



DeviceXPlorer OPC Server
User's Guide
(Server Edition)



Revision 032

August 2023

[Precautions regarding this manual]

- (1) Read this manual and understand it thoroughly before operating the product.
- (2) This manual explains the details of the product's functions; it does not guarantee that the product is suitable for your particular purposes.
- (3) Publishing or copying this manual or any part thereof without permission is strictly prohibited.
- (4) The content of this manual is subject to change without notice.
- (5) The content of this manual applies to program versions 7.2.0.1 or higher.

[Applicability of this product]

TAKEBISHI provides no implied warranty other than the terms stipulated in the warranty conditions.

Conditions regarding the use of this product includes using the product in applications that will not result in serious accident in the event of a product failure or malfunction and systematic implementation of backup and failsafe functions in preparation for occurrences of failures and malfunctions.

This product is designed and manufactured as a general-purpose product for general industrial applications. For this reason, this product is not applicable for use with the following types of specialized devices or systems. If this product is used in such applications, TAKEBISHI shall not be liable for any loss of product quality, performance, or safety including but not limited to non-performance of contractual obligations, warranty against defects, warranty against quality issues, tort liability, and product liability.

- Applications that have a significant impact on the public including nuclear and other power plants operated by power companies.
- Railway, government, or other applications that require special quality assurance.
- Applications that have significant impact on human life and property including aviation, medical, railway, combustion/fuel systems, passenger vehicles, manned transport devices, entertainment machines, and safety equipment.

However, use in these types of applications may be allowable if the customer acknowledges that special quality requirements cannot be accommodated for such specialized applications. Please contact us to discuss your specific application.

TAKEBISHI shall not be responsible for any damages including incidental damages incurred by the customer or a third party due to the use of this product or due to unforeseen product defects.

[Copyrights]

TAKEBISHI Corporation holds the copyrights to the programs contained in this product, the online manual, and other documentation. It is prohibited to copy this content or to transfer to a third party. It is also prohibited to record the content, using video tape or other media, without permission.

[Trademarks]

All company names, product names and trademarks that appear in this manual are trademarks of their respective owners.

1	Introduction.....	8
1.1	Overview	8
1.2	Features	8
1.3	Protocol for Communication between Applications	9
1.4	Licenses.....	9
2	Configuration.....	10
2.1	System Configuration.....	10
2.2	Operating Environment.....	11
2.3	Support Controller	12
3	General Performance Specifications	15
3.1	Specifications Table	15
3.2	OPC DA Interfaces	16
3.2.1	Prog.ID.....	16
3.2.2	Item ID.....	16
3.2.3	Access Path	17
3.2.4	OPC Custom Interfaces.....	17
3.2.5	OPC Automation Interface	20
3.2.6	Quality flag.....	20
3.2.7	Error code.....	21
3.3	OPC A&E Interface.....	23
3.3.1	Prog.ID.....	23
3.3.2	ItemID	23
3.3.3	Custom Interfaces	24
3.4	OPC UA Interface	25
3.4.1	Specifications Table	25
3.4.2	Service	26
3.5	SuiteLink Interfaces	27
3.5.1	Application Name	27
3.5.2	Topic Names and Item Names.....	27
3.6	DxpLink Interface.....	28
3.6.1	DxpLink Server Settings	28
3.6.2	DxpLink Communication Settings	28
3.6.3	DxpLink Communication.....	29
3.6.4	DxpLink Protocol Specification.....	30
3.6.4.1	Enumerate Device (GetDevice)	30
3.6.4.2	Enumerate Tag (GetTags).....	31
3.6.4.3	Read Value of Tag (GetTagsValue).....	32
3.6.4.4	Write Value of Tag (SetTagsValue).....	33
3.6.4.5	Constants	34
3.7	Modbus/TCP Server Interface.....	35
3.8	HTTP Client Interface.....	35
3.9	MQTT Client Interface	35
4	Operation	36
4.1	Tutorial	36
4.1.1	Installing and Setting Up an Ethernet Adapter.....	36
4.1.2	Installing DeviceXPlorer	37
4.1.3	Starting DeviceXPlorer.....	37
4.1.4	Start Page	37
4.1.5	Project Wizard.....	37
4.1.6	Accessing DeviceXPlorer from a Client	40
4.2	Screen Layout	42
4.2.1	Menus	45

4.2.2	Toolbar	48
4.2.3	Another Shortcut Key	48
4.2.4	Property Grid	50
4.2.5	Clip of List View	50
4.3	Ports	51
4.3.1	Creating New Ports	52
4.3.2	Changing the Port Name	53
4.3.3	Port Settings	53
4.3.4	Setting Ethernet Ports	55
4.3.4.1	Timeout Settings for TCP / IP	57
4.3.5	Serial Port Settings	59
4.3.6	Port Generator	60
4.3.7	Dynamic Port	62
4.4	Devices	63
4.4.1	Creating a New Device	64
4.4.1.1	Not select a Port	64
4.4.1.2	Select a Port	65
4.4.2	Connection Test	67
4.4.3	Changing the Device Name	68
4.4.4	Device Settings	68
4.4.5	Device Options	69
4.4.6	Device User Authority	71
4.4.7	Changing Ports	72
4.4.8	Device Generator	73
4.4.9	Dynamic Device	75
4.5	Groups	76
4.5.1	Creating a New Group	77
4.5.2	Changing the Group Name	78
4.5.3	Group Settings	78
4.5.4	Deleting Group	80
4.6	Device Folder	81
4.6.1	Creating a New Device Folder	82
4.6.2	Changing the Group Name	83
4.6.3	Deleting Device Folder	84
4.7	Tags	85
4.7.1	Dynamic Tags and Static Tags	85
4.7.2	Creating a New Tag	86
4.7.3	Changing a Tag Name	87
4.7.4	Properties (General)	89
4.7.5	Properties (Scale Settings)	90
4.7.6	Properties (Simulation)	91
4.7.7	Properties (Alarm)	92
4.7.8	Properties (Historical)	93
4.7.9	Saving Settings	93
4.7.10	Adding Tags Continuously	93
4.7.11	Tag Generator	93
4.7.12	Extended Tag Name Specification	96
4.7.12.1	No Specification	97
4.7.12.2	Specifying the Bit Position (:nn)	97
4.7.12.3	Byte Type (:L, :LU, :H, :HU)	97
4.7.12.4	Word Type (:W, :U, :WU)	97
4.7.12.5	Long Type (:D, :DU)	97

4.7.12.6	LongLong Type (:I, :IU)	97
4.7.12.7	Single Precision Real Number Type (:R)	97
4.7.12.8	Double Precision Real Number Type (:DR).....	97
4.7.12.9	BCD Type (:B).....	97
4.7.12.10	Read-only/Write-only (:OR, :OW)	98
4.7.12.11	Array (:A99999).....	98
4.7.12.12	String (:S99999).....	98
4.7.12.13	File Batch Write(:F999999).....	98
4.7.12.14	Text Format Binary Type (:B99999)	98
4.7.12.15	Text Format Octal Type (:O99999).....	99
4.7.12.16	Text Format Decimal Type (:D99999).....	99
4.7.12.17	Text Format Hexadecimal Type (:H99999)	99
4.7.12.18	Bit Inverse.....	99
4.7.12.19	Boolean conversion	100
4.7.12.20	Simulation mode (Sin)	100
4.7.12.21	Simulation mode (Shared Memory)	100
4.7.12.22	Simulation mode (Ramp)	100
4.7.12.23	Simulation mode (Random)	100
4.7.13	System Tags.....	101
4.7.13.1	Global System Tags	101
4.7.13.2	Device System Tags	103
4.7.13.3	Connection System Tags	104
4.7.14	Tag Monitor	105
4.7.15	Offline Mode	106
4.7.16	Write Tag Value.....	107
4.7.17	Watch	108
4.7.18	Array Viewer.....	109
4.7.19	Register Monitor.....	110
4.8	Method Tag.....	113
4.8.1	Creating a New Method Tag	113
4.8.2	Execute Method Tag	114
4.9	Structure Tag	115
4.9.1	Define Structure Template	115
4.9.2	Creating a New Structure Tag.....	119
4.9.3	How to access Structure Tags from Client	121
4.10	Bridge.....	122
4.10.1	Bridge Setting	122
4.11	Messages	125
4.11.1	Message.....	125
4.11.2	How to save messages automatically	128
4.12	Diagnosis.....	128
4.12.1	Interface Diagnosis	129
4.12.2	Client Diagnosis.....	130
4.12.3	Group Diagnosis	130
4.12.4	Tag Diagnosis	131
4.12.5	Event Diagnosis.....	132
4.13	IoT Interface.....	133
4.13.1	OPC DA Server	133
4.13.2	OPC UA Server	135
4.13.2.1	Generate Certificate Setting “UA Server”	137
4.13.2.2	Revers Connect Setting	138
4.13.3	SuiteLink Server	138

4.13.4	DxpLink Server.....	140
4.13.5	HTTP Client.....	141
4.13.5.1	HTTP-based Setting	141
4.13.5.2	HTTP group Setting	143
4.13.5.3	Outgoing Message(HTTP)	144
4.13.5.4	HTTP Tag List.....	145
4.13.6	MQTT Client.....	148
4.13.6.1	MQTT-based Setting.....	148
4.13.6.2	MQTT group Setting	150
4.13.6.3	Outgoing Message(MQTT)	151
4.13.6.4	MQTT Tag List.....	151
4.13.6.5	Subscribe Message Format.....	153
4.13.6.6	Connection settings with Azure IoT Hub	154
4.13.6.7	Azure IoT Hub connection settings in DeviceXPlorer.....	160
4.14	Historical Data	161
4.14.1	Historical Tag Setting	163
4.14.1.1	Add from Historical Tag List	163
4.14.1.2	Add from tag setting	164
4.14.1.3	Add by dragging and dropping tags into the historical group	165
4.15	Project.....	166
4.15.1	Common Properties, General	167
4.15.2	Common Properties, Settings.....	167
4.15.3	Project Properties, General.....	168
4.15.4	Project Properties, GUI.....	170
4.15.5	Project Properties, Log.....	171
4.16	Service Activation	172
4.16.1	Registering DeviceXPlorer as a service program.....	172
4.16.2	How to show DeviceXPlorer setting dialog of service program	174
4.17	Importing and Exporting.....	175
4.17.1	Import[Definition]	176
4.17.1.1	Import MX OPC Server DA/ UA Configuration File	177
4.17.2	Export[Definition].....	179
4.17.3	Import[TagMemory].....	180
4.17.4	Export[TagMemory].....	180
4.17.5	Tag Import.....	181
4.17.5.1	Import the tag definition file of KEPServerEX	182
4.17.5.2	Error List	185
4.18	Hot Configuration	186
4.19	Redundant Communication	187
4.19.1	Warm Standby	188
4.19.2	Cold Standby.....	188
4.20	Communication Period	188
4.21	Slow Polling Mode.....	189
4.22	User Management	190
4.22.1	Configuration.....	190
4.22.2	Contents List.....	191
4.22.3	Auto Login	192
4.23	Change display language dynamically.....	193
4.24	DCOM Configuration	194
4.24.1	Overview	194
4.24.2	Procedure	195
5	Script.....	206

5.1	Event Script Settings	206
5.2	Script Specification	208
5.3	Extend Specification	208
5.3.1	Reserved Word.....	208
5.3.2	Constant Value	208
5.3.3	Classes.....	210
5.3.3.1	Tag Access Class.....	211
5.3.3.2	Group Access Class	212
5.3.3.3	Device Access Class.....	212
5.3.3.4	Struct Access Class	213
5.3.3.5	Port Access Class.....	213
5.3.3.6	Event Script Access Class	213
5.3.3.7	Method Access Class.....	214
5.3.3.8	HTTP Client Access Class.....	215
5.3.3.9	HTTP request class.....	216
5.3.3.10	HTTP response class	216
5.3.3.11	MQTT Client Access Class.....	217
5.3.3.12	MQTT Publish Message Class	218
5.3.3.13	Time Class	218
5.3.3.14	Json Object Class	219
5.3.4	Functions.....	220
5.3.5	Access Macros.....	222
5.3.6	Priority of Finding Object.....	223
5.3.7	Units of Running Script.....	224
5.3.8	About Script Writing.....	224
5.4	Samples.....	225
5.4.1	Basic Example	225
5.4.2	Easy CSV Logging.....	226
5.4.3	Example which defines the class in the external file (Bridge).....	227
5.4.4	Example which defines the class in the external file (Timer).....	230
6	Troubleshooting.....	232
6.1	Error Messages	232
6.1.1	The source of the Response timeout error in the Ethernet connection.....	232
6.1.2	Ethernet Error Code (Winsock Error Code)	233
6.1.3	Serial Error Code	233
6.2	Others.....	234
6.2.1	If the application screen does not display properly.....	234
7	Appendix.....	235
7.1	How to use of the OPC Test Client.....	235
7.2	About Sample Programs.....	235
7.3	User Support	235
7.4	Copyright Information.....	235

1 Introduction

This manual explains how to use DeviceXPlorer OPC Server (hereafter referred to as DeviceXPlorer).

For information on the settings, operating method, and error codes, etc., of the PLCs, refer to the "User's Guide (PLC Communication Edition)".

For details on how to install DeviceXPlorer and its license, refer to "User's Guide (Setup Edition)".

1.1 Overview

OPC (Ole for Process Control) is an interface specification for communication between applications that was formulated by the OPC Foundation, a standardization body made up of hardware and software vendors from around the world. There are a number of OPC interfaces, including OPC DA for communication with field devices, OPC HDA for accessing historical data, and OPC AE for monitoring alarms. And The Unified Architecture (UA) is THE next generation OPC standard that provides a cohesive, secure and reliable cross platform framework for access to real time and historical data and events.

DeviceXPlorer is communication software that is compliant with the OPC DA interface. It provides functions to communicate with PLC. Using DeviceXPlorer allows the user application to access field devices such as PLCs without being constrained by the PLC type or the communication interface. In addition, the user application (hereafter referred to as the client) can be set up in either a local or remote PC, allowing the construction of distributed systems.

1.2 Features

The main features of DeviceXPlorer are as follows:

- Passed with the OPC compliance test.
- Confirmed connectivity with package products from around the world, through OPC Interoperability Workshop in the United States and Japan.
- Compliant with OPC DA2.05A / OPC DA3.0.
- Compliant with OPC A&E 1.10.
- Compliant with OPC UA.
- Supports SuiteLink.
- Supports DxpLink. This is original protocol based on HTTP Communication between DeviceXPlorer.
- Supports Ethernet communication and serial communication from a single server. Also supports communication with multiple PLCs such as MELSEC (Mitsubishi Electric Corporation), SYSMAC (OMRON Corporation), and TOYOPUC (JTEKT Corporation).
- Provides a variety of sample programs as accessories, for easy access from Visual Basic, VB.NET, and Excel, as well as Visual C/C++ and C#.NET.
- Provides scale conversion and BCD conversion inside DeviceXPlorer, reducing the work of sequence programming.
- Because communication with DeviceXPlorer is possible simply by setting up a communication unit such as Ethernet and RS232C on the PLC side, you do not need to add extra sequence programs in order to use DeviceXPlorer.
- It is possible to predefine separate device names in DeviceXPlorer for the PLC device names. It is also possible to access the PLCs without defining names, to save setup time.
- DeviceXPlorer can operate as a Windows service program.
- Because redundant communication is supported, it is possible to continue communication in the event of a communication failure by automatically switching the communication route.
- A simulation function allows you to develop an OPC client even in an environment in which it is not possible to use a PLC.
- DeviceXPlorer has the script function. Even when processing cannot be included in the client and PLC side, it can respond easily.

1.3 Protocol for Communication between Applications

DeviceXPlorer supports 6 types of protocol (OPC, OPC UA, SuiteLink, DxpLink, HTTP and MQTT) for communication with upper-tier applications.

[OPC]

OPC is a standard specification based on the Microsoft OLE/COM technology. It provides mutual connectivity between industrial applications, field devices and business applications, etc. In the past, application developers had to develop drivers for accessing field devices, but the use of OPC interfaces makes it possible to achieve seamless communication between HMI, SCADA, industrial applications and field applications, and also made reuse possible. It also allows for remote computing via DCOM (Distributed COM) in network environments.

[OPC UA]

OPC UA (Unified Architecture) is the new specification of OPC. This is based on SOAP/XML/Web Service which spread follows as the next-generation information infrastructure. In addition to integration of the existing specification (DA, A&E, HDA), the data exchange of a full-scale object base becomes possible by defining an information model.

The scope of OPC UA makes possible the safety and the stable data exchange which cannot stop in a plant floor but can be applied also to the domain of MES located more in a higher rank, or ERP.

[SuiteLink]

SuiteLink is a TCP/IP-based protocol developed by Wonderware. SuiteLink improves the reliability of communication by using TCP/IP, and also affixes time stamp and quality information, allowing the construction of highly reliable networks.

[DxpLink]

DxpLink is a HTTP based protocol developed by Takebishi.

This transmits the data between DeviceXPlorer(s) seamlessly.

[HTTP]

HTTP is a text-based protocol that can be used over TCP/IP. This is prevalently used in Web-based systems as a RESTful service. The DeviceXPlorer can exchange text messages via HTTP communication and link IoT services that have a REST API, packages, and information terminals.

(e.g. FUJITSU COLMINA, NTT Docomo Toammi for Docomo)

[MQTT]

MQTT is a lightweight and high-speed messaging protocol used on top of TCP/IP.

MQTT communication is based on a publish-subscribe model. MQTT simplifies data exchange between IoT services and the DeviceXPlorer, which creates the opportunity to utilize powerful analysis capabilities of IoT services.

(e.g. Amazon Web Services, Microsoft Azure, FUJITSU COLMINA)

1.4 Licenses

One DeviceXPlorer license can operate one PC.

The max number of accessible modules is 255 per one DeviceXPlorer.

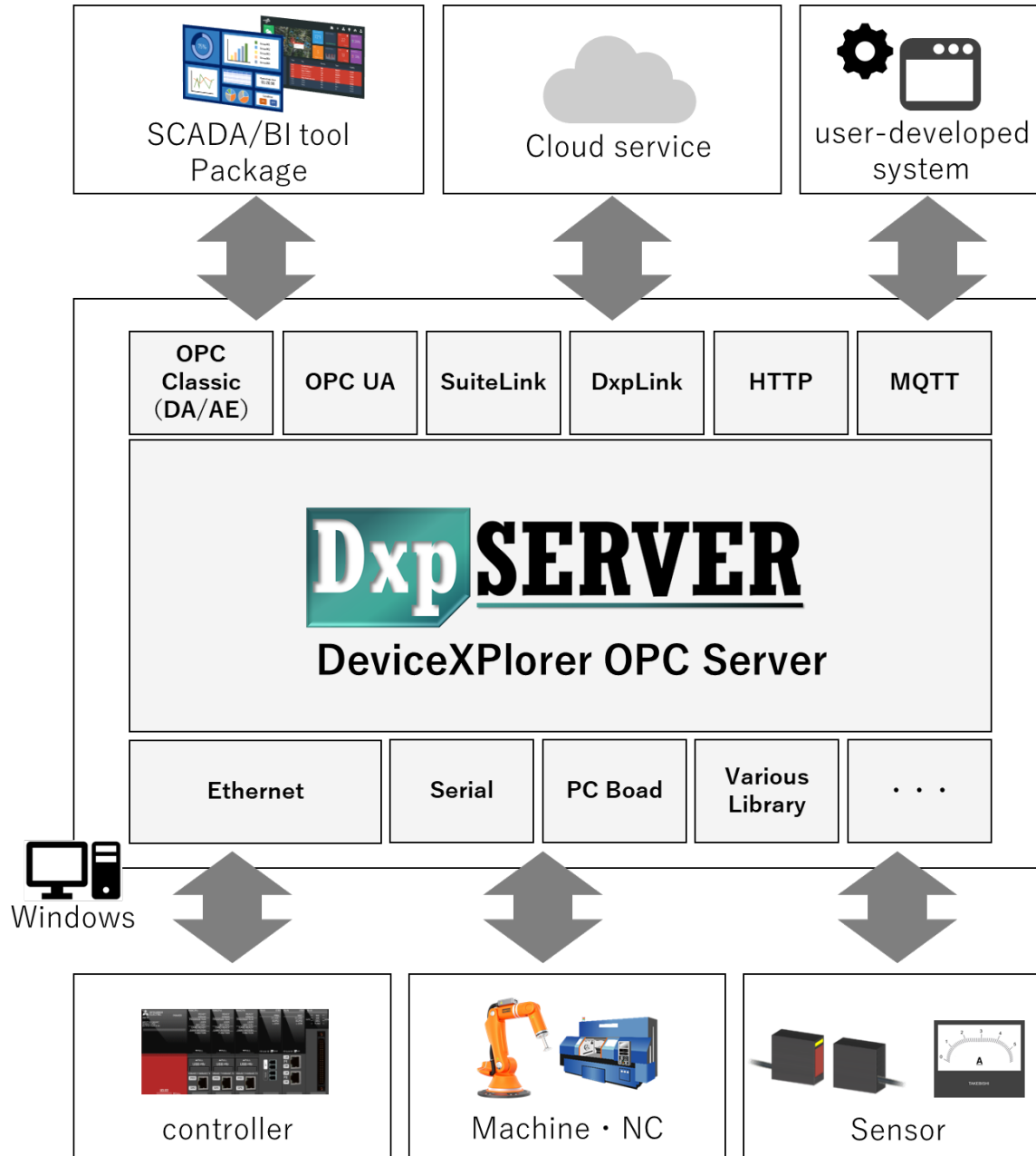
There is no constraint on the number of clients.

For the difference between each edition, refer to "User's Guide (Install Edition)".

2 Configuration

2.1 System Configuration

The DeviceXPlorer system configuration is as shown below.



- A single server supports multiple PLCs.
- The development languages that can be used to access each interface are as shown in the table below.
- Remote connection from a separate node via DCOM is possible. This is supported by OPC DA and OPC A&E.
- DxpLink, Data communications can be performed between DeviceXPlorer by DxpLink.
- MQTT and HTTP functions can be used to link with each company's cloud service.

2.2 Operating Environment

DeviceXPlorer operates in the following environments:

Item	Description
Computer	Computer running Windows
OS	Windows 11 (21H2, 22H2) *1
	Windows 10 IoT Enterprise 2021 LTSC
	Windows 10 IoT Enterprise 2019 LTSC
	Windows 10 IoT Enterprise 2016 LTSC
	Windows 10 (1809, 1903, 1909, 2004, 20H2, 21H2, 22H2) *1
	Windows Server 2022
	Windows Server 2019
	Windows Server 2016
	Windows Server 2012 R2
	Windows Server 2012
Application Type	32 bit application (WIN32 / x86)
	64 bit application (WIN64 / x64) *2 *3
Memory	1GB or more
Hard disk	1GB or more
Disc device	CD-ROM drive
LAN adapter	Adapter capable of operating on Windows
Framework	.NET Framework 3.5 (32 bit application only)

*1 There is a possibility that not be guaranteed to operate on the OS owing to the Windows Update.

In Windows10 Home Edition, Windows Update type is fixed "Current Branch"(update is performed automatically), so we are recommend that Pro Edition or above.

*2 SuiteLink interface does not support 64 bit native application. So 64 bit DeviceXPlorer does not support SuiteLink.

*3 64 bit DeviceXPlorer does not support the following communications because of limitation of communication library.

- MELSEC EZSocket / GOT connection.
- SYSMAC FinsGateway / SysmacGateway / CX-Compolet connection
- TOYOPUC CPU Port connection
- FANUC FOCAS connection
- Mitsubishi CNC EZSocketNc connection

2.3 Support Controller

DeviceXPlorer supports the following controllers:

Name	Interface	Target Module name
Mitsubishi Electric suite	Ethernet	MELSEC iQ-R / iQ-F / Q / L / QnA / A / FX series GOT series MC AppBuilder cooperation function
	Serial	
	Interface Board	
	EZSocket (including GX Simulator)	
	GOT	
Omron suite	Ethernet	SYSMAC-NX/ NJ/CJ / CS / CV / C series
	Serial	
	SysmacGateway	
	FinsGateway	
	CX-Compolet	
Yokogawa Electric suite	Ethernet	e-RT3 / FA-M3V / FA-M3(R)series
	Serial	
Jtekt suite	Ethernet	TOYOPUC-Nano / Plus / PC10 / PC3J / PC2J series
	CPU port (communication DLL)	
Hitachi suite	Ethernet	HIDIC-EHV / EH / H series
		HIDIC-S10 series
Modbus suite	Ethernet (Modbus/TCP)	Products supported by Modbus Mitsubishi Modbus module / Electric Controller / EcoMonitorLight OMRON K6CM, K7DD Hakko MONITOUCH Koyo DirectLOGIC/KOSTAC series IDEC MICRO Smart azbil(Yamatake) controller Hioki E. SmartSite2300 series MEE YA01-SAE / AE sensor RKC controller CHINO recorder/controller Eurotherm controller M-System remote IO / Tower Light Anywire DB A40 series Wago I/O System Cognex In-Sight Delta DVP series Panasonic FP / KT / KW series 3S CoDeSys SoftPLC Yokogawa STARDOM / Recorder / Control and Measurement Station Moxa switching hub Hitachi HX series IAI Position controller PHOENIX CONTACT remote IO MTT Alchis series PATLITE LA6 / AirGRID WD PRO series GRAPFTEC GLT400 series
	Serial (Modbus / RTU, ASCII)	

Keyence suite	Ethernet	KV8000 / 7500 / 7000 / 5500 / 5000 / 3000 / 1000 / 700 / Nano
Sharp suite	Ethernet	Satellite JW series
Fuji suite	Ethernet	MICREX-SX / F series
	Serial	
Yaskawa suite	Ethernet (MEMOBUS)	MP 3300 / 3200 / 2400 / 2300 / 2200 / 900 series
	Serial(MEMOBUS)	
Toshiba suite	Ethernet	nv / V / PROSEC T series
Panasonic suite	Ethernet	FP-X / FP2(SH)series
Rockwell suite	EtherNet/IP	GuardLogix / ControlLogix / CompactLogix / FlexLogix / DriveLogix series
Siemens suite	Ethernet	SIMATIC S7-1500 / S7-1200 / S7-400 / S7-300 / S7-200 series
Sbus suite	Ethernet	SAIA PCD1, PCD2, PCD3 series
GE suite	Serial(SNP)	90-30 series
Lsis suite	Ethernet	XGT series (XGK / XGI / XGR)
Kawasaki suite	Ethernet(DLL communication)	C / D/D+ / E / F / S/S+ / T / duAro D / duAro F controller
Yaskawa robot suite	Simulator Connection	Robot Simulator
Yamaha robot suite	Ethernet	RCX340 series / RCX-Studio(RCX340)
IAI robot suite	Serial	RSEL,X-SEL,TT,SSEL,ASEL,PSEL,MSEL series
Fanuc suite	Ethernet (FANUC FOCAS)	FANUC-PMC series
	Serial(DPRNT)	
Mitsubishi CNC	Serial(DPRNT)	M800/M80/E80,M700V/M70V/E70,M700/M70,C80,C70,M600,M60/60S,E60/E68 series C6/C64/C64T
	EZSocketNc	
Mitsubishi EDM	Ethernet	Wire, Die-sinker, Fine hole EDM
Shibaura Machine	Ethernet	TOSNUC series, TC200 series, TCmini series
	Serial	
Sumitomo Heavy Industries	Ethernet	SEEV-A, SE-DUZ, SEEV-A-HD, SE-HSZ, SE-HP, CL7000 series
Japan Steel Works	HTTP communication	J-ADS,J-AD,J-EL3,J-E3 series
Azbil(Yamatake) suite	Ethernet	NX series / CPL
	Serial	
JCC suite	Ethernet	Early Observer MEL-E Uno
Mitsubishi RFID suite	Ethernet	RF-RW series
Omron RFID suite	Serial	V680 series
Cognex BCR suite	Ethernet	DataMan
Keyence BCR suite	Ethernet	BC(BCR)/SR
MARS TOHKEN SOLUTION BCR suite	Ethernet	TFIR-3182
Mitsubishi Energy Saving Modules suite	HTTP communication	EcoServerⅢ / E-Energy
MTConnect suite	MTConnect	Products supported by MTConnect
DPRNT suite	Serial(DPRNT)	Equipment with serial output function
BACnet suite	Ethernet	Products supported by BACnet
IEC60870-5-104 Client	Ethernet	Products supported by IEC60870-5-104
IEC61850MMS Client	Ethernet	Wago PFC200 series ABB REX640 series ABB REF620 series Mitsubishi Electric MELPRO-D series

		Xeras IEC61850 Simulator
DNP3 Master	Ethernet	Products supported by DNP3
EtherNet/IP suite	EtherNet/IP	Products supported by EtherNet/IP
Modbus Server suite	Ethernet	Modbus/TCP Client GRAPFTEC MT1000 series
ODBC Client	ODBC	ODBC Database
OPC DA Client	OPC DA	OPC DA OPC Server
OPC UA Client	OPC UA	OPC UA OPC Server
OPC UA Subscriber	OPC UA	OPC UA Publisher
DDE Client	DDE	DDE Server
DxpLink Client	DxpLink (HTTP communication)	DxpLink Server (DeviceXPlorer)
Water CPS Application I/F	Water CPS Application I/F	Water CPS Application
Shared Memory	Shared memory	Internal memory of DeviceXPlorer

* For target module details, refer to the “User’s Guide (PLC Communication Edition)”.

* Please contact us if you wish to use OPC UA subscribers.

3 General Performance Specifications

3.1 Specifications Table

Item		Description
Service Activation		Possible
Language		Japanese, English, Simplified Chinese, Traditional Chinese, Korean
Max. ports/PLCs		255
Supported protocols		OPC-DA, OPC A&E, OPC UA, SuiteLink, DxpLink, HTTP, MQTT
Accessible devices		Refer to User's Guide (PLC Communication Edition).
Specification of different item (tag) names		Possible
OPC	OPC Item ID	Use delimiters to specify device, group and tag names.
	OPC DA	Version
		2.05A/3.0
		Prog.ID
		Takebishi.Dxp.7
		CLASS ID
		8D8CD4DC-1EA6-45bf-9AAE-A69D53A50007
		OPC Access Path
		Possible
	OPC A&E	Version
		1.10
		Prog.ID
		Takebishi.DxpAe.7
		CLASS ID
		8D8CD4DC-1EA6-45bf-9AAE-A69D53A50207
	OPC UA	URI
		opc.tcp://[machinename]:52250
		Protocol
		TCP , OPC UA Pub/Sub(UADP)
SuiteLink	Application Name	Any (default: "DXPV7")
	Topic name	Any (default: "Device name")
DxpLink	Port no	Any (default: 9980)
HTTP	Version	1.1
	Function Type	Client
	HTTP Authentication	None, basic authentication, digest authentication OAuth2.0 authorization(only ROPC)
	HTTP Method	GET, PUT, POST, DELETE
		OPTIONS, HEAD, TRACE
	Encryption	SSL/TLS
	Proxy Server	Supported (username/password authentication also supported)
	Max Connections	464servers
MQTT	Version	3.11, 3.1, Auto
	Function Type	Client (publisher)
	Protocol	TCP/IP (WebSockets not supported)
	Authentication	<MQTT broker> Username/password <Azure IoT Hub> Symmetric key in the identity registry Shared access policy, X509 certificates
	Encryption	SSL/TLS
	QoS	QoS0, QoS1, QoS2
	Retain/Will	Supported
	Max Connections	64brokers
Data type	VT_BOOL	Logical values
	VT_I1	8-bit signed integers
	VT_UI1	8-bit unsigned integers
	VT_I2	16-bit signed integers
	VT_UI2	16-bit unsigned integers
	VT_I4	32-bit signed integers
	VT_UI4	32-bit unsigned integers

	VT_R4	32-bit real number
	VT_I8	64-bit signed integers
	VT_UI8	64-bit unsigned integers
	VT_R8	64-bit real number
	VT_BSTR	Character string
	VT_ARRAY	Array
Extended format	Word device bit position specification	
	16-bit signed integers	
	16-bit unsigned integers	
	32-bit signed integers	
	32-bit unsigned integers	
	64-bit signed integers	
	64-bit unsigned integers	
	Integer BCD conversion	
	Single precision floating point	
	Double precision floating point	
	Character string	
	Text format binary	
	Text format octal	
	Text format decimal	
	Text format hexadecimal	
	Bit inverse	
	Boolean conversion	
	Batch File Write	
	Array	
	Read-only	
	Write-only	

3.2 OPC DA Interfaces

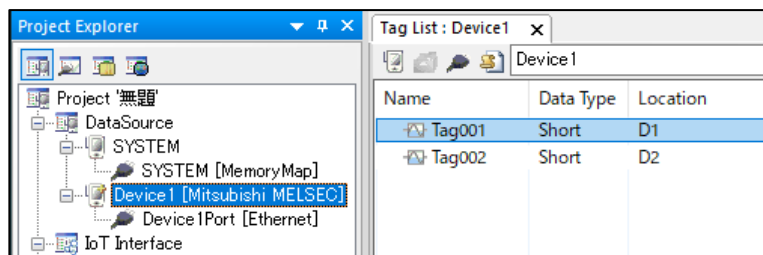
3.2.1 Prog.ID

The Prog.ID for DeviceXPlorer is “Takebishi.Dxp.7” or “Takebishi.Dxp”.

3.2.2 Item ID

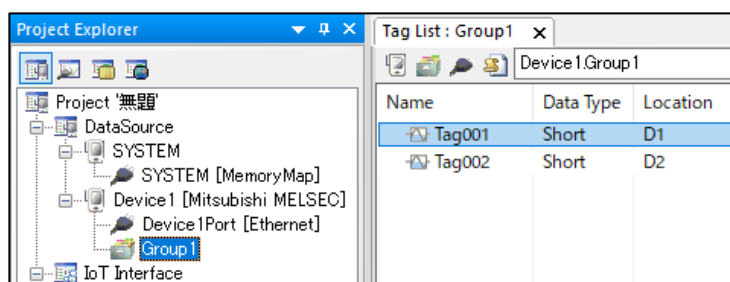
Specify the OPC Item ID as the device name, group name(Structure Tag name), and tag name set in DeviceXPlorer, separated by periods (.).

To access “Tag1” in the figure below, specify “Device1.Tag” as the item ID.



If there is a group attribution, specify the group name in between the device name and the tag name.

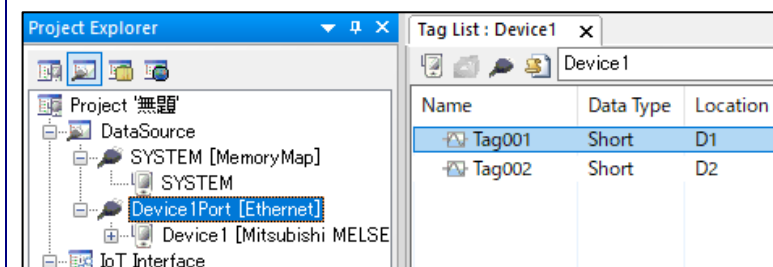
To access “Tag001” in the figure below, specify “Device1.Group1.Tag001” as the item ID.



Note

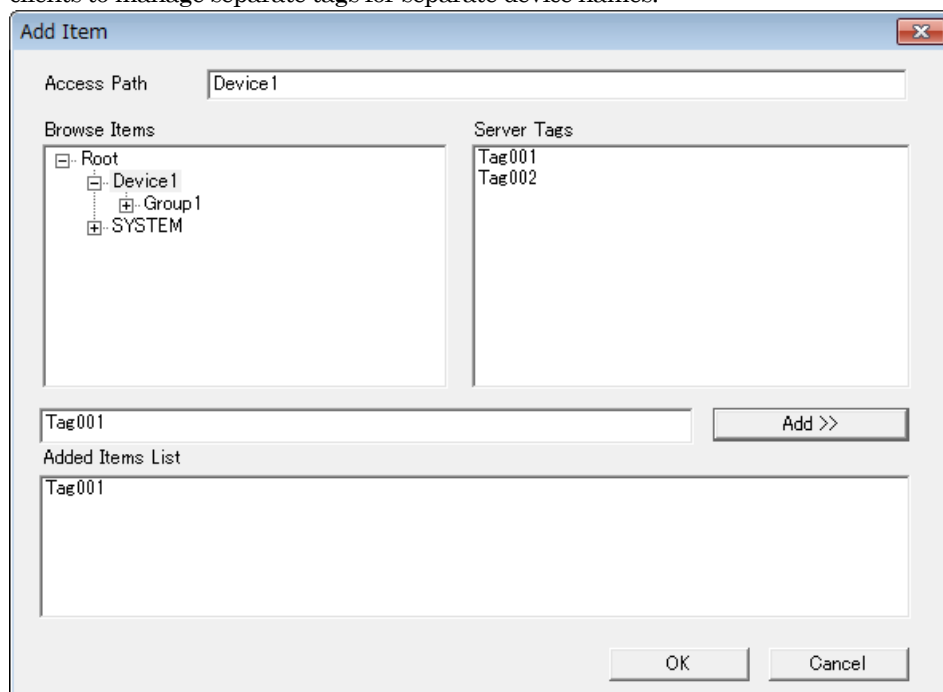
The item ID for DeviceXPlorer does not include the port name.

The port view appears as shown below, but the item ID is “Device1.Tag2”.



3.2.3 Access Path

If you specify an access path, you can omit the device name and group name from the item ID. For example, to access “Tag1” of “Device1”, you would normally need to specify “Device1.Tag1”, but if you specify “Device1” as the access path, the item ID becomes “Tag1”. By specifying an access path, you can specify simpler item IDs, which makes it possible for clients to manage separate tags for separate device names.



3.2.4 OPC Custom Interfaces

The support statuses of OPC DA custom interfaces are as shown below. The tables should be interpreted as follows:

Interface name	OPC DA interface name
DA2	The OPC DA2.05A interface is shown as “o”.

DA3	The OPC DA3.0 interface is shown as “○”.
Opt	The optional interface is shown as “○”.
Member function	Name of the member function in that interface
Support status	OK: Supported, N/A: Not supported

[OPC Server Object]

These are the main objects that expose the OPC server.

Interface name	DA2	DA3	Opt	Member function	Support status
IOPCCommon Shared interface	○	○		SetLocaleID	OK
				GetLocaleID	OK
				QueryAvailableLocalIDs	OK
				GetErrorString	OK
				SetClientName	OK
IOPCServer Main interface	○	○		AddGroup	OK
				GetErrorString	OK
				GetGroupByName	OK
				GetStatus	OK
				RemoveGroup	OK
				CreateGroupEnumerator	OK
IConnectionPointContainer Interface for accessing the connection points for IOPCShutdown	○	○		EnumConnectionPoints	OK
				FindConnectionPoints	OK
IOPCItemProperties Interface for browsing the available item properties	○	○		QueryAvailableProperties	OK
				GetItemProperties	OK
				LookutItemIDs	OK
IOPCBrowseServerAddressSpace Interface for browsing the available item IDs	○	○	○	QueryOrganization	OK
				ChangeBrowsePosition	OK
				BrowseOPCItemIDs	OK
				GetItemID	OK
				BrowseAccessPaths	N/A*1
IOPCServerPublicGroups Interface for managing public groups	○	○	○	GetPublicGroupByName	N/A
				RemovePublicGroup	N/A
IPersistFile Interface for storing and reading files of OPC server configuration information	○	○	○	IsDirty	N/A
				Load	N/A
				Save	N/A
				SaveCompleted	N/A
				GetCurFile	N/A
IOPCBrowse Interface for browsing the address space and obtaining item properties		○		GetProperties	OK
				Browse	OK
IOPCItemIO Interface for directly obtaining an item value without creating a group		○		Read	OK
				WriteVQT	OK

*1 E_NOTIMP will be returned.

[OPCGroup Object]

These objects allow the OPC server to manage the item collection.

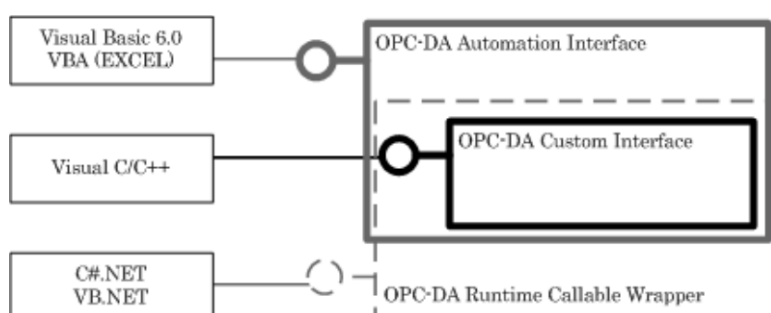
Interface name	DA2	DA3	Opt	Member function	Support status
IOPCGroupStateMgt Interface that manages the status of the whole group	○	○		GetState	OK
				SetState	OK
				SetName	OK
				CloneGroup	OK
IOPCPublicGroupStateMgt Interface for changing to a public group	○	○	○	GetState	N/A
				MoveToPublic	N/A
IOPCGroupStateMgt2 Interface for setting and running KeepAlive		○		SetKeepAlive	OK
				GetKeepAlive	OK
IOPCItemMgt Interface that controls the operation of items in the group	○	○		AddItems	OK
				ValidateItems	OK
				RemoveItems	OK
				SetActiveState	OK
				SetClientHandles	OK
				SetDatatypes	OK
				CreateEnumerator	OK
IOPCSyncIO Interface for synchronous reading / writing	○	○		Read	OK
				Write	OK
IOPCSyncIO2 Interface for synchronous reading using MaxAge / writing to the quality flag or time stamp		○		ReadMaxAge	OK
				WriteVQT	OK
IOPCAsyncIO Interface for asynchronous access	○	○		Read	OK
				Write	OK
				Refresh	OK
				Cancel	OK
IOPCASyncIO2 Interface for asynchronous access	○	○		Read	OK
				Write	OK
				Refresh	OK
				Cancel2	OK
				SetEnable	OK
				GetEnable	OK
IOPCASyncIO3 Interface for asynchronous access (using MaxAge)		○		ReadMaxAge	OK
				WriteVQT	OK
				Refresh2	OK
IConnectionPointContainer Interface for accessing the connection points for IOPCShutdown	○	○		EnumConnectionPoints	OK
IDataObject Interface to run on an OPC group	○	○		DAdvise	OK
				DUnadvise	OK
				FindConnectionPoint	OK
IOPCItemDeadbandMgt Interface for specifying the dead band of an item		○		SetItemDeadband	OK
				GetItemDeadband	OK
				ClearItemDeadband	OK
IOPCItemSamplingMgt Interface for specifying the sampling interval for each item		○	○	SetItemSamplingRate	OK
				GetItemSamplingRate	OK
				ClearItemSamplingRate	OK
				SetItemBufferEnable	OK
				GetItemBufferEnable	OK

[EnumOPCItem Attributes Object]

Interface name	DA2	DA3	Opt	Member function	Support status
IEnumOPCItemAttributes Searches the group items and attributes	○	○		Next	OK
				Skip	OK
				Reset	OK
				Clone	OK

3.2.5 OPC Automation Interface

The automation interface is a wrapper interface for the purpose of using custom interfaces from interpreter languages such as VB and VBA. It offers the same interface functions as the custom interface. For details, see the OPC DA Automation Interface Specification (issued by the OPC Foundation).



3.2.6 Quality flag

The quality flags returned by DeviceXPlorer to the client are as follows.

For details of each quality flag, see the OPC Data Access specification that comes with DeviceXPlorer.

Quality flag	Value	Description
OPC_QUALITY_BAD	0x00	NG
OPC_QUALITY_UNCERTAIN	0x40	Value undefined
OPC_QUALITY_GOOD	0xC0	OK
OPC_QUALITY_CONFIG_ERROR	0x04	Configuration error
OPC_QUALITY_NOT_CONNECTED	0x08	Not connected
OPC_QUALITY_DEVICE_FAILURE	0x0C	Device failure
OPC_QUALITY_SENSOR_FAILURE	0x10	Sensor failure
OPC_QUALITY_LAST_KNOWN	0x14	Due to a communication error, the last correctly read value is used
OPC_QUALITY_COMM_FAILURE	0x18	Communication error
OPC_QUALITY_OUT_OF_SERVICE	0x1C	Out of service (invalid item)
OPC_QUALITY_WAITING_FOR_INITIAL_DATA	0x20	Waiting for initial data
OPC_QUALITY_LAST_USABLE	0x44	Last usable value
OPC_QUALITY_SENSOR_CAL	0x50	Sensor being calibrated
OPC_QUALITY_EGU_EXCEEDED	0x54	Engineering value conversion result exceeded range
OPC_QUALITY_SUB_NORMAL	0x58	Few data sources
OPC_QUALITY_LOCAL_OVERRIDE	0xD8	Value forcibly overwritten

3.2.7 Error code

When accessing an OPC interface, DeviceXPlorer may return a COM error or an OPC error. These errors are explained in the table below.

Note
If you are using Visual Basic, the occurrence of a COM error or an OPC error is an exception. By trapping the exception, you can check the error code.

[COM Errors]

COM error	Value	Description
E_FAIL	0x80004005	Processing failed (this error is returned when write processing failed due to a communication problem)
E_INVALIDARG	0x80070057	Invalid argument
E_NOINTERFACE	0x80004002	The interface does not exist
E_NOTIMPL	0x80004001	The interface was not implemented
E_OUTOFMEMORY	0x8007000E	Insufficient memory
CONNECT_E_ADVISELIMIT	0x80040201	The maximum number of connection points is already reached, and no more can be accepted (this error is returned when asynchronous writing is continuously implemented while a communication error is occurring)
OLE_E_NOCONNECTION	0x80040004	The communication point does not exist
DV_E_FORMATETC	0x80040064	The format specified in FORMATETC is invalid

[OPC Errors]

OPC error	Value	Description
OPC_E_INVALIDHANDLE	0xC0040001	Invalid handle
OPC_E_BADTYPE	0xC0040004	Cannot convert to the specified data type
OPC_E_PUBLIC	0xC0040005	Cannot implement for public group
OPC_E_BADRIGHTS	0xC0040006	Operation denied due to the item's access rights (attempted to write to a ReadOnly file)
OPC_E_UNKNOWNITEMID	0xC0040007	The item ID does not exist (unregistered tag requested of OPC server)
OPC_E_INVALIDITEMID	0xC0040008	Invalid item ID (item ID not specified)
OPC_E_INVALIDFILTER	0xC0040009	Invalid filter string
OPC_E_UNKNOWNPATH	0xC004000A	Cannot recognize access path
OPC_E_RANGE	0xC004000B	Value out of range
OPC_E_DUPLICATENAME	0xC004000C	Duplicate name
OPC_S_UNSUPPORTEDRATE	0x0004000D	Server does not support the requested rate
OPC_S_CLAMP	0x0004000E	The written value was clamped
OPC_S_INUSE	0x0004000F	Cannot continue operation because the object is in use
OPC_S_UD_COMM_ERROR	0x00048000	Function is succeeded. But quality has problem.
OPC_E_UD_COMM_ERROR	0x80048000	Communication is failure.
OPC_E_INVALIDCONFIGFILE	0xC0040010	Invalid file format set for server
OPC_E_NOTFOUND	0xC0040011	Requested object not found
OPC_E_INVALID_PID	0xC0040203	Invalid property ID
OPC_E_DEADBANDNOTSET	0xC0040400	No dead band set for the item
OPC_E_DEADBANDNOTSUPPORTED	0xC0040401	The item does not support a dead band
OPC_E_NOBUFFERING	0xC0040402	A sampling buffer is not supported
OPC_E_INVALIDCONTINUATIONPOINT	0xC0040403	The continuation point for the Browse interface is invalid
OPC_S_DATAQUEUEOVERFLOW	0xC0040404	The server's sampling buffer overflowed
OPC_E_RATENOTSET	0xC0040405	No sampling rate was set
OPC_E_NOTSUPPORTED	0xC0040406	Writing quality flags and time stamps not supported

3.3 OPC A&E Interface

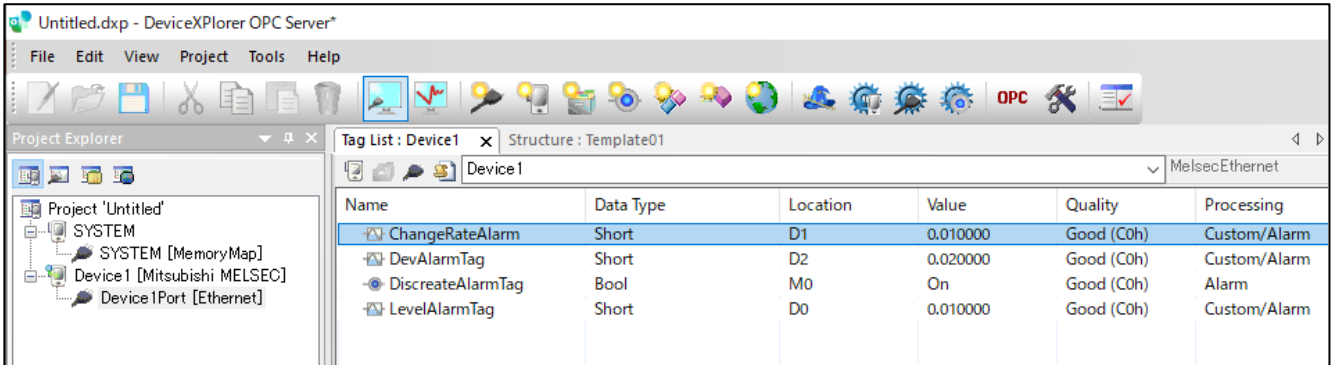
DeviceXPlorer supports OPC A&E1.10.

3.3.1 Prog.ID

The Prog.ID for DeviceXPlorer is “Takebishi.DxpAe.7” or “Takebishi.Dxp.Ae”.

3.3.2 ItemID

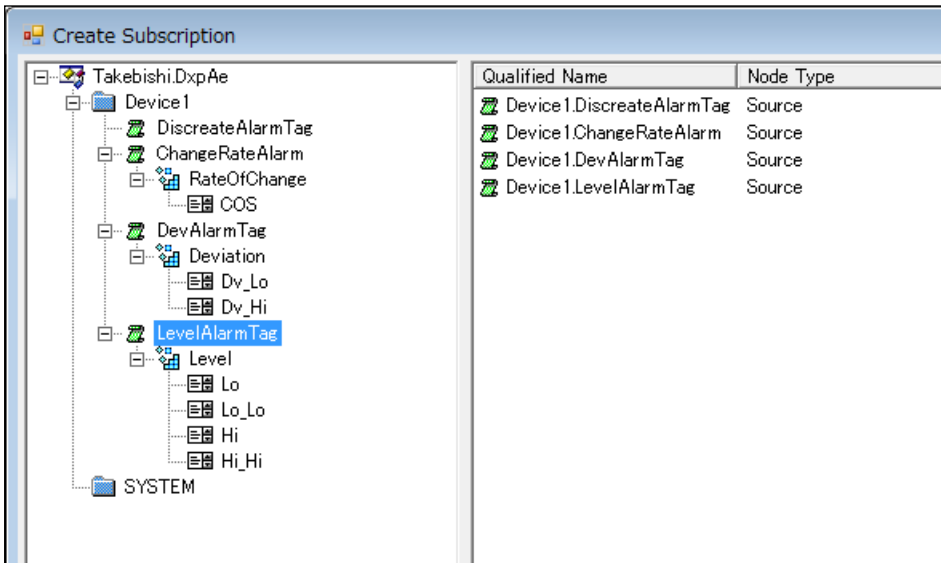
OPC ItemID is the same as that of the view of an OPC DA interface. Please refer to 3.2.2 for details.



The screenshot shows the DeviceXPlorer OPC Server interface. The 'Tag List : Device1' tab is active, displaying a table of tags for 'Device1'. The table has columns for Name, Data Type, Location, Value, Quality, and Processing. The tags listed are ChangeRateAlarm, DevAlarmTag, DiscreateAlarmTag, and LevelAlarmTag.

Name	Data Type	Location	Value	Quality	Processing
ChangeRateAlarm	Short	D1	0.010000	Good (Coh)	Custom/Alarm
DevAlarmTag	Short	D2	0.020000	Good (Coh)	Custom/Alarm
DiscreateAlarmTag	Bool	M0	On	Good (Coh)	Alarm
LevelAlarmTag	Short	D0	0.010000	Good (Coh)	Custom/Alarm

In an OPC A&E client, it looks as follows.



The screenshot shows an OPC A&E client interface. On the left is a tree view of the 'Takebishi.DxpAe' device structure, including 'Device1' and its various tags. On the right is a table with 'Qualified Name' and 'Node Type' columns, listing the tags and their source types.

Qualified Name	Node Type
Device1.DiscreateAlarmTag	Source
Device1.ChangeRateAlarm	Source
Device1.DevAlarmTag	Source
Device1.LevelAlarmTag	Source

Important

In order to treat as an alarm tag, it is necessary to configure an alarm tag property appropriately. Please refer to 4.7.7 for an alarm configuration.

3.3.3 Custom Interfaces

The support statuses of OPC AE custom interfaces are as shown below.

Object	Interface name	Opt	Member function	Support status
OPCEventServer Object	IOPCCCommon		SetLocaleID	OK
			GetLocaleID	OK
			QueryAvailableLocalIDs	OK
			GetErrorString	OK
			SetClientName	OK
	IOPCEventServer		GetStatus	OK
			CreateEventSubscription	OK
			QueryAvailableFilters	OK
			QueryEventCategories	OK
			QueryConditionNames	OK
			QuerySubConditionNames	OK
			QuerySourceConditions	OK
			QueryEventAttributes	OK
			TranslateToItemIDs	OK
			GetConditionState	OK
			EnableConditionByArea	OK
			EnableConditionBySource	OK
			DisableConditionByArea	OK
			DisableConditionBySource	OK
			AckCondition	OK
			CreateAreaBrowser	OK
	IOPCEventServer2	○	EnableConditionByArea2	N/A*1
			EnableConditionBySource2	N/A*1
			DisableConditionByArea2	N/A*1
			DisableConditionBySource2	N/A*1
			GetEnableStateByArea	N/A*1
			GetEnableStateBySource	N/A*1
	IConnectionPointContainer		EnumConnectionPoints	OK
			FindConnectionPoint	OK
OPCEventAreaBrowser Object	IOPCEventAreaBrowser	○	ChangeBrowsePosition	OK
			BrowseOPCAreas	OK
			GetQualifiedAreaName	OK
			GetQualifiedSourceName	OK
OPCEventSubscription Object	IOPCEventSubscriptionMgt		SetFilter	OK
			GetFilter	OK
			SelectReturnedAttributes	OK
			GetReturnedAttributes	OK
			Refresh	OK
			CancelRefresh	OK
			GetState	OK
			SetState	OK
	IConnectionPointContainer		EnumConnectionPoints	OK
			FindConnectionPoint	OK

*1 E_NOTIMP will be returned.

3.4 OPC UA Interface

DeviceXPlorer supports OPC UA.

3.4.1 Specifications Table

The specification table of OPC UA is shown below. Endpoint URL, security policy, and security mode settings It can be changed on the common property(Tool menu "Option").

Version	1.04
Type	Server
Endpoint URL	opc.tcp://[IP address or machine name]:52250
Profile	Standard UA Server Profile Historical Raw Data Server Facet Historical Data Insert Server Facet Historical Data Update Server Facet Historical Data Replace Server Facet A&C Base Condition Server Facet A&C Refresh2 Server Facet A&C Address Space Instance Server Facet A&C Enable Server Facet A&C Acknowledgeable Alarm Server Facet A&C Alarm Server Facet A&C Exclusive Alarm Server Facet A&E Wrapper Facet Reverse Connect Client Facet
Security Policy	None, Basic128Rsa15, Basic256, Basic256Sha256, Aes128Sha256RsaOaep, Aes256Sha256RsaPss
Security Mode	None, Sign, SignAndEncrypt
User Authentication	Anonymous, User Name/Password, X509 User Certificate
Product URI	urn:Takebishi:DeviceXPlorerOPCServer
Encoding	Binary
Maximum Session Count	100
Maximum Session Count /Client	100
Maximum Subscription Count / Session	Unlimited
MonitoredItem / Subscription	Unlimited
Minimum Sampling Interval	50ms
Minimum Publish Interval	100ms
IndexRange	Not Support
OPC UA Pub/Sub	-
Transport Profile	http://opcfoundation.org/UA-Profile/Transport/pubsub-udp-uadp
Minimum Publish Interval	50ms
Maximum WriterGroup Count	8

*User authentication settings can be set from Use Manager element. * Please contact us if you wish to use OPC UA Pub

lisher.

3.4.2 Service

The support statuses of OPC UA Service are as shown below.

Service Set	Service	Function (Our classification)	Support status
Discovery Service Set	FindServers	Enum	×
	GetEndpoints	Core	○
	RegisterServer	Enum	×
Secure Channel Service Set	OpenSecureChannel	Core	○
	CloseSecureChannel	Core	○
Session Service Set	CreateSession	Core	○
	ActivateSession	Core	○
	CloseSession	Core	○
	Cancel	Core	○
NodeManagement Service Set	AddNodes	NodeManager	×
	AddReferences	NodeManager	×
	DeleteNodes	NodeManager	×
	DeleteReferences	NodeManager	×
View Service Set	Browse	Browse	○
	BrowseNext	Browse	○
	TranslateBrowsePathToNode Ids	Browse	○
	RegisterNodes	View	×
	UnregisterNodes	View	×
Query Service Set	QueryFirst	Query	×
	QueryNext	Query	×
Attribute Service Set	Read	DA	○
	HistoryRead	Historical	○
	Write	DA	○
	HistoryUpdate	Historical	○
Method Service Set	Call	Method	○
MonitoredItem Service Set	CreateMonitoredItems	DA	○
	ModifyMonitoredItems	DA	○
	SetMonitoringMode	DA	○
	SetTriggering	DA	×
	DeleteMonitoredItems	DA	○
Subscription Service Set	CreateSubscription	DA/A&C	○
	ModifySubscription	DA/A&C	○
	SetPublishingMode	DA/A&C	○
	Publish	DA/A&C	○
	Republish	DA/A&C	○
	TransferSubscriptions	DA/A&C	×
	DeleteSubscriptions	DA/A&C	○

3.5 SuiteLink Interfaces

DeviceXPlorer supports SuiteLink interface of wonderware.

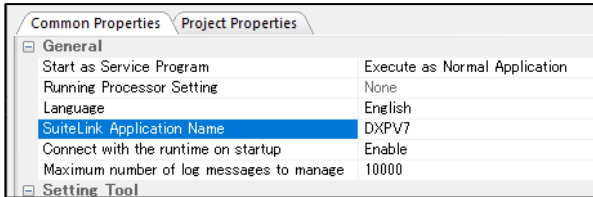
Important

- Tags of LONGLONG and ULONGLONG are not supported for SuiteLink interface.
- To use SuiteLink interface, need to set up the Wonderware component.

3.5.1 Application Name

The application name for DeviceXPlorer is “DXPV7”.

The application name can be changed on the Common Properties screen (Tools -> Options)



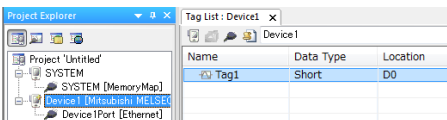
Item	Description
SuiteLink Application Name	Specify SuiteLink Server name. Default is “DXPV7”.
Notify Cache first	Send cache value of tag at first notification immediately when SuiteLink clients connect the server. You can avoid SuiteLink timeout on SuiteLink client.

3.5.2 Topic Names and Item Names

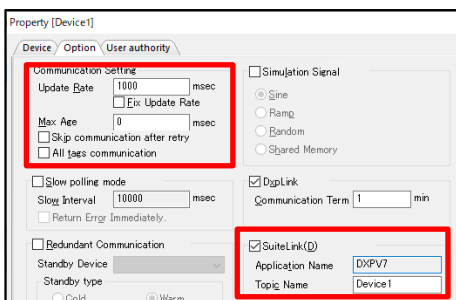
DeviceXPlorer allows you to set a topic name for each device.

The default topic name is “Device name” and the default item name is “Tag name”.

To access “Tag1” in the figure below, specify “Device1” as the topic name and “Tag1” as the item name.



You can specify a topic name and update interval for each device on the “Device Options Screen”.



Item	Description
SuiteLink	Place a check in this box to enable the SuiteLink interface.
Application Name	Shows the currently set SuiteLink application name. This property can be changed only in Common Properties (Tools -> Options).
Topic Name	Allows you to set a topic name of your choice for each device.
Update Rate	Allows you to set the interval for communication with the PLC.

3.6 DxpLink Interface

DeviceXPlorer corresponds to the DxpLink interface (HTTP communication) which is an original interface of our company.

3.6.1 DxpLink Server Settings

DxpLink Server function can be changed on Common Properties screen (menu Tools -> Options).



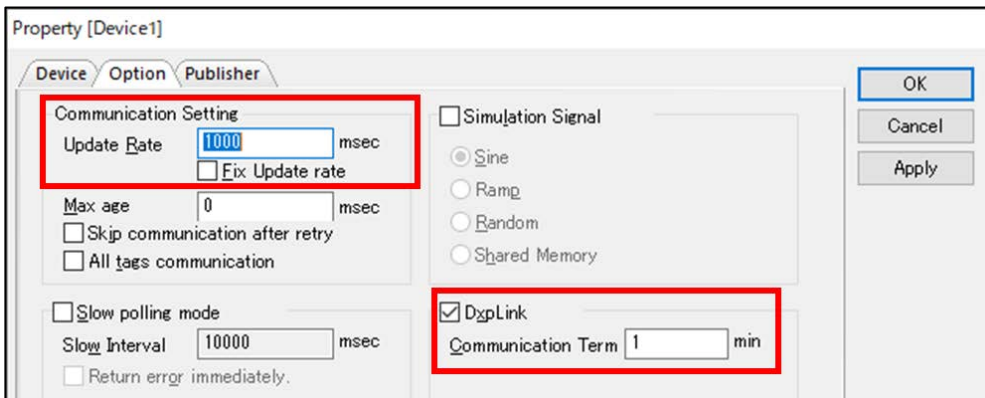
IoT Interface

Port No ☒ Enabled

Item	Description
DxpLink Server Function	Place a check in this box to enable the DxpLink Server function.
Port No	Specify the port No of DxpLink Serer.

3.6.2 DxpLink Communication Settings

DxpLink communicates per Device. Enable / Disable of DxpLink function can be specified on "Device Options".



Property [Device1]

Device Option Publisher

Communication Setting

Update Rate msec ☐ Fix Update rate

Max age msec

☐ Skip communication after retry

☐ All tags communication

☐ Slow polling mode

Slow Interval msec

☐ Return error immediately.

☐ Simulation Signal

☒ Sine

☐ Ramp

☐ Random

☐ Shared Memory

☒ DxpLink

Communication Term min

OK Cancel Apply

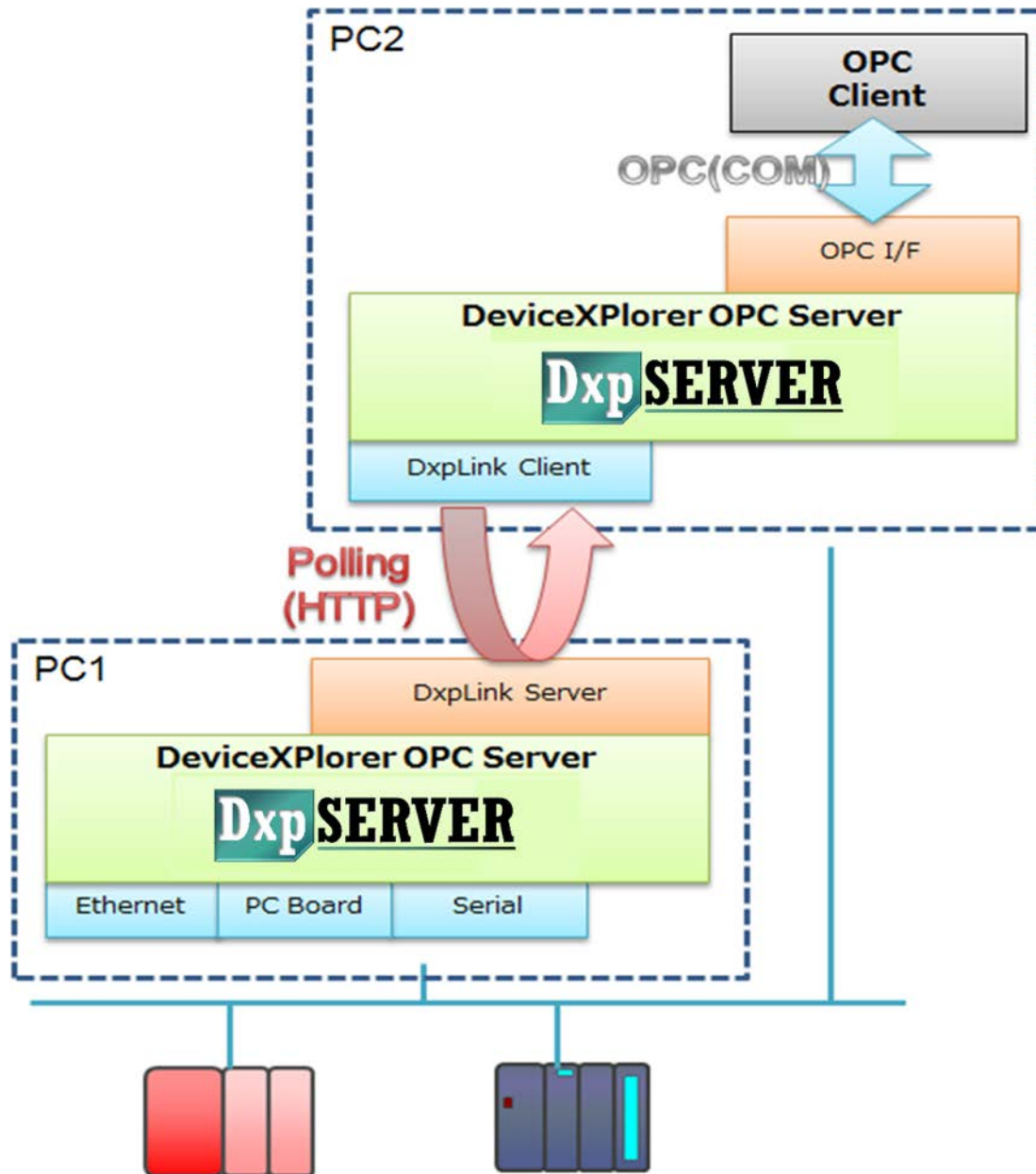
Item	Description
Server Function	Place a check in this box to enable the DxpLink Interface.
Communication Term	Specify the communication term(min) which is communication time after DxpLink client requests.
Update Rate	Allows you to set the interval for communication with the PLC.

3.6.3 DxpLink Communication

DeviceXPlorer has DxpLink Server function and DxpLink Client function.

DxpLink is HTTP communications protocol of our original.

The environment where DCOM connection is difficult, when the communication interface by the side of PLC is restricted, PLC data can be seamlessly accessed by using a DxpLink function.



3.6.4 DxpLink Protocol Specification

The DxpLink communication is realized by exchanging XML documents by POST command of HTTP from the client.

Contents	Contents
HTTP Version	1.1
Maximum Clients	64
Request Method	POST
Message Body	XML Document (XML1.0)

The XML document for request from DxpLink client and response from DeviceXPlorer is based on the following commands.

3.6.4.1 Enumerate Device (GetDevice)

■ Request

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Transportation Command="GetDevice"/>
```

Name	Kind	Type	Contents
Transportation			
Command	Attribute	String	Fixed "GetDevice"

■ Response

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Transportation Command="GetDevice" ErrorCode="0">
  <Delimiter Extend="." Namespace=""/>
  <Device Name="Device1" Type="MelsecEthernet"/>
  <Device Name="Device2" Type="LinkClient"/>
  <Device Name="Device3" Type="MicrexEthernet"/>
  <Device Name="SYSTEM" Type="SystemDevice"/>
</Transportation>
```

Name	Kind	Type	Contents
Transportation			
Command	Attribute	String	Fixed "GetDevice"
ErrorCode	Attribute	32bit Unsinged Int	
Delimiter	Element	Delimiter	Refer to type of "Delimiter"
Device	Element	Device	Refer to type of "Device" Enumerated the number of tags in the target path.

Name	Kind	Type	Contents
Delimiter			
Extend	Attribute	Character	Extend Item Name Delimiter
Namespace	Attribute	Character	Namespace Delimiter

Name	Kind	Type	Contents
Device			
Name	Attribute	String	Device Name
Type	Attribute	String	The kind of Device (Component name)

3.6.4.2 Enumerate Tag (GetTags)

■ Request

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Transportation Command="GetTags">
  <TargetPath Name="Device1"/>
</Transportation>
```

Name	Kind	Type	Contents
Transportation			
Command	Attribute	String	Fixed "GetTags"
TargetPath	Element	TargetPath	Refer to type of "TargetPath"

Name	Kind	Type	Contents
TargetPath			
Name	Attribute	String	The target path of device of group to enumerate tag. eg "Device1Group1"

■ Response

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Transportation Command="GetTags" ErrorCode="0">
  <Delimiter Extend="." Namespace="."/>
  <TargetPath Name="Device1">
    <Tag Name="Tag1" Size="1" Type="SHORT"/>
    <Tag Name="Tag2" Size="1" Type="STRING"/>
    <Tag Name="Tag3" Size="6" Type="SHORT | ARRAY"/>
  </TargetPath>
</Transportation>
```

Name	Kind	Type	Contents
Transportation			
Command	Attribute	String	Fixed "GetTags"
ErrorCode	Attribute	32bit Unsinged Int	
Delimiter	Element	Delimiter	Refer to type of "Delimiter"
TargetPath	Element	TargetPath	Enumerated the number of tags in the target path.

Name	Kind	Type	Contents
Delimiter			
Extend	Attribute	Character	Extend Item Name Delimiter
Namespace	Attribute	Character	Namespace Delimiter

Name	Kind	Type	Contents
TargetPath			
Name	Attribute	String	Device Name
Tag	Element	Tag	Refer to type of "Tag" Enumerated tags in the target path.

Name	Kind	Type	Contents
Tag			
Name	Attribute	String	Tag Name
Size	Attribute	32bit Unsinged Int	The number of element of array. "1" is returned when the tag is not array.
Type	Attribute	String	The datatype string of tag. Refer to "The datatype string of tag."

3.6.4.3 Read Value of Tag (GetTagsValue)

■ Request

Example

```
<?xml version="1.0" encoding="utf-8"?>
<Transportation Command="GetTagValues">
  <TargetPath Name="Device1">
    <Tag Name="Short" />
    <Tag Name="String" />
    <Tag Name="ShortArray" />
  </TargetPath>
</Transportation>
```

Name	Kind	Type	Contents
Transportation			
Command	Attribute	String	Fixed "GetTagValues"
TargetPath	Element	TargetPath	Refer to type of "TargetPath"

Name	Kind	Type	Contents
TargetPath			
Name	Attribute	String	The target path of device of group to enumerate tag. eg "Device1Group1"
Tag	Element	Tag	Refer to type of "Tag" Specify tags you want to read.

■ Response

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Transportation Command="GetTagValues" ErrorCode="0">
  <TargetPath Name="Device1">
    <Tag Name="Short" Size="1" Type="SHORT">
      <Value>0</Value>
      <TimeStamp>2012/09/27 06:10:24 GMT</TimeStamp>
      <Quality>192</Quality>
    </Tag>
    <Tag Name="String" Size="1" Type="STRING">
      <Value>testvalue</Value>
      <TimeStamp>2012/09/27 06:10:24 GMT</TimeStamp>
      <Quality>192</Quality>
    </Tag>
    <Tag Name="ShortArray" Size="10" Type="SHORT | ARRAY">
      <Value>0,0,0,0,0,0,0,0,0,0</Value>
      <TimeStamp>2012/09/27 06:10:24 GMT</TimeStamp>
      <Quality>192</Quality>
    </Tag>
  </TargetPath>
</Transportation>
```

Name	Kind	Type	Contents
Transportation			
Command	Attribute	String	Fixed "GetTagValues"
ErrorCode	Attribute	32bit Unsinged Int	
TargetPath	Element	TargetPath	Refer to type of "TargetPath"

Name	Kind	Type	Contents
TargetPath			
Name	Attribute	String	Device Name
Tag	Element	Tag	Refer to type of "Tag" The tags you specified.

Name	Kind	Type	Contents
Tag			
Name	Attribute	String	Tag name
Size	Attribute	32bit Unsinged Int	The number of element of array. "1" is returned when the tag is not array.
Type	Attribute	String	The datatype string of tag. Refer to "The datatype string of tag."
Value	Element	String	The value which is based on the datatype of tag.
TimeStamp	Element	String	
Quality	Element	16bit Unsinged Int	

3.6.4.4 Write Value of Tag (SetTagsValue)

Request

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Transportation Command="SetTagValues">
  <TargetPath Name="Device1">
    <Tag Name="Tag">
      <Value>123</Value>
    </Tag>
  </TargetPath>
</Transportation>
```

Name	Kind	Type	Contents
Transportation			
Command	Attribute	String	Fixed "SetTagValues"
TargetPath	Element	TargetPath	Refer to type of "TargetPath"

Name	Kind	Type	Contents
TargetPath			
Name	Attribute	String	The target path of device of group to enumerate tag. eg "Device1Group1"
Tag	Element	Tag	Refer to type of "Tag" Specify tags you want to write value.

Name	Kind	Type	Contents
Tag			
Name	Attribute	String	Tag Name
Value	Element	String	Specify the value which is based on datatype of tag.

Response

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Transportation Command="SetTagValues" ErrorCode="0">
  <TargetPath Name="Device1"/>
</Transportation>
```

Name	Kind	Type	Contents
Transportation			
Command	Attribute	String	Fixed "SetTagValues"
ErrorCode	Attribute	32bit Unsinged Int	
TargetPath	Element	TargetPath	Refer to type of "TargetPath"

Name	Kind	Type	Contents
TargetPath			
Name	Attribute	String	Device Name

3.6.4.5 Constants

■ The Datatype String of Tag

Datatype of tag	Array	The Datatype String of tag for DxpLink
BOOL		"BOOL"
BYTE		"BYTE"
UBYTE		"UBYTE"
SHORT		"SHORT"
USHORT		"USHORT"
LONG		"LONG"
ULONG		"ULONG"
LONGLONG		"LONGLONG"
ULONGLONG		"ULONGLONG"
FLOAT		"FLOAT"
DOUBLE		"DOUBLE"
STRING		"STRING"
BOOL	●	"BOOL ARRAY"
BYTE	●	"BYTE ARRAY"
UBYTE	●	"UBYTE ARRAY"
SHORT	●	"SHORT ARRAY"
USHORT	●	"USHORT ARRAY"
LONG	●	"LONG ARRAY"
ULONG	●	"ULONG ARRAY"
LONGLONG	●	"LONGLONG ARRAY"
ULONGLONG	●	"ULONGLONG ARRAY"
FLOAT	●	"FLOAT ARRAY"
DOUBLE	●	"DOUBLE ARRAY"
STRING	●	"STRING ARRAY"

3.7 Modbus/TCP Server Interface

For details, see the document “DeviceXPlorer OPC Server User’s Guide (PLC Communication Edition)” (DeviceXPlorer_Plc.pdf).

3.8 HTTP Client Interface

Version	1.1
Function Type	Client
HTTP Authentication	None, basic authentication, digest authentication OAuth2.0 authorization(only ROPC)
HTTP Method	GET, PUT, POST, DELETE
	OPTIONS, HEAD, TRACE
Encryption	SSL/TLS
Proxy Server	Supported (username/password authentication also supported)

3.9 MQTT Client Interface

Version	3.11, 3.1, Auto
Function Type	Client (publisher / subscriber)
Protocol	TCP/IP (WebSockets not supported)
Authentication	<MQTT broker> Username/password <Azure IoT Hub> Symmetric key in the identity registry Shared access policy, X509 certificates
Encryption	SSL/TLS
QoS	QoS0, QoS1, QoS2
Retain/Will	Supported

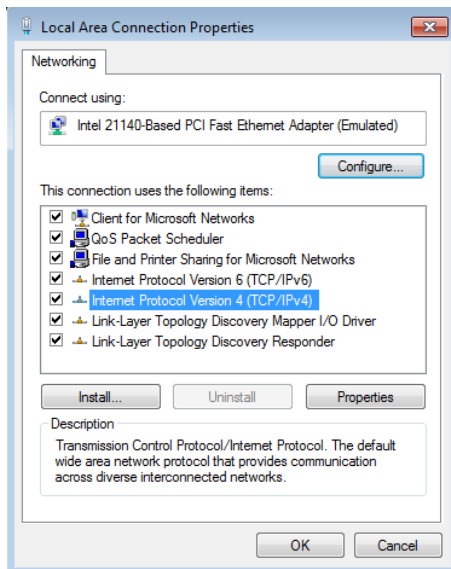
4 Operation

4.1 Tutorial

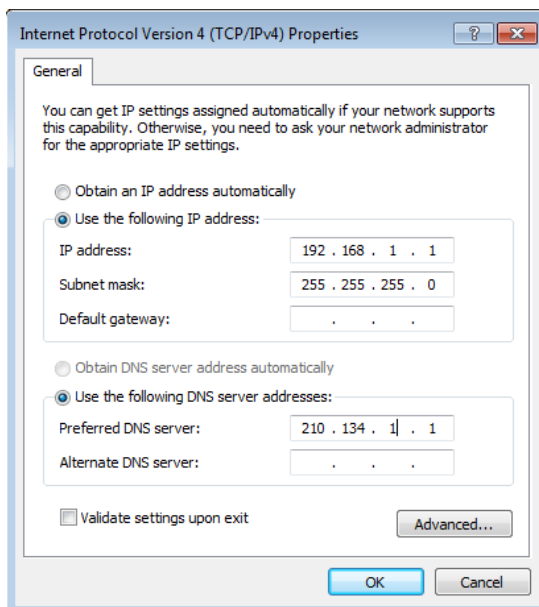
If you are using DeviceXplorer for the first time, follow the steps below to make the necessary settings and easily check operation.

4.1.1 Installing and Setting Up an Ethernet Adapter

Install an Ethernet adapter to the local PC and open Control Panel -> Local Area Connection -> Properties.



Open the “Internet Protocol (TCP / IP)” properties screen, set the IP address and host name, etc., and click “OK”.



4.1.2 Installing DeviceXPlorer

Insert the installation disk into the CD drive of the PC and install DeviceXPlorer.

Note

Install DeviceXPlorer using an account with Administrator privileges.

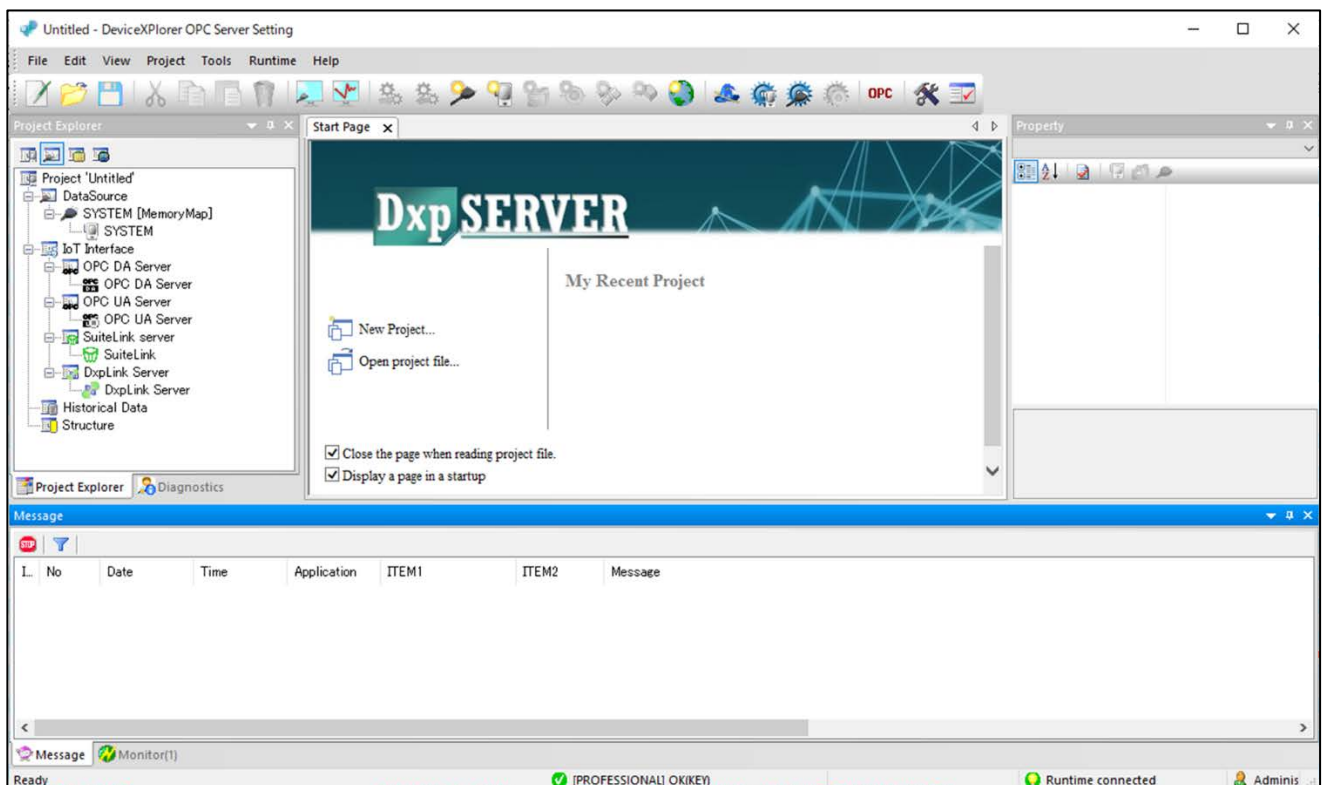
4.1.3 Starting DeviceXPlorer

From the Programs menu, start "DeviceXPlorer OPC Server 7"->"DeviceXPlorer OPC Server 7".

4.1.4 Start Page

Starting of device Explorer will display a start page as follows.

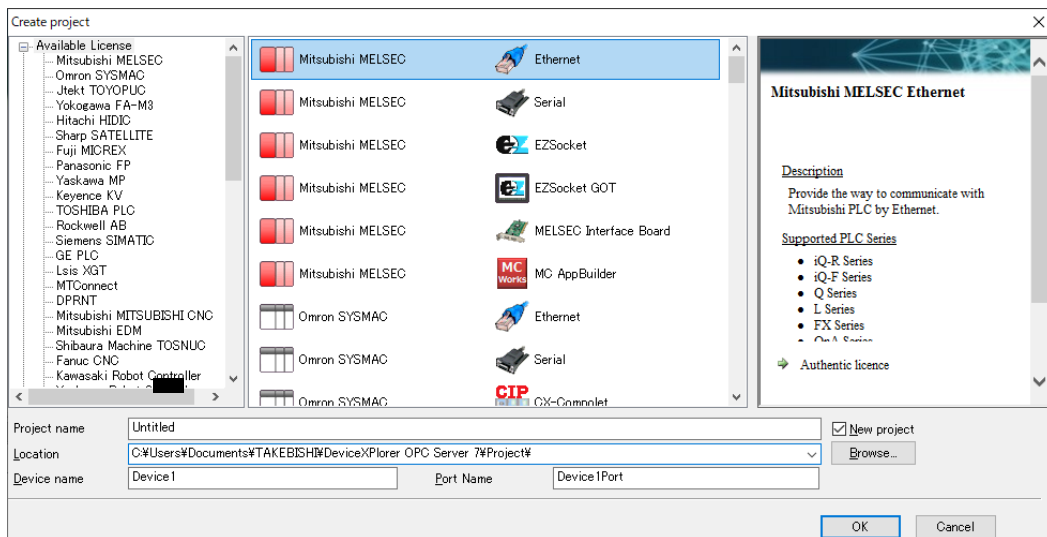
Please push "New Project" and create a new project.



Item	Description
New Project	Creates a new project.
Open project file	Opens a project.
My Recent Project	Shown recently used project file.
Close the page when Reading project file.	Select whether close Start Page when DeviceXPlorer load project file.
Display a page in a startup	Select whether Start Page is displayed when DeviceXPlorer start.

4.1.5 Project Wizard

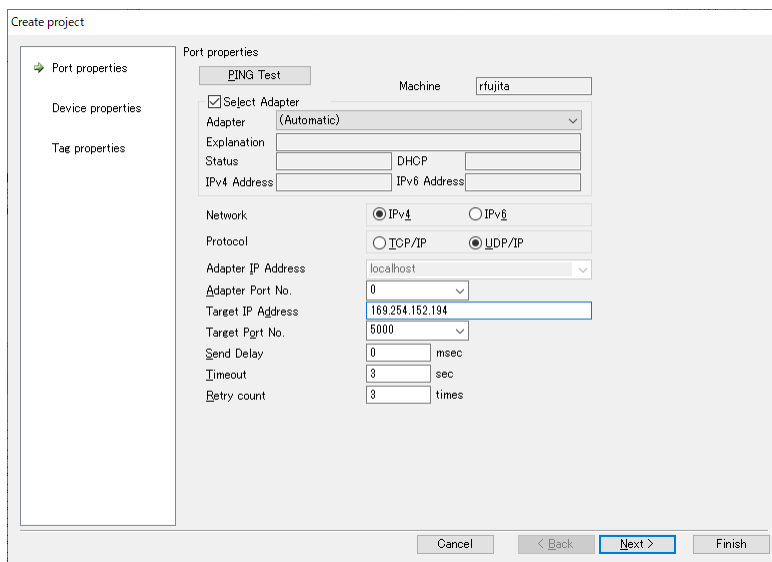
- (1) Since a project wizard will be displayed if a new project is created, select the target for communication, and input a Device name and Port name, and click the "OK."



Item	Description
Effective license	The components in which the license is effective are enumerated.
Installed Component	The components which are installed are enumerated.
Project name	Specify project name.
Location	Specify project file path.
Device name	Input Device name for creating Device.
Port name	Input Port name for creating Port.

(2) Specify the port information. You can confirm the existence of the IP address you entered by clicking the “PING” button.

For details, refer to 4.3.



- (3) Specify the Device information. You can check the communication with the PLC by clicking the “Connection test” button .
For details, refer to 4.4.

Create project

Port properties

Device properties

Tag properties

Connection Test

CPU TYPE

☐ IQ-B ☐ IQ-E ☒ Q ☐ L ☐ QnA ☐ A ☐ FX

☐ Use G4E Protocol

☐ Use Block Access Command

NETWORK

Network No. 0

PC No. 255

Unit Station No. 0

Unit I/O No.

Multiple CPU System

☒ None ☐ 1 ☐ 2 ☐ 3 ☐ 4

Redundancy CPU System

☒ None ☐ Control ☐ Standby

Communication

CPU monitoring timer 10 sec

Remote Password

ACCESS POINTS

Bit R/W 960 Bit Poke 188

Word R/W 960 Word Poke 160

Cancel < Back Next > Finish

- (4) Specify the Tag information. For details, refer to 4.7.

Create project

Port properties

Device properties

Tag properties

Tag Name Tag D0

☐ Auto Format

Description

Device Type D

Data Register D

Device No. 0

Block No. 0

Data Type SHORT

Byte Position Lower Byte

Bit Position 0

Length 1

☐ Byte Swap

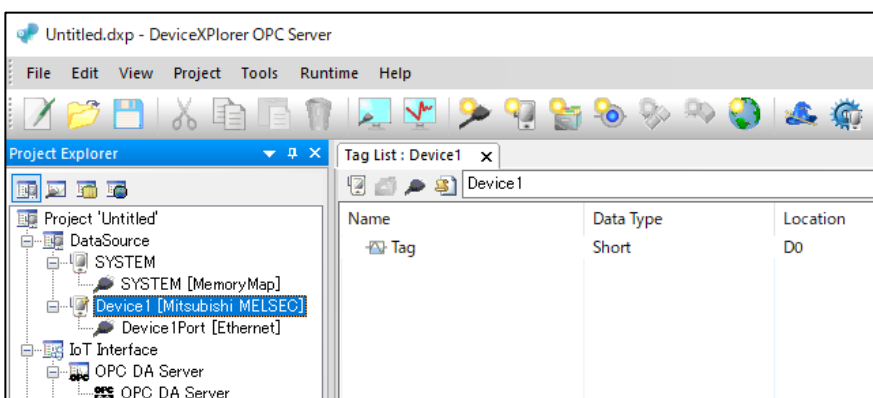
☐ Array

Attribute ReadWrite

Calculation NONE

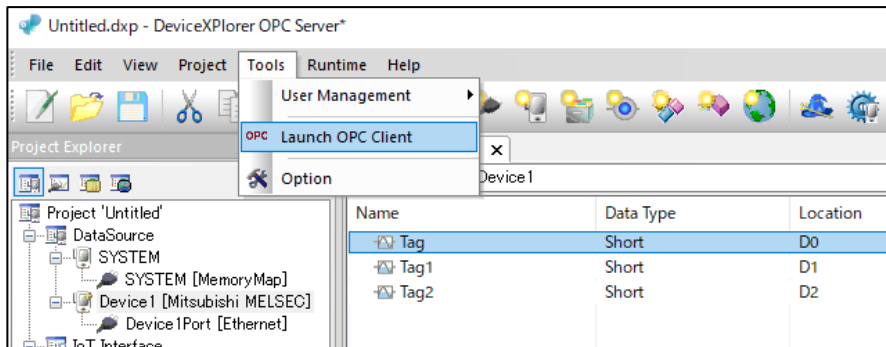
Cancel < Back Next > Finish

- (5) Above, Port settings and Device settings indispensable for communication with partner apparatus, and Tag settings were completed.
The rest should perform addition of a tag and various option setup according to a system.

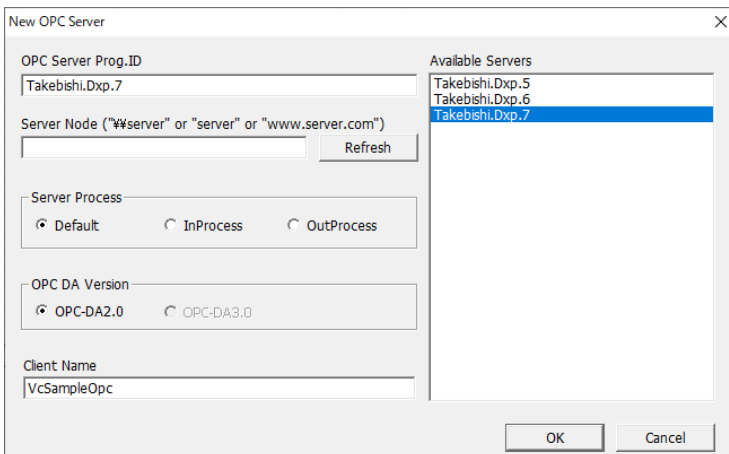


4.1.6 Accessing DeviceXPlorer from a Client

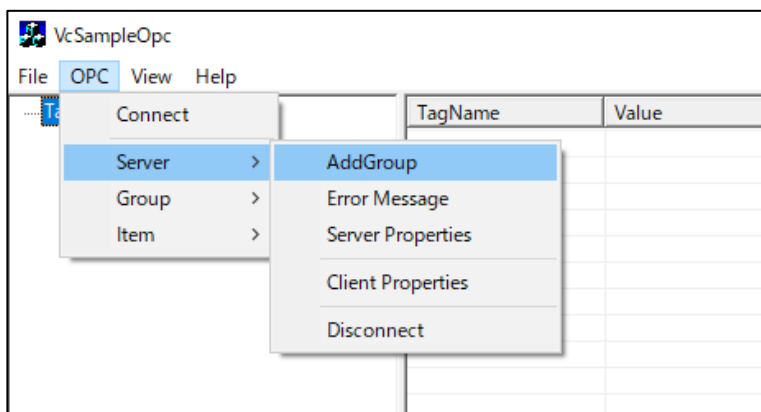
From the Tools menu, select “Start OPC Test Client”.



In the OPC Test Client menu, select “Connect” and connect to “Takebishi.Dxp.7”.



Next, select “Server” and “AddGroup” in “OPC” menu, and add group.



Add Group

OPC Group Name: Group0

Active: ☒

Update Interval: 1000 ms

Deadband: 0 % FS

Time Bias: 0 min

Locale ID: 0

OK Cancel

Next, select "Group" and "AddItem" in "OPC" menu, and add tags.

VcSampleOpc

File OPC View Help

Connect

Server >

Group > AddItem

Item >

TagName Value

AsyncIO2 Refresh (Cache)

AsyncIO2 Refresh (Device)

AsyncIO2 Cancel

Active

Advise

UnAdvise

Group Properties

RemoveGroup

Add Item

Access Path

Browse Items

Server Tags

Device1.Tag;Device1.Tag1;Device1.Tag2

Add >>

Added Items List

Device1.Tag

Device1.Tag1

Device1.Tag2

OK Cancel

Check that the client successfully obtained data. (If "0xC0" appears under Quality, communication is functioning normally.)

VcSampleOpc

File OPC View Help

Takebishi.Dxp.7

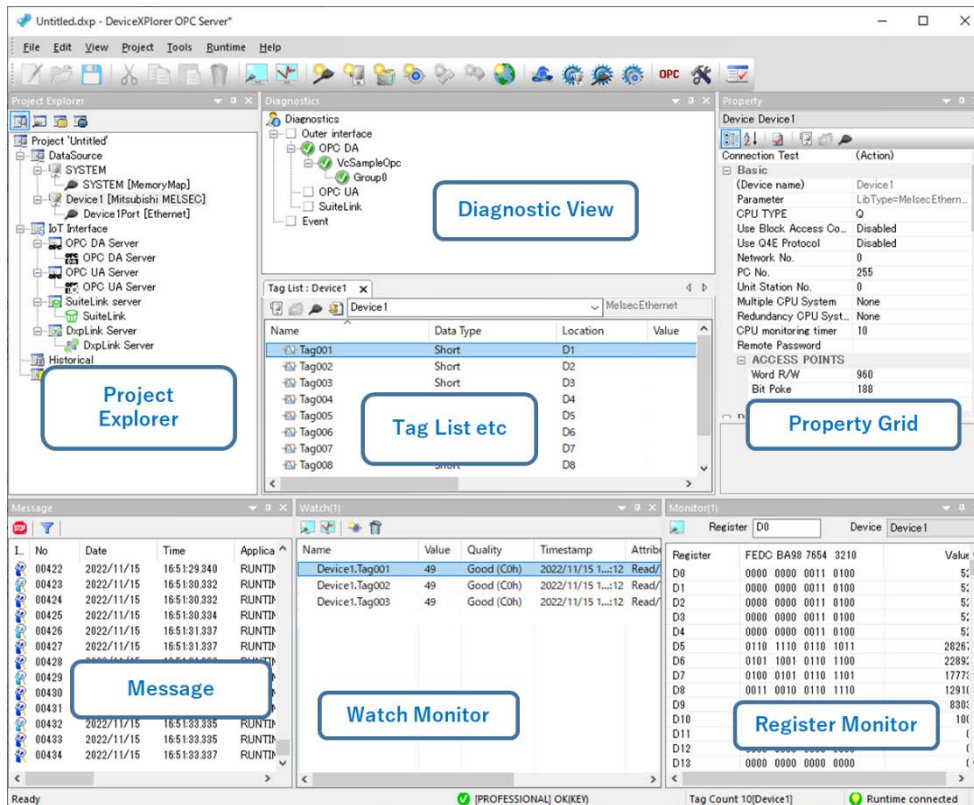
Group0

TagName	Value	Quality	TimeStamp
Device1.Tag	78	0xC0	11/08/2022 15:33:20.921
Device1.Tag1	78	0xC0	11/08/2022 15:33:20.921
Device1.Tag2	78	0xC0	11/08/2022 15:33:20.921

Date	Time	Message
2022/11/08	15:30:42.782	Connected to [Takebishi.Dxp.7] / hr= [0]
2022/11/08	15:31:23.945	IOPCServer::AddGroup Called / hr= [0]
2022/11/08	15:32:14.742	IOPCItemMgt::AddItem Called / hr= [0]
2022/11/08	15:32:14.742	Success
2022/11/08	15:32:18.274	IOPCSyncIO::Read Called / hr= [1]

Ready NUM

4.2 Screen Layout

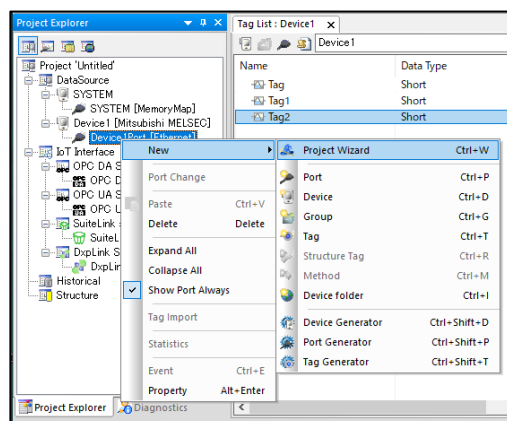


[Project Explorer]

Project Explorer allows you to visually monitor the setting status of ports, devices, and groups, and allows you to create or edit them. Using the toolbar, you can also sort by device, port, or protocol (PLC).

Icon	Name	Description
	Device View	A tree with a device as a root in a hierarchical structure.
	Port View	A tree with port information as a root in a hierarchical structure.
	Protocol View	A tree with PLC information as a root in a hierarchical structure.
	Device Folder View	A tree with Device Folder as a root in a hierarchical structure.

Right-click Project Explorer to view the pop-up menu of related functions.



[Diagnostic View]

Diagnostic View allows you to check information such as the access status of each interface and the current value of each tag.

The screenshot shows the 'Diagnostics' window. On the left is a tree view with the following structure:

- Diagnostics
 - Outer interface
 - OPC DA (selected)
 - VcSampleOpc
 - Group0
 - OPC UA
 - urn:Z1710-02:UnifiedAu
 - MonitoredItems
 - Device1
 - SuiteLink
 - Device1
 - Event
 - Device1

On the right is a table titled 'Diagnostics List' with the following data:

Client Name	Groups	Items	Active Items	Error Items
✓ VcSampleOpc	1	10	10	0

[Tag List etc.]

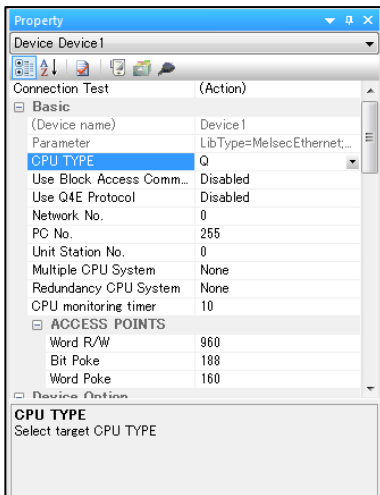
The property information of Tag List, Structure Template Tag List, Script List, Sent tag list for Iot system, target Target tag list of historical group ,etc. are shown.

The screenshot shows several overlapping windows:

- Tag List : Device1**: Shows a list of tags (Tag001 to Tag005).
- Structure : Template02**: Shows a list of members (Member01 to Member04).
- Tag List : SYSTEM**: Shows a list of scripts (Script1 to Script5).
- Event List : SYSTEM**: Shows a table with columns: Event, Type, Condition, Run.
- Tag List : SYSTEM**: Shows a table with columns: No, Name, Comment, User defined, UserDefine.
- IoT Interface : Mqtt01.MqttClientGroup01**: Shows a table with columns: Name, Attribute.
- Historical group : HistoricalGroup1**: Shows a table with columns: Name, Attribute.

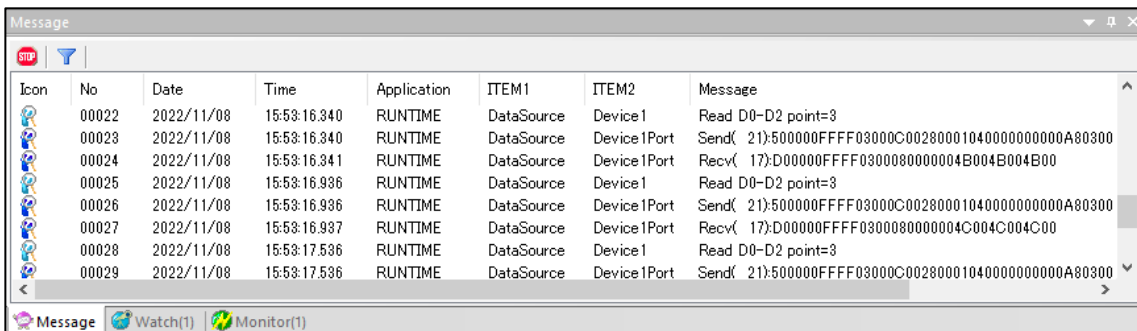
[Property Grid]

This grid shows information of selecting Object. And you can configure object.



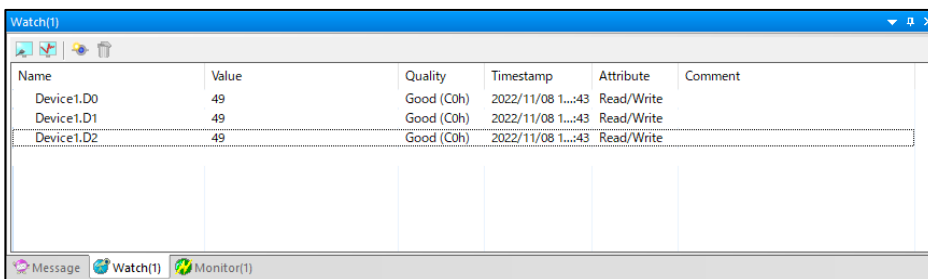
[Message View/Watch View]

This shows messages such as communication error information and diagnostic information.



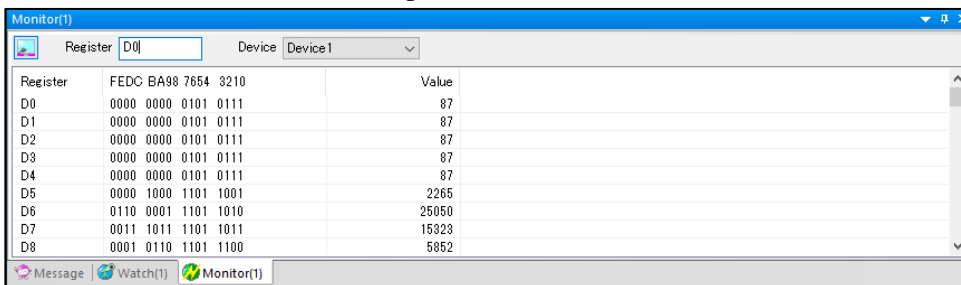
[Watch Monitor]

Arbitrary tags can be registered and you can monitor the current value.



[Register Monitor]

Even if there is no ladder tool, the register value on PLC can be monitored.



4.2.1 Menus

The menus available in DeviceXPlorer are shown in the table below.

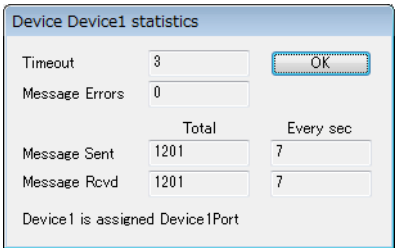
[File Menu]

Item		Description
New		Creates a new project.
Open		Opens a project.
Close		Closes a project.
Save		Overwrites and saves the project file.
Save as		Saves the project under a new name.
Import	Definition	Imports project information.
	TagMemory	Imports Tag memory and DeviceXPlorer shifts to Offline Mode.
Export	Definition	Exports project information in the XML or CSV format.
	TagMemory	Exports Tag memory(tag name, value, quality, timestamp).
The recently project list		Shown recently used project file.
Exit		Exists DeviceXPlorer.

[Edit Menu]

Item		Description
Cut		Cuts the selected tag(s).
Copy		Copies the selected tag(s).
Paste		Pastes tag information.
Delete		Deletes the selected device, group or tag.
Select all		Selects all the tags included in the selected devices.
Find		Searches for a specified tag name.
Array Viewer		Show array viewer.
Write Value to Tag		Write current value to tag.
Properties		Shows the Properties screen for the selected item.

[View Menu]

Item		Description
Monitor	Device monitor	Reads the tag value and displays it to allow you to check the communication status and value. Communication is performed only for the device shown in the window.
	Cache monitor	Shows the value of the tag being accessed by the client. Values are not shown for tags not being accessed.
Offline		Shifts to Offline Mode.
Communication information		Shows the communication status and number of errors for each device. 
Message filters	Type Filter	Specify the items to be displayed in the message and the log level.
	Text Filter	Filter the message display.
Pause Message Display		Temporarily pauses the output of trace information.
Window	Project Explorer	Shows Project Explorer.
	Message View	Shows Message View.
	Property View	Shows Property View
	Diagnostics View	Shows Diagnostics View.
	Watch	Shows Watch Monitor.(Max 8 view)
	Monitor	Shows Register Monitor.(Max 8 view)
	Start Page	Shows Start Page
Toolbar		Switches the toolbar display on and off.
Status bar		Switches the status bar display on and off.
Value Format	Binary	Specifies tag value format for tag monitor and tag diagnostics.
	Octal	
	Decimal	
	Hexadecimal	
Update Views		Updates the displayed content.

[Project Menu]

Item		Description
Project wizard		Starts the project wizard to generate ports, devices, and tags all at once.
Insert	Port	Creates new communication port information.
	Device	Creates new device information.
	Group	Creates new group information.
	Tag	Creates new tag information.
	Structure Tag	Creates new Structure Tag information.
	Method	Crates new Method.
	Device folder	Create new Device folder.
	Device Generator	Starts Device Generator.
	Port Generator	Starts Port Generator.
	Tag Generator	Starts Tag Generator.
Properties	Port (P)	Sets communication port information.
	Device (D)	Sets the device information.
	Group (G)	Sets group information.
	Tag (T)	Sets tag information.
Set as startup project		Sets the current project file as a startup project.
Project1 properties		Opens the Properties screen (configuration information) for the project.

[Tools Menu]

Item		Description
User Management	Configuration	Sets the User information.
	Login	Logins
	Logout	Logouts
Start OPC test client		Starts the OPC test client.
Options		Opens the Properties screen (shared information) for the project.

[Runtime]

Item		Description
Connect	Setting Tool -> Runtime	Connect to runtime and synchronizes configuration base on Setting Tool.
	Runtime -> Setting Tool	Connect to runtime and synchronizes configuration base on Runtime.
Disconnect		Disconnect the Runtime.
Reboot		Restart the Runtime.
Stop		Stops the Runtime.

[Help Menu]

Item		Description
Manuals	User's Guide (Server Edition)	Shows Server Edition.
	User's Guide (PLC Communication Edition)	Shows PLC Communication Edition.
Online Registration		Performs user registration for the product.
Send mail to support		Starts the mailer.
Connect to Product's Web Site		Opens the product information site.
License		Shows and sets license information.
About DeviceXPlorer		Shows the versions information for DeviceXPlorer.

4.2.2 Toolbar

Toolbar and shortcut key are assigned to following as.

FUNCTION	ICON	SHORTCUT
New		Ctrl + N
Open		Ctrl + O
Save As		Ctrl + S
Cut		Ctrl + X
Copy		Ctrl + C
Past		Ctrl + V
Delete		DEL
Device Monitor		F3
Cache Monitor		F4
Update settings to Runtime		
Download settings from Runtime		
Create Port		Ctrl + P
Create Device		Ctrl + D
Create Group		Ctrl + G
Create Tag		Ctrl + T
Create Structure Tag		Ctrl + M
Create Method		Ctrl + I
Create Device Folder		Ctrl + W
Project Wizard		Ctrl + W
Device Generator		Ctrl + Shift + D
Port Generator		Ctrl + Shift + P
Tag Generator		Ctrl + Shift + T
OPC Test Client	OPC	
Option		
Project Property		Alt + F7

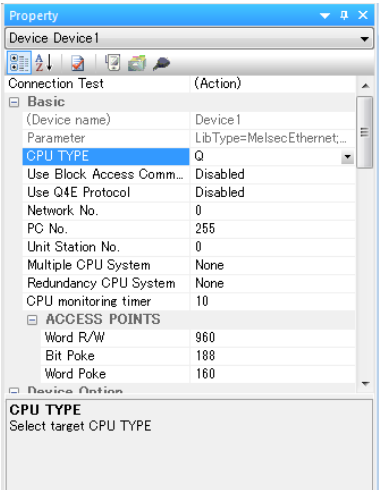
4.2.3 Another Shortcut Key

FUNCTION	SHORTCUT
Write value	(While selecting tag) Shift + Enter
Array Viewer	(While selecting tag) Ctrl + Enter

Change the Name	(While selecting Device or Port, etc.)F2
-----------------	--

4.2.4 Property Grid

The property information is expressed to a property grid as the click of the project name on Project Explorer, a Device name, or a Port name, the click of the tag name in a Tag list view, etc.
A check and change of a settings can be performed intuitively, without opening a dialog.

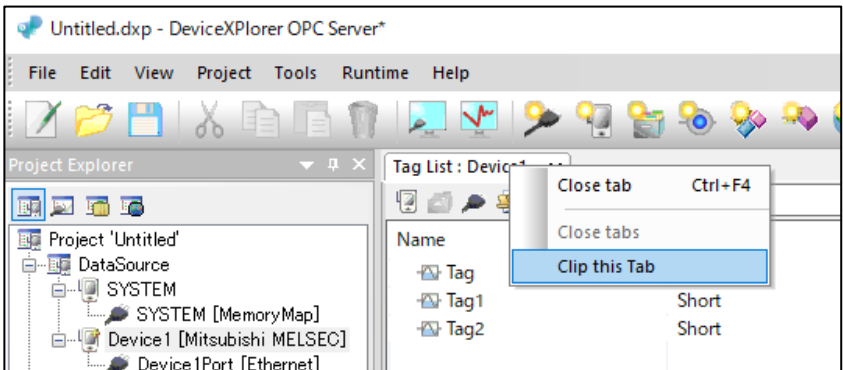


Property Grid icons are assigned to following as.

INCON	Description	INCON	Description
	Shows properties.		Shows Device settings dialog.
	Shows properties alphabetical.		Shows Group settings dialog.
	Apply button (Properties loaded are update.)		Shows Port settings dialog.

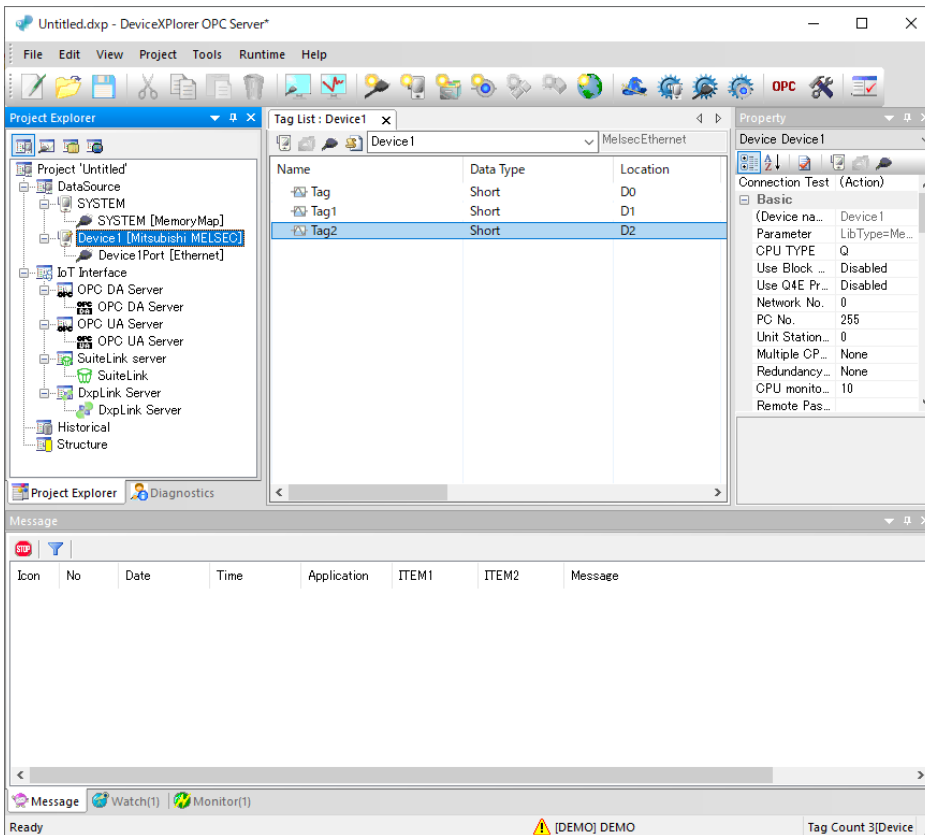
4.2.5 Clip of List View

A tab clip can be carried out if tabs, such as a Tag list, Structure list, an event script list, are right-clicked.
With clipping, it can leave a tab to always display.



4.3 Ports

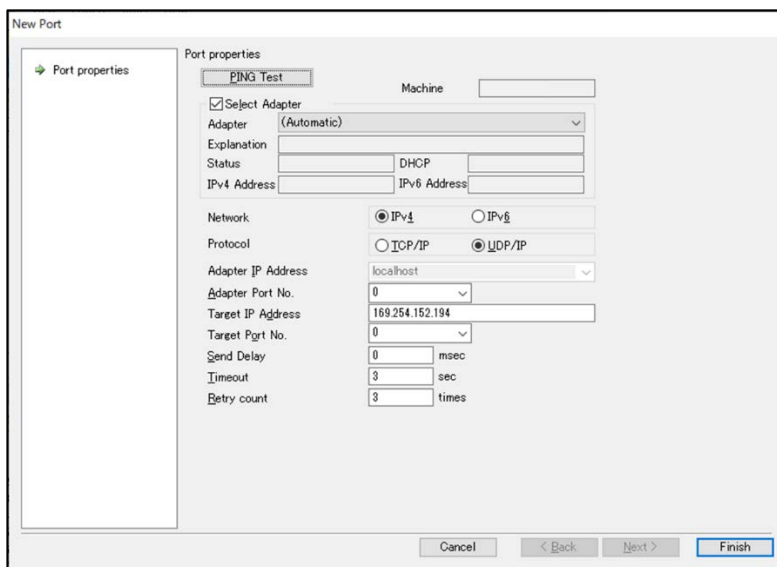
Select Project View -> Ports to view port information. Port information shows a dialog that varies depending on the type of communication port, such as Ethernet or serial.



Important

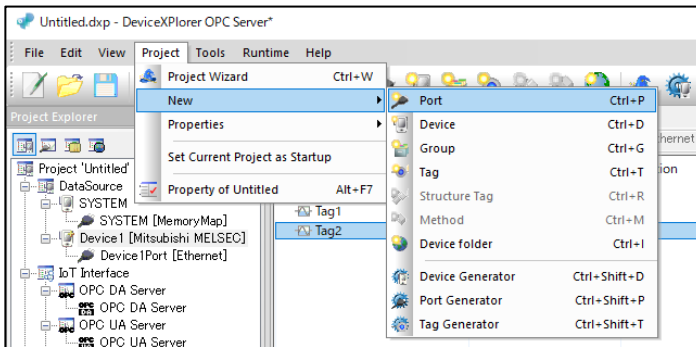
- You can create up to 255 ports.
- A Port name is a name on management and is not related to the name space accessed from a client.

The following are displayed in Port Settings of Ethernet. Setting properties differ for every communication port. Please refer to 4.3.4 if you use Ethernet port or refer to 4.3.5, if you use Serial port. For other ports, refer to the “User’s Guide (PLC Communication Edition)”.

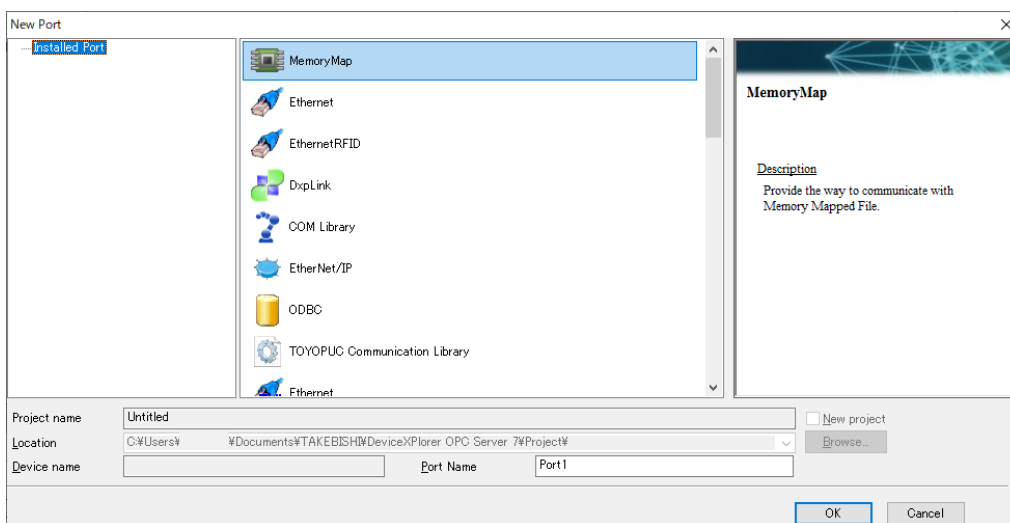


4.3.1 Creating New Ports

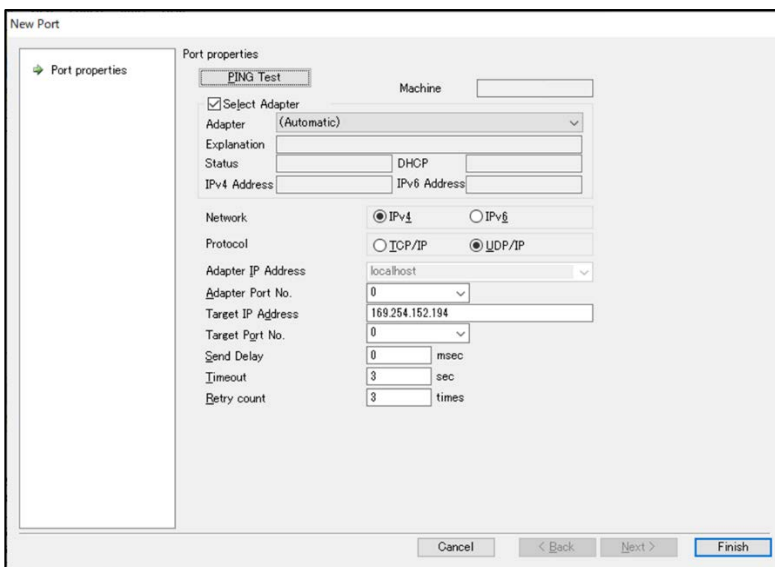
From the Projects menu, select Insert-> Port.



Select the port type and input the Port name.



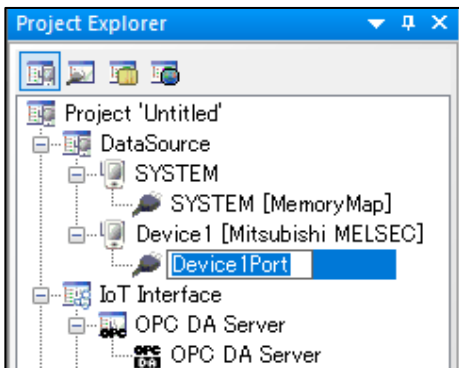
The port dialog will appear. Enter the communication parameters and click “Finish”.



4.3.2 Changing the Port Name

You can change the port name by either clicking on the port name again once it is selected, or by using the F2 key to enable in-place activation.

* You cannot change to a port name that already exists.

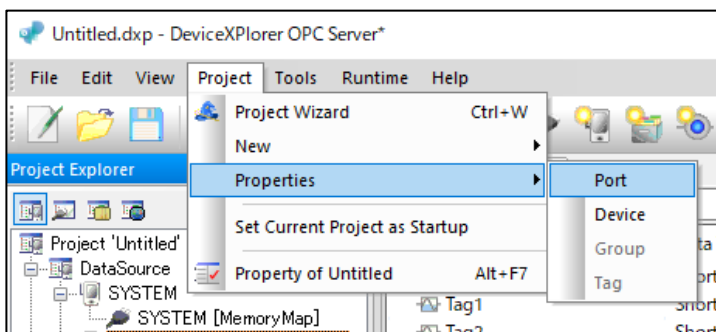


4.3.3 Port Settings

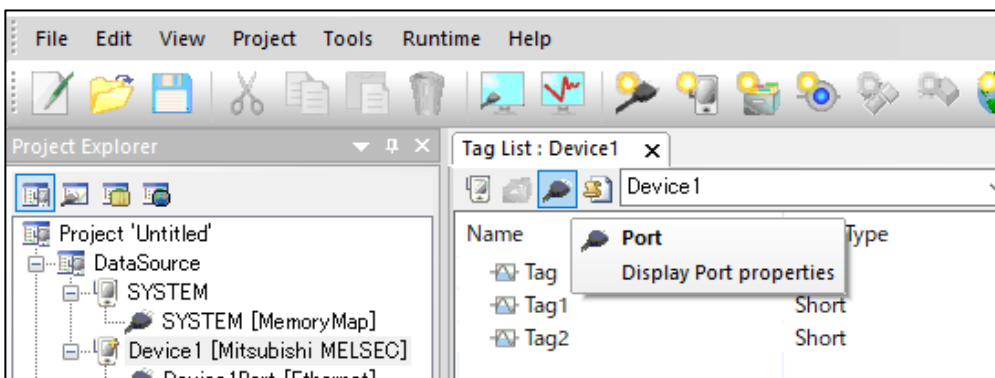
To display Port Setting dialog,

- You select a Port name in Project Explore and select Properties->Port.
- You click Port Jump button on Tag List View.
- You Right-Click and select "Property".

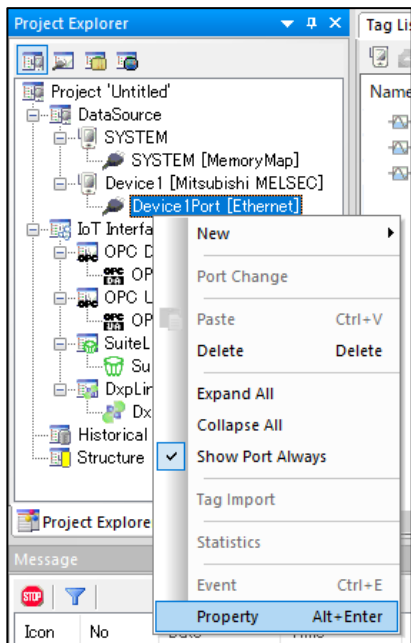
[Project menu]



[Port Jump button]



[Right Click]



4.3.4 Setting Ethernet Ports

Property [Device1Port]

Port

PING Test

Machine

☒ Select Adapter

Adapter (Automatic)

Explanation

Status DHCP

IPv4 Address IPv6 Address

Network ☒ IPv4 ☐ IPv6

Protocol ☐ TCP/IP ☒ UDP/IP

Adapter IP Address localhost

Adapter Port No. 0

Target IP Address 192.168.19.186

Target Port No. 1025

Send Delay 0 msec

Timeout 3 sec

Retry count 3 times

Item	Description
PING Test	Sends and receives a Ping command to / from the IP address of the PLC. In the environment where the transmission of a message of the PING command is limited, you cannot use this function.
Machine	Shows the computer name.
Select Adapter	Check if you fix adapter.
Adapter	Shows a list of the adapters installed on the computer.
Explanation	Shows Ethernet adapter name.
Status	Shows "Connect" when a LAN cable is connected.
DHCP	Shows "Enable" when DHCP is in use.
IPv4 Address	Shows the IPv4 address of the selected adapter.
IPv6 Address	Shows the IPv6 address of the selected adapter.
Network	Select IPv4 or IPv6.
Protocol	Select TCP/IP or UDP/IP.
Adapter IP address	Shows the IP address of the selected adapter. When you does not fix adapter, you can specify the IP address of the adapter or machine name that will communicate with the PLC. When you specify 'localhost' or '0.0.0.0', DeviceXPlorer chose available adapter automatically.
Adapter Port No.	Specify the port number of the selected adapter. If you specify port number "0", the OS will automatically assign an available port. The port number can be specified 0 or any numbers between 1024 and 65535.
Target IP address	Specify the machine IP address or machine name of the Target.
Target Port No.	Specify the port number of the Target. Set a number between 0 and 65535.
Send Delay	Specify the waiting time before sending the next command after a response is received from the other equipment on the millisecond time scale. Set a number between 0 and 65535.
Timeout	Specify the waiting time between sending a request message and receiving a response message. If a response cannot be returned within the specified time, it results in a timeout error. Set a number between 0 and 60.
Retry Count	Specify the number of times to retry when a timeout error occurs. Set a number between 0 and 10.

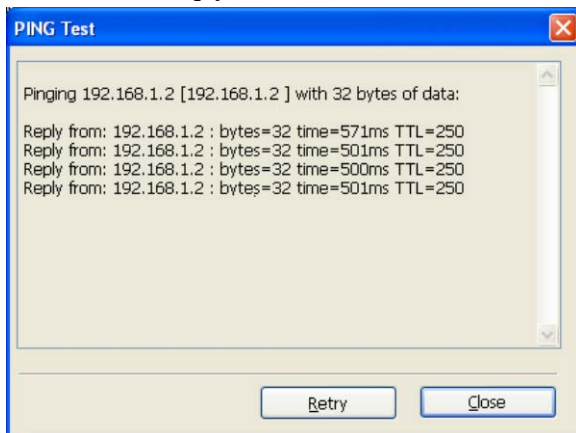
Note

When you specify port number to something other than 0 in using TCP / IP, if the connection of the port has lost, the port becomes TIME_WAIT status. Until the port becomes CLOSED status from TIME_WAIT, the port is not available. This is the specification of TCP/IP. In this time, the DeviceXPlorer cannot communicate with the controller through this port, and DeviceXPlorer outputs the log 'Open Error(Code=10048)'. If you use TCP / IP, it is recommended to specify 0 to the adapter port number and use un passive open mode at the PLC side.

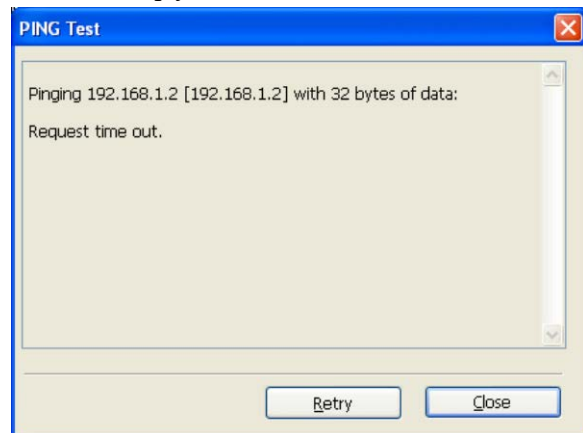
[PING]

PING is a function that checks the existence of the recipient IP address by sending an ICMP request packet to the set recipient IP address and receiving a response packet. When PING is implemented, the following information is displayed.

< Successful: "Reply from ..." >



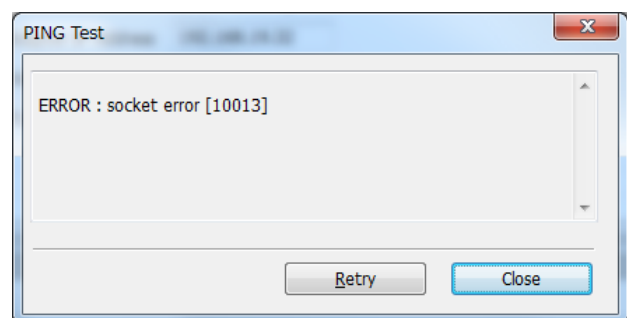
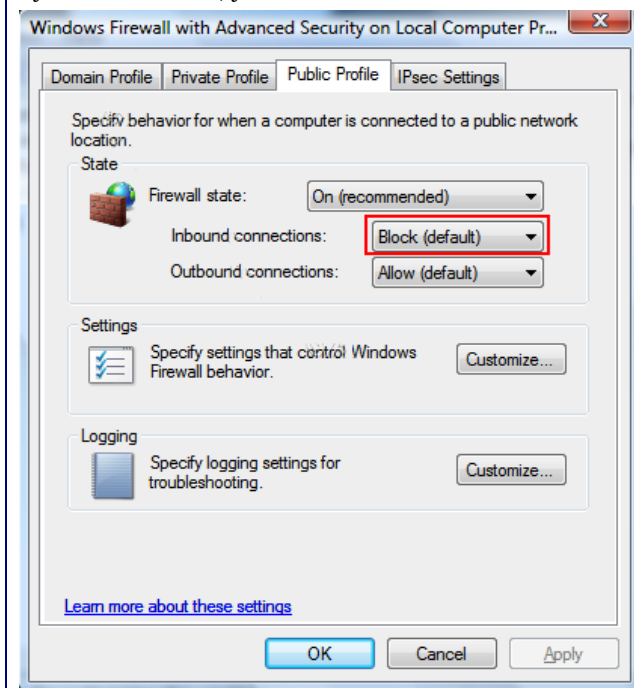
< Failed: "Reply Timeout ..." >



Important

For using PING on Windows Vista or later OS, you should set "Inbound connection" of "Public Profile" in Administration Tools-"Secured Windows Firewall" to "Allow".

And if UAC is valid, please start DeviceXPlorer as "run by Administrator". If UAC is valid and you don't start as "run by Administrator", you will see error "socket error [10013]"



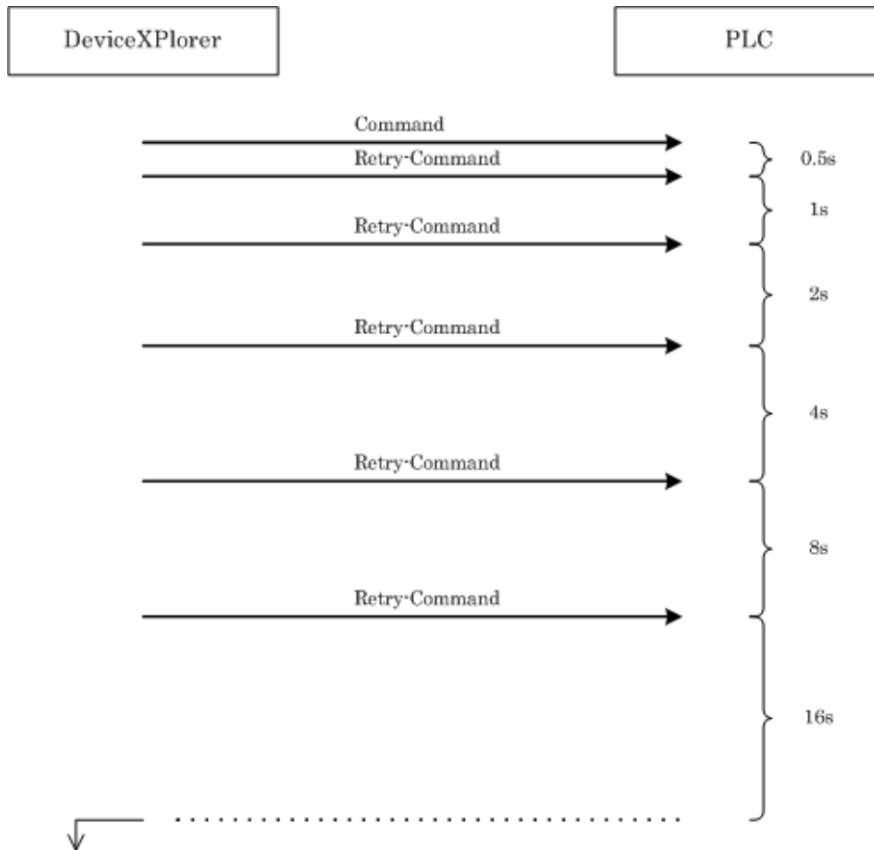
4.3.4.1 Timeout Settings for TCP / IP

When using the TCP protocol, it is necessary to set an appropriate timeout value and number of retries in DeviceXPlorer.

The following is an explanation of the relationship between the TCP communication operation and the timeout setting, assuming that the PC is running on Windows OS.

[Sending Commands from PC]

After a PC has sent a command, if there is no acknowledgement returned from the PLC, the command is resent at the TCP level at the intervals shown below.



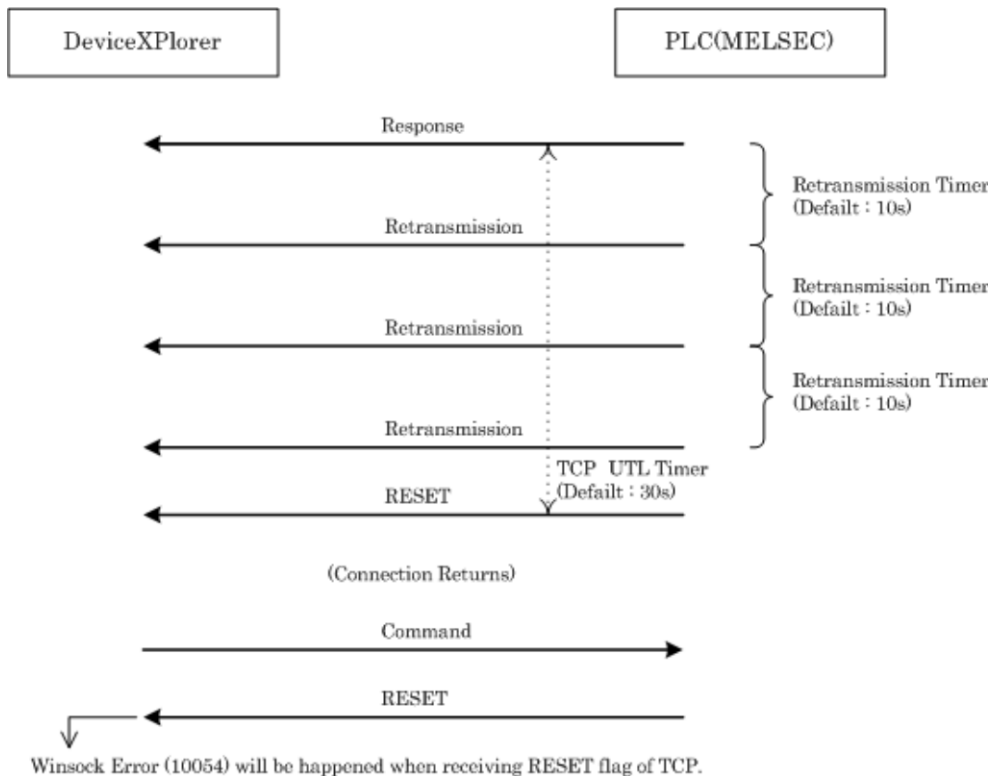
Winsock Error (10054) will be happened if ACK is not returned.

In Windows OS, the default RTO (Retransmission Time-Out) for Ethernet is 0.5 seconds. When the maximum number of retransmissions has been reached (default: 5 times / 31.5 seconds), error 10054 (the connection was reset from the remote side) is returned to the application layer. DeviceXPlorer then cuts the connection and reopens it.

The RTO is changed dynamically, and the timeout value increases by doubling and redoubling.

[Sending Response to PC from the PLC (MELSEC-Q)]

In the MELSEC-Q series, if there is no acknowledgement from the PC after the PLC has sent a response, retransmission occurs at the TCP level in accordance with the TCP retransmission timer (default: 10 seconds). When the TCP ULP timer value is exceeded, a RESET flag is sent.



While the line is disconnected, the RESET flag cannot reach DeviceXPlorer. Once the connection is restored and a command is sent from DeviceXPlorer, the RESET flag is returned as a response. At the same time, 10054 is returned to the application layer (indicating that the connection was reset from the remote side), so DeviceXPlorer cuts the connection and reopens it.

Important

If the TCP protocol is selected, we recommend that the timeout be set to "31.5 seconds or more" and the number of retries be set to "0".

4.3.5 Serial Port Settings

Port

Com Port

1

Baud Rate

19200

Parity

☒None☐Odd☐Even

Flow Control

Hardware (RTS/CTS)

Data Bit

☐7bit☒8bit

Stop Bit

☒1bit☐2bit

Send Delay

0

msec

Timeout

3

sec

Retry Count

3

times

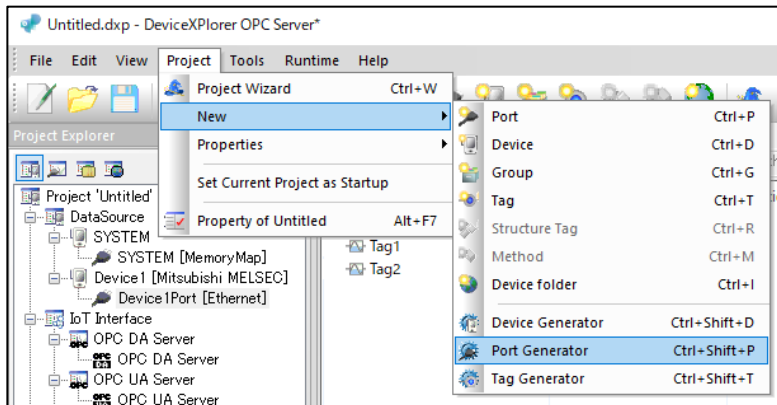
Item	Description
Com Port	Set COM port number between 1 and 256.
Baud rate	Select baud rate for communication.
Parity	Set the parity bit, on or off.
Flow control	Select “Hardware (RTS / CTS)” to perform RTS / CTS flow control.
Data length	Specify the data length.
Stop bit	Set the stop bit.
Send delay	Specify the waiting time before sending the next command after a response is received from the other equipment on the millisecond time scale. Set a number between 0 and 65535.
Timeout	Specify a number between 0 and 60 (on the second time scale) as the time to wait between sending a request message and receiving a response message.
Retry Times	Specify the number of times to retry when a timeout error occurs. Set a number between 0 and 10.

Note

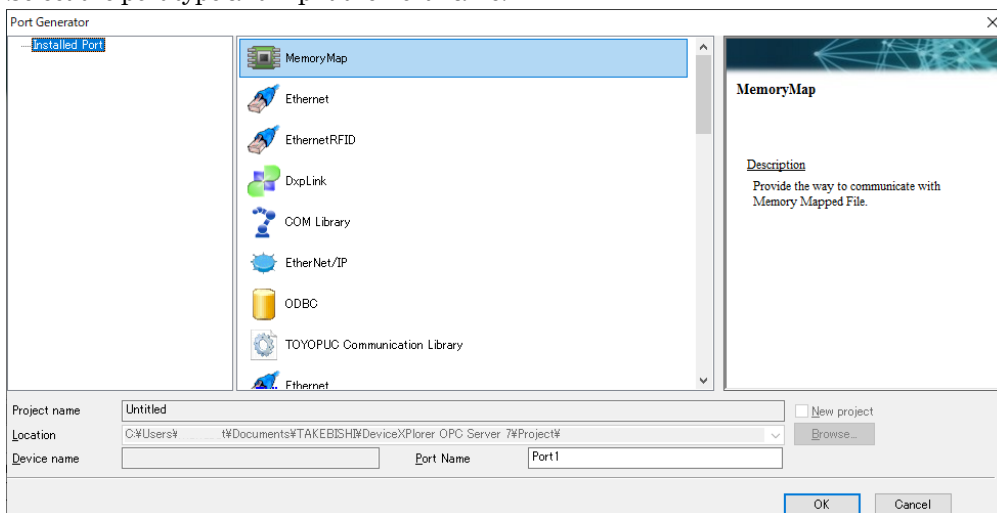
In binary communication, communication is 8 bits / character, so the data length must be set to “8”.

4.3.6 Port Generator

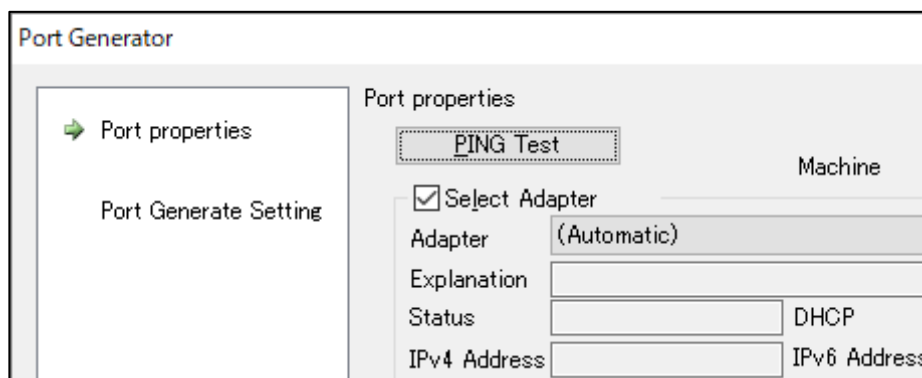
The package generation of the port setup can be carried out.
It can start from a project menu to the Insert ->Port Generator.



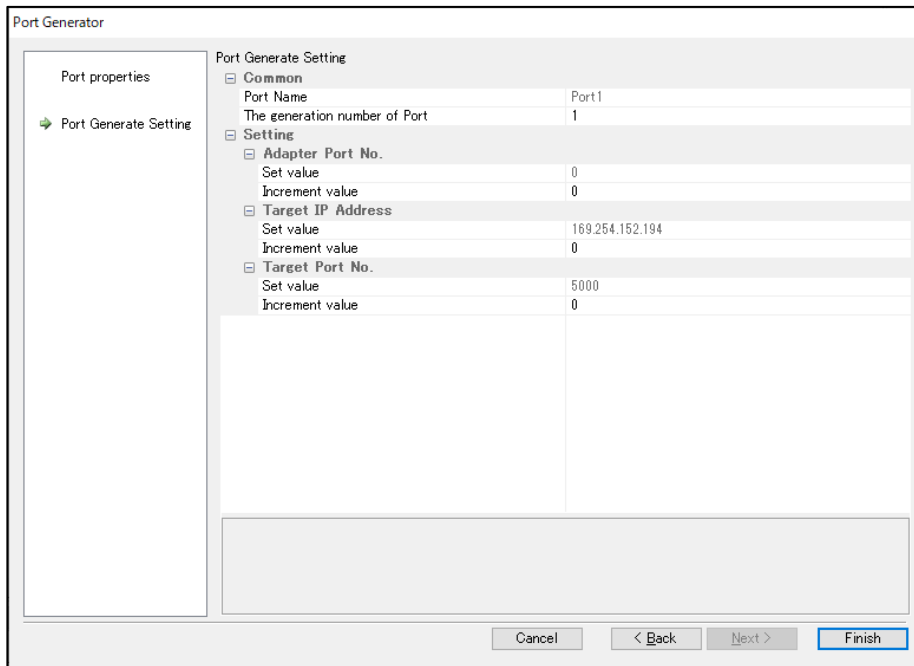
Select the port type and input the Port name.



The port dialog will appear. Enter the communication parameters and click “Next”.



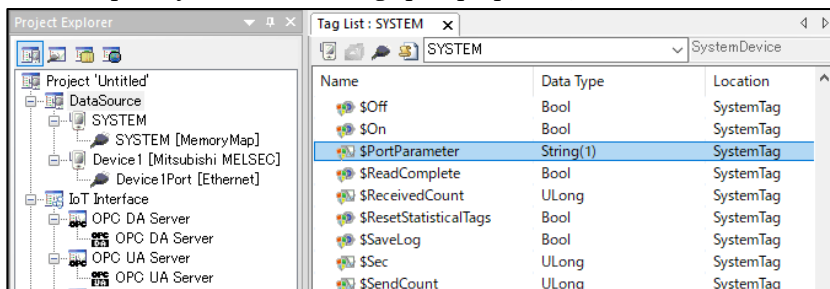
Enter the generation number of Port and Increment value and click “Finish”.



Item	Description
Port Name	Shows the 1st Port name.
The generation Number of Port	Specify the number of creation between 1 and 255.
Properties	The property which can be configured changes with communication ports. An increase can be specified depending on the Port property.

4.3.7 Dynamic Port

To create port dynamic or to change port properties, SYSTEM device has “\$PortParameter”.

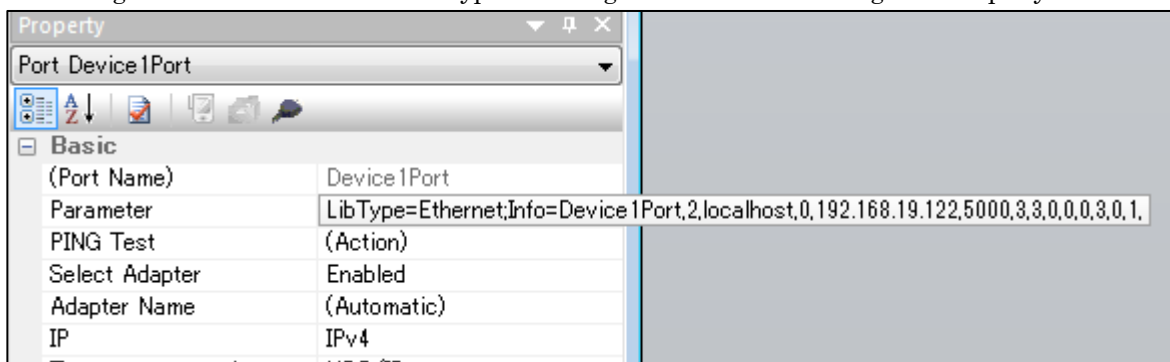


Name	Type	Contents	Data Type	Attribution
\$PortParameter	Global System Tag	The parameter system tag for port generation.	STRING	R/W

For example, to create Ethernet Port, you write this tag follows.

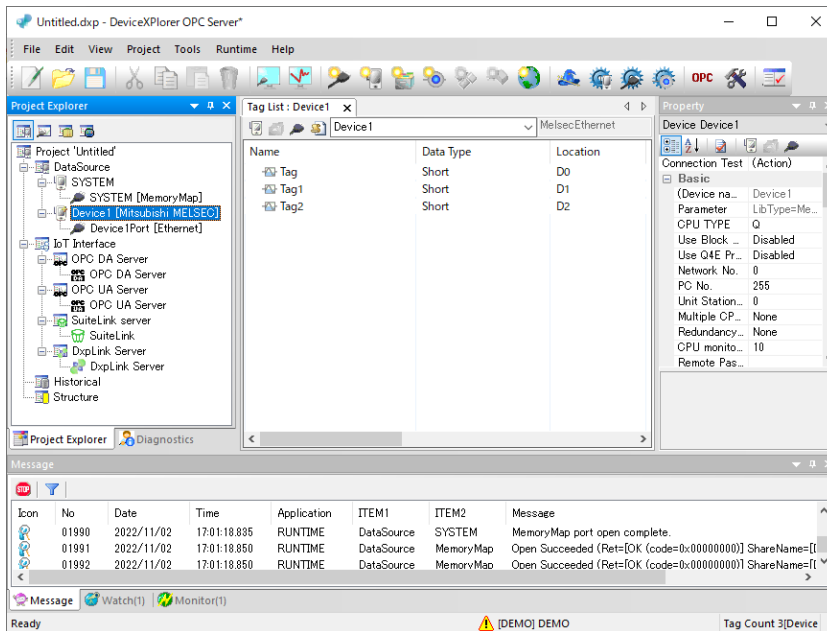
```
LibType=Ethernet;Info=Device1Port,2,localhost,0,192.168.19.122,5000,3,3,0,0,3,0,1,
```

This string format is different from Port type. You can get this Parameter string from Property Grid.



4.4 Devices

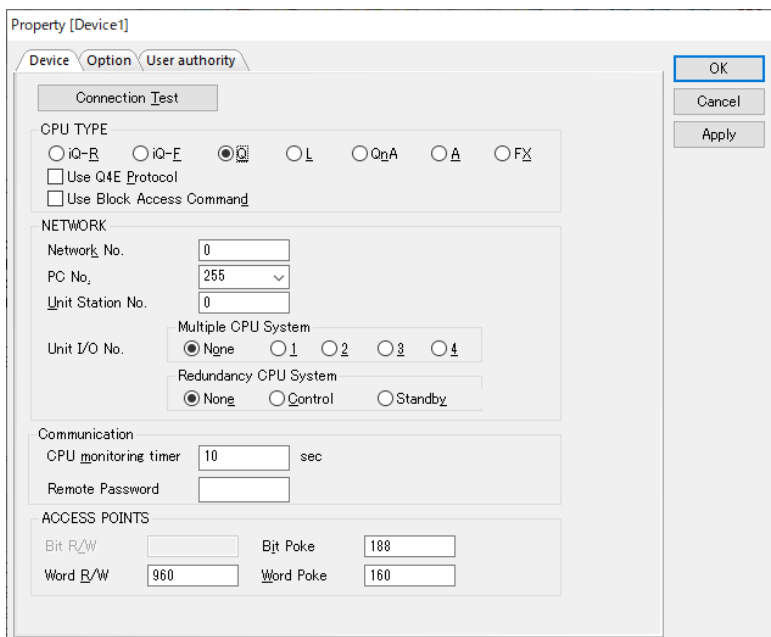
Device has a properties to communicate with target.
Select Project View -> Devices to view device information.



Important

- A Device name is concerned with the name space accessed from a client. As shown in the above, To access Tag001 in Device1, it becomes the name space "Device1.Tag001."
- There is no restriction on the maximum number of devices.

Since Device properties differ for every device type, for detail, refer to the “User’s Guide (PLC Communication Edition)”.
The following are displayed in Device of MelsecEthernet.

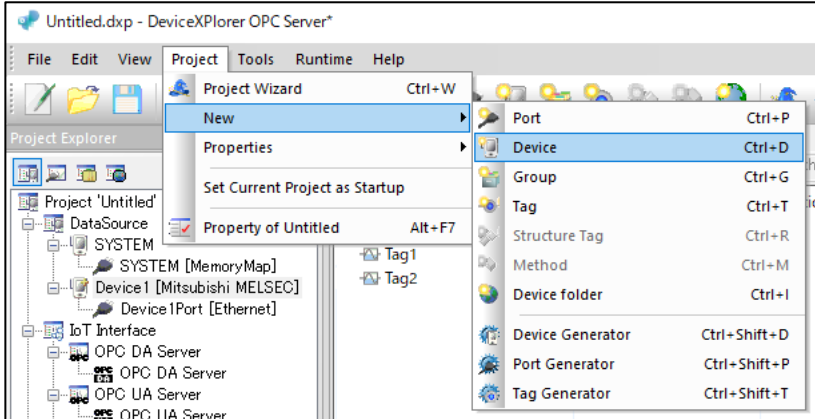


4.4.1 Creating a New Device

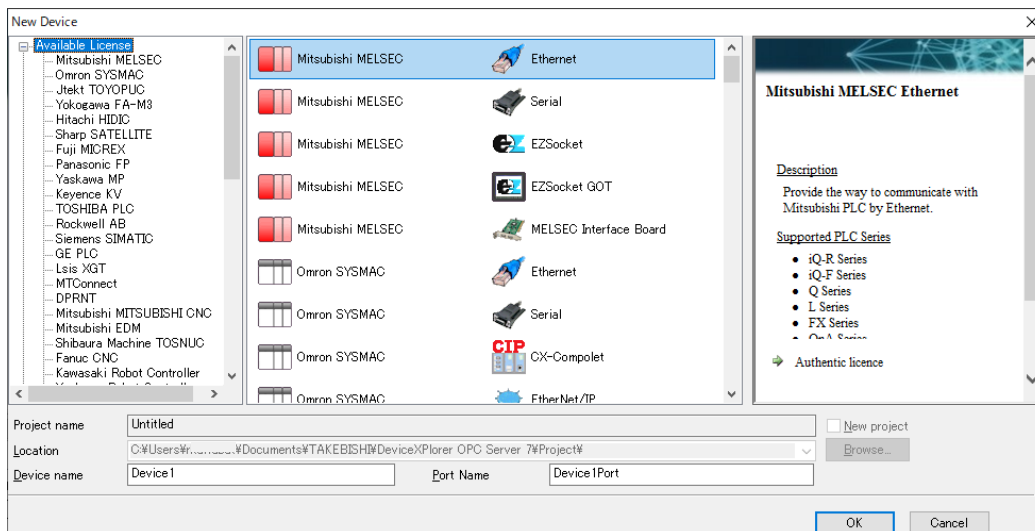
Setup steps differ by the case whether select a port or not.

4.4.1.1 Not select a Port

In Project Explorer, nothing is chosen but the "Insert->Device" from a project menu.

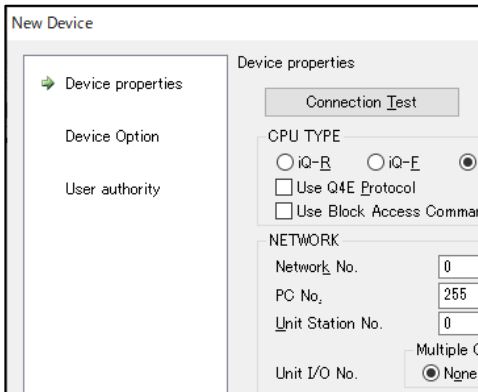


Select the Device and Port type and input Device name and Port name.

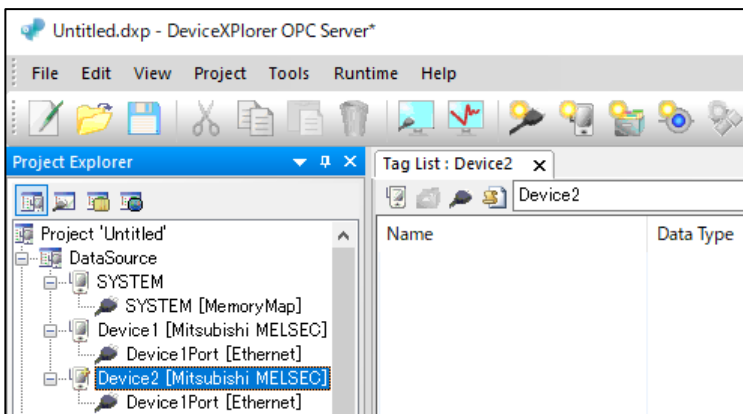


Item	Description
Device Name	Specify Device name for creation.
Port Name	Specify Port name for creation.

According to a wizard, Port Settings, Device Settings, and Device option are Settings.

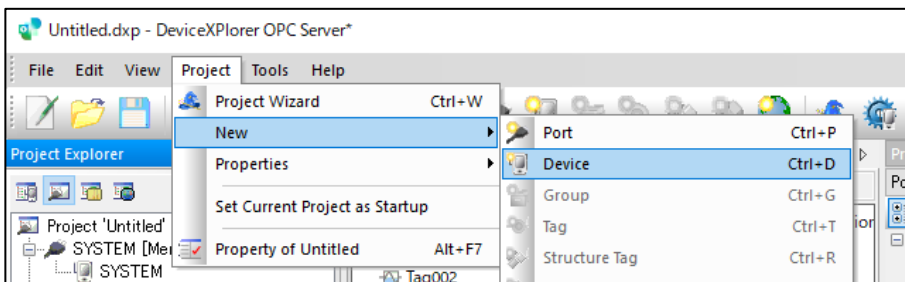


Create Port and Device by completing settings.

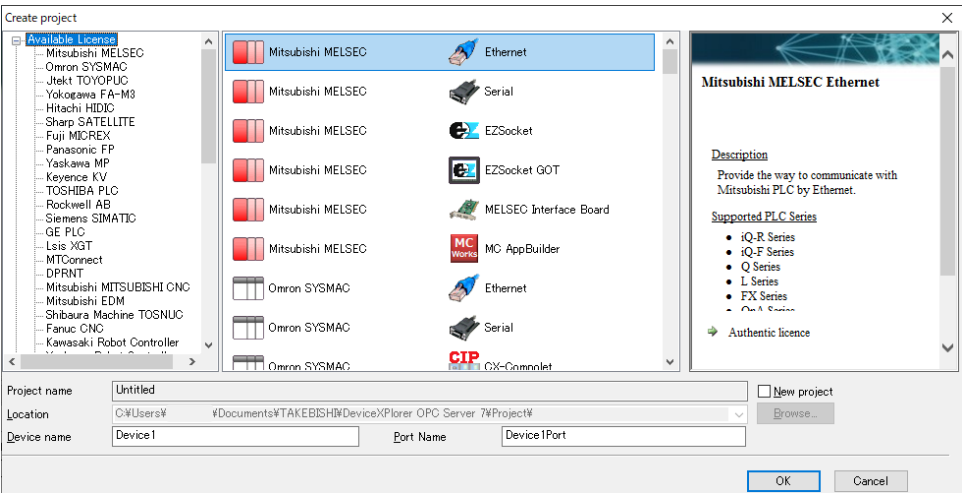


4.4.1.2 Select a Port

In Project Explorer, by selecting a port and "New->Device" from a project menu.

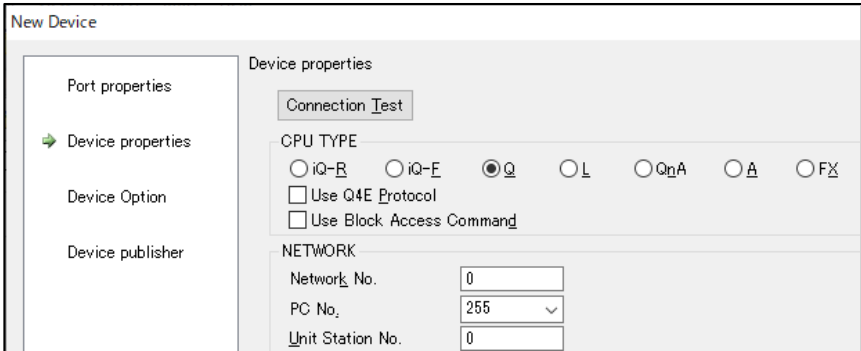


Select the Device type and input Device name.

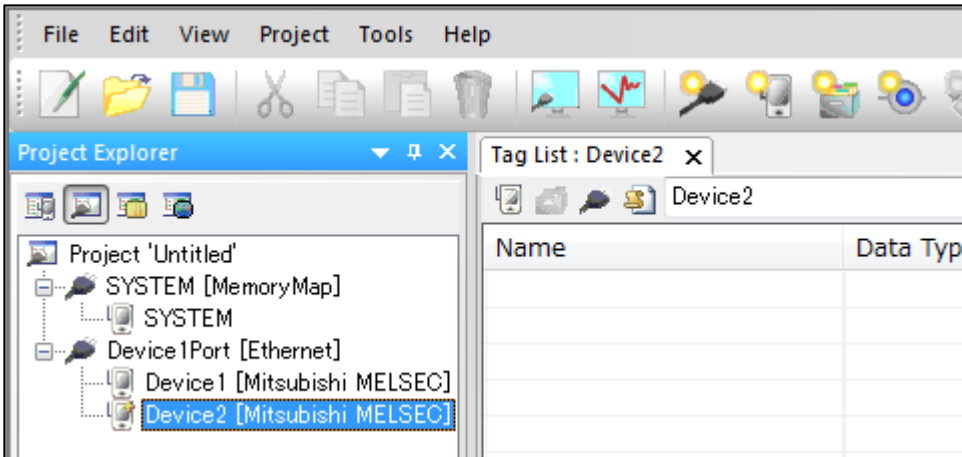


Item	Description
Device Name	Specify Device name for creation.
Port Name	Read-Only.

According to a wizard, Device Settings, and Device option are Settings.



Create Device by completing settings.



4.4.2 Connection Test

The "connection test" is prepared for Device Settings, and it can be checked whether it can communicate.

If you push "connection test" button, the following screens will be displayed and a connection test will be performed.

Item	Description
Result	Success: Connection Test Success Result: ConnectionTest Success /device=Device1, quality=Good[0xc0] Ethernet port closed.
	Error: Connection Test Error Result: ConnectionTest Error /device=Device1, quality=Comm Failure[0x18] Ethernet port closed.
Retry	Retry Connection Test.
Close	Close this window.

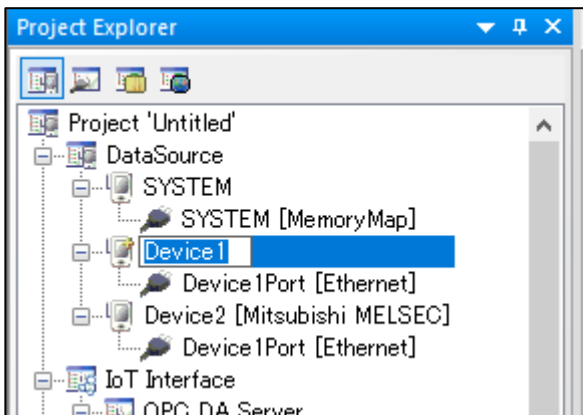
Important

- Connection Test may be failed in the situations where communication has already been performed, such as under OPC client connection and Device Monitor. (in exceeding the number connectable simultaneous such as the case which the session of TCP/IP is already established.)
- In Modbus connection, the 1st holding register is used for connection test, and when Modbus controller returns reply message, the Connection test succeeds.(no check for reply message contents)

4.4.3 Changing the Device Name

You can change the device name by either clicking on the device name again once it is selected, or by using the F2 key to enable in-place activation.

* You cannot change to a device name that already exists.

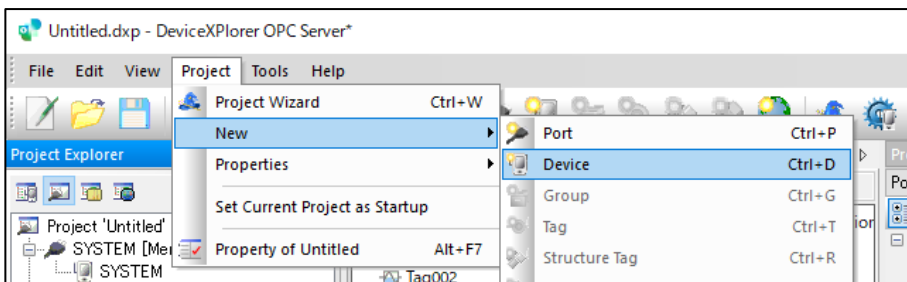


4.4.4 Device Settings

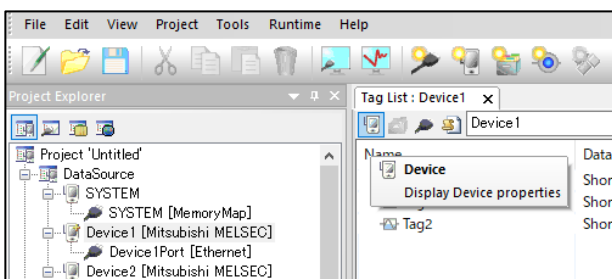
To display Port Setting dialog,

- You select a Device name in Project Explore and select Properties->Device.
- You click Device Jump button on Tag List View.
- You Right-Click and select "Property".

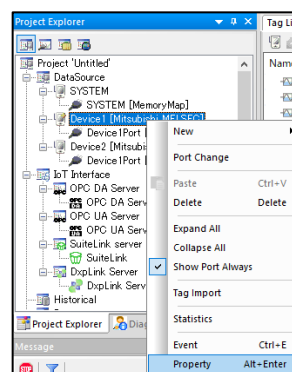
[Project menu]



[Device Jump button]



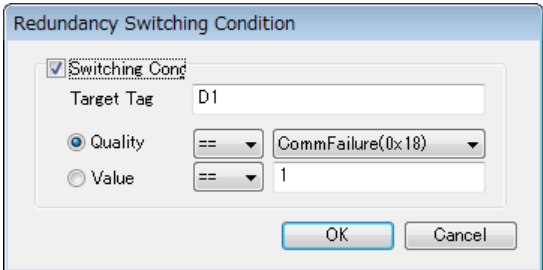
[Right-Click]



The following screens are displayed in MELSEC Ethernet.
For detail, refer to the “User’s Guide (PLC Communication Edition)”.

4.4.5 Device Options

This tab allows you to set proprietary device information that does not depend on the PLC, such as the simulation mode and redundant communication.

Item	Description
Communication Settings	Basic communication setting
Update Rate	It is a communication cycle used for SuiteLink, DxpLink , script.
Fix Update Rate	Enable to fix 'Update Rate' of Device option. When you don't fix it, DeviceXPlorer communicates by setting in the period when it is most suitable for every tag in the practice period of designation and the script in the update period from OPC client.
Max age	The allowed time of the latest value is specified. If 10000msec, it treats as the latest value for 10 seconds. It does not support in SuiteLink and DxpLink.
Skip communication after Retry	When a timeout error occurs, DXP Server cancel the transmission and reception of the following communication packet.
Slow polling mode	Enable slow poling mode. When communication error occurs, DeviceXPlorer slows reading period.
Slow Interval	Specify the interval in Slow polling mode.
Return error immediately	In Slow polling mode, DeviceXPlorer returns error immediately without communicating.
Redundant Communication	Enable the Redundancy to switch communication path.
Standby device	This is redundancy communication by using standby device/port.
Standby type Cold / Warm	Cold: When communication trouble occurs, DeviceXPlorer communicates with standby device for the first time. Warm: To switch quickly, usually communicate with standby device.
Switching Type Auto / Manual	Auto: Switching automatically when communication trouble occurs. Manual: Switching by System Tag "\$StandBy".
Return to primary automatically	In the case of warm standby and Automatic, this can change to a basic system automatically at the time of the communication return of the basic system.
Retry by Both System	Try to connect a standby system in case that failed to connect.
Switching Condition	Usually , when timeout occurs and when all retries go wrong, it changes. The change conditions using a tag can be specified by putting in a check. 
Port close when switching	Close the port when switching.
Simulation Signal	Enable simulation.
Sin	The simulation value varies with the sine curve.
Ramp	The simulation value varies incrementally.
Random	The simulation value varies randomly.
Shared Memory	The simulation value is maintained statically.
DxpLink	Enable to DxpLink
Communication Term	Specify the communication term(min) which is communication time after DxpLink client requests.
SuiteLink	Enable the SuiteLink interface.
Application Name	Shows the currently set SuiteLink application name.
Topic Name	Allows you to set a topic name of your choice for each device.
Other	Other settings
Save Unknown Item	Automatically registers tag, which is registered from the client, to the Tag View.

Reject Unknown Item	Rejects dynamic tags.
Display System Tag	Shows system tags.
System Tag uses \$ char	Uses the “\$” sign at the beginning of system tags.

4.4.6 Device User Authority

Configure settings when using user privileges.

In communication with upper-level systems (such as OPC UA) using user name and password, you can specify access privileges from IoT systems when each user is specified.

Property [Device1]

Device Option **User authority**

User list

- Administrator
- Engineer
- Operator

permissions

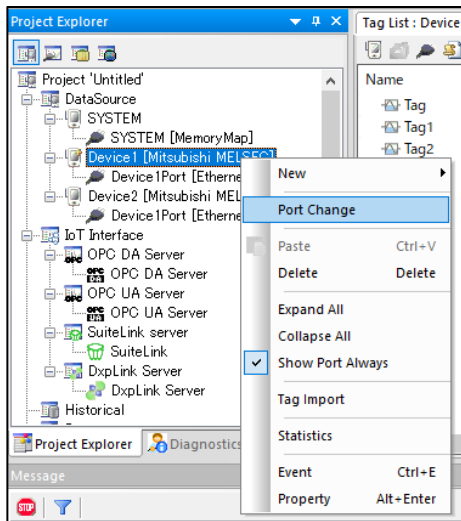
Permission
Full control
Reading
Write and run
Read history value
Insert historical value
History value update
Delete history value

OK
Cancel
Apply

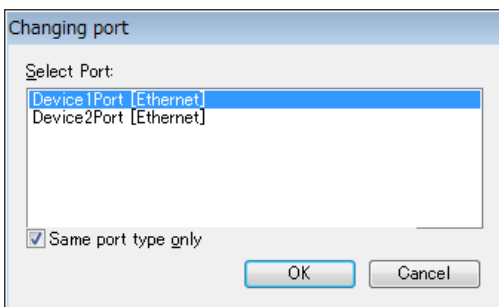
Item	Description
Full Control	Allow all permissions.
Reading	Value readout Allows the reading of values.
Write and run	Allows writing of values.
Read history value	Allows reading of historical values.
Insert historical value	Allows insertion of historical values.
History value update	Allows updating of historical values.
Delete history value	Allows erasure of historical values.

4.4.7 Changing Ports

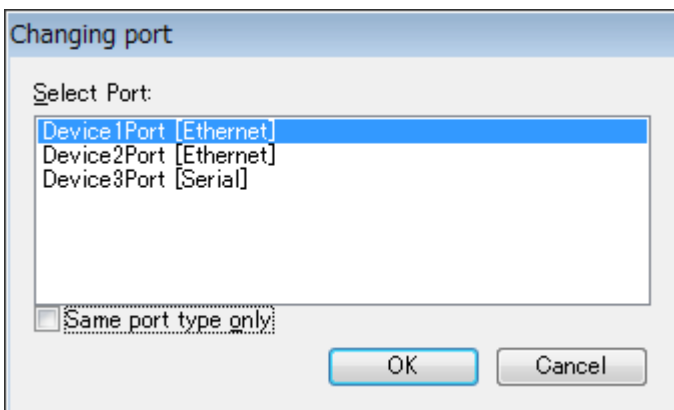
To change a port assigned to a device, select a device node in the Project Explorer, show the pop-up menu, and click “Port Change”.



The available ports will appear. Select the port you wish to change to.



If there is no check in the “Same type of port” checkbox, all the ports will be shown and it will be possible to change to a port of a different type.



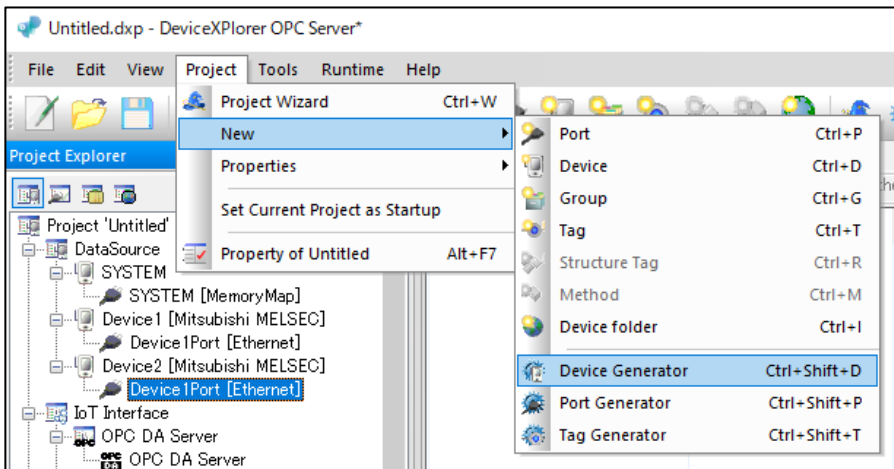
Important

If you change to a port of a different type, the information set for the device and the tag will be lost.

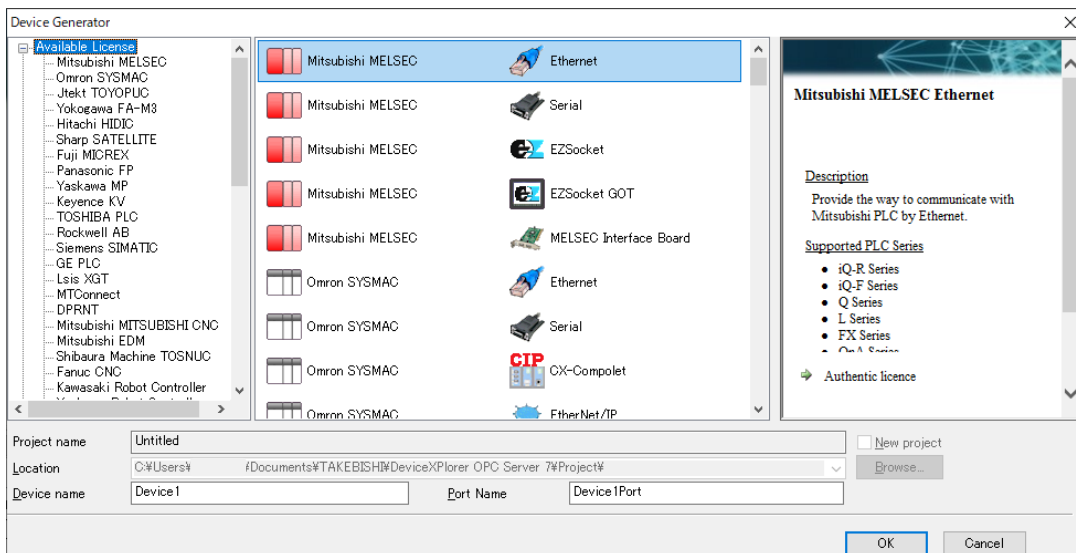
4.4.8 Device Generator

The package generation of the port setup can be carried out.

It can start from a project menu to the Insert ->Device Generator.



Select the device and port type and input the Device name and Port name.



Generation wizard steps differ by the case whether select a port or not.

	Not selected Port	Select Port
1	Port Properties	(none)
2	Device Properties	Device Properties
3	Device Option	Device Option
4	Device Generate Settings	Device Generate Settings
5	Port Generate Settings	(none)

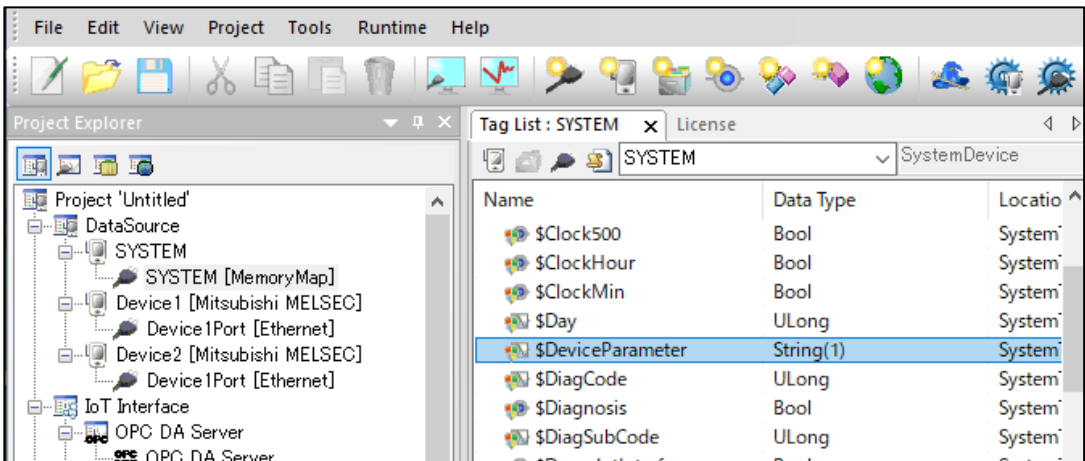
Enter the generation number of Device and Increment value and click “Finish”.

The screenshot shows the 'Device Generator' application window. On the left is a sidebar with four items: 'Device properties', 'Device Option', 'User authority', and 'Device Generate Setting' (which is selected and has a green arrow next to it). The main area is titled 'Device Generate Setting' and contains two expandable sections. The 'Common' section is expanded, showing 'Device name' as 'Device3' and 'The generation number of Device' as '1'. The 'Setting' section is also expanded, showing 'PC No.' with 'Set value' as '255' and 'Increment value' as '0'. At the bottom of the window are four buttons: 'Cancel', '< Back', 'Next >', and 'Finish' (which is highlighted with a blue border).

Item	Description
Device Name	Shows the 1st Device name.
The generation Number of Device	Specify the number of creation between 1 and 255.
Properties	The property which can be configured changes with device type. An increase can be specified depending on the Port property.

4.4.9 Dynamic Device

To create device dynamic or to change device properties, SYSTEM device has “\$DeviceParameter”.

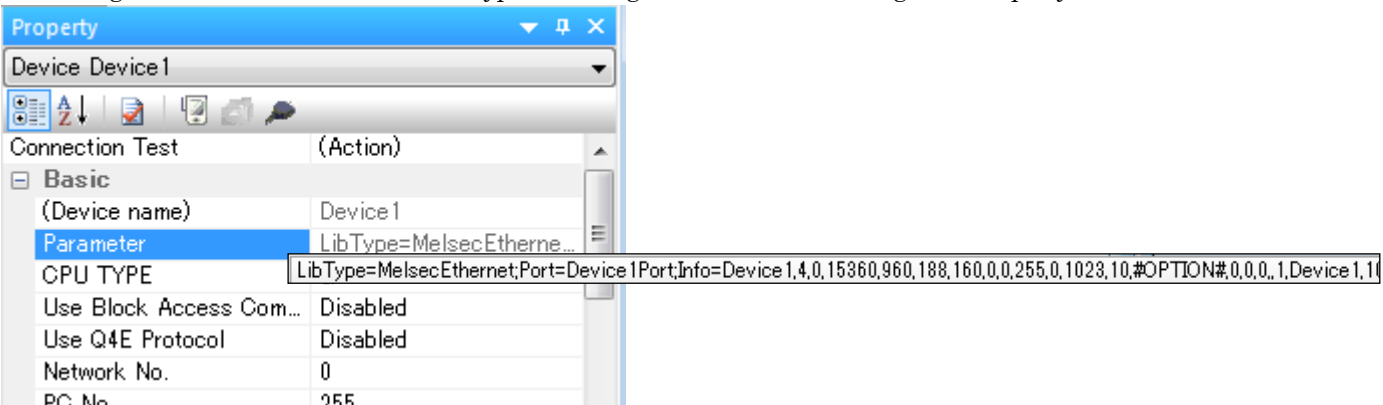


Name	Type	Contents	Data Type	Attribution
\$DeviceParameter	Global System Tag	The parameter system tag for device generation.	STRING	R/W

For example, to create device1, you write this tag follows.

```
LibType=MelsecEthernet;Port=Device1Port;Info=Device1,4,0,15360,960,188,160,0,0,255,0,1023,10,  
#OPTION#,0,0,0,1,Device1,1000,0,0,1,1,0,0,0,0,10000,0,1,1,1,1,0,D1,0,0,24,0,1,1,1,1
```

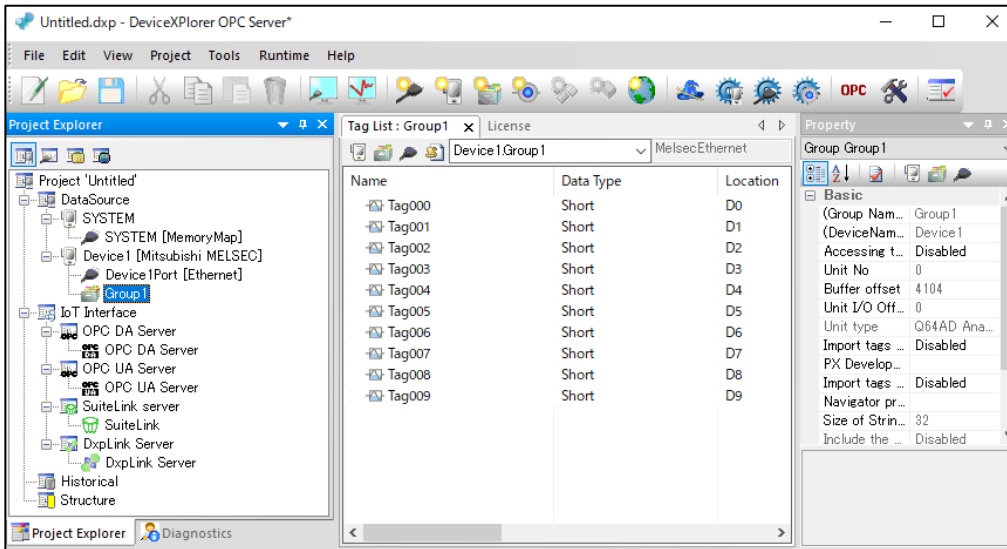
This string format is different from Device type. You can get this Parameter string from Property Grid.



4.5 Groups

A Group is a unit which manages tags.

Select Project View -> Groups to view group information.



Important

- A Group name is concerned with the name space accessed from a client. As shown in the above, To access Tag001 in Group1 of

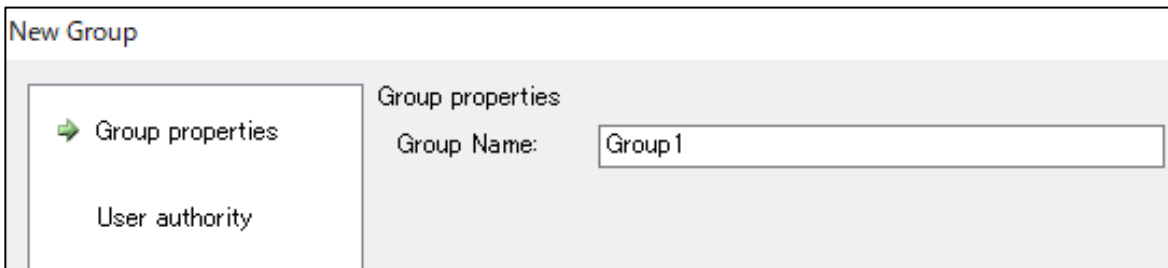
Device1, it becomes the name space "Device1.Group1.Tag001."

- There is no restriction on the number of groups.

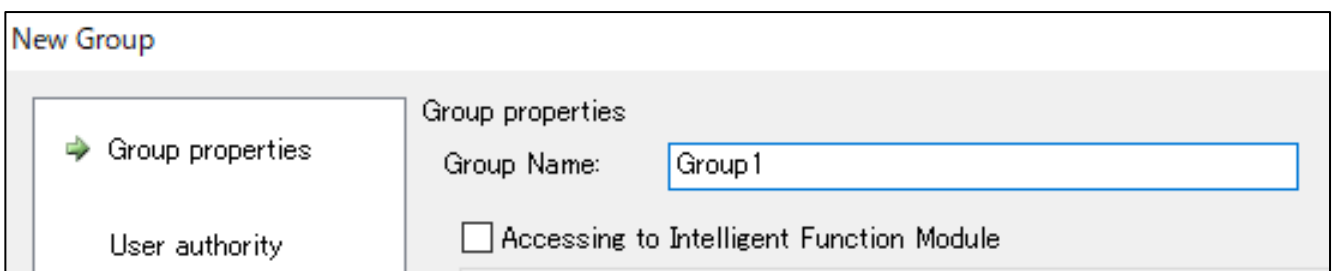
Usually, Group has only name. But some devices have properties.

Since Group properties differ for every device type, For detail, refer to the "User's Guide (PLC Communication Edition)".

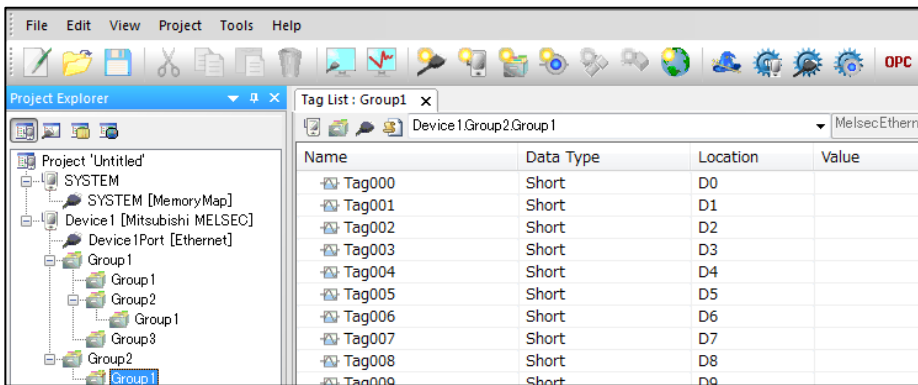
[Only Group name] Shared memory etc.



[Some properties] MelsecEthernet etc.

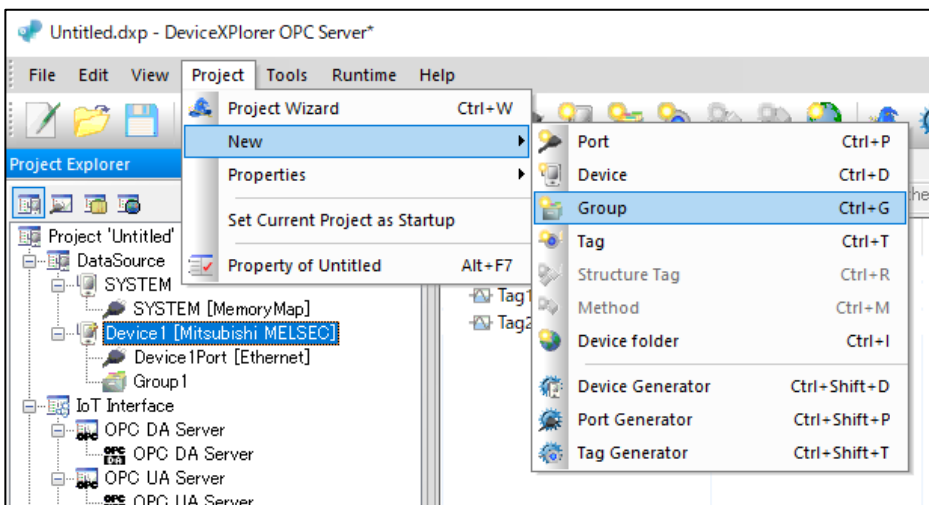


Group can be freely created by a layered structure to a Device subordinate.

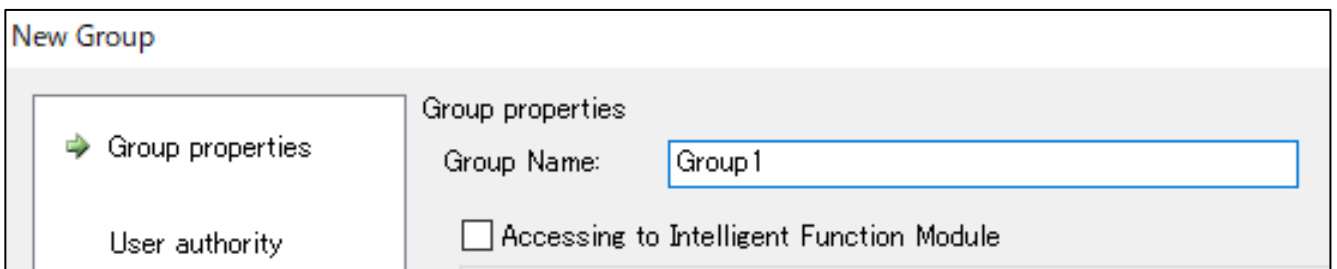


4.5.1 Creating a New Group

From the Projects menu, select Insert-> Group.

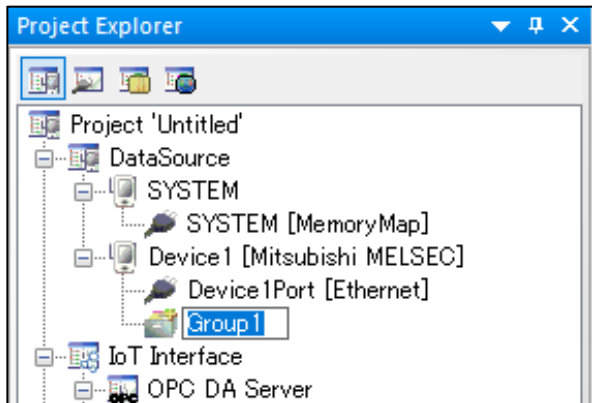


The Group dialog will appear: Enter the Group name and other properties and click “Finish”.



4.5.2 Changing the Group Name

You can change the group name by either clicking on the group name again once it is selected, or by using the F2 key to enable in-place activation.



Note

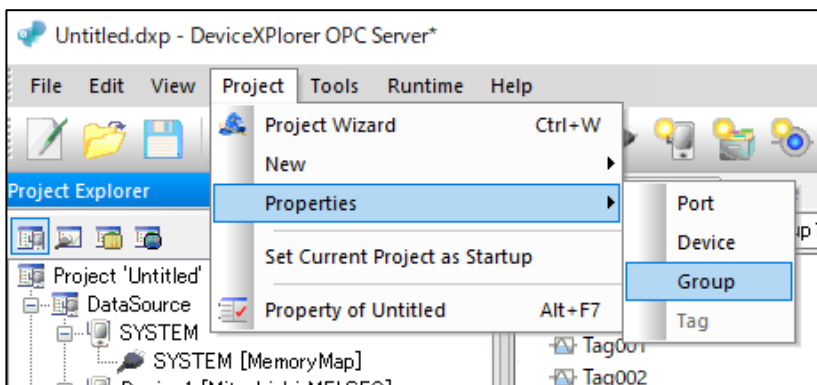
It is necessary to set a name that is not duplicated within the same layer.

4.5.3 Group Settings

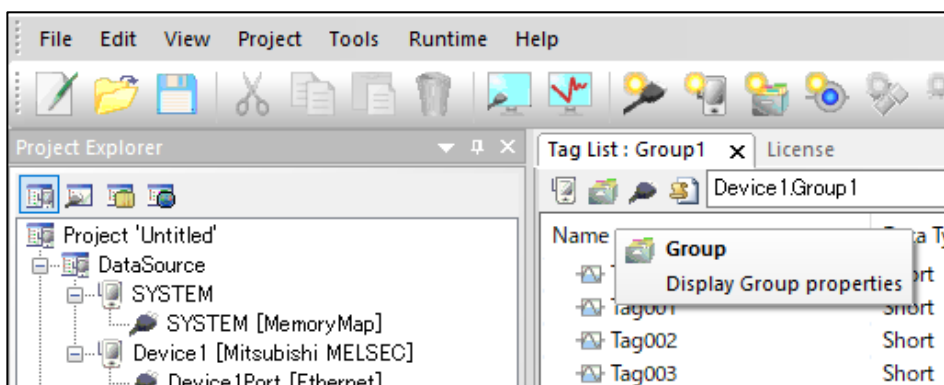
To display Group Setting dialog,

- You select a Group name in Project Explore and select Properties->Group.
- You click Group Jump button on Tag List View.
- You Right-Click and select "Property".

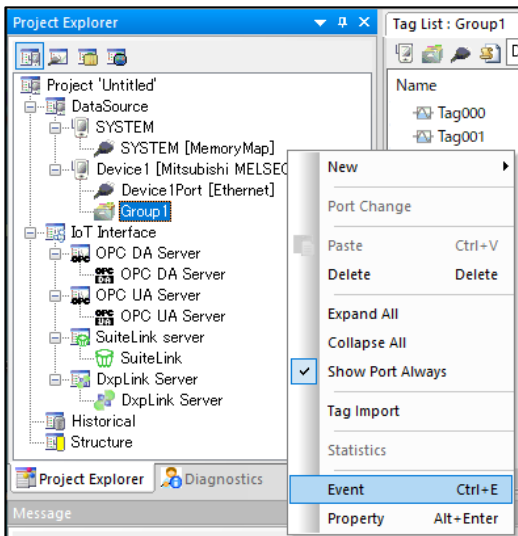
[Project menu]



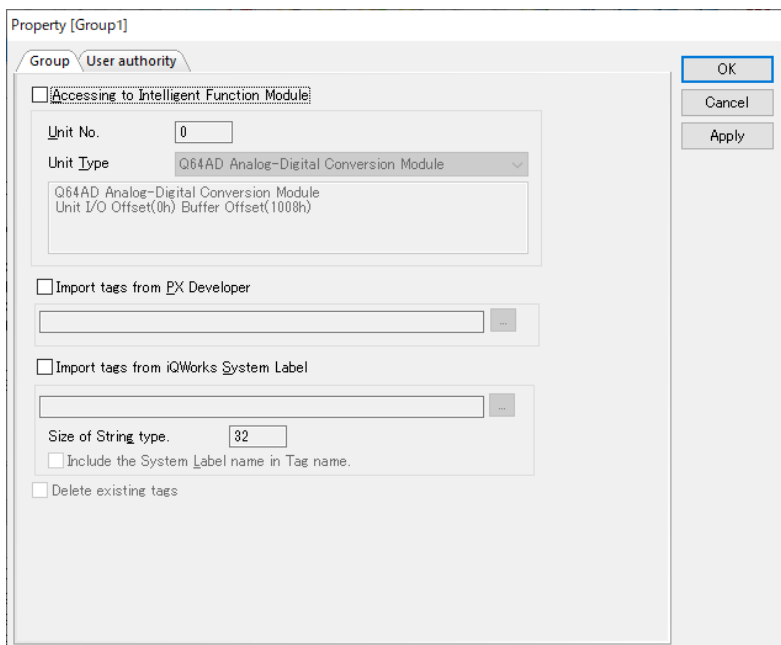
[Group Jump button]



[Right Click]



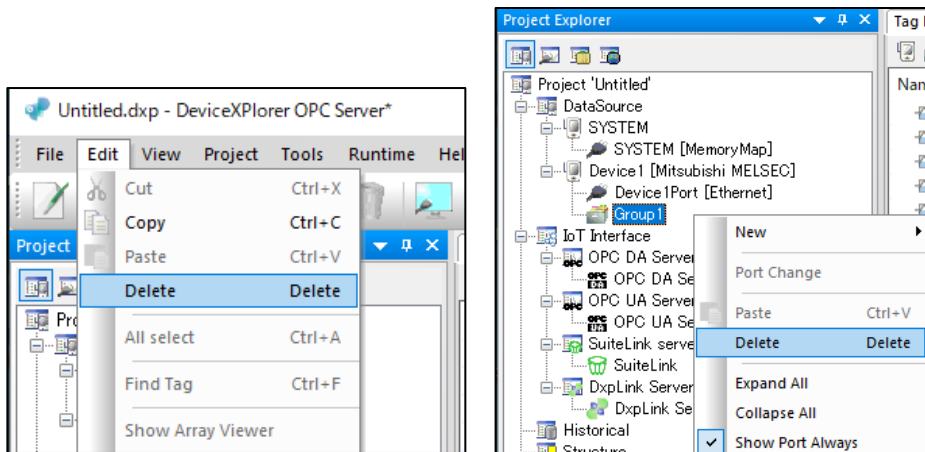
The group dialog will appear. Enter the communication parameters and click “OK”.
For detail, refer to the “User’s Guide (PLC Communication Edition)”.



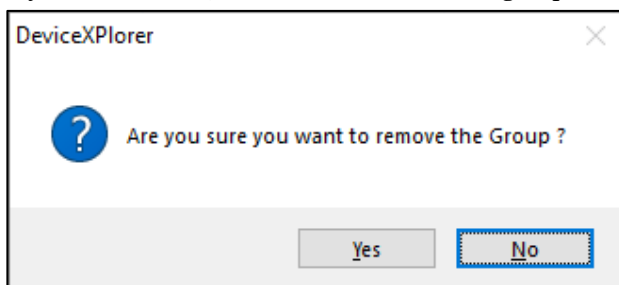
4.5.4 Deleting Group

To delete Group

- You select a Group name in Project Explorer and select Edit->Delete.
- You Right-Click and select "Delete".



If you select delete, DeviceXplorer delete the group and tags which exist under the group.

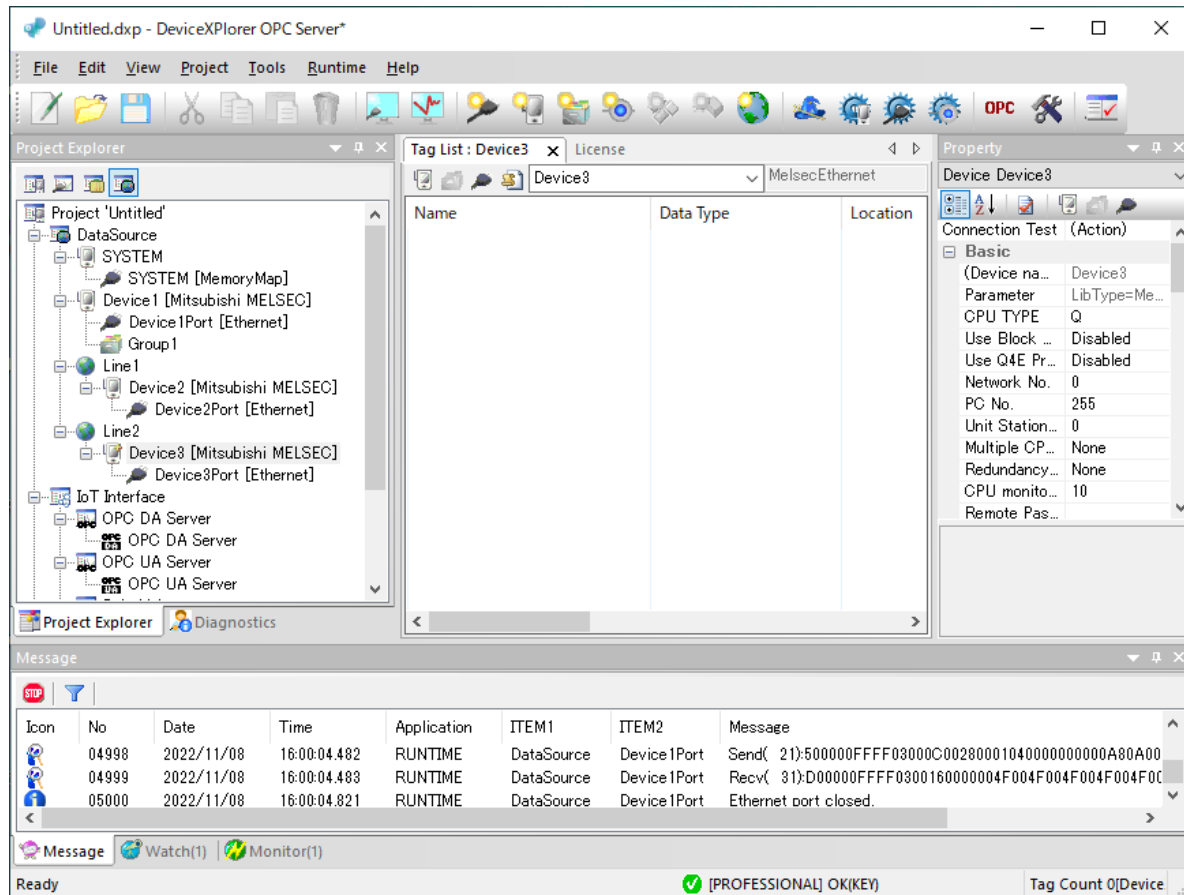


Important

- It cannot delete, when the tag in a group is communicating.

4.6 Device Folder

A Device Folder is a unit which manages devices.



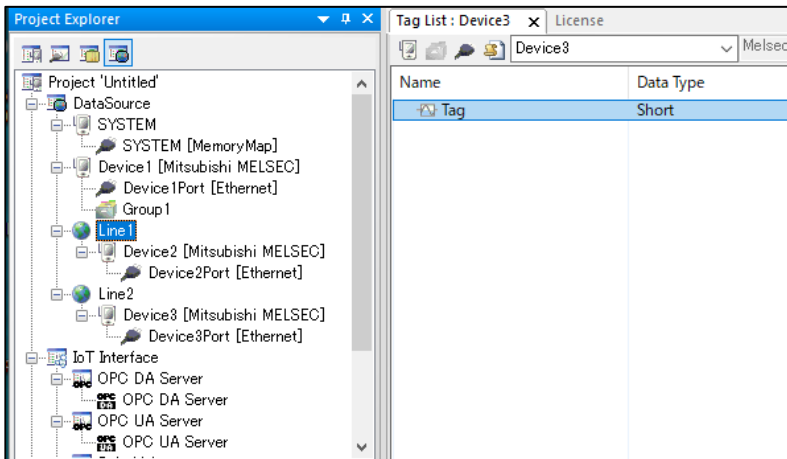
Important

- Device Folder does not be concerned with the name space.

Device Folder has only name.



Device Folder can be freely created by a layered structure to a Device subordinate.

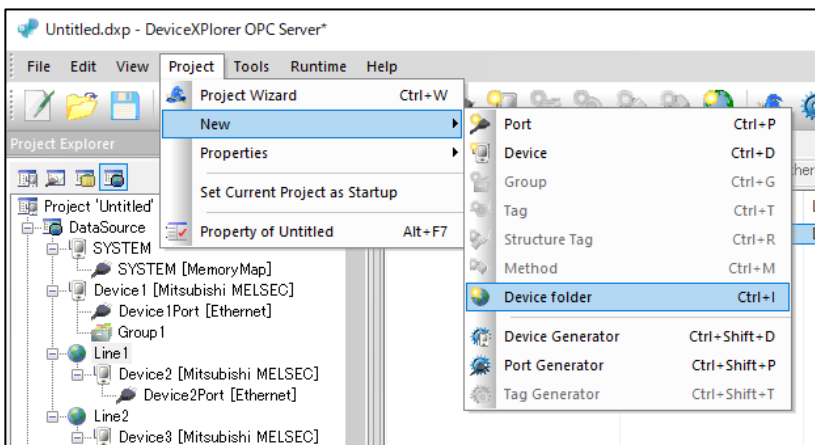


Important

-There is no restriction on the number of Device Folders.

4.6.1 Creating a New Device Folder

From the Projects menu, select Insert-> Device Folder.

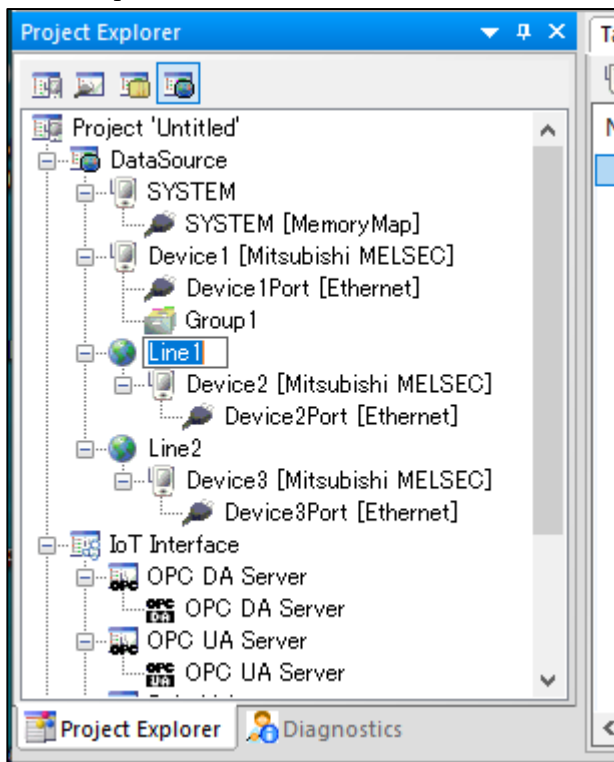


The Device Folder dialog will appear. Enter the Device Folder name and click "Finish".



4.6.2 Changing the Group Name

You can change the group name by either clicking on the group name again once it is selected, or by using the F2 key to enable in-place activation.



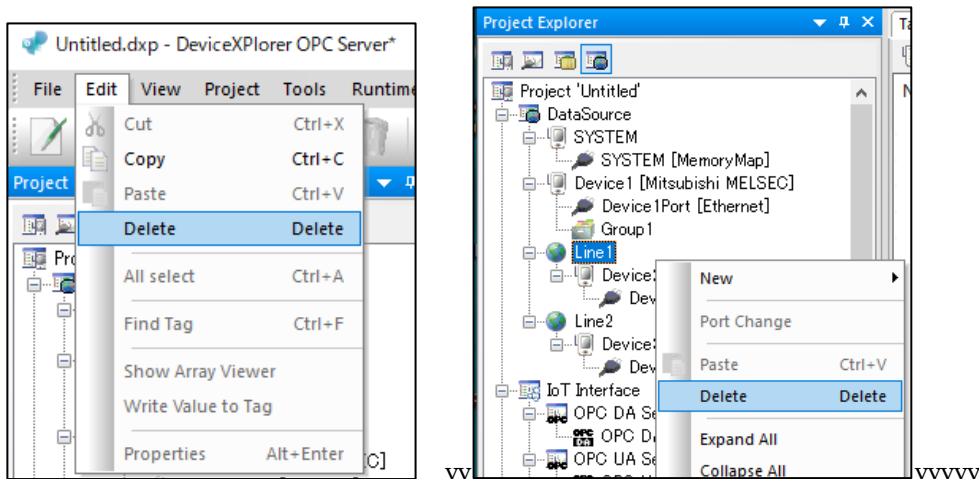
Note

It is necessary to set a name that is not duplicated within the same layer.

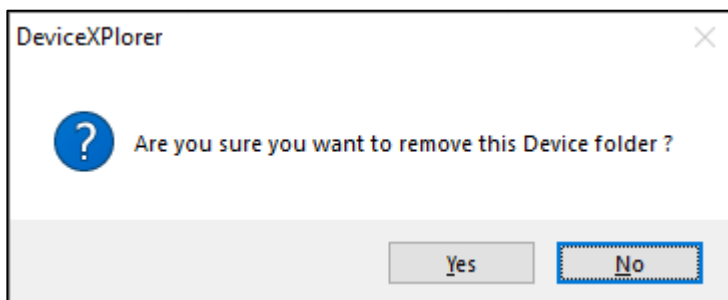
4.6.3 Deleting Device Folder

To delete Device Folder

- You select a Device Folder in Project Explore and select Edit->Delete.
- You Right-Click and select "Delete".



If you select delete, DeviceXplorer delete the group and tags which exist under the Device Folder.

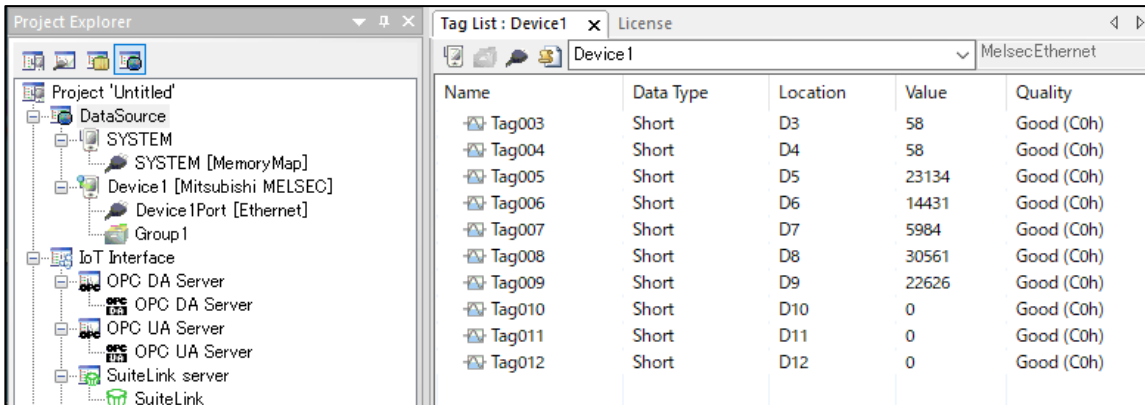


Important

- Even if it deletes a Device Folder, the Device of contents, Groups and Tags are not deleted.

4.7 Tags

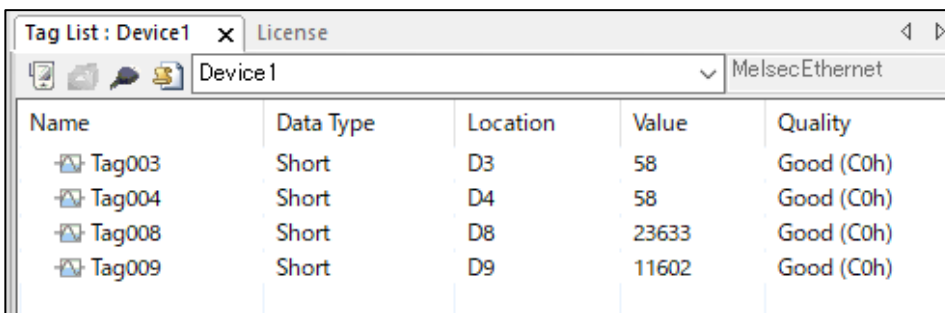
Tags define names and attributes of each register of the PLC and are used by the client to access data.



Name	Data Type	Location	Value	Quality
Tag003	Short	D3	58	Good (C0h)
Tag004	Short	D4	58	Good (C0h)
Tag005	Short	D5	23134	Good (C0h)
Tag006	Short	D6	14431	Good (C0h)
Tag007	Short	D7	5984	Good (C0h)
Tag008	Short	D8	30561	Good (C0h)
Tag009	Short	D9	22626	Good (C0h)
Tag010	Short	D10	0	Good (C0h)
Tag011	Short	D11	0	Good (C0h)
Tag012	Short	D12	0	Good (C0h)

4.7.1 Dynamic Tags and Static Tags

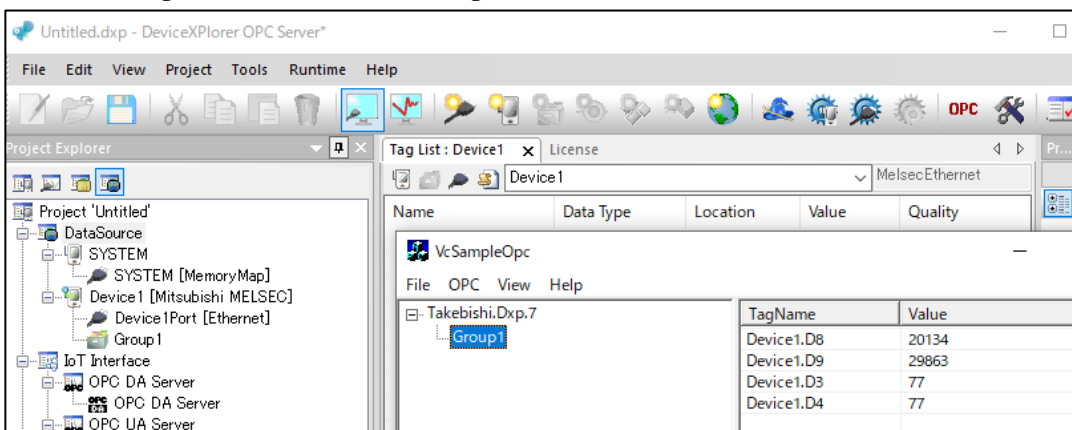
A tag that is defined in advance on DeviceXPlorer is called a “static tag”. Using a static tag allows you to attach a different tag name to the tag, as shown below, and to change the scale conversion settings. You can check the properties of static tags in the Tags tab of the Property View.



Name	Data Type	Location	Value	Quality
Tag003	Short	D3	58	Good (C0h)
Tag004	Short	D4	58	Good (C0h)
Tag008	Short	D8	23633	Good (C0h)
Tag009	Short	D9	11602	Good (C0h)

It is also possible to access data easily without tag definition. Tags that are not defined in advance are called “dynamic tags”. Dynamic tags are accessed by entering the DeviceXPlorer device name plus the PLC register name.

In the below figure, the value from data register “D0” is obtained from the PLC set as “Device1”.



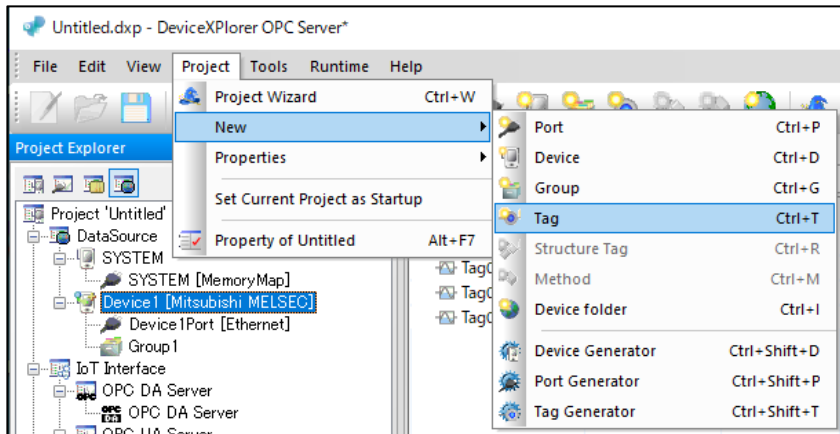
Name	Data Type	Location	Value	Quality
VcSampleOpc				
Takebishi.Dxp.7				
Group1				
Device1.D8			20134	
Device1.D9			29863	
Device1.D3			77	
Device1.D4			77	

Important

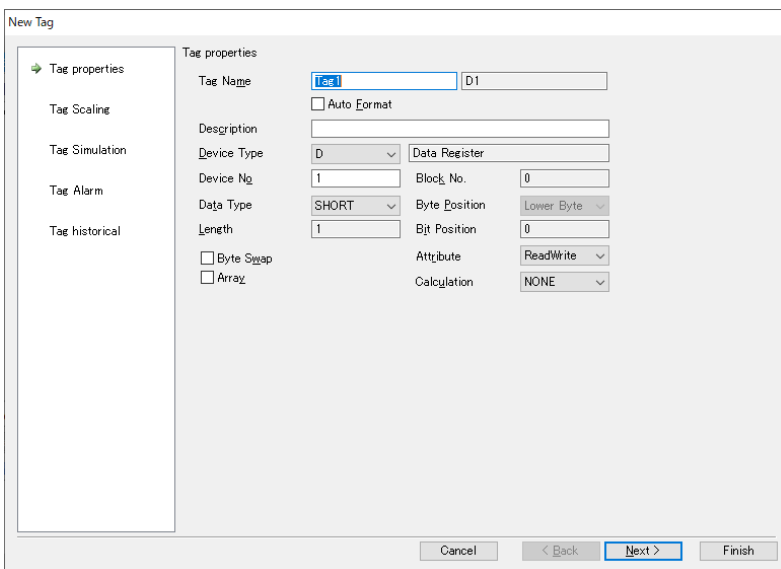
- DeviceXPlorer has no restriction on the number of tags, so it is necessary to ensure that the CPU performance and memory are high enough to accommodate the system (number of tags) you are using. If the number of PLCs is more than 50 or if the total number of tags is over 100,000, we recommend that you distribute the load to the DeviceXPlorer across several PCs.
- Dynamic tags are not supported via OPC UA communication.

4.7.2 Creating a New Tag

From the Projects menu, select New -> Tag.

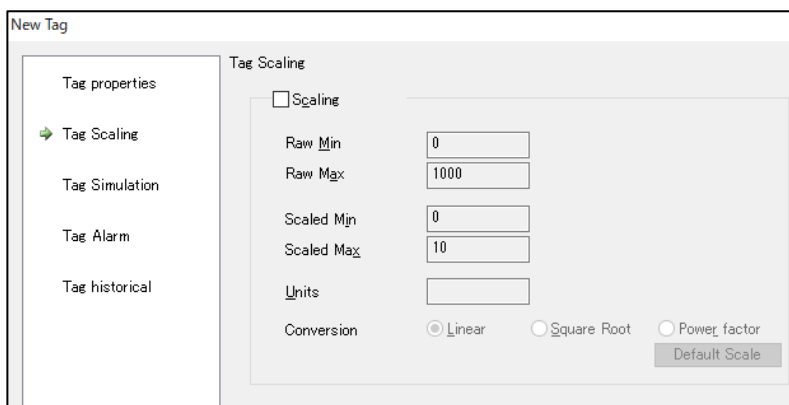


Set the tag parameters and click “Next”.

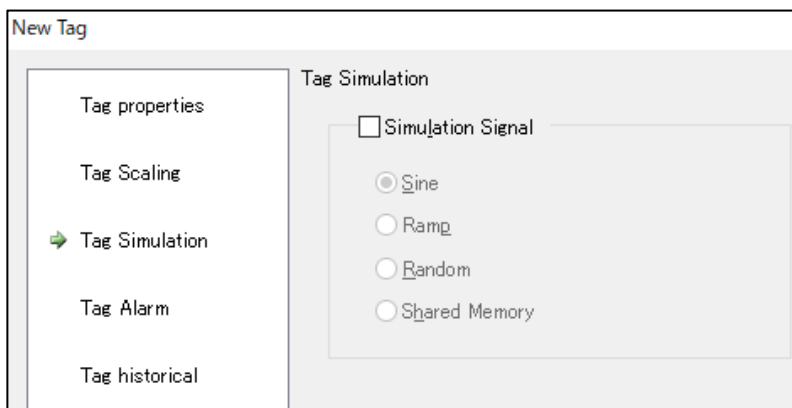


To perform scale conversion, place a check in the box for “Scale Conversion”, “Simulation” and “Alarm” “set the parameters.

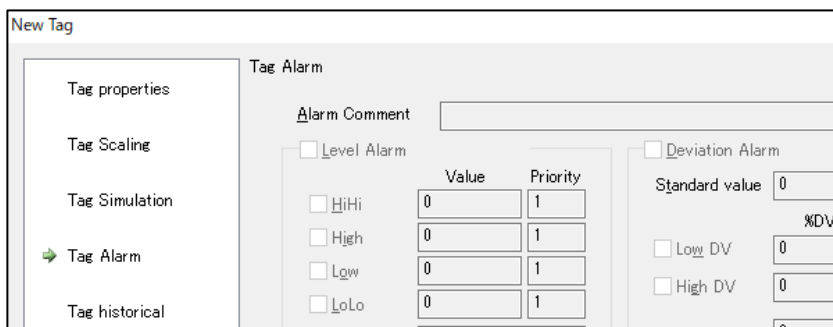
< Scale Conversion>



<Simulation>

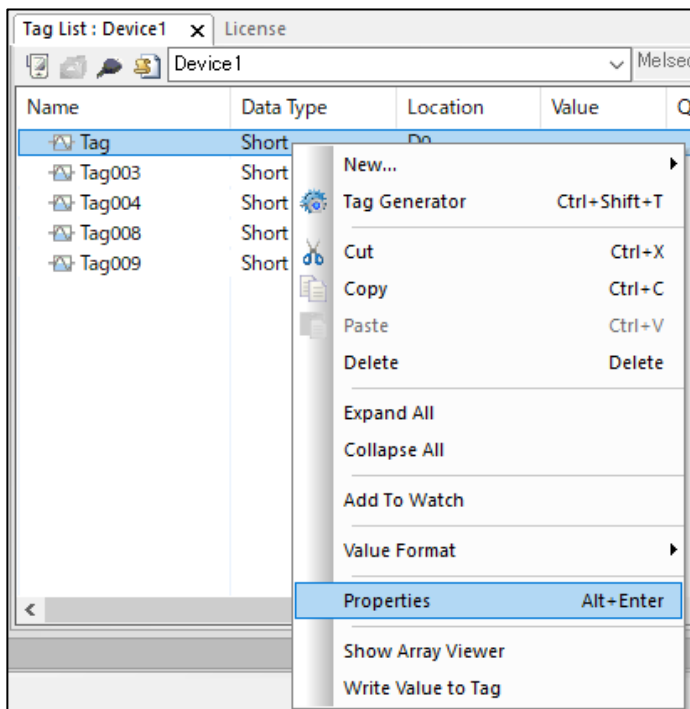


<Alarm>



4.7.3 Changing a Tag Name

Select the tag name you want to change from the Tag List and open the Properties screen from the pop-up menu.



This will allow you to change the set tag name to a name of your choice.

Property of Tag [D0]

General Scale Simulation Alarm Historical

Tag Name

☐ Auto Format

Description

Device Type Data Register

Device No. Block No.

Data Type Byte Position

Length Bit Position

☐ Byte Swap Attribute

☐ Array Calculation

New Save << >> OK Cancel

Note

It is necessary to set a name that is not duplicated.

4.7.4 Properties (General)

This tab allows you to set properties such as the tag name and data format.

The dialog settings for MELSEC Ethernet are as shown in the table below.

The settings vary for each PLC; for details, see the User's Guide (PLC Communication Edition).

Item	Description
Tag name	Specify a tag name if you want to attach a different name for the PLC device. The access format for dynamic tags is shown to the right of the tag name.
Automatic format	Automatically generates a tag name.
Comment	Allows you to add a comment to the tag name.
Device type	Specify a type of the PLC device to be accessed.
Device number	Specify the device number. Depending on the device type, the device number is entered in either decimal or hexadecimal notation.
Block number	Specify a block number if you have specified extended file register (ER) as the device type.
Data type	Select a data type. BOOL: Logical BYTE: 8-bit signed integers UBYTE: 8-bit unsigned integers SHORT: 16-bit signed integers USHORT: 16-bit unsigned integers LONG: 32-bit signed integers ULONG: 32-bit unsigned integers FLOAT: 32-bit real numbers LONGLONG: 64-bit signed integers ULONGLONG: 64-bit unsigned integers DOUBLE: 64-bit real numbers STRING: Character string
Size	Specify the size if you have specified a character string or an array. Character string: If "1" is specified, the space for one word is occupied. Array (BOOL): If "1" is specified, the space for one word (16 bits) is occupied. Array (BYTE): If "1" is specified, the space for one word (2 bytes) is occupied. Array (SHORT): If "1" is specified, the space for one word is occupied. Array (LONG / FLOAT): If "1" is specified, the space for two words is occupied. Array (LONGLONG / DOUBLE): If "1" is specified, the space for four words is occupied.

Item	Description
Byte position	If the BYTE type is selected, select either high or low byte.
Bit position	If you specified BOOL as the data type, specify the bit position.
Attribution	Specify a tag attribution. ReadWrite: Reading and writing possible ReadOnly: Read only WriteOnly: Write only
System calculation	[BCD calculation] If the data type is BYTE, SHORT, or LONG, you can perform BCD calculation. [Binary/octal/decimal/hexadecimal text type] If the data type is STRING, you can select any format.
Byte swapping	To swap the high byte for the low byte, place a check in this box. This is enabled for the SHORT or LONG data types.
Array	Place a check in this box to define the tag as an array.

4.7.5 Properties (Scale Settings)

Set the scale settings to convert values received from the PLC to engineering values.

Item	Description
Enable scale conversion	Place a check in this box to implement scale conversion processing.
Pre-conversion – Min. (M)	Enter the minimum value of the value range that can be obtained from the PLC.
Pre-conversion – Max. (A)	Enter the maximum value of the value range that can be obtained from the PLC.
Post-conversion – Min. (M)	Enter the minimum effective value.
Post-conversion – Max. (A)	Enter the maximum effective value.
Engineering unit	Specify this if you are setting a unit.
Conversion method	Specify “linear” or “square root” or “Power factor” as the conversion method.

Note

- The OPC DA dead band function is effective for tags for which scale conversion is set.
- If the values obtained from the PLC exceed the pre-conversion range, the quality flag will be 0x50 (Engineering Unit Exceeded).
- The tag's data type before conversion is ignored and it becomes 64 bit floating point type (DOUBLE type).

4.7.6 Properties (Simulation)

Enable simulation settings if you want to return values of your choice without communicating with the PLC.

The screenshot shows a dialog box titled 'Property of Tag [D0]'. It has five tabs: 'General', 'Scale', 'Simulation', 'Alarm', and 'Historical'. The 'Simulation' tab is selected. Inside the dialog, there is a checkbox labeled 'Simulation Signal' which is checked. Below this checkbox, there are four radio button options: 'Sine' (which is selected), 'Ramp', 'Random', and 'Shared Memory'. At the bottom of the dialog, there are buttons for 'New', 'Save', '<<', '>>', 'OK', and 'Cancel'. The 'OK' button is highlighted with a blue border.

Item	Description
Enable simulation	Perform simulation by tag.
Sine	Returns values that make a sine curve.
Ramp	Returns values that increase in increments of 1.
Random	Returns random values.
Shared memory	Returns static values. Values written from the client are maintained.

4.7.7 Properties (Alarm)

With the value acquired from PLC with an OPC A&E interface, alarm and when carrying out event surveillance, an alarm setup is performed.

Property of Tag [D0]

General Scale Simulation **Alarm** Historical

Alarm Comment: Ararm1

☒ Level Alarm

	Value	Priority
☒ HiHi	90	1
☒ High	80	1
☒ Low	40	1
☒ LoLo	20	1
Deadband	0	

☒ Ratio Alarm

Change: 10 % / sec

Priority: 1

☒ Deviation Alarm

Standard value: 0

	%DV	Priority
☐ Low DV	0	1
☐ High DV	0	1
Deadband	0 %	

☐ Discrete Alarm

Event Type: Simple Event

Occurrence: Cause by On(TRUE)

Priority: 1

New Save << >> OK Cancel

Item		Description
Alarm Comment		Specify the Alarm comment.
Level Alarm	LoLo/Low/High/HiHi	Select Enable or Disable.
	Value	Specify the threshold value.
	Priority	Specify the priority.
	Deadband	Set the deadband of level Alarm. If a current value returns more than deadband, the alarm restores.
Ratio Alarm	Change	Set the change ratio (percentage) per standard time.
	Unit	Select the time unit "Sec" or "Min" or "Hour"
	Priority	Specify the priority.
Deviation Alarm	Standard value	Specify the standard value.
	Low DV	Specify the deviations from Standard value.
	High DV	Specify the deviations from Standard value.
	Deadband	Set the deadband of Deviation Alarm. If a current value returns more than deadband, the alarm restores.
Discrete Alarm	Event Type	This alarm can be defined only in BOOL tag. Select "Single Event" or "Tracking Event" or "Condition Event".
	Occurrence	Select the alarm occurrence, "On" or "Off"
	Priority	Specify the priority.

Note

- To configure Level alarm, Ratio Alarm and Deviation Alarm, the Tag's scale conversion should be defined.
- To configure Discrete alarm, the Tag can be defined as BOOL.

4.7.8 Properties (Historical)

This setting is used when the historical function is used with the OPC UA interface. To use the historical function, enable the setting.

The screenshot shows a dialog box titled "Property of Tag [D0]". It has five tabs: "General", "Scale", "Simulation", "Alarm", and "Historical". The "Historical" tab is selected. Inside the dialog, there are two radio buttons: "Enable" (which is selected) and "Disable". Below these are two dropdown menus. The first is labeled "Historical group" and has "<No Selection>" selected. The second is labeled "Attribute" and has "Read/Update" selected. At the bottom of the dialog, there are five buttons: "New", "Save", "<<", ">>", and "OK" (which is highlighted with a blue border), and "Cancel".

Item	Description
Historical group	Select the group to which the tag belongs from the historical groups you have set up.
Attribute	Set the ReadOnly/Update permission attribute of the value.

*Refer to 4.13.11 for historical function settings.

4.7.9 Saving Settings

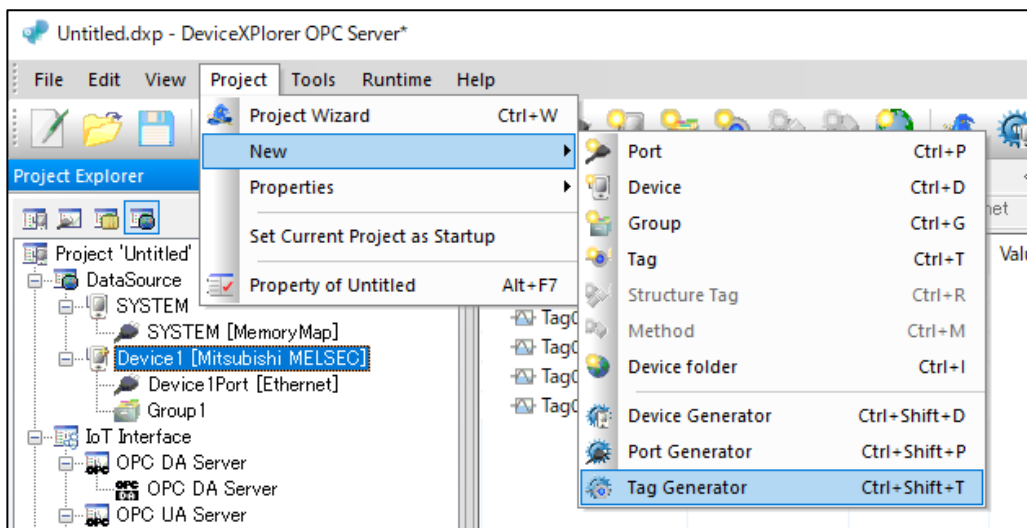
Save settings by clicking the “Save” button or clicking the “OK” button and closing the dialog.

4.7.10 Adding Tags Continuously

If you click the “Save” button and then the “New” button in succession, the device number will be automatically incremented.

4.7.11 Tag Generator

Using the tag generator allows you to generate tags of a particular format at once.

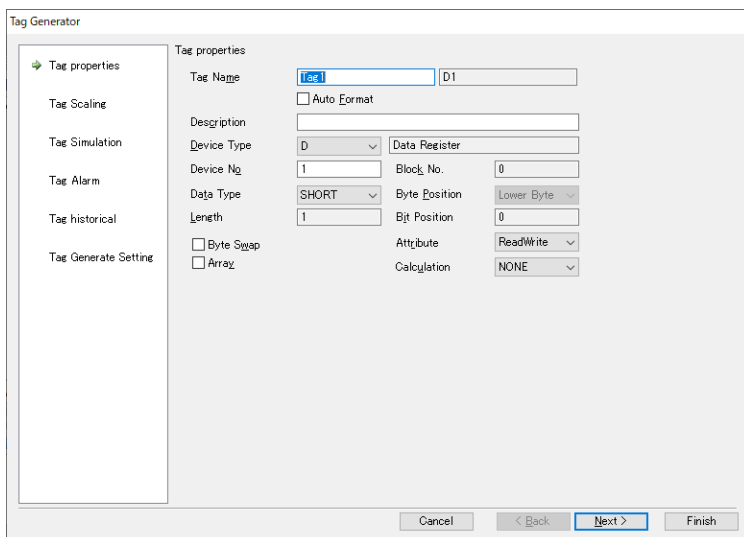


Note

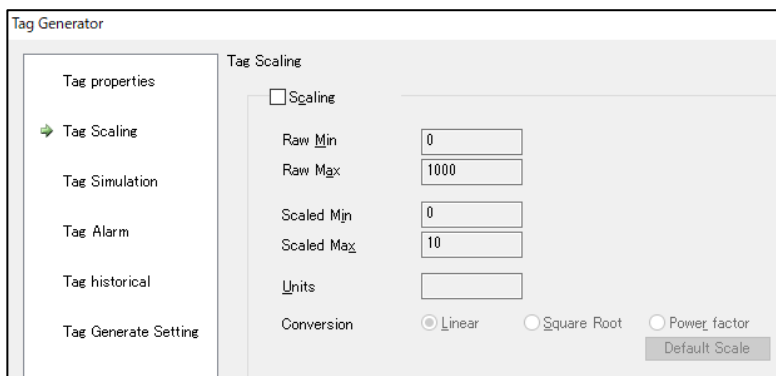
The number included in the tag name and the device number are automatically incremented.

[Operation]

(1) Set the base tag information.



(2) To perform scale conversion, place a check in the box for “Scale Conversion”, “Simulation” and “Alarm” “set the parameters.



(3) Set the number of tags to be generated at once and the increase step.

Item	Default	Description
Start Location	(Base Tag Name)	The base tag name is shown.
Generate Count	10	Specify the count which will be generated.
Increase step	1	Specify the increment by which to increase the device number.
Ignore Data Size	OFF	Ignores the data type and increases the device number in increments of 1.
Digit	9	Specify the digit of auto-generated tag name in Zero Supply.
Zero Supply	OFF	Please check if the tag name need zero supply.

Note

If an automatically generated tag name already exists, a serial number is attached after the tag name.

Name	Data Type	Location
Tag001	Short	D1
Tag002	Short	D2
Tag003	Short	D3
Tag003 (1)	Short	D3
Tag004	Short	D4
Tag005	Short	D5
Tag006	Short	D6
Tag007	Short	D7
Tag008	Short	D8

For example, if you generated 10 tags from Tag001, and Tag003 already exists, the generated tag will become "Tag003 (1)" as shown above.

4.7.12 Extended Tag Name Specification

If you are not registering tag information in advance but are registering and accessing items dynamically, specify the device name and data type as extended specification. For extended specification, follow the rules listed below.

Extended specification	Description	Note
No Specification	Bit device: Logical type	
	Word device: 16-bit signed integers	
:00::15	Logical	Cannot specify for bit devices.
:L	8-bit signed integers	
:LU	8-bit unsigned integers	
:H	8-bit signed integers	
:HU	8-bit unsigned integers	
:W	16-bit signed integers	
:WU	16-bit unsigned integers	
:D	32-bit signed integers	
:DU	32-bit unsigned integers	
:R	32-bit real numbers	
:I	64-bit signed integers	
:IU	64-bit unsigned integers	
:DR	64-bit real numbers	
:B	Integer BCD conversion	
:OR	Read-only	
:OW	Write-only	
:A99999	Array	
:S99999	Character string	D1.S5 = "ABCDEFGHJIJ"
:F99999	File Batch Write	D1:F5="C:\test.txt"
:B99999	Binary text	D1.B2 = "11111111111111110000111100001111"
:O99999	Octal text	D1.O5 = "1,1234,77777,0,177777"
:D99999	Decimal text	D1.D5 = "1,2,32767,-32768,0"
:H99999	Hexadecimal text	D1.H5 = "1234FFFF99990000ABCD"
:V	Bit inverse	
:BL	Boolean conversion	Only some models correspond.
:SS	Simulation mode(Sin)	
:SD	Simulation mode(Shared Memory)	
:SI	Simulation mode(Ramp)	
:SR	Simulation mode(Random)	

Note

To use a combination of more than one extended specification, you can use a colon as a separator.
For example, a write-only 32-bit unsigned integer array would be specified as "D1:DU:OW:A100".

4.7.12.1 No Specification

If you are not using extended specification, the standard data type is the 16-bit signed integer for word devices and the logical type for bit devices.

Example: For MELSEC, the No. 1000 data register is “D1000”, and the link relay No. 1F is “B1F”.

4.7.12.2 Specifying the Bit Position (:nn)

If you specify a separator and bit position at the end of a tag name, it can be handled as if it were the logical type.

Example: The 0th bit of D1 is “D1:0” and the 15th bit of D1 is “D1:15”.

Note
Only certain PLCs can write to these tags.

4.7.12.3 Byte Type (:L, :LU, :H, :HU)

By specifying a low byte and a high byte, you can treat the register value as an 8-bit signed or unsigned integer.

Example: In “D1:H”, the higher 8 bits of D1 are accessed as an integer.

4.7.12.4 Word Type (:W, :U, :WU)

The register value is treated as a 16-bit signed or unsigned integer.

Example: In “D1:WU”, D1 is accessed as an unsigned integer.

4.7.12.5 Long Type (:D, :DU)

The register value is treated as a 32-bit signed or unsigned integer.

Example: In “D1:D”, D1 and D2 are accessed as a 32-bit signed integer.

4.7.12.6 LongLong Type (:L, :LU)

The register value is treated as a 64-bit signed or unsigned integer.

Example: In “D1:L”, D1 – D4 are accessed as a 64-bit signed integer.

4.7.12.7 Single Precision Real Number Type (:R)

The register value is treated as a 32-bit real number.

Example: If the two words of D1 and D2 contain “0x42F6CCCD”, “D1:R” is read as “123.4”.

4.7.12.8 Double Precision Real Number Type (:DR)

The register value is treated as a 64-bit real number.

Example: If the two words of D1 – D4 contain “0x40FE240C9FBE76C9”, “D1:DR” is read as “123456.789”.

4.7.12.9 BCD Type (:B)

The register value (BCD value) is converted to BIN value when it is read, and when it is written, the written value (BIN value) is converted to BCD value. If BIN conversion is not possible (when reading is not possible), the quality flag will be “0x54” (Engineering Units Exceeded), and the value will be undefined. If BCD conversion is not possible (when writing is not possible), an error will be returned and the value will not be written.

Example: If 1234 (hexadecimal) is contained in D1 and “D1:B” is read, “1234 (decimal)” will be returned.

Example: If “5678 (decimal) is” written to “D1.B”, “5678 (hexadecimal)” will be written.

4.7.12.10 Read-only/Write-only (:OR, :OW)

The read-only tag attribution can be used to prevent inadvertent writing. In addition, if there are many tags that do not require reading, you can use the write-only tag to improve performance.

Example: “D1:OR” allows read-only accesses to D1.

4.7.12.11 Array (:A99999)

If you add a separator and specify the number of elements after the tag name, the tag will be treated as an array-type tag. Array-type tags perform batch-access to a PLC, which optimizes the communication time.

Example: D1:A10 accesses D1 - D10 as a 16-bit integer array with 10 elements.

Example: D1:D:A64 accesses D1 - D64 as a 32-bit integer array with 64 elements.

Note

- If the number of elements exceeds the batch R/W points, the synchronicity of the data will not be maintained.
- The maximum number of elements is 65535.

4.7.12.12 String (:S99999)

If you specify “:S99999” (with 99999 being the number of registers to access), the register value is treated as a character string.

Example: If D0=0x4241, D1=0x4443, D2=0x4645, D3=0x4847 and D4=0x4A49, if “D0:S5” is accessed, the character string “ABCDEFGHJ” is obtained.

Note

- If the number of elements exceeds the batch R/W points, the synchronicity of the data will not be maintained.
- The maximum number of elements is 65535.

4.7.12.13 File Batch Write(:F99999)

By specifying “:F99999” (9999 is between 1 and 65535) behind a tag name, the contents of the text file can be written in at once.

You create CSV (comma separated form) file , and writes the path (full path) of a text file in a tag at the time of writing. If it reads, the written-in file path is acquirable.

Example:

You write CSV file path (full path) to “D0:F5”, DeviceXPlorer write values between D0 and D4 by the head of the data of the CSV file.

4.7.12.14 Text Format Binary Type (:B99999)

If you specify “:B99999” after the tag name, the register value can be treated as a text format binary type. When requesting writing, if the value to be written contains something other than numbers (0 - 1), that value and subsequent values will be ignored.

Example: If D0=0xF0F0, accessing “D0:B1” obtains the character string “1111000011110000”.

Example: If “111a0000” is written to “D0:B1”, “111” is written.

Note

- If the number of elements exceeds the batch R/W points, the synchronicity of the data will not be maintained.
- The maximum number of elements is 65535.

4.7.12.15 Text Format Octal Type (:O99999)

If you specify “O99999” after the tag name, the register value can be treated as a text format octal type. When requesting writing, if the value to be written contains something other than numbers (0 - 7), that value and subsequent values will be ignored.

Example: If D0=123, D1=99, and D2=1, accessing “D0:O3” obtains the character string “173,143,1”.

Example: If “135790” is written to “D0:O1”, “1357” is written.

Note

- If the number of elements exceeds the batch R/W points, the synchronicity of the data will not be maintained.
- The maximum number of elements is 65535.

4.7.12.16 Text Format Decimal Type (:D99999)

If you specify “D99999” after the tag name, the register value can be treated as a text format decimal type. When requesting writing, if the value to be written contains something other than numbers (0 - 9), that value and subsequent values will be written as “0”.

Example: If D0=123, D1=99, and D2=1, accessing “D0:D3” obtains the character string “123,99,1”.

Example: If “9,a,8” is written to “D0:D3”, the value is written as D0=9, D1=0, and D2=0.

Note

- If the number of elements exceeds the batch R/W points, the synchronicity of the data will not be maintained.
- The maximum number of elements is 65535.

4.7.12.17 Text Format Hexadecimal Type (:H99999)

If you specify “H99999” after the tag name, the register value can be treated as a text format hexadecimal type. When requesting writing, if the value to be written contains something other than numbers (0 - 9), that value and subsequent values will be written as “0”.

Example: If D0=0x123 and D1=0x10, accessing “D0:H2” obtains the character string “01230010”.

Example: If “98765X” is written to “D0:H3”, the value is written as D0=0x9876, D1=5, and D2=0.

Note

- If the number of elements exceeds the batch R/W points, the synchronicity of the data will not be maintained.
- The maximum number of elements is 65535.

The table below shows the scale conversion matrix for each text format.

Binary	Octal	Decimal	Decimal (unsigned)	Hexadecimal
0	0	0	0	0
11111111	377	255	255	FF
111111111111	7777	4095	4095	FFF
0111111111111111	77777	32767	32767	7FFF
1000000000000000	100000	-32768	32768	8000
1111111111111111	177777	-1	65535	FFFF

4.7.12.18 Bit Inverse

If you specify “V” after the tag name, DeviceXPlorer inverse Boolean value.

Example: If M0=”False”, accessing “M0:V” obtains the “True”.

4.7.12.19 Boolean conversion

If you specify “BL” after the tag name, DeviceXPlorer converts to Boolean .
If value is 0, the tag is False(Off). Otherwise if value is not 0, the tag is True(On).
Example: If Field1 is set value 123, "Field1:BL" is True(On).
Example: If Field1 is set value 0, "Field1:BL" is False(Off).

4.7.12.20 Simulation mode (Sin)

If you specify “SS” after the tag name, DeviceXPlorer treats the Tag as a simulation(Sin) tag.

4.7.12.21 Simulation mode (Shared Memory)

If you specify “SD” after the tag name, DeviceXPlorer treats the Tag as a simulation(Shared memory) tag.

4.7.12.22 Simulation mode (Ramp)

If you specify “SI” after the tag name, DeviceXPlorer treats the Tag as a simulation(Ramp) tag.

4.7.12.23 Simulation mode (Random)

If you specify “SR” after the tag name, DeviceXPlorer treats the Tag as a simulation(Random) tag.

4.7.13 System Tags

4.7.13.1 Global System Tags

The following system tags are valid for DeviceXPlorer as a whole. These system tags are in SYSTEM device.
(Access example: SYSTEM.\$Clock1000, SYSTEM.\$TagCount)

Name	Tag name	Type	Attribu tion	Note
Version information	\$Version	VT_BSTR	R	Ex: "Version 7.0.0.1"
License status	\$License	VT_BOOL	R	License Activated: FALSE DEMO or STOP mode : TRUE
License mode	\$LicenseMode	VT_BSTR	R	Show current license mode
License edition	\$LicenseEdition	VT_BSTR	R	Show current license edition
Clock (100 ms)	\$Clock100	VT_BOOL	R	Repeats 50 ms on followed by 50 ms off.
Clock (200 ms)	\$Clock200	VT_BOOL	R	Repeats 100 ms on followed by 100 ms off.
Clock (500 ms)	\$Clock500	VT_BOOL	R	Repeats 250 ms on followed by 250 ms off.
Clock (1000 ms)	\$Clock1000	VT_BOOL	R	Repeats 500 ms on followed by 500 ms off.
Clock (2000 ms)	\$Clock2000	VT_BOOL	R	Repeats 1 second on followed by 1 second off.
Clock (1 min.)	\$ClockMin	VT_BOOL	R	Repeats 30 seconds on followed by 30 seconds off.
Clock (1 hour)	\$ClockHour	VT_BOOL	R	Repeats 30 minutes on followed by 30 minutes off.
Heartbeat	\$HeartBeat	VT_UI4	R	Ring counter that increments every second (can check that server is operating)
Always ON	\$On	VT_BOOL	R	
Always OFF	\$Off	VT_BOOL	R	
Self-diagnosis error	\$Diagnosis	VT_BOOL	R	On while error is occurring
Self-diagnosis error code	\$DiagCode	VT_UI4	R/W	The last error code that occurred.
Self-diagnosis sub error code	\$DiagSubCode	VT_UI4	R/W	The last detail error code that occurred.
Clock data (hour)	\$Hour	VT_UI4	R	Stores the PC's time data
Clock data (minute)	\$Min	VT_UI4	R	
Clock data (second)	\$Sec	VT_UI4	R	
Clock data (millisecond)	\$Msec	VT_UI4	R	
Date data (year)	\$Year	VT_UI4	R	
Date data (month)	\$Month	VT_UI4	R	
Date data (day)	\$Day	VT_UI4	R	
Log save implementation flag	\$SaveLog	VT_BOOL	R/W	If TRUE is written, the log message is saved, and it is changed to FALSE after the request is received.
Log saving file name	\$LogFile	VT_BSTR	R/W	Can specify both absolute and relative paths. If there is no specification (NULL), it is saved in the YYYYMMDDHHMMSS.CSV format.

Name	Tag name	Type	Attribution	Note
Script Log flag	\$DumpScript	VT_BOOL	R/W	Corresponds to View -> Script
Sent/received trace flag	\$DumpCom	VT_BOOL	R/W	Corresponds to View -> Communication data
DxpLink trace flag	\$DumpDxpLink	VT_BOOL	R/W	Corresponds to View -> DxpLink
SuiteLink trace flag	\$DumpDDE	VT_BOOL	R/W	Corresponds to View -> SuiteLink
OPC access log flag	\$DumpOpc	VT_BOOL	R/W	Corresponds to View -> OPC Access
Trace type flag (INFO)	\$TraceInfo	VT_BOOL	R/W	Outputs an INFO message
Trace type flag (WARNING)	\$TraceWarning	VT_BOOL	R/W	Outputs a WARNING message
Trace type flag (ERROR)	\$TraceError	VT_BOOL	R/W	Outputs an ERROR message
Trace type flag (FATAL)	\$TraceFatalError	VT_BOOL	R/W	Outputs an ERROR message
Connected clients	\$ClientCount	VT_UI4	R	Number of connected clients
Client errors	\$ClientErrors	VT_UI4	R	Number of clients experiencing errors
Connected groups	\$GroupCount	VT_UI4	R	Number of OPC groups and SuiteLink topics that are being accessed from outside
Group errors	\$GroupErrors	VT_UI4	R	Number of groups experiencing errors
Total tags	\$TagCount	VT_UI4	R	Total number of active tags
Tag errors	\$TagErrors	VT_UI4	R	Number of tags experiencing errors
Port Parameter	\$PortParameter	VT_BSTR	R/W	Dynamic Port parameter.
Device Parameter	\$DeviceParameter	VT_BSTR	R/W	Dynamic Device parameter.
Reset Statistical Information	\$ResetStatisticalTags	VT_BOOL	R/W	By writing TRUE, DeviceXplorer resets the following system tags. \$ElapseMin,\$ElapseMax,\$Elapse,\$ErrorCounts,\$SendCount,\$ReceivedCounts

4.7.13.2 Device System Tags

The following system tags are valid for individual devices. These system tags are in each device.
(Access example: Device1.\$Status, Device2.\$Simulate)

Name	Tag name	Type	Attri butio n	Note
Communication status	\$Status	VT_BOOL	R/W	Communicating: TRUE, Stopped: FALSE In simulation mode, it is free to change the value of this tag.
Message/communication errors	\$ErrorCounts	VT_UI4	R	Number of errors that have occurred
Messages sent	\$SendCounts	VT_UI4	R	Number of messages sent
Messages received	\$ReceivedCounts	VT_UI4	R	Number of messages received
Max. number of batch R/W of bit device	\$BitRWRange	VT_UI4	R	Maximum number of batch R/W of bit device
Max. number of batch R/W of word device	\$WordRWRange	VT_UI4	R	Maximum number of batch R/W of word device
	\$ReadRange	VT_UI4	R	
Max. number of points for random write of a bit device	\$BitWRange	VT_UI4	R	Maximum number of points for random write of a bit device
Max. number of points for random write of a word device	\$WordWRange	VT_UI4	R	Maximum number of points for random write of a word device
Redundant communication standby status	\$Standby	VT_BOOL	R/W	Active system when redundant communication is used Communicating on primary device: FALSE Communicating on secondary device: TRUE
Simulation status	\$Simulate	VT_BOOL	R/W	Running in simulation mode: TRUE Communicating with PLC: FALSE
Communication port name	\$PortName	VT_BSTR	R	Port name
Lap time (current)	\$Elapse	VT_UI4	R	The time required for the communication. If more than one OPC group is accessing the device at once, the lap time is set for each group.
Lap time (minimum)	\$ElapseMin	VT_UI4	R	
Lap time (maximum)	\$ElapseMax	VT_UI4	R	
Communication load rate (%)	\$ElapseRate	VT_UI4	R	Current lap time ÷ update interval x 100 (%)
Update Rate	\$DeviceUpdateRate	VT_UI4	R/W	Update rate for defined in Device Option.
Simulation type	\$SimulationType	VT_UI4	R/W	Simulation mode 0:Sin 1:Ramp 2:Random 3:Shared Memroy
Slow Polling Mode	\$SlowpollingMode	VT_BOOL	R/W	TRUE:Slow Polling FALSE:Normal
Connection Test	\$ConnectionTest	VT_BOOL	R	Execute "Connection Test". Success : TRUE Failed : FALSE
Primary device Communication	\$PrimaryStatus	VT_BOOL	R/W	Primary device Communicating: TRUE, Stopped: FALSE

Status				
Standby device Communication Status	\$SecondaryStatus	VT_BOOL	R/W	Standby device Communicating: TRUE, Stopped: FALSE

4.7.13.3 Connection System Tags

The following system tags are valid for individual connections. These system tags are in each connection. These system tags' value is different per connection. (Access example: SYSTEM.\$UpdateInterval, SYSTEM.\$Elapse)

Name	Tag name	Type	Attri butio n	Note
Update interval	\$UpdateInterval	VT_UI4	R/W	The current update interval. If you specify "0", communication will stop.
Error code	\$ErrorCode	VT_UI4	R	The last error code that occurred.
Error text	\$ErrorText	VT_BSTR	R	The text of the last error that occurred.
Message/communication errors	\$ErrorCount	VT_UI4	R	Number of errors that have occurred
Messages sent	\$SendCount	VT_UI4	R	Number of messages sent
Messages received	\$ReceivedCount	VT_UI4	R	Number of messages received
Write status	\$WriteComplete	VT_BOOL	R	"0" when a write request is received and "1" when write processing is finished (successfully completed/failed).
Read status	\$ReadComplete	VT_BOOL	R/W	"1" when all active tags are read, returning to "0" when a tag is added, and then becoming "1" again when that tag is read. When "0" is written from a client, it becomes "1" again after all reading is complete.
Registered tags	\$ItemCount	VT_UI4	R	Number of registered tags
Number of active tags	\$ActiveItemCount	VT_UI4	R	Number of active tags
Number of error tags	\$ErrorItemCount	VT_UI4	R	Tags with quality flags other than "good".
Messages	\$MessageCount	VT_UI4	R	Total number of read/write messages
Read requests	\$PollNow	VT_BOOL	R/W	Refreshes (rereads) all tags as soon as "1" is written. Valid even when "0" is set as the update interval.
Lap time (current)	\$Elapse	VT_UI4	R	The time required for the communication
Lap time (minimum)	\$ElapseMin	VT_UI4	R	
Lap time (maximum)	\$ElapseMax	VT_UI4	R	
Communication load rate (%)	\$ElapseRate	VT_UI4	R	Current lap time ÷ update interval x 100 (%)

Note

In the time of client unit communication invalidity (default configuration), only the following 4 connection tags can be available.

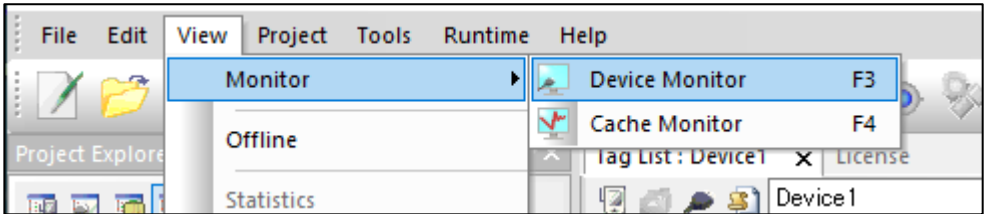
- \$UpdateInterval
- \$ItemCount
- \$ActiveItemCount
- \$ErrorItemCount

4.7.14 Tag Monitor

The current value of static tags can be monitored from the DeviceXPlorer screen regardless of whether or not there is access from a client.

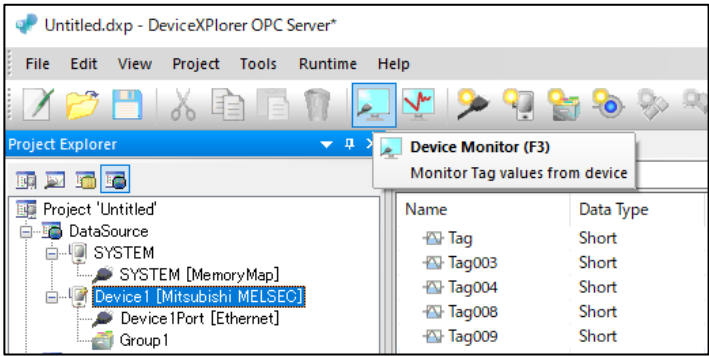
[Operation]

Select “Device Monitor” or “Cache Monitor” from the View menu.



Item	Description
Device monitor	Communicates with the PLC and monitors live data.
Cache monitor	Does not communicate with the PLC but monitors cached values in DeviceXPlorer. The values of only the tags currently accessed by clients are updated.

When tag monitoring is performed, the current values and information such as the quality flags appear on the screen.



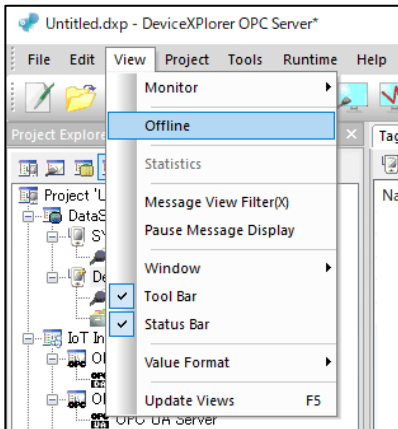
Note

The update interval when the device monitor function is used is value of Monitor Rate in “Common Properties, General”.

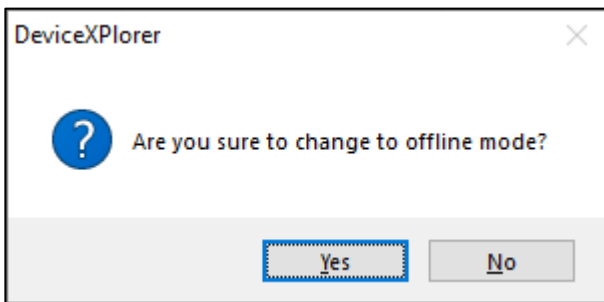
4.7.15 Offline Mode

If DeviceXPlorer shifts to off-line mode, it will stop performing communication with target modules entirely, and will process with the cache value which it has in the inside.

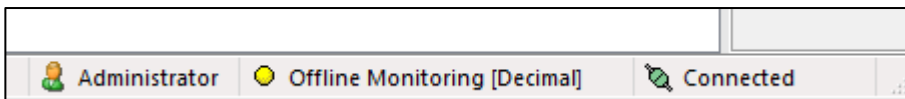
If DeviceXPlorer shifts to this mode, the processing to a client etc. will also be affected.



A confirmation message is displayed when shifting to off-line mode.



During Offline Mode execution, it is displayed on a lower right status bar as “Offline Monitoring”.

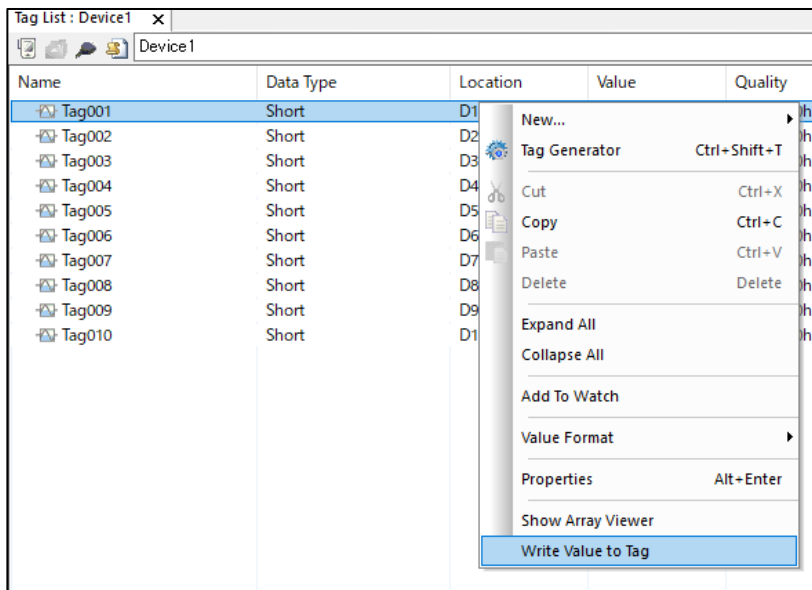


Important

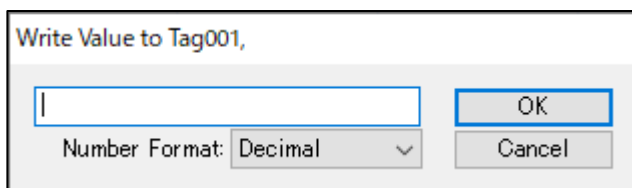
If DeviceXPlorer shifts to Offline Mode, it stops performing communications processing entirely with target modules. When you shift during system running, be careful enough.

4.7.16 Write Tag Value

During tag monitoring, you can write value to a tag.
You select the tag, and select "Write Value to Tag...".



Since the follows dialog is displayed, You input a write value and click "OK". DeviceXPlorer will write the value to target module.

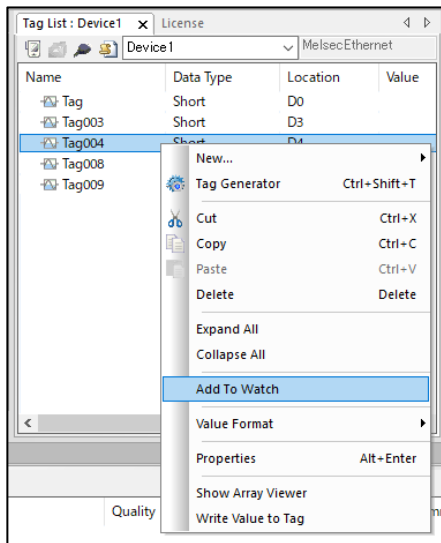


Note

"GUI->Writing Tag value From GUI" needs to be "Permit Writing" of Project Properties.

4.7.17 Watch

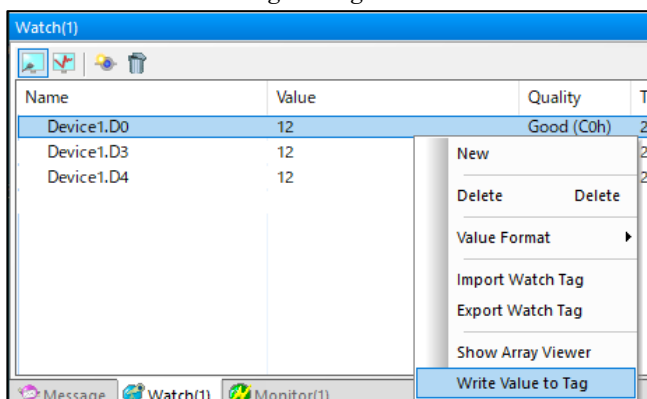
The watch function allows you to dynamically monitor the current values of tags if you drag & drop the tags you want to monitor from the Tag View.



It is also possible to monitor dynamic tags by directly entering the tag names.

Name	Value	Quality	Timestamp	Attribute	Comment
Device1.D0	95	Good (C0h)	2022/11/08 1...:55	Read/Write	
Device1.D3	95	Good (C0h)	2022/11/08 1...:55	Read/Write	
Device1.D4	95	Good (C0h)	2022/11/08 1...:55	Read/Write	

You can also save the tag list registered for the watch function in a file using the pop-up menu.



Note

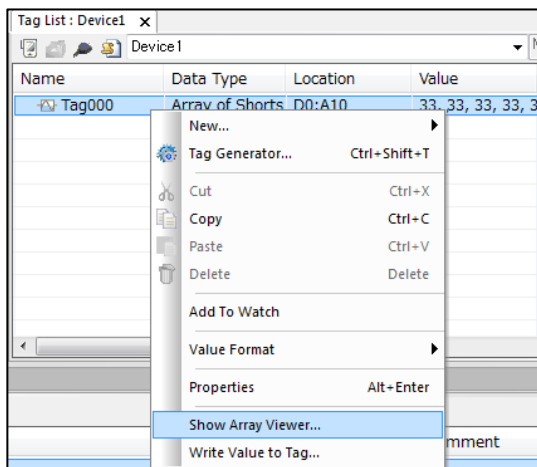
If you register an inappropriate tag, "N/A" will appear.

Name	Value	Quality	Timestamp
Device1.DE	N/A		
Device1.M0:V	Off	Good (C0h)	2012/05/1...

4.7.18 Array Viewer

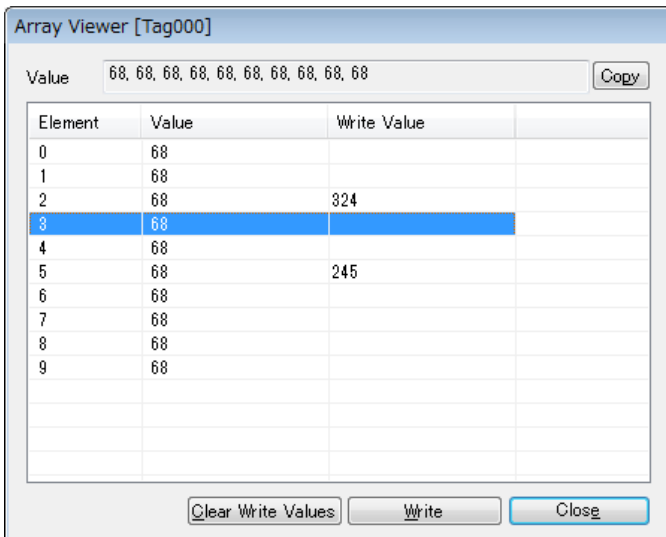
Array Viewer is the function for monitor tag value of array.

Please select “Show Array Viewer” from right-click menu or select tag and push Ctrl + Enter key, so the Array Viewer will show.



You can check the tag value and write value by “Write”.

If you write the value, please change the value and push “Write” button.

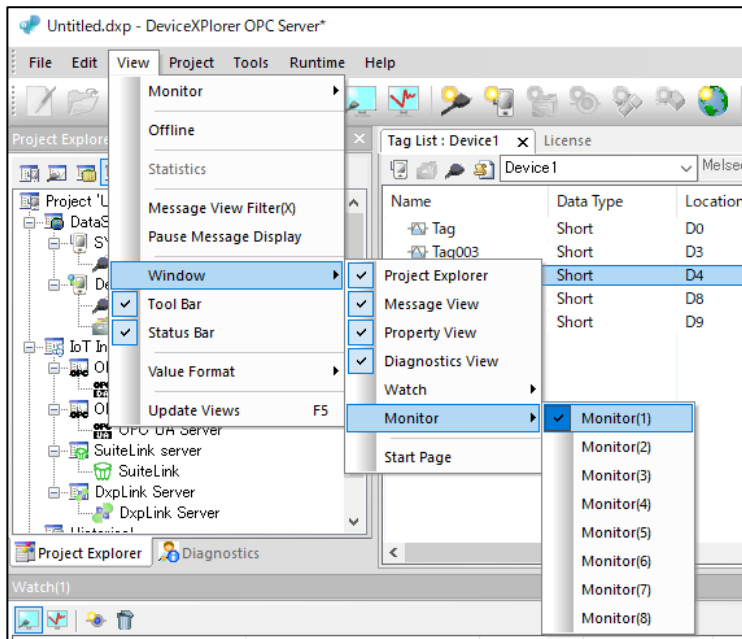


Item	Description
Value / Copy	Show the tag value. Copy button copies the values to clipboard.
Element	The number of the element.
Value	Show the target tag's value.
Write Value	Edit each writing values.
Clear Write Values	Clear all writing values.
Write	Write the value to the target tag.

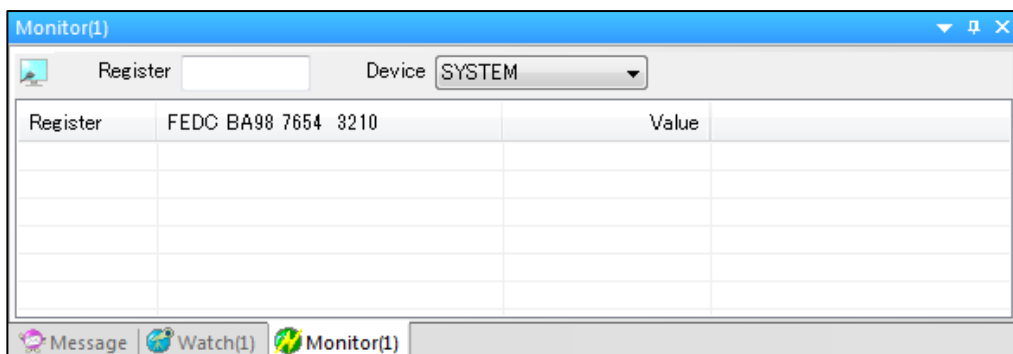
4.7.19 Register Monitor

Register Monitor allows you to monitor the memory values on target module.

Register Monitor can be started from View->Window->Monitor->Monitor(1)



The monitor windows are the following windows.



Item	Description
Device	Specify the target Device
Register	Specify register(memory name) in the target module.

Important

Register Monitor becomes a monitor function for PLC drivers.

The models (for example, ODBC Client, DDE Client etc.) which don't have register area cannot use this function.

You select a device, and input register name, and press Enter Key. Register monitor will start. For example, in MELSEC, if D0 is inputted into a register, it can monitor as follows.

The screenshot shows the software interface with three main windows:

- Project Explorer:** Displays a tree structure of the project, including 'Project 'Untitled'', 'DataSource', 'SYSTEM', 'Device 1 [Mitsubishi MELSEC]', 'Device1Port [Ethernet]', 'Group1', 'IoT Interface', 'OPC DA Server', 'OPC UA Server', 'SuiteLink server', 'SuiteLink', 'DxpLink Server', and 'DxpLink Server'.
- Tag List: Device1:** A table showing tags for 'Device1' with columns 'Name', 'Data Type', 'Location', and 'Value'.

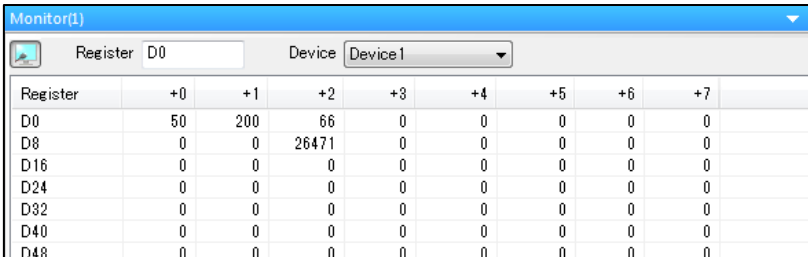
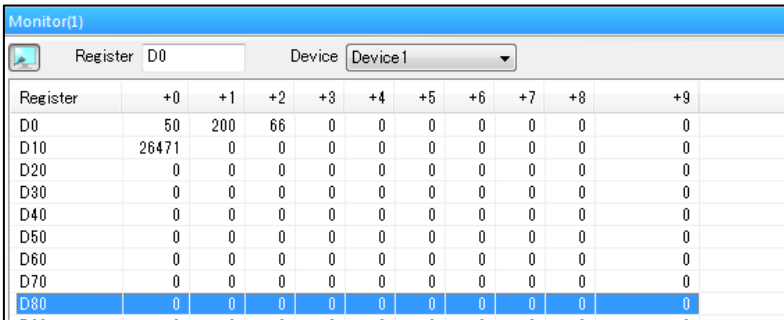
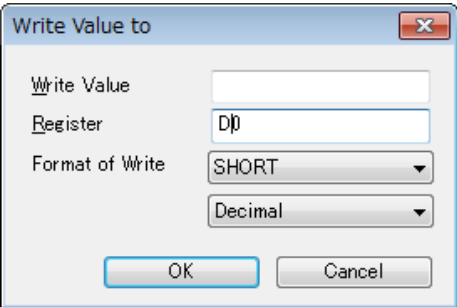
Name	Data Type	Location	Value
Tag	Short	D0	
Tag003	Short	D3	
Tag004	Short	D4	
Tag008	Short	D8	
Tag009	Short	D9	
- Monitor(1):** A window for monitoring registers. It has a 'Register' field set to 'D0' and a 'Device' dropdown set to 'Device1'. Below is a table of registers:

Register	FEDC BA98 7654 3210	Value
D0	0000 0000 0101 1110	94
D1	0000 0000 0101 1110	94
D2	0000 0000 0101 1110	94
D3	0000 0000 0101 1110	94
D4	0000 0000 0101 1110	94
D5	0110 1101 1111 0001	28145
D6	0101 1110 1111 0010	24306
D7	0101 0000 1111 0011	20723
D8	0100 0011 1111 0100	17396
D9	0011 0111 1111 0101	14325
D10	0000 0000 0000 0000	0

The following things can choose a register monitor with a right-click menu.

The screenshot shows the 'Monitor(1)' window with a right-click context menu open over the 'D0' register row. The menu options are:

- Display
 - BYTE(B)
 - UBYTE(Y)
 - SHORT(S)** (selected)
 - USHORT(H)
 - LONG(L)
 - ULONG(O)
 - LONGLONG(N)
 - ULONGLONG(G)
 - FLOAT(F)
- Hexadecimal
- Multi-register(B)
- Ten points back
- Write Value

Item	Description
Display	Select data type for display.
Hexadecimal	Display values in hexadecimal.
Multi-register	Many registers are shown as below. (This is 8 points per line) 
Ten points back	Many registers are shown as below. (This is 10 points per line) 
Write Value	Input a Write Value and write Register. 

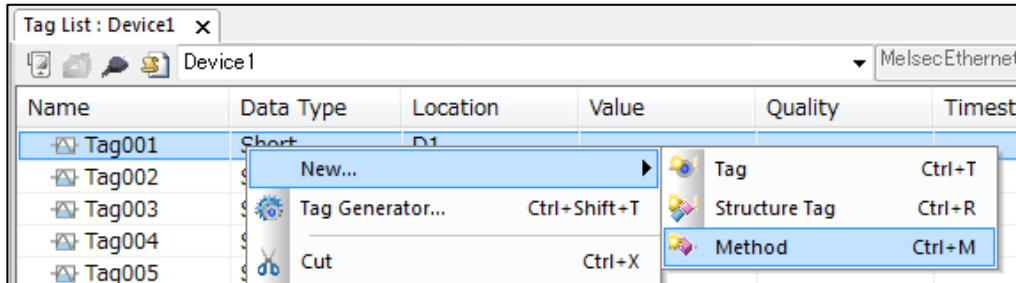
4.8 Method Tag

You can create a method tag with script processing.

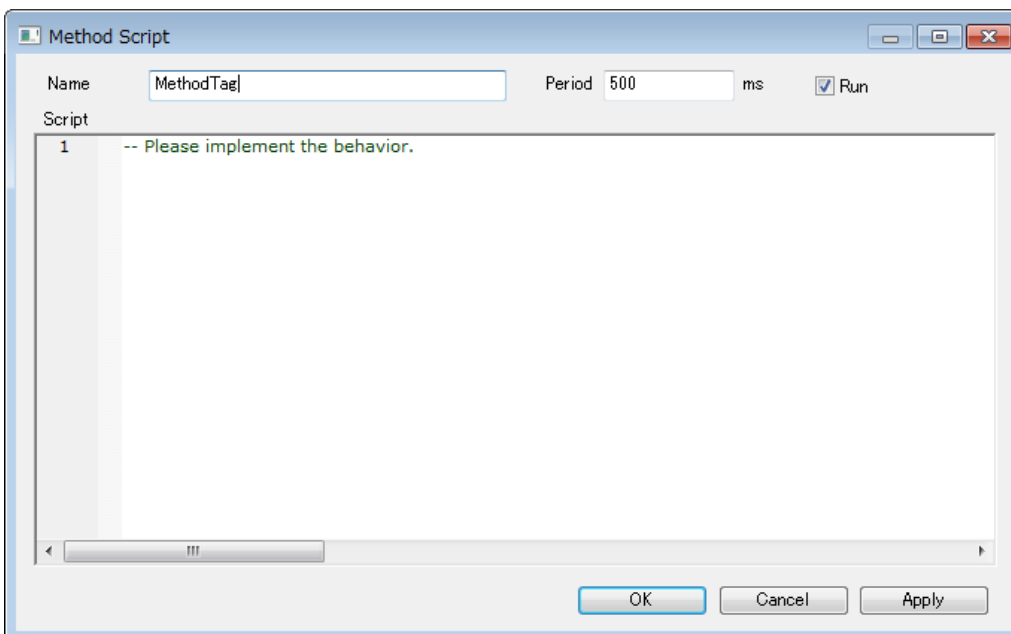
If you write value is to a method tag, DeviceXPlorer executes the script logic of Method Tag.

4.8.1 Creating a New Method Tag

From the Tag List menu, select New -> Method.



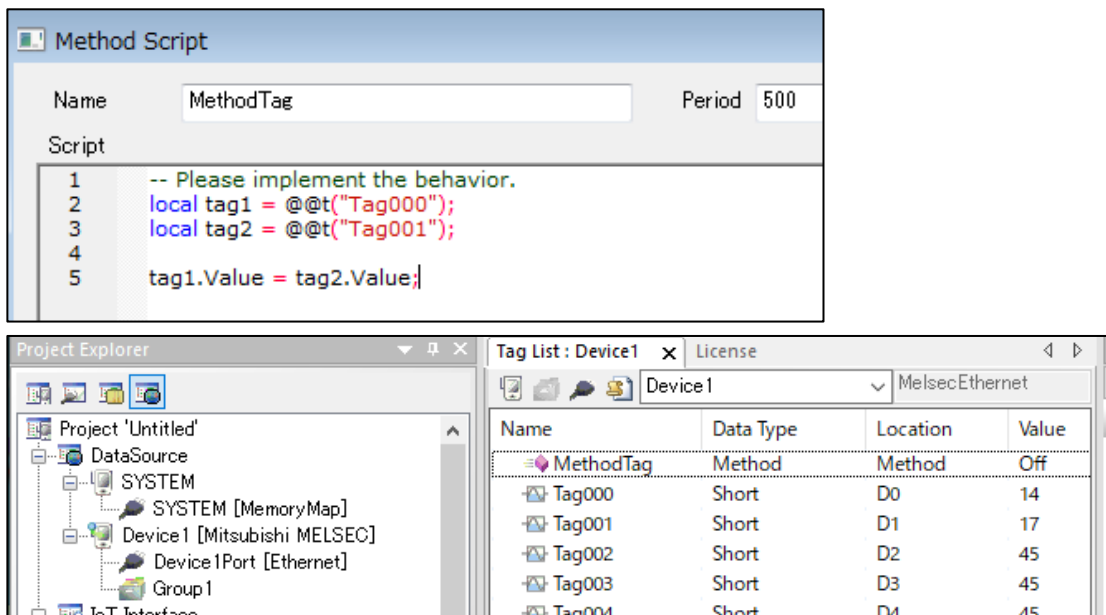
Edit script.



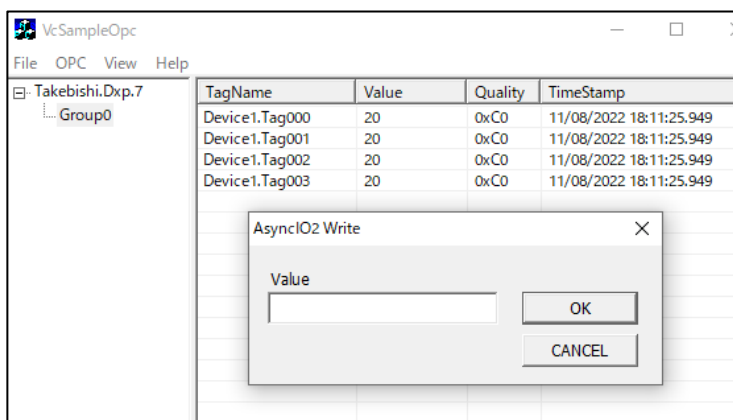
Item	Description
Name	Specify the Method Tag name.
Period	Specify the update rate for refresh tag's cache value which used in script.
Run	If you execute script, Check the box.
Script	Programing. Refers to 5

4.8.2 Execute Method Tag

You create a Method tag like follows.



If a value is written in from a client to this tag, the script will be performed.



Name	Data Type	Location	Value	Quality
MethodTag	Method	Method	On	Good (C0h)
Tag000	Short	D0	12345	Good (C0h)
Tag001	Short	D1	12345	Good (C0h)
Tag002	Short	D2	66	Good (C0h)

About the execution result of a method, when validating a script log, it can check with a message view.

I...	No	Date	Time	Type	Message
	00001	2012/05/13	20:46:47.421	Script	Called Method. (Script: Device1.MethodTag)
	00002	2012/05/13	20:46:47.421	SYSTEM	Write succeed.Device1.MethodTag [Val=-1]

4.9 Structure Tag

The model with two or more member variables of a structure is defined, and the same tag structure can be used only by generating a structure tag based on the definition. The structure has 2 type Structure Tag and Structure Template.

Item	Description
Structure Template	Definition of structure which has some members.
Structure Tag	Tag which created based on Structure Template.

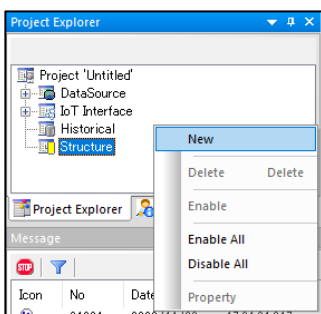
Important

Tag settings in ODBC Client Connection don't support Structure Tag.

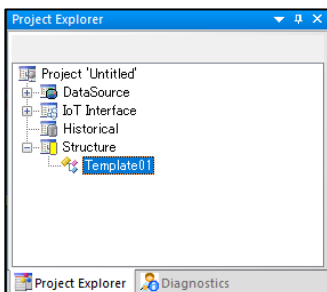
4.9.1 Define Structure Template

To use structure tags, first create a structure template.

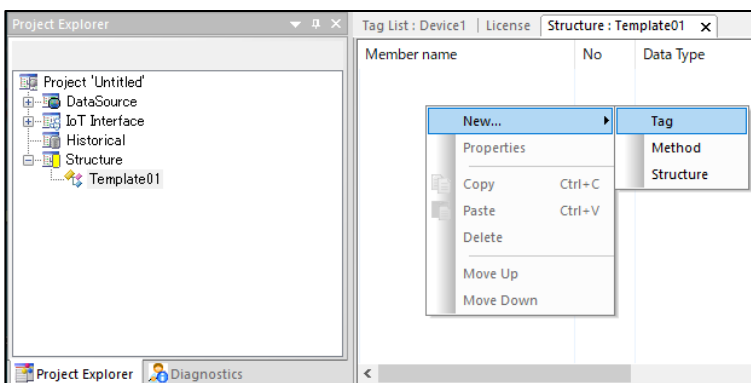
In the Project Explorer, right-click on "Structure" and select "Create New" from the right-click menu.



"Template01" is created as a Structure Template.



By clicking "Template01", Member list is shown. You select "Tag"/"Structure" from right-click menu of the component.



You configure the member properties .

<Tag>

Property of Member01 []

General Scale Simulation Alarm Historical

Member name

Description

Data Type Attribute

Length Calculation

☐ Byte Swap

☐ Array

Item	Description
Member name	Specify a member name.
Description	Allows you to add a comment to the member.
Data type	Select a data type.
Size	Specify the size if you have specified a character string or an array.
Attribution	Specify a member attribution.
System calculation	Select a calculation.
Byte swapping	To swap the high byte for the low byte, place a check in this box.
Array	Place a check in this box to define the tag as an array.

<Structure>

Property of StructTag01 []

General

Structure Tag Name

Description

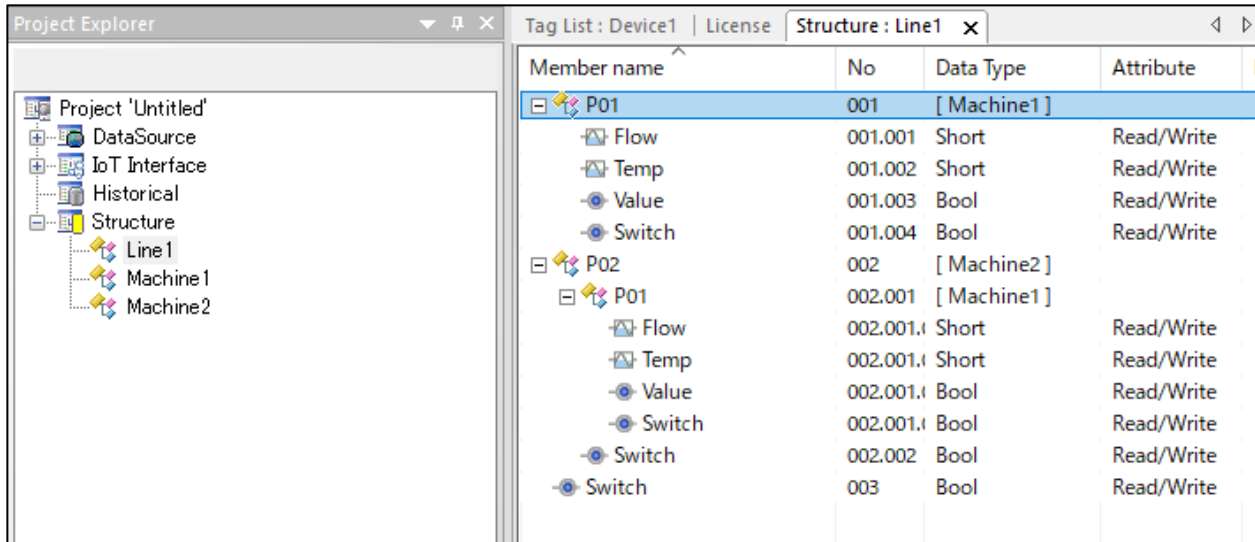
Structure Type

Start Register

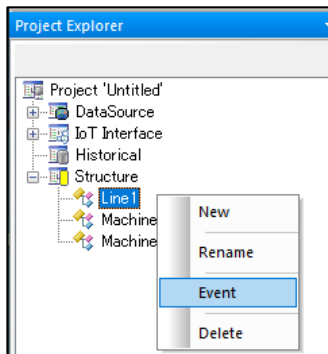
Member name	Data Type	Register	Attribute	Processing	Comment
☐ Member01	Short	D0	Read/Write		
☐ Member02	Short	D0	Read/Write		
☐ Member03	Short	D0	Read/Write		

Item	Description
Member name	Specify a member name.
Description	Allows you to add a comment to the member.
Structure Type	Select a Structure Template which defined already.
Member List	Show members of selected Structure Template.

By using two or more Structure Templates, the hierarchy Structure Template as follows can be created.



The right-click menu of Structure Template view is as follows.

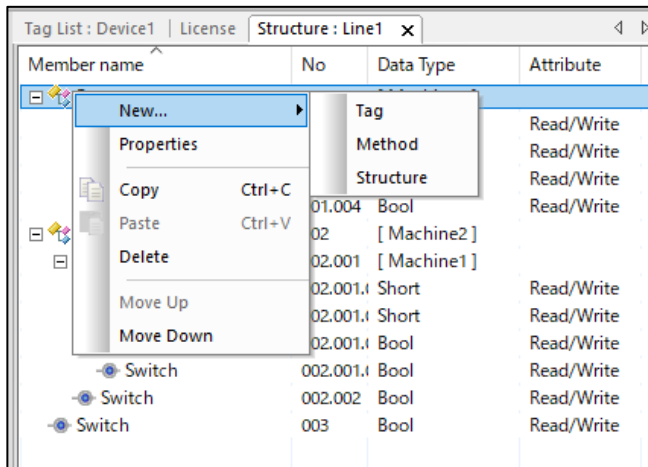


Item	Description
New	Create a new Structure Template.
Change the name	Change the selecting Structure Template name.
Event Script	Create a new Event Script of Structure Template.
Delete	Delete the Structure Template. If selected Structure Template is referred from another Structure Template, This structure template doesn't allow to delete.

Important

You can't delete the selected Structure Template is referred from another Structure Template.
You can't delete the selected Structure Template, if Structure Tag which created based on this Template is communicating.

The right-click menu of Structure Template members list is as follows.



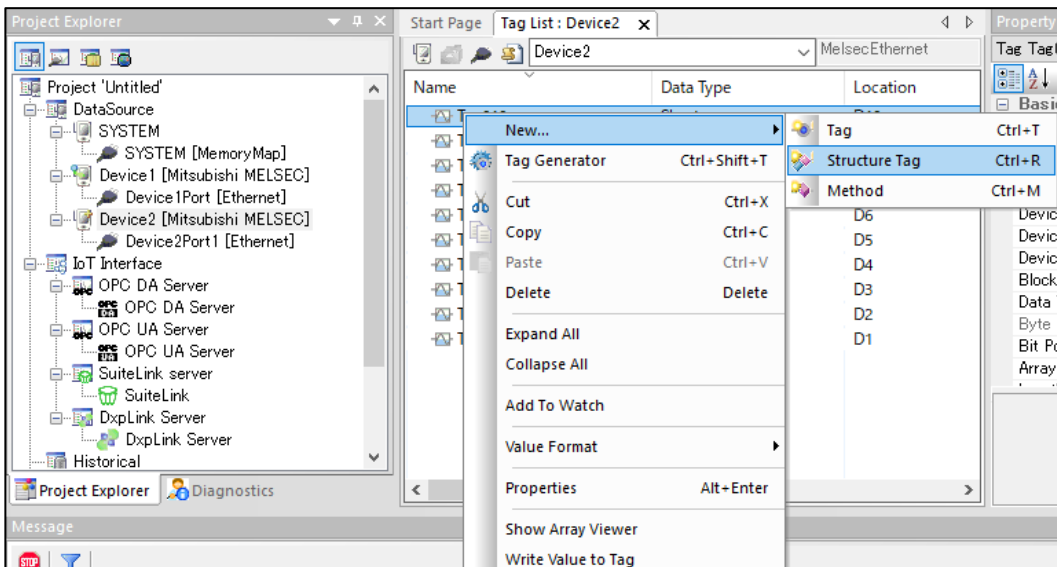
Item	Description
New	Create a new Structure Template member Tag or Structure.
Property	Configure the member properties.
Copy	Copy the selected member.
Paste	Paste the copied member.
Delete	Delete the selected member.
Move Up	The row of the member is moved upwards.
Move Down	The row of the member is moved downwards.

Important

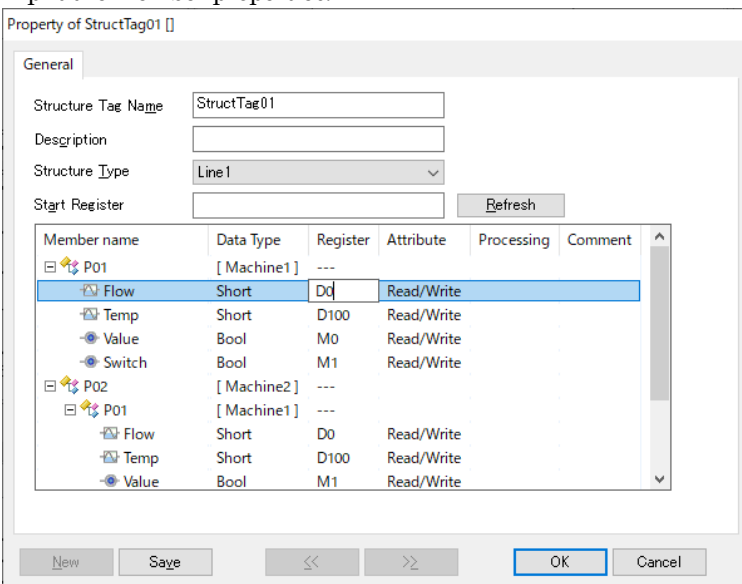
You can't delete the selected Structure Template member, if Structure Tag which created based on this Template is communicating.

4.9.2 Creating a New Structure Tag

You select "New->Structure Tag" from right-click menu of the Tag List.



Input the member properties.



Item	Description
Structure Tag Name	Specify the Structure Tag name.
Description	Allows you to add a comment to the Structure Tag.
Structure Type	Select a Structure Template which defined already.
Start Register	If you specify start register and push "Refresh" button, based on the start register, structure member's register will be assigned automatically.
Member List	Show members of selected Structure Template.

Important

A register's contents of a setting change to a character string with extended specification by the register who assigns. For example, if "D0" is inputted into a Float type member and a setup is saved to him, it will be registered as "D0:R."

You can monitor Structure Tags in Tag List.

Name	Data Type	Location	Value	Quality
Tag001	Short	D1	41	Good (C0h)
StructTag01 [Line1]				
Switch	Bool	M4	Off	Good (C0h)
P02 [Machine2]				
Switch	Bool	M3	Off	Good (C0h)
P01 [Machine1]				
Value	Bool	M1	Off	Good (C0h)
Temp	Short	D100	0	Good (C0h)
Switch	Bool	M2	Off	Good (C0h)
Flow	Short	D0	41	Good (C0h)
P01 [Machine1]				
Value	Bool	M0	Off	Good (C0h)
Temp	Short	D100	0	Good (C0h)
Switch	Bool	M1	Off	Good (C0h)
Flow	Short	D0	41	Good (C0h)

You can expand members by click [+], collapse by [-].

<Expand members>

P02 [Machine2]		
Switch	Bool	M3
P01 [Machine1]		
Value	Bool	M1
Temp	Short	D100
Switch	Bool	M2
Flow	Short	D0

<Collapse members>

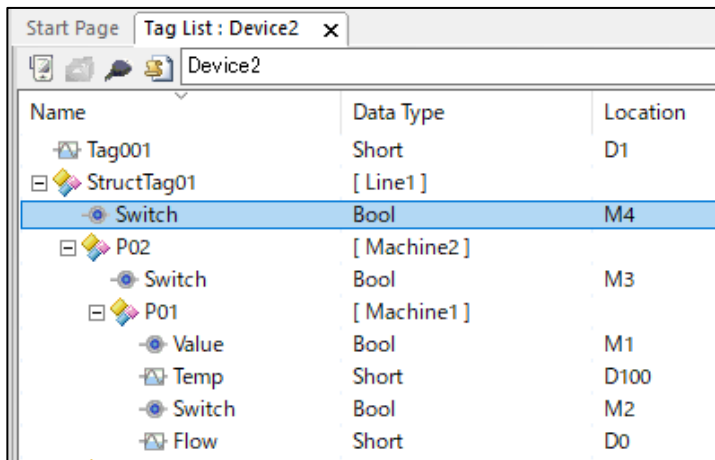
P02 [Machine2]		
Switch	Bool	M3
P01 [Machine1]		

Also, you can expand and collapse all structure tag's members by right-click menu. "Expand All" and "Collapse All".

Name	Data Type	Location	Value
Tag001	Short	D1	56
StructTag01 [Line1]			
Switch	Bool		
P02 [Machine2]			
Switch	Bool		
P01 [Machine1]			
Value	Bool		
Temp	Short		
Switch	Bool		
Flow	Short		
P01 [Machine1]			
Value	Bool		
Temp	Short		
Switch	Bool		
Flow	Short		

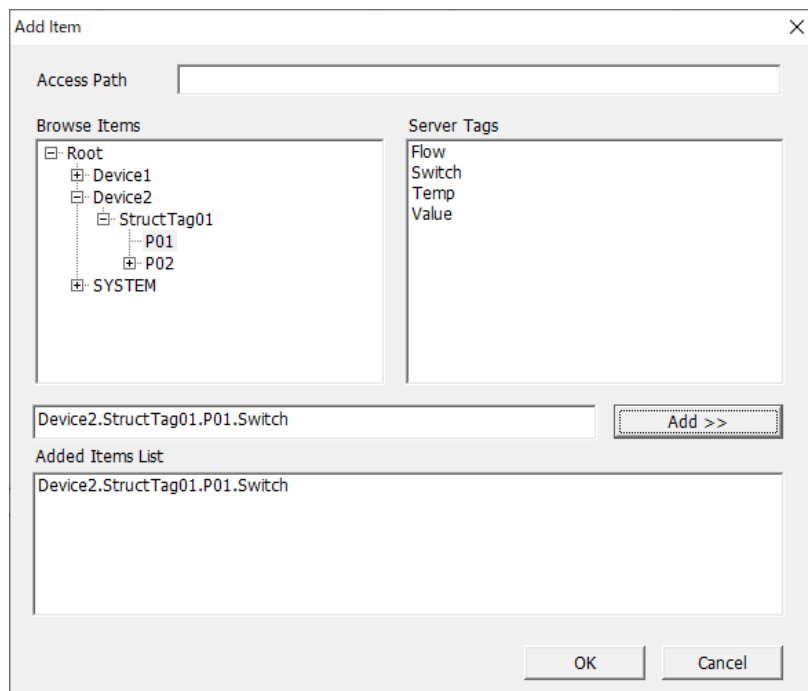
4.9.3 How to access Structure Tags from Client

When you access from a client, please access as a "Structure Tag name . member"
In Switch of the following, it accesses as "Device1.StructTag01.P01.Switch."



Name	Data Type	Location
Tag001	Short	D1
StructTag01	[Line1]	
Switch	Bool	M4
P02	[Machine2]	
Switch	Bool	M3
P01	[Machine1]	
Value	Bool	M1
Temp	Short	D100
Switch	Bool	M2
Flow	Short	D0

It is as follows if it browses from a client.



Access Path:

Browse Items:

- Root
 - Device1
 - Device2
 - StructTag01
 - P01
 - P02
 - SYSTEM

Server Tags:

- Flow
- Switch
- Temp
- Value

Device2.StructTag01.P01.Switch

Added Items List:

- Device2.StructTag01.P01.Switch

Important

The name accessed from a client is the same as the view of a group.

Keep in mind that a group "AAA" and the structure tag "AAA" are treated as the same object.

4.10 Bridge

You can transfer the value from Tag to Tag, Group to Group and Device to Device, by using bridge function.

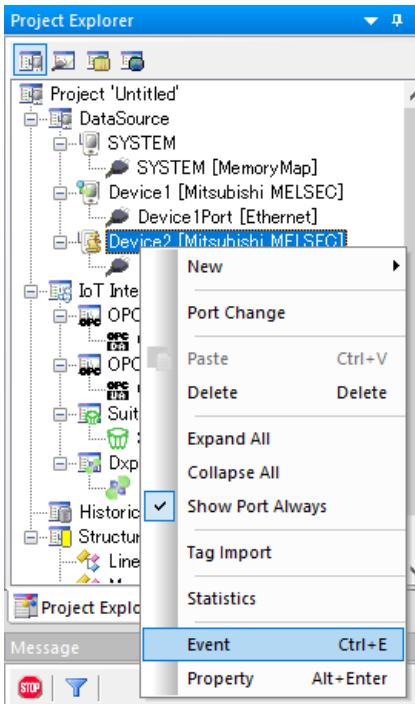
Important

- There is no upper limit to the number of bridge settings, but please note that the communication load will increase.
- Up to 32 items can be set in one bridge setting.
- Please note that executable number of scripts (including bridges) is limited depending on the edition of DeviceXPlorer.

4.10.1 Bridge Setting

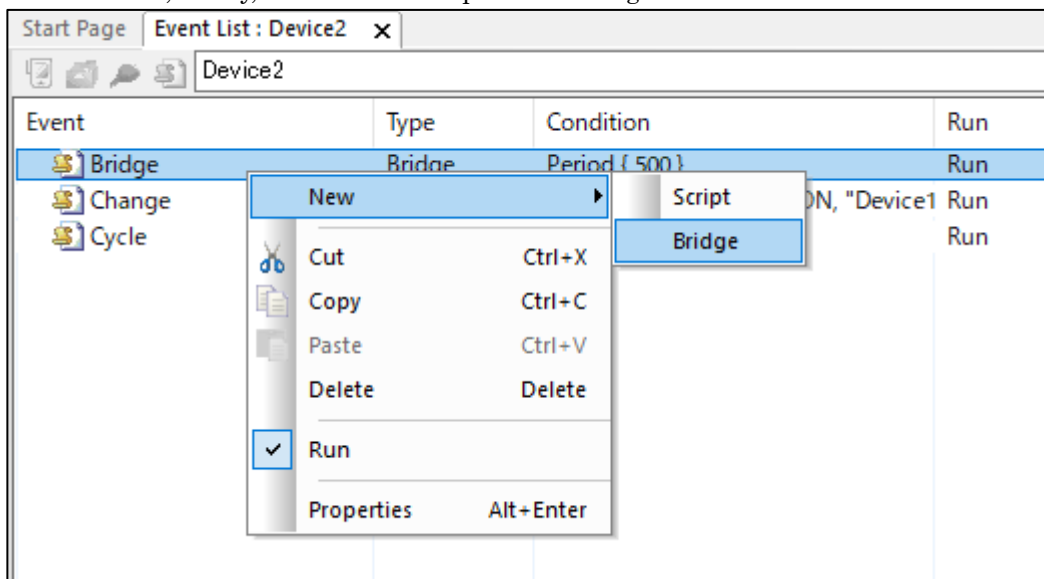
Tag, Group and Device have bridge function.

Event list will be displayed when select [Event] from right-click menu for Device and Group.



Events will be displayed in event list.

You can create, modify, delete and run/stop event from right-click menu.



Input bridge setting.

Event Bridge

Name: Bridge

☒ Run

Event: Period Period: 500 ms

Tag: While ON

Method	Destination	Source	Expression(x)
Tag	Device1.Tag000	<-- Device2.Tag001	

Add Delete OK Cancel Apply

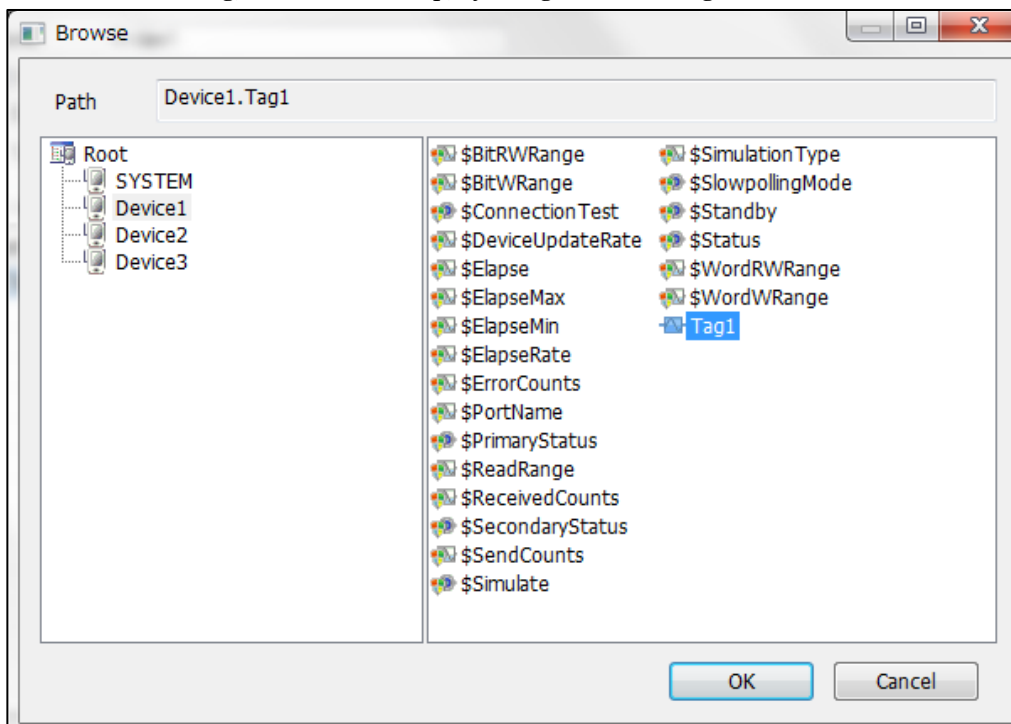
Item		Information
Name		Setting bridge name.
Run		Specify run state of bridge. If no check, bridge is not performed
	Event	Specify event condition to execute bridge. [Period] - - - Bridge is performed the specified cycle (fixed cycle). [Tag Change] - - - Bridge is performed when the value of the tag changed. [Run Event] - - - Bridge is performed at the time of event engine starting of the parent Device. (*) [Stop Event] - - - Bridge is performed at the time of event engine stopping of the parent Device. (*)
Run	Tag	When "Tag Change" is chosen in Event, a specification tag and conditions can be specified. Static tag and dynamic tag can be specified. While on the other hand, array and string tag cannot be specified. [While ON] - - -While a specification tag turns on, it performs at a case (when it is except zero). [Change] - - - Bridge is performed at the time of specified tag's Value or Quality or Timestamp changing. [Change(Value)] - - - Bridge is performed at the time of specified tag's Value changing. [Change (Quality)] - - - Bridge is performed at the time of specified tag's Quality changing. [Change (Time stamp)] - - - Bridge is performed at the time of specified tag's Timestamp changing. [Rise] - - - Bridge is performed at the time of specified tag changes to ON from OFF.

		[Fall] - - Bridge is performed at the time of specified tag changes to OFF from ON
	Period	Specify running interval and communication interval of bridge

* Event engine starts at the time of starting and stops at the time of application closing. And event engine will reboot to the timing when user creates, modifies and deletes event.

Item		Information
Bridge	Method	Specify the method of bridge.
	Destination	Specify the path of Tag, Device and Group which will be forwarded value. You can chose the path by direct input upon in-place editing, or Browse dialog (*) which will be shown by clicking [...] button.
	Source	Specify the path of Tag, Device and Group which transferring value. You can chose the path by direct input upon in-place editing, or Browse dialog (*) which will be shown by clicking [...] button.
	Expression(x)	Perform arithmetic operation via script. When you specify Group at Method, cannot perform conversion of value. Example: x * 100
Add		Add new bridge setting.
Delete		Delete the selected bridge settings
OK		Confirm the editing bridge setting.
Cancel		Cancel the editing bridge setting.

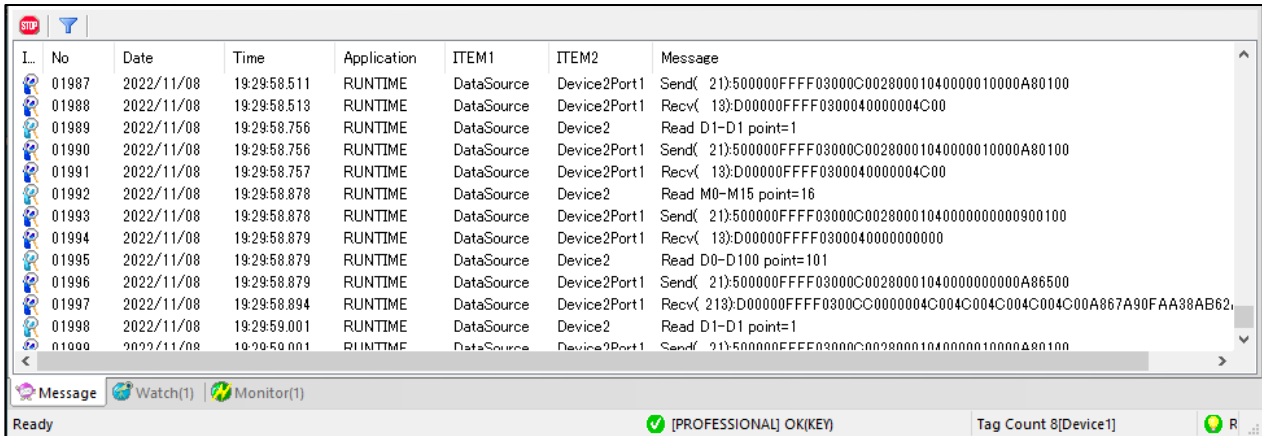
* You can choose Tag, Device and Group, by using Browse dialog function.



4.11 Messages

4.11.1 Message

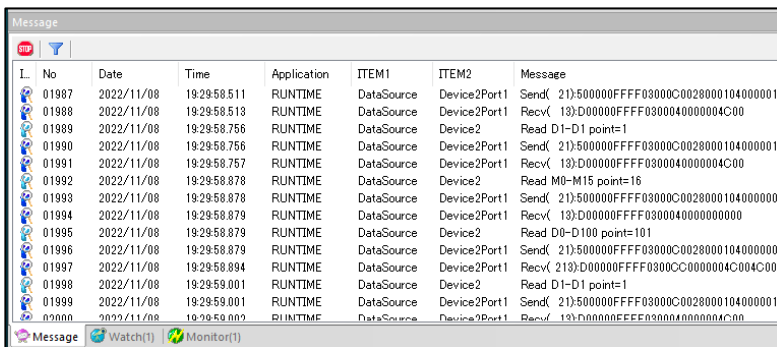
DeviceXPlorer divides messages into four levels of importance for processing (information, warning, error, and fatal error).



The importance of the message is indicated by the icons in the table below.

Icon	Category	Description
	Trace Detail	Trace Detail message for communication, etc.
	Trace	Trace message for communication, etc.
	Information	Information message for communication parameters, etc.
	Warning	Warning level message
	Error	Error level message

It is possible to use the pop-up menu or toolbar to display only the information you need by Icon Filter and Text Filter or to temporarily stop output.



[Pause]

Messages output can be suspended. Although the message is outputted in fact, updating on a display is stopped. The display of a message is updated by clicking Pause again.

Important

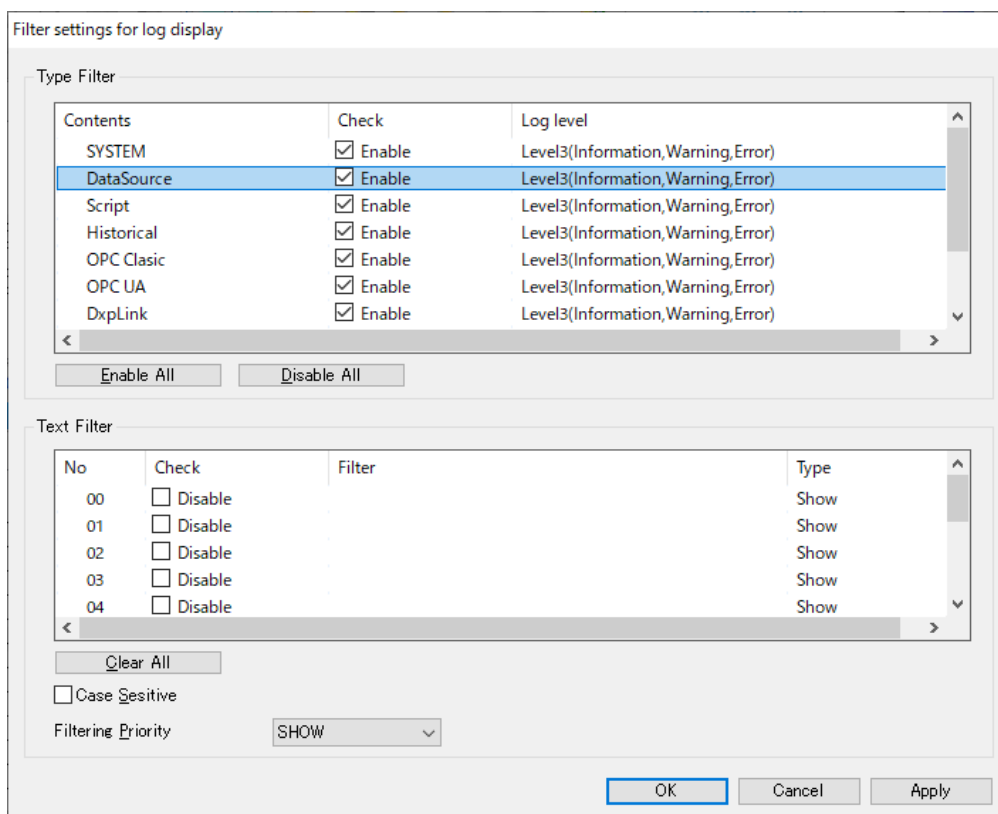
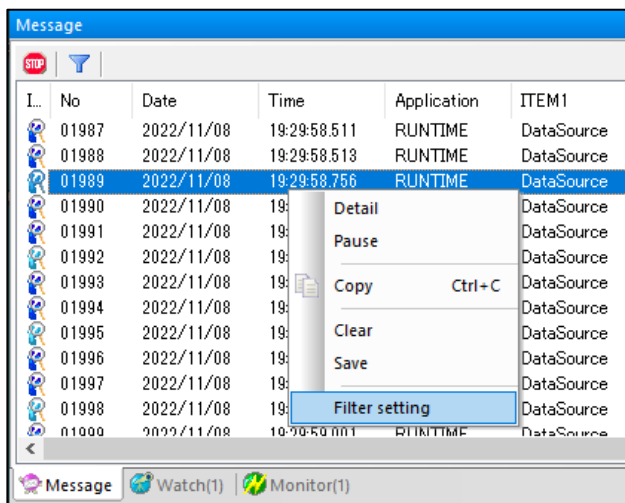
The function that saves log file doesn't be suspended, if you configure "Auto Message Save" as "Save".

[Clear]

Messages are cleared from Message View.

[Filter Setting]

You can filter the display of messages by right-clicking on the message screen and selecting Filter Settings.



<Text Filter>

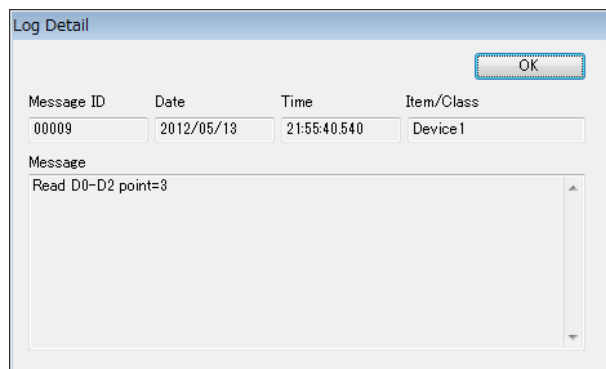
You can filter what is displayed in the message by specifying a string.

<Type Filter>

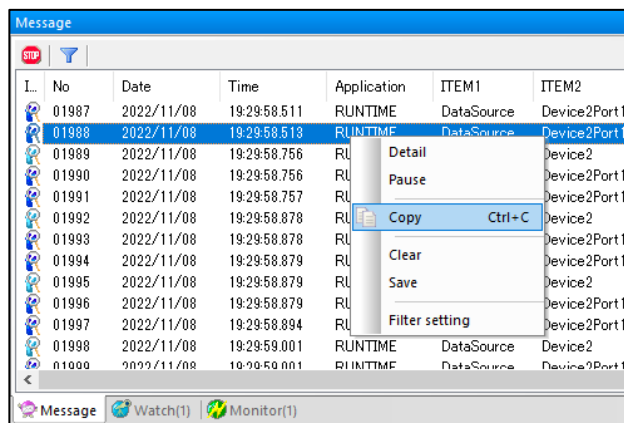
Item	Description
SYSTEM	Specifies the system log output level to be displayed.
DataSource	Specifies the DataSource log output level to be displayed.
Script	Specifies the Script log output level to be displayed.
Historical	Specifies the Historical log output level to be displayed.

OPC Classic	Specifies the OPC Classic log output level to be displayed.
OPC UA	Specifies the OPC UA log output level to be displayed.
DxpLink	Specifies the DxpLink log output level to be displayed.
SuiteLink	Specifies the SuiteLink log output level to be displayed.
HTTP Client	Specifies the HTTP Client log output level to be displayed.
MQTT Client	Specifies the MQTT Client log output level to be displayed.

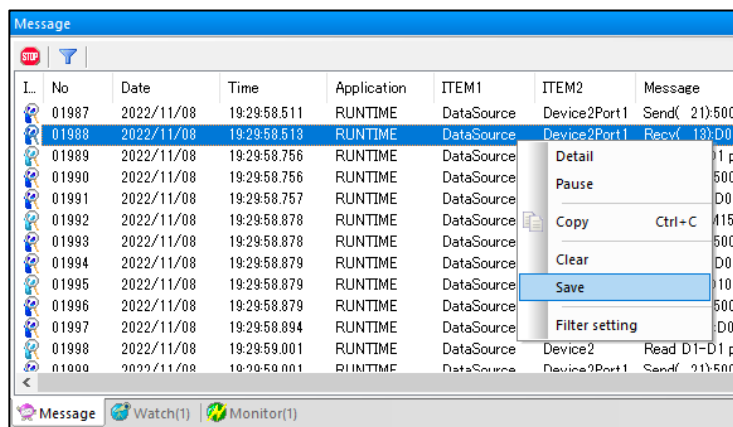
By double-clicking the message line, you can view a detailed dialog. It is possible to copy error messages.



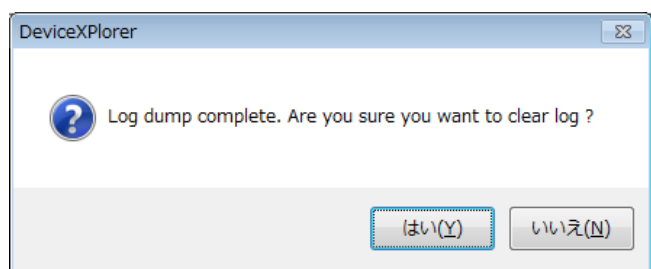
By selecting "Copy" from pop-up menu., you can copy the message to clipboard.



By selecting "Save" from the pop-up menu, you can save the message to csv file.

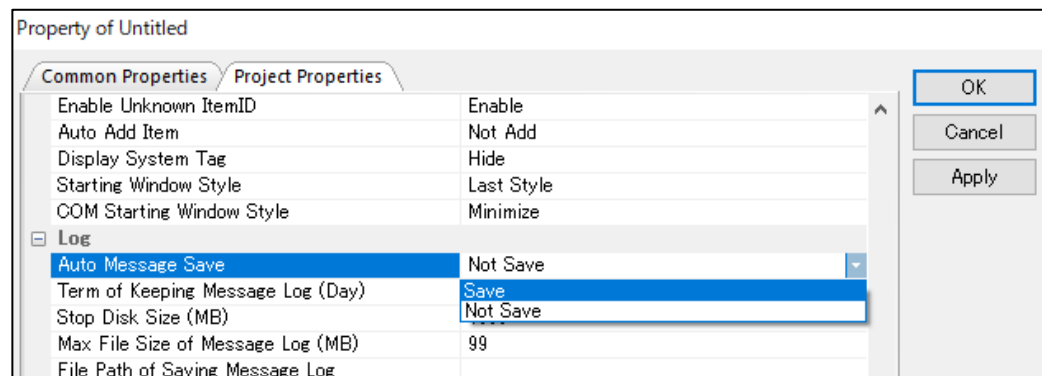


It is checked after CSV file preservation whether messages are cleared or not.



4.11.2 How to save messages automatically

You configure “Auto Message Save” as “Save” in Project Properties (4.15).

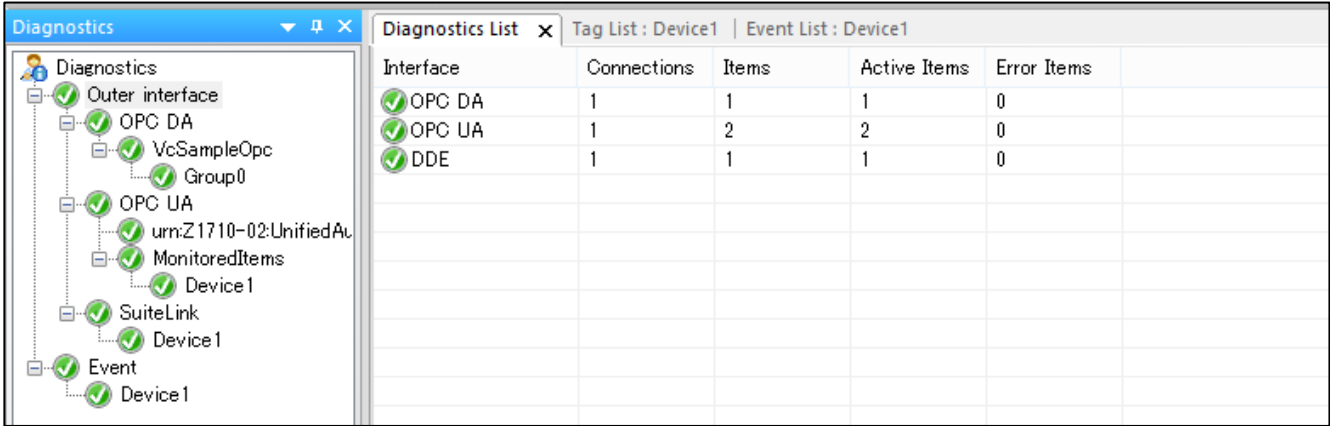


4.12 Diagnosis

You can use the diagnostic function to check the access status of a client or investigate the cause of a problem. Undernoted icons show diagnosis condition.

Icon	Description
	Condition is Good.
	Good and Bad are mixed.
	Condition is Bad. Communication error is occurring.
	Condition is not Active status.

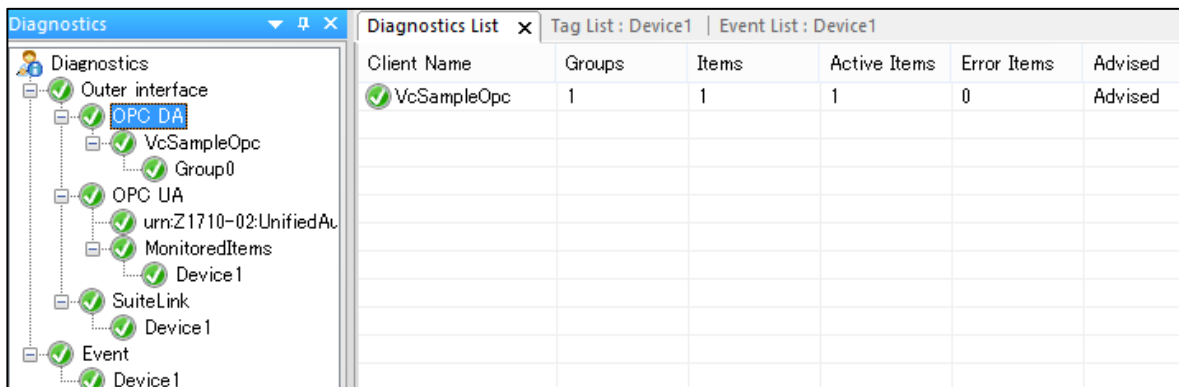
4.12.1 Interface Diagnosis



Item	Description
Interface	OPC interfaces and SuiteLink interfaces
Connections	Number of clients connected to each interface
Items	Number of registered tags
Active items	Number of active tags
Error items	Number of error tags

4.12.2 Client Diagnosis

This shows information for the connected clients.



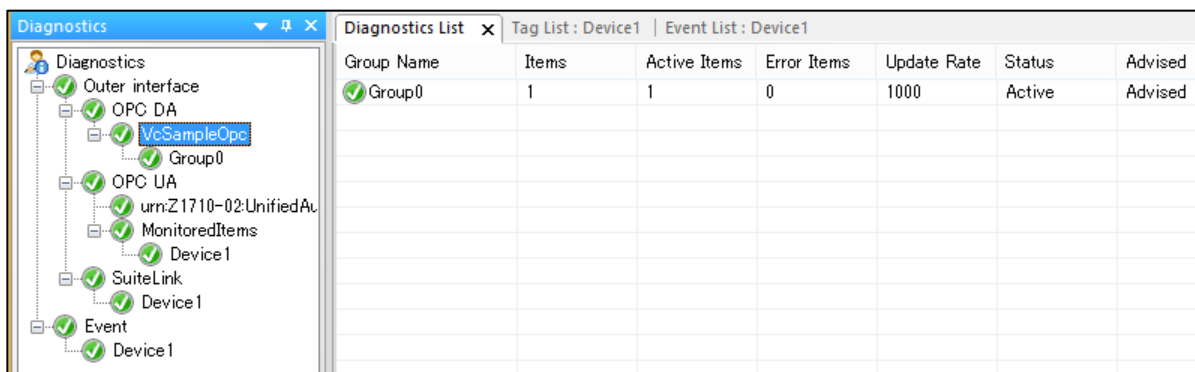
The screenshot shows the 'Diagnostics' window with the 'OPC DA' tree on the left. The 'Diagnostics List' tab is active, displaying a table with the following data:

Client Name	Groups	Items	Active Items	Error Items	Advised
VcSampleOpc	1	1	1	0	Advised

Item	Description
Client	Client handle and client name
Groups	Number of registered groups
Items	Number of registered tags
Active items	Number of active tags
Error items	Number of error tags
Advised	The status of advise connection for OPC server object.

4.12.3 Group Diagnosis

This shows the information of the group to which the client is registered.



The screenshot shows the 'Diagnostics' window with the 'OPC DA' tree on the left. The 'Diagnostics List' tab is active, displaying a table with the following data:

Group Name	Items	Active Items	Error Items	Update Rate	Status	Advised
Group0	1	1	0	1000	Active	Advised

Item	Description
Group name	OPC group name
Items	Number of registered tags
Active items	Number of active tags
Error items	Number of error tags
Update Rate	Update interval value set for the group
Status	Active/Inactive status of group
Advised	The status of advised connection for OPC group object.

4.12.4 Tag Diagnosis

This shows information on the tags included in the group.

Name	Data Type	Location	Value	Quality	Timestamp	Attribute	Processing	Comment	Status
Device1.Tag	Short	D0	0	Good (C0h)	2019/11/12...	Read/Write			Active

Item	Description
Name	Tag name
Data Type	Tag data type
Location	Physical address to refer
Value	The most recent cache value
Quality	Quality of tag
Time stamp	Time stamp
Attribute	Attribute of tag
Processing	Existence of scale setting
Comment	Comment of tag
Status	Active / InActive status of tag

Undernoted icons show the diagnosis condition of tag.

Icon	Description
	Tag quality is Good.
	Tag quality is Uncertain.
	Tag quality is Bad. Communication error is occurring.
	Tag quality is Waiting for initial data.
	Tag condition is not active status.

You can write the value to tag as well as the tag monitor. You can register the tag to the watch view by drag and drop.

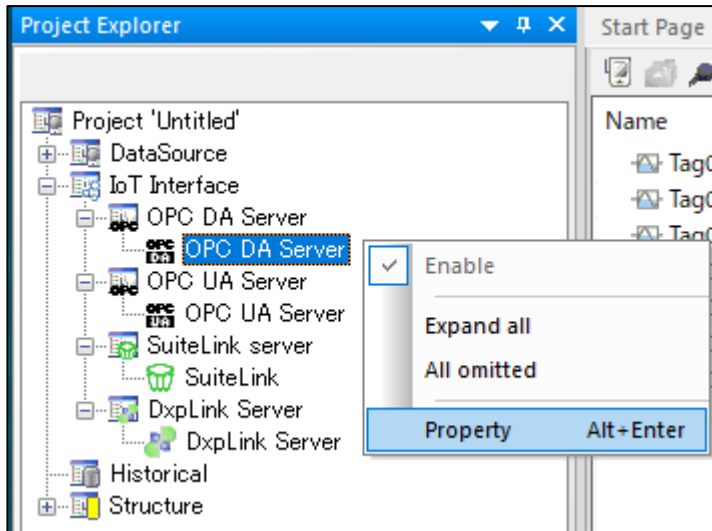
Name	Data Type	Location	Value	Quality	Timestamp
Device1.Tag	Short	D0	0	Good (C0h)	2019/11/12...

4.13 IoT Interface

4.13.1 OPC DA Server

Configure the OPC DA Server settings.

Right-click "OPC DA Server" from External System Linkage in the Project Explorer and select Properties to configure the OPC DA Server settings.



IoT Interface property [OPC DA Server]

IoT Interface

Update Rate: 500 msec

Timer Tick for Async Update Mode (msec): 10 msec

☒ Return cache in error

☐ Return success for communication failure at reading

Initial DataChange Timeout: 15000 msec

OPC Thread Priority: NORMAL

Acceptable Max Count of Async Demand: 1000

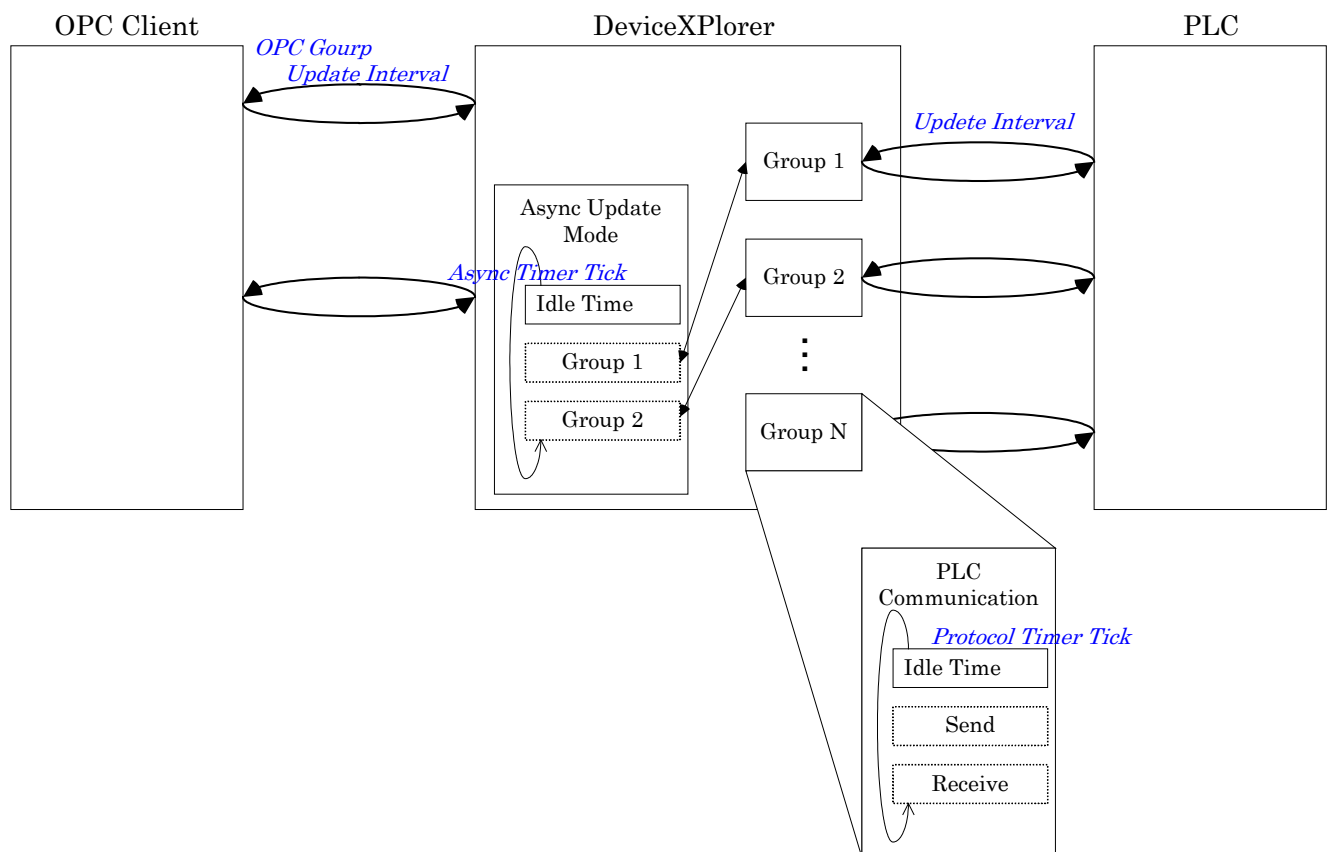
OK Cancel Apply

Item	Default	Reboot*	Description
OPC Update Rate	500 (ms)	Need	Specifies the interval for notifying an OPC client of data changes. This setting value is used when OPC client specifies OPC UpdateRate as "0".
Timer Tick for Async Update Mode	10 (ms)	Need	Specifies the interval for notification of data changes when the asynchronous update mode is on.
Return cache in error	Enabled	---	Return cache value when error is occurred. If the setting is disabled, the EMPTY value is returned.

Return success for communication failure at reading	Disable	---	Return success code 'OPC_S_UD_COMM_ERROR(0x00048000)' for communication failure at reading. If this is enabled, return failure code 'OPC_E_UD_COMM_ERROR(0x80048000)'.
Initial DataChange Timeout	15000	---	The timeout for getting first value when initial DataChange. DeviceXPlorer sets Last Usable Value(0x44) or Waiting for Initial Data(0x20) to the quality of the items that are past timeout, and notifies to the client.
OPC Thread Priority	NORMAL	Need	Specifies the thread priority. [High] [Low] CRITICAL > HIGHEST > NORMAL > LOWEST > IDLE
Acceptable Max Count of Async Demand (OPC)	1,000	Need	Sets the maximum number of asynchronous requests (OPC interface) that can be accepted. If asynchronous requests continue to be called faster than the actual communication can handle them, once a specified number of requests build up DeviceXPlorer returns CONNECT_E_ADVISELIMIT (0x80040201).

* If the settings are changed while the OPC client is connected, the settings will not be updated for the connected OPC client. To reflect the settings, reconnect the OPC client.

[Settings and Internal Operations]



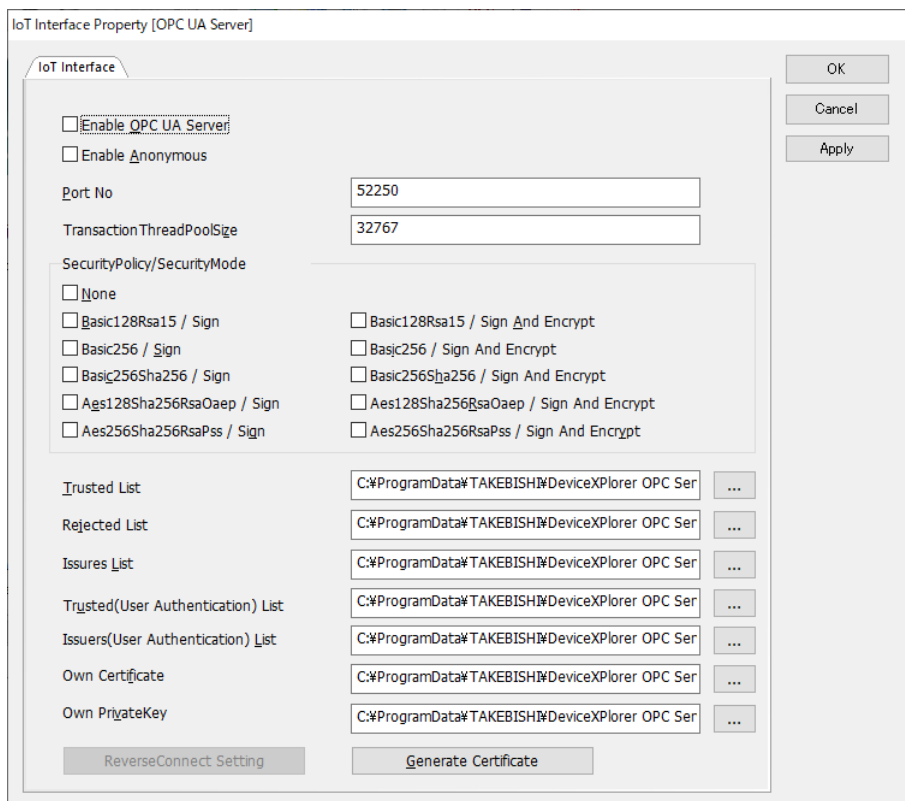
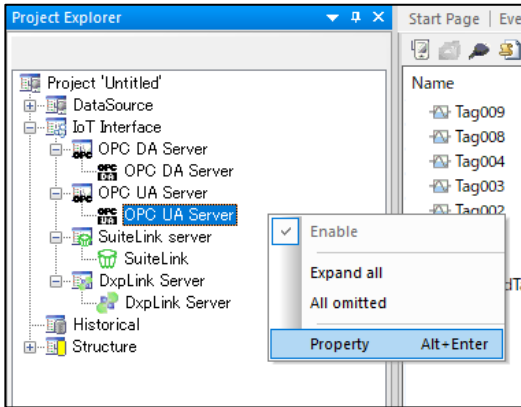
* The "Update Interval" in the figure above is a setting that is made by the OPC client when AddGroup() is running.

Project settings are comprised of common properties and project properties. The settings that affect the application as a whole are "Common Properties", while the items that are limited to a particular project are set as "Project Properties". The project settings screen can be opened from "Properties" under the "Projects" menu, or from "Options" under the "Tools" menu.

4.13.2 OPC UA Server

Configure the OPC UA server settings.

Right-click "OPC UA Server" from External System Linkage in the Project Explorer and select Properties to configure the OPC UA server settings.



Item	Default	Description
Enable UA Server	Disable	Select whether you use OPC UA Server function is, or not.
Enable Anonymous	Disable	Select whether to allow anonymous connections.
Port No	52250	Specify the port number.
Transaction ThreadPoolSize	32767	Specify the number of transactions that can be accepted simultaneously.
Security Policy/ Security Mode		
None	Disable	Select the restriction of the security Policy/ Security mode to be connected.
Basic128Rsa15 / Sign		
Basic128Rsa15 /		

	Sign AndEncrypt Basic256 / Sign Basic256 / Sign AndEncrypt Basic256Sha256 / Sign Basic256Sha256 / Sign AndEncrypt Aes128Sha256 RsaOaep / Sign Aes128Sha256 RsaOaep / Sign AndEncrypt Aes256Sha256 RsaPss / Sign Aes256Sha256 RsaPss / Sign AndEncrypt	
Trusted List*1	C:\ProgramData\TA KEBISHI\DeviceXPl orer OPC Server 7\pkiserver\trusted	Specify the directory where trusted certificates are stored.
Rejected List	C:\ProgramData\TA KEBISHI\DeviceXPl orer OPC Server 7\pkiserver\rejected	Specify the directory for storing rejected certificates. If you want to allow the connection, move it to the trusted directory.
ISSUERS List*1	C:\ProgramData\TA KEBISHI\DeviceXPl orer OPC Server 7\pkiserver\issuers	Specify the directory for storing the certificate and revocation list file (CRL).
Trusted(User Authentication) List*1	C:\ProgramData\TA KEBISHI\DeviceXPl orer OPC Server 7\pkiserver\user_tru sted	Specify the directory to store the certificate when connecting using an X509 authenticated user.
Issuers(User Authentication) List*1	C:\ProgramData\TA KEBISHI\DeviceXPl orer OPC Server 7\pkiserver\user_iss uers	Specify the directory for storing the revocation list file (CRL) when connecting using an X509 authenticated user.
Own Certificate	C:\ProgramData\TA KEBISHI\DeviceXPl orer OPC Server 7\pkiserver\own\cer ts\DxpOpcUaServer.d er	Specify own certificate path.
Own PrivateKey	C:\ProgramData\TA KEBISHI\DeviceXPl orer OPC Server 7\pkiserver\own\pri vate\DxpOpcUaServe r.pem	Specify own private key path.

Generate Certificate	—	Refer to 4.13.2.1
Revers Connect	—	Refer to 4.13.2.2

*1 Store the certificate file in certs under the specified folder and the revocation list in crl.

4.13.2.1 Generate Certificate Setting “UA Server”

The screenshot shows a 'Generate Certificate' dialog box with the following fields and values:

- Organization: TAKEBISHI
- OrganizationUnit: (empty)
- Locality: (empty)
- State: KYOTO
- Country: JP
- DomainName: takebish
- IP Address: (empty)
- RSA Key Strength: 2048bits
- Signature Algorithm: Sha256
- Certificate Validity (Year): 3

Buttons: OK, Cancel

Item	Description
Organization	Set Organization to be embedded in the server certificate.
OrganizationUnit	Set the Organization Unit to be embedded in the server certificate.
Locality	Set Locality to be embedded in the server certificate.
State	Set the State to be embedded in the server certificate.
Country	Set the Country to be embedded in the server certificate.
DomainName	Set the domain Name to be embedded in the server certificate.
IP Address	Set the IP Address to be embedded in the server certificate.
RSA Key Strength	Set the RSA Key Strength to be c
Signature Algorithm	Specify the Signature Algorithm embedded in the server certificate.
Certificate Validity (Year)	Specify the Certificate Validity(Year) embedded in the server certificate.

*UA server certificates are generated when no certificate exists at application startup.

UA client certificates are generated when a certificate does not exist when connecting to the UA server.

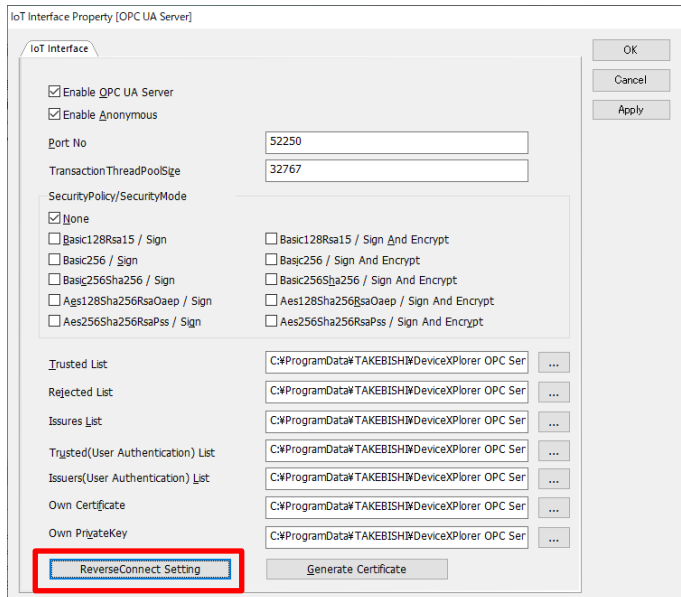
4.13.2.2 Revers Connect Setting

ReversConnect configuration.

ReversConnect requires the specification of the server and client endpoint URLs and server certificates on the client side in advance. The client makes a connection upon receiving a connection request from the specified server.

[DeviceExplorer Settings]

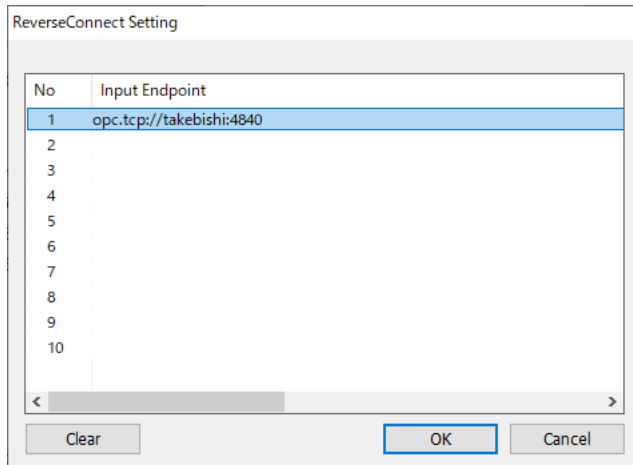
Click on "ReversConnect Settings" to open the settings window.



In the configuration screen, set the client's Endpoint URL, which is specified on the client side.

DeviceXPlorer makes a connection request to the client according to the configured Endpoint URL.

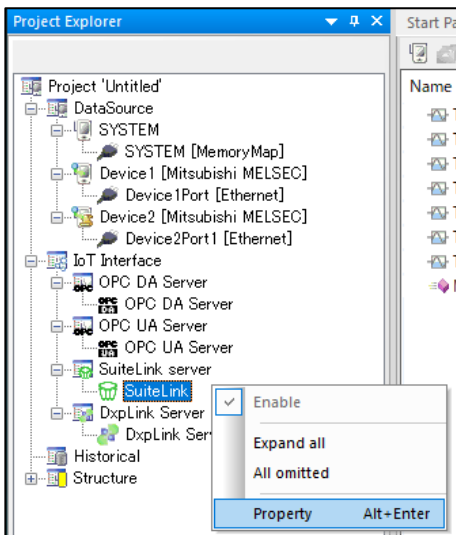
Specify "opc.tcp:// Machine name : Port number" for Endpoint.



4.13.3 SuiteLink Server

Configure the SiteLink server settings.

Right-click "SuiteLink Server" from External System Linkage in the Project Explorer and select Properties to configure the SiteLink server settings.



IoT Interface property [SuiteLink]

IoT Interface

☐ Notify Cache First

Acceptable Max Count of

OK

Cancel

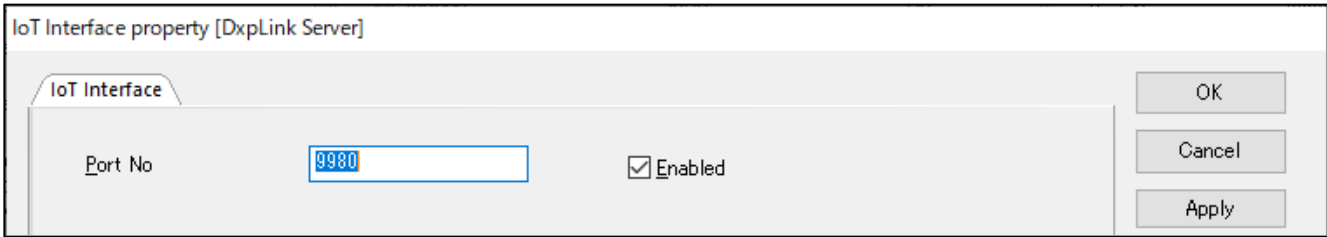
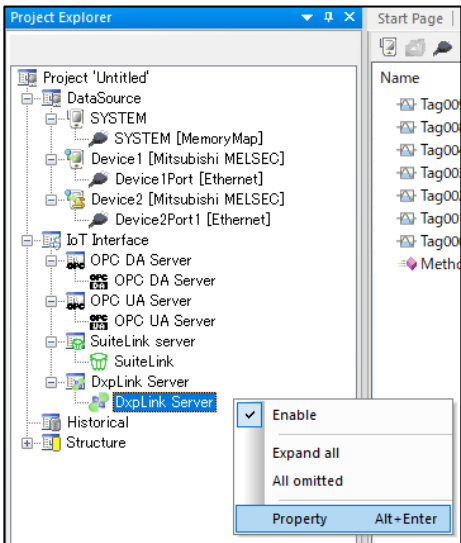
Apply

Item	Default	Description
Notify Cache first	Disable	Send cache value of tag at first notification immediately when SuiteLink clients connect the server. You can avoid timeout on SuiteLink client.
Acceptable Max Count of Async Demand	10,000	Sets the maximum number of asynchronous writing requests (SuiteLink interface) that can be accepted. If asynchronous requests continue to be called faster than the actual communication can handle them, once a specified number of requests build up DeviceXPlorer returns error.

4.13.4 DxpLink Server

Configure the DxpLink server settings.

Right-click "DxpLink Server" from External System Linkage in the Project Explorer and select Properties to configure the DxpLink server settings.



Item	Default	Description
DxpLink Server Function	Enable	Place a check in this box to enable the DxpLink Server function.
DxpLink Port No	9980	Specify the port No of DxpLink Serer.

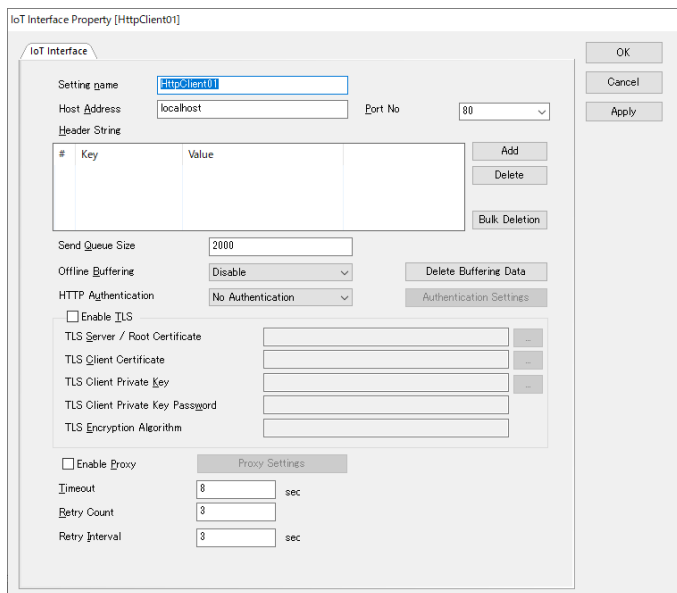
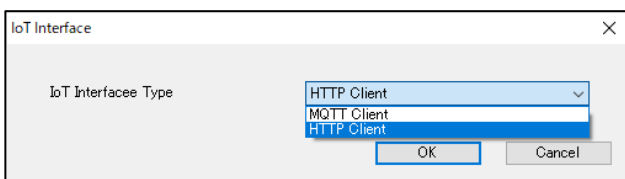
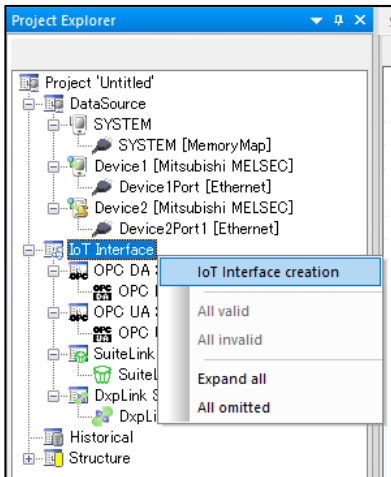
4.13.5 HTTP Client

Configure the HTTP client settings.

4.13.5.1 HTTP-based Setting

Configure the HTTP client settings.

Right-click "External System Linkage" in the Project Explorer and select "Create External System Linkage". Select "HTTP Client" from "External System Linkage Type" and configure the settings.



Item	Description
Setting name	Sets the name of the external system configuration.
Host address	Specifies the host address.
Port No	Specify the port number. Generally, 80 is used for HTTP and 443 for HTTPS.
Hesder string	Specifies the header string to be inserted when creating a new message.
Send queue size	Specifies the maximum number of messages waiting to be sent.
Offline buffering	Enable to temporarily buffer messages waiting to be sent. If set to "Always", all

	messages will be buffered. When set to "Enable in case of communication failure," messages are buffered when the connection with the server times out.
HTTP authentication	Specify the HTTP authentication type. Basic authentication, Digest authentication, and OAuth2.0 can be specified.
Use TLS	
TLS Server / Root Certificate	Specifies the HTTPS server or root certificate.
TLS Client Certificate	Specifies the device explorer certificate.
TLS client private key	Specifies the private key for DeviceXPlorer.
Client private key password	Specify the client private key if it has a password set.
TLS encrypt algorithm	Specifies the TLS encryption algorithm. See CipherList of OpenSSL. If not specified, "ALL" is assumed.
Use proxy	Enable when using the proxy setting.
Waiting for response	Specifies the timeout period to wait for the completion of the transmission process. 0 specifies the timeout period to wait until the process succeeds or fails.
Number of Send retry	Specifies the number of times to retry transmission in case of transmission failure.
Send retry interval	Specifies the minimum interval between transmission retries.

If proxy use is enabled, advanced proxy settings are required.

The screenshot shows a web-based configuration interface titled "IoT Interface Detailed Settings [Proxy]". Inside, there's a section titled "IoT Interface Detailed Settings". It contains the following fields:

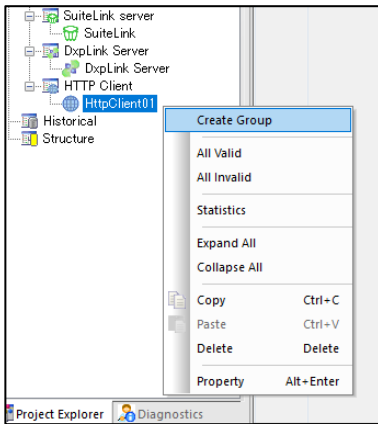
- Proxy Address:** A text input field.
- Proxy Port Number:** A text input field with the value "80" entered.
- Proxy Authentication:** A checkbox that is currently unchecked.
- Proxy Username:** A text input field, visible only if the checkbox is checked.
- Proxy Password:** A text input field, visible only if the checkbox is checked.

Item	Description
Proxy Address	Specify the machine name or IP address of the proxy server.
Proxy Port Number	Specify the port number of the proxy.
Proxy Authentication	Enable if the proxy server requires user authentication.
TLS Server / Root Certificate	Specify the user name required for proxy authentication.
TLS Client Certificate	Specify the password required for proxy authentication.

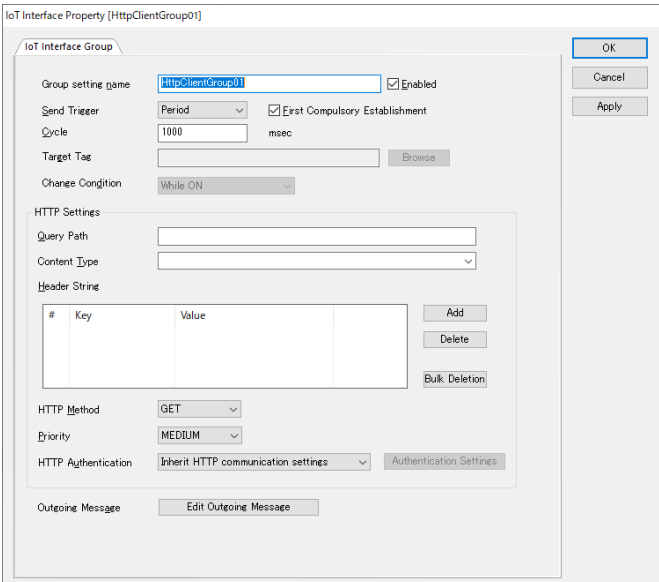
4.13.5.2 HTTP group Setting

HTTP group is performed. This is where you configure the settings for sending data.

Right-click on the HTTP base setting you created and select "Create Group".



Properties set the period, query path, header string, etc.



Item	Description
Group setting name	Specifies the group setting name.
Enabled	Enables/disables the operation.
Send Trigger	Specify the type of message to be sent.
Cycle	Specifies the message sending cycle.
Target Tag	Specify the target of tag change event.
Change Condition	Specify the event to be executed when the tag value changes in a tag change event.
First Compulsory	The trigger is forced to be established once at the beginning of the operation.
HTTP settings	
Query Path	Specifies the URL relative path followed by the domain.
Content Type	Specifies the content type of the outgoing message.
Header String	Specify the header string to be inserted when creating a new message.
HTTP Method	Specify the HTTP method.
Priority	Select the priority level. If multiple transmission requests are generated, transmission processing is performed starting with the request with the highest priority.

HTTP Authentication	Select the HTTP authentication type.
---------------------	--------------------------------------

4.13.5.3 Outgoing Message(HTTP)

Select Edit Outgoing Message in the HTTP Group Settings to edit the content of the message to be sent.

Item	Description																		
Outgoing Message																			
<@SERVERTIMESTANP>	Inserts the system timestamp at the time the message was created. Also specify the time format as <@SERVERTIMESTAMP("yyyy/MM/dd hh:mm:ss.fff")>. Refer to the table below for the specification character string of the format. <table> <tr> <th>character string</th><th>Data</th></tr> <tr> <td>y</td><td>Year</td></tr> <tr> <td>M</td><td>Month</td></tr> <tr> <td>d</td><td>Day</td></tr> <tr> <td>h</td><td>Hour(0-23)</td></tr> <tr> <td>m</td><td>Minute</td></tr> <tr> <td>s</td><td>Second</td></tr> <tr> <td>S</td><td>UNIX time</td></tr> <tr> <td>f</td><td>Milli second</td></tr> </table>	character string	Data	y	Year	M	Month	d	Day	h	Hour(0-23)	m	Minute	s	Second	S	UNIX time	f	Milli second
character string	Data																		
y	Year																		
M	Month																		
d	Day																		
h	Hour(0-23)																		
m	Minute																		
s	Second																		
S	UNIX time																		
f	Milli second																		
<@TAGS_START>	Repeat the range enclosed by {#each VALUE} and {/each} for the number of tags set in the tag list, reflecting the current value of each tag.																		
<@TAGS_END>	Repeat the range enclosed by {#each VALUE} and {/each} for the number of tags set in the tag list, reflecting the current value of each tag.																		
<@USERDEFINE>	Inserts the "user-defined" string set in the tag list of the HTTP group.																		
<@TAGNAME>	Inserts the "tag name" string set in the tag list of the HTTP group settings.																		
<@VALUE>	Inserts the value of the tag at the time of message creation.																		
<@QUALITY>	Inserts the quality flag of the tag when creating a message.																		
<@TIMESTANP>	Inserts the timestamp of the tag when composing a message. The format can be specified in the same way as SERVERTIMESTANP.																		
<@COMMENT>	Inserts a "comment" set in the tag list of HTTP group settings.																		
Insert	Inserts the selected outgoing message.																		
Line feed code	Specify the line feed code. (CR, CR+LF, LF)																		
Delimiter	Specify the delimiter character.																		
Preview	Confirm the preview of the outgoing message.																		

4.13.5.4 HTTP Tag List

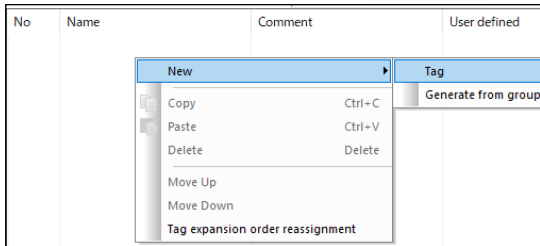
IoT Interface : HttpClient01.HttpClientGroup01 x Tag List : Group1			
No	Name	Comment	User defined
1	Device1.ArrayTag		
2	Device1.StringTag		
3	Device1.Tag		
4	Device1.Group1.Tag		

Item	Description
No	Display the order contained in sending message. The order can be changed by right-clicking.
Name	Display the tag name of the tag that sends.
Comment	Display comments set in tag settings. Comments can be included in sending message.
User defined	The user can freely enter characters. User defined can be included in sending message.

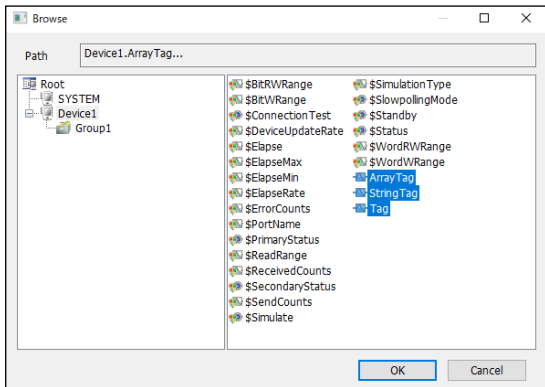
4.13.5.4.1 Add from HTTP Tag List

<Add by Tag>

Right-click on HTTP tag list and click New > Tag.



Select the tags and click OK.

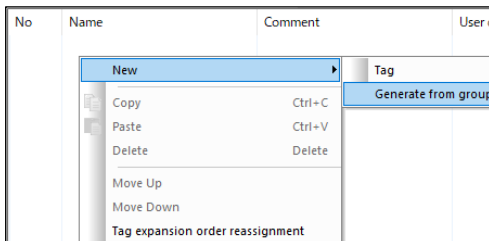


Added Tags to HTTP tag list.

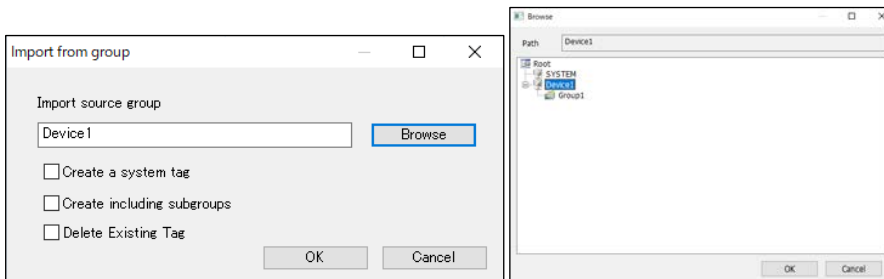
No	Name	Comment	User defined
1	Device1.ArrayTag		
2	Device1.StringTag		
3	Device1.Tag		

<Add by Device or Group>

Right-click on HTTP tag list and click New > Generate from group.



Select the device or group and click OK.

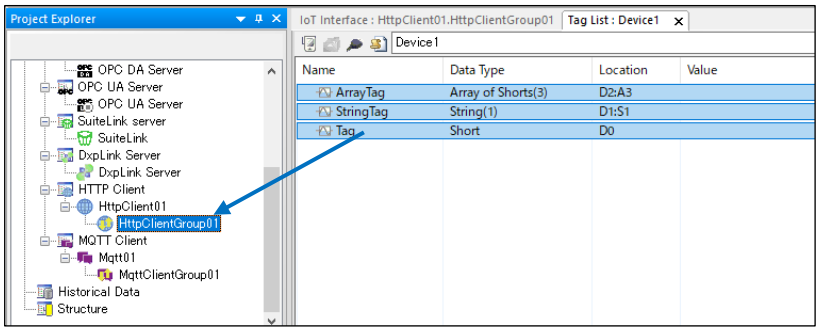


Added Tags to HTTP tag list.

No	Name	Comment	User defined
1	Device1.ArrayTag		
2	Device1.StringTag		
3	Device1.Tag		

4.13.5.4.2 Add by dragging and dropping tags

Add tags to the HTTP tag list by dragging and dropping tags from DataSource tag list to the HTTP group.



IoT Interface : HttpClient01.HttpClientGroup01 x Tag List : Device1			
No	Name	Comment	User defined
1	Device1.ArrayTag		
2	Device1.StringTag		
3	Device1.Tag		

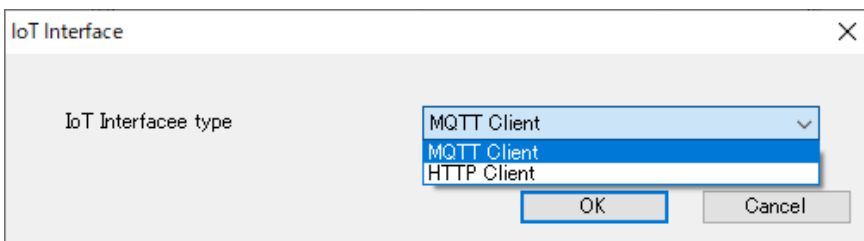
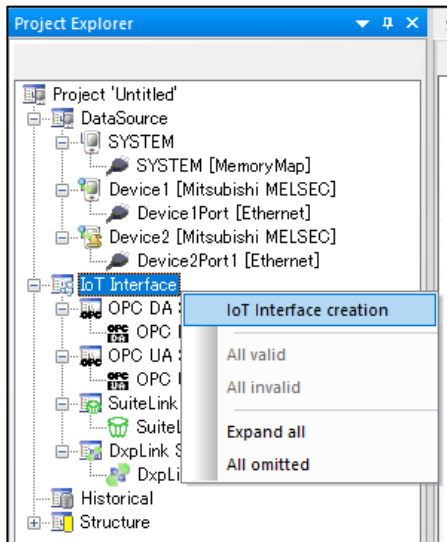
4.13.6 MQTT Client

Configure MQTT client settings.

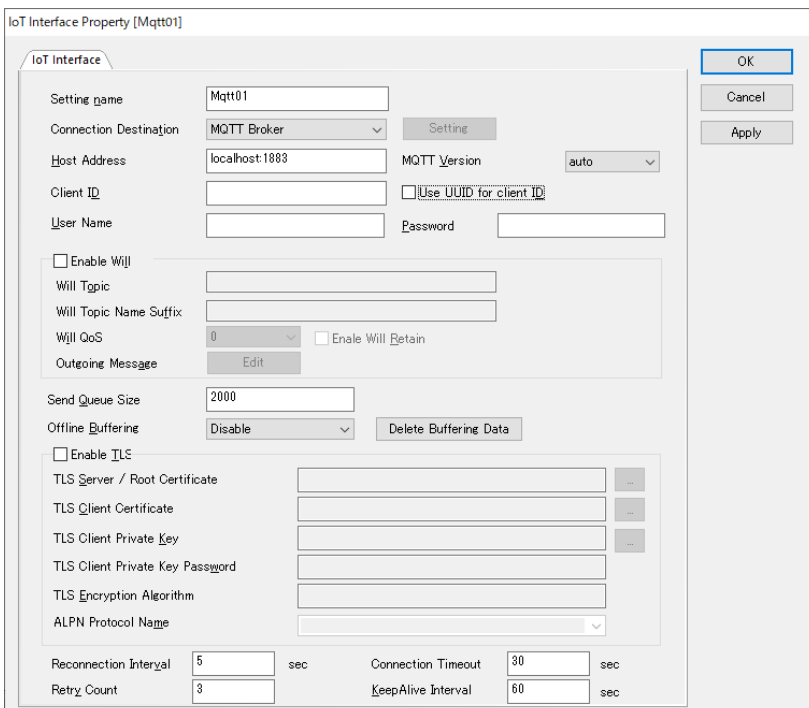
4.13.6.1 MQTT-based Setting

MQTT base configuration.

Right-click "External System Linkage" in the Project Explorer and select "Create External System Linkage". Select "MQTT Client" from "External System Linkage Type" and configure the settings.



Selecting an MQTT client will display its properties.

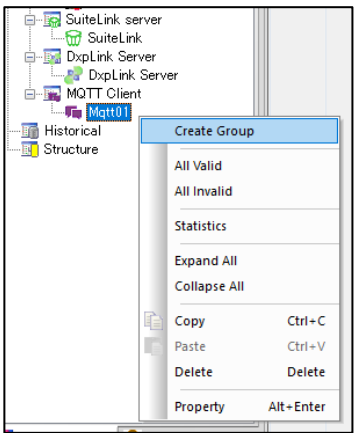


Item	Description
Setting name	Set the name of the external system settings.
Connection destination	Select a connection destination; you can choose an MQTT broker or Azure IoT Hub.
Host address	Enter the host address of the sending server.
MQTT version	Specifies the version of the MQTT specification used for MQTT communication.
Client ID	Specify any identifier that is unique within the broker.
User name	Specifies the login username for the MQTT broker.
Password	Specifies the login password for the MQTT broker.
Will use	
Will topic	Specifies the topic name where the Will message will be left.
Will topic suffix	Specify the Will topic name suffix.
Will QoS	Select QoS0, QoS1, or QoS2.
Outgoing message	Specify the Will message.
Send queue size	Specifies the maximum number of messages waiting to be sent.
Offline buffering	Enable to temporarily buffer messages waiting to be sent. If set to "Always", all messages will be buffered. When set to "Enable in case of communication failure," messages are buffered when the connection with the server times out.
Use TLS	
TLS Server / Root Certificate	Specifies the HTTPS server or root certificate.
TLS Client Certificate	Specifies the device explorer certificate.
TLS client private key	Specifies the private key for DeviceXPlorer.
Client private key password	Specify the client private key if it has a password set.
TLS encrypt algorithm	Specifies the TLS encryption algorithm. See CipherList of OpenSSL. If not specified, "ALL" is assumed.
ALPN protocol name	Specifies the ALPN protocol name.
Reconnection interval	Specifies the interval between reconnections when a connection is dropped.
Connection timeout	Specifies the timeout period for communication with the server.
Send retry interval	Specifies the number of transmission retries when a message transmission with a QoS level (quality of service) of 1 or higher fails.
KeepAlive interval	Specifies the interval between connection checks with the MQTT broker. 0 disables it.

4.13.6.2 MQTT group Setting

MQTT group is performed. This is where you configure the settings for sending data.

Right-click on the MQTT base setting you created and select "Create Group".



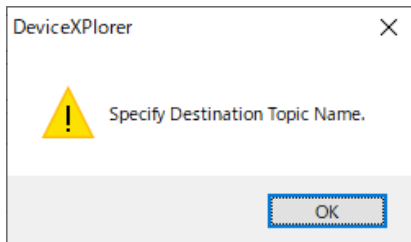
Properties set the period, topic name, etc.

A screenshot of the 'IoT Interface Group' configuration window. The 'Group setting name' is 'MQTTClientGroup01' and it is 'Enabled'. Under 'Publish settings', 'Send Trigger' is set to 'Period' with a 'Cycle' of '1000' msec. 'First Compulsory Establishment' is checked. 'Target Tag' is empty, and 'Change Condition' is 'While ON'. The 'MQTT Settings' section includes 'Destination Topic Name', 'Topic Name Suffix', 'Publish QoS' set to '0', and 'Enabled Retain' (unchecked). The 'Outgoing Message' section has 'Message Type' set to 'String', an 'Edit Outgoing Message' button, and an empty 'Outgoing File Path'. The 'Subscribe settings' section is checked, with 'Incoming topic name' set to 'Topic1' and 'Subscribe QoS' set to '0'. 'Skip null values' is unchecked.

Item	Description
Group setting name	Specifies the group setting name.
Enabled	Enables/disables the operation.
Publish settings	Enable Publish.
Send Trigger	Specify the type of message to be sent.
Cycle	Specifies the message sending cycle.
Target Tag	Specify the target of tag change event.
Change Condition	Specify the event to be executed when the tag value changes in a tag change event.
First Compulsory	The trigger is forced to be established once at the beginning of the operation.
MQTT settings	
Destination Topic Name	Specifies the destination topic name.
Topic Name Suffix	Specifies the topic name suffix.
Publish QoS	QoS0, QoS1, and QoS2 for Publish can be selected.
Enabled Retain	Enable if you want to use the message Retain (to be kept as a cache on the MQTT server) function.

Outgoing Message	
Message type	Select the type of message to be sent.
Outgoing Message	Specify the message to send.
Outgoing File Path	Specify the file path of the outgoing message.
Subscribe settings	Enable Subscribe.
Incoming topic name	Specify the topic name to receive.
Subscribe QoS	QoS0, QoS1, and QoS2 for Subscribe can be selected.
Skip null values	Skip writing to the tag when receiving a message that contains NULL in the value. If it is disable, a write error will occur.

If "OK" is clicked with no topic name specified, the following message is displayed.



4.13.6.3 Outgoing Message(MQTT)

Select Edit Outgoing Message in the MQTT Group Settings to edit the content of the message to be sent. Please refer to 4.13.5.3 Outgoing Message(HTTP).

4.13.6.4 MQTT Tag List

IoT Interface : HttpClient01.HttpClientGroup01 x Tag List : Group1			
No	Name	Comment	User defined
1	Device1.ArrayTag		
2	Device1.StringTag		
3	Device1.Tag		
4	Device1.Group1.Tag		

Item	Description
No	Display the order contained in sending message. The order can be changed by right-clicking.
Name	Display the tag name of the tag that sends and receives.
Attribute	Select attributes for whether to Publish or Subscribe. Select from Publish/Subscribe, Publish Only, and Subscribe Only.
Comment	Display comments set in tag settings. Comments can be included in sending message.
User defined	The user can freely enter characters. User defined can be included in sending message.

4.13.6.4.1 Add from MQTT Tag List

See 4.13.6.4.1 for how to add tags from the MQTT tag list.

4.13.6.4.2 Add by dragging and dropping tags

Add tags to the MQTT tag list by dragging and dropping tags from DataSource tag list to the MQTT group.

Project Explorer

- IoT Interface
 - OPC DA Server
 - OPC DA Server
 - OPC UA Server
 - OPC UA Server
 - SuiteLink server
 - SuiteLink
 - DxpLink Server
 - DxpLink Server
 - HTTP Client
 - MQTT Client
 - Mqtt01
 - MqttClientGroup01
 - Historical Data
 - Structure

Tag List : Device1 x IoT Interface : Mqtt01.MqttClientGroup01

Device 1

Name	Data Type	Location	Value
ArrayTag	Array of Shorts(3)	D2:A3	
StringTag	String(1)	D1:S1	
Tag	Short	D0	

IoT Interface : Mqtt01.MqttClientGroup01 x

No	Name	Attribute	Comment	User defined
1	Device1.Tag	Publish/Subscribe		
2	Device1.ArrayTag	Publish/Subscribe		
3	Device1.StringTag	Publish/Subscribe		

4.13.6.5 Subscribe Message Format

The value of the subscribed message is written to the tag added to MQTT tag list.

A write error will occur when the key name of a tag that attribute is Publish Only is received.

IoT Interface : Mqtt01.MqttClientGroup01 × Tag List : Group1				
No	Name	Attribute	Comment	User defined
1	Device1.Tag	Publish/Subscribe		
2	Device1.ArrayTag	Publish/Subscribe		
3	Device1.StringTag	Publish/Subscribe		
4	Device1.Group1.Tag	Publish/Subscribe		

If the settings are as shown above, Tags are written by receiving the following JSON message.

Specify the name displayed in the tag name for the key name. It is not necessary to match the order displayed in the tag list with the order of the messages.

```
{
    "Device1.Tag" : 1,
    "Device1.StringTag" : "AA",
    "Device1.ArrayTag" : [1, 2, 3],
    "Device1.Group1.Tag" : 100
}
```

Important

Please note that if you receive a JSON message with a hierarchical structure as shown below, tag isn't written.

```
{
    "Device1" : { "Tag" : 1 },
    "Data" : { "Device1.Tag" : 1 }
}
```

4.13.6.6 Connection settings with Azure IoT Hub

This tutorial presents an example of sending data to Azure IoT Hub from the Device Explorer and use the actions function to record the number of times the Azure IoT Hub has been received, and how to receive messages from the Azure IoT Hub at the Device Explorer.

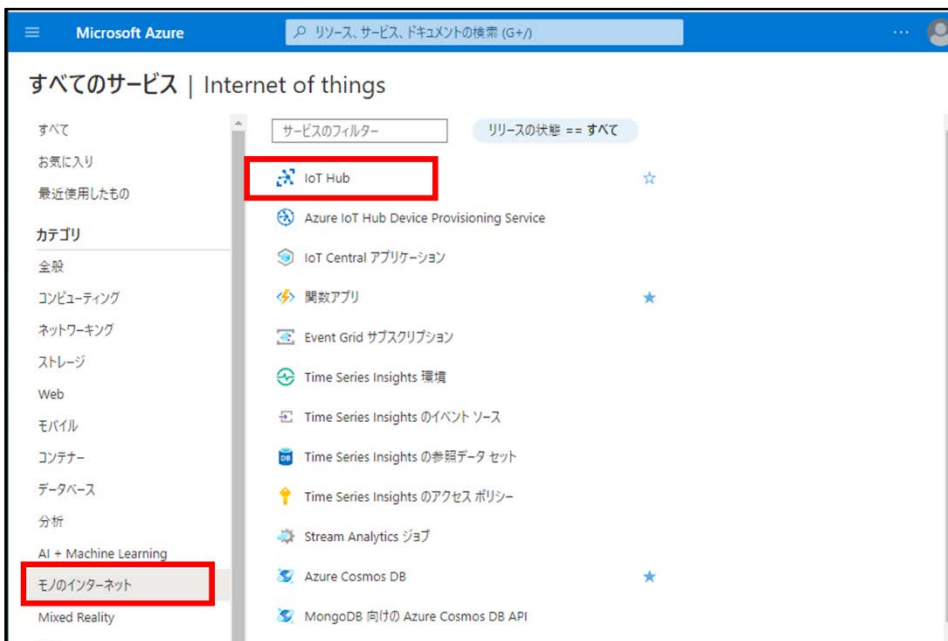
Note

The information, setting procedures, and images included in this report are as of May 2019. The settings and screens may differ when using the product.

4.13.6.6.1 Creating Azure IoT Hub

After logging in to Azure, click Portal to open the Home window.

Then click “All Services > Internet of Thing > IoT Hub”.



Click “Add” to create a IoT Hub.



elect "Subscription", "Region", and enter any "Resource Group Name" and "IoT Hub Name".

This time, the region is selected as "Japan West", the resource group is named as "test", and the IoT Hub is named as "takebishi".

After completing the entry, click "Next: Network".

The screenshot shows the 'IoT Hub' creation page in the Microsoft Azure portal, specifically the 'Basic' tab. The page is titled 'IoT ハブ' (IoT Hub) and 'Microsoft'. Below the title, there are tabs for '基本' (Basic), 'ネットワーク' (Network), '管理' (Management), 'アドオン' (Add-on), 'タグ' (Tag), and '確認および作成' (Review and Create). The '基本' tab is active. The page contains several form fields for configuration:

- サブスクリプション *** (Subscription): A dropdown menu with a red box around it.
- リソース グループ *** (Resource Group): A dropdown menu with 'test' selected and a red box around it.
- IoT Hub 名 *** (IoT Hub Name): A text input field with 'takebishi' entered and a red box around it.
- 領域 *** (Region): A dropdown menu with 'Japan West' selected and a red box around it.

At the bottom of the page, there are three buttons: '確認および作成' (Review and Create), '< 前へ' (Previous), and '次へ: ネットワーク >' (Next: Network).

On the Network tab, specify "Public Access" or "Private Access". In this case, we will set Private Access. After completing the settings, click "Next: Manage".

The screenshot shows the 'IoT Hub' creation page in the Microsoft Azure portal, specifically the 'Network' tab. The page is titled 'IoT ハブ' (IoT Hub) and 'Microsoft'. Below the title, there are tabs for '基本' (Basic), 'ネットワーク' (Network), '管理' (Management), 'アドオン' (Add-on), 'タグ' (Tag), and '確認および作成' (Review and Create). The 'ネットワーク' tab is active. The page contains several form fields for configuration:

- 接続の構成 *** (Connection Configuration): A section with two radio buttons: 'パブリック アクセス' (Public Access) and 'プライベート アクセス (推奨)' (Private Access (Recommended)). The 'プライベート アクセス (推奨)' option is selected.
- このリソースが作成された後に、この設定を変更したり、別の接続方法を構成したりすることができます。** (After this resource is created, you can change this setting or configure a different connection method.)

At the bottom of the page, there are three buttons: '確認および作成' (Review and Create), '< 前へ: 基本' (Previous: Basic), and '次へ: 管理 >' (Next: Manage).

Select "Price and Scaletier" and click "Confirm and Create" to begin deployment.
This time, "F1:free" is set.

Microsoft Azure

ホーム > IoT Hub >

IoT ハブ

Microsoft

基本 ネットワーク **管理** アドオン タグ 確認および作成

それぞれの IoT ハブには、特定レベルの特定数のユニットがプロビジョニングされます。レベルとユニット数によって、1 日に送信できるメッセージの最大クォータが決まります。 [詳細](#)

スケーリング レベルとユニット

価格とスケールティア *

IoT ハブ F1 のユニット数

IoT ハブをスケーリングできる場合を決定します。ニーズが増大した場合は、後でこれを変更できます。

価格とスケールティア	F1	device-to-cloud メッセージ	有効
1 日あたりのメッセージ	8,000	メッセージ ルーティング	有効
月あたりのコスト	0.00 USD	cloud-to-device コマンド	有効
Defender for IoT	対象外	IoT Edge	有効
デバイスの更新プログラム	対象	デバイス管理	有効

ロールベースのアクセス制御

アクセス許可モデルを Azure ロールベースのアクセス制御 (RBAC) のみに、または共有アクセス ポリシーと RBAC の組み合わせに変更します。 [詳細情報](#)

☐ RBAC のみ

確認および作成 < 前へ: ネットワーク 次へ: アドオン >

Click "Create" to begin deployment.

Microsoft Azure

ホーム > IoT Hub >

IoT ハブ

Microsoft

基本 ネットワーク 管理 アドオン タグ **確認および作成**

基本

サブスクリプション [Redacted]

リソース グループ test

領域 Japan West

IoT Hub 名 takebishi

ディザスター リカバリーが有効 はい

ネットワーク

接続の構成 プライベート アクセス

プライベート エンドポイント接続 なし

公衆ネットワーク アクセスを許可する 無効

管理

価格とスケールティア F1

IoT ハブ F1 のユニット数 1

1 日あたりのメッセージ 8,000

Device-to-cloud パーティション 2

月あたりのコスト 0.00 USD

Defender for IoT 無効

タグ

作成 < 前へ: タグ 次へ > Automation オプション

Make sure that the status changes from "Created" to "OK" after deployment is completed.

The screenshot shows the Microsoft Azure portal interface. The main heading is "takebishi-117163314 | 概要". Below the heading, there's a search bar and several action buttons: 削除, キャンセル, 再デプロイ, ダウンロード, and 最新の情報に更新. The main content area shows a green checkmark and the text "デプロイが完了しました". Below this, it says "デプロイ名: takebishi-117163314", "サブスクリプション: [redacted]", and "リソースグループ: test". The deployment started on "2022/11/7 16:33:15". A table shows the deployment details with columns: リソース, 種類, 状態, and 操作の詳細. The table has one row: takebishi, Microsoft.Devices/Iot..., and OK. The status "OK" is highlighted with a red box. Below the table, there's a section for "次の手順" (Next steps) with links for IoT device addition, device messaging rules, and a button to "リソースに移動". On the right sidebar, there are sections for Cost Management, Microsoft Defender for Cloud, and a free Microsoft tutorial.

Click "Home" and click created IoT Hub.

The screenshot shows the Microsoft Azure portal interface for IoT Hub. The main heading is "IoT Hub". Below the heading, there's a search bar and several action buttons: + 作成, ビューの管理, 更新, CSV にエクスポート, クエリを開く, and タグの割り当て. There's a filter bar with "任意のフィールドのフィルター...", "サブスクリプション 次の値と等しいすべて", "リソース グループ 次の値と等しいすべて", "場所 次の値と等しいすべて", and "フィルターの追加". Below the filter bar, there's a table with columns: 名前, 種類, リソース グループ, 場所, and サブスクリプション. The table has one row: takebishi, IoT Hub, test, Japan West, and [redacted].

You can check the "Host Name" for using in the DeviceExplorer settings.

The screenshot shows the Microsoft Azure portal interface for the details of the "takebishi" IoT Hub. The main heading is "takebishi IoT Hub". Below the heading, there's a search bar and several action buttons: 移動, 削除, 最新の情報に更新, and フィードバック. There's a warning banner about IoT Hub and Data Protection Manager. Below the banner, there's a section for "基本" (Basic) with fields: リソース グループ (test), 状態 (Active), 場所 (Japan West), サービス リージョン (Japan West), サブスクリプション ([redacted]), タグ ([redacted]), and タグを追加するにはここをクリック. The "Host Name" field is highlighted with a red box and contains the value "takebishi.azure-devices.net". Other fields include 価格とスケールティア (F1 - 無料) and IoT Hub のユニット数 (1).

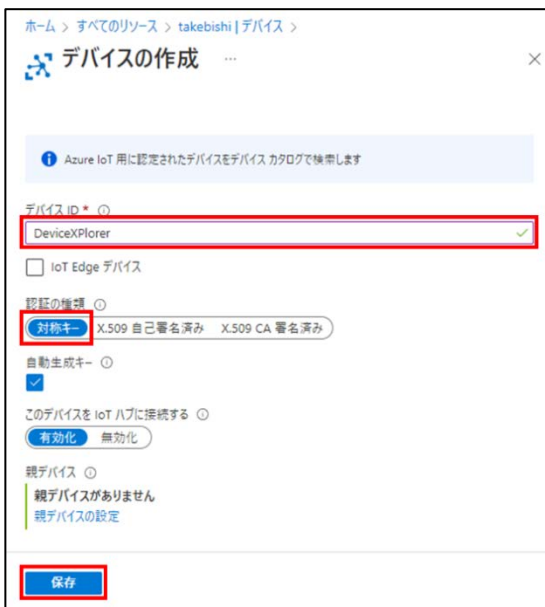
Create a IoT device by clicking “Device > Adding Devices”.



Enter an arbitrary Device ID and set the authentication type to "Symmetric Key".

This time, the Device ID is "DeviceXPlorer".

After completing the input, click “Save”.



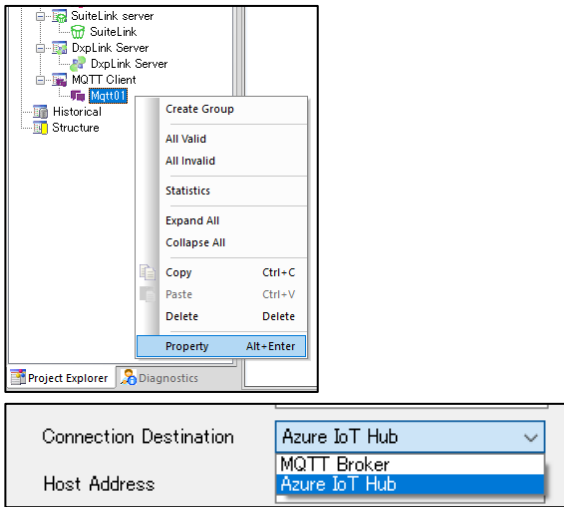
When you select the created Device, you can check the Device ID and main key for using in the DeviceXPlorer settings.



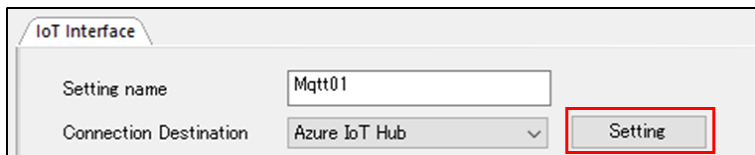
4.13.6.7 Azure IoT Hub connection settings in DeviceXPlorer.

Configure Azure IoT Hub connection settings in DeviceXPlorer.

Select Properties from the newly created MQTT settings for external system integration. Select Azure IoT Hub from the connection type.



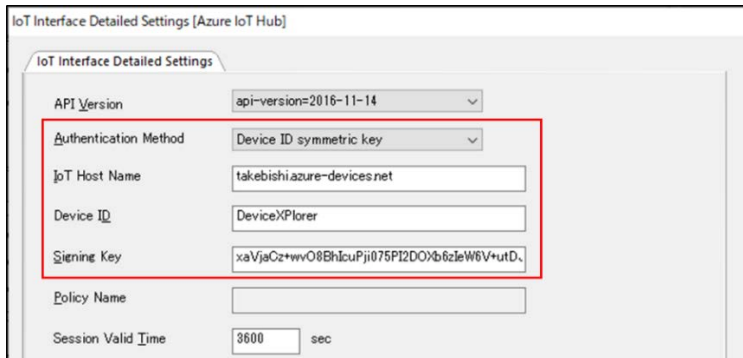
Click Settings to configure the connection settings with Azure IoT Hub.



This time, use "Device ID Symmetric Key" for "Authentication Method".

Enter the "IoT Host Name," "Device ID," and "Signing Key" as they appeared in 5.13.6.3.1, respectively.

*For "Signature Key," enter the primary key.



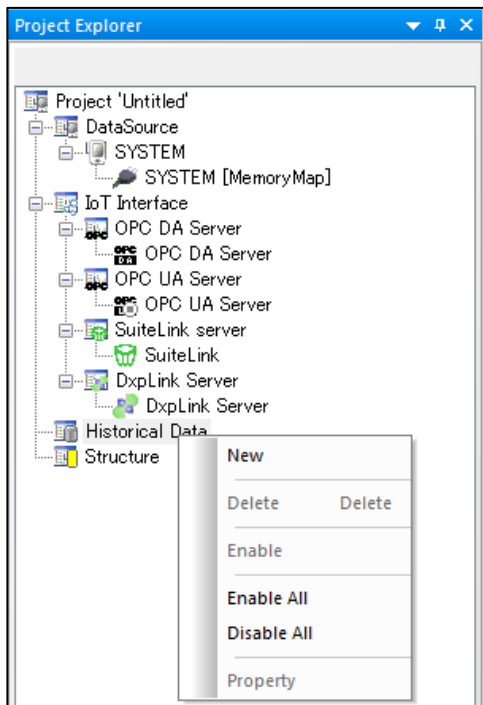
Item	Description
API Version	Select the API version of Azure IoT Hub.
Authentication Method	Select the authentication method for Azure IoT Hub.
IoT Host Name,	Specify the IoT host name of Azure IoT Hub.
Device ID	Specify the Device ID of Azure IoT Hub.
Signing Key	Specify the signing key for Azure IoT Hub.
Policy Name	Specify the policy name of Azure IoT Hub.
Session Valid Time	Set the session validity time. After the set time has elapsed, it will reconnect voluntarily.

After creating the MQTT base configuration, refer to 5.13.6.2 to configure it to send data to Azure IoT Hub.

4.14 Historical Data

Configure historical data. The historical values of tags can be logged on the database, and OPC UA Historical Access makes it possible to refer to and change historical data.

Right click on "Historical Data" in the Project Explorer and select "New".



When a new one is created, the Historical Property Settings shows.

Property [HistoricalGroup1]

Historical group

Historical Group Name: HistoricalGroup1

☒ Enable

Communication Cycle: 1000 msec

Logging Monitoring Cycle: 1000 msec ☒ Data change logging

File Path:

Maximum Number of Records: 100000

Stop Disk Size (MB): 1000

Delete All Records

OK, Cancel, Apply

Item	Description
Historical group name	Set the historical group name.
Enable	Enable logging for historical groups. Enabling it will start logging tags that belong to the historical group.
Communication cycle	Specify the communication cycle of tags belonging to the historical group. If the tag is already communicating, the cycle with the earliest communication cycle is given priority.
Logging monitoring cycle	Set the interval for logging the tag data. Important Tags belonging to the historical group are temporarily accumulated in the buffer when the tag is read or written. The accumulated buffer is logged to the database in batch at the interval set in the logging monitoring interval.
Data change logging	Log when a tag's value or quality flag changes.
File path	Specifies the file path of the database that stores tag data.
Maximum number of Records	Specifies the maximum number of records of historical values that can be logged to the database. If the history value exceeds the maximum number of records, the oldest data is deleted and the new data is logged.
Stop Disk Size(MB)	Specify free space of the disk in which Log Message are saved in MB. Stops saving when the remaining disk capacity becomes less than the specified capacity.
Delete all records	Deletes all records (history values) stored in the database.

Important
Up to 64 historical groups can be created.

4.14.1 Historical Tag Setting

Add tags to the historical group you created. By adding it, it becomes possible to save the history value of the tag. Please note that method tags cannot be added to historical groups.

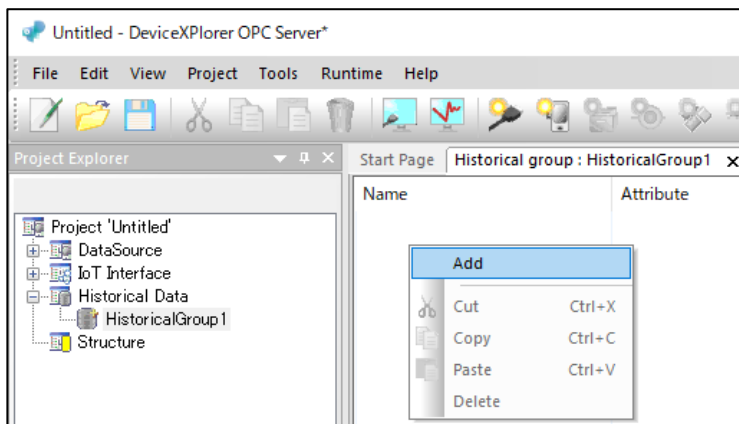
Important

Up to 1024 tags can be added per group.

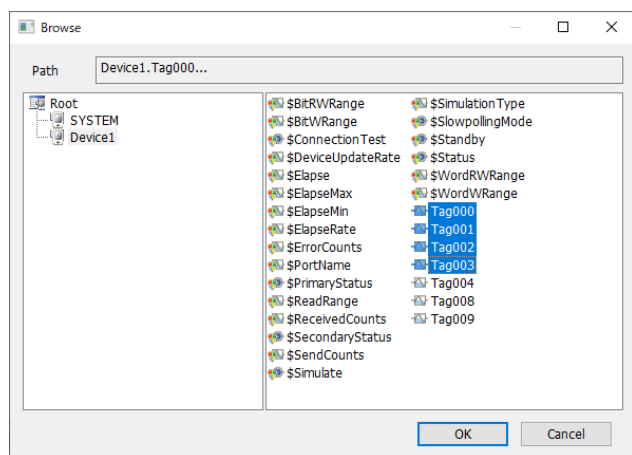
4.14.1.1 Add from Historical Tag List

Clicking on the historical group displays the historical tag list.

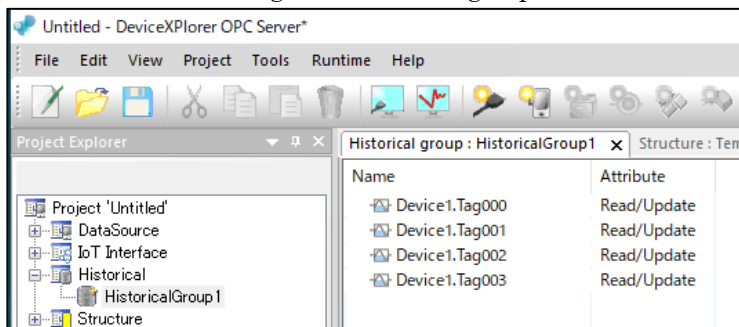
Right-click on the Historical Tag List and select Add.



You can see the tags you have set for the data source on the browse dialog and add the specified tag to the historical group.

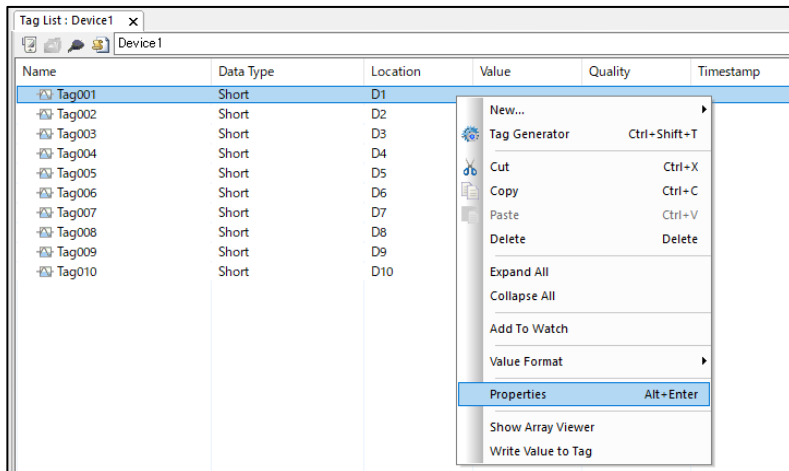


Click "OK" to add the tag to the historical group.

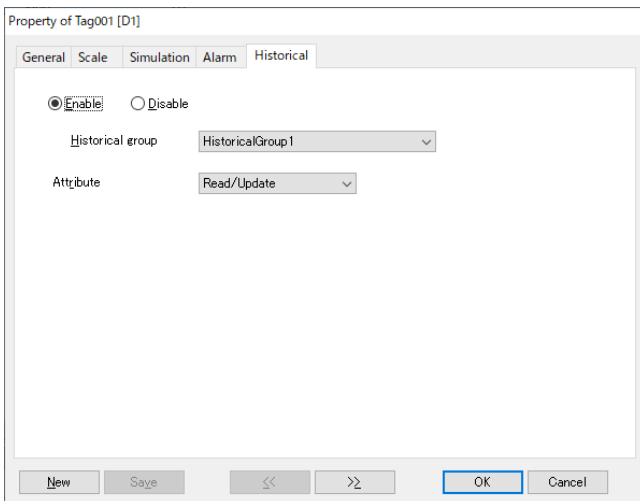


4.14.1.2 Add from tag setting

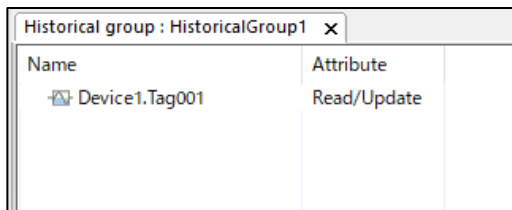
Right-click the tag you want to add to the historical group from the tag list and open the property.



Enable and select the historical group to add the tag from the Historical tab.



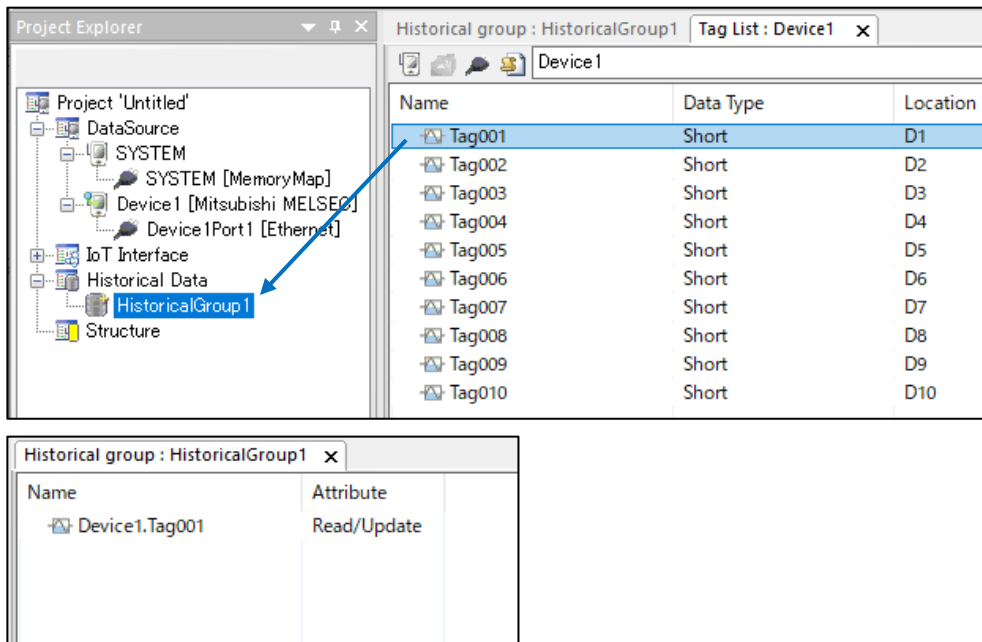
Click OK to add the tag to the historical group.



4.14.1.3 Add by dragging and dropping tags into the historical group

Select the tag you want to add to the historical group from the tag list.

A tag is added by dragging and dropping it to the historical group to be added.



4.15 Project

Project settings consist of common properties and configuration properties. Setting items that affect the entire application are set as "Common Properties" and items limited to each project are set as "Configuration Properties". The project settings screen can be opened from "Properties" in the "Project" menu or "Options" in the "Tools" menu.

Property of Untitled

Common Properties Project Properties

☐ General

Start as Service Program	Execute as Normal Application
Running Processor Setting	None
Language	English
SuiteLink Application Name	DXPV7
Connect with the runtime on startup	Enable
Maximum number of log messages to manage	100000

☐ Setting Tool

Monitor Rate (msec)	600
Automatic Login User	Administrator
Project file save folder	
Startup File	

OK Cancel Apply

Note

DeviceXPlorer settings are saved to a specified folder as files with the extension "*.dvp".

4.15.1 Common Properties, General

This is where you set the parameters that affect the software as a whole.

Item	Default	Reboot	Description
Start as Service Program	Execute as Normal Application	Need	If this is “Service Program”, DeviceXPlorer is run as a Windows service program.* Service Program: Run as Service. Execute as Normal Application: Run as Normal application.
Running Processer Setting	(No specification)	---	Specifies running CPU processer of DeviceXPlorer. If this setting is not specified, DeviceXPlorer runs with all CPU processer.
Language	English	---	Specifies the language of the application. You can specify “Japanese”, “English” or “Chinese (Simplified)” or “Chinese(Traditional)” or “Korean(Hangul)”.
SuiteLink Application Name	DXPV7	Need	Specifies the SuiteLink application name.
Maximum number of log messages to manage	100000	---	Specify the maximum number of log messages to be managed. (Setting range: 100 to 9999999)
Startup File	(No specification)	Need	Specifies the project file to be loaded automatically when DeviceXPlorer starts. If there is no specification, the last project file used will be loaded.

* When connecting with OPC server which runs as windows service, the DCOM configuration is required. About the DCOM configurations, please refer to 4.24.

4.15.2 Common Properties, Settings

Item	Default	Reboot	Description
Monitor Rate	600 (ms)	---	Specify the interval of monitor function and diagnostics view on OPC Server. CPU usage rises when you set small value. Set recommendable value more than 500msec.
Auto Login User	Administrator	Need	Select the login user when DeviceXPlorer starts.
Project file save folder	My Document	---	Specify the folder for saving project file. If you specify empty string, project folder will be My Document/TAKEBISHI/Project folder.
Always connect with the Runtime	Enable	---	Specify whether to automatically connect to the Runtime. When enabled, the runtime will automatically reconnect to the runtime even if it is disconnected due to a connection is lost or other reason. If the user stops or disconnects the runtime by operating, it will not be reconnected.

4.15.3 Project Properties, General

This is where you set the parameters that only affect the project.

Item	Default	Reboot	Description
Reflect Poke	Reflect cache memory immediately	---	Select the timing that tags cache value is updated on poke. “Reflect cache memory immediately” Immediately when DeviceXPlorer receives a write request . “Reflect cache memory after writing” After execute poke and poke is success. “Not reflect cache memory” Not update cache value on poke.
The Priority of the Writing in processing	Giving Priority to Writing	Need	If you select “Giving Priority to Writing”, when writing is requested, writing is performed before reading. You can specify “Giving Priority to Writing” or “Giving Priority to Reading”.
Communication per Client	Disable	Need	Communicate by a client and a subscription group unit and perform movement of the DXP2007 equivalency. Communication load becomes higher than default(Invalidity). If use this function, Slow polling mode is not available.
Multi Access Mode	Enable	Need	Only when “Communication per Client” is “Enable”, this setting is effective. If the multi-access mode is on, when a tag that covers several devices is registered for a single OPC group, parallel processing is performed for each device, which makes it possible to increase throughput. If the multi-access mode is not used, communication proceeds in order one device at a time, so that if a communication failure occurs for one device, the communication processing for the other devices is delayed. You can specify “Enable” or “Disable”.
Async Update Mode	Enable	Need	If the asynchronous update mode is on, a DataChange event occurs for the tags that have been read, out of the tags registered for a single OPC group. You can specify “Enable” or “Disable”.
Multi Access for Sync Async	Disabled	Need	Only when “Communication per Client” is “Enable”, this setting is effective. Specify whether to use multi access for sync or async request from clients. If this setting is enable, many threads are made for reading and writing.
Multi Access Method for Sync Async	Read/Write	Need	Specify the process for multi-access, when Multi Access for Sync Async is enabled.
Write Optimization Type	Request Order	Need	Specify the method of optimization about communicating with PLC to write value. You can select “Request Order”, “Address Order” or “Optimization Priority”.
Reduce write-queue	Omit same value	Need	Specify the method to reduce writing-queue of async request. This setting influences some internal handling of GUI, script, OPC UA, DxpLink. It do not influence the sync or async writing from OPC DA client, and the writing

			from SuiteLink client. 'Omit same value' DeviceXPlorer omits requests of writing same value to the same tag in once, so you can reduce the load of the PLC communication appropriately (Default). 'Omit except latest value' DeviceXPlorer omits requests of writing value to the same tag in once. DeviceXPlorer writes only latest value, so you can reduce the load of PLC communication the maximum. Because the old requests in the writing-queue is not written, you can choose this if you allow omission. 'No Reducing' DeviceXPlorer writes all requests without omitting anything. In the case which a writing speed to PLC is not enough for, the full error of the writing-queue occurs and may not be accepted a new request of writing.
Simulation Max	100	---	Specify max value to change value at simulation mode (Sine / Ramp / Random).
Simulation Cycle	1000	---	Specify cycle to change value for Sign and Ramp type at simulation mode.
Write non-numeric values	Disable	---	Specify whether to write non-numeric values(NaN/Infinite) to floating type tags or not.
Skip bridging for tag which is bad quality	Does not skip bridging	---	About bridge function when source tag is bad quality, copying process of this tag is skipped.
Reset of the System Tag(Statistics)	None	---	Set the reset timing of the System Tag (\$ElapseMin,\$ElapseMax,\$Elapse,\$ErrorCounts,\$SendCounts,\$RecievedCounts).
Script Garbage Collection	5000	---	Run the Full-Garbage-Collection-Cycle of script (Lua) per specified interval. If 0 is set, user can control garbage collection by self.
Auto Port Close	Close	---	If you select "Close", the ports will close when there is no access from a client. This improves convenience when more than one application is using a port such as RS232C that must be accessed exclusively. You can specify "Close" or "Not Close".
Acceptable Max Count of Async Demand (Common)	1,000	Need	Sets the maximum number of asynchronous requests (Script, DxpLink etc.) that can be accepted. If asynchronous requests continue to be displayed error message in message view.
Protocol Timer Tick	10 (ms)	Need	Sets the operating interval for sending and receiving messages. If the value is too small, it increases the CPU usage rate. Normally, a setting within the range of 10 - 100 ms is recommended.
Process Priority	Normal	Need	Specifies the process priority. [High] [Low] REALTIME > HIGH > NORMAL > LOW
Namespace Delimiter	. (Dot)	Need	Specifies the delimiter for Item ID.

Extend Tag Delimiter	: (Cologne)	Need	Specifies the delimiter for extended tag name of dynamic tag.
----------------------	-------------	------	---

4.15.4 Project Properties, GUI

This sets the parameters related to the display.

Item	Default	Reboot	Description
Writing Tag Value From GUI	Permit Writing	---	Allows the operator to write values in the Tag View. You can specify "Prohibit Writing" or "Permit Writing".
Keep Monitor Value	Keep	---	Keeps the current value of the displayed tag if the monitoring is stopped in Tag View. If the value is not kept, it is cleared. You can specify "Keep" or "Clear".
Task Tray	Not Put in	---	If you select "Put in", when the window is minimized, it is not shown in the task bar but is stored in the task tray. You can specify "Put in" or "Not Put in".
Max Display Message	1,000	Need	Sets the maximum message lines shown in Message View.
Enable Unknown ItemID	Enable	---	Allows access to items that have not been registered in advance. You can specify "Enable" or "Disable".
Auto Add Item	Not Add	---	If you select "Yes", when an unregistered item is accessed, the item is automatically registered. You can specify "Add" or "Not Add".
Display System Tag	Hide	---	Shows system tags in Tag View. You can specify "Hide" or "Display".
Starting Window Style	Last Style	---	Specifies the window style of when the application starts. Last Style Stores the window style. Normal: Always Window. Minimize: Always Minimize.
COM Starting Windows Style	Minimize	---	Specifies the window style of when the application is started automatically by COM.

* The settings you make for "Use of unregistered items", "Automatic item registration", and "Show system tags" become the defaults for newly created devices. If you want to change the settings for each device, you can change them on the Device Properties screen

4.15.5 Project Properties, Log

This sets the parameters for logging messages shown in Message View.

Item	Default	Reboot	Description
Auto Message Save	Not Save	---	It is possible to automatically save messages output to Message View. Messages are saved even during a pause in the displayed messages. You can specify "Not Save" or "Save".
Storage Period	0 (days)	---	The stored data is deleted when the specified storage period expires. If you specify "0", the data is not deleted.
Stop Disk Size	100 (MB)	---	If the remaining disk space falls below the specified amount, file saving stops.
Max File Size of Message Log	99(MB)	---	A name is created for the log file in the "YYYYMMDD-nnn.LOG" format, and when the maximum size is reached, a new file is created. The number nnn is an automatically attached branch number between 001 and 999. If the branch number exceeds 999, the file name becomes YYYYMMDD-1000.LOG. If you specify "0", the file size will not be checked.
File Path of Saving Message Log	(Current folder)	---	Specifies the path for storing log data. You can specify "Output" or "Not Output".
System Log output level	Level3	---	Specifies the system log output level.
DataSource communication output level	Level3	---	Specifies the DataSource log output level.
Script log output level	Level3	---	Specifies the Script log output level.
Historical log output level	Level3	---	Specifies the Historical log output level.
OPC Classic communication output level	Level3	---	Specifies the OPC Classic log output level.
OPC UA communication output level	Level3	---	Specifies the OPC UA log output level.
DxpLink communication output level	Level3	---	Specifies the DxpLink log output level.
SuiteLink communication output level	Level3	---	Specifies the SuiteLink log output level.
HTTP Client communication output level	Level3	---	Specifies the HTTP Client log output level.
MQTT Client communication output level	Level3	---	Specifies the MQTT Client log output level.

Important

Setting the log output level to "level 4" or higher will significantly degrade performance.

4.16 Service Activation

DeviceXPlorer can be started as a Windows service program.

Important

Please configure after start as “Run as administrator” when UAC is ON in on Windows Vista or later OS.

DeviceXPlorer runs on the service session (session 0) of Windows when you use service in Windows Vista or later.

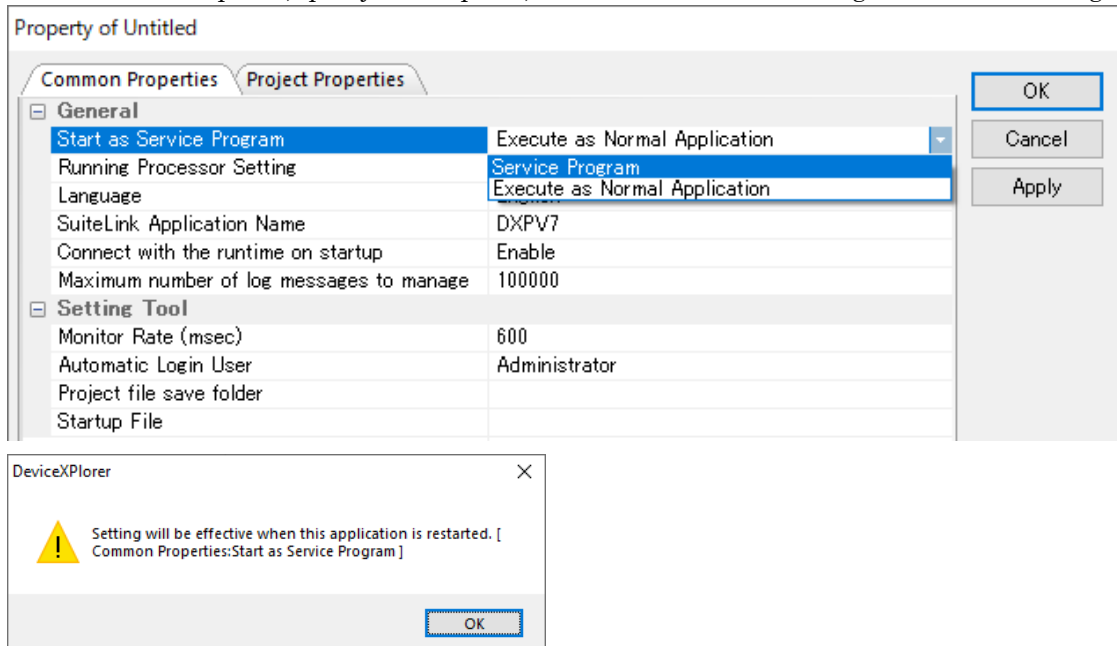
SuiteLink clients running on other sessions such as desktop cannot connect to DeviceXPlorer.

MELSEC EZSocket, MELSEC interface board and Mitsubishi CNC connection do not work in running as service.

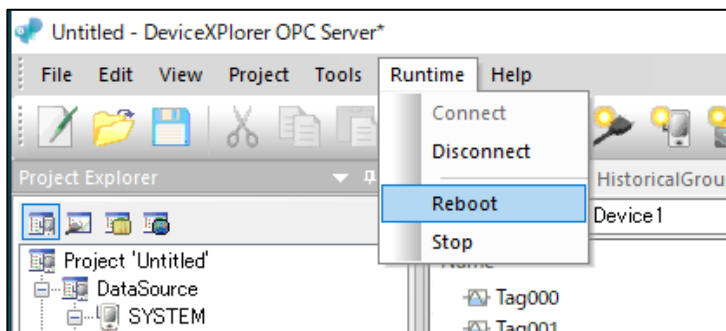
DCOM setting will be initialized if you register / unregister service.

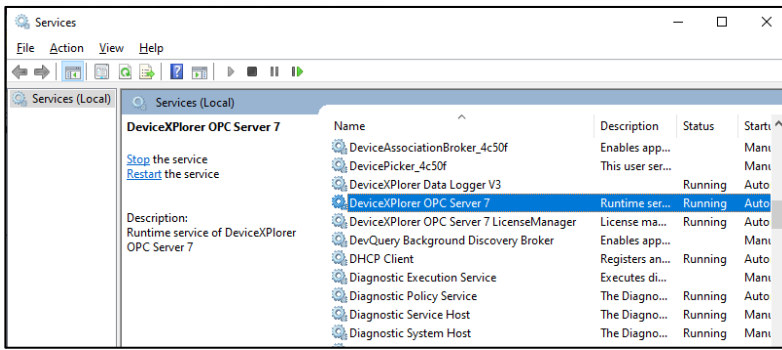
4.16.1 Registering DeviceXPlorer as a service program

(1) Go to Tools -> Options, specify “Startup file”, and set “Start as Service Program” to “Service Program”.

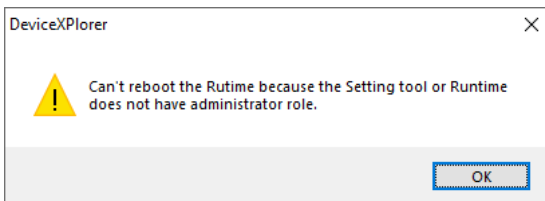


(2) Select "Restart" from the "Runtime" menu to restart the runtime, and it will be registered as a service program. You can check it from "Administrative Tools" -> "Services" in the control panel.

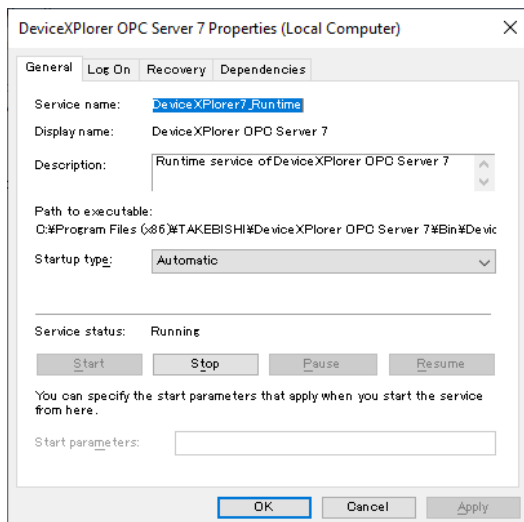




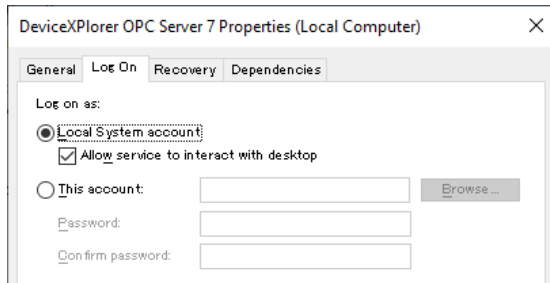
If the following dialog box appears, start DeviceXplorer by "Run as administrator".
 *"DeviceXplorer_Runtime.exe" must be started as administrator.



(3) Open the "DeviceXPlorer OPC Server 7" properties screen from "Services" in the control panel.



(4) Open the Logon tab and validate the check of "Allow service to interact with desktop" checkbox.

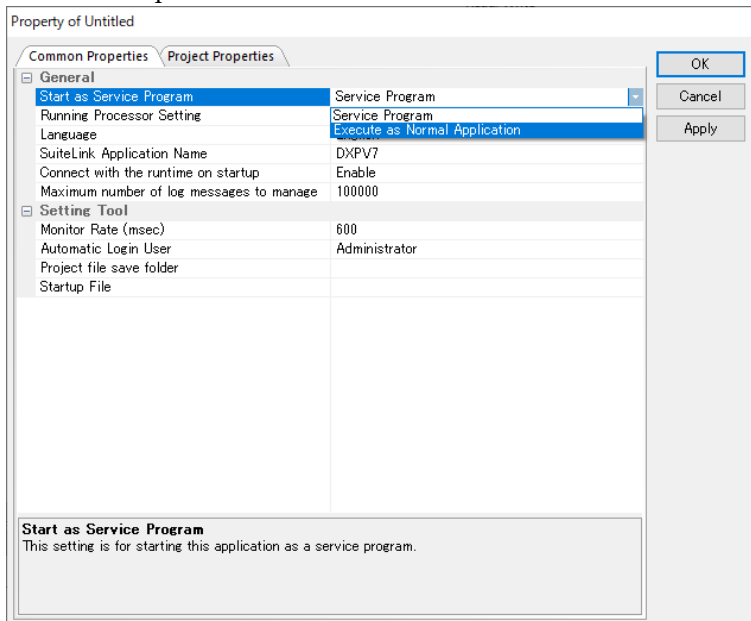


Important

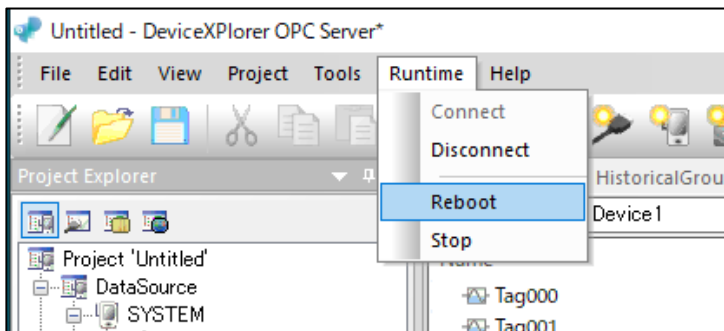
- When connecting with OPC server which runs as windows service, the DCOM configuration is required. About the DCOM configurations, please refer to 4.24.
- In Windows10, please clear the check box for "Allow service to interact with desktop".

4.16.2 How to show DeviceXPlorer setting dialog of service program

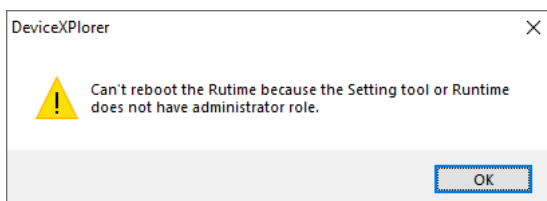
- (1) Set "Service startup" to "Start as a normal application" in "Option settings" under "Tools" in the service started device explorer.



- (2) Select "Restart" from the "Runtime" menu to restart the runtime and it will start as an application.

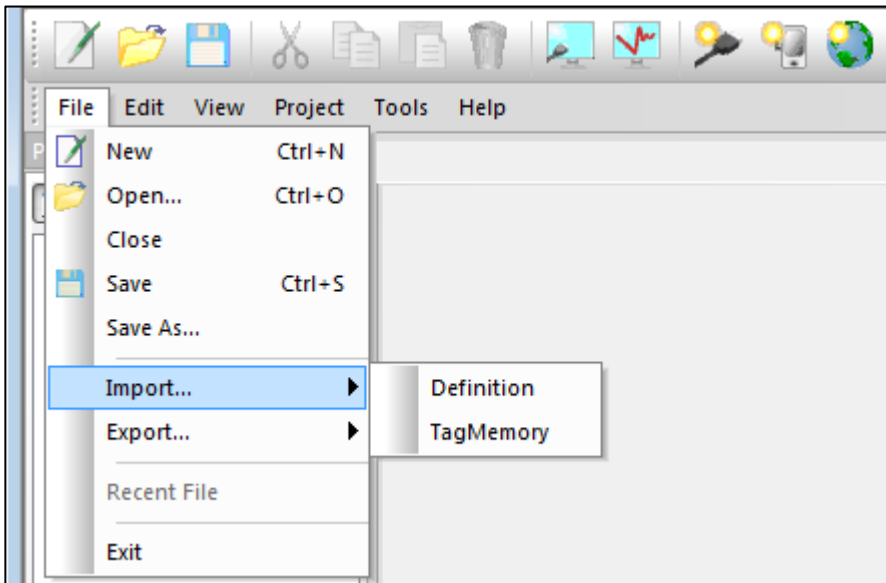


If the following dialog box appears, start Device Explorer by "Run as administrator" and then re-configure the settings.
*Because the runtime needs to be started as an administrator, terminate the task "DeviceXplorer_Runtime.exe" from the task manager and start "DeviceXplorer_Setting.exe" again.



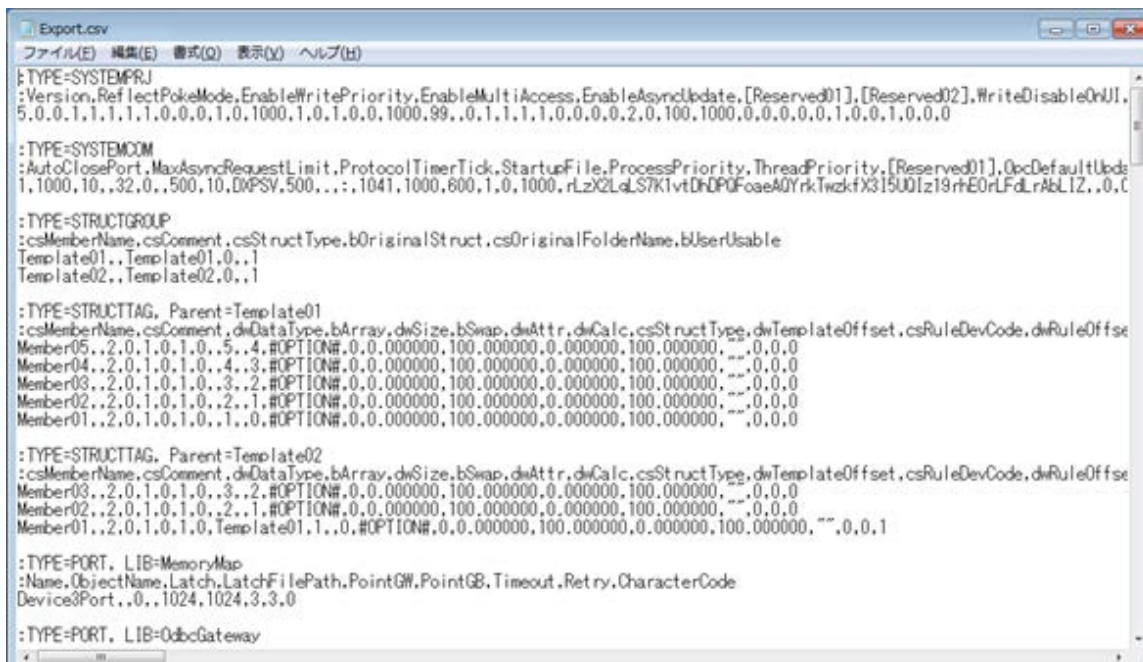
4.17 Importing and Exporting

[Import / Export] from File menu, It is possible to output a CSV format or XML format. OPC server to do the export of the tag memory is holding. It is possible import to use at offline mode.



Important

This is can Import and Export file create at Ver.4. But this cannot import CSV file is created by before Ver.3 OPC Server. When CSV file created by Ver.3, please convert at Project Converter[All Program]-[DeviceXPlorer OPC Server 7] - [Project Converter].

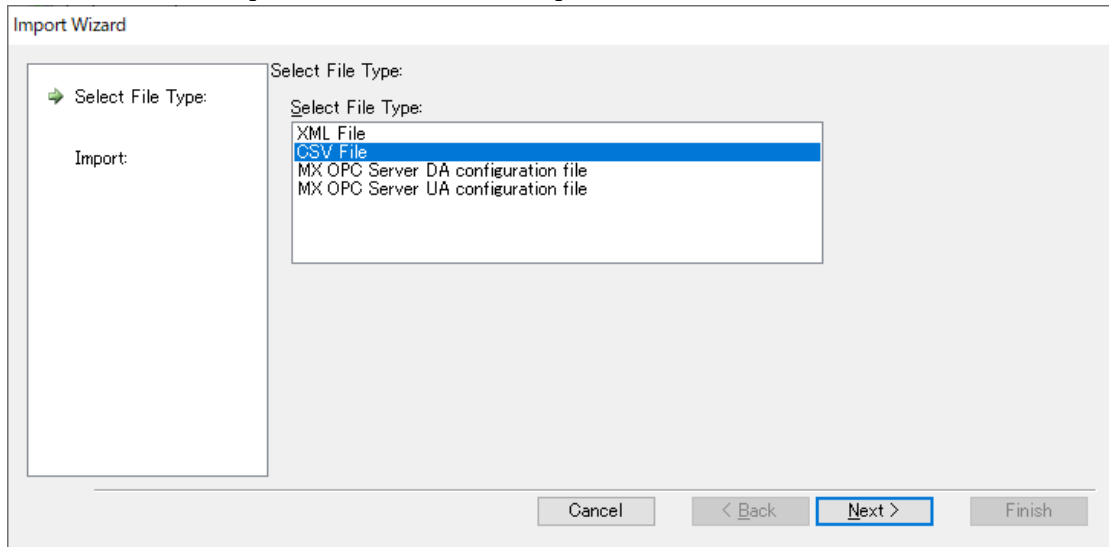


Note

When edit to file , please careful attention.

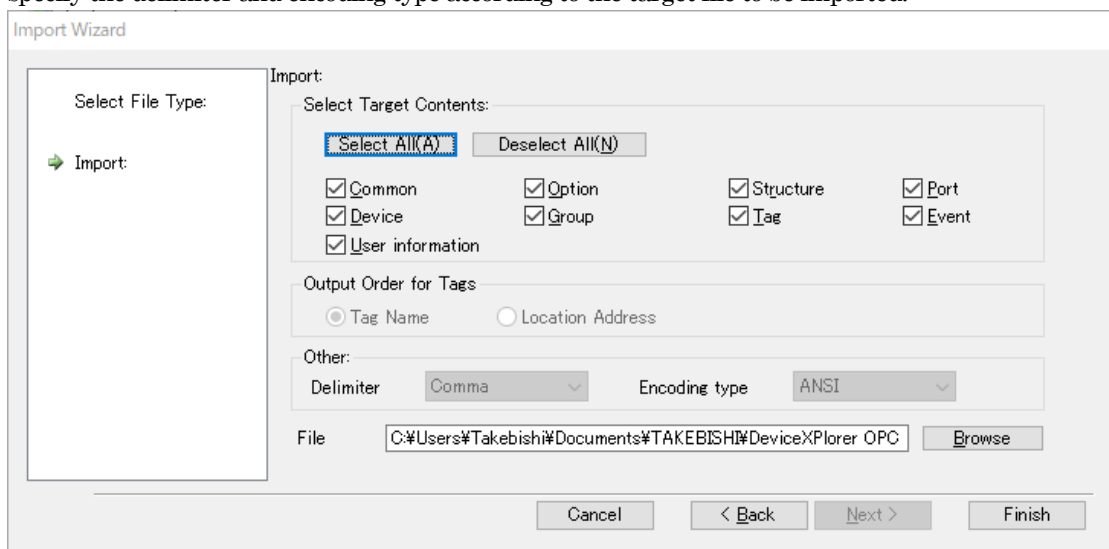
4.17.1 Import[Definition]

DeviceXPlorer can import CSV file or XML file exported from DeviceXPlorer.



Select target contents and file path. When import, cannot specify the order of output tags.

When importing the MX OPC Server DA configuration file or MX OPC Server UA configuration file, specify the delimiter and encoding type according to the target file to be imported.



4.17.1.1 Import MX OPC Server DA/ UA Configuration File

Import the CSV file or TEXT file exported by MX OPC Server DA/ UA.

Overview	
Product Name	MELSOFT MX OPC Server DA/ UA
Version	Ver.7.05(DA) / Ver3.06(UA)
Supported Device	MX Device / MX Device(ver.2)

Important

- To use this function, Melsec Ethernet connection (UA only) / Melsec EZ Socket connection function is required.
- The setting to be imported is Device setting, Group setting, and Tag setting.
Only the Port setting is generated in the default setting. Set the port before communication.

【Device Type Conversion Table】

MX OPC Server DA	DeviceXPlorer OPC Server	Explanation
B	B	Link relay
CC	CC	Counter coil
CN	CN	Counter Present value
CS	CS	Counter Contact
D	D	Data register
EG	EG	GOT register
F	F	Annunciator
L	L	Latch relay
LCC	LCC	Long counter coil
LCN	LCN	Long counter current value
LCS	LCS	Long counter contact
LSTC	LSTC	Long retentive timer coil
LSTN	LSTN	Long retentive timer current value
LSTS	LSTS	Long retentive timer contact
LTC	LTC	Long timer coil
LTN	LTN	Long timer current value
LTS	LTS	Long timer contact
M	M	Internal relay
R	R	File register
SB	SB	Link special relay
SD	SD	Special register
SM	SM	Special relay
STC	SC	Retentive Timer Coil
STN	SN	Retentive Timer Present value
STS	SS	Retentive Timer Contact
SW	SW	Link special register
TC	TC	Timer Coil
TN	TN	Timer Present value
TS	TS	Timer Contact
W	W	Link register
WR	WR	CC-Link input register
WW	WW	CC-Link output register
X	X	Input relay
Y	Y	Output relay
ZR	ZR	File register

【Data Type Conversion Table】

MX OPC Server DA	DeviceXPlorer OPC Server	Explanation
INT	SHORT	Signed 16-bit integer.
UINT, WORD	USHORT	Unsigned 16-bit integer.
REAL	FLOAT	32-bit floating point (IEEE).
LREAL	DOUBLE	64-bit floating point (IEEE).
BOOL	BOOL	Digital, one bit.
UDINT, DWORD	LONG	Unsigned 32-bit integer.
DINT	ULONG	Signed 32-bit integer.
COUNTER	SHORT	Signed 16-bit integer (Counter).
TIMER	SHORT	Signed 16-bit integer (Timer).
RTIMER	SHORT	Signed 16-bit integer (Retentive timer).
UTIMER	USHORT	Unsigned 16-bit integer (Unsigned timer).
UCOUNTER	USHORT	Unsigned 16-bit integer (Unsigned counter).
URTIMER	USHORT	Unsigned 16-bit integer (Unsigned retentive timer).
LTIMER	ULONG	Unsigned 32-bit integer (Long timer).
LCOUNTER	ULONG	Unsigned 32-bit integer (Long counter).
LRTIMER	ULONG	Unsigned 32-bit integer (Long retentive timer).
STRING	STRING	Text stored internally as a 16-bit integer (ASCII, 2 x 8-bit characters per word).
WSTRING	STRING	Text stored internally as a 16-bit integer (UNICODE, 1 x 16-bit character per word).

In MX OPC Server, the BOOL array tag is 1 bit per element. (In DxpSERVER , BOOL array tag is 16 bits per element)

When importing a BOOL array tag, convert it to cover all the devices with the BOOL type array tag set in MX OPC Server .

Example 1:

MX OPC Server (before conversion): M7 Bool array size 15 Read range: M7 to M21 (15bit)

DxpSERVER (after conversion): M0 Bool Array size 2 Read range: M0 to M32 (32bit)

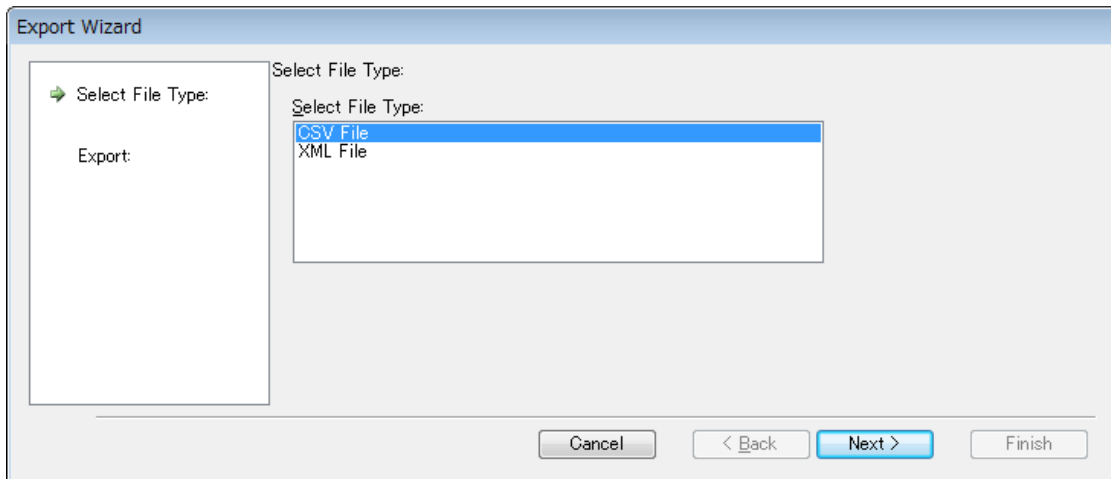
Example 2:

MX OPC Server (before conversion): D0.F Bool Array size 3 Read range: D0.F to D1.1 (3bit)

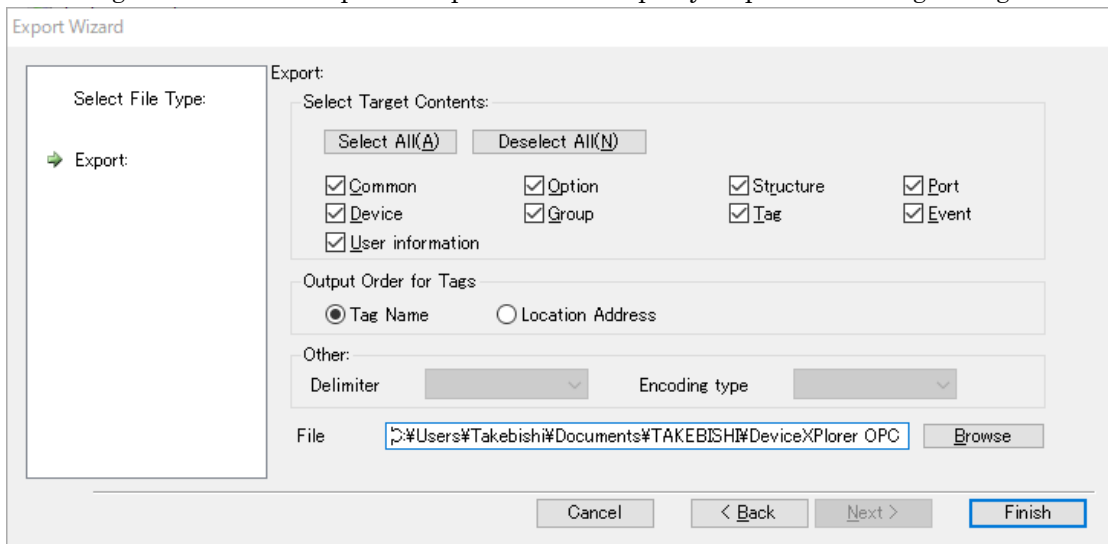
DxpSERVER (after conversion): D0 Bool Array size 2 Read range: D0.0 to D1.F (32bit)

4.17.2 Export[Definition]

DeviceXPlerer can export currently opened project file as CSV or XML format.

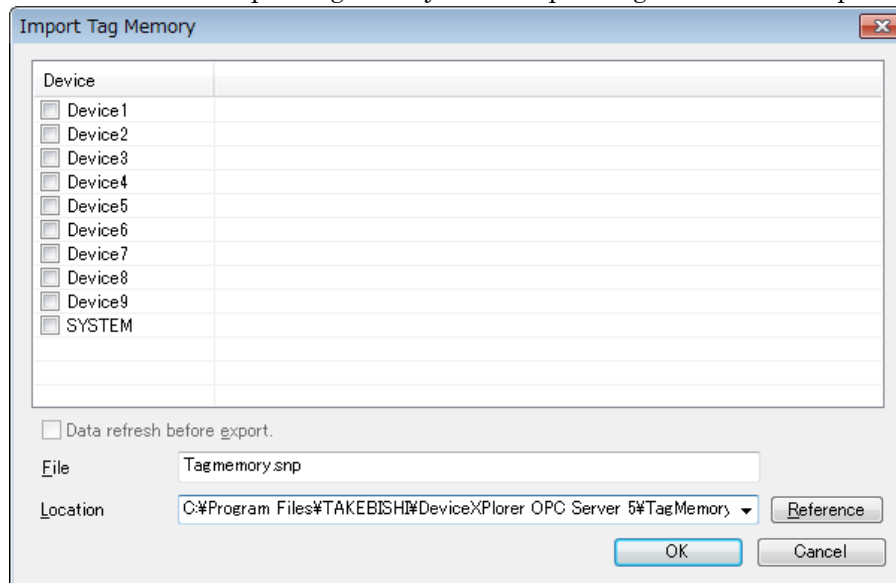


Select target contents and file path to export. And it can specify output order for tags as tag name or location address.

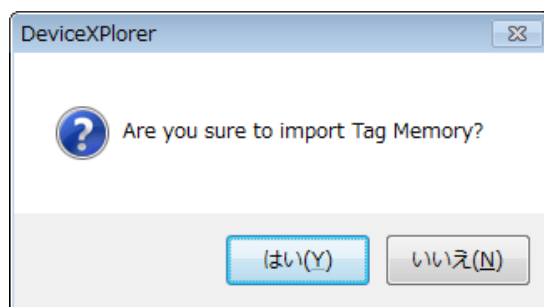


4.17.3 Import[TagMemory]

DeviceXPlorer can import TagMemory. Select import target device and file path.



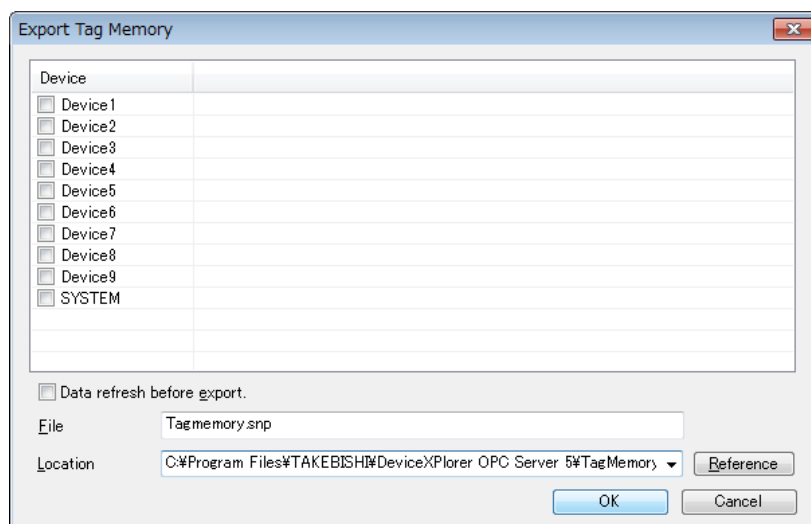
When importing, confirmation will be displayed.



4.17.4 Export[TagMemory]

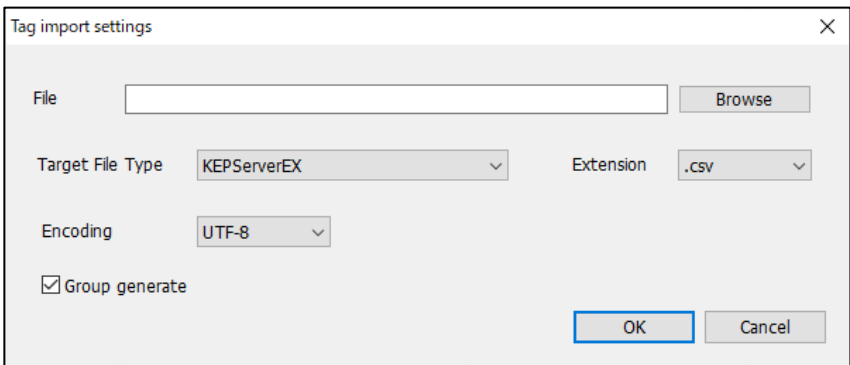
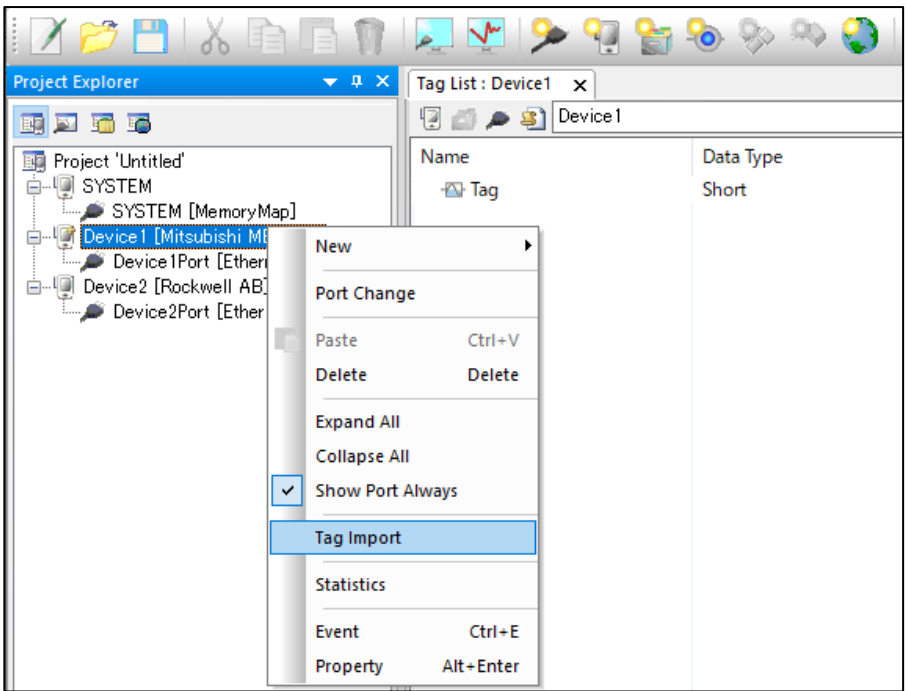
Export TagMemory(cache, timestamp, quality) that DeviceXPlorer has currently.
Select target device.

If [Data refresh before export.] is checked, DeviceXPlorer updates all data to the latest value before exporting TagMemory.



4.17.5 Tag Import

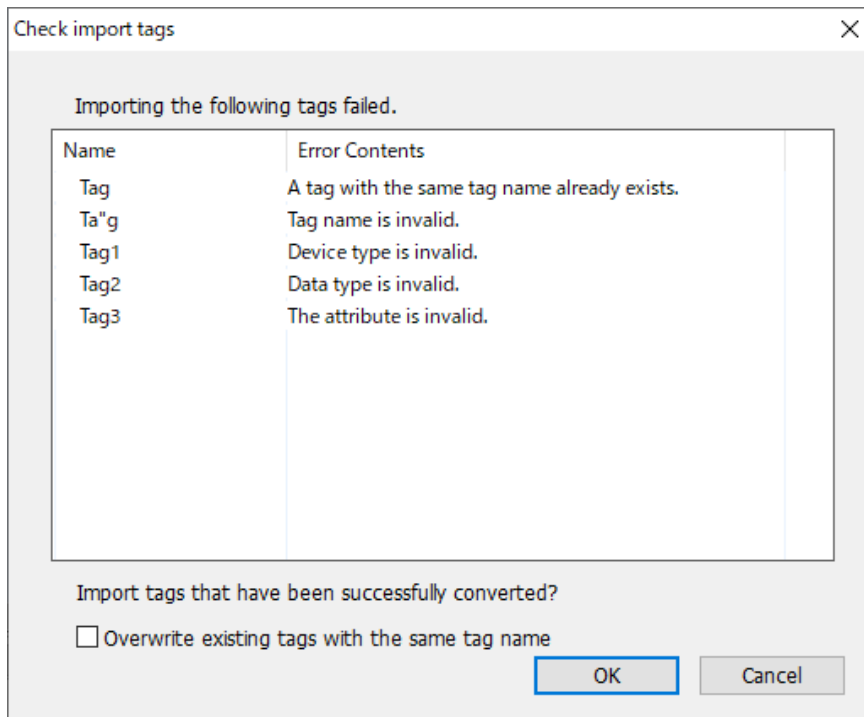
It is possible to create tags by importing the tag definition file.
Right-click a device or group node in Project Explorer, and click "Tag Import".



Items	Description
File	Select the tag definition file.
Target file type	Select the type of the target file.
Extension	Select the Extension of the target file.
Encoding	Select the Encoding of the target file.
Group generate	Create a group if it exists in the imported settings. If disabled, all tags will be added directly under the device or group without creating the group.

4.17.5.1 Import the tag definition file of KEPServerEX

Import the tag definition file exported by KEPServerEX from Kepware.



This dialog shows the name and error contents of the tag that failed conversion.

Click "OK" to import the tags that succeeded conversion.

Check "Overwrite existing tags with the same tag name" and click "OK", it is possible to import tags with the error "A tag with the same tag name already exists."

Refer to 4.17.5.2 for the detail of Error.

[Supported suite list]

Suite Name	
KEPServerEX	DeviceXPlorer OPC Server
Mitsubishi Ethernet	MELSEC Ethernet
Mitsubishi Serial	MELSEC Serial
Omron Fins Ethernet	SYSMAC Ethernet
Toyopuc Ethernet	TOYOPUC Ethernet
Yaskawa Ethernet	MP Ethernet
Yaskawa Serial	MP Serial
Keyence Ethernet	KV Ethernet
Allen-Bradley ControlLogix Ethernet	AB Ethernet
Siemens TCP/IP Ethernet	SIMATIC Ethernet
GE SNP	GE Serial
Fanuc Focas Ethernet	FANUC FOCAS
Modbus Serial	Modbus Serial
Modbus ASCII	

[Supported Data Type list]

Data Type		Description
KEPSeverEX	DeviceXPlorer	
Boolean	BOOL	1 bit ^{*1}
Char	BYTE	8 bit signed integers data
Byte	UBYTE	8 bit unsigned integers data
Short	SHORT	16 bit signed integers data
Word	USHORT	16 bit unsigned integers data
Long	LONG	32 bit signed integers data
DWord	ULONG	32 bit unsigned integers data
LLong	LONGLONG	64 bit signed integers data
QWord	ULONGLONG	64 bit signed integers data
Float	FLOAT	32 bit real value IEEE754
Double	DOUBLE	64 bit real value IEEE754
String	STRING	Null-terminated string(Unicode) ^{*2}
BCD	SHORT (Calculation: BCD Format)	2 BytePackBCD Range 0 - 9999
LBCD	LONG (Calculation: BCD Format)	4 BytePackBCD Range 0 - 99999999
Date	N/A	32 / 64bit Datetime

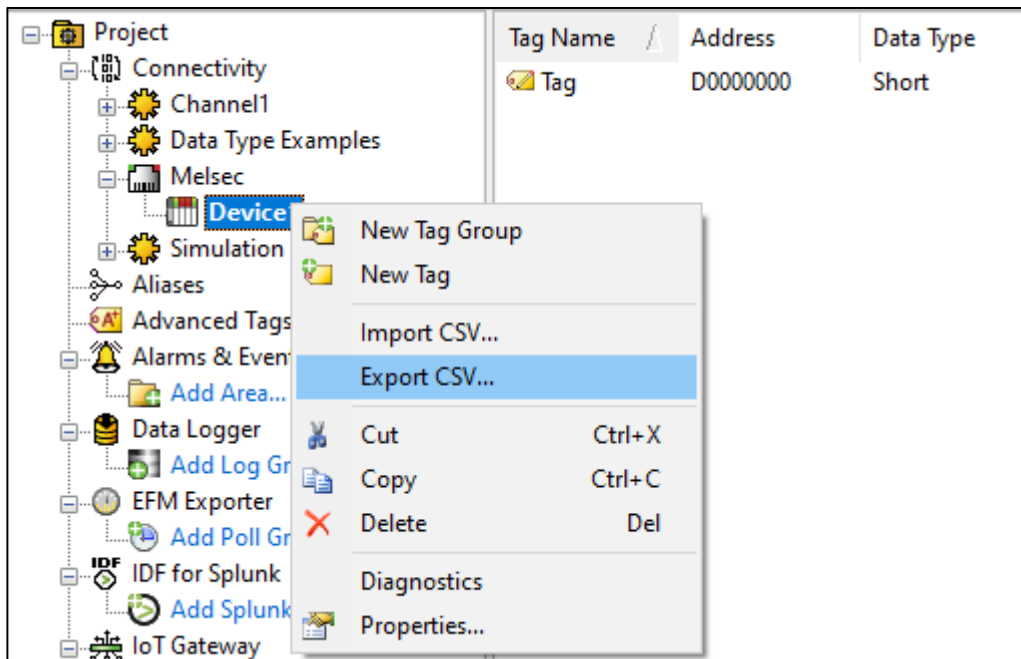
*1 In case of KEPSeverEX, units per size of logical type array tags is bit.

Whereas, it at DeviceXPlorer is byte or word (depends on the basic data type for each suite).

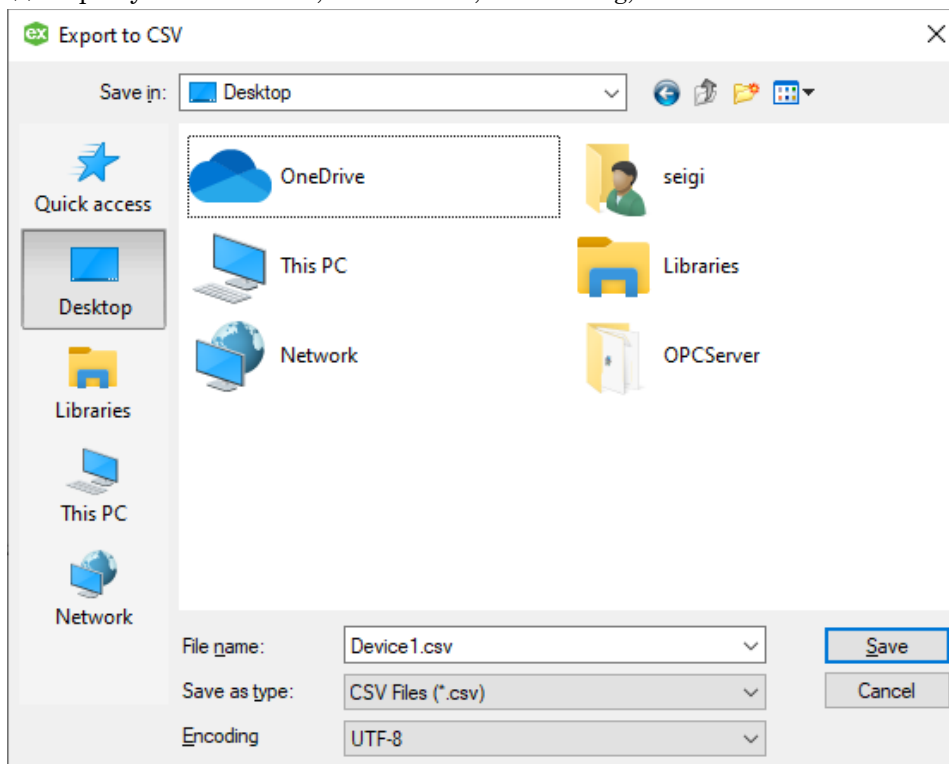
*2 Array tag is not supported.

[Tutorial exporting the tag definition file from KEPServerEX]

(1) Right-click the device (or Tag Group) and click “Export CSV...” to export the tag definition file.



(2) Specify the destination, the file name, the encoding, and click "Save".



4.17.5.2 Error List

Show the list of error output when tag definition information conversion fails.

Error	Description
Tag name is invalid.	Tag name is longer than 64 characters or might contain inhibited characters.
Device type is invalid.	Unsupported device type might be specified.
Device Number is invalid.	Device number out of range might be specified or the format might be incorrect.
Data type is invalid.	Unsupported data type might be specified.
The attribute is invalid.	Unsupported attribute might be specified
Comment is invalid.	Comments might contain inhibited characters.
Scaling is invalid.	Unsupported scaling type might be specified.
Raw Min is invalid.	Raw Low value might be incorrect.
Raw Max is invalid.	Raw High value might be incorrect.
Scaled Min is invalid.	Scaled Low value might be incorrect.
Scaled Max is invalid.	Scaled High value might be incorrect.
The unit is invalid.	Units might contain inhibited characters.
The format of Address is invalid.	The format of Address in the tag definition file of KEPServerEX might be incorrect.
Tag address is invalid.	The tag address might be incorrect. This error can occur only for AB Ethernet.
A tag with the same tag name already exists.	The same tag name of tag already exists in the tag list. Check "Overwrite existing tags with the same tag name" on the check import tag dialog, and click "OK" to overwrite existing tags and import.

Important

The name of tag definition at KEPServerEX may have characters treated as delimiter by DeviceXPlorer.
It is not possible to import the tag definition has delimiter character in tag name.

4.18 Hot Configuration

The DeviceXPlorer can change a Device settings and Tag settings also in the connection from a client. Notes are indicated to be an item reflected in below by Hot Configuration.

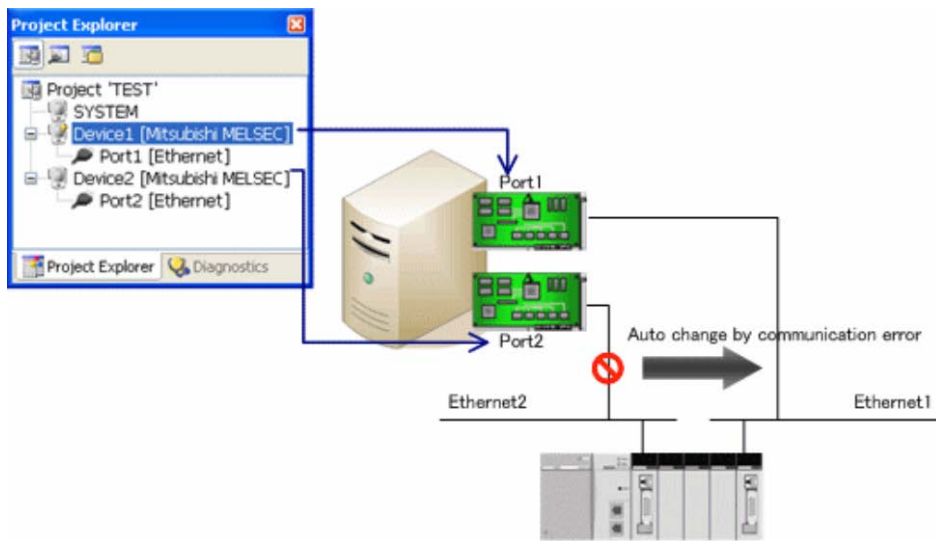
Object	Hot Configuration	Note
Common Properties	Enable	There are some properties to update after PC re-starts.
Project Properties	Enable	There are some properties to update after PC re-starts.
User Information	Enable	
Port	Enable:Add/Edit Disable:Delete	Disable to change Port name. When change properties, Port is re-connected.
Device	Enable:Add/Edit Disable:Delete	Disable to change Device name. When change properties, Port is re-connected in some models.
Device Option	Enable	Disable to change follows properties, -Topic name -System Tag uses \$ char
Group	Enable:Add/Edit Disable:Delete	Disable to change Group name.
Tag	Enable:Add/Edit Disable:Delete	
Method Tag	Enable:Add/Edit Disable:Delete	
Structure Tag	Enable:Only Add	Disable to change Structure type.
Structure Template	Enable:Only Add	Disable to edit or delete member.
Script	Enable:Add/Edit/Del	
Operation of Project	Disable	Disable to new project, to open project file, to close under communication.
Import / Export	Enable	When you execute import under communication, Configuration is skipped, not overwrite.

Important

Even if an Object cannot delete, it which is not used for communication can be deleted.

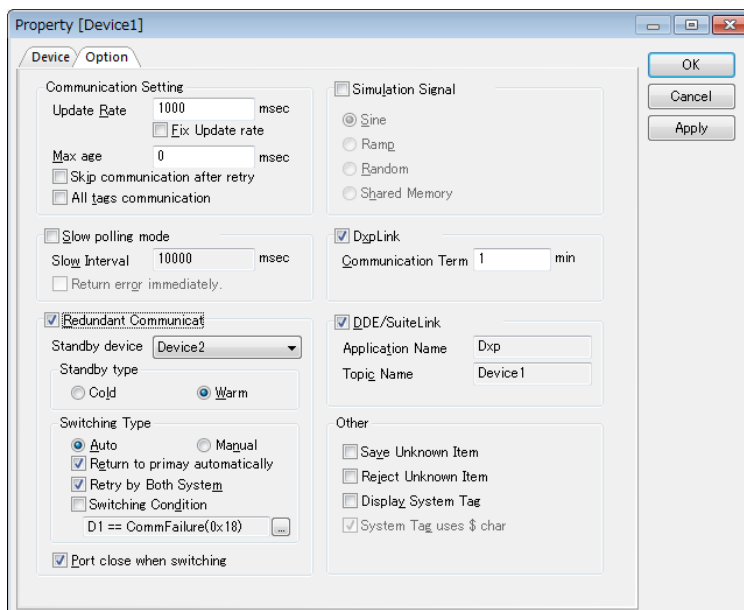
4.19 Redundant Communication

The redundant communication function continues communication by automatically switching lines when a communication failure occurs. Using this function, you can easily configure a redundant system without worrying about switching communication lines.



[Settings]

- (1) In the Device Options screen, turn on the redundant communication function.
- (2) Select “Auto switching”.
- (3) Select a registered device for the standby device.



In the figure above, communication switches to Device2 when a communication failure occurs at Device1, and returns to Device1 when a communication failure occurs at Device2.

For configuration about Redundant, please refer to 4.4.5.

4.19.1 Warm Standby

Also during normalcy system communication, it communicates by a standby system and performs a communication check.

In the case of Warm Standby, while disconnecting Primary system, when it can communicate by Standby system, it changes to a Standby system.

When Standby system is also under disconnection, it communicates with Primary system.

(It shifts to slow polling mode at the time of slow polling mode effective.) During disconnection, both systems always perform a return check by both systems, and it changes them to the system which returned.

4.19.2 Cold Standby

To the timing which was disconnected, Primary system changes communication pathway to Standby system.

When Standby system is also under disconnection, communication pathway is changed to Primary system.

Note

When automatic switching occurs, communication is not automatically restored to the normal system even when it recovers.

If you want to manually return communication to the normal system, write FALSE to the “\$Standby” system tag.

4.20 Communication Period

The communication period of DeviceXPlorer differs for every interface and configurations.

Interface	Communication Period
OPC DA	Specified by OPC client's update rate.
OPC UA	But if you check “Fix Update Rate” of Device Option, DeviceXPlorer obeys “Update Rate” of Device Option.
OPC A&E	Specified by “Update Rate” of Device Option.
SuiteLink	
DxpLink	
Script	Specified by “Update Rate” of Script.
Device Monitor	Specified by “Monitor Rate” of Common Properties.

These interfaces perform communications processing a best communication cycle. But to maintain past compatibility, only the OPC DA interface is preparing the mode which does not perform the optimal communications processing. (Client unit communication)

You can configure Client unit communication in “Communication per Client” of Project Properties.

The difference in operation by setup of the client unit communication in an OPC DA interface is as follows.

Communication per Client	Communication Period
Disable(Default)	DeviceXPlorer selects best communication cycle
Enable	Communications processing is performed the renewal cycle of OPC for every OPC group registered from an OPC client. Compared with the time of “Disable”, Communication Rate becomes high.

4.21 Slow Polling Mode

When communication error such as timeout error and open error occurs, DeviceXPlorer reduce communication rate. You can configure it on Device Option. At the time communication status is restored, DeviceXPlorer leaves Slow Polling mode.

For configuration about Slow Polling Mode, please refer to 4.4.5.

If \$SlowpollingMode is supervised, you can understands whether the device is in Slow Polling Mode or not. If this tag is TRUE, under Slow Polling Mode. If this tag is FALSE, under normal communication.

Shows message about Slow Polling.

Message	Contents
Entering Slowpolling Mode.	DeviceXPlorer enters under Slow Polling Mode.
Leaving Slowpolling Mode.	DeviceXPlorer leaves from Slow Polling Mode.

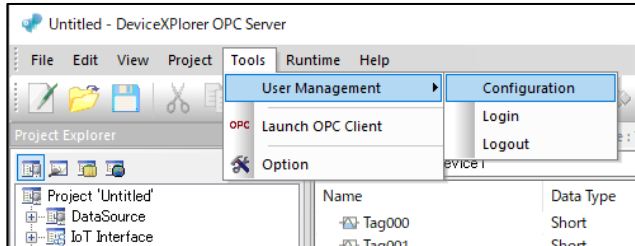
Note
If you'd like to reduce communication time under Slow Polling Mode, please check "Return error immediately".

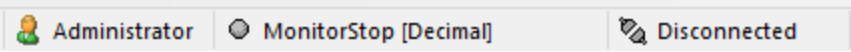
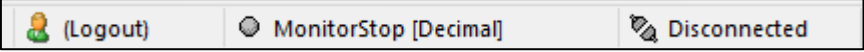
4.22 User Management

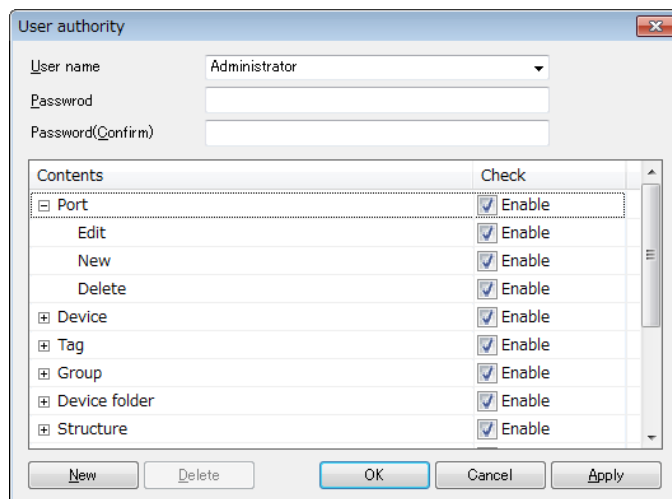
4.22.1 Configuration

You can configure of operators.

Usually, it is set up by Administrator authority (with no password), and restriction of operation is absolutely none.



Item	Description
Configuration	Configure User.
Login	Login user. The user name is displayed in status bar. 
Logout	Logout user. Displayed (Logout) in status bar. 



Item	Description
User name	Specify the User name
Password	Specify the password
Password(Confirm)	Specify the password for a check
Contents	The operation authority of the user who specified is set up. Please refer to 4.22.2 for explanation of each contents. You cannot modify “Administrator” authority.
New	Create a new user.
Delete	Delete a user. You cannot delete “Administrator”.

4.22.2 Contents List

Item		Description
Port	Edit	Enable to configure Port
	New	Enable to create Port
	Delete	Enable to delete Port
Device	Edit	Enable to configure Device
	New	Enable to create Device
	Delete	Enable to delete Device
Tag	Edit	Enable to configure Tag
	New	Enable to create Tag
	Delete	Enable to delete Tag
Group	Edit	Enable to configure Group
	New	Enable to create Group
	Delete	Enable to delete Group
Device Folder	Edit	Enable to configure Device Folder
	New	Enable to create Device Folder
	Delete	Enable to delete Device Folder
Structure	Edit	Enable to configure Structure Tag and Structure Template
	New	Enable to create Structure Tag and Structure Template
	Delete	Enable to delete Structure Tag and Structure Template
Script	Edit	Enable to configure Script
	New	Enable to create Script
	Delete	Enable to delete Script
User	Edit	Enable to configure User authority
	New	Enable to create User
	Delete	Enable to delete User
Project File	New	Enable to create new Project file
	Load	Enable to load project file
	Save	Enable to save project file
Setting of Project	Edit	Enable to configure Common Properties and Project Properties
Monitor	Read	Enable to monitor of Device Monitor, Watch Monitor, and Register Monitor.
	Write	Enable to write from GUI
Application	Exit	Enable to exit DeviceXPlorer
OPC UA	UserName/Password	Enable to connect with user name / password authentication via OPC UA.
	X509 Certificate User	Enable to connect with user X509 user authentication via OPC UA.
	SecurityAdmin	Become a user with security administrator authority in OPC UA communication.

Important

- To configure license, You need to login as “Administrator”
- User authority is invalid to the client at the time of carrying out COM starting. For example, the application which carried out COM starting also in the state of login by the user without the end authority of application carries out the end of application by the connection release from a client.

4.22.3 Auto Login

You can specify Login User when DeviceXPlorer starts in Common Properties.
If you failed to specify Automatic Login User, DeviceXPlorer start as logoff status.

The screenshot shows the 'Property of Untitled' dialog box with the 'Common Properties' tab selected. The 'Setting Tool' section is expanded, showing the 'Automatic Login User' field set to 'Administrator'. The 'General' section includes fields for 'Start as Service Program' (Service Program), 'Running Processor Setting' (None), 'Language' (English), 'SuiteLink Application Name' (DXPV7), 'Connect with the runtime on startup' (Enable), and 'Maximum number of log messages to manage' (100000). The 'Automatic Login User' field is highlighted in blue. The 'Project file save folder' and 'Startup File' fields are empty. The 'OK', 'Cancel', and 'Apply' buttons are visible on the right side.

Common Properties	
General	
Start as Service Program	Service Program
Running Processor Setting	None
Language	English
SuiteLink Application Name	DXPV7
Connect with the runtime on startup	Enable
Maximum number of log messages to manage	100000
Setting Tool	
Monitor Rate (msec)	600
Automatic Login User	Administrator
Project file save folder	
Startup File	

Automatic Login User
Select the login user. This setting becomes effective after DeviceXPlorer is restarted.

Specify User name and password.

The screenshot shows the 'Login' dialog box with a close button (X) in the top right corner. It contains two input fields: 'User name' with a dropdown menu showing 'Administrator' and 'Password' with a text box. The 'OK' and 'Cancel' buttons are at the bottom right.

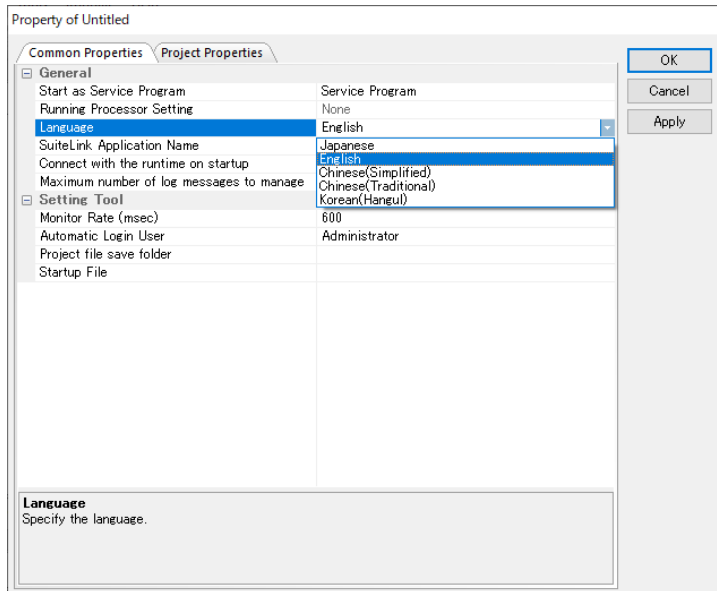
Login	
User name	Administrator
Password	
OK Cancel	

4.23 Change display language dynamically

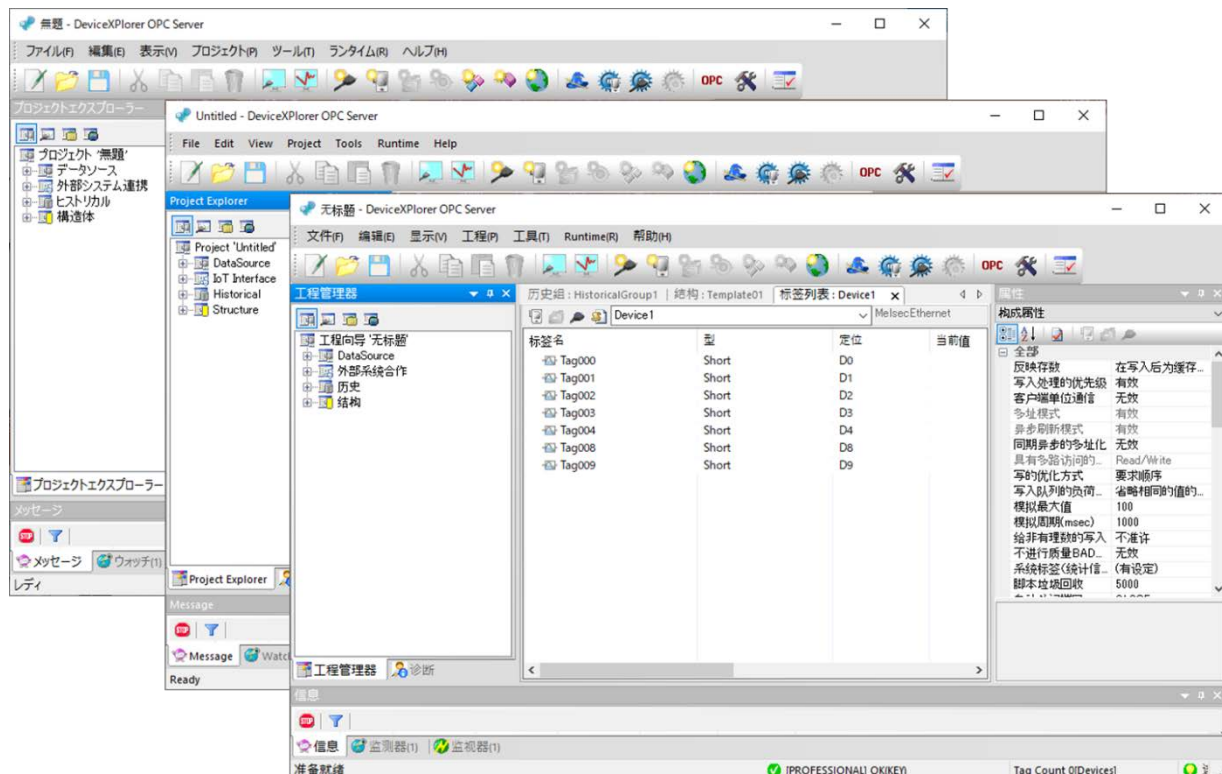
You can change the display language of DeviceXPlorer without restart application.

[Settings]

- (1) Go to Tools -> Options, specify "Language", and select other language.
- (2) Click "OK".



- (3) DeviceXPlorer changes the language.



4.24 DCOM Configuration

This section describes how to configure DCOM to connect from a remote computer using OPC. The examples provided in this section are typical configuration examples. Please modify each setting based on your system security needs.

Note

Please refer to related documents on OS for more details on DCOM.

Important

When DeviceXPlorer runs as windows service, the DCOM configuration is required.

When OPC Client application runs as windows service and connects with DeviceXPlorer, the DCOM configuration is required.

4.24.1 Overview

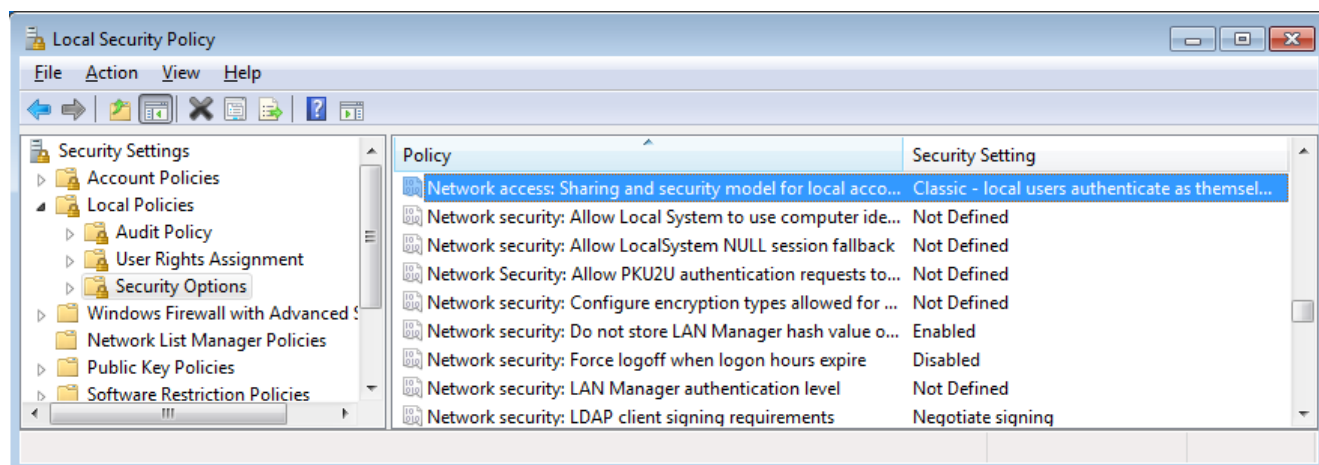
Client user must be authenticated on the machine on which the OPC server is running in order to enable DCOM connection. Authentication of the client machine user is normally done by configuring one user account and one password on both machines running in a work group. From the user account on the client machine, you log on to the local PC, remote log on to the server machine, and run the OPC server. The following description provides an example of configuration on the OPC server machine, unless otherwise indicated.

See below for a configuration example of using DCOM:

Machine	Field	Simple	Secure
Server	Authentication Level	None	Connect
	Start Access Right	Everyone	Arbitrary user name
	Access Right	Everyone	Arbitrary user name
	Identity	Interactive User	Interactive user/Launching user
Client	Pre-defined Authentication Level	None	None

* “Simple” configuration is for when security is not emphasized.

To use DCOM in a Windows workgroup environment, start “Local Security Policy” from Administrative Tools and set “Network Access: Sharing and security model for local accounts” to “Classic.” This step is not required when you wish to use DCOM in the domain environment.



Important

You cannot connect via DCOM if checking the “Windows firewall” in property window of “Local Area Connection” So you should not check this option button.

DCOM can be connected by making the firewall effective by setting security.

4.24.2 Procedure

[Start DCOM Configuration Utility]

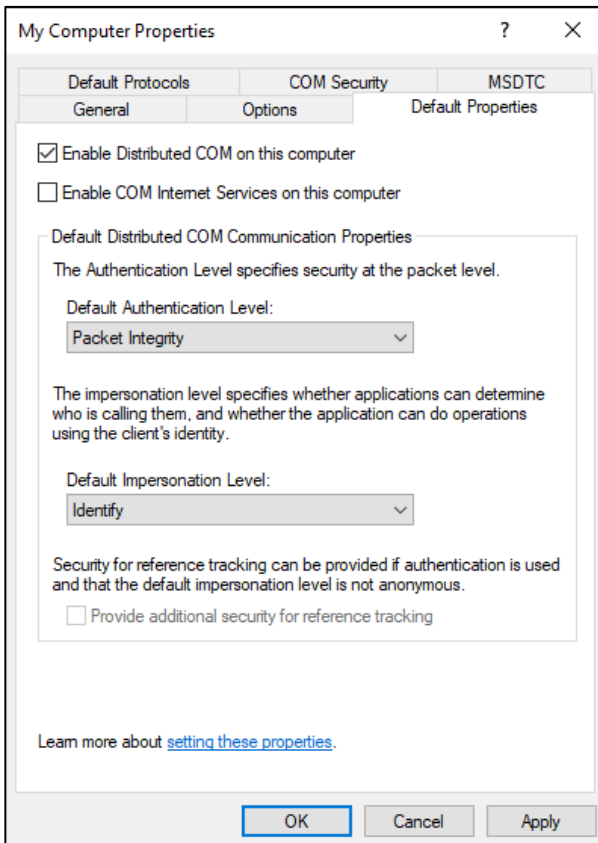
From the Start menu, select “Run” and enter “DCOMCNFG.EXE” to launch the DCOM configuration utility.

Note

You must log on with Administrator rights to configure DCOM.

[Default Properties]

Combined utilities start Navigate through “Component Services” -> “Computer” -> “My Computer.” Right mouse click on “My Computer,” open the Properties page from the pop-up menu, and select the “Default Properties” tab.



* Check the “Enable Distributed COM on this Computer” check box.

* Set “Default Authentication Level” to “Packet Integrity.”

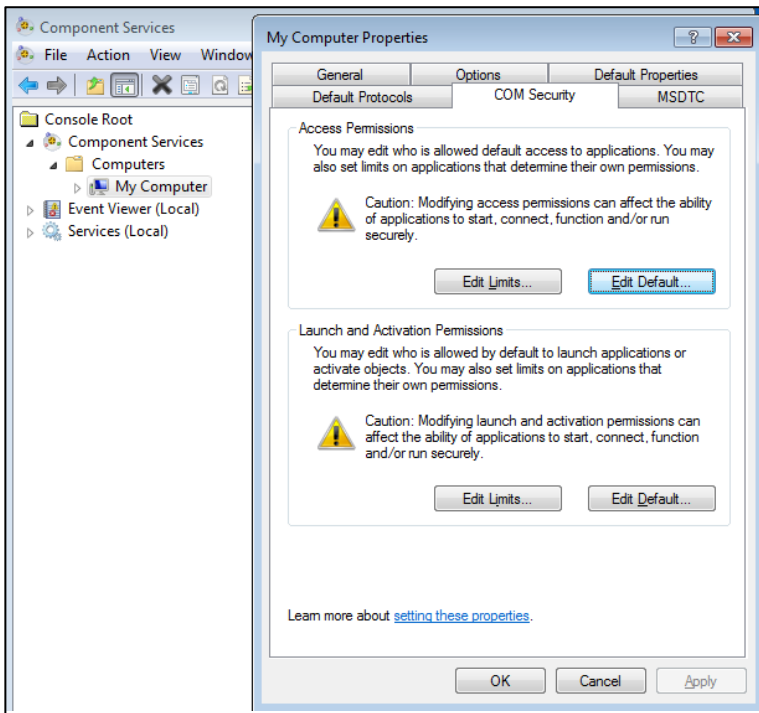
* Set “Default Impersonation Level” to “Identify.”

Important

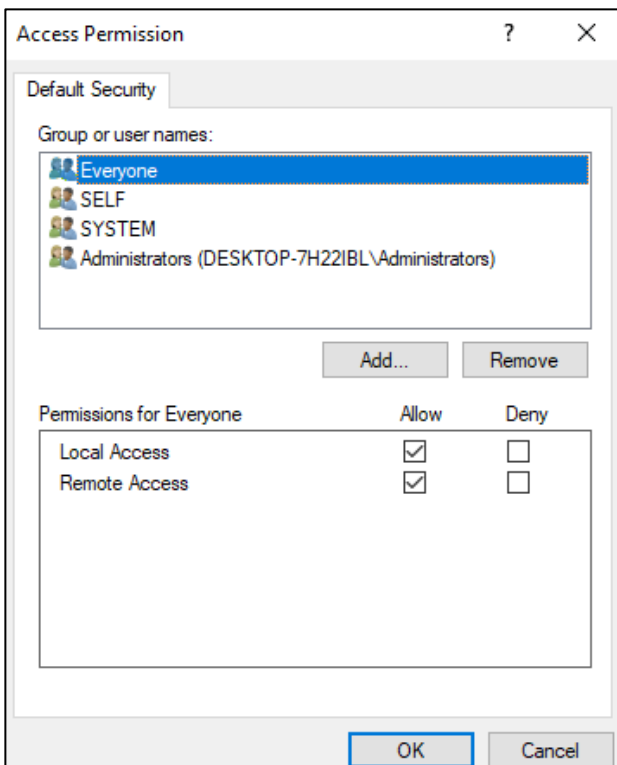
On the client machine, set “Default Authentication Level” to “Packet Integrity.”

[COM Security]

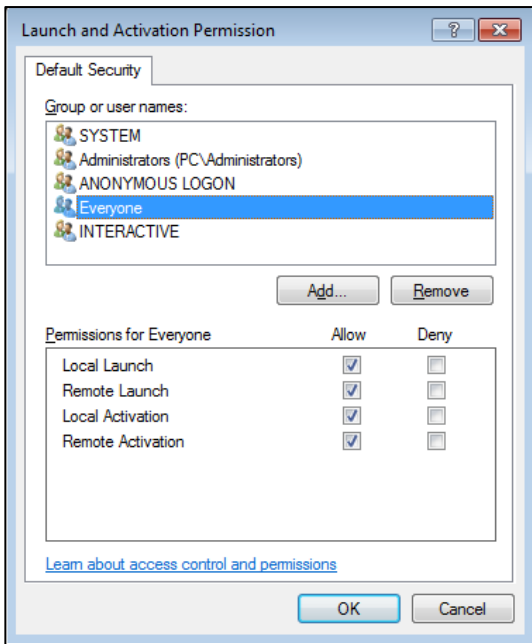
The change in the DCOM setting is needed.



"DCOMCNFG.EXE" is started from the command line or "Component Service" is opened from the management tool of control panel, and "COM security" tab is selected.



The dialog opens when "Edit limits" button of "Access permit" is pushed. "Everyone" and "SYSTEM" is selected to access OPCenum.exe remotely, and "Remote Access" is set to "Allow".



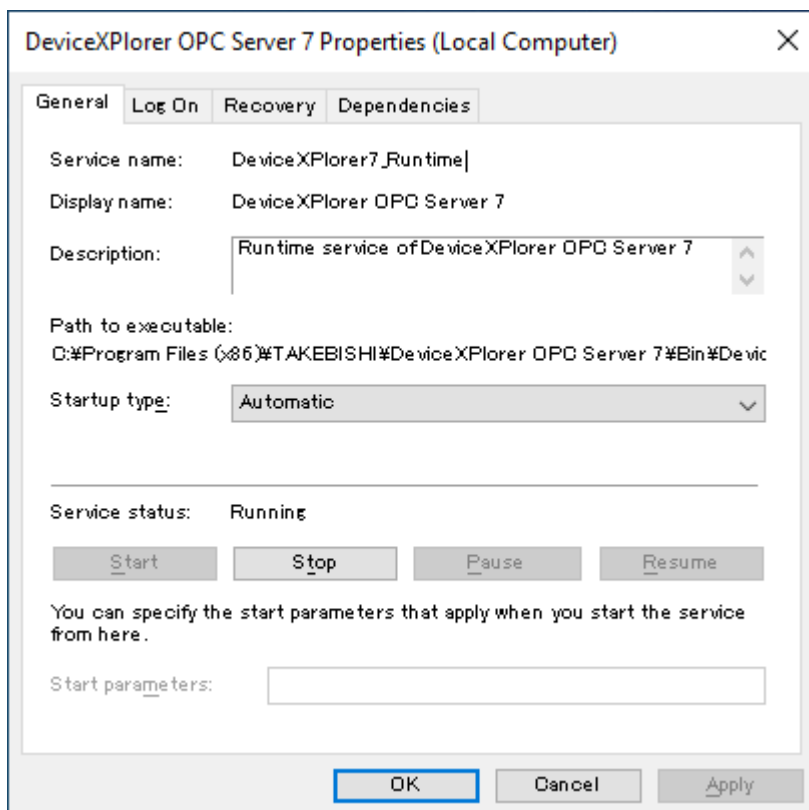
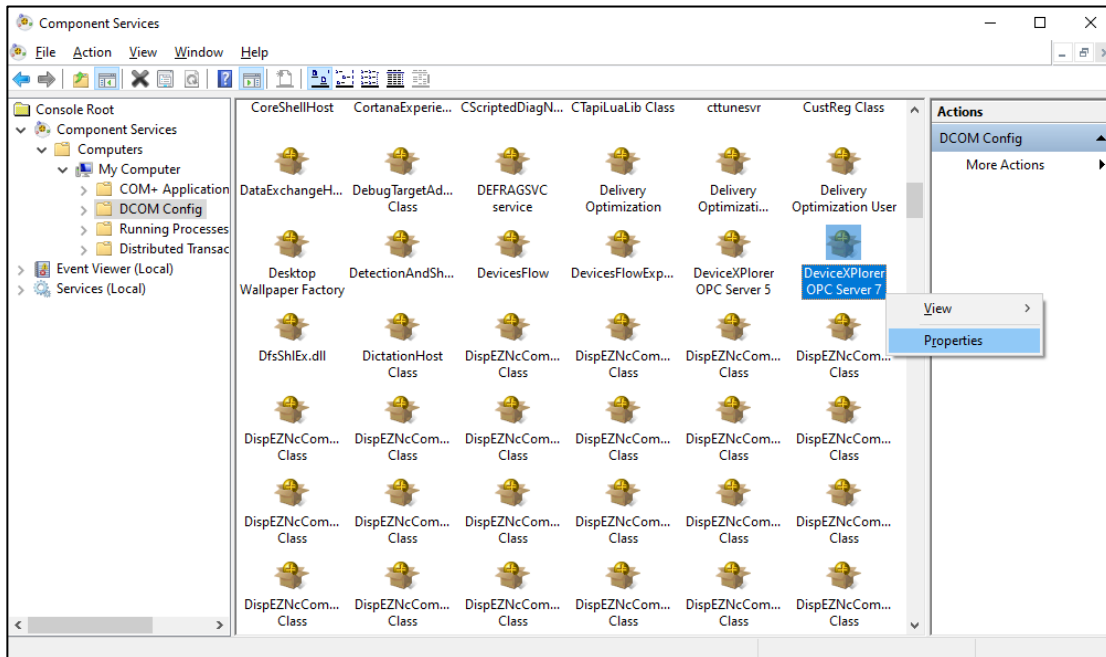
The dialog opens when "Edit limits" button of "Access permit of starting and making actively" is pushed. "Everyone" and "SYSTEM" is selected, and "Remote Launch" and "Remote Activation" are set to "Allow".

Important

If client application runs as window service, please add permission to INTERACTIVE and SYSTEM.

[Specific Properties (General)]

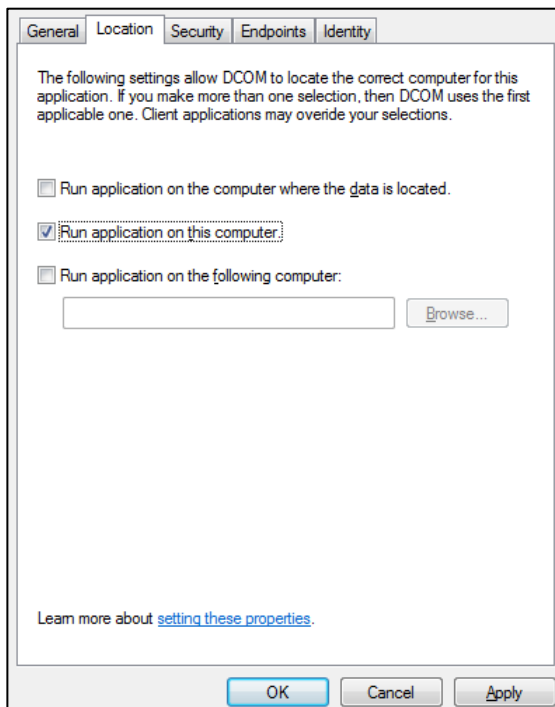
Right mouse click on "DeviceXPlorer OPC Server 7" from "DCOM Config" under "My Computer", and select Properties from the pop-up menu.



* Set the "Authentication Level" to "Packet Integrity."

[Specific Properties (Location)]

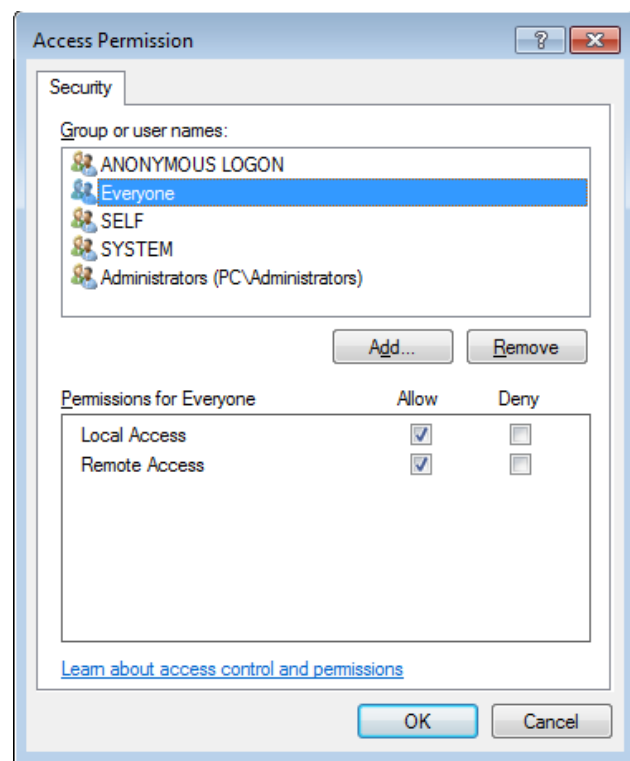
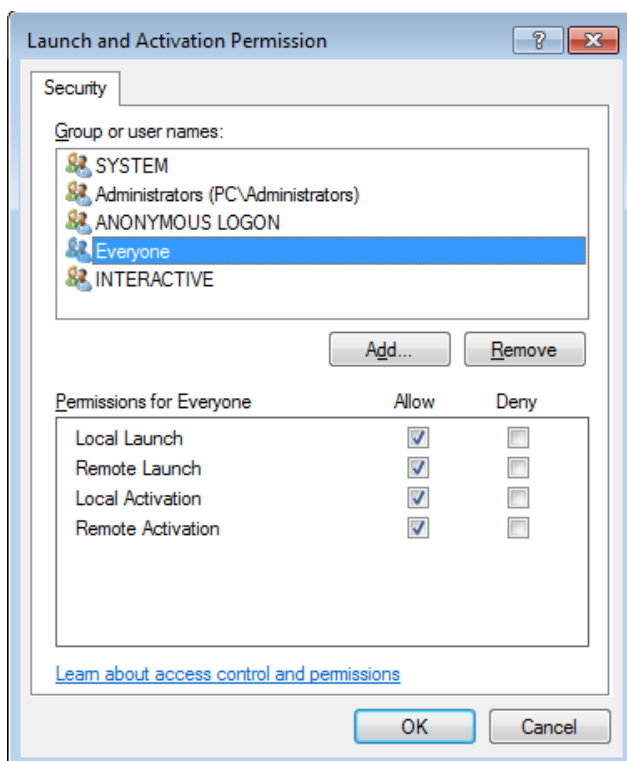
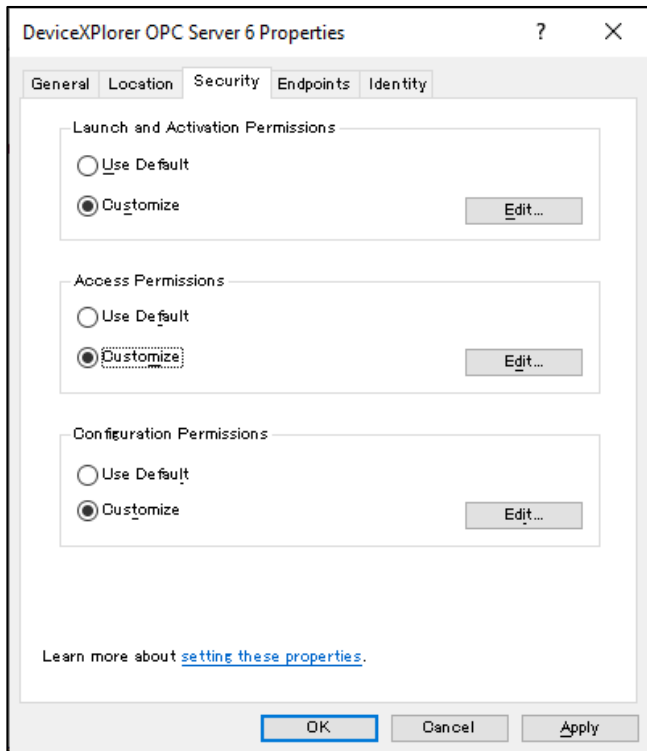
Select the “Location” tab.



*Check “Run application on this computer.”

[Specific Properties (Security)]

Select “Customize” in each field and press “Edit.”



*Add “Everyone” and “SYSTEM” to “launch permissions,” “access permissions,” and “configuration permissions.”

[Specific Properties (ID/Identity)]

Select the “ID” tab, and specify the user account that runs the application.

The screenshot shows the 'DeviceXPlorer OPC Server 6 Properties' dialog box with the 'Identity' tab selected. The dialog has a title bar with a question mark and a close button. Below the title bar are tabs for 'General', 'Location', 'Security', 'Endpoints', and 'Identity'. The 'Identity' tab contains the text 'Which user account do you want to use to run this application?'. There are four radio button options: 'The interactive user.' (selected), 'The launching user.', 'This user.', and 'The system account (services only)'. Below 'This user.' are three text input fields labeled 'User:', 'Password:', and 'Confirm password:', followed by a 'Browse...' button. At the bottom of the dialog are 'OK', 'Cancel', and 'Apply' buttons. A link 'Learn more about [setting these properties.](#)' is located at the bottom left of the main content area.

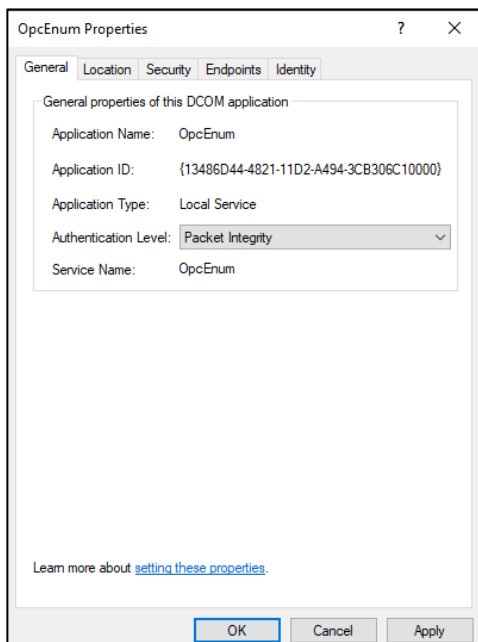
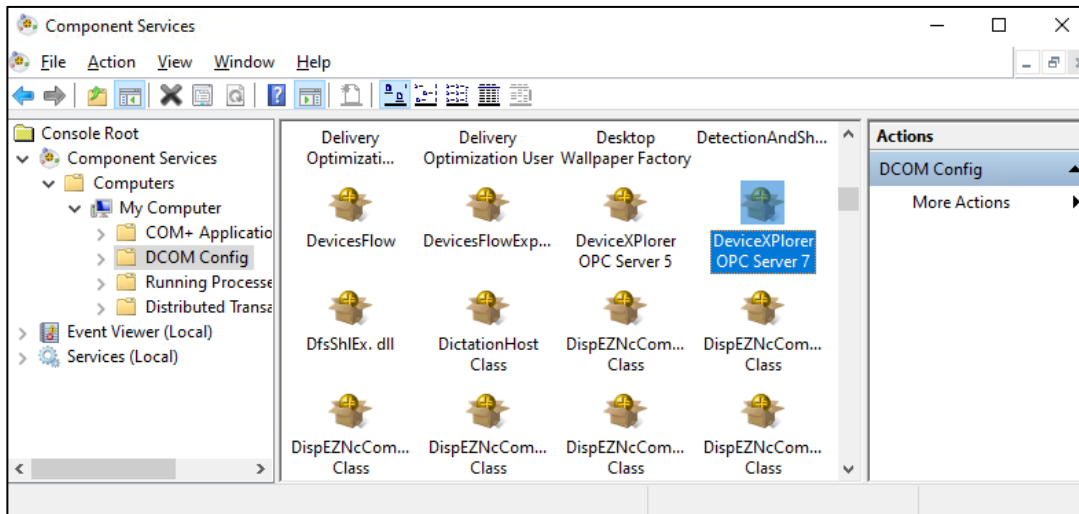
Select “This User” and specify a user that has Administrator rights. Or select “System Account” to run it as a service program. When launching DeviceXPlorer as a service, it becomes a “System Account,” and you cannot specify it as an “Interactive User” or “Launching User.”

Important

Restart computer with changing DCOM Configuration.

[OPCEnum Settings]

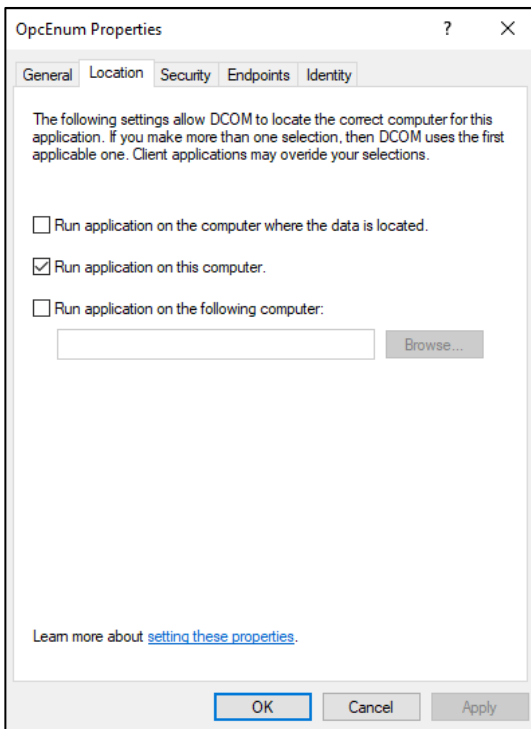
Right mouse click on "OpcEnum" from "DCOM Config" under "My Computer", and select Properties from the pop-up menu.



* Set the "Authentication Level" to "Packet Integrity".

[Specific Properties (Location)]

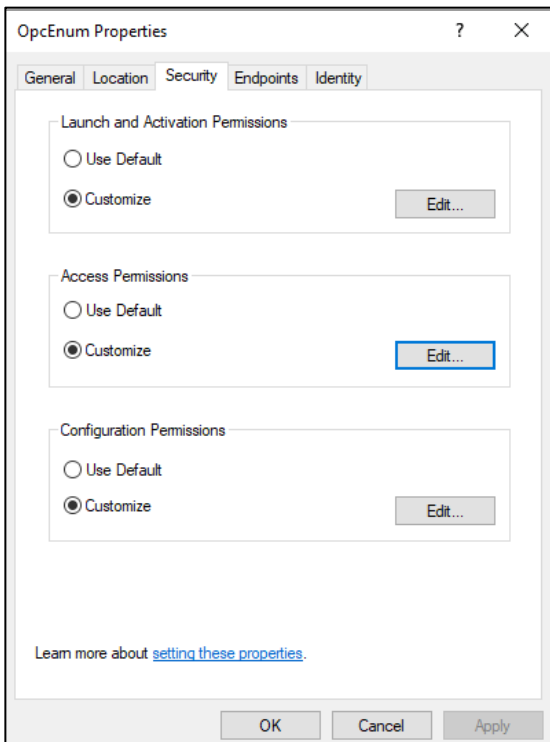
Select the “Location” tab.



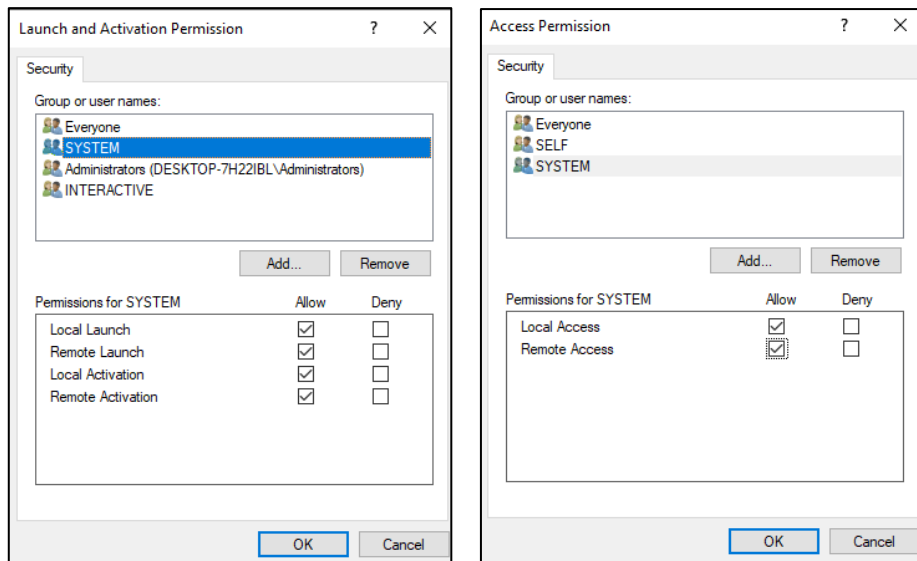
*Check “Run application on this computer.”

[Specific Properties (Security)]

Select “Customize” in each field and press “Edit.”

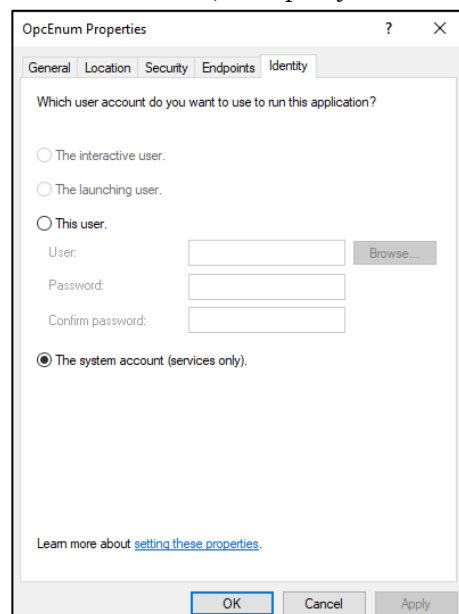


Under "Security," add "SYSTEM" and grant access. (If you cannot connect, add "Everyone" to allow access.)



[Specific Properties (ID/Identity)]

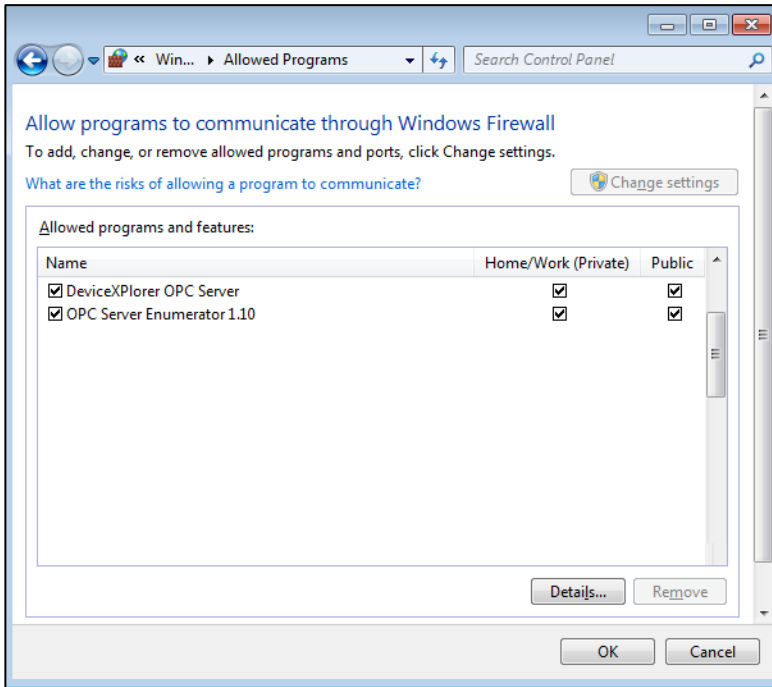
Select the "ID" tab, and specify the user account that runs the application.



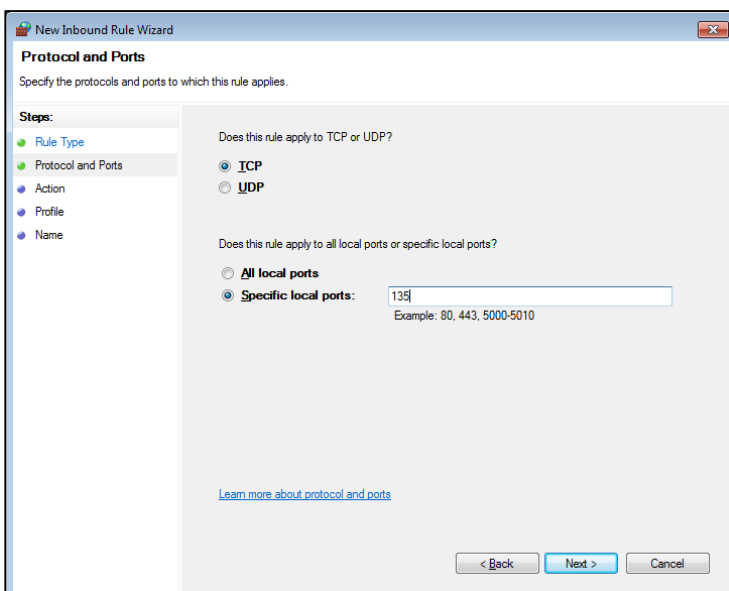
[Firewall(Exceptions)]

The Windows firewall is effectively set by default. It explains the method of connecting DCOM with DeviceXPlorer with the firewall made effective since it the next pages.

The application to pass the firewall can be specified in "Exceptions" tab. And push "Add Programs" button and add "DeviceXPlorer OPC Server 7". Moreover, to browse the installed server list from another node with the OPCENUM function, add "OpcEnum.exe".



In addition, port "135" of "TCP" used with DCOM pushing "Add Port" button is added.



5 Script

User can input original logic in DeviceXPlorer by using script. Device, Group and Struct Template can have event and methodtag can have event. About methodtag, please check 4.8.

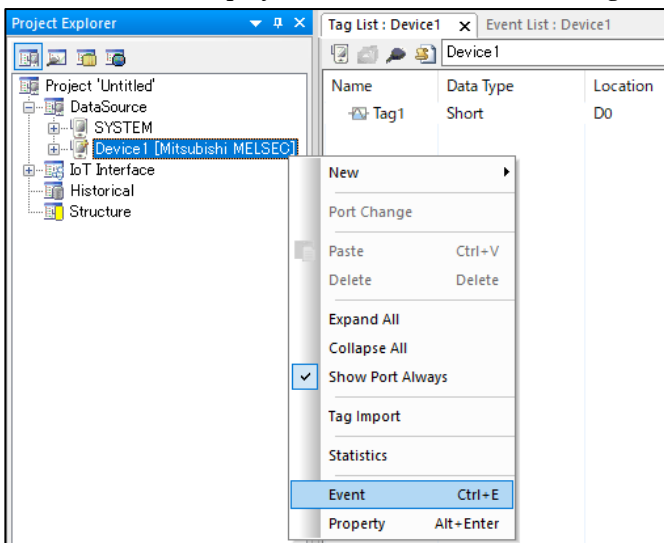
Important

- There is no upper limit to the number of script settings, but please note that the communication load will increase.
- Please note that executable number of scripts (including bridges) is limited depending on the edition of DeviceXPlorer.

5.1 Event Script Settings

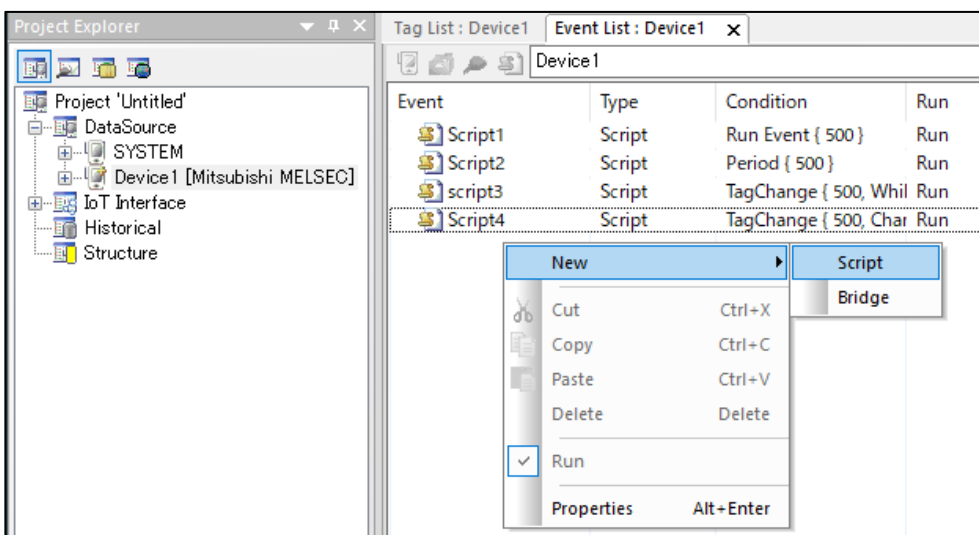
Device, Group and Struct Template have event list.

Event list will be displayed when select [Event] from right-click menu for Device and Group.

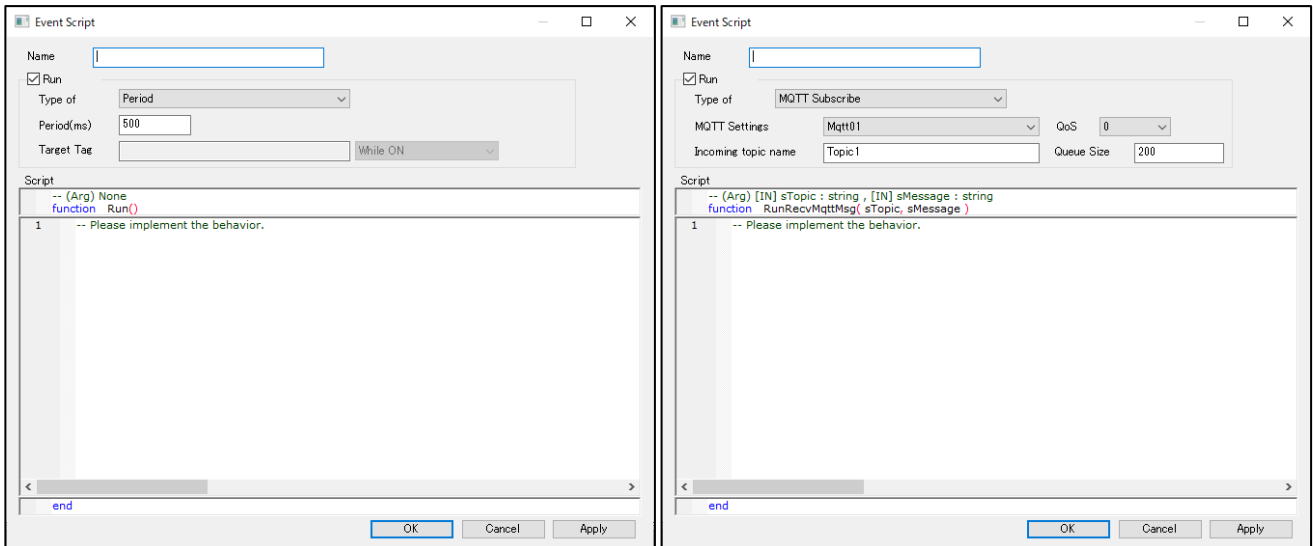


Event will be displayed in event list.

You can create, modify, delete and run/stop event from right-click menu.



Input script setting.



Item	Information
Name	Set script name.
Run	Specify run state of script. If no check, script is not performed.
Event	Specify event condition. Period - - Script is performed the specified cycle (fixed cycle). Tag Change- - Script is performed when the value of the tag changed. Run Event- - Script is performed at the time of event engine starting of the parent Device. (*) Stop Event- - Script is performed at the time of event engine stopping of the parent Device. (*) MQTT Subscribe- - Script is performed when MQTT message is received. Received topics and messages can be used in scripts as sTopic and sMessage variables.
Tag	When "Tag Change" is chosen in Event, a specification tag and conditions can be specified. Static tag and dynamic tag can be specified. You cannot use array tag and string tag. [While ON] - - While a specification tag turns on, it performs at a case (when it is except zero). [Change] - - Script is performed at the time of specified tag's Value or Quality or Timestamp changing. [Change(Value)] - - Script is performed at the time of specified tag's Value changing. [Change (Quality)] - - Script is performed at the time of specified tag's Quality changing. [Change (Time stamp)] - - Script is performed at the time of specified tag's Timestamp changing. [Rise] - - Script is performed at the time of specified tag changes to ON from OFF. [Fall] - - Script is performed at the time of specified tag changes to OFF from ON.
Period	Specify running interval and communication interval of script.
MQTT Settings	Select the connection destination to subscribe to MQTT from the MQTT base settings. Can be set only if Event is MQTT Subscribe.
Incoming topic name	Specify the topic name to subscribe to MQTT. Can be set only if Event is MQTT Subscribe.
QoS	Specify QoS from Level 0 to 2 when subscribing to MQTT. Can be set only if Event is MQTT Subscribe.
Queue Size	Specify the maximum number of MQTT receive messages waiting to be processed. Can be set only if Event is MQTT Subscribe.
Script	Describe User logic (script) . There are size restrictions of 32767 characters per one script. The lists of properties, functions and code-snippet are shown by pushing "Ctrl + Space key". Tag browse dialog is shown by pushing "Ctrl + B" or "Ctrl + Insert key".

	The function name listed at the top of the script will be the function name when calling from another script. If Event is Period, Tag Change, Start Event, Stop Event, it can be called as Run function. If Event is MQTT Subscribe, it can be called as RunMqttRecvMsg function.
--	---

* Event engine starts at the time of starting and stops at the time of application closing. And event engine will reboot to the timing when user creates, modifies and deletes script.

5.2 Script Specification

Script in DeviceXPlorer uses open script language Lua .

Please refer to the official reference of Lua for the basic specification of Lua.

Note
The Lua version used by DeviceXPlorer is 5.1.4.

5.3 Extend Specification

DeviceXPlorer has some extend specifications for Lua.

5.3.1 Reserved Word

The word which starts with Dxp cannot be used.

Object	Type	Information
Dxp	Table	The constant, a class, a function, etc. are defined as a member of this table.

5.3.2 Constant Value

Object Type Table (Dxp.ObjectType)

Constant Value	Type	Value	Information
Dxp.ObjectType			
Tag	Number	1	Tag
Device	Number	2	Device
Group	Number	3	Group
Struct	Number	4	Struct
Port	Number	5	Port
Method	Number	6	Method
EventScript	Number	7	EventScript
HttpClient	Number	8	HTTP Client
MqttClient	Number	9	MQTT Client

Tag Type (Dxp.TagType)

Constant Value	Type	Value	Information
Dxp.TagType			
Boolean	Number	1	Boolean Tag
Number	Number	2	Number Tag
String	Number	3	String Tag
Boolean Array	Number	4	Boolean Array Tag

	Number Array	Number	5	Number Array Tag
--	-----------------	--------	---	------------------

5.3.3 Classes

The following class (table) can be used.

The instance of the following classes is acquired by using the access macro provided originally by DeviceXPlorer.

Class	Information
BooleanTag NumberTag StringTag BooleanArrayTag NumberArrayTag	Tag Access Class This class offers the function accessed about a property common to some tags. There are five kinds, the object for Boolean Tag, t Number Tag, String Tag, Boolean Array Tag, and Number Array Tag, and it is necessary to use properly according to the data type of a tag.
Group	Group Access Class
Device	Device Access Class. This is derived from the Group Access Class.
Struct	Struct Intance Access Class. This is derived from the Group Access Class.
Port	Port Access Class
EventScript	Event Script Access Class
Method	Method Access Class
HttpClient	HTTP Client Access Class
HttpRequest	HTTP Request Class
HttpResponse	HTTP Response Class
MqttClient	MQTT Client Access Class
MqttPublishMessage	MQTT Publisher Class
Time	Time Class

5.3.3.1 Tag Access Class

Tag Access Class (BooleanTag / NumberTag / StringTag / BooleanArrayTag / NumberArrayTag)

Member	Type	Attribute	Information
ObjectType	Number	R	Object type (Dxp.ObjectType.**)
ObjectClassName	String	R	Object type name (Example : MelsecEthernetTag)
Path	String	R	Path
Name	String	R	Tag name
Config	String	R	Setting parameter (csv format)
Value (Different from class)	Boolean Number String	R/W	Boolean type in the case of Boolean Tag. Number type in the case of NumberTag. String type in the case of StringTag or BooleanArrayTag, and NumberArrayTag. Array tag's value becomes a comma separated value. If a new value is assigned, with a script execution thread, DeviceXPlorer will do asynchronous value writing to PLC in the end of script. Please refer to 5.3.8about writing value.
Quality	Number	R	Quality flag
Timestamp	Time	R	Timestamp
Location	String	R	Location String
TagType	Number	R	Tag's Type (Return Dxp.TagType.**)
VariantType	Number	R	Tag's data type (Return VARIANT : vt)
Comment	String	R	Comment
Simulate	Boolean	R/W	Simulation Status.

Function	Information
ToString()	[Return value] instance information (String) " <u>ObjectTypeName</u> { <u>ClassName</u> , <u>Name</u> , <u>Path</u> , <u>Location</u> , <u>Value</u> }"
ToStringValue()	[Return value] string converted from value of the tag (String)
HasValue()	[Return value] whether tag has value or not (Boolean)
GetValue(element) (Only Array tag)	[Argument value] element : element index of array (Number) [Return value] specified element value (Number) Boolean type in BooleanArrayTag Number type in NumberArrayTag
SetValue(element, value) (Only Array tag)	[Argument value] element : element index of array (Number) value : set value (Boolean or Number) [Return value] None
GetParent()	[Return value] Group (Group, Device, Struct)

5.3.3.2 Group Access Class

Group Access Class (Group)

Member	Type	Attribute	Information
ObjectType	Number	R	Object type (Dxp.ObjectType.**)
ObjectClassName	String	R	Object type name (Example : BaseGroup / MelecEthernetGroup)
Path	String	R	Path
Name	String	R	Name
Config	String	R	Setting parameter (csv format)

Function	Information
ToString()	[Return value] instance information (String) “ <i>ObjectTypeName</i> { <i>ClassName</i> , <i>Name</i> , <i>Path</i> }”
ResolvePath(<i>name</i>)	[Argument value] name : Subordinate's relative path (String) [Return value] Fullpath
FindChild(<i>name</i>)	[Argument value] name : Subordinate's relative path (String) [Return value] Object Access Class(BooleanTag , Device etc), Function <Example> <pre>local group = @@g("Group1"); local tag = group.FindChild("tag1");</pre>
GetParent()	[Return value] Group (Group, Device, Struct)
GetEvent(<i>name</i>)	[Argument value] name : event script name (String) [Return value] Event (EventScript)
GetMethod(<i>name</i>)	[Argument value] name : method tag name (String) [Return value] Method (Method)

5.3.3.3 Device Access Class

Device Access Class (Device)

Member	Type	Attribute	Information
Status	Boolean	R	Communication Status. Areas of \$Status (SystemTag)
Standby	Boolean	R/W	Redundancy Status. Areas of \$Standby (SystemTag)
Simulate	Boolean	R/W	Simulation Status. Areas of \$Simulate (SystemTag)

Function	Information
ToString()	[Return value] instance information (String) “ <i>ObjectTypeName</i> { <i>ClassName</i> , <i>Name</i> , <i>Status</i> , <i>Standby</i> , <i>Simulate</i> }”
GetPortName()	[Return value] Port Name (String)
GetPort()	[Return value] Port (Port) The instance of port related with this device is returned. Nil is returned if this function is failed.
GetStandbyDevicePath()	[Return value] Full path of standby device(String)
GetStandbyDevice()	[Return value] Standby device (Device) The instance of standby-device for this device is returned. Nil is returned if the redundant-communication is not specified.

Note

Device Access Class is derived from Group Access Class. The members (properties, functions) of Group Access Class can be used.

5.3.3.4 Struct Access Class

Struct Access Class (Struct)

Member	Type	Attribute	Information
TypeName	String	R	Type name

Function	Information
ToString()	[Return value] instance information (String) “ <u>ObjectTypeName</u> { <u>ClassName</u> , <u>Name</u> , <u>StructTypeName</u> }”

Note

Struct Access Class is derived from Group Access Class. The members (properties, functions) of Group Access Class can be used.

5.3.3.5 Port Access Class

Port Access Class (Port)

Member	Type	Attribute	Information
ObjectType	Number	R	Object type (Dxp.ObjectType.**)
ObjectClassName	String	R	Object type name (Example : EthernetPort)
Path	String	R	Path
Name	String	R	Port Name
Config	String	R	Setting parameter (csv format)

Function	Information
ToString()	[Return value] instance information (String) “ <u>ObjectTypeName</u> { <u>ClassName</u> , <u>Name</u> }”

5.3.3.6 Event Script Access Class

Event Script Access Class (EventScript)

Member	Type	Attribute	Information
ObjectType	Number	R	Object type (Dxp.ObjectType.**)
ObjectClassName	String	R	Object type name (Example : EventScript)
Path	String	R	Path
Name	String	R	Event Script Name
Config	String	R	Setting parameter (csv format)
Enabled	Boolean	R/W	Event status.

Function	Information
ToString()	[Return value] instance information (String) “ <u>ObjectTypeName</u> { <u>ClassName</u> , <u>Name</u> }”

Run()	Run Event Script manually. If Event of the called script is MQTT Subscribe, it is executed without processing regardless of the contents of the script. [Return value] None <pre><Sample1> local device = @d("Device1"); local event = device:GetEvent("script01"); event:Run(); <Sample2> local event = @e("Device1.script01"); event:Run();</pre>
RunMqttRecvMsg(sTopic, sMessage)	Run Event Script manually. If Event of the called script is not MQTT Subscribe, it is executed without processing regardless of the contents of the script. [Argument value] sTopic : Topic name (String) sMessage : Message (String) [Return value] None <pre><Sample1> local device = @d("Device1"); local event = device:GetEvent("script01"); event:RunMqttRecvMsg("Topic1" , "Hello!"); <Sample2> local event = @e("Device1.script01"); event:RunMqttRecvMsg("Topic1" , "Hello!");</pre>

5.3.3.7 Method Access Class

Method Access Class (Method)

Member	Type	Attribute	Information
ObjectType	Number	R	Object type (Dxp.ObjectType.**)
ObjectClassName	String	R	Object type name (Example : Method)
Path	String	R	Path
Name	String	R	Event Script Name
Config	String	R	Setting parameter (csv format)
Enabled	Boolean	R/W	Event status.

Function	Information
ToString()	[Return value] instance information (String) “<u>ObjectTypeName</u>{ <u>ClassName</u>, <u>Name</u>}”
Run()	Run Method manually. [Return value] None <pre><Sample1> local device = @d("Device1"); local method = device:GetMethod("Method01"); method:Run(); <Sample2> local method = @m("Device1.method01"); method:Run();</pre>

5.3.3.8 HTTP Client Access Class

6.3.3.8 HTTP Client Access Class (HttpClient)

Member	Type	Attribute	Information
ObjectType	Number	R	Object type (Dxp.ObjectType.**)
ObjectClassName	String	R	Object type name (Example : Method)
Path	String	R	Path
Name	String	R	Client Name
HostAddress	String	R	HostAddress
PortNo	Number	R	Port Number.
Header	String	R	Header
Timeout	Number	R	Timeout
RequestQueuesCount	Number	R	Request Queues Count

Function	Information
ToString()	[Return value] instance information (String) “ <i>ObjectTypeName</i> { <i>ClassName</i> , <i>Name</i> }”
CreateRequest()	Create a HTTP request class [Return value] request class of HTTP (HttpRequest)
Send (HttpRequest)	Send the HTTP request (Asynchronous) [Argument value] request class of HTTP (HttpRequest) [Return value] request id (Number) <Sample:DxpLink GetDevice> local http = @httpclient("Http01"); local req = http.CreateRequest(); req.Body = "<?xml version=¥"1.0¥" encoding=¥"UTF-8¥"?><Transportation Command=¥"GetDevice¥"/>"; req.MethodType = "POST"; local id = http.Send(req); local res = http.CheckEnd(id); while res.FinCode > 0 do res = http.CheckEnd(id); end *If CheckEnd is not executed, responses will be accumulated in DeviceXPlorer. If the number of accumulated responses exceeds 4096, the transmission will fail.
SendWithID(HttpRequest, ID)	Send the HTTP request with Id (Asynchronous). [Argument value] HttpRequest: request class of HTTP (HttpRequest) ID: request id (Number) [Return value] request id (Number) <Sample:DxpLink GetDevice(Cancele)> local http = @httpclient("Http01"); local req = http.CreateRequest(); req.Body = "<?xml version=¥"1.0¥" encoding=¥"UTF-8¥"?><Transportation Command=¥"GetDevice¥"/>"; req.MethodType = "POST"; local id = http.SendWithID(req,100); http.CancelRequest(id);

CancelRequest(ID)	cancel the http send request. [Argument value] ID: request id (Number) [Return value] canceled request id (numeric). A negative value indicates a failure.
CheckEnd(ID)	Check the complete of send the HTTP and received the response. [Argument value] ID: request id (Number) [Return value] response class of HTTP (HttpResponse)
SendSync(HttpRequest)	Send the HTTP request (Synchronous). [Argument value] HttpRequest: request class of HTTP (HttpRequest) [Return value] response class of HTTP (HttpResponse) <Sample:DxpLink GetDevice> local http = @httpclient("Http01"); local req = http.CreateRequest(); req.Body="<?xml version='1.0' encoding='UTF-8'><Transportation Command='GetDevice'>"; req.MethodType = "POST"; local res = http.SendSync(req);
GetHeader(key)	Get value of HTTP client header [Argument value] key:Key of header (String) [Return value] value for key(String)

5.3.3.9 HTTP request class

HTTP request class (HttpRequest)

Member	Type	Attribute	Information
ContentType	String	R/W	ContentType
Body	String	R/W	Body
Header	String	R	Header
MethodType	String	R/W	MethodType(example:GET, PUT, POST, DELETE, OPTIONS, HEAD, TRACE)
QueryPath	String	R/W	QueryPath
KeepAlive	Boolean	R/W	KeepAlive
Priority	Number	R/W	Priority

Function	Information
SetHeader (key, value)	[Argument value] key : Key of Header(String) value : Value of Header(String) [Return value] None
GetHeader (key)	[Argument value] key:Key of header (String) [Return value] value for key(String)

5.3.3.10 HTTP response class

HTTP request class(HttpResponse)

Member	Type	Attribute	Information
ContentType	String	R/W	ContentType
Body	String	R/W	Body
Header	String	R	Header
Result	Boolean	R/W	Result
StatusCode	Number	R/W	StatusCode

Reason	String	R/W	Reason
FinCode	Number	R/W	FinCode

Function	Information		
GetHeader (key)	[Argument value]	key:Key of header (String)	
	[Return value]	value for key(String)	

5.3.3.11 MQTT Client Access Class

6.3.3.11 MQTT Client Access Class(MqttClient)

Member	Type	Attribute	Information
ObjectType	Number	R	Object type (Dxp.ObjectType.**)
ObjectClassName	String	R	Object type name (Example : MqttClient)
Path	String	R	Path
Name	String	R	Name
HostAddress	String	R	HostAddress
Timeout	Number	R	Timeout
ClientId	String	R	ClientID
UserName	String	R	UserName
Password	String	R	Password
UseWill	Boolean	R	Use Will
WillTopic	String	R	Topic name of will
WillMessage	String	R	Message of will
WillQos	Number	R	QoS of Will
WillRetain	Boolean	R	Retain of Will

Function	Information
ToString()	[Return value] instance information (String) “ <i>ObjectTypeName</i> { <i>ClassName</i> , <i>Name</i> }”
CreatePublishMessage()	Create the Publish Message class. [Return value] Publish message (MqttPublishMessage)
Publish (publishMessage)	Send the publish message [Argument value] publishMessage : Publish message (MqttPublishMessage) [Return value] request ID(Number). A negative value indicates a failure. <Sample> <pre>local mqtt= @mqttclient("Mqtt01"); local publishMsg = mqtt:CreatePublishMessage(); publishMsg.Topic= "topic01"; publishMsg.Message= "{kwy:value}"; mqtt:Publish(publishMsg);</pre>
PublishWithID(publishMessage,ID)	Send the publish message with id. [Argument value] publishMessage : Publish message (MqttPublishMessage) ID: request ID(Number). [Return value] request ID(Number). A negative value indicates a failure. <Sample> <pre>local mqtt= @mqttclient("Mqtt01"); local publishMsg = mqtt:CreatePublishMessage (); publishMsg.Topic= "topic01"; publishMsg.Message= "{kwy:value}"; mqtt:PublishWithID(publishMsg,10);</pre>

CancelRequest(ID)	cancel the send request. [Argument value] ID: request id (Number) [Return value] canceled request id (numeric). A negative value indicates a failure. <Sample> <pre> local mqtt = @mqttclient ("Mqtt01"); local publishMsg = mqtt.CreatePublishMessage 0; publishMsg.Topic = "topic01"; publishMsg.Message = "{kwy:value}"; local id = mqtt.PublishWithID(publishMsg,10); mqtt.CancelRequest(id); </pre>
-------------------	--

5.3.3.12 MQTT Publish Message Class

MQTT Publish Message Class(MqttPublishMessage)

Member	Type	Attribute	Information
Topic	String	R/W	Topic name
Message	String	R/W	Message
UseRetain	Boolean	R/W	Use Retain
Qos	Number	R/W	Qos

5.3.3.13 Time Class

Time Class (Time)

Member	Type	Attribute	Information
High	Number	R/W	High Value
Low	Number	R/W	Low Value

Function	Information
Format (format)	[Argument] format : Time format (String) * elision is OK. [Return value] Time String (String). If the format fails, empty is returned. <Sample> <pre> local time = Dxp.GetCurrentTime0; Dxp.LogInfoMessage(time:Format("%Y/%m/%d %H:%M:%S")); </pre> Specify format of COleDateTime::Format0.
Diff(Time)	Get difference of the time of Second [Argument] Time : Time class (Time) [Return value] difference second time (Number)
Year0	[Return value] Year (Number)
Month0	[Return value] Month. 1 to 12 (Number)
Day0	[Return value] Day. 1 to 31 (Number)
Hour0	[Return value] Hour. 0 to 23 (Number)
Min0	[Return value] Minute. 0 to 59 (Number)
Sec0	[Return value] Second. 0 to 59 (Number)
WDay0	[Return value] Week day 1(Sunday) to 7 (Saturday) (Number)

5.3.3.14 Json Object Class

Json Object Class (JsonObject)

Member	Type	Attribute	Information
ObjectClassName	String	R	Object type name (Example : JsonObject)

Function	Information
ToString()	[Argument] None [Return value] value(String)
ToNumeric()	[Argument] None [Return value] value(Number)
ToBoolean()	[Argument] None [Return value] value(Boolean)
GetString(key)	[Argument] key(String) [Return value] value of key(String)
GetNumeric(key)	[Argument] key(String) [Return value] value of key(Number)
GetBoolean(key)	[Argument] key(String) [Return value] value of key(Boolean)
GetObject(key)	[Argument] key(String) [Return value] JsonObject Class(JsonObject)
SetString(key, value)	[Argument] key(String), value(String) [Return value] Return Flag(Boolean). TRUE : succeed /FALSE : failed
SetNumeric(key, value)	[Argument] key(String), value(Number) [Return value] Return Flag(Boolean). TRUE : succeed /FALSE : failed
SetBoolean(key, value)	[Argument] key(String), value(Boolean) [Return value] Return Flag(Boolean). TRUE : succeed /FALSE : failed
SetObject(key, value)	[Argument] key(String), value (JsonObject Class) [Return value] Return Flag(Boolean) . TRUE : succeed/FALSE : failed
HasString(key)	[Argument] key(String) [Return value] Exist Flag(Boolean) . TRUE : exist/FALSE : not exist
HasNumeric(key)	[Argument] key(String) [Return value] Exist Flag(Boolean) . TRUE : exist/FALSE : not exist
HasBoolean(key)	[Argument] key(String) [Return value] Exist Flag(Boolean) . TRUE : exist/FALSE : not exist
Remove (key)	[Argument] key(String) [Return value] Return Flag(Boolean) . TRUE : succeed/FALSE : failed
GetStringElement(index)	[Argument] index(Number) [Return value] value of index(String)
GetNumericElement(index)	[Argument] index(Number) [Return value] value of index(Number)
GetBooleanElement(index)	[Argument] index(Number) [Return value] value of index(Boolean)
GetObjectElement(index)	[Argument] index(Number) [Return value] JsonObject Class(JsonObject)
SetStringElement(value)	[Argument] value(String) [Return value] Return Flag(Boolean) . TRUE : succeed/FALSE : failed
SetNumericElement(value)	[Argument] value(Number) [Return value] Return Flag(Boolean) . TRUE : succeed/FALSE : failed

GetQualityString(<i>quality</i>)	This will get a string of quality flags. [Argument] quality : Quality flag of the tag(Number) [Return value] Quality flag string (String)
IsGoodQuality(<i>quality</i>)	This will determine whether the flag is Good quality. [Argument] quality : Quality flag of the tag (Number) [Return value] Good quality flag is whether (Boolean)
IsBadQuality(<i>quality</i>)	This will determine whether the flag is Bad quality. [Argument] quality : Quality flag of the tag (Number) [Return value] Bad quality flag is whether (Boolean)
IsUncertainQuality(<i>quality</i>)	This will determine whether the flag is Uncertain quality. [Argument] quality : Quality flag of the tag (Number) [Return value] Uncertain quality flag is whether (Boolean)
CopyTagsValue(<i>dst, src</i>)	This will copy hierarchically to another group, the value of the tag in the group. [Argument] dst : Destination path (String) src : Souce path (String) [Return value] Whether or not the operation was successful (Boolean)
SyncRead(<i>tags</i>)	This is done in a synchronous read. [Argument] tags : Table that contains the tag(Tag) [Return value] Result Success / Failure (Boolean) <Example> <pre> local tags = {}; -- Set the tag to be read for i=1, 10, 1 do tags[i] = @t("Device1.Tag" .. i); end -- Synchronous read local ret = Dxp.SyncRead(tags); -- Read the results local sum = 0; for i=1, 10, 1 do sum = sum + tags[i].Value; end </pre>
SyncWrite(<i>tags</i>)	This is done in a synchronous write. [Argument] tags : Table that contains the tag (Tag) [Return value] Result Success / Failure (Boolean) <Example> <pre> local tags = {}; -- Set the tag to be written. Set the value of writing. for i=1, 10, 1 do tags[i] = @t("Device1.Tag" .. i); tags[i].Value = 123; end -- Synchronous write local ret = Dxp.SyncWrite(tags); </pre>
LogMessage(<i>msg</i>)	This script outputs the log message. [Argument] msg : Output message (String)
LogInfoMessage(<i>msg</i>)	This will output the info message. [Argument] msg : Output message (String)
LogWarningMessage(<i>msg</i>)	This will output a warning message.

		[Argument] msg : Outoput message (String)
	LogErrorMessage(<i>msg</i>)	This will output an error message. [Argument] msg : Output message (String)
	GetCurrentTime()	This gets the current time. [Return value] Current time (Time)
	And(num1, num2) Or(num1, num2) Xor(num1, num2) Nor(num1, num2) Nand(num1, num2)	This is done in various bit operations (AND / OR / XOR / NOR / NAND). [Argument] num1 : Target number 1(Number) num2 : Target number 2(Number) [Return value] The result of the bitwise operation various (Number)
	Not(num)	This is done bit operations (NOT) [Argument] num : Target number (Number) [Return value] The result of the bitwise operation (NOT) .(Number)
	Sleep(num)	Suspends the execution of the current thread until the time-out interval elapses. [Argument] num : The time interval for which execution is to be suspended, in milliseconds. (Number) * Please note that execute Sleep will affect other scripts on the same device.
	CreateJsonObject()	Return the JsonObject class. [Argument] None [Return value] JsonObject class(JsonObject)
	ParseJson(JsonMsg)	Parse a Json string and return a JsonObject class. [Argument] JsonMsg : Json string (String) [Return value] JsonObject class (JsonObject)

5.3.5 Access Macros

DeviceXPlorer provides following macros originally. Access Macros creates the instance of tag, device, group, struct, method and port.

Macro	Content
@(<i>fullpath</i>)	Absolute access macro. You can access the structure, methods, and to the port, by enclosed in double quotes followed by at sign, the absolute path The nil is returned if the target object is not found. <Example1> @("Device1.Tag1").Value = 1; <Example2> local tag = @("Device1.Tag1"); tag.Value = 1;
@t(<i>fullpath</i>)	Absolute access macro for tag. This macro creates the instance of the Tag Access Class.
@d(<i>fullpath</i>)	Absolute access macro for device. This macro creates the instance of the Device Access Class.
@g(<i>fullpath</i>)	Absolute access macro for group. This macro creates the instance of the Group Access Class.
@s(<i>fullpath</i>)	Absolute access macro for struct. This macro creates the instance of the Struct Access Class.
@m(<i>fullpath</i>)	Absolute access macro for method. This macro creates the instance of the Method Access Class. This is the sample to call "SetAuto" method when this method is created in the "Device1" device.

	<Example1> local ret = @m("Device1.SetAuto"):Run(); <Example2> local method = @m("Device1.SetAuto"); local ret = method:Run();
@e(<i>fullpath</i>)	Absolute access macro for event script. This macro creates the instance of the Event Script Access Class.
@p(<i>fullpath</i>)	Absolute access macro for port. This macro creates the instance of the event Port Access Class.
@httpclient(<i>fullpath</i>)	Absolute access macro for HTTP Client. This macro creates the instance of the HTTP Client Access Class.
@mqttclient(<i>fullpath</i>)	Absolute access macro for MQTT Client. This macro creates the instance of the HTTP Client Access Class.
@@ @@(<i>relativePath</i>)	Relative access macros. This macros that you can access the object with a relative path. How to write in the description of enclosed in double quotation marks at the two and a relative path or two at sign. The nil is returned if the target object is not found. @@("RelativePath") - - - Access to child objects relative @@ - - - Themselves (this) This is the sample to output message which has value of "Tag1" tag and name of "Device1" device, when this tag is created in this device. <Example1> Dxp.LogInfoMessage(@@.Name .. "/" .. @@("Tag1"):ToStringValue0); <Example2> local device = @@; local tag = @@("Tag1"); Dxp.LogInfoMessage(device.Name .. "/" .. tag:ToStringValue0);
@@t(<i>relativePath</i>)	Relative access macro for tag. This macro creates the instance of the Tag Access Class.
@@g(<i>relativePath</i>)	Relative access macro for group. This macro creates the instance of the Group Access Class.
@@s(<i>relativePath</i>)	Relative access macro for struct. This macro creates the instance of the Struct Access Class.
@@m(<i>relativePath</i>)	Relative access macro for method. This macro creates the instance of the Method Access Class.
@@e(<i>relativePath</i>)	Relative access macro for event script. This macro creates the instance of the Event Script Access Class.

5.3.6 Priority of Finding Object

Indicates a preference for objects that can be retrieved from the Object access macro($\hat{=}$ GetObject function / GetObjectType function). For example, when there are same pot name and device name, this will be replaced by the process of obtaining a device. If you want to get the port explicitly, please use the port access macro.

Rank	Object
High	Tag, Method
	Group, Structure instance
	Event script
	Device
Low	Port

5.3.7 Units of Running Script

DeviceXPlorer has Lua Engines per device. Scripts run on threads separated by devices. The script on the "Device1" waits for finishing of waiting timeout and retrying, when you use synchronous reading (Dxp.SyncRead) for the child tag(e.g. Device1.Tag1) while the controller is down. The script on the "Device2" is not affected from waiting for "Device1".

5.3.8 About Script Writing

If a new value is assigned to the Value property of a tag access object or a new value is set by the SetValue method of an arrangement tag access object, script engine will do asynchronous write and will write a value to PLC.

Asynchronous writing is not performed when a synchronous write-in function is called after value substitution with a script.

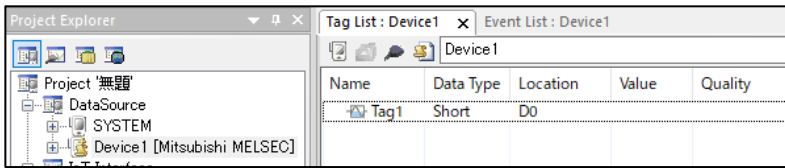
However, asynchronous writing is performed when value substitution is carried out after a synchronous write-in function call.

Demands of asynchronous writing is queueing if script writes repeatedly to the device which is down. The demands are processed and write values to the controller in turn, when the device restored. DeviceXPlorer outputs an error to the message view if the queue count reaches the upper limit. You should stop writing in the script while device is down. Otherwise, you can control by the "Reduce write-queue" at the Project Properties.

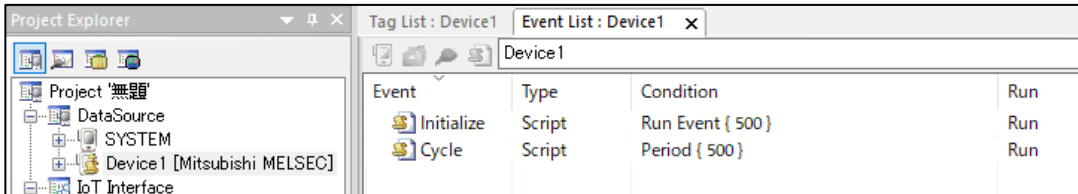
5.4 Samples

5.4.1 Basic Example

This sample shows the basic example. At first, make the tag (Device1.Tag001) in the Device1.



Make two event scripts in the Device1.



Initialize script(Run Script) initializes the global variables and functions.

Scripts are executed in the processing engine per devices (Refer to 5.3.7 Units of Running Script). So global variables and functions are able to be shared between the device and child groups.

Script Example (Initialize)

```
1  -- define the function
2  function CalcAdd( val1, val2 )
3      return val1 + val2;
4  end
6
7  -- declare global variable
8  g_counter = 0;
```

Cycle script(Period) implements the processing that continues always running. You can use global variables and functions which is defined in the Initialize script in this script. You can use “local” keyword to declare the local variable. If you want to access the tags, you can use the Tag Access Macro (@t, @@t). For accessing the properties of tag, you can use dot “.”. For calling the method of tag, you can use colon “:” (This is the specification which can omit “this” object for first argument in the Lua language).

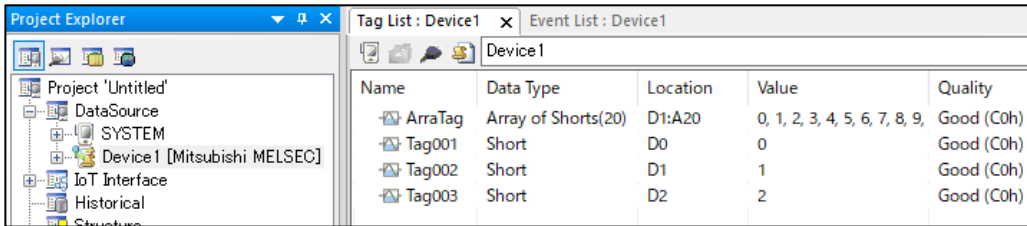
You can use the functions which is defined under the “Dxp” namespace, e.g. “Dxp.LogMessage()” function.

Script Example (Cycle)

```
1  -- Count up global variable
2  g_counter = g_counter + 1;
3
4  -- Getting tag access object with local variable
5  local tag = @@t( "Tag001" );
6
7  -- Calling the function and writing the value of tag
8  tag.Value = CalcAdd( g_counter, 1000 );
9
10 -- Output message log (WarningLog)
11 Dxp.LogWarningMessage( "counter=" .. g_counter .. " / tag=" .. tag.ToStringValue() );
12
13 -- If example
14 if( g_counter % 5 == 0 ) then
15     -- For example
16     local i;
17     for i=0, 10, 1 do
18         Dxp.LogInfoMessage( "i=" .. i );
19     end
20 end
```

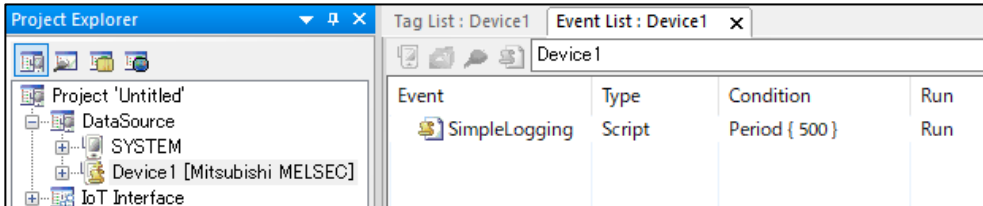
5.4.2 Easy CSV Logging

This sample shows a simple example about collecting the values of tags to the CSV file.
At first, make the tags in the Device1.



Name	Data Type	Location	Value	Quality
ArraTag	Array of Shorts(20)	D1:A20	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	Good (C0h)
Tag001	Short	D0	0	Good (C0h)
Tag002	Short	D1	1	Good (C0h)
Tag003	Short	D2	2	Good (C0h)

Make event script in the Device1.



Event	Type	Condition	Run
SimpleLogging	Script	Period { 500 }	Run

Implement the CSV logging about three tags.

Script Example (SimpleLogging)

```
1 local f, msg = io.open("C:\\¥¥Work¥¥Log.csv", "a+");
2
3 -- Check the open result
4 if (f == nil) then
5     Dxp.LogErrorMessage("File Open Error (" .. msg .. ")");
6     return;
7 end
8
9 -- Make CSV
10 local text = "";
11 text = text .. Dxp.GetCurrentTime():Format();
12 text = text .. "," .. @@t("Tag001"):ToStringValue();
13 text = text .. "," .. @@t("Tag002"):ToStringValue();
14 text = text .. "," .. @@t("Tag003"):ToStringValue();
15
16 -- Write to file
17 f:write(text .. "¥n");
18
19 -- Close the file
20 f:close();
21
22 -- Dump log
23 Dxp.LogMessage(text);
24
```

If you change the code to use the tag of array type, you can realize easily the logging about big continuous data.

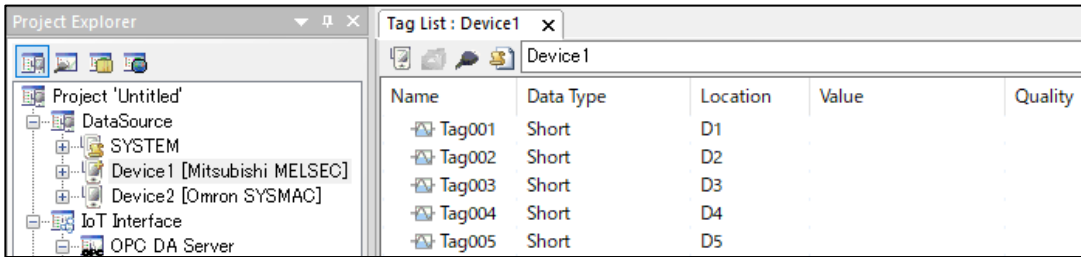
Script Example (SimpleLogging)

```
:
9 -- Make CSV
10 local text = "";
11 text = text .. Dxp.GetCurrentTime():Format();
12 text = text .. "," .. @@t("ArrayTag"):ToStringValue();
13
14 -- Write to file
15
```

5.4.3 Example which defines the class in the external file (Bridge)

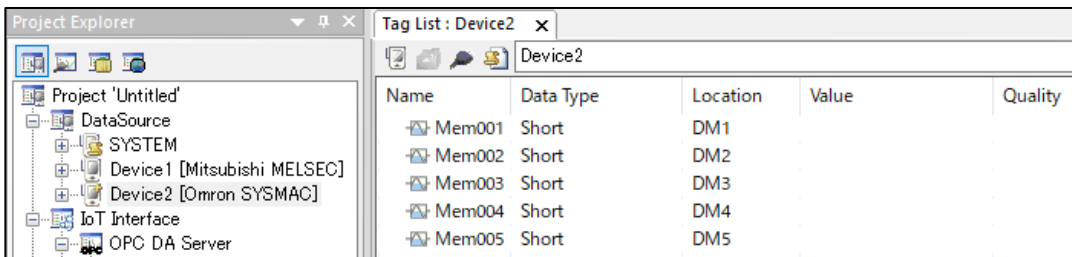
This sample shows how you can define the class in the external file with the bridge example.

At first, make the tags (Device1.Tag001 – 005, Device2.Mem001 - 005) in the devices (Device1, Device2).



The screenshot shows the 'Project Explorer' on the left with a tree view containing 'Project 'Untitled'', 'DataSource', 'SYSTEM', 'Device1 [Mitsubishi MELSEC]', 'Device2 [Omron SYSMAC]', 'IoT Interface', and 'OPC DA Server'. The 'Tag List : Device1' window is open on the right, showing a table of tags for Device1.

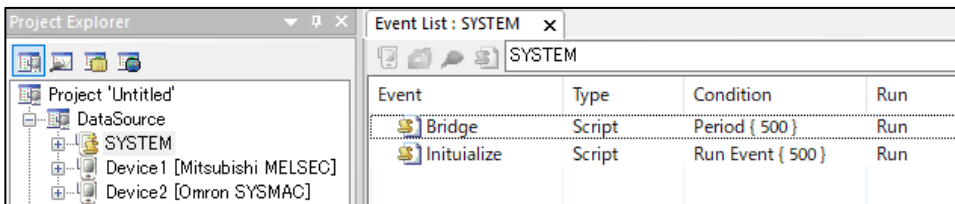
Name	Data Type	Location	Value	Quality
Tag001	Short	D1		
Tag002	Short	D2		
Tag003	Short	D3		
Tag004	Short	D4		
Tag005	Short	D5		



The screenshot shows the 'Project Explorer' on the left with the same tree view. The 'Tag List : Device2' window is open on the right, showing a table of tags for Device2.

Name	Data Type	Location	Value	Quality
Mem001	Short	DM1		
Mem002	Short	DM2		
Mem003	Short	DM3		
Mem004	Short	DM4		
Mem005	Short	DM5		

Make the event script in the SYSTEM.



The screenshot shows the 'Project Explorer' on the left with the same tree view. The 'Event List : SYSTEM' window is open on the right, showing a table of events for the SYSTEM.

Event	Type	Condition	Run
Bridge	Script	Period { 500 }	Run
Initialize	Script	Run Event { 500 }	Run

Initialize script (Run Script) loads the external script file. So you can use BridgeSample class which is defined in the external file.

About the instance of BridgeSample class you created, it stores to the global variable.

By calling the method of BridgeSample instance, you can register the path that you want.

Script Example (Initialize)

```
1 require("lua/BridgeSample");
2
3 g_bridge = BridgeSample.New();
4 g_bridge:Add("Device1.Tag001", "Device2.Mem001");
5 g_bridge:Add("Device1.Tag002", "Device2.Mem002");
6 g_bridge:Add("Device1.Tag003", "Device2.Mem003");
7 g_bridge:Add("Device1.Tag004", "Device2.Mem004");
8 g_bridge:Add("Device1.Tag005", "Device2.Mem005");
9
```

Bridge script(Period) does the bridge by calling the method of BridgeSample instance.

Script Example (Bridge)

```
1
2 g_bridge:BridgeForward();
3
```

Make the external script file in the “Lua” folder.

You should make “Lua” folder in the Bin Folder(C:\Program Files (x86)\TAKEBISHI\DeviceXplorer OPC Server 7\Bin\) if it is first time to making the external script file.

Script Example (<Bin Folder>\Lua\BridgeSample.lua)

```
1  if(LUA_TKB_BRIDGE_SAMPLE_INCLUDE == nil) then
2      LUA_TKB_BRIDGE_SAMPLE_INCLUDE = 1;
3
4      -- *****
5      -- BridgeData class
6      -- *****
7      BridgeData =
8      {
9          New = function(_DstTagPath, _SrcTagPath)
10             return
11             {
12                 SrcTagPath = _SrcTagPath,
13                 DstTagPath = _DstTagPath,
14                 SrcTag = nil,
15                 DstTag = nil,
16
17                 -- //////////////////////////////////
18                 -- PrepareTag method
19                 -- //////////////////////////////////
20                 PrepareTag = function(this)
21                     this.SrcTag = Dxp.GetTag("", this.SrcTagPath);
22                     this.DstTag = Dxp.GetTag("", this.DstTagPath);
23
24                     local found = true;
25
26                     if ( this.SrcTag == nil ) then
27                         Dxp.LogErrorMessage( "BridgeData - Not found SrcTag [" ..
28 this.SrcTagPath .. "]" );
29                         found = false;
30                     end
31                     if ( this.DstTag == nil ) then
32                         Dxp.LogErrorMessage( "BridgeData - Not found DstTag [" ..
33 this.DstTagPath .. "]" );
34                         found = false;
35                     end
36
37                     if ( found == false ) then
38                         this.SrcTag = nil;
39                         this.DstTag = nil;
40                     end
41
42                     return found;
43                 end,
44
45                 -- //////////////////////////////////
46                 -- HasValue method
47                 -- //////////////////////////////////
48                 HasValues = function(this)
49                     if ( this.SrcTag.HasValue() == false ) then
50                         return false;
51                     end
52                     if ( this.DstTag.HasValue() == false ) then
53                         return false;
54                     end
55                     return true;
56                 end,
57
58                 -- //////////////////////////////////
59                 -- BridgeForwarded method
```

```

60      -- //////////////////////////////////
61      BridgeForward = function(this)
62          this.DstTag.Value = this.SrcTag.Value;
63      end,
64
65      -- //////////////////////////////////
66      -- BridgeInverse method
67      -- //////////////////////////////////
68      BridgeInverse = function(this)
69          this.SrcTag.Value = this.DstTag.Value;
70      end
71  };
72  end
73  };
74
75  -- *****
76  -- BridgeSample
77  -- *****
78  BridgeSample =
79  {
80      New = function()
81          return
82          {
83              Bridges = {},
84
85              -- //////////////////////////////////
86              -- Add method
87              -- //////////////////////////////////
88              Add = function(this, dstPath, srcPath)
89                  -- Create Bridge Data
90                  local data = BridgeData.New( dstPath, srcPath );
91                  this.Bridges[#this.Bridges + 1] = data;
92
93                  -- At first, get tag to start communication with PLC for getting values
94                  data.PrepareTag();
95              end,
96
97              -- //////////////////////////////////
98              -- BridgeTags method
99              -- //////////////////////////////////
100             BridgeForward = function(this)
101                 -- Wait for getting values
102                 for i=1, #this.Bridges, 1 do
103                     local data = this.Bridges[i];
104                     if ( data.HasValues() == false ) then
105                         return false;      -- failed
106                     end
107                 end
108
109                 -- Do Bridge
110                 for i=1, #this.Bridges, 1 do
111                     local data = this.Bridges[i];
112                     data.BridgeForward();
113                 end
114
115                 return true; -- succeeded
116             end
117         };
118     end
119 };
120
end

```

5.4.4 Example which defines the class in the external file (Timer)

This sample shows how you can define the class in the external file with the timer example.

At first, make the tag (SYSTEM.Running).

Project Explorer

Project 'Untitled'

DataSource

SYSTEM

IoT Interface

Tag List : SYSTEM

Event List : SYSTEM

SYSTEM

Name	Data Type	Location	Value	Quality
Running	Bool	GB0	On	Good (Coh)

Make the event script in the SYSTEM.

Project Explorer

Project 'Untitled'

DataSource

SYSTEM

IoT Interface

Historical

Tag List : SYSTEM

Event List : SYSTEM

SYSTEM

Event	Type	Condition	Run
Initialize	Script	Period { 500 }	Run
StartStop	Script	TagChange { 500, Change, "SYSTEM.Running" }	Run
Timer	Script	Period { 500 }	Run

Initialize script (Run Script) loads the external script file. So you can use TimerSample class which is defined in the external file.

About the instance of TimerSample class you created, it stores to the global variable.

Script Example (Initialize)

```

1  require("lua/TimerSample");
2
3  -- Timer
4  g_timer = TimerSample.New();
5  g_timer:Stop();
6

```

StartStop script(SYSTEM.Running : Change(Value)) starts and stops the timer by calling the method of TimerSample instance.

スクリプト例(StartStop)

```

1  if (@t("SYSTEM.Running").Value == true) then
2      g_timer:Start();
3      Dxp.LogInfoMessage("Timer Start");
4  else
5      g_timer:Stop();
6      Dxp.LogInfoMessage("Timer Stop");
7  end
8

```

Timer script(Period) checks the time-up(10 second) and outputs the log message and repeats it.

Script Example (Timer)

```

1  if (g_timer.Elapsed(10) == true) then
2      Dxp.LogInfoMessage("Time up !");
3      g_timer:StartAdd(10);
4  end
5

```

Make the external script file in the "Lua" folder.

You should make "Lua" folder in the Bin Folder(C:\Program Files (x86)\TAKEBISHI\DeviceXPlorer OPC Server 7\Bin) if it is first time to making the external script file.

Script Example (<Bin Folder> \Lua\TimerSample.lua)

```

1  if (LUA_TKB_TIMER_SAMPLE_INCLUDE == nil) then
2      LUA_TKB_TIMER_SAMPLE_INCLUDE = 1;
3
4      -- *****
5      -- TimerSample class

```

```

6      -- *****
7      TimerSample =
8      {
9          New = function()
10             return
11             {
12                 Running = true,
13                 LastTime = os.time(),
14
15                 -- //////////////////////////////////
16                 -- Start method
17                 -- //////////////////////////////////
18                 Start = function( this )
19                     this.LastTime = os.time();
20                     this.Running = true;
21                 end,
22
23                 -- //////////////////////////////////
24                 -- StartAdd method
25                 -- //////////////////////////////////
26                 StartAdd = function( this, sec )
27                     local nowTime = os.time();
28                     this.LastTime = this.LastTime + sec;
29
30                     if ( this.Elapsed(sec) == true ) then
31                         this.LastTime = nowTime - sec;
32                     end
33
34                     this.Running = true;
35                 end,
36
37                 -- //////////////////////////////////
38                 -- Elapsed method
39                 -- //////////////////////////////////
40                 Elapsed = function( this, sec )
41                     if ( this.Running == false ) then
42                         return false;
43                     end
44
45                     local planTime = this.LastTime + sec;
46                     local nowTime = os.time();
47
48                     local diffSec = os.difftime( planTime, nowTime );
49                     if ( diffSec <= 0 ) then
50                         lastTime = planTime;
51                         return true;    -- Time up !
52                     end
53
54                     return false;    -- Not time up
55                 end,
56
57                 -- //////////////////////////////////
58                 -- Stop method
59                 -- //////////////////////////////////
60                 Stop = function( this )
61                     this.Running = false;
62                 end
63             };
64         end;
65     };
66
67 end
68

```

6 Troubleshooting

6.1 Error Messages

Error messages showed in log view are listed below.

Message	Contents
Open error (Code=XXXX)	Error occurred when opening port
Send error (Code=XXXX)	Error occurred when transmitting
Receive error (Code=XXXX)	Error occurred when receiving
Response timeout error (wait=xxx, got=xxx)	Response timed out. Wait=xxx indicates the byte size that the OPC Server is waiting. Got=xxx indicates the byte size that have been received actually
Size error of transmitted message	Size of the data that was actually sent was different from that of the send data. This error occurs if the cable is disconnected when the computer uses RS232C.
Size error of received message	Size of the data that was actually received was different from that of the data that it was supposed to receive. This error also occurs in RS232C communication.
Header error of received message	The head of received data is abnormal.
Received NAK (Code=XXXX)	NAK is received from PLC.
Communication Error	Any other error
Sum check error	The sum check results of received data do not correspond to the data of the message.
Bad com port handle	Requested device mode cannot be configured.
Receive queue overflow	Receive queue is overflow.
Receive overrun	Overrun error occurred when receiving data. Communicate by lowering the baud rate.
Receive parity error	Parity error occurred.
Receive framing error	Framing error occurred.
Received BREAK	BREAK was found in channels.
I/O Error	I/O error occurred.
Transmit queue overflow	Transmit queue is overflow.

* Contents of Code=XXXX depend on the port you use.

6.1.1 The source of the Response timeout error in the Ethernet connection

The outbreak factor that the Response timeout error in Ethernet connection is follows mainly. The trouble of the network adapter may be the outbreak factor.

No.	Source	Solution
1	High network load (Chronic / Momentary)	Please reduce the workload of network.
2	CPU-hogging (Chronic / Momentary)	Please reduce the workload of CPU.
3	Windows ARP (the case which time-out error occurs every 10 minutes)	Please change ARP cache settings of Windows OS.
4	Time of time-out of Ethernet port is short.	Please extend the time of Time-out of Ethernet port on DeviceXPlorer.
Below is source of timeout error in case of UDP/IP (In case of TCP/IP, Open error occurs by these source 5 and 6.)		
5	PLC power is off or the LAN cable is break.	Please check the power condition and LAN cable.
6	The Ethernet parameter of PLC and DeviceXPlorer may be mistake.	Please check the Ethernet parameter of PLC and DeviceXPlorer.

Note

TCP/IP communication is stable communication protocol. So if you can select UDP/IP or TCP/IP, please use TCP/IP.

6.1.2 Ethernet Error Code (Winsock Error Code)

Code	Contents
10024	There are no available file descriptors.
10022	Socket is not bound yet to address.
10035	Connection could not be completed because of unblocking socket. Time-out error occurred.
10036	Windows socket operation that is currently executed is blocked.
10038	Descriptor is not a socket.
10039	Address is required.
10040	Datagram was shortened because it was too long to fit the specified buffer.
10048	The defined address is currently occupied. (This error happens when you set the number other than 0 as the port number.)
10049	The specified address is not available from the local machine.
10050	Windows socket implement found out an error of network subsystem.
10051	Now it is unable to reach network from host.
10052	Windows socket implement dropped a socket. You should reset the connection.
10053	Virtual circuit is aborted because of time-out error or some other error. Check Ethernet cable.
10054	Virtual circuit was reset from the Remote. This error happens when CPU has been reset and power has been turned off.
10055	There is no available buffer range. Socket cannot be connected. Please reduce the number of connections of TCP.
10056	Socket has already been connected.
10057	Socket is not connected.
10058	Socket was shut down.
10060	Connection was attempted but not completed because it timed out. Check the Ethernet channels and PCL programming.
10061	Connection was attempted but rejected. Previous connection remains in the PLC. Reset CPU or turn off the signal for open request. Check the port number.
258	Connecting PLC is failed by timeout. Please confirm whether the PLC works well or not.
39321	

6.1.3 Serial Error Code

Code	Contents
2	Specified port does not exist.
5	Specified port already open by other process.

6.2 Others

6.2.1 If the application screen does not display properly

If the screen layout information of the application stored in the registry of the Windows OS is corrupted for some reason, the screen of the application will not be displayed properly.

In that case, close the application, execute “InitializeWorkspace.reg” in the “Bin” folder of the installation directory, initialize the screen layout, and then restart the application.

7 Appendix

7.1 How to use of the OPC Test Client

Please refer Client Sample Manual.

You can open the manual at “Client Sample Manual” at “Client Sample” at “DeviceXPlorer” at “programs” at Start Menu of Windows.

7.2 About Sample Programs

Sample programs are stored in the Samples folder of the folder of DeviceXPlorer installation directory.

Please refer to the "Samples" folder of installation media for the other sample program.

For details, please refer to a DeviceXPlorer client sample guide.

7.3 User Support

TAKEBISHI offers a user support service to customers who have completed user registration and during the maintenance contract period. If you have any questions about how to set up or operate DeviceXPlorer, please contact the support service detailed below.

User Support

<Contact Form>

<https://www.faweb.net/en/support/form/>

TEL +81-75-325-2261

Open: 9:00 - 12:00 / 13:00 - 17:00

When you contact the support service, you will be asked to confirm the following information, so please have it ready.

- (1) Product name, serial number (written on the license certificate)
- (2) PC model, OS/service packs
- (3) PLC CPU type, communication unit model
- (4) Client software version/service pack/development language

Note

The support service may not be able to answer questions that do not relate to DeviceXPlorer, such as how to set up the hardware and device driver or how to program the sequence programs and sample programs.

7.4 Copyright Information

For the copyright information of third-party used in DeviceXPlorer, refer to the path below.

<Path>

Installation folder¥Doc¥TherdParty

DeviceXPlorer OPCServer
User's Guide
Server Edition

