executable	Each Attribute value is displayed.
UserExecutable	

NodeClass: ObjectType

The following appears when the NodeClass of a Node selected in the NodeTree screen is ObjectType.

Window

NodeClass Attributes	
NodeClass Attributes	Value
IsAbstract	False

Displayed items

Item	Description
IsAbstract	An IsAbstract value is displayed.

NodeClass: VariableType

The following appears when the NodeClass of a Node selected in the NodeTree screen is VariableType.

Window

NodeClass Attributes	
NodeClass Attributes	Value
Value	
DataType	BaseDataType
ValueRank	0(OneOrMoreDimensions)
ArrayDimensions	
IsAbstract	True

Displayed items

Item	Description
Value*1*2	Each Attribute value is displayed.
DataType	
ValueRank	
ArrayDimensions	
IsAbstract	

*1 When the DataType is Float or Double, the display is as follows:

- Float: The number of significant digits in the mantissa part is 7 digits. If it is exceeded, the 7th digit is displayed as a rounded number.
- Double: The number of significant digits in the mantissa part is 15 digits. If it is exceeded, the 15th digit is displayed as a rounded number.

*2 Displayed in a JSON format when the DataType is Structure.

NodeClass: ReferenceType

The following appears when the NodeClass of a Node selected in the NodeTree screen is ReferenceType.

Window

NodeClass Attributes	
NodeClass Attributes	Value
IsAbstract	True
Symmetric	True
InverseName	

Displayed items

Item	Description
IsAbstract	Each Attribute value is displayed.
Symmetric	
InverseName	

NodeClass: DataType

The following appears when the NodeClass of a Node selected in the NodeTree screen is DataType.

Window

NodeClass Attributes	
NodeClass Attributes	Value
IsAbstract	True

Displayed items

Item	Description
IsAbstract	An IsAbstract value is displayed.

Structure Fields

When selecting a target Node^{*1} in the NodeTree screen, the contents of Structure Fields are displayed in the [Structure Fields] tab in the subwindow.

For the specifications of Structure Fields, refer to the OPC UA specifications.

- *1 The following type of Node is a target.
- Subtype of Structure with the NodeClass of DataType

Window

Structure Fields						
Name	DataType	ValueRank	ArrayDimensions	MaxStringLength	IsOptional	Description
ParentNodeId	ExpandedNodeId	-1(Scalar)		0	False	
ReferenceTypeId	NodeId	-1(Scalar)		0	False	
RequestedNewNodeId	ExpandedNodeId	-1(Scalar)		0	False	
BrowseName	QualifiedName	-1(Scalar)		0	False	
NodeClass	NodeClass	-1(Scalar)		0	False	
NodeAttributes	Structure	-1(Scalar)		0	False	
TypeDefinition	ExpandedNodeId	-1(Scalar)		0	False	

Displayed items

Item	Description
Name	The names of fields are displayed.
DataType	The data types of fields are displayed.
ValueRank	The numbers of dimensions of fields are displayed.
ArrayDimensions	The numbers of elements are displayed when fields are arrays.
MaxStringLength	The maximum lengths of character strings are displayed when the data types of fields are String or ByteString.
IsOptional	Whether fields are optional is displayed.
Description	The descriptions of fields are displayed.

Enumeration Values

When selecting a target Node^{*1} in the NodeTree screen, the contents of Enumeration Values are displayed in the [Enumeration Values] tab in the subwindow.

- *1 The following type of Node is a target.
- Subtype of Enumeration with the NodeClass of DataType (EnumStrings or EnumValues)

Window

Enumeration Values	
EnumString	Value
Server	0
Client	1
ClientAndServer	2
DiscoveryServer	3

Displayed items

Item	Description
EnumString	The EnumStrings of Enumerations are displayed.
Value	Enumeration Values are displayed.

References

The Hierarchical References and Non-Hierarchical References of a Node selected in the NodeTree screen are displayed in the [References] tab in the subwindow.

By double-clicking a References row, the cursor jumps to the References destination Node.

Window

References						
Hierarchical References						
IsForward	ReferenceType	NodeClass	Name	DataType	TypeDefinition	ModellingRule
True	Organizes	Object	Objects		FolderType	
True	Organizes	Object	Types		FolderType	
True	Organizes	Object	Views		FolderType	
Non-Hierarchical References						
IsForward	ReferenceType	Name				
True	HasTypeDefinition	FolderType				

Displayed items

○: Displayed, —: Not displayed

Item	Description	Hierarchical References	Non-Hierarchical References
IsForward	The IsForward of a Reference is displayed.	○	○
ReferenceType	The ReferenceType of a Reference is displayed.	○	○
NodeClass	The NodeClass of a Reference target Node is displayed.	○	—
Name	The name of a Reference target Node is displayed.	○	○
DataType	The DataType of a Reference target Node is displayed.	○	—
TypeDefinition	The TypeDefinition of a Reference target Node is displayed.	○	—
ModellingRule	The ModellingRule of a Reference target Node is displayed.	○	—

Importing an information model file

A Namespace and Nodes can be imported from an information model file (XML file in which information models are described).

In addition, mapping information in the information model file can be imported together.

For the XML file format, refer to the following:

☞ Page 154 XML File Format

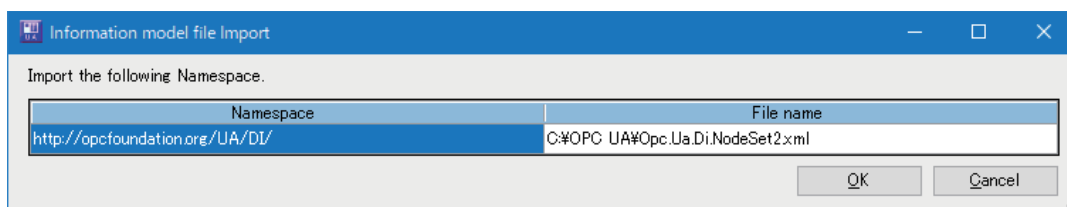
Precautions

- An information model file containing multiple Namespaces cannot be imported. Save an information model file for each Namespace in a modeling tool, then import it individually.
- If an information model file to be imported references another information model file, import the referenced information model file first.
(This is because the Namespace in the reference destination information model file is defined in the RequiredModel in the reference source information model file.)
- Import takes time according to the number of Nodes due to decoding information models and mapping information.
(Example) If the number of Nodes included in an information model file exceeds 10000, it may take one minute or more.
- Depending on the contents of an information model file and the mapping status in the configuration tool, data may be mapped to a Node that cannot be mapped. In this case, right-click the Node and select [Clear] from the shortcut menu to delete mapping information.
- An information model file with the Namespace MELNS cannot be imported.

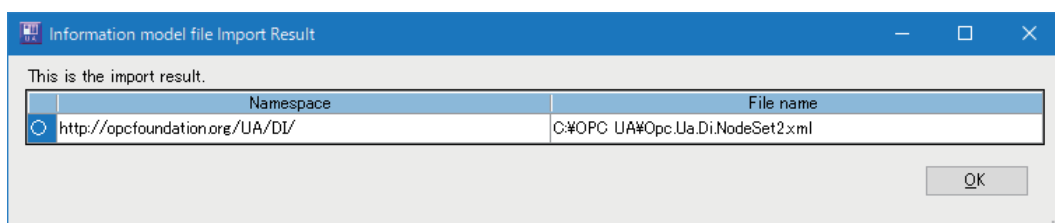
Import procedure

Operating procedure

1. Select [File] ⇒ [Import] ⇒ [Information Model File].
2. Select a file to import, and click the [Open] button.
3. Check the Namespace to import, and click the [OK] button.



4. Check the import result, and click the [OK] button.



Node setting and mapping information handling

If the Namespace is the same between an import source and import destination, the Node setting and mapping information handling after import differ depending on differences in the Node setting and whether mapping information is included.

■Handling according to whether mapping information is included in an import source and import destination

Mapping information to be retained after import differs depending on whether mapping information is included in an import source and import destination.

Mapping information		
Import source (Import target XML file)	Import destination (Configuration tool)	After import
Not included	Not included	Not included
Not included	Included	Mapping information in an import destination
Included	Not included	Mapping information in an import source
Included	Included	

■If there is no corresponding Node in an import source

The corresponding Node is deleted after import regardless of whether mapping information is included.

■If the NodeClass is different between an import source and import destination

- When it is Variable in an import source

Mapping information to be retained after import is determined according to whether it is included. ( Page 108 Handling according to whether mapping information is included in an import source and import destination)

- When it is not Variable in an import source

Mapping information of the corresponding Node is deleted after import.

■If the Attribute is different between an import source and import destination

Mapping information to be retained after import is determined according to whether it is included. ( Page 108 Handling according to whether mapping information is included in an import source and import destination)

Note that if the contents in the following Attributes are different in particular, errors or warnings are likely to be output during import.

- DataType
- ValueRank
- ArrayDimensions
- Structure
- StructureFields

Exporting an information model file

Nodes in a Namespace can be exported to an information model file.

In addition, mapping information in the Node setting can be exported together.

Export procedure

Operating procedure

1. Select [File] ⇒ [Export] ⇒ [Information Model File].
2. Select a Namespace to export, edit an information model file name^{*1*2*3}, and specify an export destination, then click the [OK] button.

*1 The default file name is the last segment delimited by a slash (/) in the NamespaceURI. If a prohibited character is included, it is deleted from the file name.

(Example) When the NamespaceURI is 'http://MyCompany/ABC,' the file name is 'ABC.'

*2 A same file name cannot be used. (Multiple Namespaces cannot be exported to one information model file.)

When the default file name is duplicated, '_1' is added to the end of the file name. The number is incremented to avoid duplication with other file names.

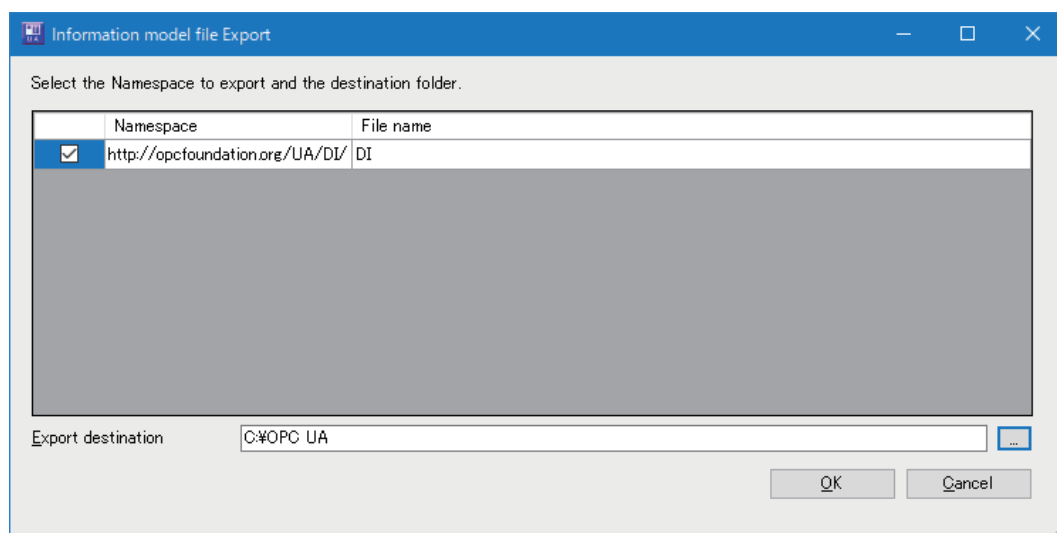
*3 When the default file name is either of the following, the file name is '_1.' The number is incremented to avoid duplication with other file names.

- Prohibited character string (reserved word)

- Null character or prohibited character string (reserved word) due to deletion of a prohibited character

For details on the prohibited characters and prohibited character strings (reserved words), refer to the following:

📖 Page 147 Available Characters



9.5 Mapping

This section describes the details on setting (mapping) global labels used for a CPU module for Nodes in an imported information model file.

Map global labels to Nodes manually. Multiple units of mapping data can be entered in a batch by pasting them from the clipboard or using Excel. (🔗 Page 112 Mapping operations (copy, cut, paste))

Note that Variables created by using the label Node builder function are automatically mapped.

Mapping targets

Mapping is available for the following items under Root > Objects in NodeTree.

- Variable
- Handle of a Method
- Argument under InputArguments of a Method

However, it is not available for Nodes that cannot be mapped. (🔗 Page 110 Targets that cannot be mapped)

In addition, mapping data for Variables created by using the label Node builder function cannot be edited.

■Targets that cannot be mapped

The following targets cannot be mapped.

- Nodes with the Namespace of "0: http://opcfoundation.org/UA/"
- Nodes not under Objects (ns=0;i=85)
- Nodes under Objects > Server (ns=0;i=2253)
- Nodes under Objects > Aliases (ns=0;i=23470)
- Nodes in a Namespace used as a RequiredModel in an information model other than MELNS*¹
- InputArguments and OutputArguments of a Method, and Arguments under OutputArguments
- Data that cannot be assigned in the following:
 - 🔗 Page 60 DataType assignment
 - 🔗 Page 65 Structure assignment
 - 🔗 Page 67 Enumeration assignment
- Data that conflicts with the restrictions on global labels (🔗 Page 77 Restrictions)
- Data that cannot be mapped in the following:
 - 🔗 Page 76 Availability and specification methods

*¹ For details on MELNS, refer to the following:

🔗 Page 51 Namespace of a label Node



When there are multiple Nodes with the same NodeId in an AddressSpace, none of them can be mapped if any of them cannot be mapped.

Mapping methods

Enter a data name in the Mapping column of a Node to be mapped in the NodeTree screen.

Point

When there are multiple Nodes with the same NodeId in an AddressSpace, the mapping information of each Node is changed accordingly if any of them is mapped.

■ For an Array

Map data to an Element displayed in a child hierarchy of a Node. For details, refer to the following:

 Page 113 Array mapping

■ For a Structure

Map data to a Field displayed in a child hierarchy of a Node. When the Field is of Structure, a Field is displayed in its child hierarchy.

When a Node of the Structure is an Array or the Field is of Structure and Array, an Element is displayed in a child hierarchy and a Field is displayed in its child hierarchy.

When the Field of the Structure is other than of Structure and is of Array, a Field is displayed in a child hierarchy and an Element is displayed in its child hierarchy.

■ For an Enumeration

Data is mapped in the same manner as a normal Node.

Mapping operations (copy, cut, paste)

In mapping, items can be copied, cut, and pasted, which allows Excel[®] and the global label setting of GX Works3 to be utilized.

The following table shows the availability of copying, cutting, and pasting each item in the NodeTree screen.

○: Available, ×: Not available

Screen	Item	Copy	Cut	Paste
NodeTree	NodeTree	○	×	×
	DisplayName	○	×	×
	NodeClass	○	×	×
	DataType (Node)	○	×	×
	TypeDefinition	○	×	×
	ValueRank	○	×	×
	ArrayDimensions	○	×	×
	AccessLevel	○	×	×
	Description	○	×	×
	Mapping	○	○	○

Note that the availability differs for each selection method for a target as shown in the following table.

○: Available, ×: Not available, —: Not applicable

Target	Selection method	Copy	Cut	Paste
Character string	Single selection	○	○	○
	Range selection (one column)*1 (Shift + click)	○	○	○
	Range selection (multiple columns)*1*2 (Shift + click)	○	○	○
	Partial selection (one column) (Ctrl + click)	×	×	×
	Partial selection (multiple columns) (Ctrl + click)	×	×	×
Others (file, image, etc.)		—	—	×

*1 For collapsed trees, hidden items in multiple selected rows are also targets.

*2 Hidden columns are also targets.

(Example) Hidden columns between multiple selected columns

(Example) Hidden columns included when pasting an item

Array mapping

If a Node is an Array, ArrayElements are expanded in its child hierarchy. The contents differ depending on the ValueRank and ArrayDimensions.

When setting mapping array specification, a specified Rank and Dim 1 to 3 are regarded as the ValueRank and ArrayDimensions, respectively, and an Element of the Array is expanded in the child hierarchy. ( Page 115 Mapping array specification)

No.	ValueRank ^{*1}	ArrayDimensions ^{*1}	Element of an Array expanded in a child hierarchy ^{*2}	Display in the DataType (Node) column
1	Can be assigned.	Can be assigned.	Follows the ValueRank and ArrayDimensions.	 Page 114 When the ValueRank can be assigned
2		Can be assigned with limitations.	Follows mapping array specification. ^{*3*4*5}	• Node: Blue • Array element: Blue  Page 114 When the ValueRank can be assigned with limitations
3		Cannot be used.	Nothing is displayed in a child hierarchy.	Node: Red  Page 114 When the ValueRank cannot be used
4	Can be assigned with limitations.		Follows mapping array specification. ^{*3*5}	Same as No.2
5	Cannot be used.		Nothing is displayed in a child hierarchy.	Same as No.3

*1 For details on the assignment availability, refer to the following:

 Page 64 Array assignment method

*2 For multiple dimensions, [{number of elements}] for the number of dimensions is displayed in order of [One dimension], [Two dimensions], and [Three dimensions] from the left.

*3 For handling when not setting mapping array specification, refer to the following:

 Page 64 Limited assignment

*4 If setting mapping array specification but not mapped, its setting is discarded at the following timings:

- Saving a project
- Writing to a module
- Exporting a file

*5 Arguments under InputArguments and OutputArguments of a Method cannot be used (No.3).

■ When the ValueRank can be assigned

NodeTree	DataType (Node)
0:Root	
0:Objects	
2:ArrayNode	Boolean[2][3]
[0]	Boolean[3]
[0]	Boolean
[1]	Boolean
[2]	Boolean
[1]	Boolean[3]
[0]	Boolean
[1]	Boolean
[2]	Boolean

■ When the ValueRank can be assigned with limitations

NodeTree	DataType (Node)
0:Root	
0:Objects	
2:ArrayNode	Boolean[1][1]
[0]	Boolean[1]
[0]	Boolean

■ When the ValueRank cannot be used

NodeTree	DataType (Node)
0:Root	
0:Objects	
2:ArrayNode	Boolean[-]

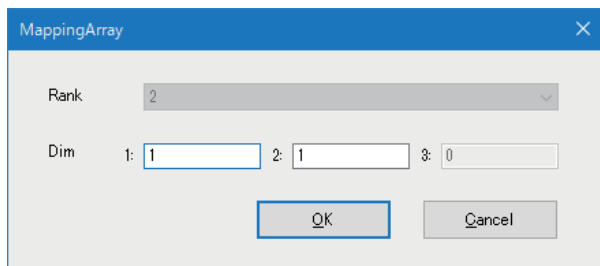
Mapping array specification

Mapping array specification can be set for an Array Node if its ValueRank can be assigned with limitations.

The number of dimensions (Rank) and the numbers of elements (Dim 1 to 3) can be specified for an array element to be mapped.

Window

Right-click the row of the Node (including its child hierarchies) in the NodeTree screen, and select [Mapping Array Specification] from the shortcut menu.



The image shows a dialog box titled "MappingArray" with a close button (X) in the top right corner. Inside the dialog, there is a "Rank" label followed by a dropdown menu showing the value "2". Below this, there is a "Dim" label followed by three input fields labeled "1:", "2:", and "3:". The first field contains "1", the second contains "1", and the third contains "0". At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Displayed items

Item	Description
Rank	Select the number of dimensions. For mapping array specification for a Field, 2 or 3 cannot be set.
Dim 1 to 3	Set the number of elements in each dimension.

- To clear the setting of mapping array specification

Right-click the row of the Node (including its child hierarchies), and select [Mapping Array Specification Clear] from the shortcut menu.

9.6 Global Label Linking

This section shows the details on creating Nodes based on label information of global labels.

Importing global labels

Label information^{*1} used for building label Nodes can be imported from a GX Works3 project.

Imported label information is displayed in the [Label] tab in the bottom window. ( Page 38 Bottom window - [Label] tab)

^{*1} Refers to global labels and structures used for the global labels.

Precautions

- To import global labels, GX Works3 (Version 1.040S or later) must be installed.
- Unconverted GX Works3 projects cannot be imported.
- GX Works3 projects that cannot be opened in installed GX Works3 cannot be imported.
- Do not use a project being operated in GX Works3. (Label information may not match due to the timing of saving the GX Works3 project and importing global labels.)

Import procedure

Operating procedure

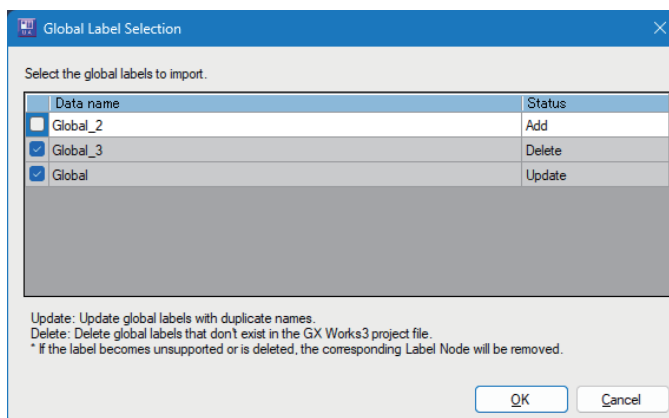
1. Select [File] ⇒ [Import] ⇒ [Global Label].
2. Select an import source GX Works3 project, and click the [Open] button.

Point

- When the security version of the selected project is '2,' the "Project Authentication (GX Works3 Project)" screen appears. Enter the project password set in GX Works3 to perform project authentication.
Note that when using a project with security version '2,' GX Works3 with software version '1.096A' or later must be installed.
- When the selected project is a safety project, the "User Authentication (GX Works3 Project)" screen appears. Enter the user name and password set in GX Works3 to perform user authentication.

3. Select setting data to import, and click the [OK] button.

Note that if there are any global labels to be updated or deleted, the target setting data is automatically selected for import. (The checkbox cannot be unselected.)



4. Check the import result, and click the [OK] button.

Building label Nodes

Objects and mapped Variables can be created from label information imported by importing global labels.

Build label Nodes in the bottom window ([Label] tab). Nodes can be created on NodeTree simply by selecting the checkbox of an item to create a Node.

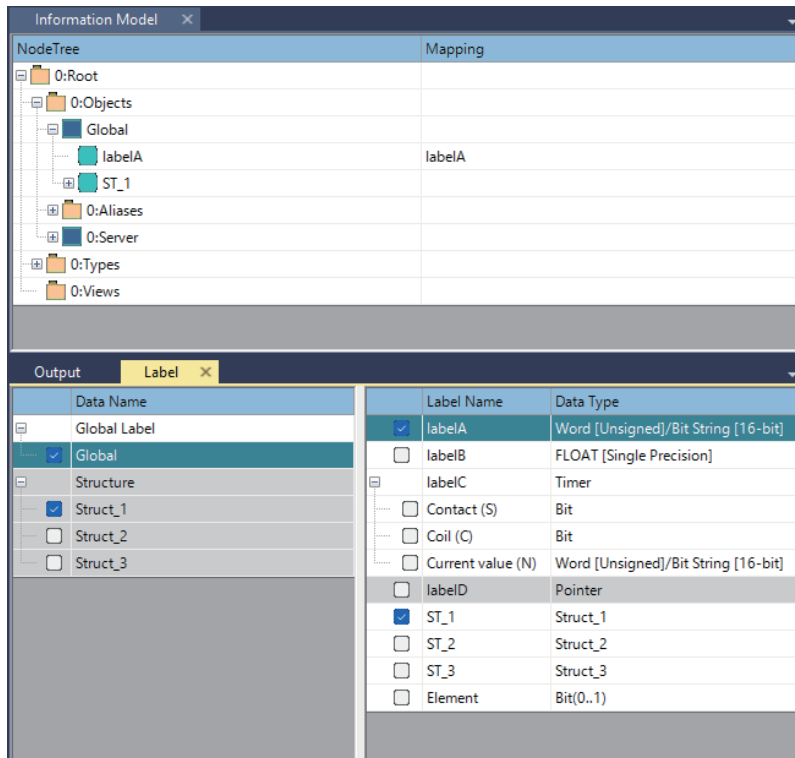
For details on the function, refer to the following:

☞ Page 49 Label Node builder

For an operating procedure example, refer to the following:

☞ Page 26 Setting procedure for an OPC UA server module

Window



For details on the screen, refer to the following:

☞ Page 38 Bottom window - [Label] tab

☞ Page 98 NodeTree

Methods for creating, deleting, and updating label Nodes

The following tables show the methods for creating, deleting, and updating label Nodes.

■ Create

Label Node type	Method	Creation destination
Label Node (Object)	Select the checkbox of global label setting data in the bottom window ([Label] tab).	• Under Root > Objects
Label Node (Variable)	Select the checkbox of a target global label in the bottom window ([Label] tab).	• Under Root > Objects • Under the corresponding label Node (Object) in the creation destination, if any
Label Node (Structure)	Automatically created when creating a label Node (Variable) that uses a target structure.	• Under Root > Types > DataTypes > BaseDataType > Structure
Label Node (Field)	Automatically created when creating a label Node (Variable) that uses a target structure and setting "Variables for Structure Fields" to "Create" in the optional operation setting. (☞ Page 131 Optional Operation Setting) Variables (label Nodes (Fields)) for structure members grayed out in the bottom window ([Label] tab) are not created.	• Under a label Node (Variable) that uses the target structure
Label Node (Element)	Automatically created when creating a label Node related to a target array ^{*1} and setting "Variables for Array Elements" to "Create" in the optional operation setting. (☞ Page 131 Optional Operation Setting)	• Under a label Node related to the target array ^{*1}

*1 The following are applicable:

- Array type label Node (Variable)
- Structure array type label Node (Variable)
- Array type label Node (Field)

Point

By creating Variables for structure members (label Nodes (Fields)) and Variables for array elements (label Nodes (Elements)), each Variable can be used individually by OPC UA clients, reducing the amount of data during communication. However, the number of Variables and the Variable data size increase.

■ Delete

Label Node type	Method
Label Node (Object)	Unselect the checkbox of global label setting data in the bottom window ([Label] tab). Otherwise, select a target Node on NodeTree (multiple selections allowed), and press the  key. A label Node is also deleted when deleting the global label setting data in GX Works3 then importing the global label. When deleting a label Node (Object), the Nodes under the deleted target are also deleted.
Label Node (Variable)	Unselect the checkbox of a target global label in the bottom window ([Label] tab). Otherwise, select a target Node on NodeTree (multiple selections allowed), and press the  key. A label Node is also deleted when deleting the global label in GX Works3 then importing the global label. When deleting a label Node (Variable), the Nodes under the deleted target are also deleted.
Label Node (Structure)	Delete all label Nodes (Variables) that use a target structure. Otherwise, update all label Nodes (Variables) that use the target structure to non-structure types. ^{*1}
Label Node (Field)	Set "Variables for Structure Fields" to "Do not create" in the optional operation setting. (☞ Page 131 Optional Operation Setting) Otherwise, update a label Node (Variable) that uses the target structure to a non-structure type. ^{*1}
Label Node (Element)	Set "Variables for Array Elements" to "Do not create" in the optional operation setting. (☞ Page 131 Optional Operation Setting) Otherwise, update a label Node related to the target array ^{*2} to one not related to the array. ^{*1}

*1 Update them by importing global labels.

*2 The following are applicable:

- Array type label Node (Variable)
- Structure array type label Node (Variable)
- Array type label Node (Field)

■Update

Label Node type	Method
Label Node (Object)	None (If changing the setting data name, the setting data is treated as different data.)
Label Node (Variable)	Change the label information in GX Works3, then import the global label.
Label Node (Structure)	Change the label information in GX Works3, then import the global label.
Label Node (Field)	Change the label information in GX Works3, then import the global label.
Label Node (Element)	Change the label information in GX Works3, then import the global label.

9.7 Namespace

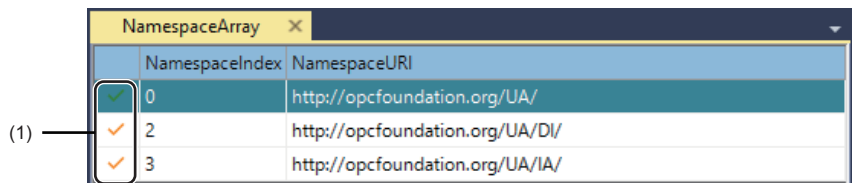
This section shows the screen for displaying information on Namespaces in a project.

NamespaceArray

Information on Namespaces is displayed in the main window.

Window

Select "Namespaces" in the navigation tree.



Displayed items

Item	Description
(1) Import status of a RequiredModel	The status of importing a RequiredModel is displayed. ✓ : One or more RequiredModels with a different version from a specified one exist. ✓ : All specified RequiredModels are properly imported.
NamespaceIndex	The NamespaceIndex is displayed.
NamespaceURI	The NamespaceURI is displayed.
Version	The version of a Namespace is displayed.
PublicationDate	The publication date of a Namespace is displayed.

Point

An item not displayed by default is displayed by right-clicking the column title and selecting the item.

Right-click menus

The following menus are displayed by right-clicking a row.

- Information Model Import: To import an information model file. (Page 107 Importing an information model file)
- Information Model Export: To export an information model file. (Page 109 Exporting an information model file)
- Delete^{*1*2}: To delete a selected Namespace.
- Selection of emphasis color^{*3}: To change the color of a selected Namespace and its corresponding NodeTree to a selected color.
- Information Model: To display the NodeTree screen. (Page 98 NodeTree)

*1 ns=0 (http://opcfoundation.org/UA/) cannot be deleted.

*2 Can be selected when selecting the following Namespaces:

- MELNS
- Namespace in the bottom row excluding MELNS

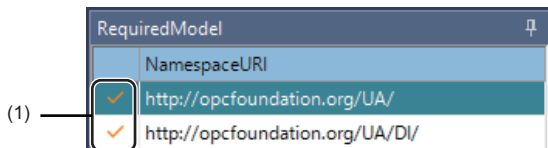
*3 A color already used cannot be selected.

RequiredModel

RequiredModels of a Namespace are displayed in the subwindow.

Window

Select a NamespaceURI under Namespaces in the navigation tree.



Displayed items

Item	Description
(1) Import status	The status of importing each RequiredModel is displayed. ✓ : The version of an imported RequiredModel differs from a specified one. ^{*1} ✓ : A specified RequiredModel is properly imported.
NamespaceURI	The NamespaceURI is displayed.
Version	The version of a Namespace is displayed.
PublicationDate	The publication date of a Namespace is displayed.

^{*1} The version of ns=0 cannot be changed in the configuration tool; however, the Namespace (ns=0) can be used even if the version differs. Note that when adding Types in a new version, instances (Objects and Variables) using the added Types are not supported.

Point

An item not displayed by default is displayed by right-clicking the column title and selecting the item.

9.8 User Setting

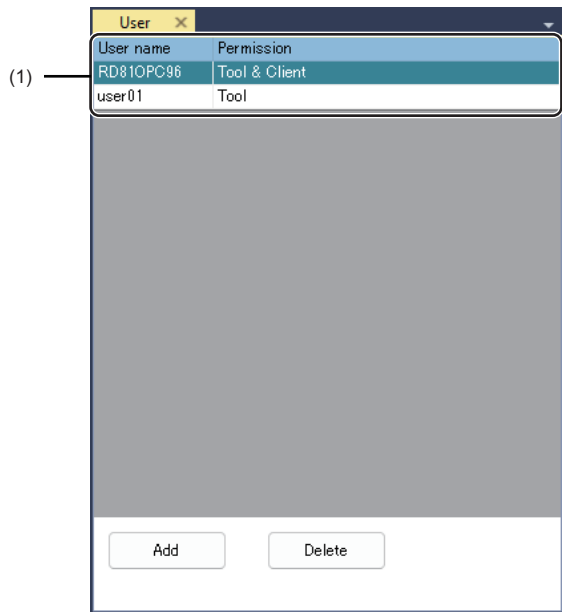
This section shows the screen for setting a user used for user authentication in communication between the configuration tool or an OPC UA client and an OPC UA server module.

A user can be defined as a user of an OPC UA server module if the communication with the following is allowed.

- Configuration tool only
- OPC UA client only
- Both of them

Window

Select "User" in the navigation tree.



Displayed items

Item	Description
(1) User list	Users are listed. By selecting a user, the setting can be edited in the subwindow. (🔗 Page 123 User setting)
[Add] button	Click this to create a new user. (🔗 Page 123 User setting)
[Delete] button	Click this to delete a selected user.

User setting

A user can be set or edited in the subwindow.

Window

User Setting

User name

user01

Permission

Tool & Client

Apply

Displayed items

Item	Description
User name	Set or edit a user name. For available characters, refer to the following:  Page 147 Available Characters
Permission	Set or edit user permission (communication target).
[Apply] button	Click this to enter a password and apply the setting. ( Page 124 "Password input" screen)

Point

The default user setting is as follows:

- User name: RD81OPC96
- Password: MITSUBISHI

■ "Password input" screen

Password input

Password

☐ Change the password

Changing

Confirmation

OK

Cancel

Displayed items

Item	Description
Password	■For creating a new user Set a password (6 to 32 characters including upper-case and lower-case alphabetical characters, and numbers). For available characters, refer to the following:  Page 147 Available Characters ■For editing a user Enter the set password.
Change the password	Select the checkbox to change the set password.
Changing	When selecting the checkbox of "Change the password," enter a new password.
Confirmation	Enter a new password again.

9.9 Network Setting

This section shows the screen for setting the IP address of an Ethernet port and a port number used for OPC UA communication.

There are two Ethernet ports: CH1 and CH2, and only CH1 can be used for OPC UA communication and the both can be used for communication with the configuration tool.

In the subwindow, an endpoint URL used for OPC UA communication is displayed.

Window

Select "Network" in the navigation tree.

Displayed items

■Main window

Item	Description
Enable Ethernet port (CH1)	— Using the Ethernet port (CH1) is enabled (fixed).
IP Address	Set the IP address of CH1.* ¹
Subnet Mask	Set the subnet mask of CH1.
Port No.	Set a port number for OPC UA communication.
Enable Ethernet port (CH2)	— Select the checkbox to use the Ethernet port (CH2). OPC UA communication and other communication can be used in different networks.
IP Address	Set the IP address of CH2.* ¹
Subnet Mask	Set the subnet mask of CH2.
Default Gateway	To set a default gateway, select a port to set and set the address.* ²
Host Name	Set a host name (common to CH1/CH2).
[Apply] button	Click this to apply the setting.

*¹ A same IP address or an IP address in the same network cannot be set for both CH1 and CH2.

*² Only one of CH1 or CH2 can be set.

Communication with the same network as each CH is performed from the corresponding CH only. (Even if a default gateway is set for the other CH, communication is not performed from the other one.)

■Subwindow

Item	Description
End Point URL	An endpoint URL used when connecting from an OPC UA client is displayed. When changing a port number for "Port No." in the main window, the endpoint URL is also changed.

9.10 Security Setting

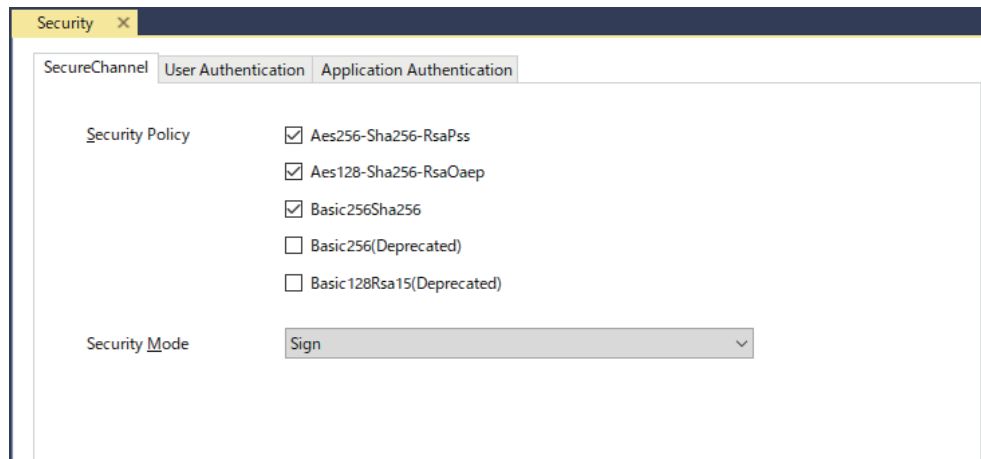
This section shows the screen for configuring the setting for two-step authentication (application authentication and user authentication) to secure OPC UA communication and for SecureChannel construction.
User authentication can also be set for communication with the configuration tool.

Window

Select "Security" in the navigation tree.

■[SecureChannel] tab

A Security Mode to prevent eavesdropping and falsification of messages in OPC UA communication and a SecurityPolicy to be its algorithm can be specified.



Displayed items

Item	Description
Security Policy	Select the checkbox of a security policy to enable. When not selecting any checkboxes, all the security policies including None are enabled. Aes256-Sha256-RsaPss: AES 256-bit encryption + SHA-256 (default: selected) Aes128-Sha256-RsaOaep: AES 128-bit encryption + SHA-256 (default: selected) Basic256Sha256: Basic 256-bit encryption + SHA-256 (default: selected) Basic256(Deprecated)*1: Basic 256-bit encryption (default: unselected) Basic128Rsa15(Deprecated)*1: Basic 128-bit encryption (default: unselected)
Security Mode	Select a security mode.*2*3 None*4: No security Sign*5: An electronic signature prevents falsification. Sign & Encrypt*5: Encryption prevents eavesdropping.

- *1 Not recommended in the OPC UA specifications, because it is considered less secure.
- *2 All security modes equal to or stronger than the selected one are enabled.
- *3 The selection is automatically changed in the following cases:
When unselecting the checkboxes of all security policies: None
When selecting any of the checkboxes of security policies while none of them is selected: Sign
- *4 Cannot be selected if any of the checkboxes of security policies is selected.
- *5 Cannot be selected if none of the checkboxes of security policies is selected.

■[User Authentication] tab

Authentication with a user name and password can be used for user authentication in communication with the configuration tool.

For user authentication (User Authentication) in OPC UA communication, the following methods can be used:

- Authentication with a certificate: A method for authentication with a user certificate (X.509). A user certificate must be uploaded to the trust list in advance by selecting [Online] ⇒ [Manage User Certificate].
- Authentication with a user name and password: A method for user authentication with a combination of a user name and password
- Authentication by Anonymous: A method for allowing anonymous users to log on

Displayed items

Item	Description
User Authentication for Connection from MX OPC UA Module Configurator-R	Select a user authentication necessity.
User Authentication for Connections from OPC UA Client	Select the checkbox of a user authentication method. For allowing authentication with a certificate, a user certificate must be uploaded to the trust list by clicking the [User Certificate Management] button.

Point

A user name and password used for user authentication can be set in the user setting. (🔍 Page 122 User Setting)

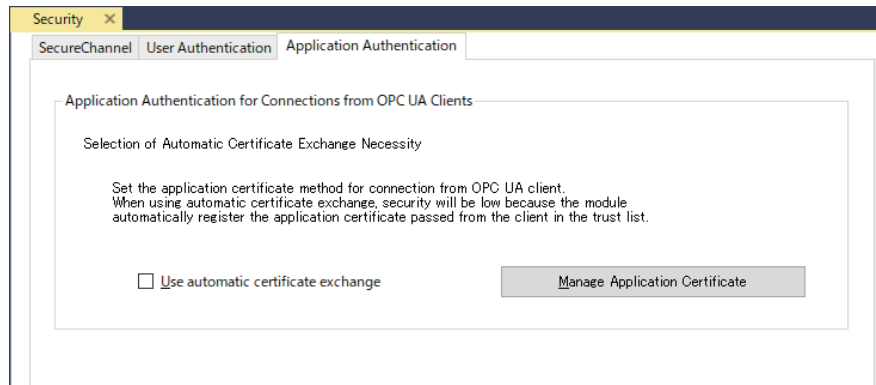
Precautions

Allowing Anonymous logon is not recommended in an open network environment such as the Internet or an intranet, because it allows connections from all users.

■[Application Authentication] tab

Application authentication (Application Authentication) is a method for mutual authentication by exchanging electronic certificates between an OPC UA server and OPC UA client.

An OPC UA server module has a function to automatically exchange an application certificate received from an OPC UA client for test operation. This allows application authentication to be passed for any OPC UA client, resulting in less security. It is recommended to disable this function during actual operation.



Displayed items

Item	Description
Application Authentication for Connections from OPC UA Clients	Select the checkbox to exchange certificates automatically. When not automatically exchanging certificates, an application certificate must be uploaded to the trust list by clicking the [Manage Application Certificate] button.

Precautions

- Automatically exchanging certificates is not recommended in an open network environment such as the Internet or an intranet, because it allows connections from all OPC UA clients.
- Automatically exchanged certificates are not displayed in the "Manage Application Certificate" screen.

9.11 Server Certificate Setting

This section shows the screen for setting the contents of a server certificate of an OPC UA server module used for application authentication.

Window

Select "Server Certificates" in the navigation tree.

Displayed items

Item	Description
OPC UA Server Name	Enter the name of an OPC UA server.
Organization name	Enter the name of an organization.
Unit	Enter the unit of an organization.
Locality	Enter the name of a region.
State	Enter a state.
Country	Enter the abbreviation for a country name.
[Apply] button	Click this to apply the setting.

Point

An OPC UA server module automatically generates a server certificate (its own application certificate) according to this setting at the first startup. For details on the expiration date, refer to the following:

MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

The generated certificate can be downloaded and used for application authentication as it is. In addition, a certificate prepared by a user can be uploaded as a server certificate and used for application authentication.

A server certificate can be downloaded and uploaded in the "Manage Application Certificate" screen. For details on the screen, refer to the following:

MELSEC iQ-R OPC UA Server Module User's Manual (Application)

9.12 Discovery Server Setting

This section shows the screen for specifying a discovery server to register an OPC UA server module.
An OPC UA client can acquire the endpoint URL of an OPC UA server module registered in a discovery server by using the Discovery Service.

Window

Select "Discovery Server" in the navigation tree.

Discovery Server

Discovery Server URL

Registration interval

10000

ms

Apply

Displayed items

Item	Description
Discovery Server URL	When using a discovery server, specify its URL. An OPC UA client can acquire the endpoint URL of an OPC UA server module from a discovery server. When not using a discovery server, leave the discovery server URL blank. The endpoint URL of an OPC UA server module can be checked in the network setting. (Page 125 Network Setting)
Registration interval	Set an interval for registering the endpoint URL of an OPC UA server module in a discovery server.
[Apply] button	Click this to apply the setting.

9.13 Optional Operation Setting

This section shows the screen for configuring the setting related to the following functions:

- Data input/output function
- Label Node builder function

Window

Select "Optional Operation Setting" in the navigation tree.

Item	Setting
[-] Data Sampling Setting	
Sampling Mode	Standard sampling
[-] Label Node Setting	
Variables for Array Elements	Do not create
Variables for Structure Fields	Do not create

Description
Set to change the behavior of data sampling.

Apply

Displayed items

Item	Description	Target function
Data Sampling Setting	Set the sampling mode for collecting data from a CPU module. For details on the sampling mode, refer to the following: ☞ Page 132 Data sampling mode	Data input/output function ☞ Page 80 Data reading
Label Node Setting	Set whether to create specific label Nodes.	Label Node builder function ☞ Page 49 Label Node builder
[Apply] button	Click this to apply the setting.	—

Data sampling mode

The following sampling modes can be selected.

Sampling mode	Description	Restriction	Influence on the scan time
Standard sampling	Data is collected by using the device/label access service processing of a CPU module.	N/A	Not affected
High-speed sampling (Only MonitoredItems)	Data is collected by using both the sequence scan synchronization sampling function and the device/label access service processing of a CPU module. - For the Subscription, data can be collected faster than standard sampling. (For the Read, it is the same as standard sampling.) - Only MonitoredItems specified by an OPC UA client are collected at high speed.	The total data size of MonitoredItems (total size of data mapped to each MonitoredItem) must be 32K words or less.	Affected (The influence level varies depending on the total size of MonitoredItems specified by an OPC UA client.)
High-speed sampling (All Variables)	Data is collected by using the sequence scan synchronization sampling function of a CPU module. - Data can be collected faster than standard sampling for both the Read and Subscription. - All mapped Variables are collected at high speed.	The total size of data that can be mapped to a Variable must be 32K words or less.	Affected

Point

In high-speed sampling, data is collected from the control CPU at high speed by using the sequence scan synchronization sampling function of a CPU module. For details on the function, refer to the manual for a CPU module used.

9.14 Module Control

This section shows the details on connecting the configuration tool to an OPC UA server module, reading, writing, or verifying a project file, and restarting or stopping an OPC UA server module or updating its setting.

Connection destination setting

For details, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

Online data operations

For details, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

Server operation

For details, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

9.15 Module Diagnostics

This section shows the details on displaying the operating status of an OPC UA server module and on management operations.

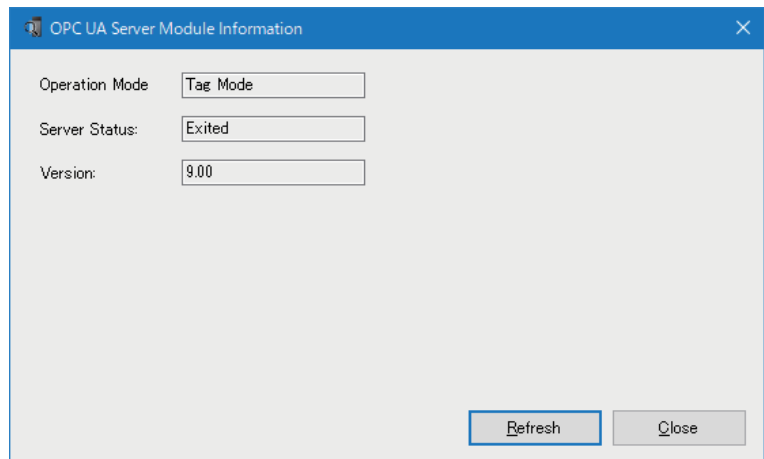
In addition, the status of an SD memory card can be displayed and the card can be formatted.

Information on an OPC UA server module

The following shows the screen for displaying the operating status of an OPC UA server module.

Window

Select [Online] ⇒ [OPC UA Server Module Information].



Displayed items

Item	Description
Operation Mode	The operation mode of an OPC UA server module is displayed.
Server Status	The status of an OPC UA server module is displayed.
Version	The software version of an OPC UA server module is displayed.
[Refresh] button	Click this to update the operating status of an OPC UA server module to the latest information.

OPC UA server module management

For details, refer to the following:

📖MELSEC iQ-R OPC UA Server Module User's Manual (Application)

SD memory card diagnostics

For details, refer to the following:

📖MELSEC iQ-R OPC UA Server Module User's Manual (Application)

9.16 Electronic Certificate Management

This section describes the details on managing user certificates and application certificates used for OPC UA communication.

User certificate management

User certificates retained in an OPC UA server module can be managed.

When using 'authentication with a certificate' as the user authentication method for OPC UA communication, upload a user certificate used for connection from an OPC UA client to the trust list of an OPC UA server module. ( Page 127 [User Authentication] tab)

For details on the "Manage User Certificate" screen, the procedures for uploading certificates to the trust list, and handling of certificates certified by a certificate authority, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

Application certificate management

Application certificates and server certificates retained in an OPC UA server module can be managed.

Upload an application certificate of an OPC UA client in application authentication for OPC UA communication to the trust list of an OPC UA server module. In addition, a server certificate can be downloaded and uploaded. Configure the setting for a server certificate in the setting screen of the server certificate. ( Page 129 Server Certificate Setting)

For details on the "Manage Application Certificate" screen, the procedures for uploading certificates to the trust list, and handling of certificates certified by a certificate authority, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

10 TROUBLESHOOTING

This chapter explains the errors which may occur when using an OPC UA server module and the troubleshooting.

10.1 Checking Method for Error Descriptions

For details, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

10.2 Checking the Module Status

For details, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

10.3 Troubleshooting by Symptom

This section shows the troubleshooting on the information model mode.

For that on the tag mode and common to each mode, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

10

Troubleshooting on the configuration tool

Function	Symptom	Check point	Corrective action
NodeTree (Importing an information model file, mapping)	A target Node does not exist.	Do all required Namespaces exist in the NamespaceArray screen?	Import an information model file containing the required Namespaces.
		Does the target Node exist in the information model file?	Review the information model file in the modeling tool, and import it again.
	An Attribute is incorrect.	Is the Attribute in the information model file correct?	
	A Reference is incorrect.	Is the Reference in the information model file correct?	
	'It failed a information model processing.' is displayed.	Are the contents of the information model file correct?	End the configuration tool, then perform the following: • Review the information model file then start the configuration tool, and import it again. • Update the configuration tool to the latest version then start it, and import the file again.
Simultaneous use of configuration tools	Configuration tools cannot be started up to the upper limit of five.	Has the running configuration tool ended due to a message such as 'It failed a information model processing.'?	End all running configuration tools, then restart them.

Troubleshooting on communication with an OPC UA client

Function	Symptom	Check point	Corrective action
AddressSpace	A target Node does not exist.	Refer to 'NodeTree' in the troubleshooting on the configuration tool.	
	An Attribute is incorrect.		
	A Reference is incorrect.		
Attribute Read	A Value cannot be read.	Is data mapped to the Node in the NodeTree screen?	Review the NodeTree screen of the configuration tool, then write and update the setting again.
		Is the CurrentRead (Bit 0) readable (1) for the AccessLevel of the Node?	Review the information model file in the modeling tool and import it in the configuration tool, then write and update the setting again.
Attribute Write	A Value cannot be written.	Is data mapped to the Node in the NodeTree screen?	Review the NodeTree screen of the configuration tool, then write and update the setting again.
		Is the CurrentWrite (Bit 1) writable (1) for the AccessLevel of the Node?	Review the information model file in the modeling tool and import it in the configuration tool, then write and update the setting again.
MonitoredItem Subscription	A Value cannot be read.	Is data mapped to the Node in the NodeTree screen?	Review the NodeTree screen of the configuration tool, then write and update the setting again.
		Is the CurrentRead (Bit 0) readable (1) for the AccessLevel of the Node?	Review the information model file in the modeling tool and import it in the configuration tool, then write and update the setting again.
	A Value does not change. (It takes time to change.)	Is data mapped to the Node in the NodeTree screen?	Review the NodeTree screen of the configuration tool, then write and update the setting again.
		Has the mapping data value actually changed?	Monitor the mapping data in GX Works3 to check the value change and timing.
		Is an appropriate value specified for the SamplingInterval in an OPC UA client? (Can changes in mapping data be detected with the value?)	- Review the SamplingInterval specified in the OPC UA client. - If the difference between the RevisedSamplingInterval and a measured value is not acceptable, reduce the load on the module by the following methods: - Increase the interval for the SamplingInterval. - Reduce the number of MonitoredItems. - Reduce the number of Subscriptions. - Reduce the number of units of mapping data. - Reduce the frequency of requests from the OPC UA client.
		Is the operation being performed with a specified SamplingInterval? (Is there any difference between 'RevisedSamplingInterval (minimum) [ms]' (Un\G12419 to 12420) and 'Actual SamplingInterval (minimum) [ms]' (Un\G12421 to 12422) in the buffer memory?)	
		Is an appropriate value specified for the PublishingInterval in an OPC UA client?	Review the PublishingInterval specified in the OPC UA client.
—	The StatusCode for Bad or Bad_NoCommunication is returned.	Has an error been output to a CPU module or OPC UA server module?	- If an error is output, check the contents described in the corrective action for the error code to take corrective action. - If no error is output, check if the requests from an OPC UA client exceed the upper limit of the performance specifications. (Page 18 Performance Specifications)
	The StatusCode for Good_Overload is returned.	Has a global label of a CPU module been added or changed?	When adding or changing a global label (including one not used for an OPC UA server module), data cannot be temporarily acquired. Wait for a while before resuming the operation.

10.4 Error Code List

For the error code list, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

10.5 Event Code List

For the event code list, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

APPENDIX

Appendix 1 External Dimensions

For details, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Startup)

Appendix 2 Module Labels

For details, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

A

Appendix 3 Input/Output Signals

For details, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

Appendix 4 Buffer Memory

For details, refer to the following.

Note that only the control CPU is an access target device in the information model mode; therefore, only 'Target device 1' of 'Target device information' (Un\G8448 to 8703) is used.

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

Appendix 5 Data Assignment Availability for Standard DataTypes

This section shows the data assignment availability for Standard DataTypes.

Standard DataTypes			Label assignment availability
BaseDataType	—		Not available
	Boolean		Refer to the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes
	ByteString	—	Not available
		Image	ImageBMP
			ImageGIF
			ImageJPG
			ImagePNG
		AudioDataType	
	DateTime	—	
		UtcTime	
	DataValue		Refer to the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes
	DiagnosticInfo		Refer to the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes
	Enumeration	—	Not available
		IdType	Refer to the following: ☞ Page 67 Data types that can be assigned to Enumerations
		NamingRuleType	
		NodeClass	
		SecurityTokenRequestType	
		MessageSecurityMode	
		RedundancySupport	
		ServerState	
	ExpandedNodeId		Refer to the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes
	Guid		
	LocalizedText		
	NodeId		
	Number	—	Not available
		Integer	—
			SByte
			Int16
			Int32
			Int64

Standard DataTypes					Label assignment availability		
BaseDataType	Number	UInteger	—		Not available		
			Byte	—	Refer to the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes		
				AccessLevelType	Same as the Byte in the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes		
				EventNotifierType			
				UInt16	—	Refer to the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes	
			AccessRestrictionsType		Same as the UInt16 in the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes		
			UInt32		—	Refer to the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes	
					AccessLevelExType	Same as the UInt32 in the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes	
				AttributeWriteMask			
				PermissionType			
			UInt64	—	Refer to the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes		
				BitFieldMaskDataType	Same as the UInt64 in the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes		
			Double	—		Refer to the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes	
				Duration		Same as the Double in the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes	
			Float		Refer to the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes		
			Decimal		Refer to the following: ☞ Page 63 Data types that can be assigned to other DataTypes		
		QualifiedName		Refer to the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes			
		String	—		☞ Page 60 Data types that can be assigned to Built-in DataTypes		
			LocaleId		Same as the String in the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes		
			NumericRange				
	NormalizedString						
	DecimalString						
	DurationString						
	TimeString						
	DateString						

Standard DataTypes		Label assignment availability
BaseDataType	Structure	—
		Argument
		Refer to the following: ☞ Page 65 Data types that can be assigned to Fields of a Structure
		UserIdentityToken
		—
		Not available
		UserNameIdentityToken
		Refer to the following: ☞ Page 65 Data types that can be assigned to Fields of a Structure
		X509IdentityToken
		AnonymousIdentityToken
		AddNodesItem
		AddReferencesItem
		DeleteNodesItem
		DeleteReferencesItem
		ApplicationDescription
		BuildInfo
		SamplingIntervalDiagnosticsDataType
		ServerDiagnosticsSummaryDataType
		ServerStatusDataType
		SessionDiagnosticsDataType
		SessionSecurityDiagnosticsDataType
		ServiceCounterDataType
		StatusResult
		SubscriptionDiagnosticsDataType
		ModelChangeStructureDataType
		SemanticChangeStructureDataType
		SignedSoftwareCertificate
		TimeZoneDataType
		EnumValueType
		—
		EnumField
		OptionSet
		Not available
		Union
		Not available
		StructureField
		Refer to the following: ☞ Page 65 Data types that can be assigned to Fields of a Structure
		DataTypeDefinition
		—
		Not available
		StructureDefinition
		Refer to the following: ☞ Page 65 Data types that can be assigned to Fields of a Structure
		EnumDefinition
		RedundantServerDataType
	XmlElement	Refer to the following: ☞ Page 60 Data types that can be assigned to Built-in DataTypes

Appendix 6 Available Characters

This section shows the characters that can be used for each setting item.

Screen/operation	Item	Available character	Reference
NodeTree	Mapping column	Characters that can be represented in Unicode ^{*1}	Page 98 NodeTree
Importing an information model file	Folder path	Depends on the specifications of an operating system.	Page 107 Importing an information model file
	File name	Depends on the specifications of an operating system. ^{*2}	
Exporting an information model file	File name	Depends on the specifications of an operating system.	Page 109 Exporting an information model file
	Output destination	Depends on the specifications of an operating system.	
User Setting	User name	☞ Page 148 Available ASCII characters	Page 123 User setting
User Setting (password setting)	Password		Page 124 "Password input" screen
	Changing		
	Confirmation		
Network setting	IP Address	Numerals (decimal)	Page 125 Network Setting
	Subnet Mask		
	Port No.		
	Default Gateway		
	Host Name	☞ Page 148 Available ASCII characters ^{*3}	
Discovery server setting	Discovery Server URL	☞ Page 148 Available ASCII characters^{*4} • ;, /, . (period)	Page 130 Discovery Server Setting
	Registration interval	Numerals (decimal)	
Server certificate setting	OPC UA Server Name	☞ Page 148 Available ASCII characters	Page 129 Server Certificate Setting
	Organization name		
	Unit		
	Locality		
	State		
	Country	☞ Page 148 Available ASCII characters ^{*3}	
Open	Folder path	Depends on the specifications of an operating system.	Page 96 Opening a project
	File name	Depends on the specifications of an operating system. ^{*2}	
Save As	Folder path	Depends on the specifications of an operating system.	Page 97 Saving a project
	File name	Depends on the specifications of an operating system. ^{*2}	
Target Setting	IP Address	Numerals (decimal)	☞ MELSEC iQ-R OPC UA Server Module User's Manual (Application)
	User Name	☞ Page 148 Available ASCII characters	
	Password	☞ Page 148 Available ASCII characters	
OPC UA Server Module Management	Prohibition release password	U+0020 to U+007E (ASCII characters excluding control characters and DEL)	

*1 The following cannot be used:

- Symbols: Single-byte space, !, ", #, \$, %, &, ', (,), *, +, /, - (hyphen), ., :, ;, <, =, >, ?, @, \, ^, ` , {, |, }, ~, environment dependent character, Unicode double-byte space
- Surrogate pair characters: 0xD800 to 0xDBFF, 0xDC00 to 0xDFFF
- Others: U+0000 to U+001F, U+0080 to U+009F, U+00A0 to U+00BF, U+FFFE, U+FFFF

*2 Prohibited characters and prohibited character strings (reserved words) are as follows:

- Prohibited characters: <, >, :, ", /, \, |, ?, *, NUL
- Prohibited character strings (reserved words): CON, PRN, AUX, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, COM¹, COM², COM³, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, LPT¹, LPT², LPT³

*3 Excluding lower-case characters.

*4 Excluding underscores (_).

Available ASCII characters

Characters in the shaded area can be used.

	0	1	2	3	4	5	6	7
0	NUL		(SP)	0	@	P	'	p
1			!	1	A	Q	a	q
2			"	2	B	R	b	r
3			#	3	C	S	c	s
4			\$	4	D	T	d	t
5			%	5	E	U	e	u
6			&	6	F	V	f	v
7			'	7	G	W	g	w
8			(	8	H	X	h	x
9			)	9	I	Y	i	y
A			*	:	J	Z	j	z
B			+	;	K	[	k	{
C			,	<	L	\	l	
D			-	=	M	]	m	}
E			.	>	N	^	n	~
F			/	?	O	_	o	

Appendix 7 Processing Time

Processing time

This section shows the measurement results for the processing times required for the Attribute Read, Attribute Write, and Subscription.

Note that the processing time may increase depending on any of the following factors:

- Usage environment (personal computer, network, and SD memory card)
- Sequence scan time
- Access status from a personal computer, HMI, or another intelligent function module to a CPU module
- Settings of an OPC UA server module

Use the measurement results as a reference for processing time.

The measurement result for the Subscription can be used as a reference for the number of MonitoredItems and the SamplingInterval requested from an OPC UA client.

Measurement conditions

- In measuring the processing times of the Attribute Read and Attribute Write, the time is measured from when an OPC UA client sends a Read Request or Write Request to an OPC UA server module to when a response is returned.
- For a Subscription, the interval at which an OPC UA client receives the Publish of a Node registered in a MonitoredItem is measured.

Item		Description
Access target device	CPU module	R08CPU
	Scan time	20 ms (A program with a specified scan time not a constant scan is used.)
	Device/label access service processing setting	10% (default value)
	Program content	A program for incrementing 50% of mapped data every scan
	Access target	Own station (control CPU)
OPC UA server module	Request	• Attribute Read • Attribute Write • Subscription
	Number of Variables requested	■Attribute Read 100, 1000, 5000, 10000, 20000, 25000 ■Subscription For standard sampling: 100, 500, 1000, 2000, 3000 For others: 100, 1000, 5000, 10000, 15000 ■Attribute Write 100, 500, 1000, 2000, 3000
	Number of units of mapped data	25000
	Model definition	100 Variables are defined for each of 250 Objects.
	DataType of a Variable	Int16
	Data type of mapped data	Word [signed]
	Security policy	Basic256Sha256
	Security mode	Sign & Encrypt
	Data sampling setting	• Standard sampling • High-speed sampling (MonitoredItems only) • High-speed sampling (all Variables)
OPC UA client	CPU	Intel® Core™ i7-8700K 3.70 GHz
	Memory	16 GB
	OS	Windows® 11 Enterprise
	SDK used	UA-.NETStandard-1.4.368.58 of the OPC Foundation
	SamplingInterval (for a Subscription)	• 50 ms • 200 ms
Others	Number of measurements	An average calculated by measuring for 100 times

Measurement results

The following shows the measurement results for the processing time.

Point

When the number of Variables requested (number of units of mapped data and size) is large, increasing the SamplingInterval specified by a client may reduce the processing time of a Subscription.

For verification, specify an actual update cycle for each unit of data (cycle in which values change in a program) and an appropriate SamplingInterval based on an expected monitoring interval.

■Standard sampling

• Attribute Read

Request	Number of Variables requested					
	100	1000	5000	10000	20000	25000
Attribute Read [ms]	47.3	428.3	2124.1	4239.0	8444.3	10551.6

• Subscription

Request	Sampling Interval	Number of Variables requested				
		100	500	1000	2000	3000
Subscription [ms]	50 ms	50.0	120.5	226.6	461.1	701.1
	200 ms	200.0	208.0	222.0	428.1	671.6

• Attribute Write

Request	Number of Variables requested				
	100	500	1000	2000	3000
Attribute Write [ms]	55.6	290.6	569.1	1170.7	1636.8

■High-speed sampling (MonitoredItems only)

• Attribute Read

Request	Number of Variables requested					
	100	1000	5000	10000	20000	25000
Attribute Read [ms]	46.2	427.9	2126.2	4241.5	8443.8	10525.9

• Subscription

Request	Sampling Interval	Number of Variables requested				
		100	1000	5000	10000	15000
Subscription [ms]	50 ms	50.0	64.0	260.0	546.6	850.7
	200 ms	200.0	200.1	242.1	490.3	742.3

• Attribute Write

Request	Number of Variables requested				
	100	500	1000	2000	3000
Attribute Write [ms]	55.5	269.3	377.1	1143.1	1695.0

■ High-speed sampling (all Variables)

- Attribute Read

Request	Number of Variables requested					
	100	1000	5000	10000	20000	25000
Attribute Read [ms]	14.5	125.8	628.9	1205.6	2306.5	2885.9

- Subscription

Request	Sampling Interval	Number of Variables requested				
		100	1000	5000	10000	15000
Subscription [ms]	50 ms	826.6	852.1	927.6	1040.1	1179.6
	200 ms	827.6	843.5	904.1	992.1	1074.1

- Attribute Write

Request	Number of Variables requested				
	100	500	1000	2000	3000
Attribute Write [ms]	892.3	1102.9	1369.8	1824.3	2461.4

Startup time

The following shows the measurement results for the startup time of an OPC UA server module according to the information model definition and whether there is mapping.

Note that the startup time may increase depending on any of the following factors:

- Usage environment (SD memory card)
- Type of a CPU module
- Types of other modules mounted on the base unit/number of modules

If a Node with many References is included in an information model^{*1}, the startup time will increase. Less than 100 References are recommended per Node.

The startup time varies widely depending on the information model definition. Verify it sufficiently before use.

Use the measurement results as a reference for startup time.

^{*1} For example, there are 100 or more Variable Nodes immediately under one Object.

Measurement conditions

The time is measured from when the power is ON (SM400 is ON) to when an OPC UA server module starts operating (X1 is ON).

Item		Description
Configuration	CPU module	R08CPU
	OPC UA server module	RD81OPC96
	SD memory card (inserted in an OPC UA server module)	2 GB
Information model		Information models from the following points: • Variation in startup time depending on the number of Variables • Variation in startup time depending on the number of Objects • Variation in startup time depending on the number of XML files
Data sampling setting		• Standard sampling • High-speed sampling (MonitoredItems only) • High-speed sampling (all Variables)
Others	Number of measurements	An average calculated by measuring for 3 times

Measurement results

The following shows the measurement results for the startup time.

■ Variation in startup time depending on the number of Variables

Number of XML files	Number of Objects	Number of Variables	Mapping	Startup time [second]		
				Standard sampling	High-speed sampling (MonitoredItems only)	High-speed sampling (all Variables)
1	1	100	Mapped (all Variables)	18.3	18.2	18.2
		1000		21.4	21.1	21.8
		5000		43.9	43.5	47.0
		10000		122.5	98.6	104.9
		20000		316.5	316.3	328.1
		25000		480.4	477.5	499.4
1	1	100	Not mapped	18.2	18.2	18.2
		1000		20.1	19.7	19.8
		5000		38.9	30.4	30.5
		10000		87.2	54.1	54.2
		20000		296.5	145.1	145.3
		25000		453.8	212.1	213.2

■ Variation in startup time depending on the number of Objects

Number of XML files	Number of Objects	Number of Variables (per Object)	Mapping	Startup time [second]		
				Standard sampling	High-speed sampling (MonitoredItems only)	High-speed sampling (all Variables)
1	1	5000	Mapped (all Variables)	43.9	43.5	47.0
	10	500		32.7	32.4	35.5
	100	50		32.0	31.8	35.2
	1000	5		33.2	32.8	35.4
1	1	5000	Not mapped	38.9	30.4	30.5
	10	500		27.5	25.9	26.1
	100	50		27.2	25.6	26.0
	1000	5		28.3	27.1	26.9

■ Variation in startup time depending on the number of XML files

Number of XML files	Number of Objects (per XML file)	Number of Variables (per Object)	Mapping	Startup time [second]		
				Standard sampling	High-speed sampling (MonitoredItems only)	High-speed sampling (all Variables)
1	100	50	Mapped (all Variables)	32.0	31.8	35.2
5	20			32.9	32.7	36.1
10	10			33.4	33.2	36.5
1	100	50	Not mapped	27.2	25.6	26.0
5	20			27.9	28.5	28.6
10	10			28.3	28.8	29.1

Appendix 8 XML File Format

This section shows the format specifications of an XML file used for the import or export function for information model files.

XML file specifications

Use an XML file compliant with the OPC UA specifications. *1
*1 Only English, numbers, and symbols are supported for description of the NamespaceURI.

Description rules for mapping information

Describe mapping information of a Node and data in the Extensions of an XML file.
Do not describe data conversion information (Enumeration value conversion, etc.) in an XML file, but retain it in a project file.

Description rules for the xmlns

Use the following xmlns to describe mapping information in an XML file.
• <https://www.mitsubishielectric.com/fa/OPCUA/extension/>

■Description method

Describe 'xmlns' directly in the Extension. (A method for description in the xmlns attribute of the UANodeSet tag is not supported.)
• Description in the Extension
<DataMapping Data="label01" xmlns="https://www.mitsubishielectric.com/fa/OPCUA/extension/" />

Description rules for the Extension

Define mapping information in the Extension.
It must be defined in a combination of an element and attribute.
• <{Element} {Attribute}="{Attribute value}"/>

Item	Description
Element	Case-sensitive
Attribute	Case-sensitive. A space is required immediately before an attribute.
Attribute value	Not case-sensitive
Tag	Line feeds cannot be inserted in a tag.

Ex.
For an element: DataMapping, attribute: Data, attribute value: label01
• <DataMapping Data="label01"/>

■Element

Item		Description	Format
DataMapping	—	Describe mapping information of a Node and data. Describe this element under Extensions → Extension of the UaVariable tag.	📖 Page 157 DataMapping
	Element	When a Node (Variable) to be mapped is an Array, describe mapping information of each element and data.	📖 Page 157 Element
	Field	When a Node (Variable) to be mapped is a Structure, describe mapping information of each Field and data.	📖 Page 159 Field
MethodMapping	—	Describe mapping information of a Method. Describe this element under Extensions → Extension of the UaMethod tag.	📖 Page 160 MethodMapping
	Handle	Describe mapping information of data to notify a program of the start of a Method.	
	Input	Describe mapping information of InputArguments of a Method and data.	
	Argument	Describe mapping information of each element of InputArguments and data as a child element of an Input.	
	Element	When an Argument is an Array, describe mapping information of each element and data. The structure is the same as Element of DataMapping.	
	Field	When an Argument is a Structure, describe mapping information of each Field and data. The structure is the same as Field of DataMapping.	

■Attribute

Element: DataMapping

Attribute ^{*1*2}	Attribute value ^{*3}	Description	Range	Element type (◎: Required, ○: Optional, —: Not supported ^{*4})		
				DataMapping	Element	Field
IsArray	true	When a Node is an Array, describe "true." (Required when a Node is an Array)	true	○	—	○
IsStructure	true	When a Node is a Structure, describe "true." (Required when a Node is a Structure)	true	○	—	○
Num	Element number	Describe the element number of an Array. (Example) "0"	0 to 10239	—	◎	—
Name	Name	Describe the Name of a Field. (Example) "ProductName"	—	—	—	◎
Data ^{*5}	Data name	Describe a global label. (Example) "label01"	1 to 800 characters	○	○	○
Rank	Number of array dimensions	When the ValueRank is any of the following, specify the number of dimensions of an array to be mapped. • -3 (ScalarOrOneDimension) • -2 (Any) • 0 (OneOrMoreDimensions)	• -1: Scalar • 1 to 3: Number of dimensions	○ ^{*6}	—	○ ^{*6}
Dim1	Number of one-dimensional elements	Specify the number of elements in one dimension in either of the following cases: • The Rank is not 0. • The ArrayDimensions in one dimension are 0 (Any).	1 to 10240 ^{*7}	○	—	○
Dim2	Number of two-dimensional elements	Specify the number of elements in two dimensions in either of the following cases: • The Rank is 2 or higher. • The ArrayDimensions in two dimensions are 0 (Any).	1 to 10240 ^{*7}	○	—	○
Dim3	Number of three-dimensional elements	Specify the number of elements in three dimensions in either of the following cases: • The Rank is 3. • The ArrayDimensions in three dimensions are 0 (Any).	1 to 10240 ^{*7}	○	—	○
Sign	Change determination information	It is used to determine if mapping information is changed in a location other than the configuration tool. (Not editable)	—	○	—	—

Element: MethodMapping

Attribute ^{*1,2}	Attribute value ^{*3}	Description	Range	Element type (⊙: Required, ○: Optional, —: Not supported ^{*4})			
				Method Mapping	Handle	Input	Argument
IsArray	true	When a Node is an Array, describe "true." (Required when a Node is an Array)	true	—	—	—	○
IsStructure	true	When a Node is a Structure, describe "true." (Required when a Node is a Structure)	true	—	—	—	○
Num	Element number	Describe the element number of an Array. (Example) "0"	0 to 10239	—	—	—	⊙
Name	Name	Describe the Name of a Field. (Example) "ProductName"	—	—	—	—	—
Data ^{*5}	Data name	Describe a global label. (Example) "label01"	1 to 800 characters	—	⊙	—	○
Rank	Number of array dimensions	When the ValueRank is any of the following, specify the number of dimensions of an array to be mapped. • -3 (ScalarOrOneDimension) • -2 (Any) • 0 (OneOrMoreDimensions)	• -1: Scalar • 1 to 3: Number of dimensions	—	—	—	—
Dim1	Number of one-dimensional elements	Specify the number of elements in one dimension in either of the following cases: • The Rank is not 0. • The ArrayDimensions in one dimension are 0 (Any).	1 to 10240 ^{*7}	—	—	—	—
Dim2	Number of two-dimensional elements	Specify the number of elements in two dimensions in either of the following cases: • The Rank is 2 or higher. • The ArrayDimensions in two dimensions are 0 (Any).	1 to 10240 ^{*7}	—	—	—	—
Dim3	Number of three-dimensional elements	Specify the number of elements in three dimensions in either of the following cases: • The Rank is 3. • The ArrayDimensions in three dimensions are 0 (Any).	1 to 10240 ^{*7}	—	—	—	—
Sign	Change determination information	It is used to determine if mapping information is changed in a location other than the configuration tool. (Not editable)	—	○	—	—	—

*1 When not describing an attribute, either of the following applies depending on whether it is required or optional.

- Required: Mapping information is treated as not described, and this is reported on the configuration tool.
- Optional: An attribute is treated as omitted.

*2 When describing an unavailable attribute (not listed in the table), mapping information is treated as not described, and this is reported on the configuration tool.

*3 If it is an unavailable numerical value or character, either of the following applies depending on whether it is required or optional.

- Required: Mapping information is treated as not described, and this is reported on the configuration tool. This treatment also applies to a null character (" " only).
- Optional: An attribute is treated as omitted.

*4 Ignored when describing an attribute.

*5 When a Node is an Array or Structure, describe an attribute in the Element or Field.

- When describing an attribute in the DataMapping, this is reported on the configuration tool.
- Describe an attribute in an Element indicating a high dimension (example: two dimensions when the ValueRank is 2). When describing an attribute in an Element other than a high dimension, this is reported on the configuration tool.
- When a Field is an Array, or an Array and Structure, describe an attribute in the Element. When describing an attribute in the Field tag, this is reported on the configuration tool.
- When a Field is a Structure (parent), describe an attribute in the Field of its child. When describing an attribute in the Field tag of the parent, this is reported on the configuration tool.

*6 Cannot be optional if mapping array specification is set. If it is optional, the Dim1 to 3 are ignored.

*7 If the upper limit is exceeded, an attribute is treated as omitted.

In addition, if the product of the numbers of elements in valid dimensions exceeds the upper limit (10240), the Dim1 to 3 are treated as omitted.

Extension format

The following shows the Extension format.

If the format is not respected, mapping information is treated as not described and this is reported on the configuration tool.

Point

xmlns="https://www.mitsubishielectric.com/fa/OPCUA/extension/" and Sign="xxxxxxx" are omitted in the format explained below.

For actual description, describe 'xmlns' in the Extension. (📖 Page 154 Description rules for the xmlns)

■DataMapping

The DataMapping format is as follows:

- <DataMapping Data="{Label name}"/>

■Element

The Element format is as follows:

Describe mapping information of each element of an Array and data in the Element. For a multidimensional Array, describe it in an Element hierarchy, and the deeper the hierarchy, the higher the dimension.

- ValueRank = 1, ArrayDimensions = {3} (example: Int32[3])

```
<DataMapping IsArray="true">
```

```
  <Element Num="0" Data="{Label name}"/>
```

```
  <Element Num="1" Data="{Label name}"/>
```

```
  <Element Num="2" Data="{Label name}"/>
```

```
</DataMapping>
```

- ValueRank = 2, ArrayDimensions = {2,3} (example: Int32[2][3])

```
<DataMapping IsArray="true">
```

```
  <Element Num="0">
```

```
    <Element Num="0" Data="{Label name}"/>
```

```
    <Element Num="1" Data="{Label name}"/>
```

```
    <Element Num="2" Data="{Label name}"/>
```

```
  </Element>
```

```
  <Element Num="1">
```

```
    <Element Num="0" Data="{Label name}"/>
```

```
    <Element Num="1" Data="{Label name}"/>
```

```
    <Element Num="2" Data="{Label name}"/>
```

```
  </Element>
```

```
</DataMapping>
```

- ValueRank = 3, ArrayDimensions = {2,3,4} (example: Int32[2][3][4])

```

<DataMapping IsArray="true">
  <Element Num="0">
    <Element Num="0">
      <Element Num="0" Data="{Label name}"/>
      <Element Num="1" Data="{Label name}"/>
      <Element Num="2" Data="{Label name}"/>
      <Element Num="3" Data="{Label name}"/>
    </Element>
    <Element Num="1">
      <Element Num="0" Data="{Label name}"/>
      <Element Num="1" Data="{Label name}"/>
      <Element Num="2" Data="{Label name}"/>
      <Element Num="3" Data="{Label name}"/>
    </Element>
    <Element Num="2">
      <Element Num="0" Data="{Label name}"/>
      <Element Num="1" Data="{Label name}"/>
      <Element Num="2" Data="{Label name}"/>
      <Element Num="3" Data="{Label name}"/>
    </Element>
  </Element>
  <Element Num="1">
    <Element Num="0">
      <Element Num="0" Data="{Label name}"/>
      <Element Num="1" Data="{Label name}"/>
      <Element Num="2" Data="{Label name}"/>
      <Element Num="3" Data="{Label name}"/>
    </Element>
    <Element Num="1">
      <Element Num="0" Data="{Label name}"/>
      <Element Num="1" Data="{Label name}"/>
      <Element Num="2" Data="{Label name}"/>
      <Element Num="3" Data="{Label name}"/>
    </Element>
    <Element Num="2">
      <Element Num="0" Data="{Label name}"/>
      <Element Num="1" Data="{Label name}"/>
      <Element Num="2" Data="{Label name}"/>
      <Element Num="3" Data="{Label name}"/>
    </Element>
  </Element>
</DataMapping>

```


The Rank and Dim1 to Dim3 formats are as follows:

The Rank and Dim1 to Dim3 can be set in the "MappingArray" screen of the configuration tool. (🔍 Page 115 Mapping array specification)

- ValueRank = 1, ArrayDimensions = {0}, and Dim1 = 3 (example: Int32[3])

```
<DataMapping IsArray="true" Rank="1" Dim1="3">
  <Element Num="0" Data="{Label name}"/>
  <Element Num="1" Data="{Label name}"/>
  <Element Num="2" Data="{Label name}"/>
</DataMapping>
```

- ValueRank = -2, ArrayDimensions = {0}, and Rank = 1, Dim1 = 3 (example: Int32[3])

```
<DataMapping IsArray="true" Rank="1" Dim1="3">
  <Element Num="0" Data="{Label name}"/>
  <Element Num="1" Data="{Label name}"/>
  <Element Num="2" Data="{Label name}"/>
</DataMapping>
```

- ValueRank = 0, ArrayDimensions = {0,0}, and Rank = 2, Dim1 = 2, Dim2 = 3 (example: Int32[2][3])

```
<DataMapping IsArray="true" Rank="2" Dim1="2" Dim2="3">
  <Element Num="0">
    <Element Num="0" Data="{Label name}"/>
    <Element Num="1" Data="{Label name}"/>
    <Element Num="2" Data="{Label name}"/>
  </Element>
  <Element Num="1">
    <Element Num="0" Data="{Label name}"/>
    <Element Num="1" Data="{Label name}"/>
    <Element Num="2" Data="{Label name}"/>
  </Element>
</DataMapping>
```

■Field

The Field format is as follows:

Describe mapping information of each Field of a Structure and data in the Field. When the Field is a Structure, describe it in a hierarchy of the Field tag. In addition, when the Structure or Field is an Array, describe it in a combination with an Element.

- For a Structure

```
<DataMapping IsStructure="true">
  <Field Name="{Name }" Data="{Label name}"/>
  <Field Name="{Name }" Data="{Label name}"/>
</DataMapping>
```

- When a Field of a Structure is a Structure (one of the three Fields is a Structure)

```
<DataMapping IsStructure="true">
  <Field Name="{Name}" IsStructure="true">
    <Field Name="{Name}" Data="{Label name}"/>
    <Field Name="{Name}" Data="{Label name}"/>
  </Field>
  <Field Name="{Name}" Data="{Label name}"/>
  <Field Name="{Name}" Data="{Label name}"/>
</DataMapping>
```

- When a Structure is an Array (for the Array, ValueRank = 1, ArrayDimensions = {2})

```
<DataMapping IsArray="true" IsStructure="true">
  <Element Num="0">
    <Field Name="{Name}" Data="{Label name}"/>
    <Field Name="{Name}" Data="{Label name}"/>
  </Element>
  <Element Num="1">
    <Field Name="{Name}" Data="{Label name}"/>
    <Field Name="{Name}" Data="{Label name}"/>
  </Element>
</DataMapping>
```

- When a Field of a Structure is an Array (one of the three Fields is an Array (for the Array, ValueRank = 1, ArrayDimensions = {2}))

```
<DataMapping IsStructure="true">
  <Field Name="{Name}" IsArray="true">
    <Element Num="0" Data="{Label name}"/>
    <Element Num="1" Data="{Label name}"/>
  </Field>
  <Field Name="{Name}" Data="{Label name}"/>
  <Field Name="{Name}" Data="{Label name}"/>
</DataMapping>
```

The format for descriptions in combination with the Rank and Dim1 to Dim3 is as follows:

The Rank and Dim1 to Dim3 can be set in the "MappingArray" screen of the configuration tool. ( Page 115 Mapping array specification)

- When a Structure is an Array (for the Array, ValueRank = 1, ArrayDimensions = {0}, and Dim1 = 2)

```
<DataMapping IsArray="true" IsStructure="true" Rank="1" Dim1="2">
  <Element Num="0">
    <Field Name="{Name}" Data="{Label name}"/>
    <Field Name="{Name}" Data="{Label name}"/>
  </Element>
  <Element Num="1">
    <Field Name="{Name}" Data="{Label name}"/>
    <Field Name="{Name}" Data="{Label name}"/>
  </Element>
</DataMapping>
```

- When a Field of a Structure is an Array (one of the three Fields is an Array (for the Array, ValueRank = 0, ArrayDimensions = {0}, and Rank = 1, Dim1 = 2))

```
<DataMapping IsStructure="true">
  <Field Name="{Name}" IsArray="true" Rank="1" Dim1="2">
    <Element Num="0" Data="{Label name}"/>
    <Element Num="1" Data="{Label name}"/>
  </Field>
  <Field Name="{Name}" Data="{Label name}"/>
  <Field Name="{Name}" Data="{Label name}"/>
</DataMapping>
```

■MethodMapping

The MethodMapping format is as follows:

```
<MethodMapping>
  <Handle Data="{Label name}"/>
  <Input>
    <Argument Num="0" Data="{Label name}"/>
    <Argument Num="1" Data="{Label name}"/>
    <Argument Num="2" Data="{Label name}"/>
  </Input>
</MethodMapping>
```

When an Argument is an Array or Structure, the description method is the same as DataMapping.

- When an Argument is an Array (the first of the two Arguments is an Array with ValueRank = 1 and ArrayDimensions = {3})

```
<MethodMapping>
  <Handle Data="{Label name}"/>
  <Input>
    <Argument Num="0" IsArray="true">
      <Element Num="0" Data="{Label name}"/>
      <Element Num="1" Data="{Label name}"/>
      <Element Num="2" Data="{Label name}"/>
    </Argument>
    <Argument Num="1" Data="{Label name}"/>
  </Input>
</MethodMapping>
```

- When an Argument is a Structure (the first of the two Arguments is a Structure)

```
<MethodMapping>
  <Handle Data="{Label name}"/>
  <Input>
    <Argument Num="0" IsStructure="true">
      <Field Name="{Name}" Data="{Label name}"/>
      <Field Name="{Name}" Data="{Label name}"/>
    </Argument>
    <Argument Num="1" Data="{Label name}"/>
  </Input>
</MethodMapping>
```

- When an Argument is a Structure, and a Field of the Structure is a Structure (the first of the two Arguments is a Structure, and one of the three Fields is a Structure)

```
<MethodMapping>
  <Handle Data="{Label name}"/>
  <Input>
    <Argument Num="0" IsStructure="true">
      <Field Name="{Name}" IsStructure="true">
        <Field Name="{Name}" Data="{Label name}"/>
        <Field Name="{Name}" Data="{Label name}"/>
      </Field>
      <Field Name="{Name}" Data="{Label name}"/>
      <Field Name="{Name}" Data="{Label name}"/>
    </Argument>
    <Argument Num="1" Data="{Label name}"/>
  </Input>
</MethodMapping>
```

- When an Argument is a Structure and Array (the first of the two Arguments is a Structure and Array, and the Array has ValueRank = 1 and ArrayDimensions = {2})

```
<MethodMapping>
  <Handle Data="{Label name}"/>
  <Input>
    <Argument Num="0" IsArray="true" IsStructure="true">
      <Element Num="0">
        <Field Name="{Name}" Data="{Label name}"/>
        <Field Name="{Name}" Data="{Label name}"/>
      </Element>
      <Element Num="1">
        <Field Name="{Name}" Data="{Label name}"/>
        <Field Name="{Name}" Data="{Label name}"/>
      </Element>
    </Argument>
    <Argument Num="1" Data="{Label name}"/>
  </Input>
</MethodMapping>
```

Appendix 9 Software Licenses and Copyrights

This section describes the licenses and copyrights of software used in information model mode.

For those for the tag mode and common to each mode, refer to the following:

📖 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

Official OPC UA .NET Standard Stack from the OPC Foundation

This repository includes the UA .NET Stack sample libraries and sample applications, all based on the .NET StandardLibrary.

The UA .NET StandardLibrary Stack follows a dual-license:

OPC Foundation Corporate Members: RCL

Everybody else: GPL 2.0

RCL enables OPC Foundation members to deploy their applications using the software without being required to disclose the application code. Non-members licensed under GPL 2.0 must disclose their application code when using the software.

If OPC Foundation Corporate Membership is terminated, the licence reverts to GPL 2.0 whenever any fixes or updates published on GitHub are applied to the former member's application. This includes updates acquired by fetching any Git commit made after membership termination or the use of binaries distributed in OPC NuGet packages that were posted after the membership termination date.

Note: Dual license applies to this repository only; GPL 2.0 applies to all other repositories

All samples, the platform layers, and files generated with the ModelCompiler by means of the OPC UA NodeSet are provided under the MIT license.

SVG.NET

Microsoft Public License (Ms-PL)

This license governs use of the accompanying software. If you use the software, you accept this license. If you do not accept the license, do not use the software.

1. Definitions

The terms "reproduce," "reproduction," "derivative works," and "distribution" have the same meaning here as under U.S. copyright law.

A "contribution" is the original software, or any additions or changes to the software.

A "contributor" is any person that distributes its contribution under this license.

"Licensed patents" are a contributor's patent claims that read directly on its contribution.

2. Grant of Rights

(A) Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create.

(B) Patent Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use,

sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software.

3. Conditions and Limitations

(A) No Trademark License- This license does not grant you rights to use any contributors' name, logo, or trademarks.

(B) If you bring a patent claim against any contributor over patents that you claim are infringed by the software, your patent license from such contributor to the software ends automatically.

(C) If you distribute any portion of the software, you must retain all copyright, patent, trademark, and attribution notices that are present in the software.

(D) If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license.

(E) The software is licensed "as-is." You bear the risk of using it. The contributors give no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this license cannot change. To the extent permitted under your local laws, the contributors exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement.

CSharp_TreeListView

MIT License

Copyright (c) 2018 TangHanF

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ComponentOne

ComponentOne

Copyright (C) 2004 GrapeCity inc.

A

Appendix 10 Use in a Redundant System

For details, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

Appendix 11 Added and Changed Functions

For details, refer to the following:

 MELSEC iQ-R OPC UA Server Module User's Manual (Application)

A

INDEX

A

Application authentication 128

C

Call 57
CreateMonitoredItems 54

D

Data reading 80
Data writing 87
DeleteMonitoredItems 56

G

Global label 76
Global label-related data update 88

H

Handle 79
High-speed sampling 80,84

L

Label Node 50

M

Mapping array specification 115
MELNS 51
Method 79,90
ModifyMonitoredItems 56

N

Node 78

R

Read 52

U

User authentication 127

V

Variable 78

W

Write 53

MEMO

REVISIONS

*The manual number is given on the bottom left of the back cover.

Revision date	*Manual number	Description
May 2024	SH(NA)-082679ENG-A	First edition
April 2025	SH(NA)-082679ENG-B	■Added or modified parts Section 2.1, Section 4.1, Section 4.2, Section 8.1, Section 8.2, Section 8.3, Section 9.4, Section 9.5, Section 9.12, Section 10.3, Appendix 7, Appendix 8, Appendix 11
June 2025	SH(NA)-082679ENG-C	■Added or modified parts Section 2.1, Section 9.4, Appendix 6
April 2026	SH(NA)-082679ENG-D	■Added or modified parts TERMS, Section 2.1, Chapter 3, Section 4.1, Section 4.2, Section 8.1, Section 8.2, Section 8.3, Section 8.6, Section 9.4, Section 9.5, Section 9.6, Section 9.7, Section 9.8, Section 9.13, Section 10.3, Appendix 7

Japanese manual number: SH-082678-D

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2024 MITSUBISHI ELECTRIC CORPORATION

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company.

However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

- (1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.
- (2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
 1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
 2. Failure caused by unapproved modifications, etc., to the product by the user.
 3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
 4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
 5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
 6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
 7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

- (1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued. Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.
- (2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to:

- (1) Damages caused by any cause found not to be the responsibility of Mitsubishi.
- (2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.
- (3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.
- (4) Replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

INFORMATION AND SERVICES

For further information and services, please contact your local Mitsubishi Electric sales office or representative.
Visit our website to find our locations worldwide.

MITSUBISHI ELECTRIC Factory Automation Global Website
Locations Worldwide
www.MitsubishiElectric.com/fa/about-us/overseas/

TRADEMARKS

Intel is a trademark of Intel Corporation or its subsidiaries.

Microsoft, Excel, and Windows are trademarks of the Microsoft group of companies.

Unicode is either a registered trademark or a trademark of Unicode, Inc. in the United States and other countries.

The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.

In some cases, trademark symbols such as TM or [®] are not specified in this manual.

COPYRIGHTS

For the open source software and third party software components used in this product, refer to the following:

 Page 162 Software Licenses and Copyrights

TRANSLATION OF THE ORIGINAL INSTRUCTIONS

SH(NA)-082679ENG-D(2604)MEE

MODEL: RD81OPC96-U-JM-E

mitsubishi electric corporation

HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA 461-8670, JAPAN

When exported from Japan, this manual does not require application to the
Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.