TAKEBISHI

DeviceXPlorer OPC Server

User's Guide

(Server Edition)

Revision 028

July 2022

[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 6.7.0.1 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]

[Trademarks]

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)".

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 4 types of protocol (OPC, OPC UA, SuiteLink, and DxpLink) 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.

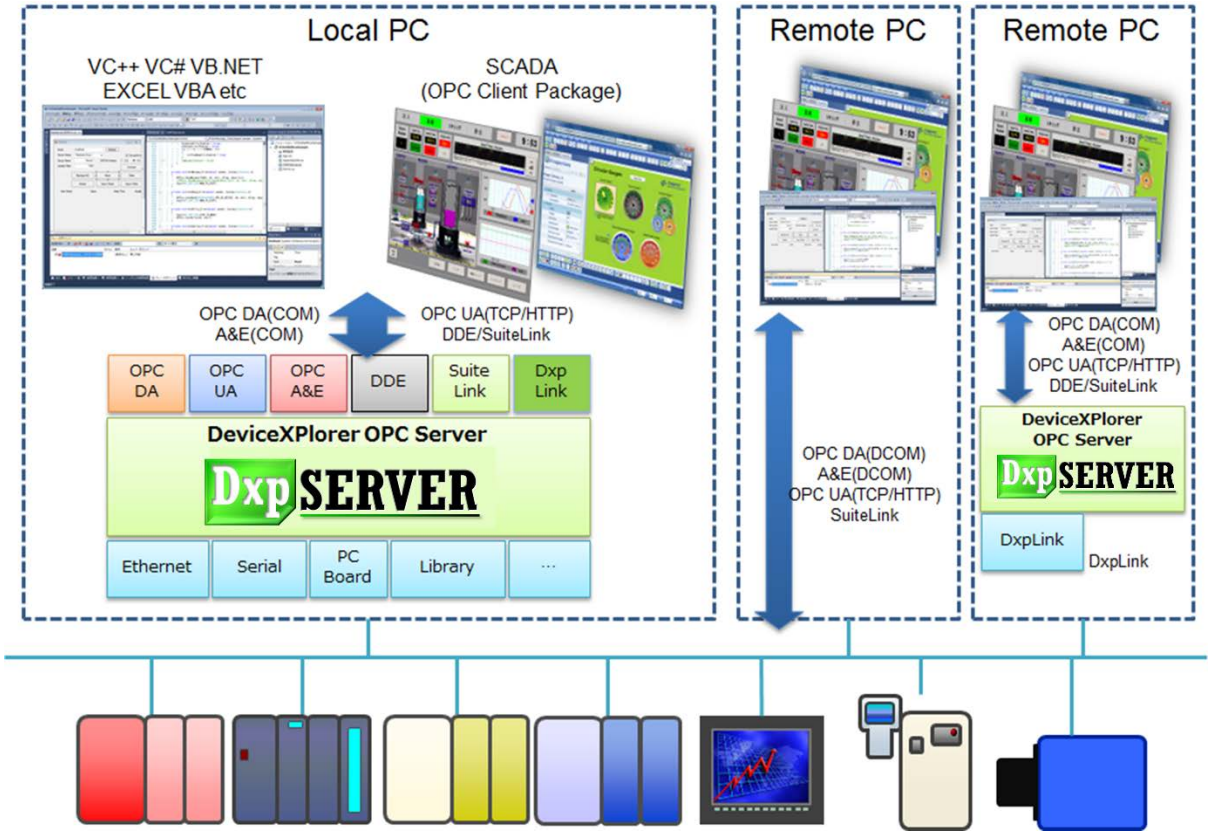
1.4 Licenses

One DeviceXPlorer license is required for each PC in which the software is installed.
The max number of accessible modules is 255 per one DeviceXPlorer.
There is no constraint on the number of clients.
Refer to the 3.6 for the difference between Enterprise edition, Advanced edition and Standard 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.

[Supported Development Languages]

Interface		Excel	VB6.0	VB.NET	C#.NET	VC6.0/8.0	SCADA
OPC DA	Custom interface					OK	OK
	Automation interface*1	OK	OK	OK			
	RCW			OK	OK		
OPC UA				OK	OK	OK	OK
OPC AE			OK	OK	OK	OK	OK
SuiteLink*2							OK
DxpLink		It can be used only for communication between DeviceXPlorer.					

*1 Automation interface don't support Windows x64 Edition.

*2 For using SuiteLink interface, installing Wonderware product is necessary.

2.2 Operating Environment

DeviceXPlorer operates in the following environments:

Item	Description
Computer	Computer running Windows
OS	Windows 11 (21H2) *1
	Windows 10 IoT Enterprise 2019 LTSC
	Windows 10 IoT Enterprise 2016 LTSB
	Windows 10 (1809, 1903, 1909, 2004, 20H2, 21H2) *1
	Windows 8.1
	Windows 7
	Windows Server 2022
	Windows Server 2019
	Windows Server 2016
	Windows Server 2012 R2
	Windows Server 2012
	Windows Server 2008 R2
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
- YASKAWA Robot Controller 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	
	EtherNet/IP	
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
Modbus suite	Ethernet (Modbus/TCP)	Products supported by Modbus Mitsubishi Modbus module / Electric Controller / EcoMonitorLight OMRON K6CM Hakko MONITOUCH Koyo DirectLOGIC/KOSTAC series IDEC MICRO Smart azbil(Yamatake) controller Hioki E. SmartSite2300 series 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
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	
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	Products supported by IEC61850MMS
DNP3 Master	Ethernet	Products supported by DNP3
EtherNet/IP suite	EtherNet/IP	Products supported by EtherNet/IP
Modbus Server suite	Ethernet	Modbus/TCP Client GRAPHTEC 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)
Shared Memory	Shared memory	Internal memory of DeviceXPlorer

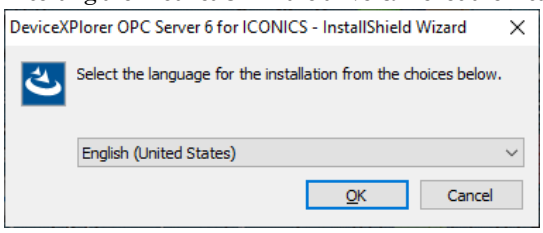
* For target module details, refer to the "User's Guide (PLC Communication Edition)".

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

3 Installation

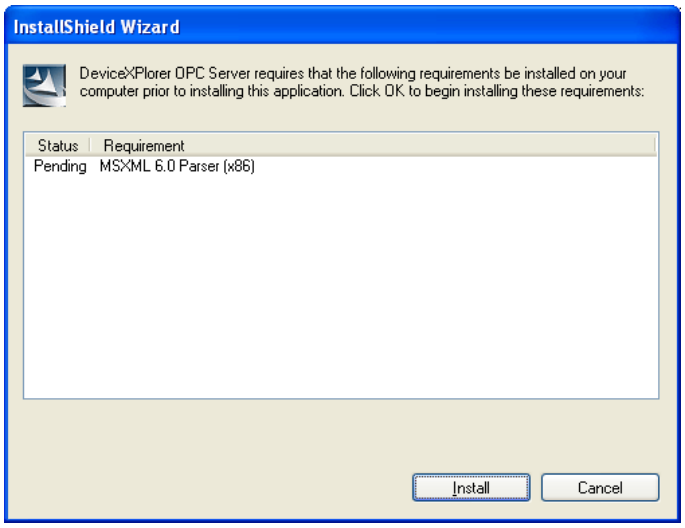
3.1 Install Procedure

(1) Inserting the Product CD in the drive launches the installer. Choose setup language in dialog.

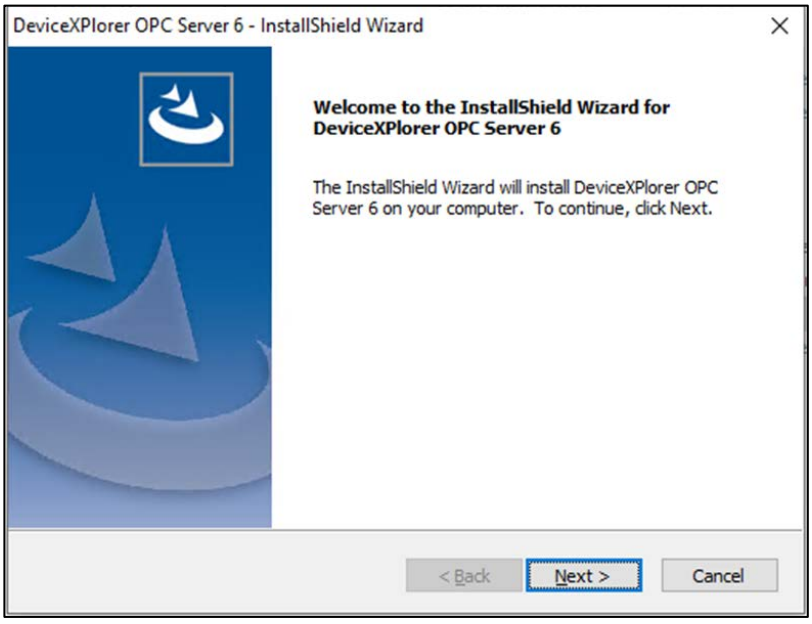


Important
- You must have Administrator rights to set up. - .NET Framework 3.5 setup is required before installing the 32-bit application.

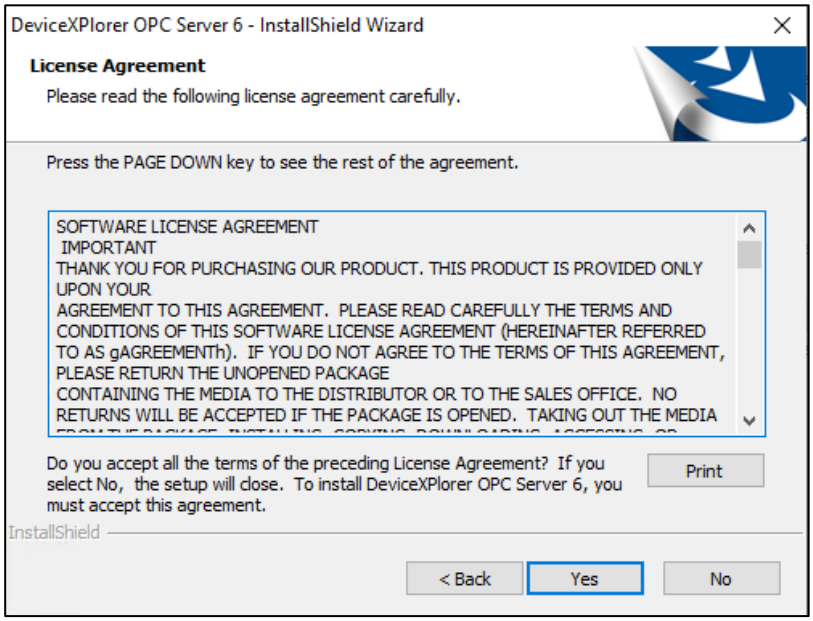
(2) Install the components which are necessary to operate DeviceXPlorer. Press "Install".



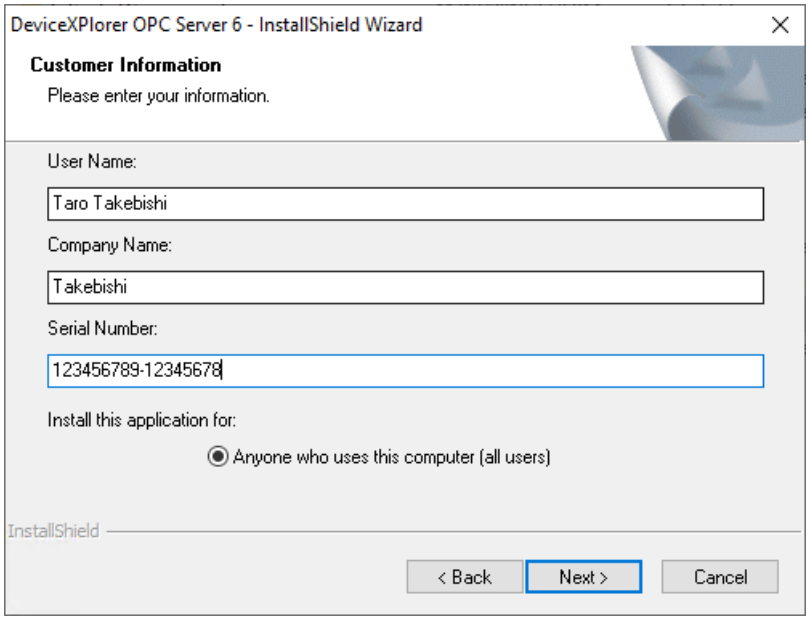
(3) When the InstalShield Wizard is displayed, press "Next."



(4) Agree with the License Agreement, and click "Yes."



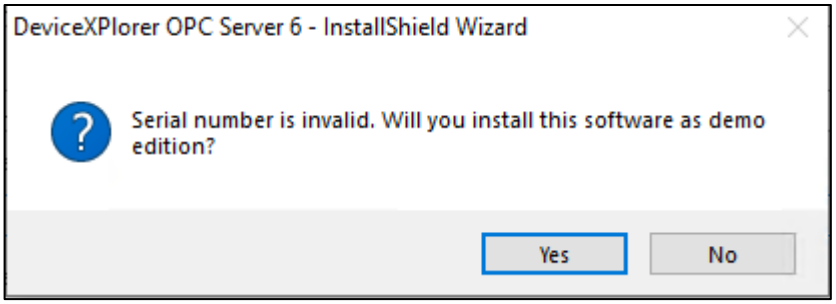
(5) Input user information and Serial Number. Click "Next."



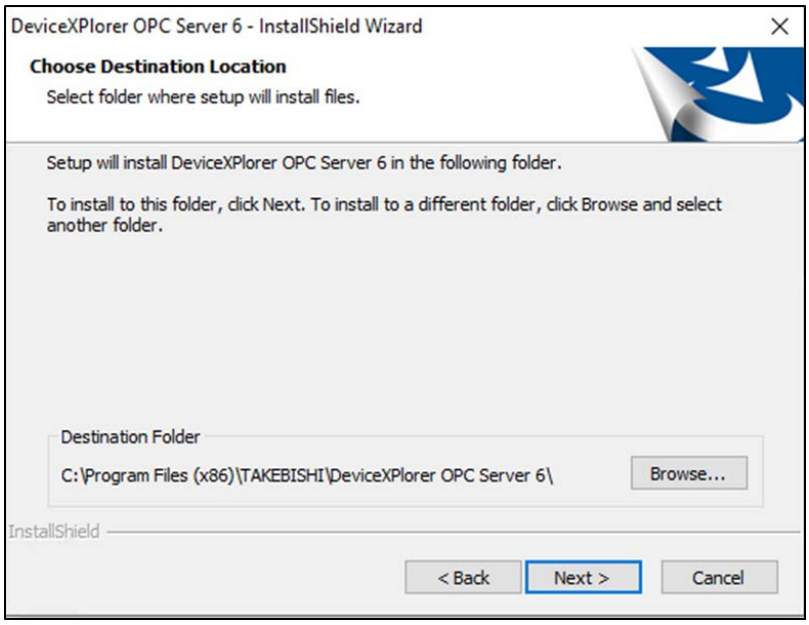
Note

Please input the Serial Number that has been described to the product accurately.

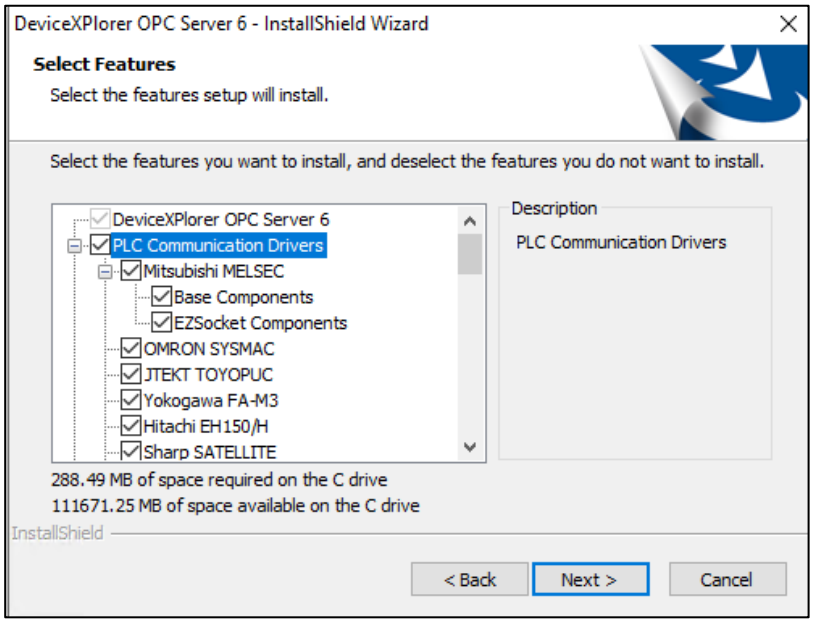
If inputting invalid Serial Number, following dialog will be displayed. If you would like to use as demo edition, please input a hyphen symbol "-" in Serial Number.



(6) Select where you wish to install the system or components you wish to install as may be necessary.

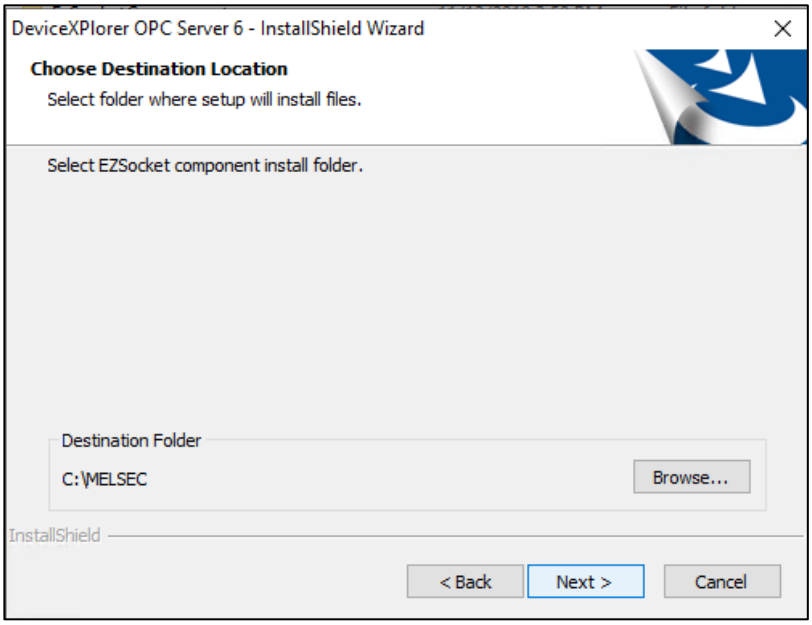


(7) Select Installing component.



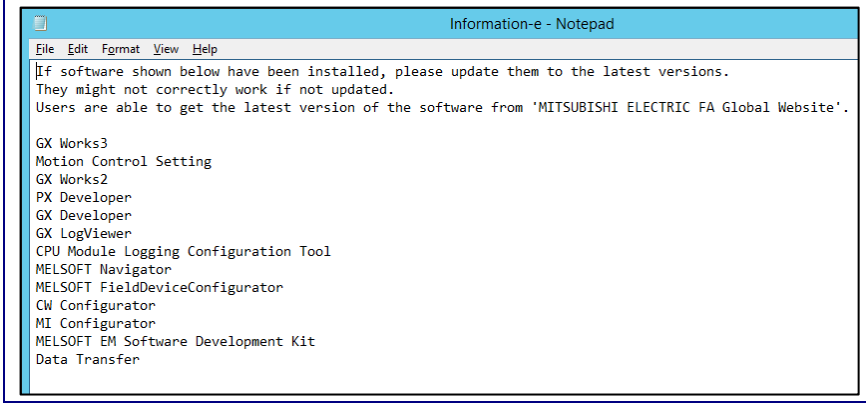
Note
The communication function selected here can be used. If you want to use all communication functions, check all of them.

(8) If MELSEC EZSocket component is selected, select the installation destination of EZSocket component.

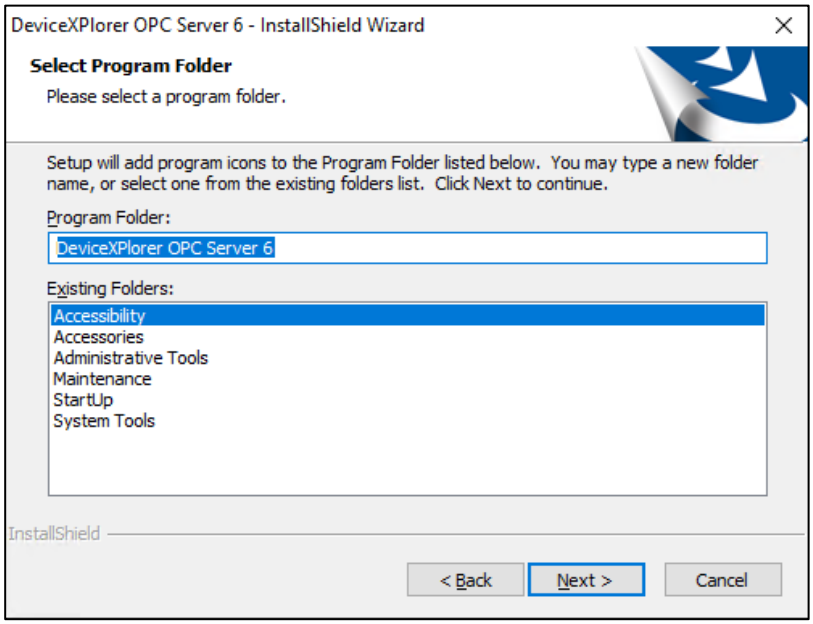


Note

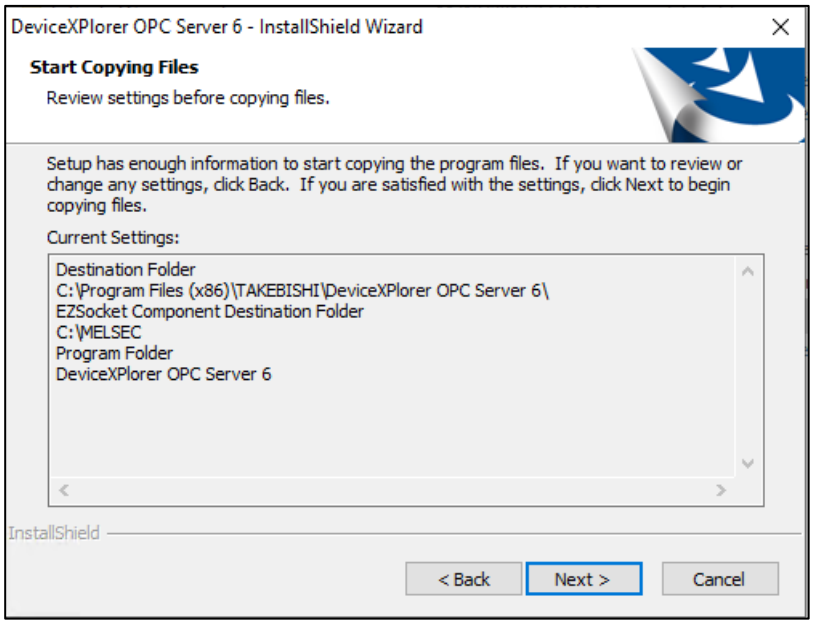
When EZSocket components installation, following message may be displayed. In that case, update the software shown in the message.



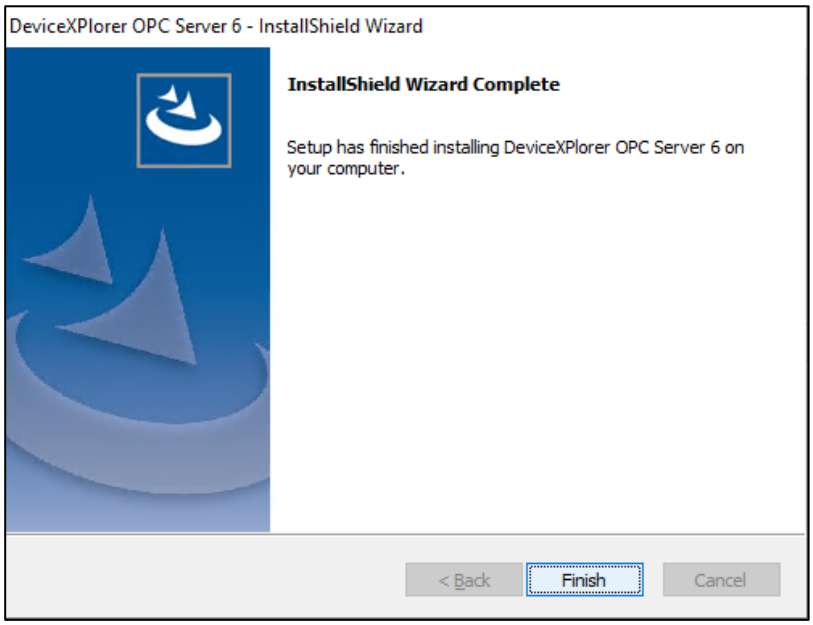
(9) Select the program folder.



(10) Preparation for installation is now complete. Press 'Install' to start installation.



(11) When the Windows component is updated, it is necessary to restart PC.



3.2 Installation Files

DeviceXPlorer installation sets up the following files.

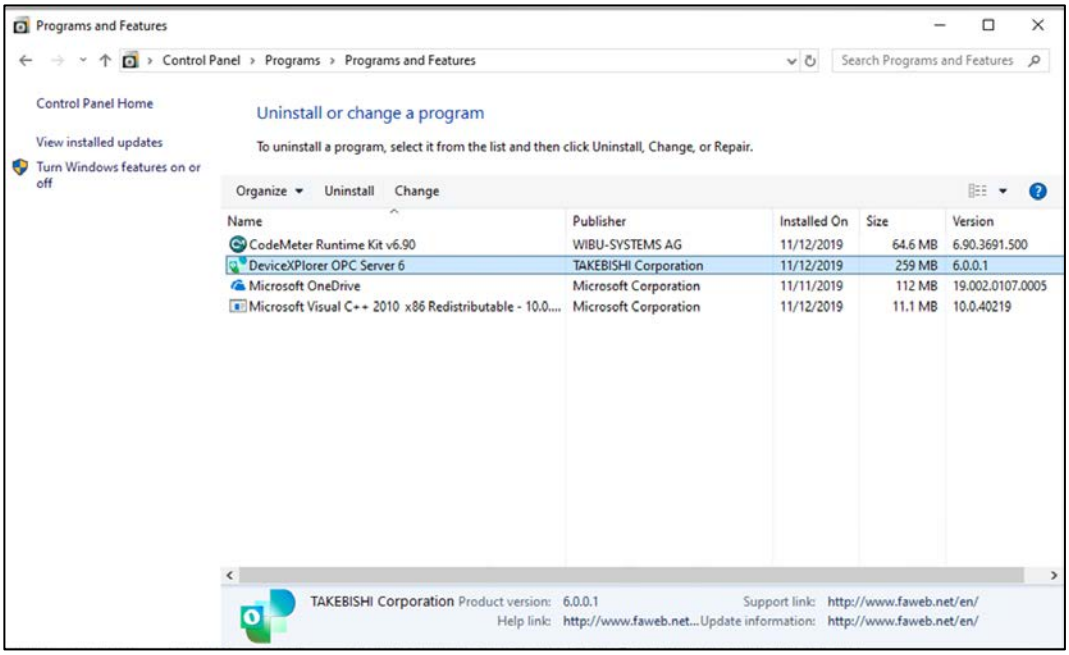
Description	Setup path	Note
DeviceXPlorer	Installation folder ¥BIN	
User's Guide	Installation folder ¥Doc	Server Edition PLC Communication Edition Software license agreement
Option File	Installation folder ¥Option	License Request File template
OPC DA components	Windows system folder	OPC DA2.0 / 3.0 custom interface OPC DA2.0 automation interface
Windows components	Windows system folder	VC runtime library

3.3 Uninstall Procedure

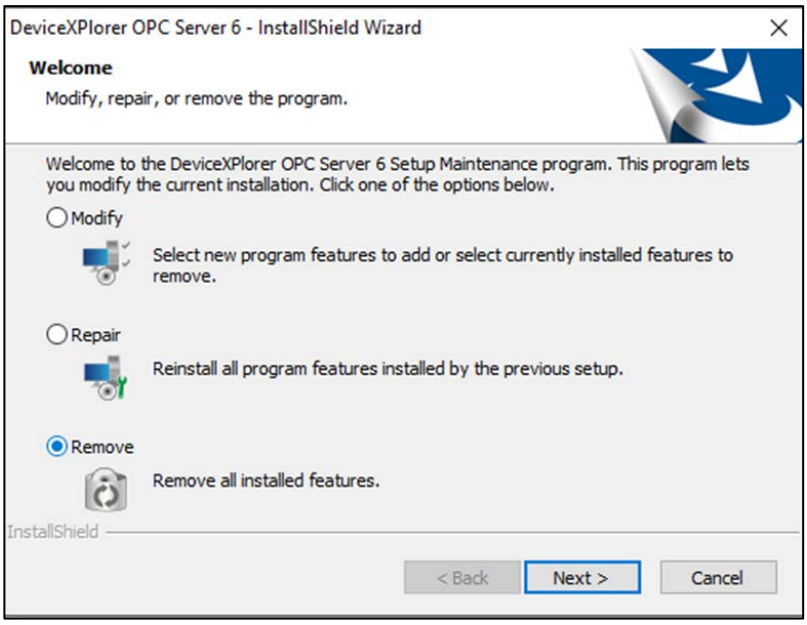
You can uninstall DeviceXPlorer by following the Procedures below.

(1) Disconnect all connections from client applications and quit DeviceXPlorer.
If it is running as a service program, go to the "Control Panel -> Services", and stop "DeviceXPlorer OPC Server 6".

(2) In the Control Panel, select "Add/Remove Program"



(3) Remove "DeviceXPlorer OPC Server 6".



(4) Even after DeviceXPlorer has been uninstalled, project file created by the user will remain.

Important
If you have activated the software key, please deactivate the software key before uninstalling. (Refer to 3.4.12)

3.4 License Activation

License activation is necessary in order to run DeviceXPlorer properly.
The license agreement for our software products grants the right to install and use the product on a specific PC.
Depending on the product type, a software key or a hardware key may be required to operate the product.

Important
DeviceXPlorer license is activated through following 2 steps.. (1) Activation with serial number of DeviceXPlorer. (2) Activation with hardware key or software key

3.4.1 Serial Number

The serial number is a unique number that you receive when you purchase a product.
After purchase, the customer must register as a user based on this serial number.
Also, using this serial number, it is necessary to register the serial number on DeviceXPlorer.

Important
DeviceXPlorer periodically conducts a serial number check to see if there are multiple DeviceXPlorer running on PCs in the same LAN. Note that if DeviceXPlorer sharing the same serial number are found, operation will stop immediately.

3.4.2 Software Key

If you purchased software key product, "Ticket ID" is required to activate the software key. The "Ticket ID" can be acquired by registering the exact serial number at user registration after purchase, and it is required to activate the software.

3.4.3 Hardware Key

If your product requires a hardware key, both the serial number and the hardware key must be set up on the same PC.
When the hardware key included in the package is inserted into the PC where DeviceXPlorer is installed, the hardware key is activated.

Note
The hardware key must always be inserted to the PC while DeviceXPlorer is running..

3.4.4 License Status and Application Operation

You could check a table below shows the relations between license activation status and application operation and confirm the license status on the "License" screen of the "Help" menu.

License status	Status	Application operation
Normal Mode	OK(KEY)	The license is effective, so no further actions are required.
Demo Mode	DEMO	Operates in demo mode for 1 hour.
NO KEY	NO KEY	Hardware key or software key is required.
STOP	STOP	The operation stops. Activate license is required.
Temporary Mode	TERM	A temporary license has been applied. The license is valid until valid date. In software key edition, you need to activate the key within 14 days after installation. On the user registration page, enter your serial number and other required information, submit it. We will inform you of the ticket ID necessary for license activation. URL https://www.faweb.net/en/user-form/ If you can't get a ticket ID after user registration, would you contact us to the following email address? Email address: fa-support@takebishi.co.jp
GRACE Mode	GRACE	If the software key or hardware key becomes temporarily unrecognizable, the application will continue to operate under the original license state as a grace period of 180 minutes in GRACE mode. However, if the application has been running for a long time, this grace period may be extended up to 10 days.

The items on the "License" screen are as follows.

Item	Description
Component	[PLATFORM] component: Application base. For others, the component name for each connected device is displayed.
Version	[PLATFORM] component: License name is displayed. ENTERPRISE, ADVANCED, STANDARD or DEMO. For others, the version of the component is displayed.
Status	The license status of each component is displayed.
Serial Number	The applied serial number is displayed.
Key information	The applied key information is displayed.
Valid Date	The expiration date is displayed when the status is [TERM]. In GRACE mode, the remaining grace period is displayed.

3.4.5 Serial Number

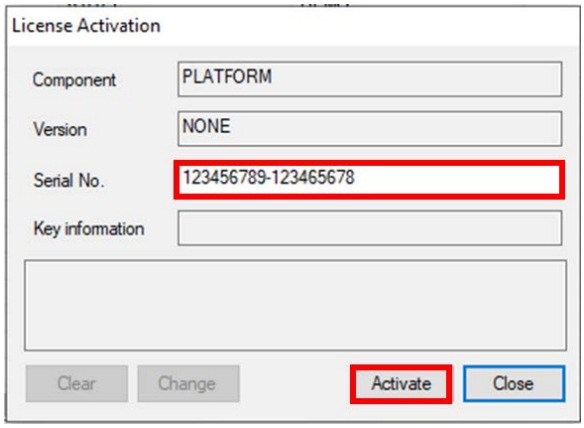
3.4.5.1 Serial Number Registration

DeviceXPlorer serial number authentication procedure.

(1) Open the license screen from "Help" on DeviceXPlorer.

(2) Select the “Component” to be authenticated, right-click and click “License Activation”.

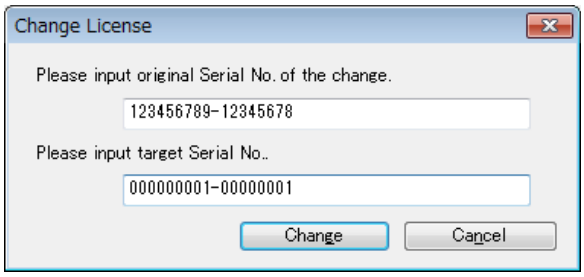
(3) Input the serial number and click “Activate” to complete.



3.4.5.2 Change the activated Serial Number

You could change the activated serial number.

It is necessary to input the original serial number and the target serial number.



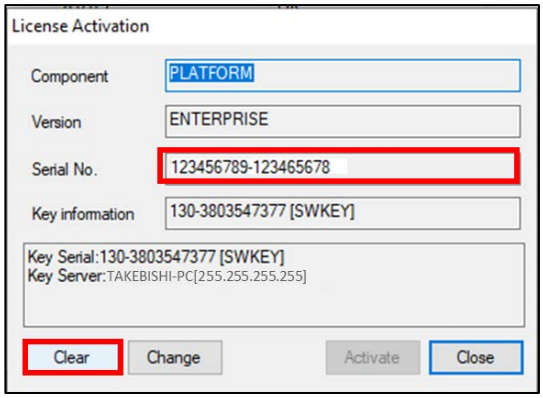
3.4.5.3 Deactivate the Serial Number.

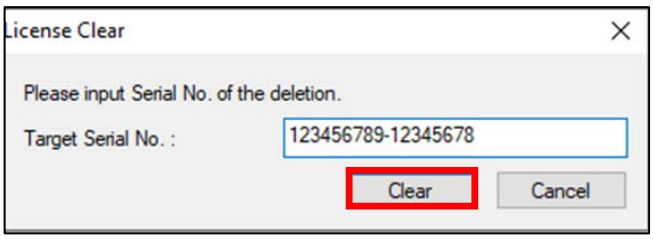
The Procedures of deactivate the serial number.

(1) Open the license screen from "Help" on the OPC server.

(2) Select the “Component” to be deactivated, right-click and click “License Activation”.

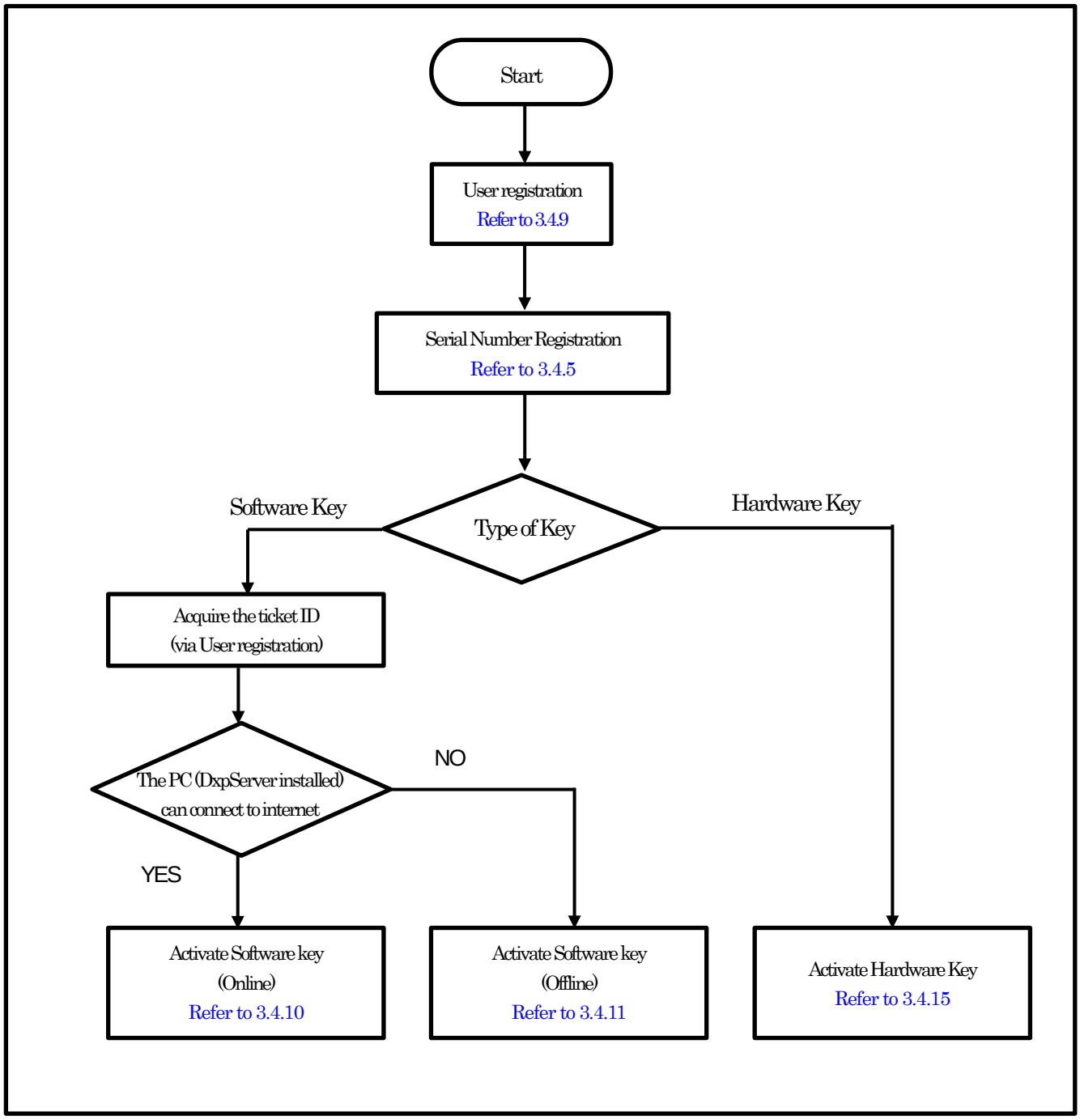
(3) Input the serial number and click “Clear” to complete.





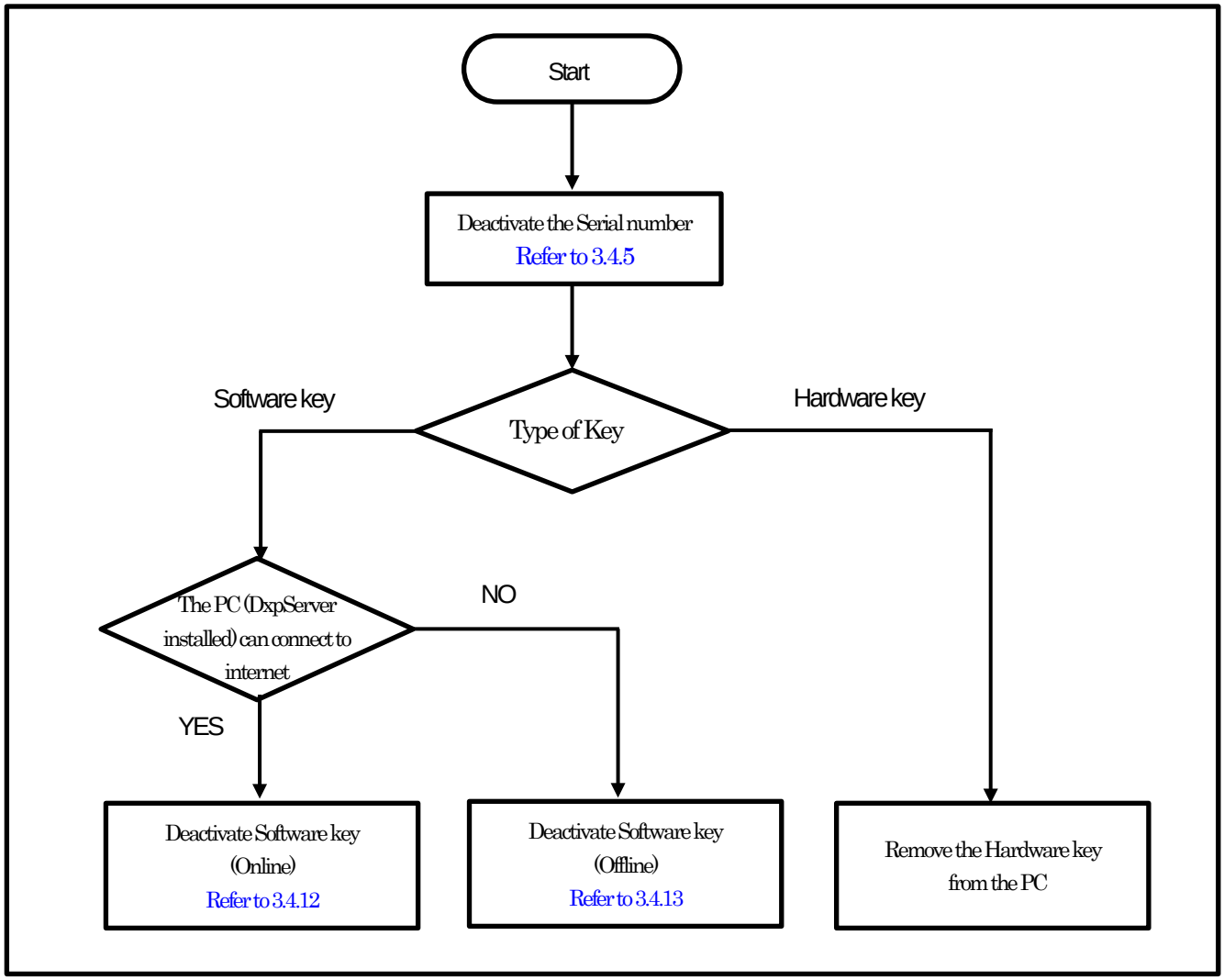
3.4.6 Activation of License Flow

The activation of license flow for software keys and hardware keys is shown below.



3.4.7 Deactivation of License Flow

The deactivation of license flow for software keys and hardware keys is shown below.



Important
If you are unable to deactivate the software key due to PC failure or such, please contact our support.

3.4.8 Transfer of License Flow

When transferring a license to another PC, deactivate the license at the source PC and activate the license at the destination PC.

3.4.8.1 Deactivate the license at the source PC

Deactivate the license at the source PC with licensed DeviceXPlorer by following steps..

(1) Deactivate the serial number of DeviceXPlorer.
(2) If hardware key is activated, remove the hardware key.
(3) If the software key is activated, deactivate the software key.
Refer to "3.4.12 Deactivate the Software Key (Online)" or "3.4.13 Deactivate the Software Key (Offline)".
(4) Uninstall DeviceXPlorer.

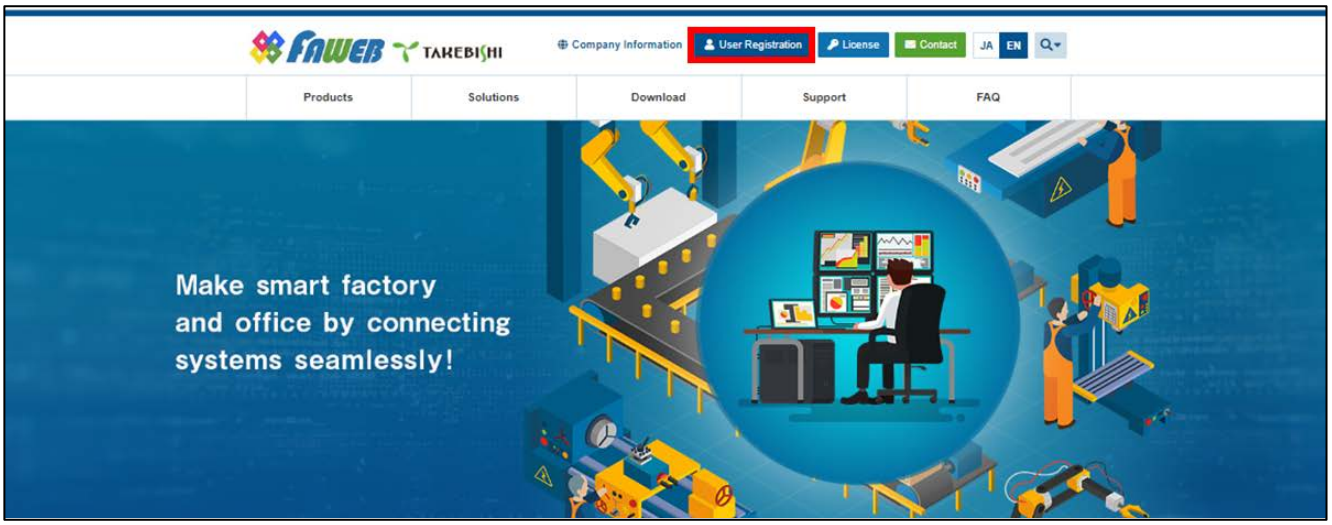
3.4.8.2 Activate the license in the transferred PC

Activate the license at the destination PC with DeviceXPlorer installed by following the steps below.

(1) Register the serial number of DeviceXPlorer.
(2) In case of hardware key, attach the hardware key to the PC.
(3) In case of software key, activate the software key.
Refer to "3.4.10 Activate Software key (Online)" or "3.4.11 Activate the Software Key (Offline)".

3.4.9 User Registration

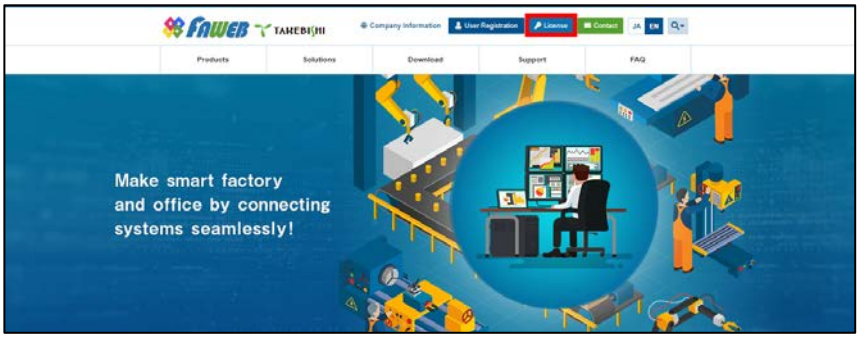
Please complete user registration on the FAWEB website, (FAWEB: https://www.faweb.net/en/) with the correct serial number (9 digits -8 digits) that you acquired when purchased.
If you purchased the software key product, a ticket ID will be sent to your e-mail when you complete use registration on the FAWEB site. The ticket ID is required to activate the software key.
If you purchase Hardware key, Ticket ID will not be sent.



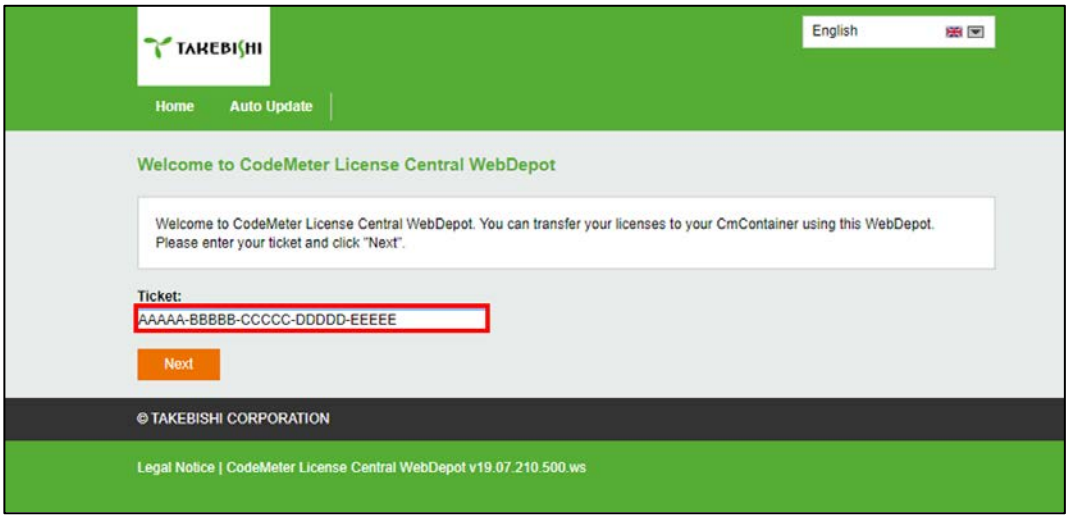
3.4.10 Activate the Software Key (Online)

The following shows the procedure for activating software key when a PC with DeviceXPlorer installed can connect to the Internet.
* The ticket ID is required to activate. Please complete 3.4.9 User Registration to acquire the ticket ID..

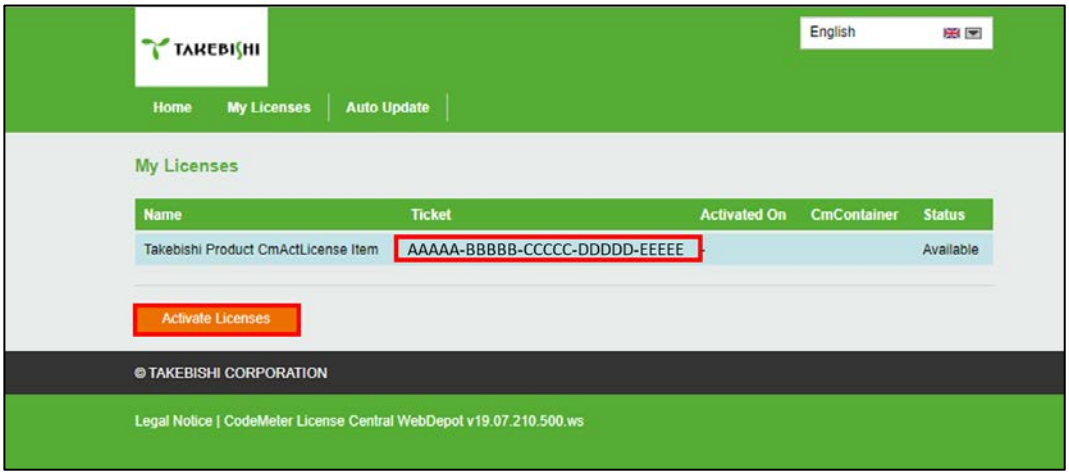
(1) Open the license registration web page ("License") in FAWEB.
(The license registration web page: http://license.faweb.net/)



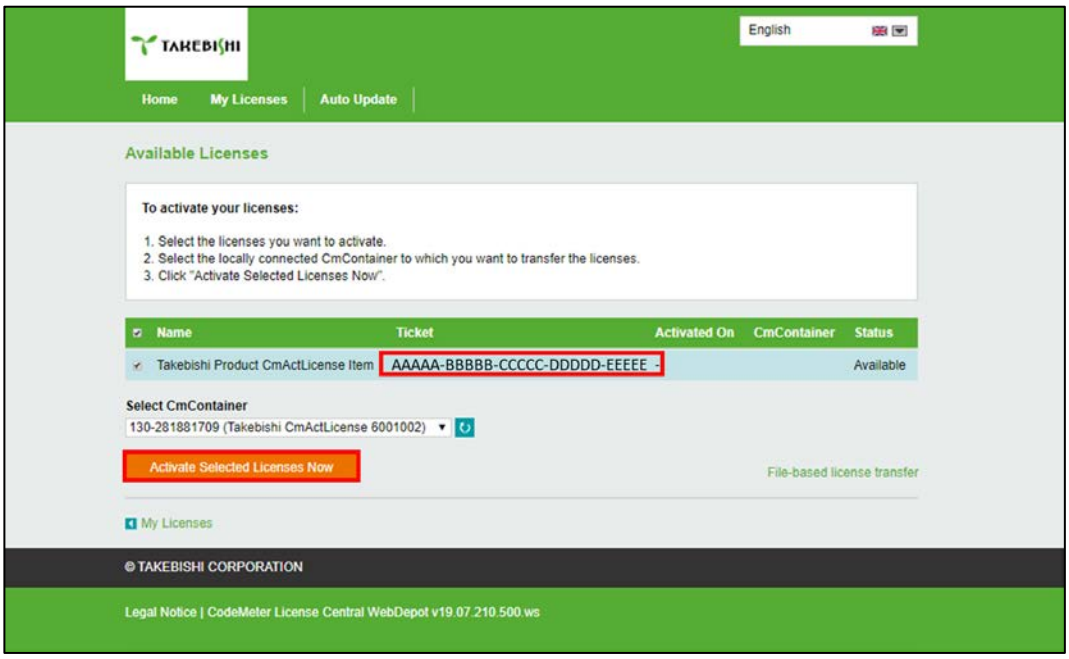
(2) Input the ticket ID in the "Ticket", which you acquired at user registration on the license registration page, and click "Next.



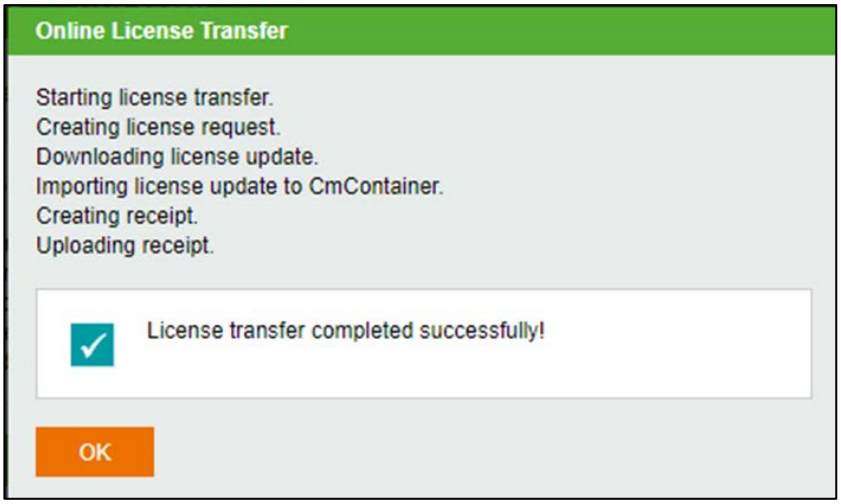
(3) My Licenses screen is displayed. Confirm that the ticket ID is correctly filled, and click "Activate Licenses".



(4) Click “Activate Selected License Now”.
Use the ticket ID acquired by this operation to activate the license.



(5) The following screen is displayed when process is completed.
When the message “License transfer completed successfully!” is displayed, Software Key Activation is complete.



3.4.11 Activate the Software Key (Offline)

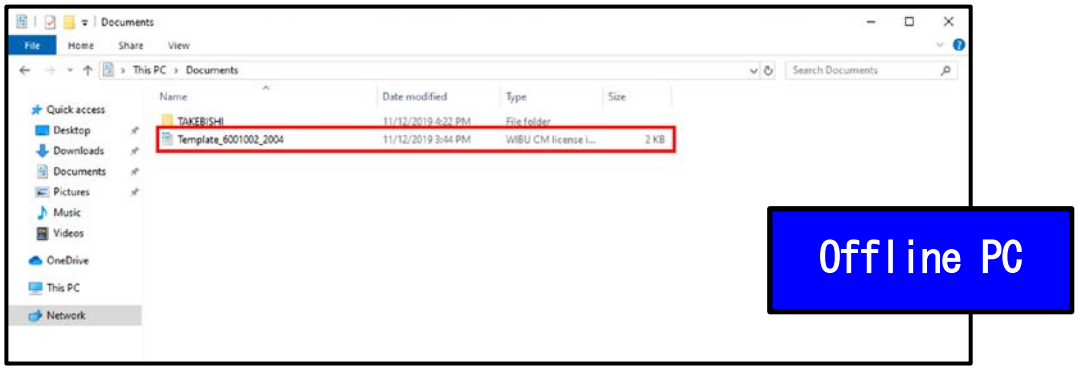
The following shows the Software Key Activation procedure when a PC on which DeviceXPlorer is installed cannot connect to the Internet.

* Use a PC that can access the Internet (hereinafter referred to as an Online PC) to activate a license for a PC (hereinafter referred to as an Offline PC) that has a DeviceXPlorer that cannot access the Internet.

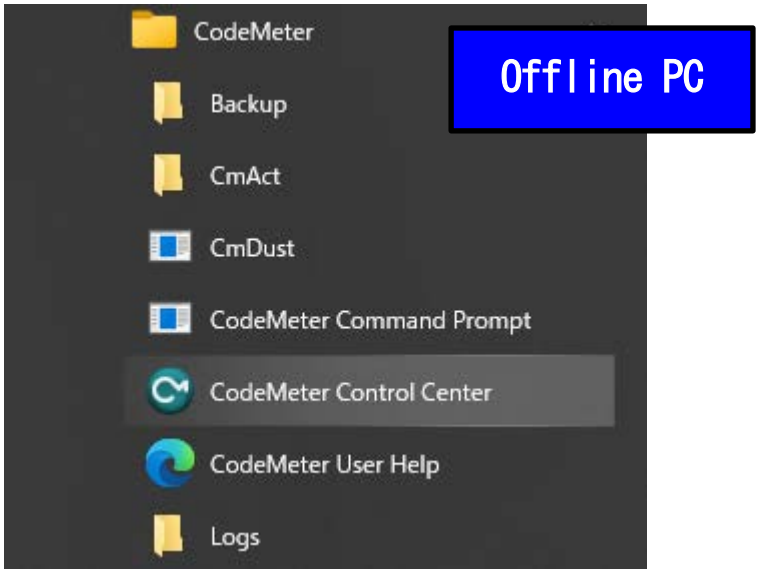
* The ticket ID is required to activate. Please complete 3.4.9 User Registration to acquire the ticket ID..

(1) Place the "License Request File Template" stored on the installation disc (Option¥SWkeyOfflineActivation) or in the installation directory of the DeviceXPlorer (Option¥SWkeyOfflineActivation¥) in the any location on the offline PC.

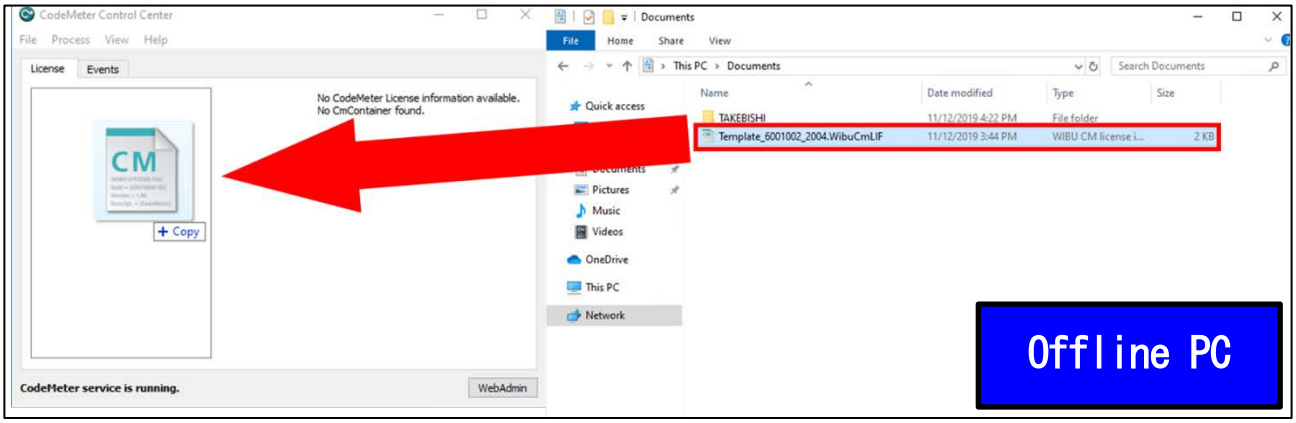
License request file template
"Template_6001002_2004.WibuCmLIF"



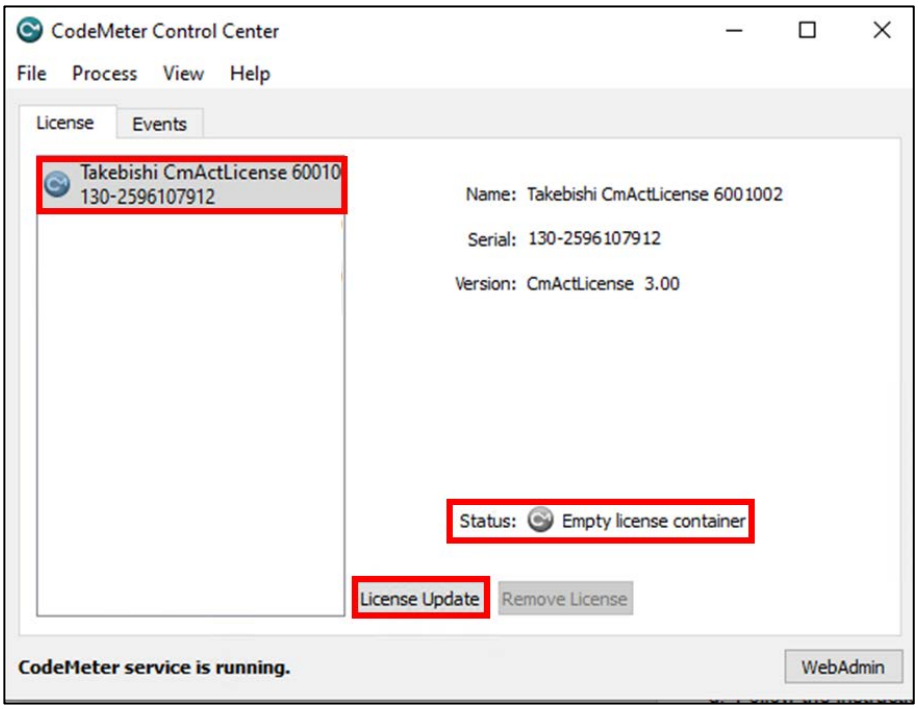
(2) On the offline PC, start the "CodeMeter Control Center" from the Start menu or the following path.
CodeMeter Control Center File path to store;
C:¥Program Files (x86)¥CodeMeter¥Runtime¥bin¥CodeMeterCC.exe



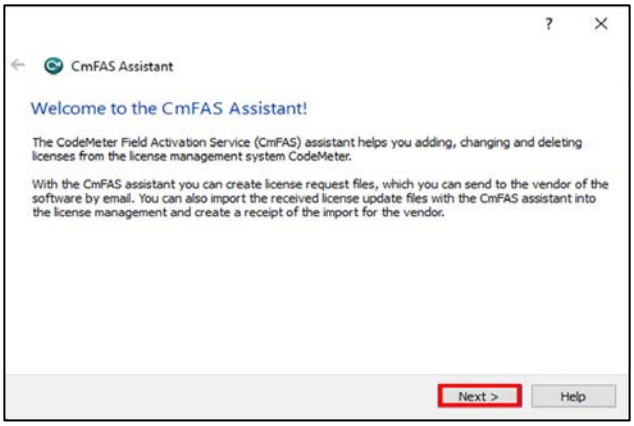
(3) Open CodeMeter Control Center, and Copy & Drop the “License Request File template" on “License” of CodeMeter Control Center.



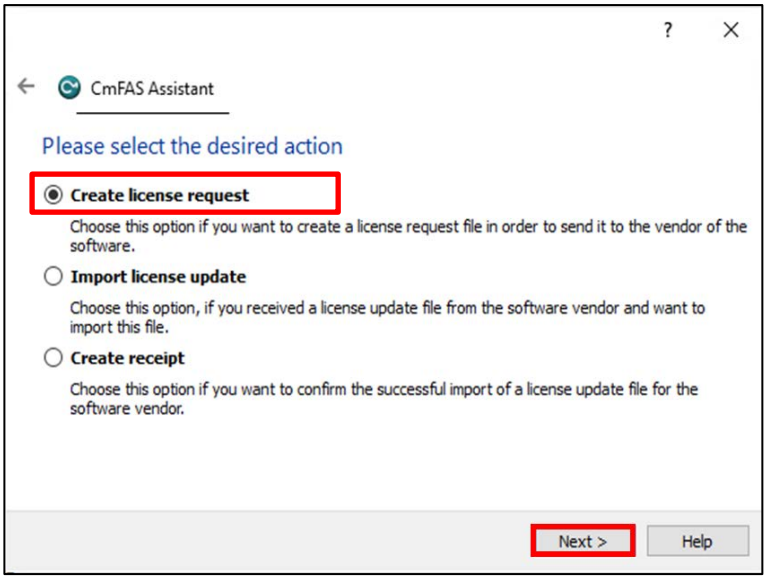
(4) After placing "License Request File template”, “Empty License Container” is displayed on “License” in CodeMeter Control Center. Select the added license container and click “Update License”.



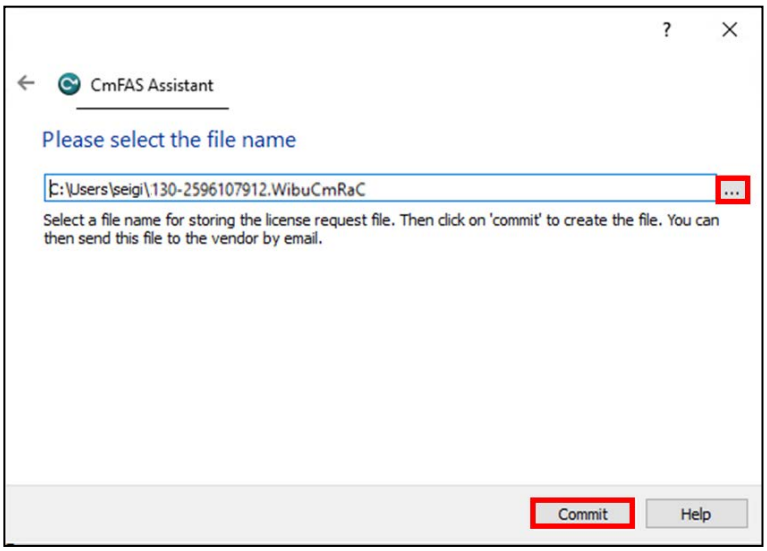
(5) Click the “Next.>” button.



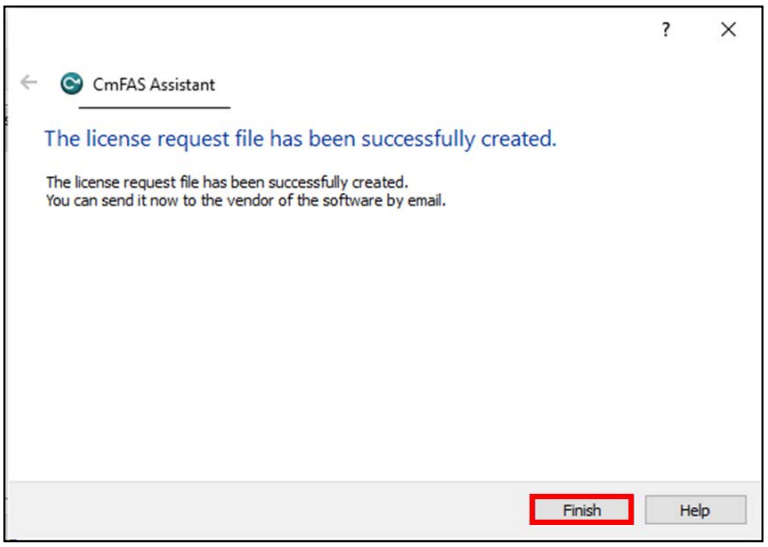
(6) Check “Create license request” and click “Next”.



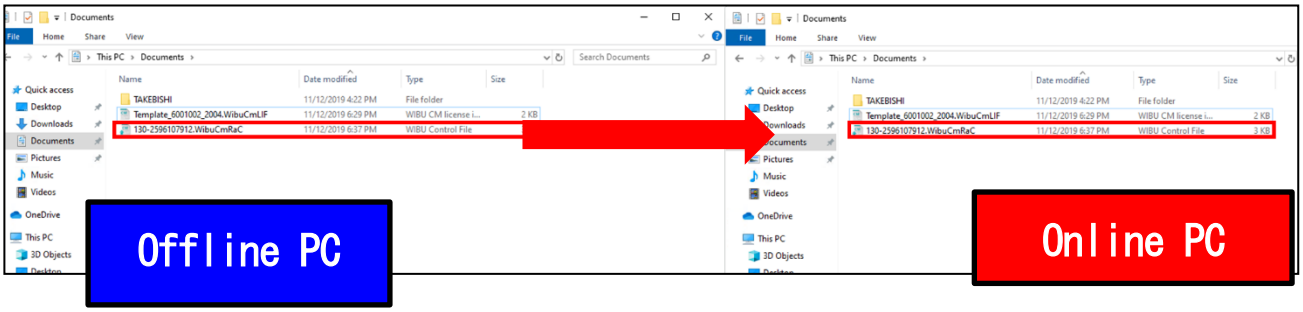
(7) Save the created “License Request File " in any folder, specify the save folder to save and click “Next”.



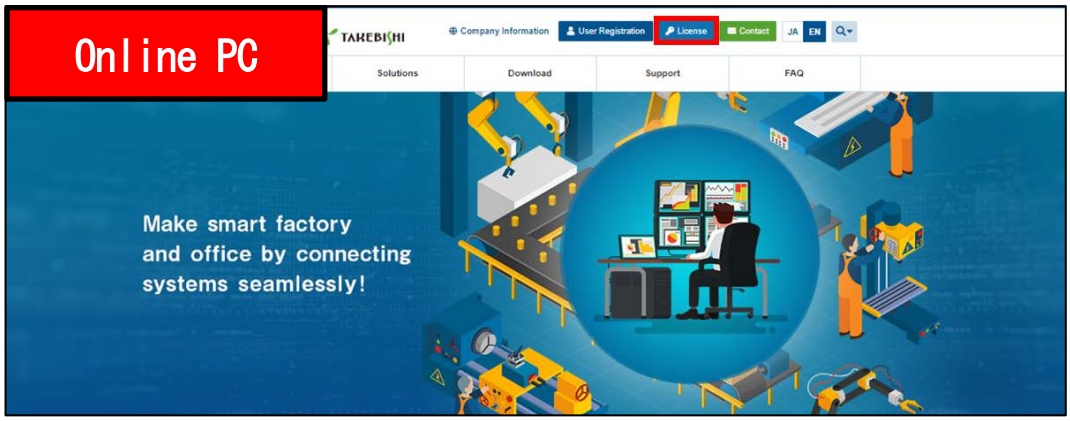
(8) The “" License Request File” is created, the following screen is displayed.
Click “Finish”. Generation of a “License Request File” based on the template is completed by the above Procedures.



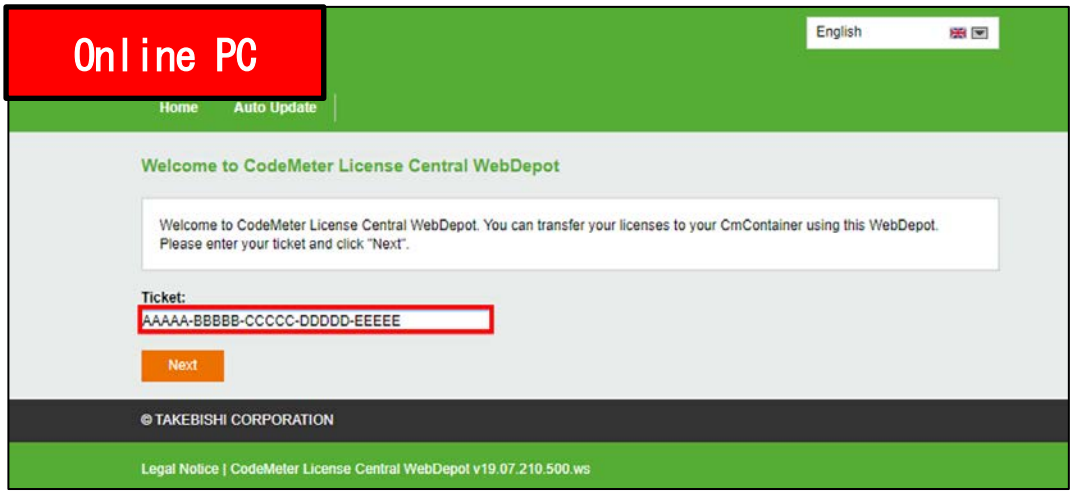
(9) Transfer the “License Request File” that created in Offline PC to any folder on the Online PC.



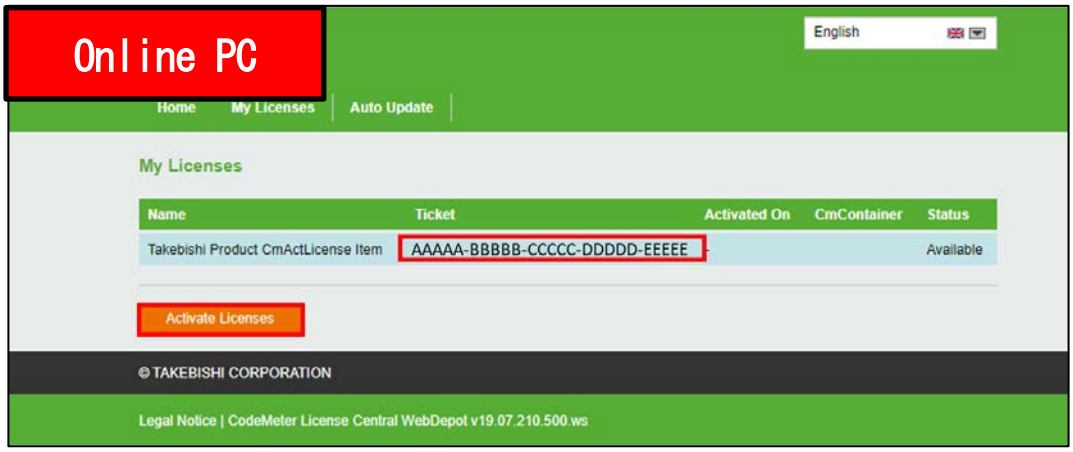
(10) Open the license registration web page("License") in FAWEB on the Online PC.
("License" in FAWEB : http://license.faweb.net/index.php)



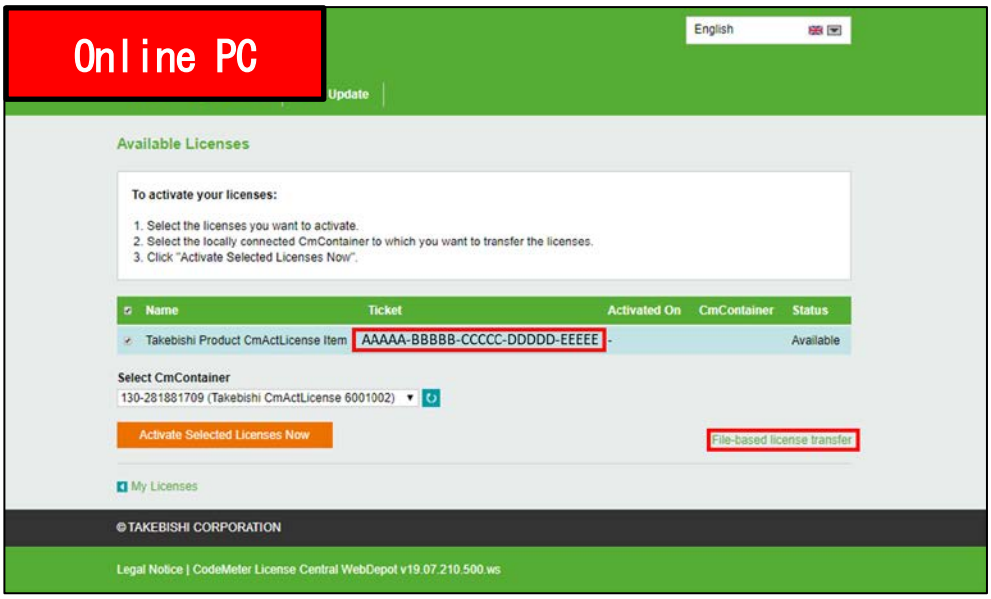
(11) Input the ticket ID in the "Ticket:", which you acquired at user registration on the license registration page, and click "Next.



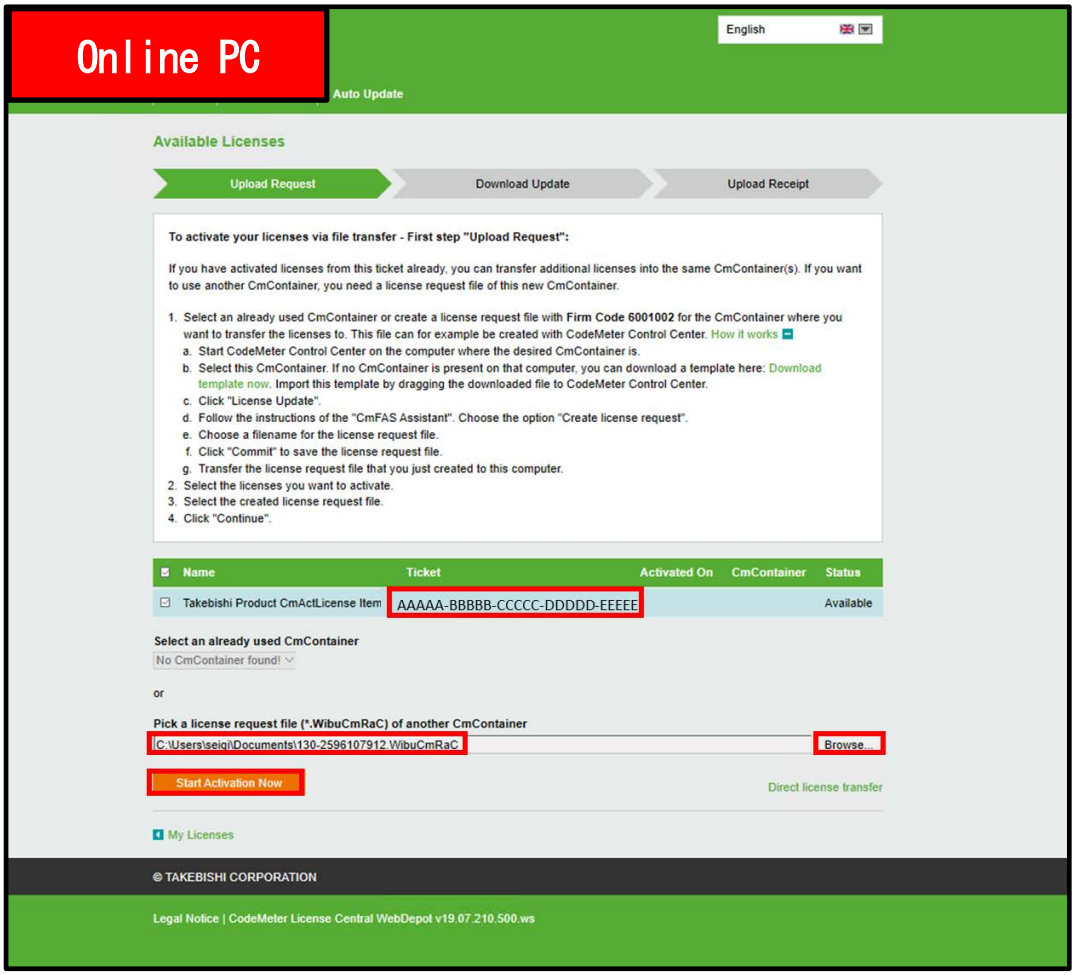
(12) My Licenses screen is displayed. Confirm that the ticket ID is correctly filled, and click "Activate Licenses".



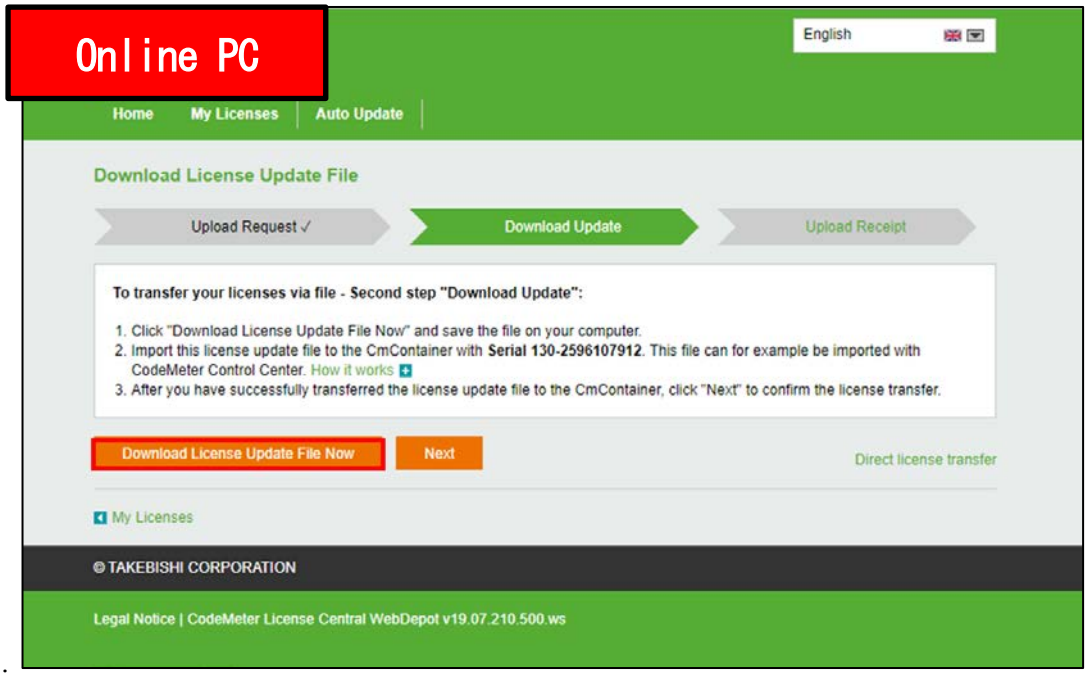
(13) Available Licenses screen is displayed. Click "Offline License Transfer". Click “File-based license transfer”.



(14) The following screen is displayed. Click on "Select File" and select the "License Request File" that transferred in procedure (9). Then click on "Start Activation Now".

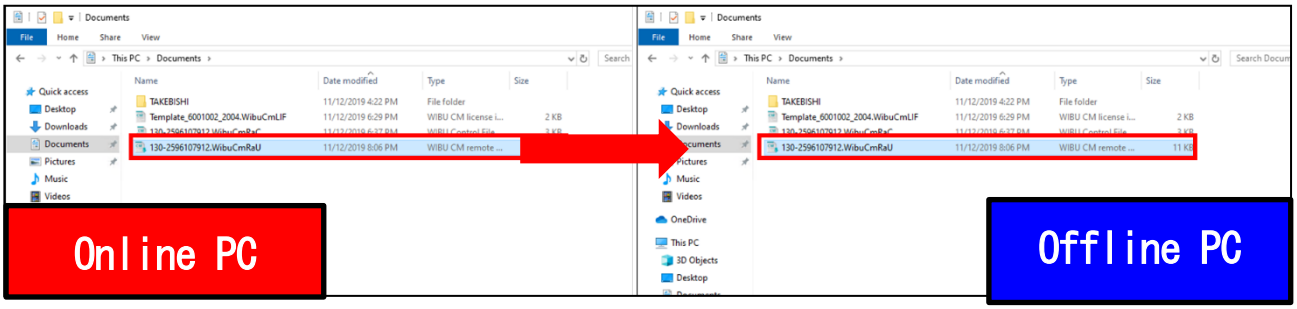


(15) Click “Download License Update File Now” to create the “License Update File ”.

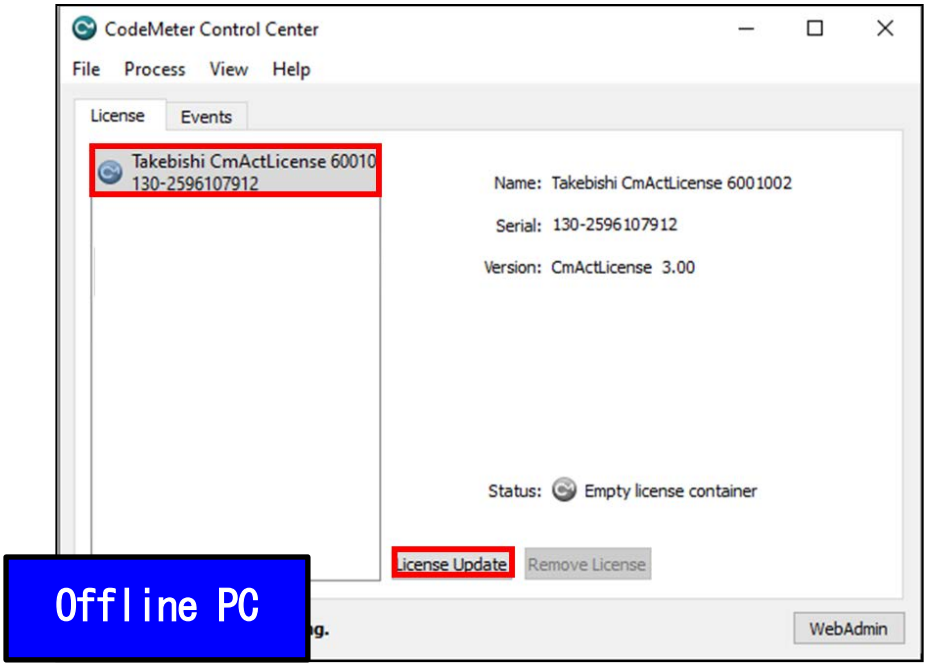


.

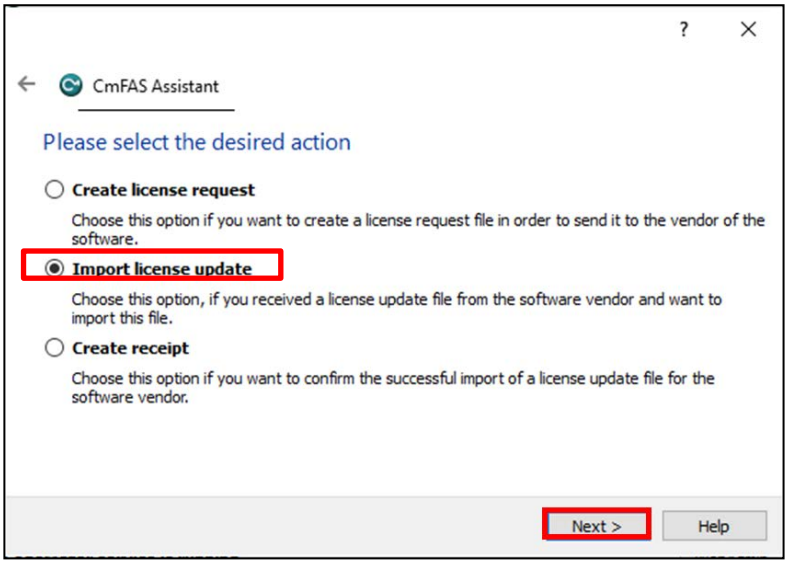
(16) Copy the "License Update File" to any location on offline PC.



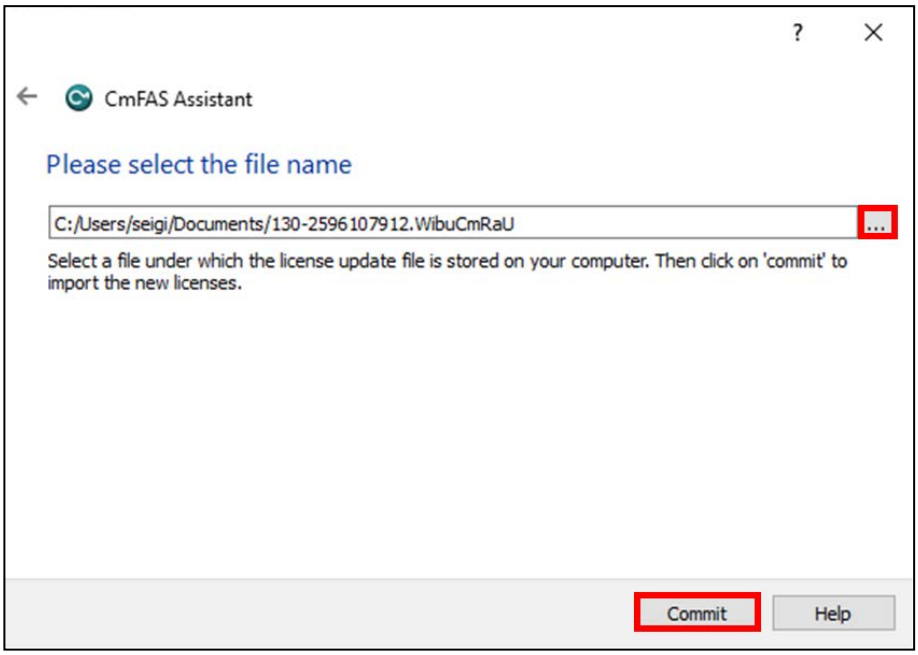
(17) Start CodeMeter Control Center on an Offline PC. Select the license that is "Empty license container" and Click "License Update".

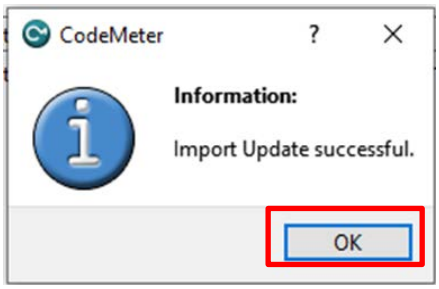


(18) Check “Import license update” and click “Next”.

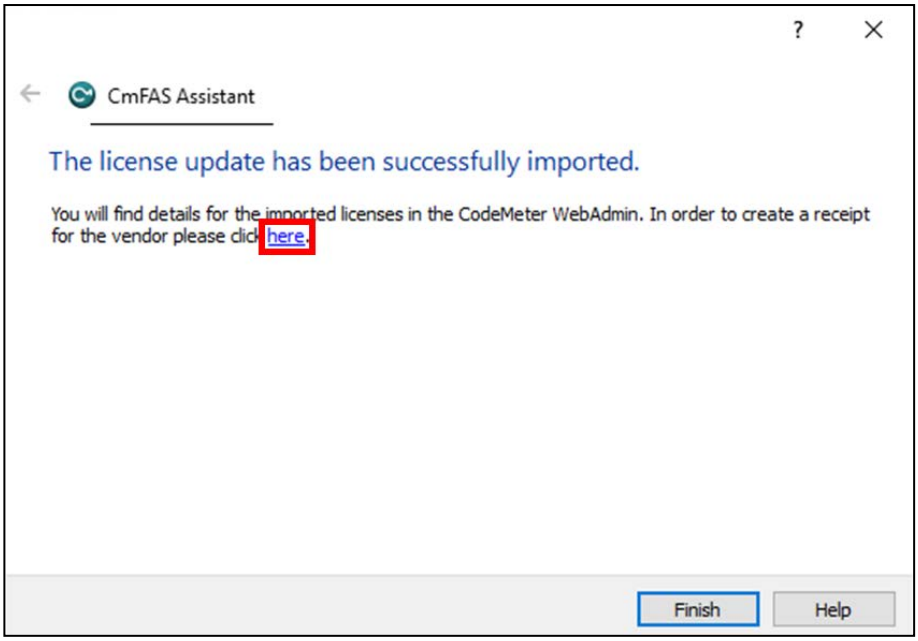


(19) Select “License Update File” saved in Offline PC and click “Commit”.





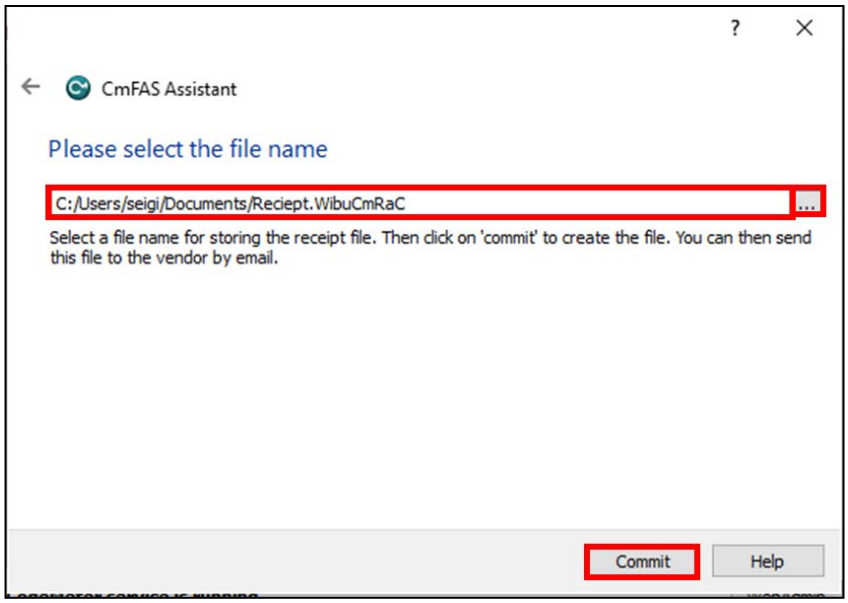
(20) To confirm that the license of the Offline PC has been updated, create a receipt file from “here” on this screen.



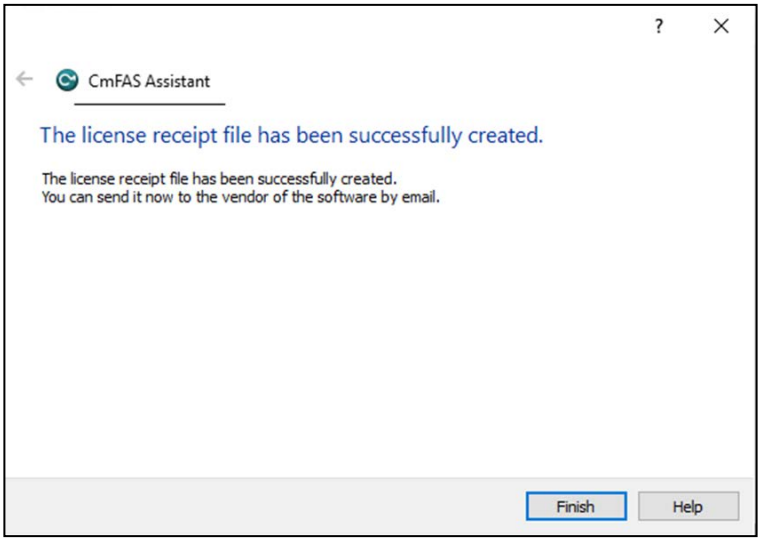
Important

When you activate the software key in an offline environment, you must create a receipt file and upload it to "License" page in FAWEB. Unless you upload the receipt file, you will not be able to perform some operations such as disabling the software key.

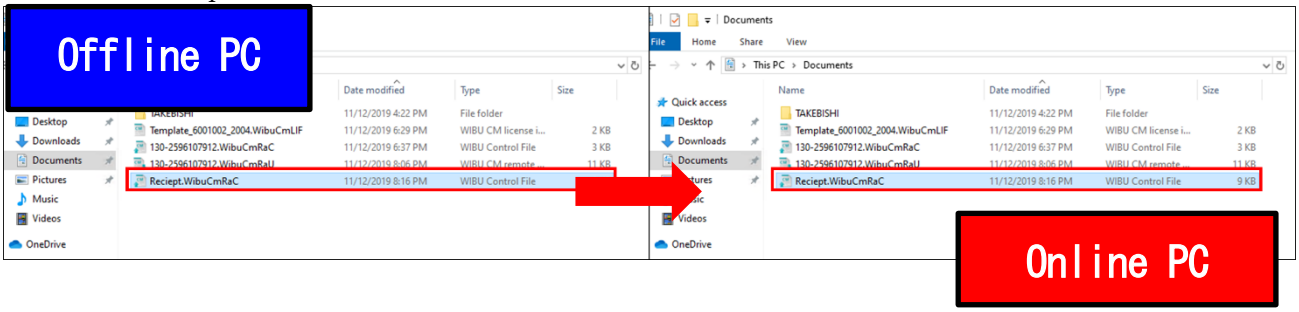
(21) Specify any folder and save the created receipt on the Offline PC and click “Commit”.
*The file is generated with the same name as the "online request file", change the file name if necessary.



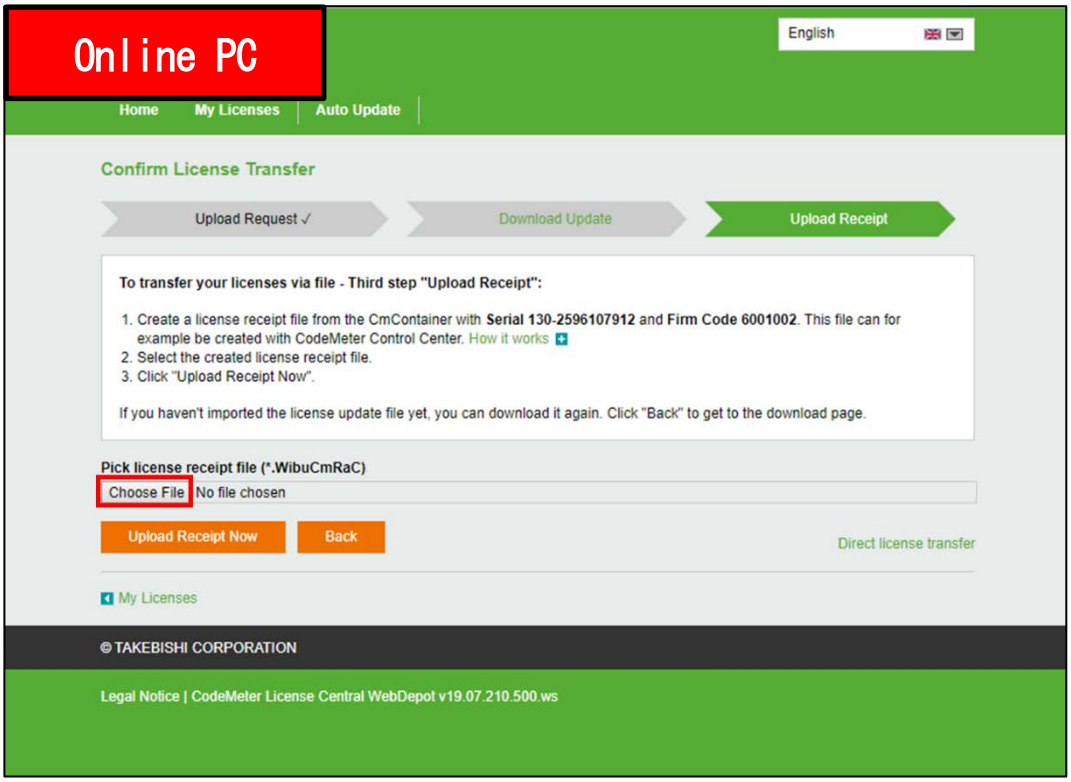
(22) When the created license file is saved, the following screen is displayed.



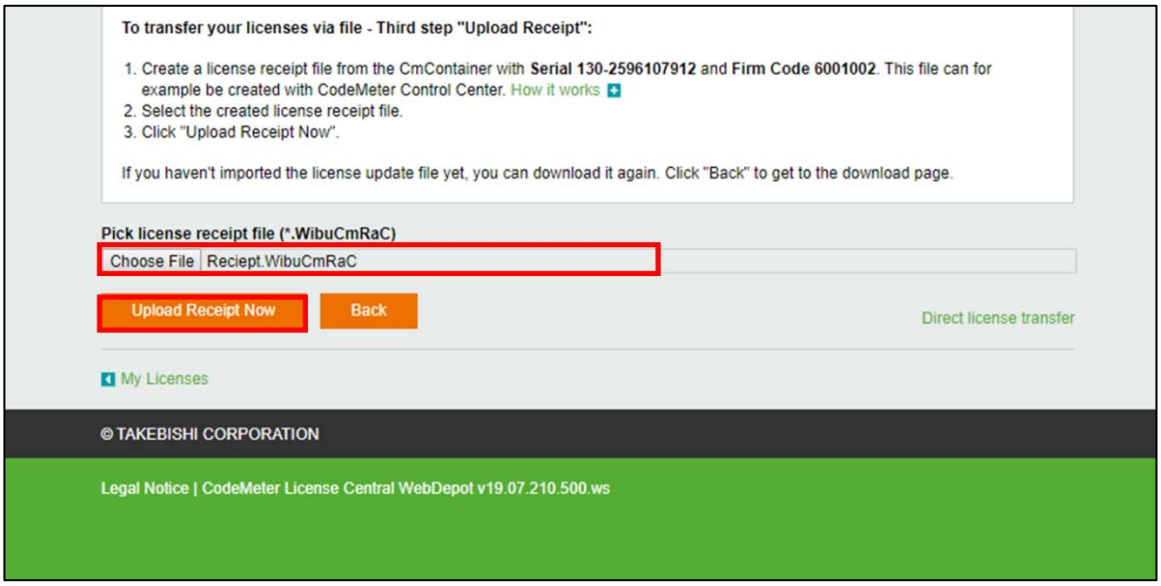
(23) Transfer the “receipt file” saved in Offline PC to on Online PC.



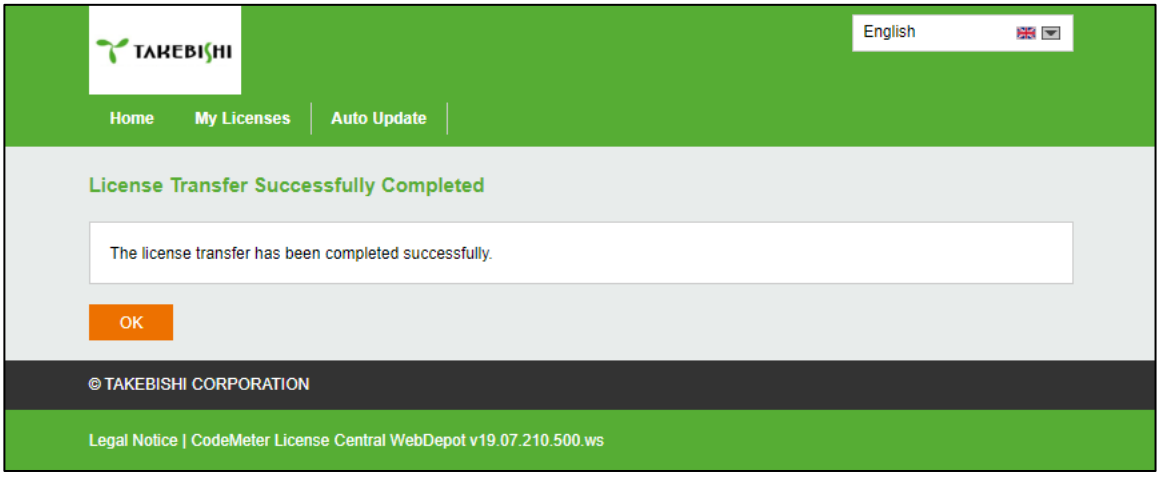
(24) Open the license registration web page on the Online PC and upload the receipt transferred from the Offline PC. Click "Select File" and select the "receipt file".



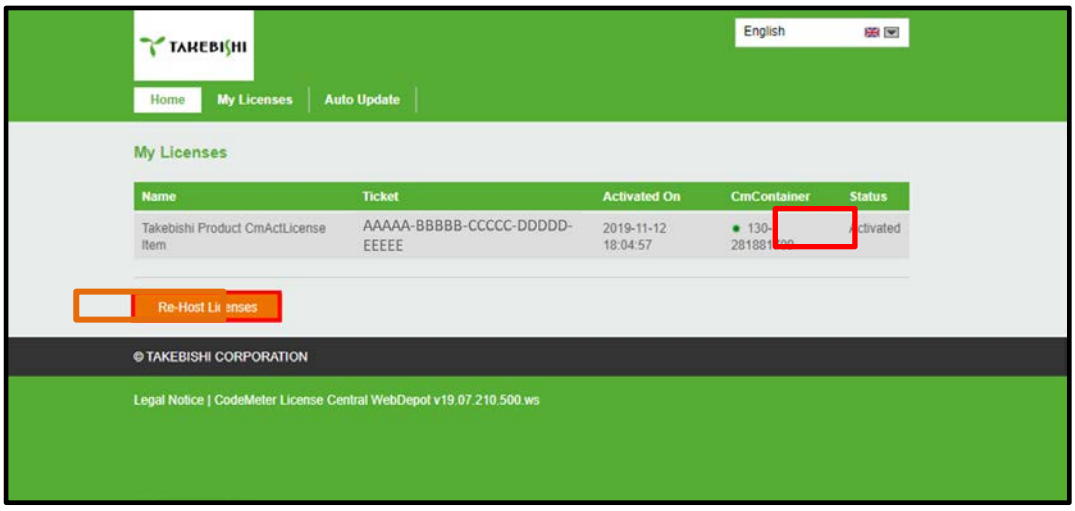
(25) After selecting “Receipt File”, click “Upload Receipt Now”.



(26) When uploading of "receipt file" is completed, the following screen is displayed.

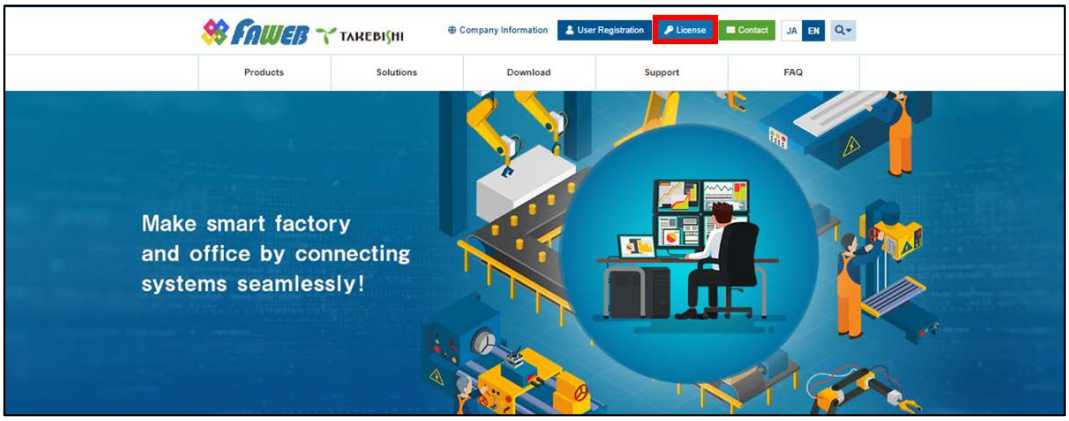


(27) When you click "OK", the following screen is displayed and Software Key Activation for the Offline PC is complete.

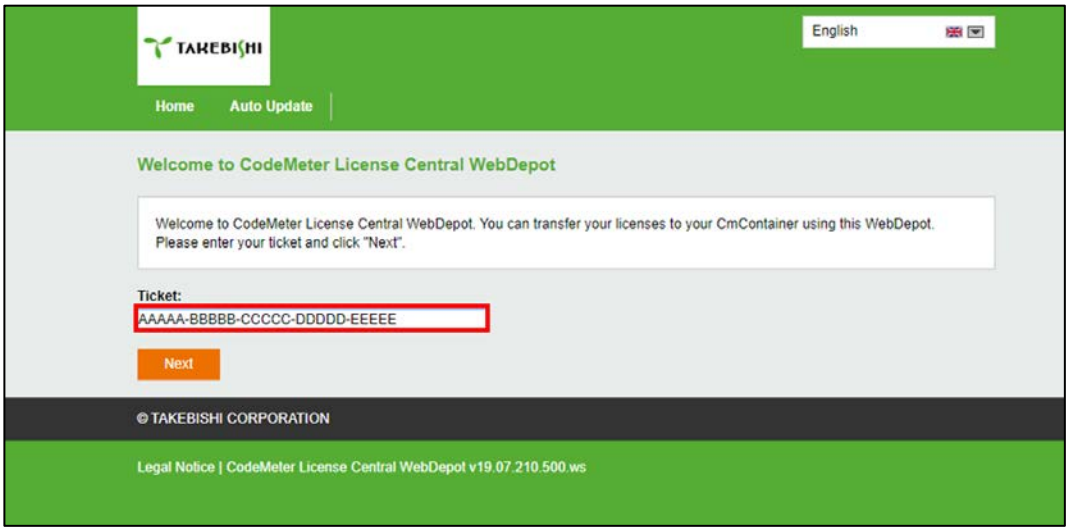


3.4.12 Deactivate the Software Key (Online)

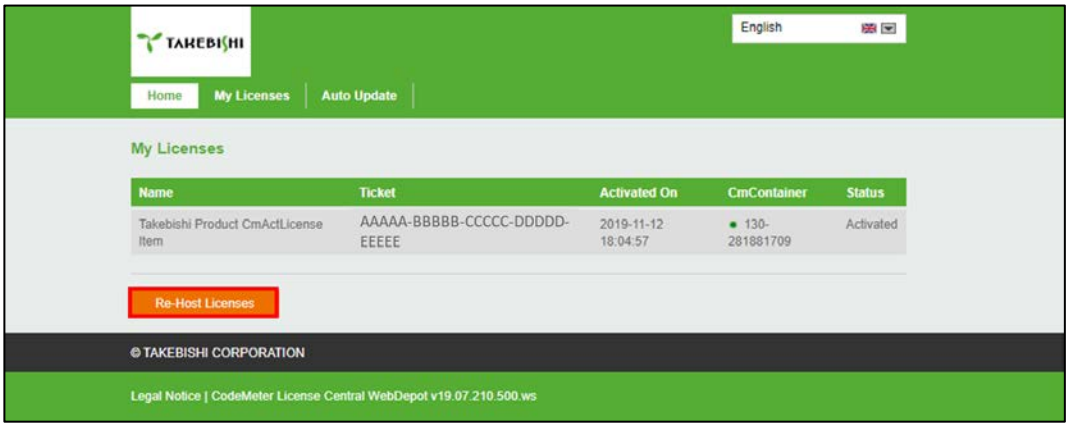
(1) Open the license registration web page (“License” http://license.faweb.net/index.php) in FAWEB.



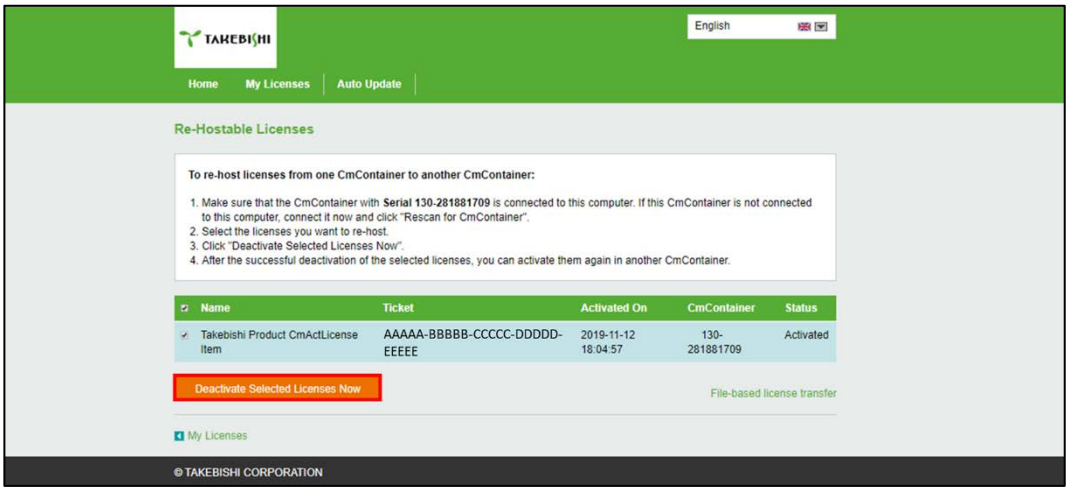
(2) Input the ticket ID in the "Ticket:", which you acquired at user registration on the license registration page, and click "Next. (Ticket ID linked to product serial number.)



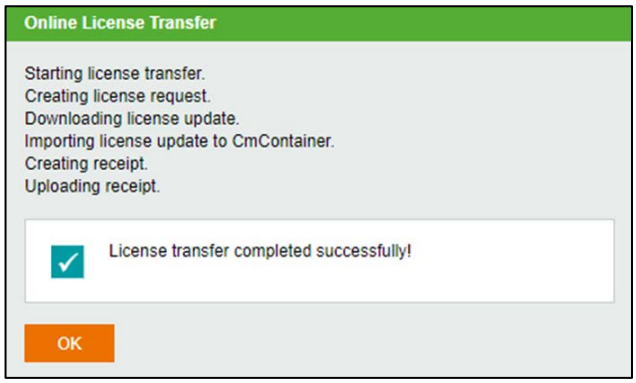
⑶ The screen will change to the My License Display screen. In order to proceed 'deactivate the software key', Click "Re-host Licenses".



⑷ "Re-Hostable Licenses" screen is displayed. Click "Deactivate Selected License Now".



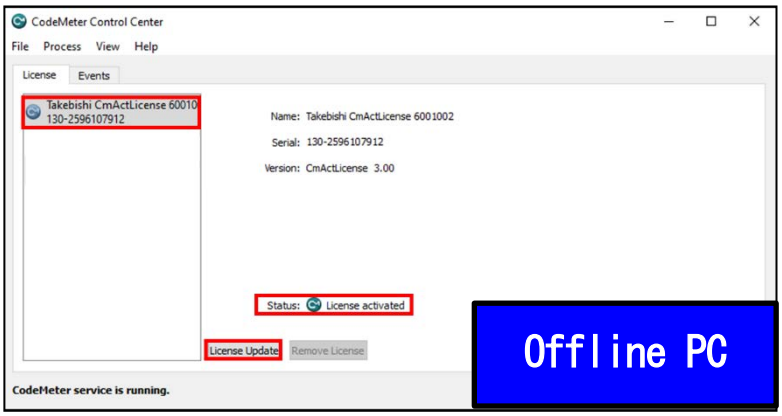
⑸ The following screen is displayed when process is completed..
When the message "License transfer completed successfully! " is displayed, Software Key Activation is complete.



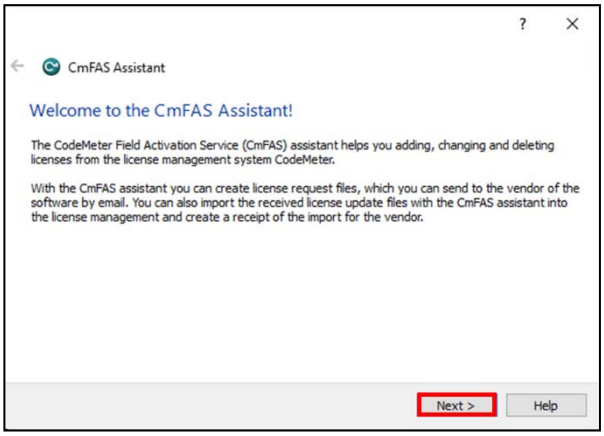
3.4.13 Deactivate the Software Key (Offline)

The following shows the procedure for deactivate the software key when the PC on which DeviceXPlorer is installed cannot connect to the Internet.
* Use a PC that can access the Internet (hereinafter referred to as Online PC) to transfer the software key of the PC (hereinafter referred to as Offline PC) where DeviceXPlorer that cannot access the Internet is installed.

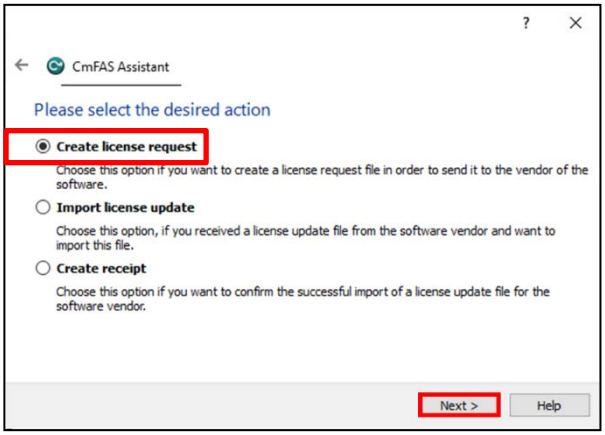
(1) Open "CodeMeter Control Center" on the Offline PC.
Select the license you want to transfer, and click “License Update”.



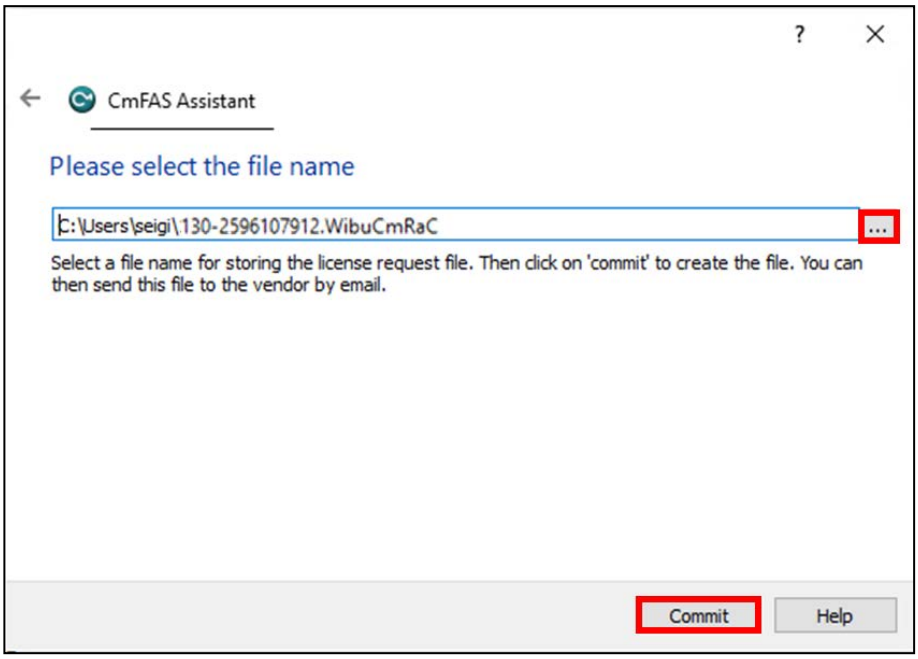
(2) Click the “Next”.



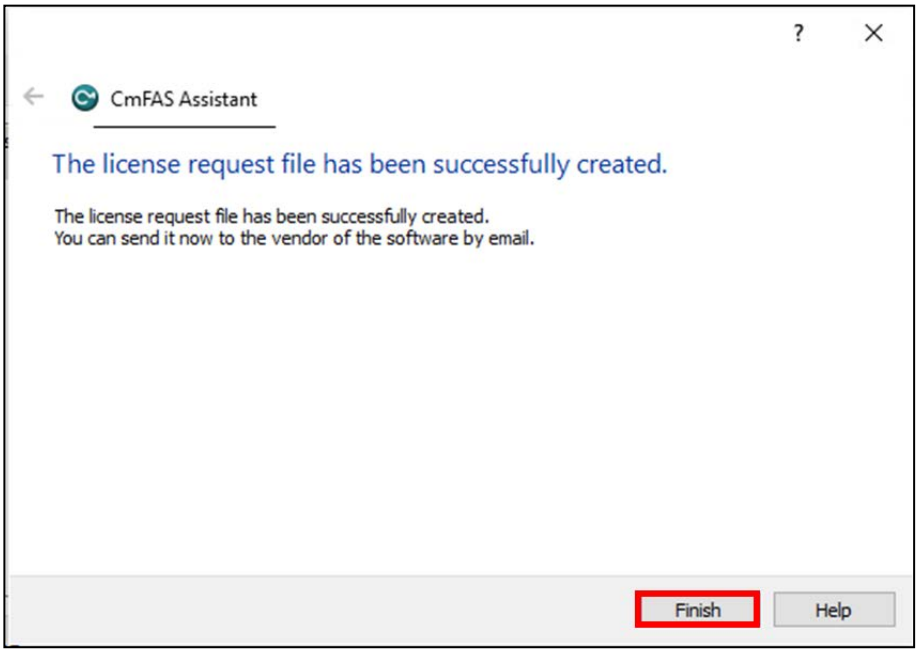
(3) Check “Create license request” and click “Next”.



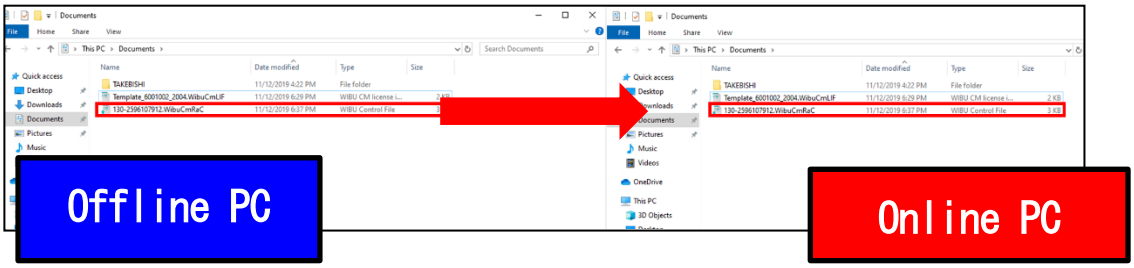
(4) Save the created “License Request File” in any folder, click “Commit”.



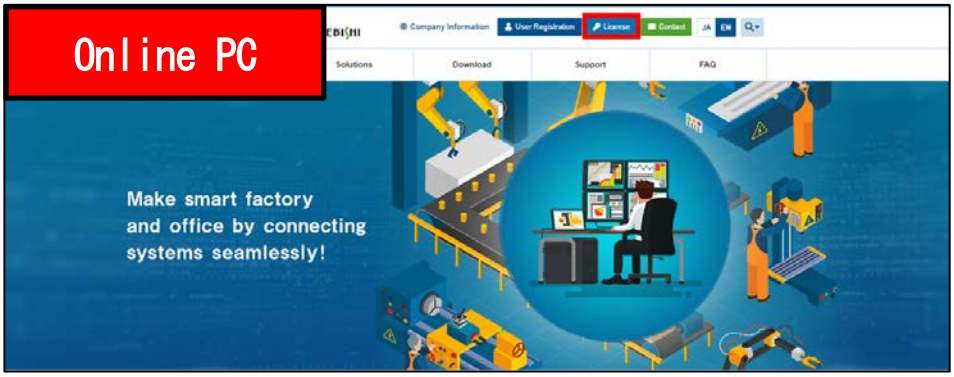
(5) The “License Request File” is created, the following screen is displayed.
Click “Finish”.



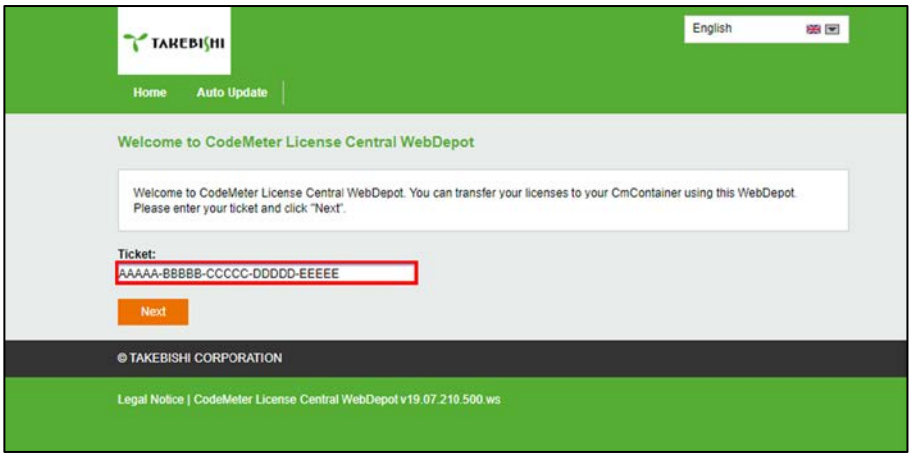
(6) Copy the “License Request File” from the folder of the Offline PC to the Online PC.



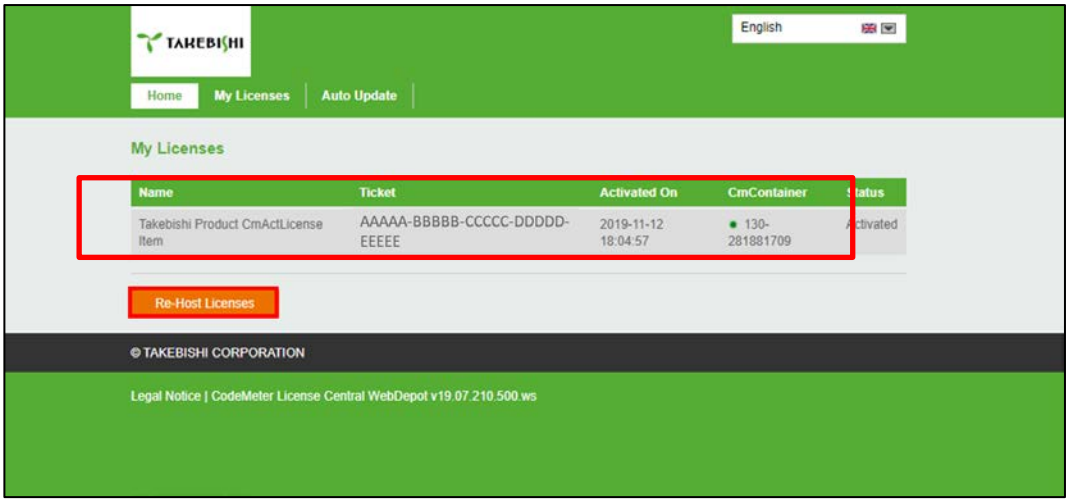
(7) Open the license registration web page (“License”) in FAWEB.
(The license registration web page: http://license.faweb.net/index.php)



(8) Input the ticket ID got at the time of user registration, and click “Next”.
(Ticket ID linked to product serial number is issued.)



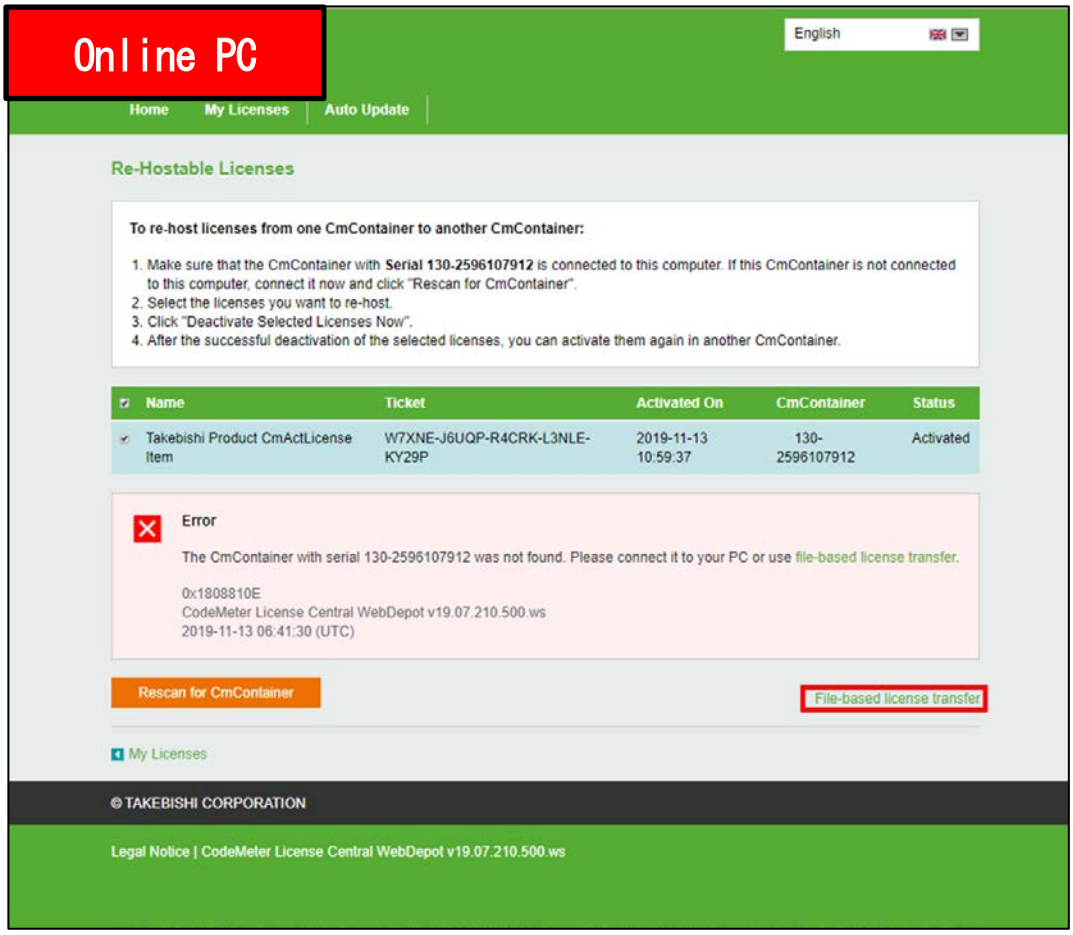
(9) My Licenses screen is displayed. (The license installed on the offline PC is displayed.)
Click on "Re-Host license" to proceed the procedure of deactivating the software key.



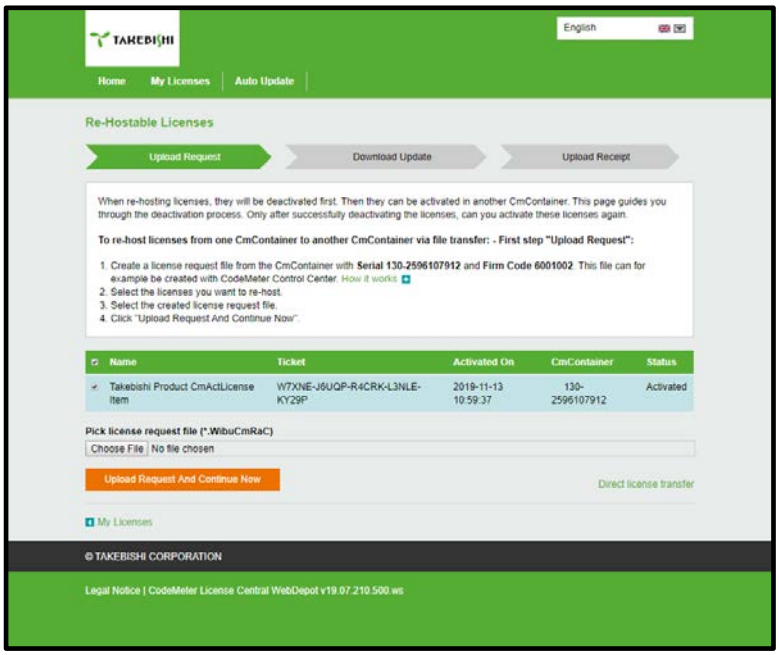
(10) The following screen is displayed. Click "File-based license transfer".

*An error message may displayed, which is no problem.

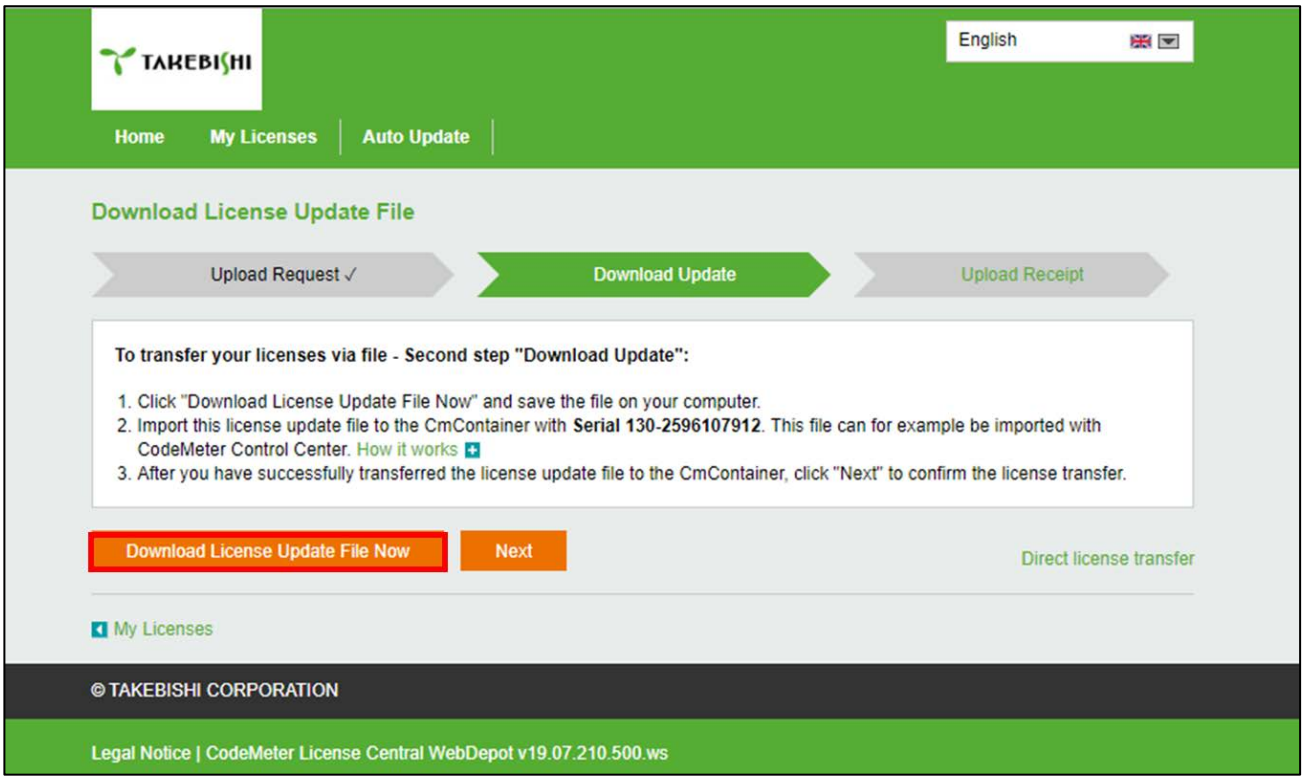
This is because the selected license does not exist on the Online PC.



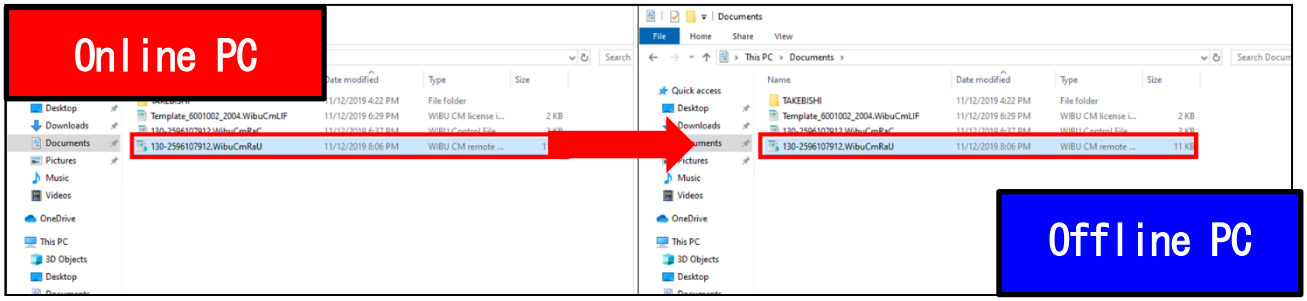
(11) When the License Transfer screen is displayed, click 'Choose File' and select the 'License Request File' that you copied from your offline PC. Click on 'Upload Request and Continue Now'.



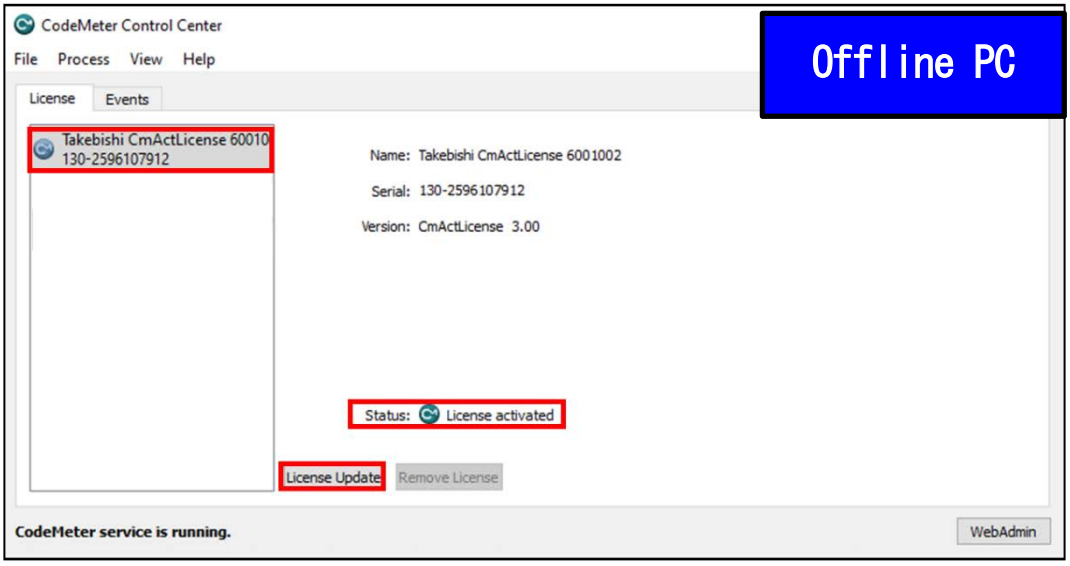
(12) Create the “license update file”on the following Procedures.
Click “Download License Update File Now”.



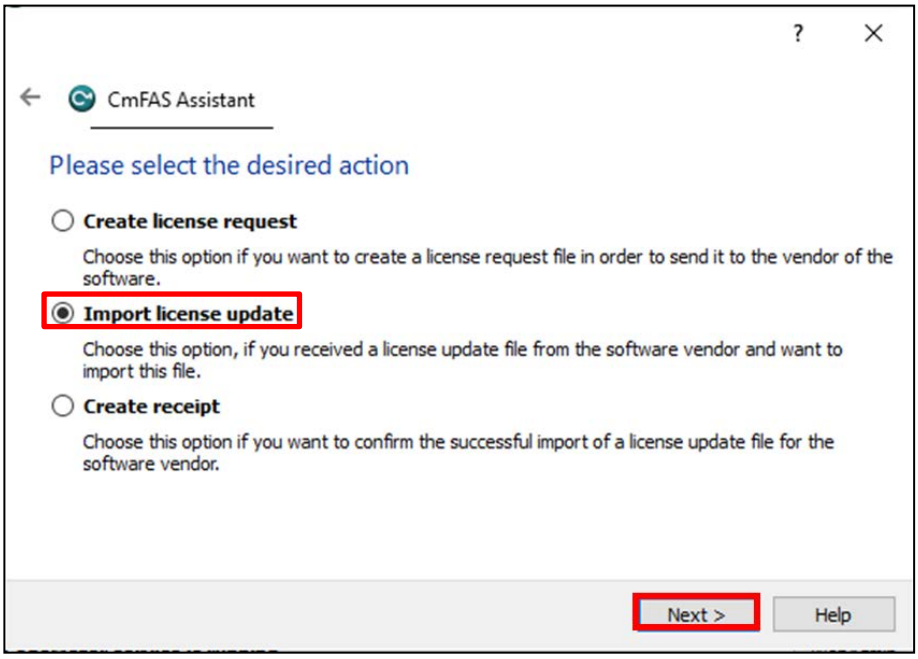
(13) Copy the "License Update File" in any folder on the Offline PC.



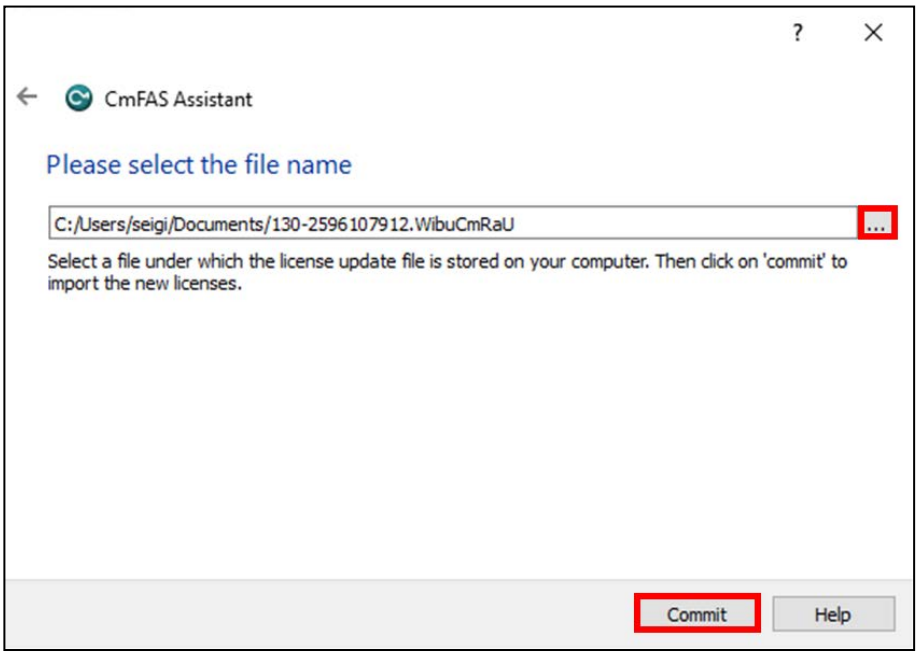
(14) Start the CodeMeter Control Center on your offline PC. Select the license to deactivate and click “License Update”.



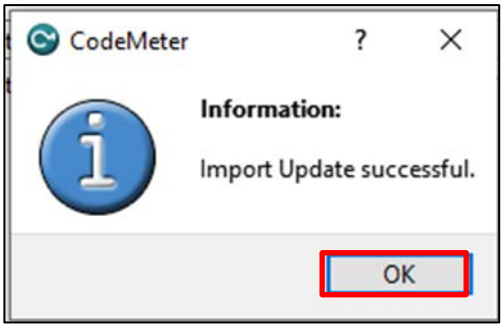
(15) When the following screen is displayed, select “Import license update” and click “Next”.



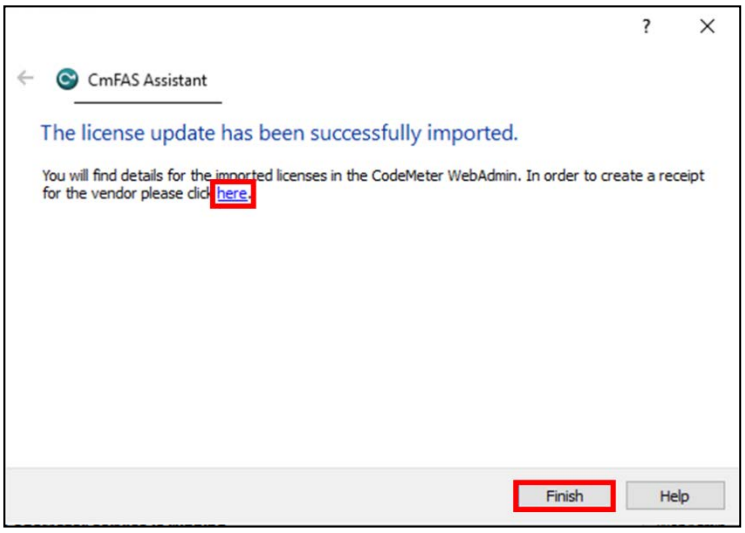
(16) Select “License Update File” saved in any folder and click “Commit”.



(17) When “License Update File” is imported, the following screen is displayed.



(18) To confirm that the license of the Offline PC has been updated, create a receipt file from “here” on this screen.

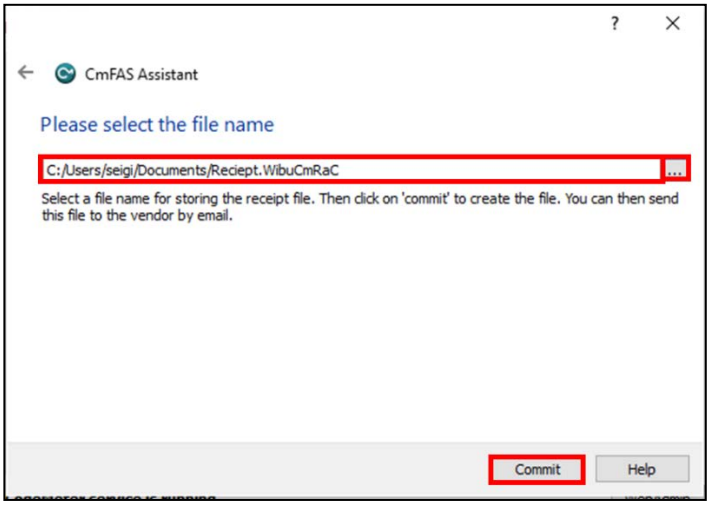


Important

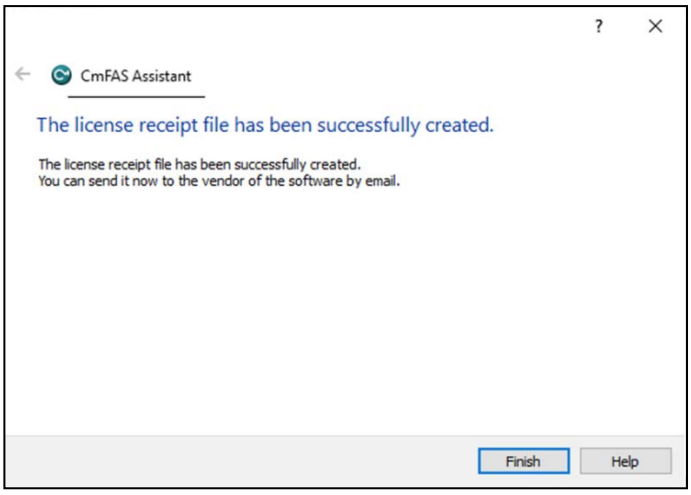
When you deactivate the software key in an offline environment, you must create a receipt file and upload it to "License" page in FAWEB. Unless you upload the receipt file, you will not be able to perform operations such as activating the software key.

(19) Specify where to save of the created receipt file in any folder on the Offline PC and click “OK”.

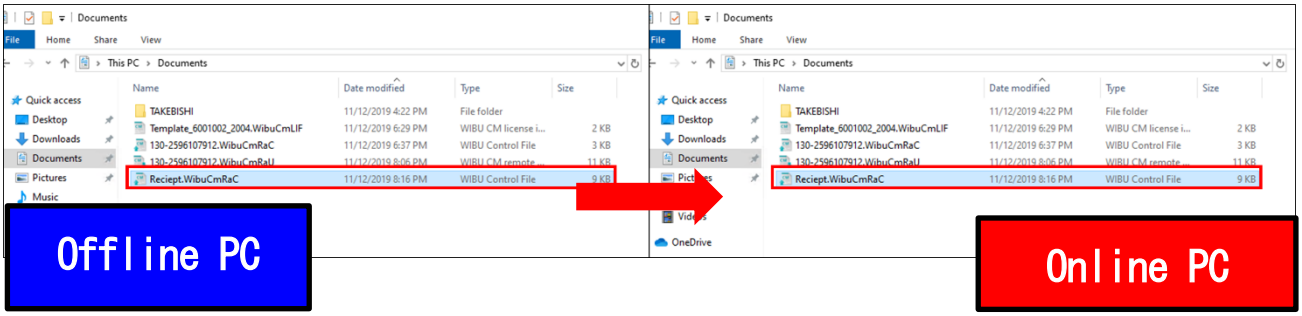
* The created “receipt file” is automatically named same as the “online request file”. Change the file name if necessary.



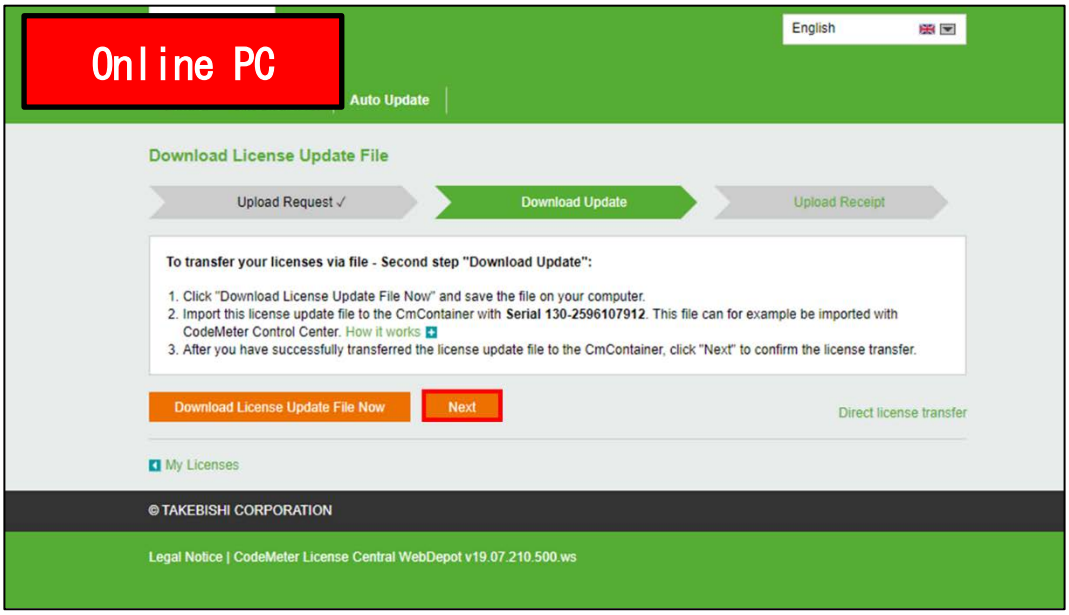
(20) When the created license file is saved, the following screen is displayed.



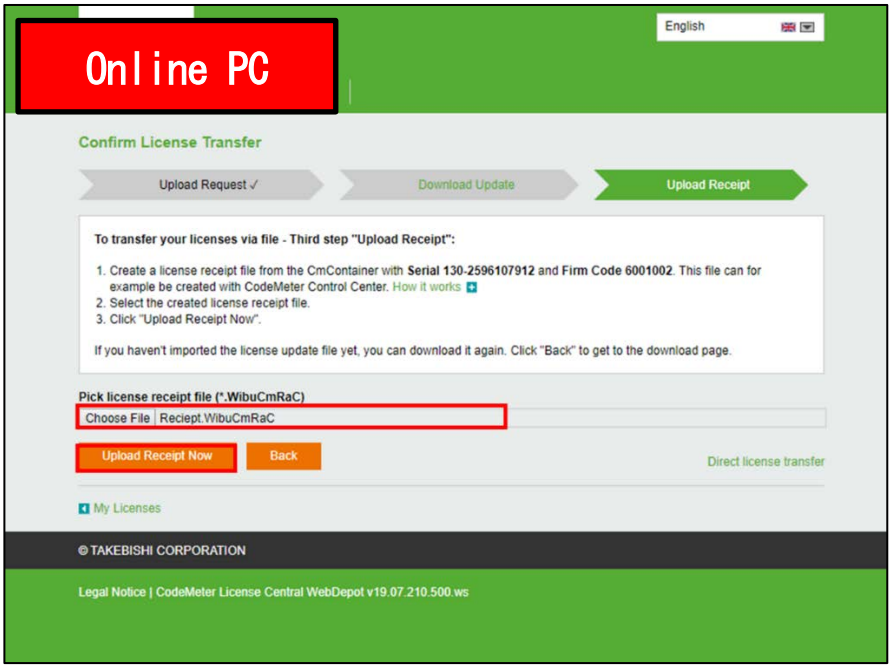
(21) Copy the "receipt file" saved in Offline PC to on Online PC.



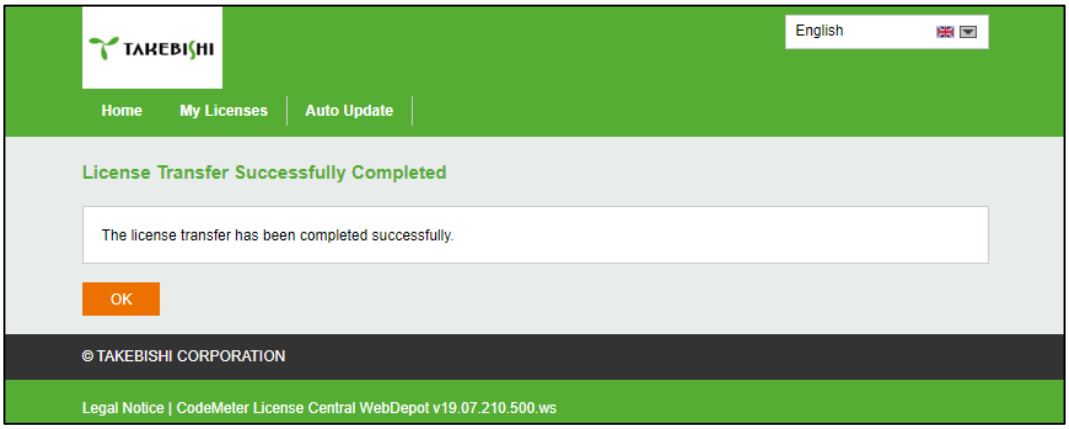
(22) Open the license registration web page on the Online PC, click "Next".



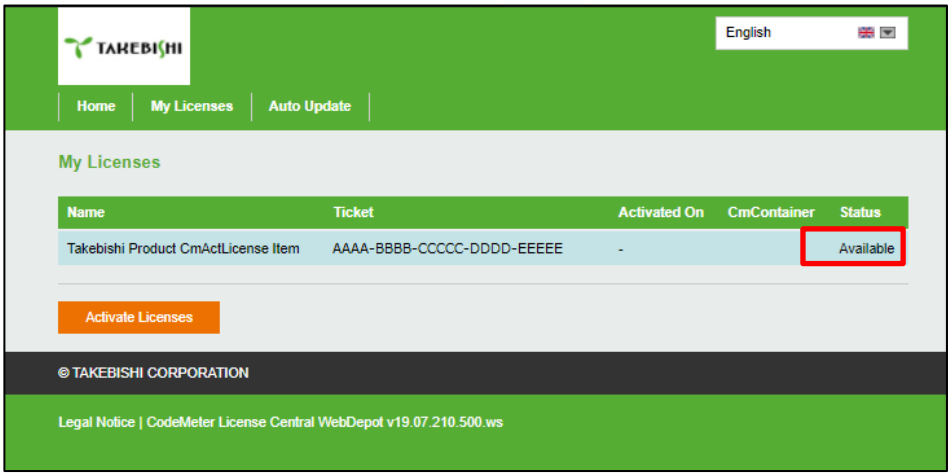
(23) Upload the receipt file copied from the Offline PC. Click on "Choose File" and specify where to store the receipt file. After specifying the receipt file, click "Upload Receipt Now".



(24) When “receipt file” is uploaded, the following screen is displayed to complete the deactivation of the software key.



When deactivation of the DeviceXPlorer on the Offline PC is completed, the status will be displayed as “Available”.



3.4.14 Using the Software Key in the Virtual Environment

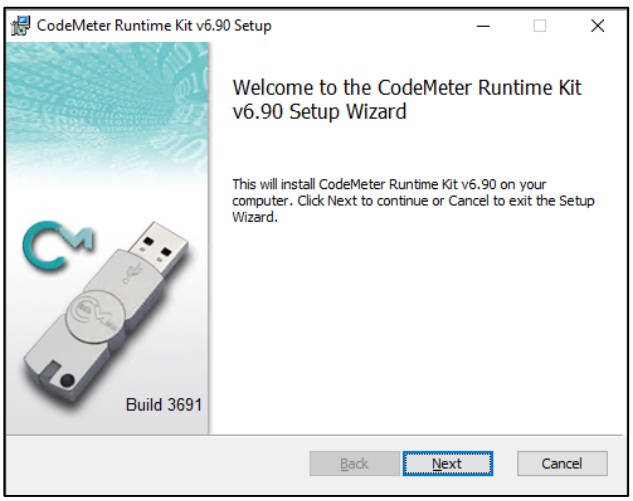
This section describes the procedure for using a hardware key license on a DeviceXPlorer in a virtual environment. Software key can be activate directly on the guest OS. However, the software key may not be activated correctly due to changes in the operating environment. Therefore, it is recommended to activate the software key on the host OS and use the license key from the guest OS.
Please follow the steps below to set up the host OS environment.

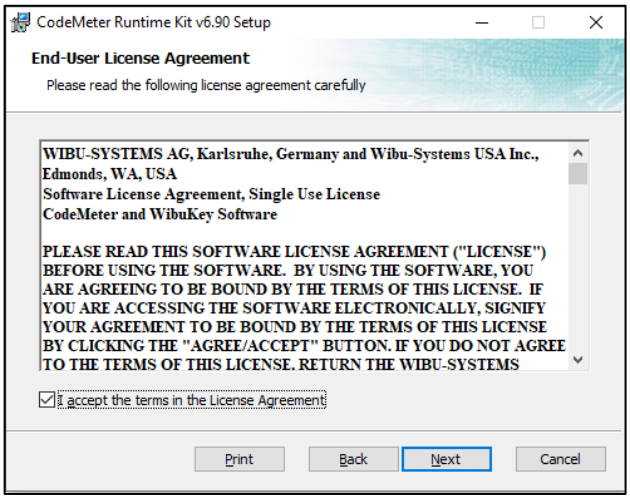
3.4.14.1 Set up CodeMeter Runtime on Host OS

In an environment where DeviceXPlorer is not installed, CodeMeter Runtime is required to be installed to use the license key.

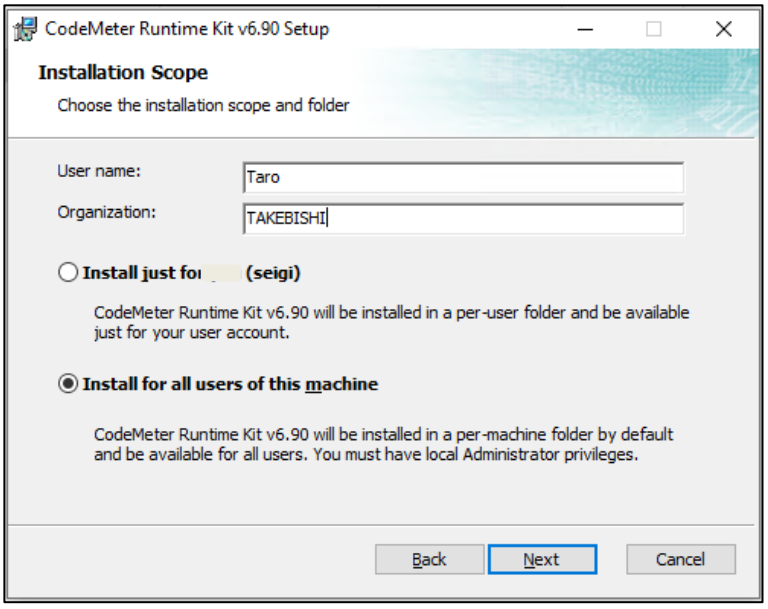
(1) Download the "CodeMeter Runtime setup file From the FAWEB download page (https://www.faweb.net/en/download/), and install it.

(2) When the setup screen starts, click "Next", check "I accept the license agreement", and click "Next".

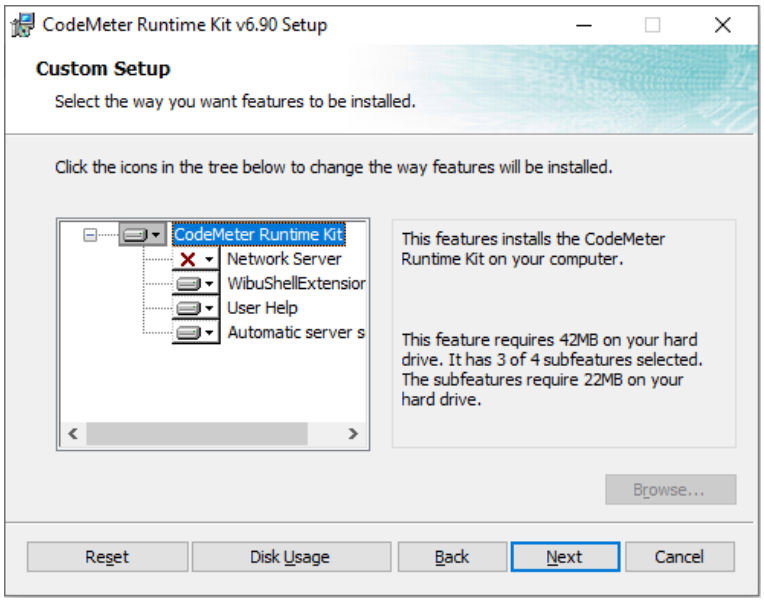




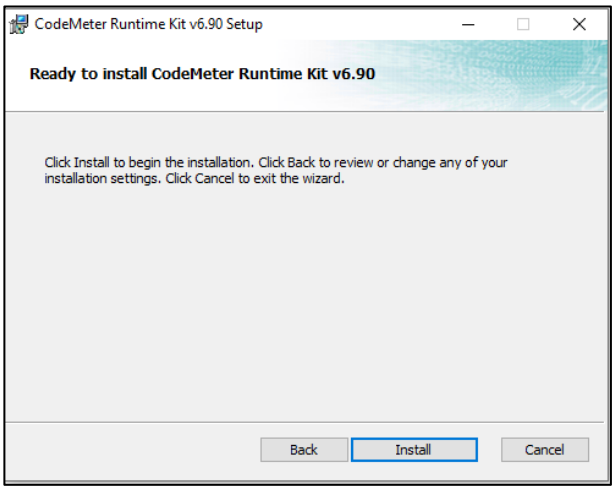
(3) Input the user information, and click “Next”.

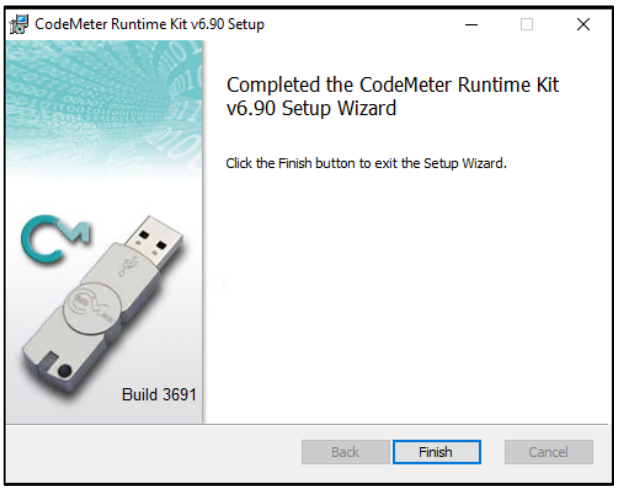


(4) Set the function to be set up as shown below, and click "Next".

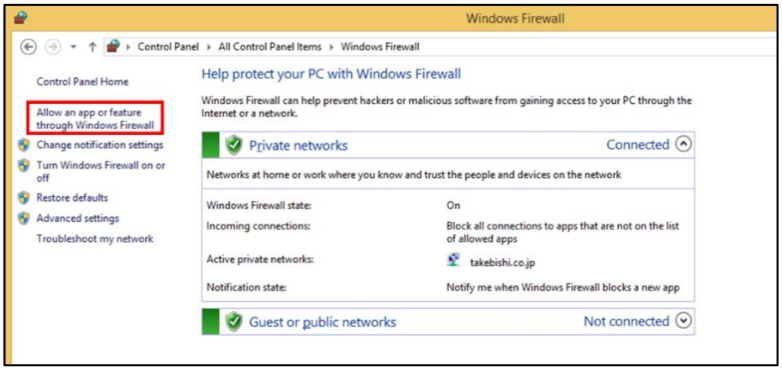


(5) When the following screen displayed, click "Install" to start the installation.

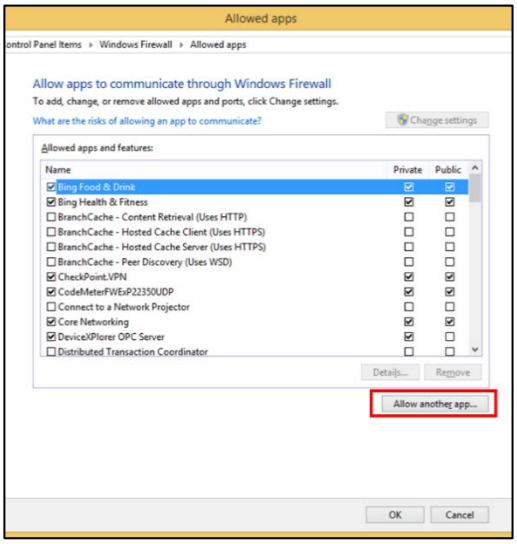


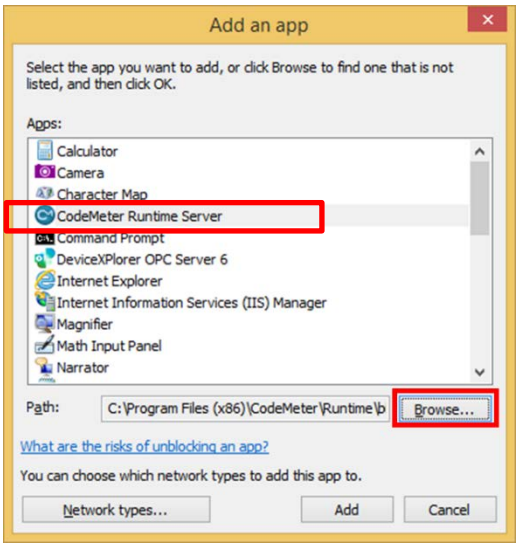


(6) The following describes the procedure to allow communication through CodeMeter's firewall in the firewall settings of the host OS. Open the “Allow an app or feature through Windows Firewall”. (Control Panel > System and Security > Windows Firewall)

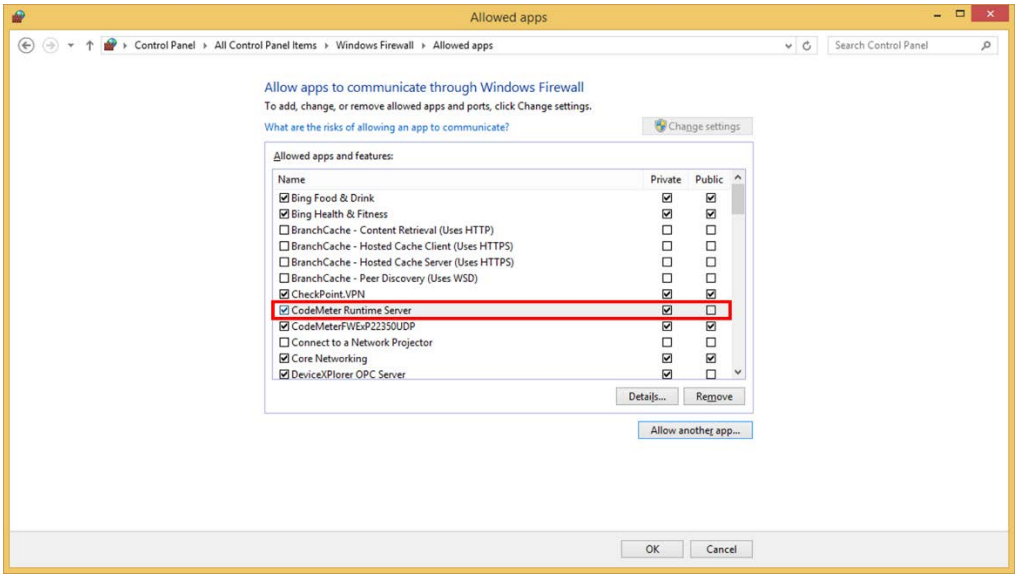


(7) Click the “Allow another app...” and add ”CodeMeterRuntimeServer” from “Browse..”. Allow communication through CodeMeter's firewall. Path: C:¥Program Files (x86)¥CodeMeter¥Runtime¥bin¥CodeMeter.exe





(8) Set up private and public according to the network environment.

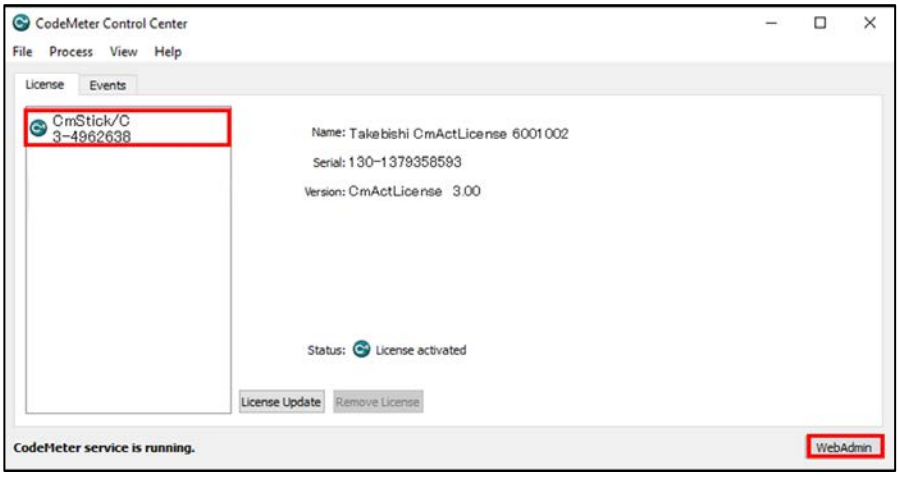


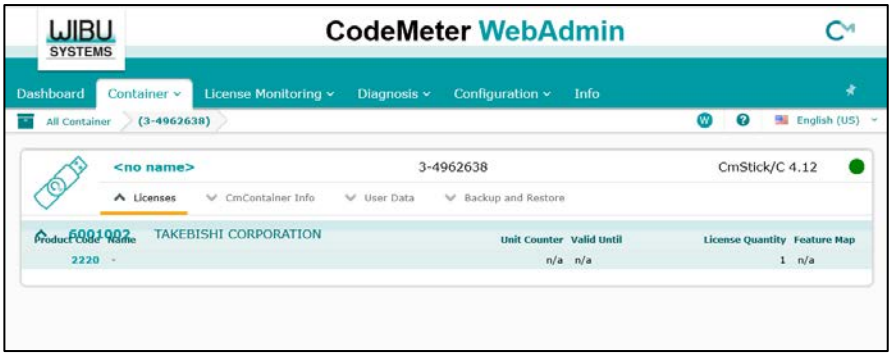
3.4.14.2 License activation on the host OS

Hardware key: Attach the hardware key to the host OS.
Software key: Activate the software key on the host OS, Please refer to Software key activation procedure.

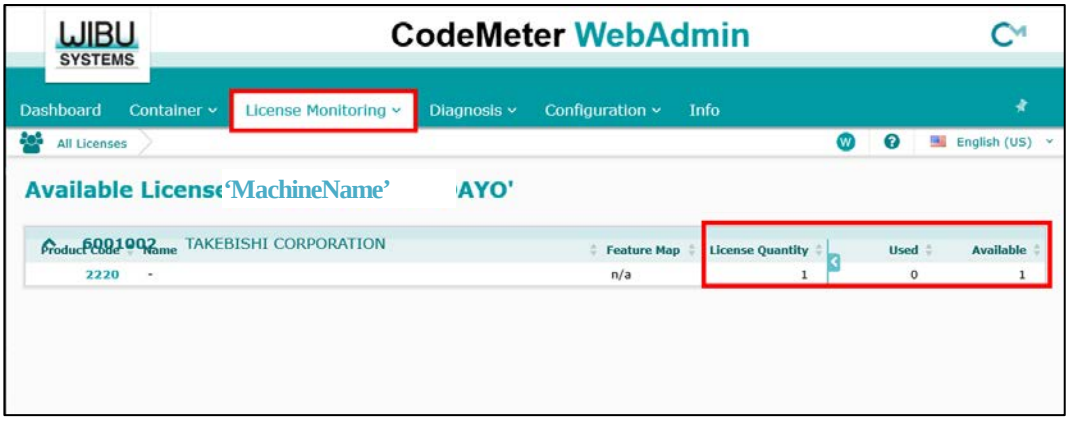
3.4.14.3 License server activation on the host OS

(1) Open "CodeMeter Control Center" from the Startup menu. Make sure CmStick is listed for hardware key, or "Takebishi CmActLincense" is listed for software key. Then click "WebAdmin" to open the "CodeMeter WebAdmin" in the browser.

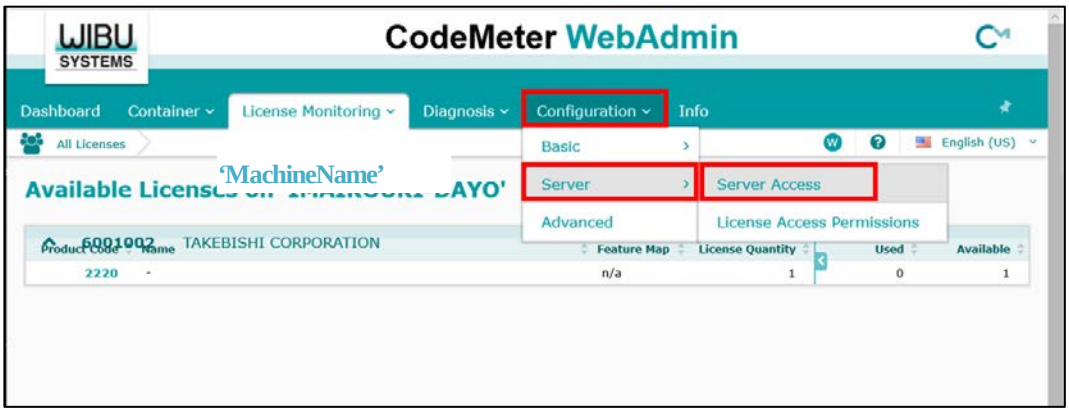




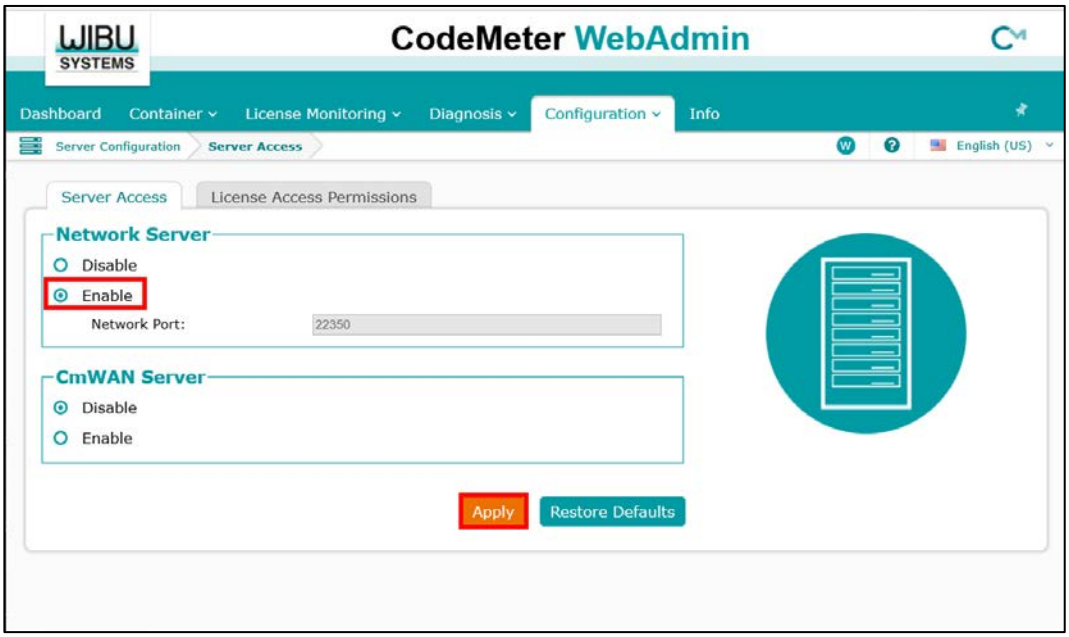
(2) Click “License Monitoring” on “CodeMeter WebAdmin” and confirm that “6001002-2220” license is available.



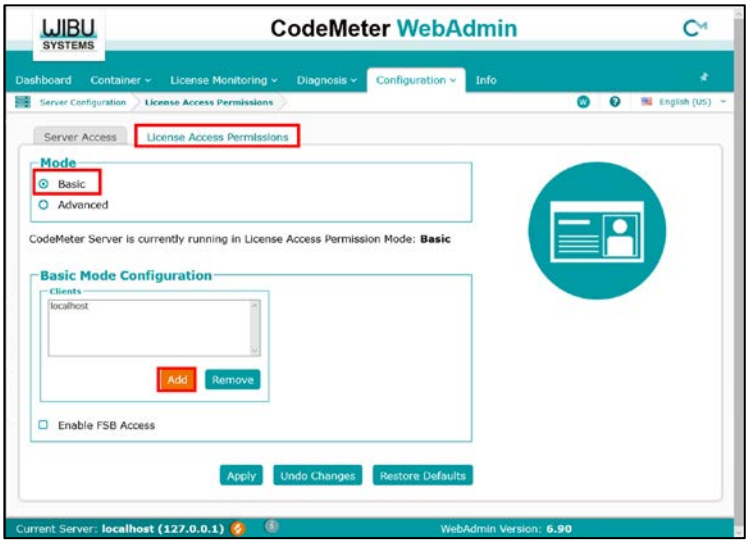
(3) Move the cursor to the setting and click “Server” > “Server Access”.



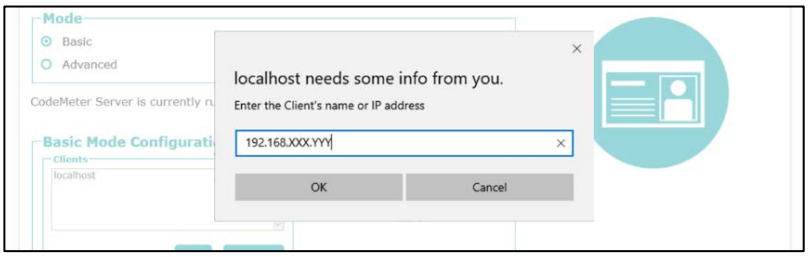
(4) “Server Access” screen appears. Check “Enable” for “Network Server” and click “Apply”.



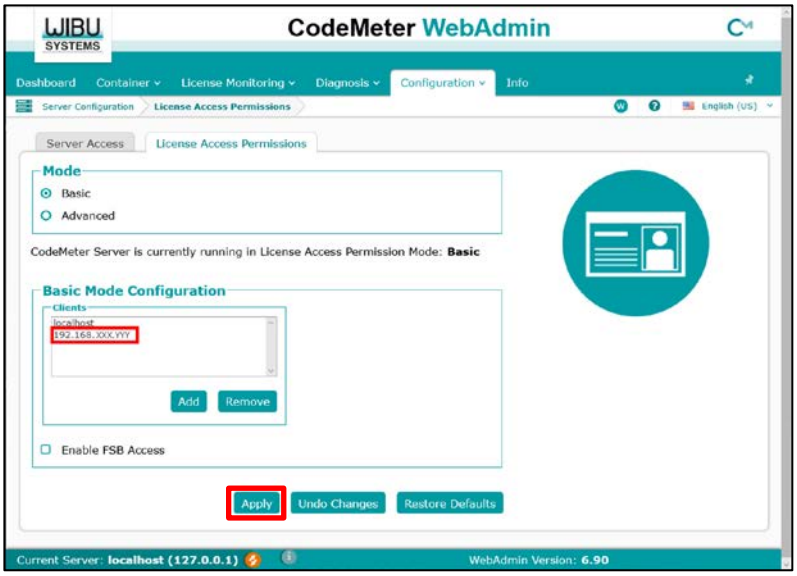
(5) Click “License Access Permission”, then click “Basic” in “Mode” and “Add” in “Basic Mode Configuration”.



(6) Input the IP address assigned to the guest OS.



(7) Confirm that the IP address of the guest OS entered in “Client” of “Basic Mode Configuration” has been added, and click “Apply”. (This procedure adds a PC that can access the license with the hardware key.)



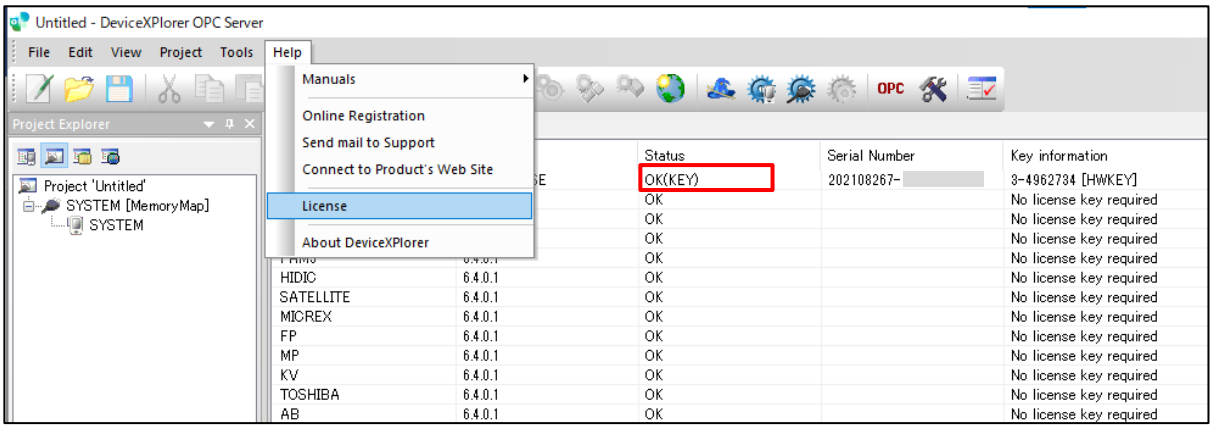
Important

Be sure to set the license access permission. (Guest OS IP specification)

Otherwise, the license assigned to the hardware key may be used from an unintended PC on the network..

3.4.14.4 License activation on the Guest OS

On the guest OS, open the DeviceXPlorer and check the license.
If the serial number is registered correctly, the host OS key is activated and the key information is displayed. If the status is OK (KEY), the license is activated correctly.
If the key information shows "License key required", the key on the host OS is not activated correctly. In this case, check the environment again.

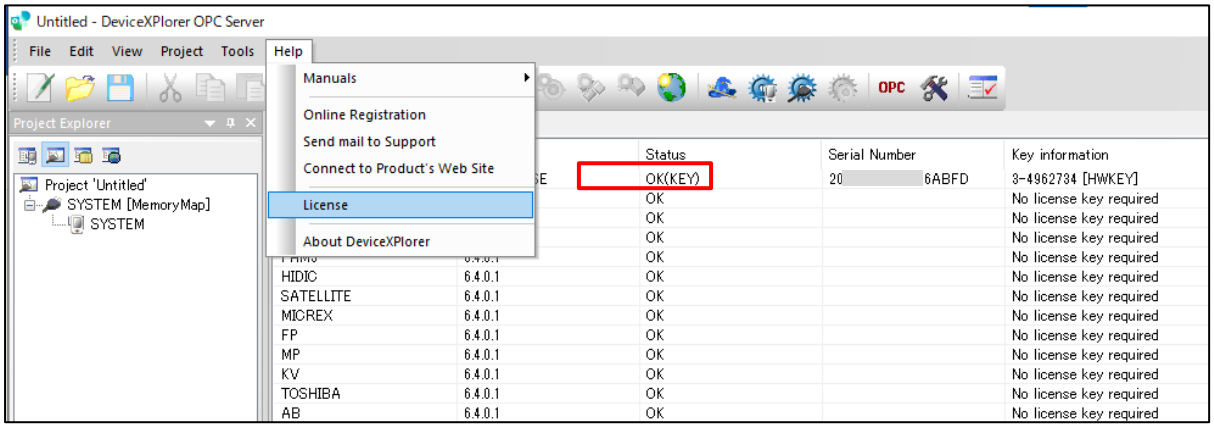


3.4.15 Using the Hardware Key

This section describes the procedure to use hardware key with DeviceXPlorer.

(1) Insert the hardware key into USB port of PC where DeviceXPlorer is installed.
*The Hardware Key must be inserted to PC while DeviceXPlorer is operating.

(2) If the key information is on the license screen and the status is "OK(KEY)", the key is activated correctly.



3.4.16 Using the Hardware Key in the Virtual Environment

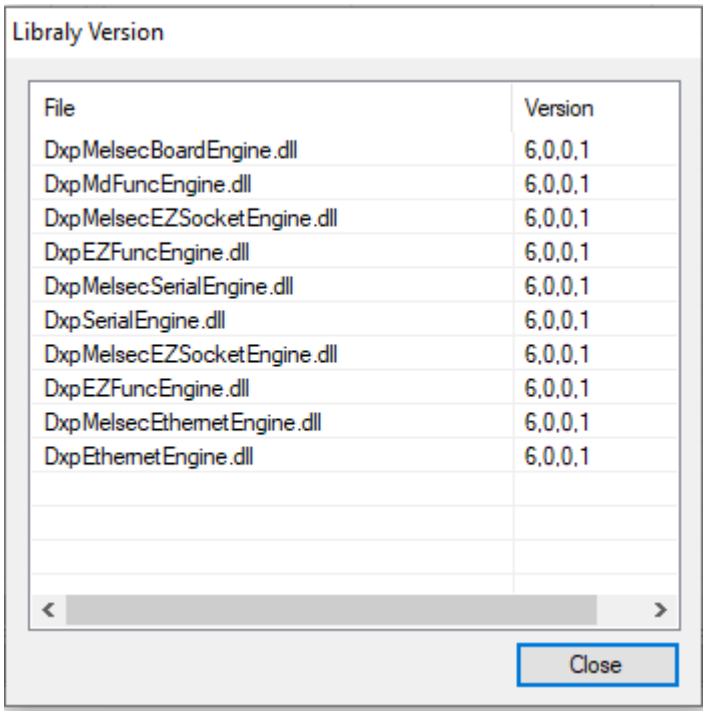
This section describes the procedure to use a hardware key with the DeviceXPlorer in a virtual environment of a guest OS.

If the virtual environment allows the guest OS to authenticate the USB device (hardware key) of the host OS, then there is no problem. However, depending on the virtual environment, the guest OS may not be able to authenticate the USB device (hardware key) of the host OS.

In such a virtual environment, please follow the procedure below to use the hardware Key in the DeviceXPlorer of the guest OS, which is activated by the host OS.
"3.4.14.1 Set up CodeMeter Runtime on Host OS"
"3.4.14.2 License activation on the host OS"
"3.4.14.3 License server activation on the host OS"
"3.4.14.4 License activation on the Guest OS"

3.5 Library Information

By selecting "Library Information" from the pop-up menu on the list displayed in the license, you can check the library information used by each component.



3.6 The Differences between each Editions

dition	Accessible models	Executable Script (Include Bridge)	OPC UA Server I/F	Modbus/TCP Server I/F
Enterprise	All models	Unlimited	Supported	Supported
Advanced	1 model	1 piece only / 1 second fixed cycle	Supported	Not Supported
Standard	1 model	1 piece only / 1 second fixed cycle	Not Supported	Not Supported

3.7 Communication Suite that can be Selected for Advanced / Standard Edition

The difference between Enterprise edition and Standard edition is listed below.

Suite	
Mitsubishi MELSEC iQ-R/Q/L/FX/QnA/A Series	YASKAWA Robot Controller
OMRON SYSMAC-NJ/CJ/CS/CP/CV/C Series	KEYENCE KV Series
JTEKT TOYOPUC-PC10/PC3J/PC2J Series	ROCKWELL AB Logix Series
YOKOGAWA FA-M3 Series	SIEMENS SIMATIC-S7 Series
HITACHI EH/H Series	AZBIL(YAMATAKE) CPL Protocol
SHARP SATELLITE JW Series	Modbus/TCP, Modbus/RTU, Modbus/ASCII
FUJI MICREX-SX/F Series	FANUC PMC Series
PANASONIC FP-X/FP7/FP2 Series	KAWASAKI Robot Controller
YASKAWA MP Series (MEMOBUS)	OPC DA2.05A/3.0 Client function
LSIS XGT Series	

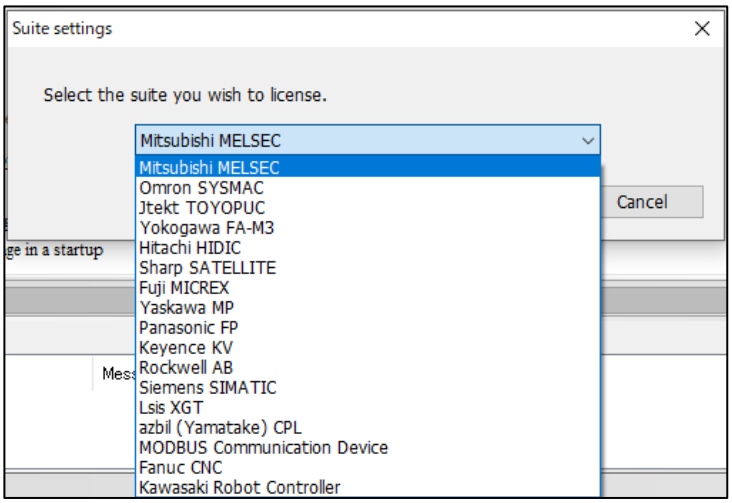
3.7.1 Selecting the Communication Suite

The procedure for selecting the communication suite in the advanced edition and standard edition is shown below.

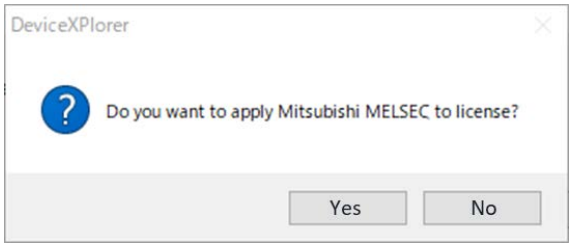
Important

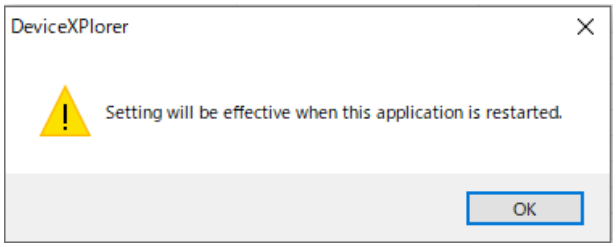
When selecting a communication suite, launch DeviceXPlorer by "run as administrator" of right click menu.

(1) After activate license for Advanced Edition or Standard Edition, if no communication suite is selected, the following dialog will be displayed. Please select the suite.



(2) Click " Yes "





Note

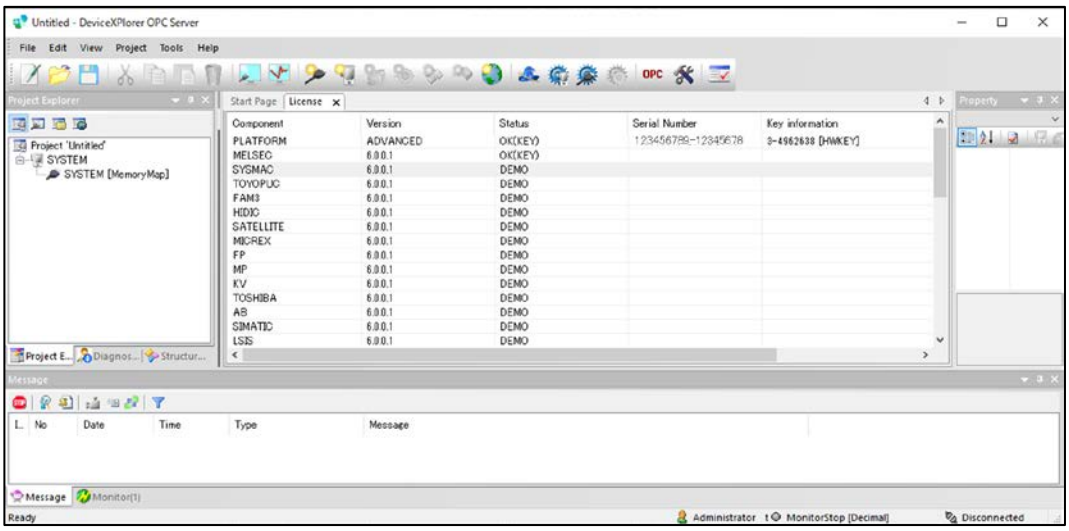
Communication suite will be valid after restart DeviceXPlorer.

3.7.2 Changing the Communication Suite from license menu.

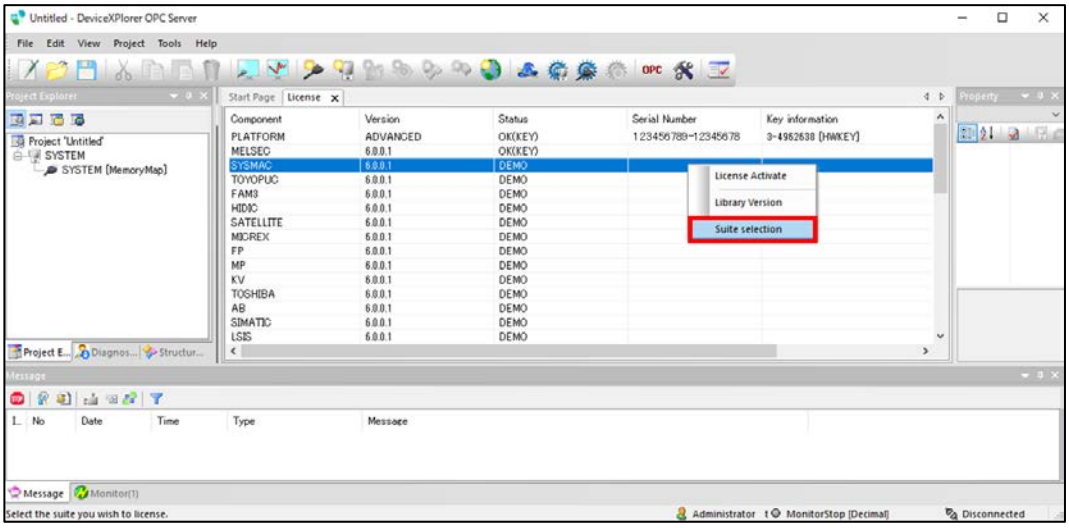
The procedure for selecting the communication suite in the advanced edition and standard edition from license menu is shown below.

Important
When selecting a communication suite, launch DeviceXPlorer by "run as administrator" of right click menu.

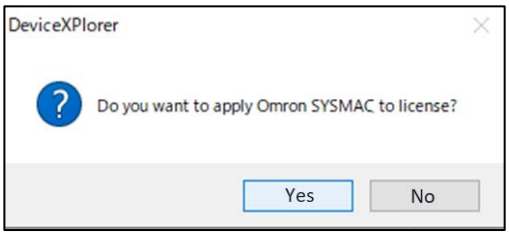
(1) Select "License" of "Help" menu.

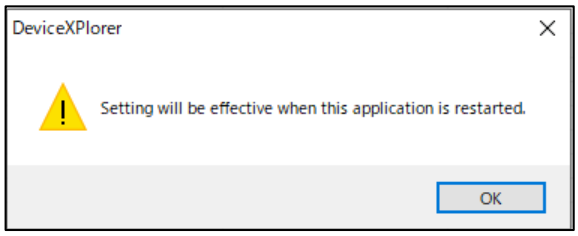


(2) Move the cursor to the component to be changed, and right-click and click "Select Suite



(3) Click "Yes".

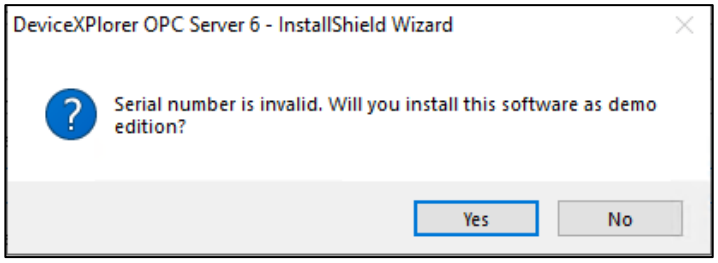




Note
Communication suite will be valid after restart DeviceXPlorer.

3.8 Demo edition

If install DeviceXPlorer as demo edition, input a hyphen symbol "-" in Serial Number when installing.
There is not the functional limit in demo edition. But demo edition runs for only one hour.



3.9 Notes on the usage by the non-administrator rights

The user cannot run the following operations, if user of the non-administrator rights uses DeviceXPlorer.
So, please be careful.

- Install / Uninstall
- Register as service program / Unregister from service program
- License Activation, Change, Clear, Transport
- Selection of communication model

[Limitation]

UAC	How to perform operation mentioned above
UAC ON	Please run by administrator.
UAC OFF	Please run by user account which has administrator rights.

3.10 Notes at 64-bit application

3.10.1 Notes on using the OPC Classic (OPC DA / OPC AE) server function

When the 64-bit DeviceXPlorer is used, to enumerate some OPC Server's from a 64-bit OPC Client application, it is necessary to enumerate using OpcEnum of the 64-bit application.
If OpcEnum registered as a service is "C:¥Windows¥SysWOW64¥OpcEnum.exe", it is necessary to carry out service registration of the "C:¥Windows¥System32¥OpcEnum.exe."
Please execute "Reg_OPCEnum32to64.Bat" in installation path as "Run as administrator".

3.10.2 Unsupported connections in 64-bit application

Some communication models are not supported in the 64-bit application.

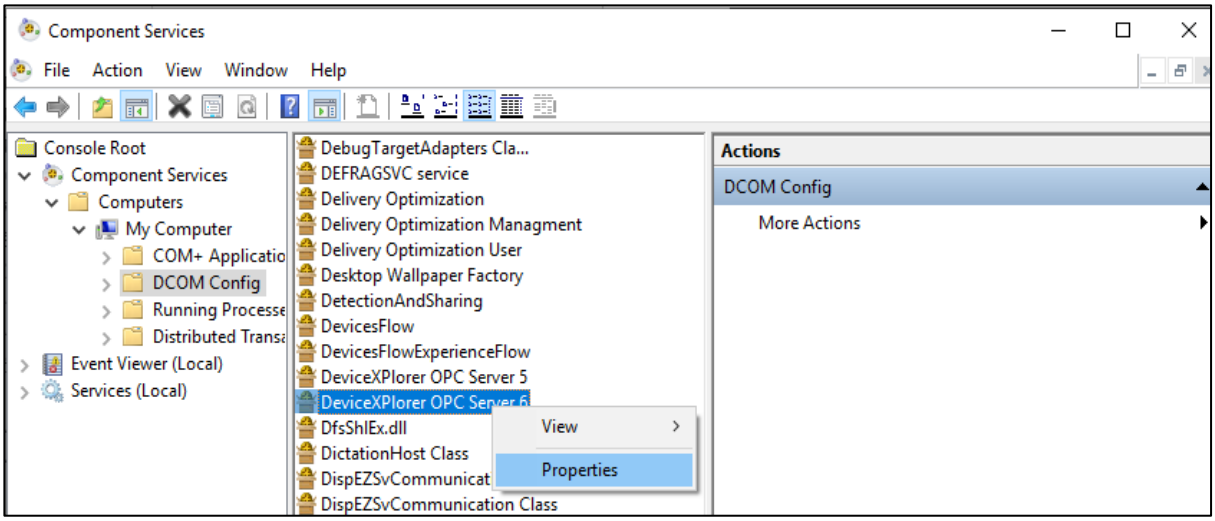
- MELSEC EZSocket / GOT connection
- SYSMAC FinsGateway / SysmacGateway /CX-Compolet connection
- TOYOPUC CPU port connection
- FANUC FOCAS connection
- YASKAWA robot connection
- Mitsubishi CNC EZSocketNc connection

3.11 Note at connecting with GENESIS 64 / MC Works64 via OPC DA interface

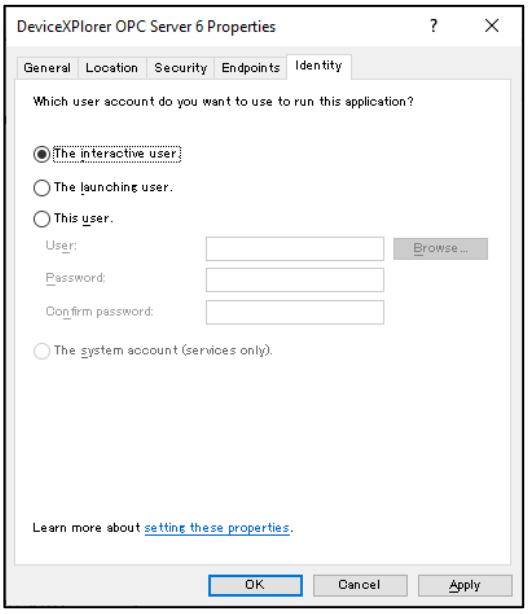
When connecting using the OPC DA interface from an application that starts as a service such as GENESIS64 / MC Works64, it is necessary to set the DCOM settings appropriately in advance. A setting example is shown below.
Please refer to 5.22 for details on DCOM settings.

(1) [Start DCOM Configuration Utility]
From the Start menu, select "Run" and enter "DCOMCNFG.EXE" to launch the DCOM configuration utility.

(2) [Specific Properties (General)]
Right mouse click on "DeviceXPlorer OPC Server 6" from "DCOM Config" under "My Computer", and select Properties from the pop-up menu.



(3) [Specific Properties (ID/Identity)]
Select the "ID" tab, and specify "The interactive user" and apply setting.



(4) Restart the PC
By restarting the PC, the application started as a service will be able to connect to DeviceXPlorer OPC Server via the OPC DA interface.

4 General Performance Specifications

4.1 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	
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.6	
		CLASS ID	8D8CD4DC-1EA6-45bf-9AAE-A69D53A50006	
		OPC Access Path	Possible	
	OPC A&E	Version	1.10	
		Prog.ID	Takebishi.DxpAe.6	
		CLASS ID	8D8CD4DC-1EA6-45bf-9AAE-A69D53A50206	
	OPC UA	URI	opc.tcp://[machinename]:52240	
		Protocol	TCP, OPC UA Pub/Sub(UADP)	
SuiteLink		Application Name	Any (default: "DXPV6")	
		Topic name	Any (default: "Device name")	
DxpLink		Port no	Any (default: 9980)	
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

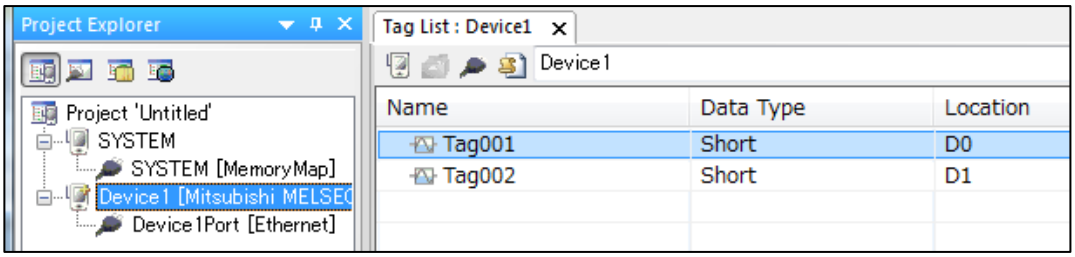
4.2 OPC DA Interfaces

4.2.1 Prog.ID

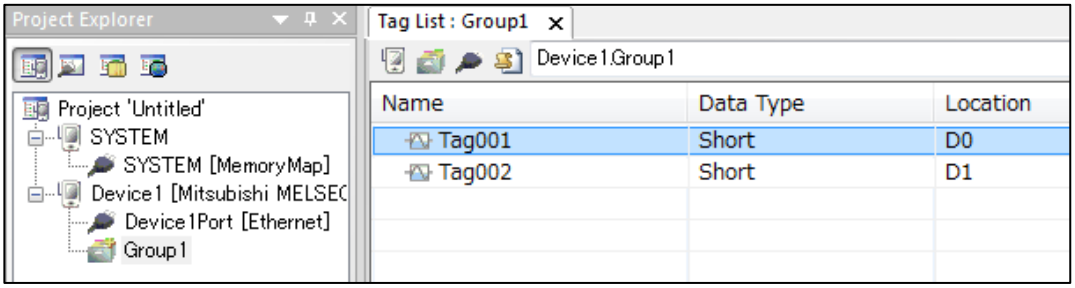
The Prog.ID for DeviceXPlorer is “Takebishi.Dxp.6” or “Takebishi.Dxp”.

4.2.2 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.

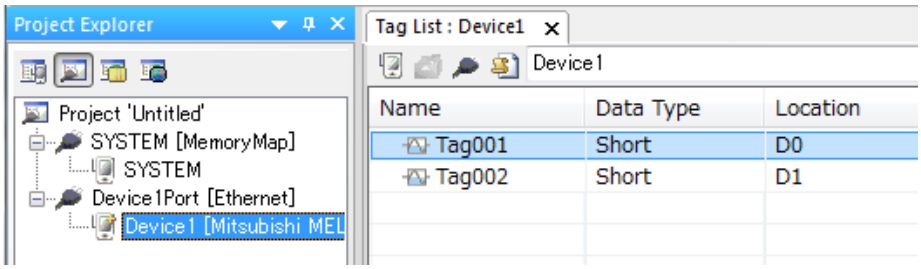


If there is a group attribution, specify the group name in between the device name and the tag name.
To access “Tag001” in the figure below, specify “Device1.Group1.Tag001” as the item ID.



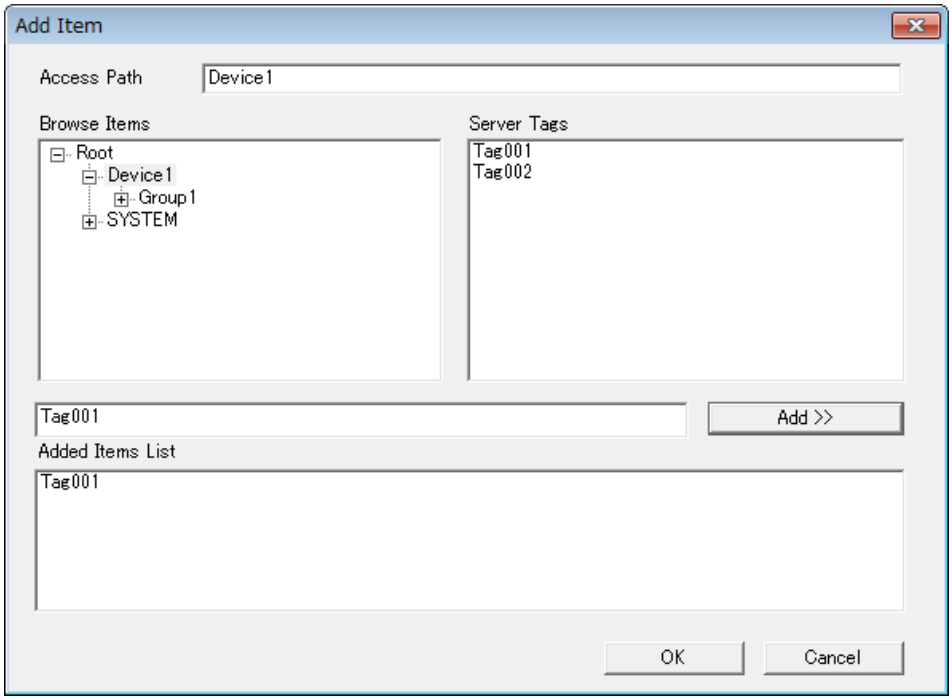
Note

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”.



4.2.3 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.



4.2.4 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 "○".
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	○	○		EnumConnectionPoints	OK

Interface for accessing the connection points for IOPCShutdown				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

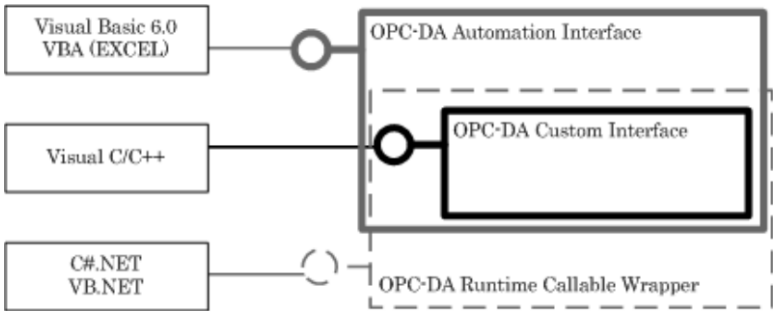
				Refresh	OK
				Cancel	OK
IOPCASyncIO2 Interface for asynchronous access	○	○		Read	OK
				Write	OK
				Refresh	OK
				Cancel2	OK
				SetEnable	OK
				GetEnable	OK
IOPCASyncIO3 Interface for asynchronous access (using MaxAge)		○		ReadMaxAge	OK
				WriteVQT	OK
				Refresh2	OK
IConnectionPointContainer Interface 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

4.2.5 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).



4.2.6 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

4.2.7 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

4.3 OPC A&E Interface

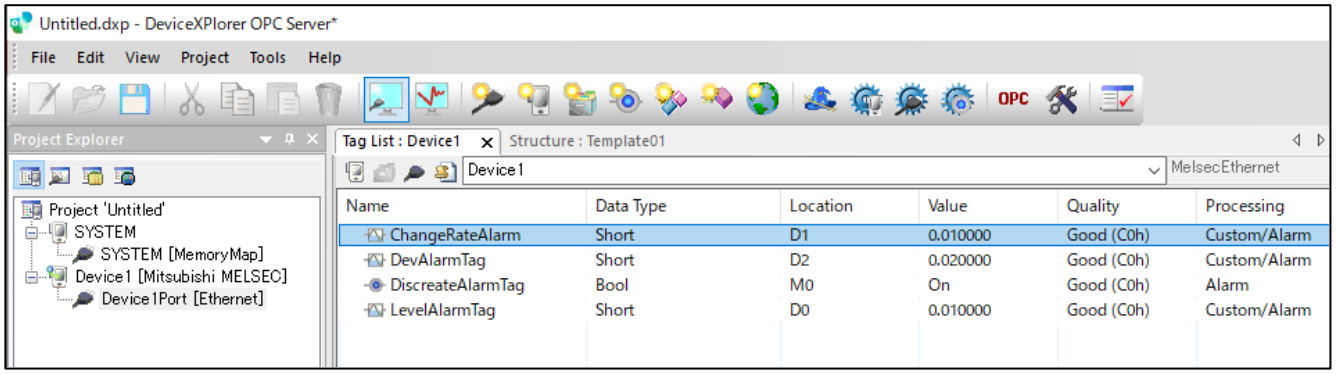
DeviceXPlorer supports OPC A&E1.10.

4.3.1 Prog.ID

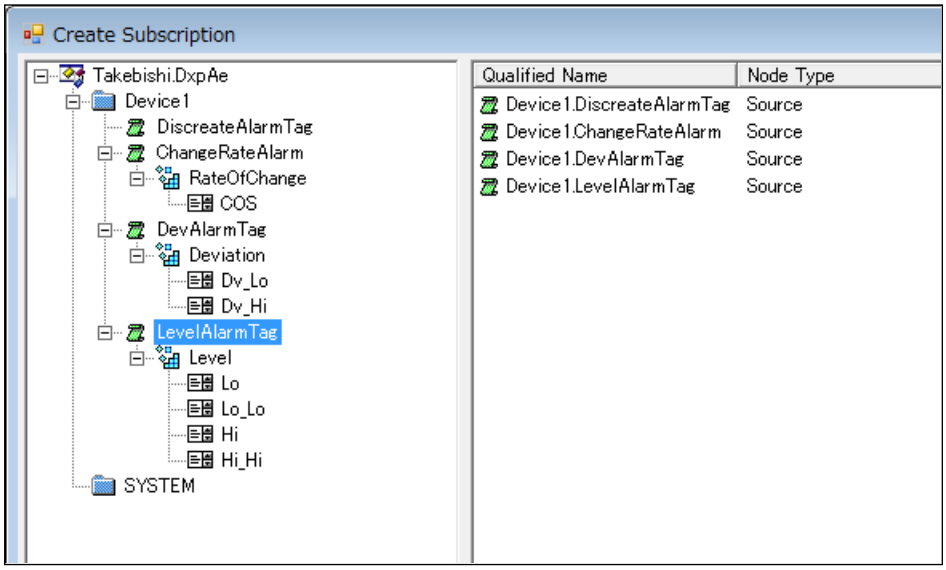
The Prog.ID for DeviceXPlorer is “Takebishi.DxpAe.6” or “Takebishi.Dxp.Ae”.

4.3.2 ItemID

OPC ItemID is the same as that of the view of an OPC DA interface. Please refer to 4.2.2 for details.



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



Important

In order to treat as an alarm tag, it is necessary to configure an alarm tag property appropriately.
Please refer to 5.7.7 for an alarm configuration.

4.3.3 Custom Interfaces

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

Object	Interface name	Opt	Member function	Support status
OPCEventServer Object	IOPCCommon		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.

4.4 OPC UA Interface

DeviceXPlorer supports OPC UA.

4.4.1 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]:52240
Profile	Standard UA Server Profile
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	50
Maximum Session Count /Client	50
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 User Management.

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

4.4.2 Service

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

Service Set	Service	Function (Our classification)	Support status
Discoverty 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	○
	TranslateBrowsePathToNodeIds	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	○
	SetPulishingMode	DA／A&C	○
	Publish	DA／A&C	○
	Republish	DA／A&C	○
	TransferSubscriptions	DA／A&C	×
	DeleteSubscriptions	DA／A&C	○

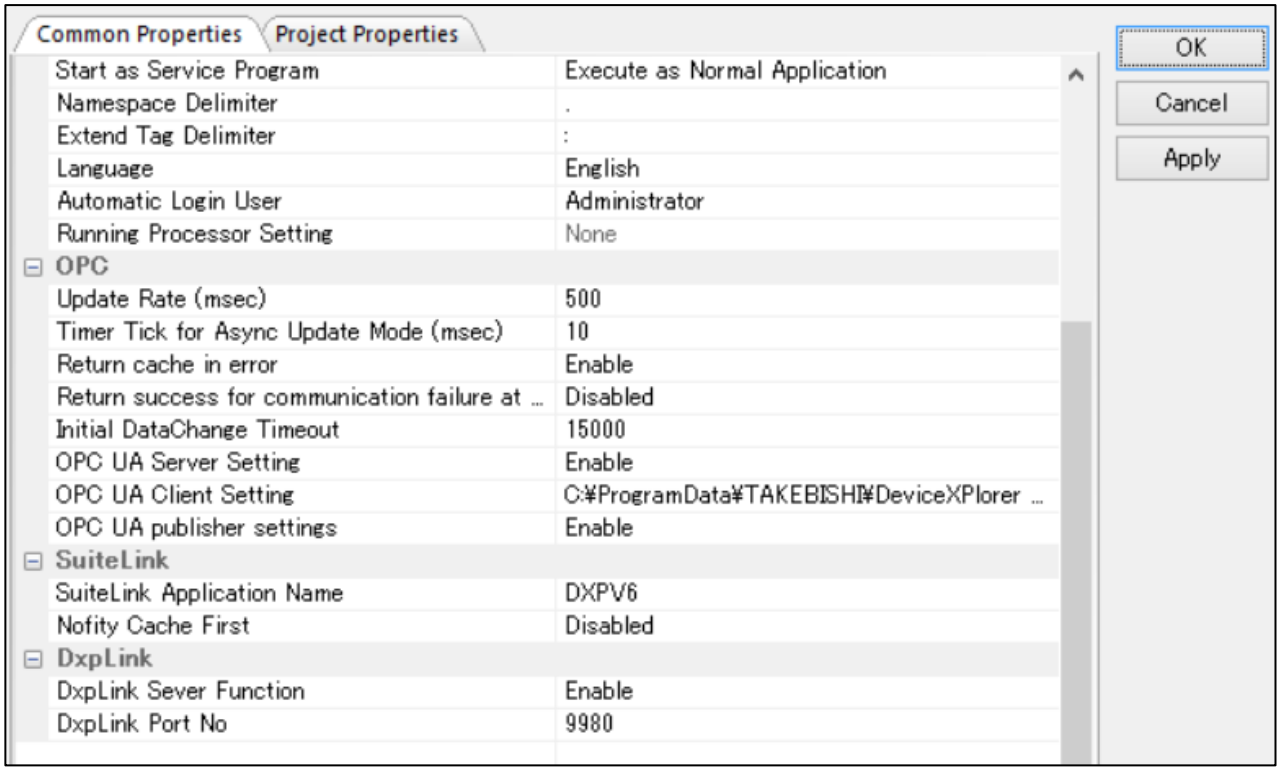
4.5 SuiteLink Interfaces

DeviceXPlorer supports SuiteLink interface of wonderware.

Important
Tags of LONGLONG and ULONGLONG are not supported for SuiteLink interface.

4.5.1 Application Name

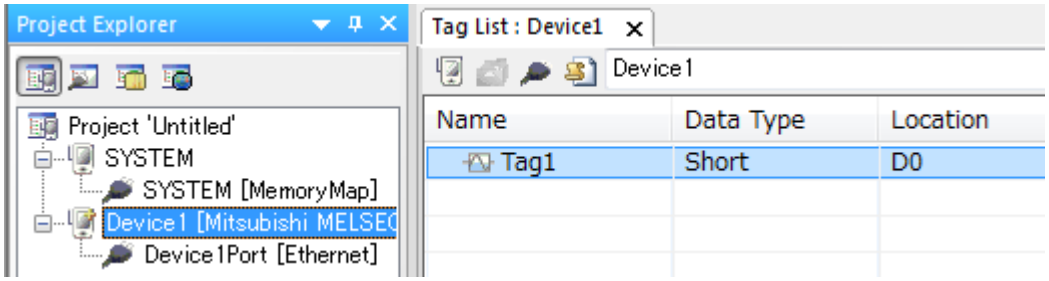
The application name for DeviceXPlorer is “DXPV6”.
The application name can be changed on the Common Properties screen (Tools -> Options)



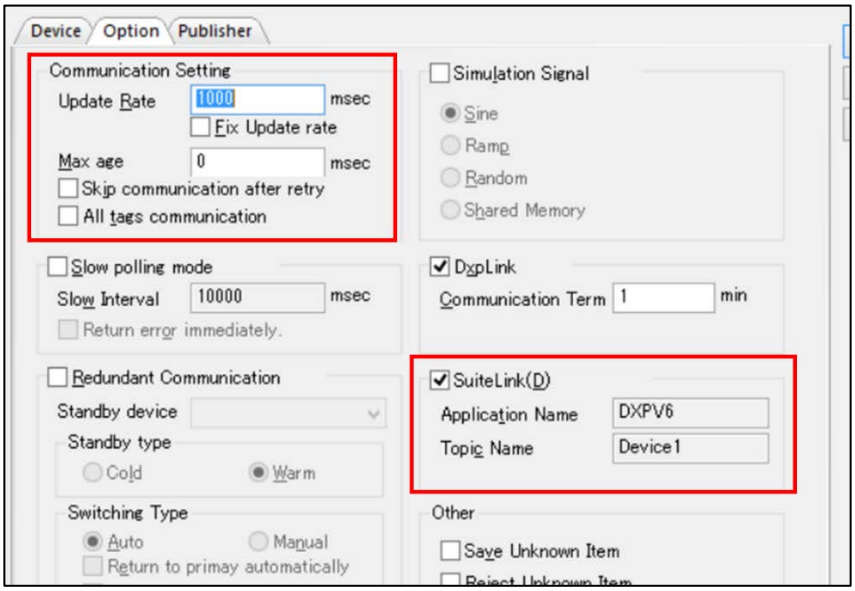
Item	Description
SuiteLink Application Name	Specify SuiteLink Server name. Default is “DXPV6”.
Notify Cache first	Send cache value of tag at first notification immediately when SuiteLink clients connect the server. You can avoid SuiteLink timeout on SuiteLink client.

4.5.2 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".



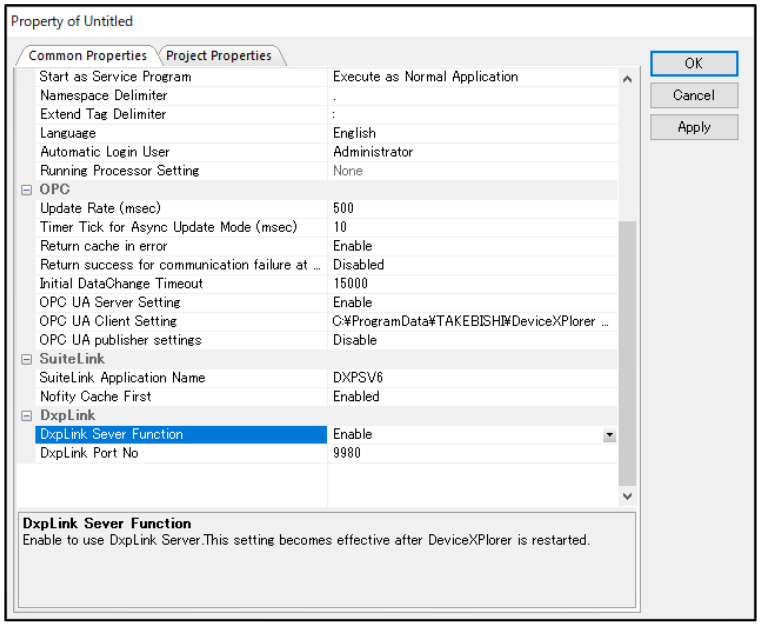
Item	Description
SuiteLink	Place a check in this box to enable the SuiteLink interface.
Application Name	Shows the currently set SuiteLink application name. This property can be changed only in Common Properties (Tools -> Options).
Topic Name	Allows you to set a topic name of your choice for each device.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

4.6 DxpLink Interface

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

4.6.1 DxpLink Server Settings

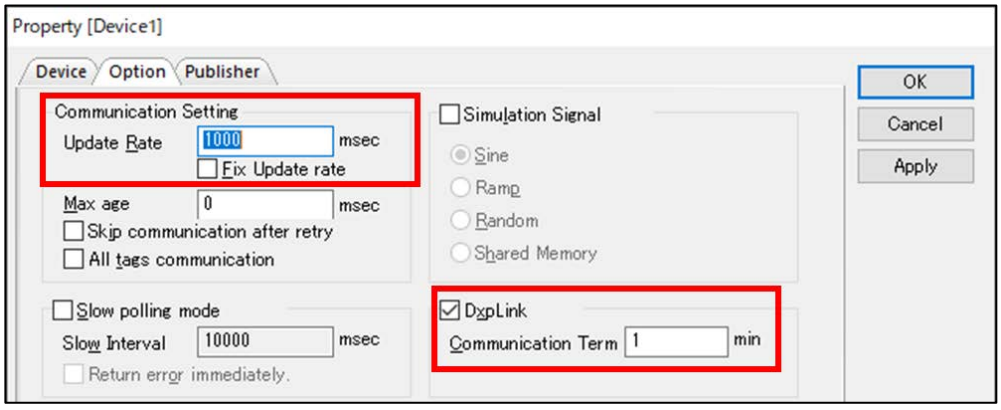
DxpLink Server function can be changed on Common Properties screen (menu Tools -> Options).



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.

4.6.2 DxpLink Communication Settings

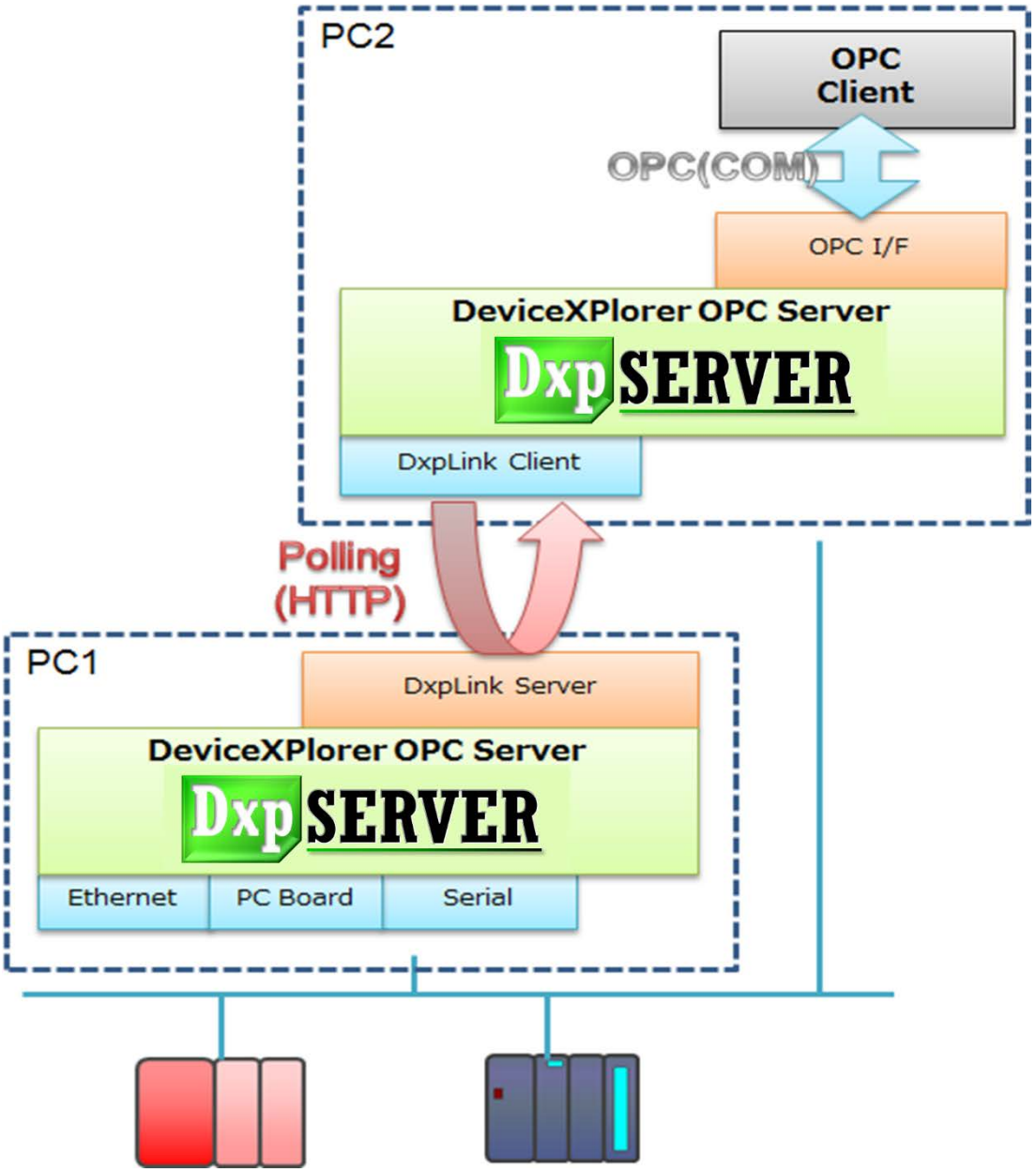
DxpLink communicates per Device. Enable / Disable of DxpLink function can be specified on "Device Options".



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.

4.6.3 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.



4.6.4 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.

4.6.4.1 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)

4.6.4.2 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."

4.6.4.3 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	

4.6.4.4 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

4.6.4.5 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"

4.7 Modbus/TCP Server Interface

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

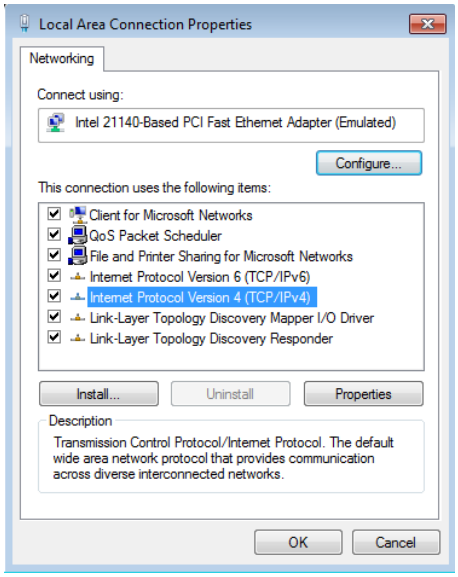
5 Operation

5.1 Tutorial

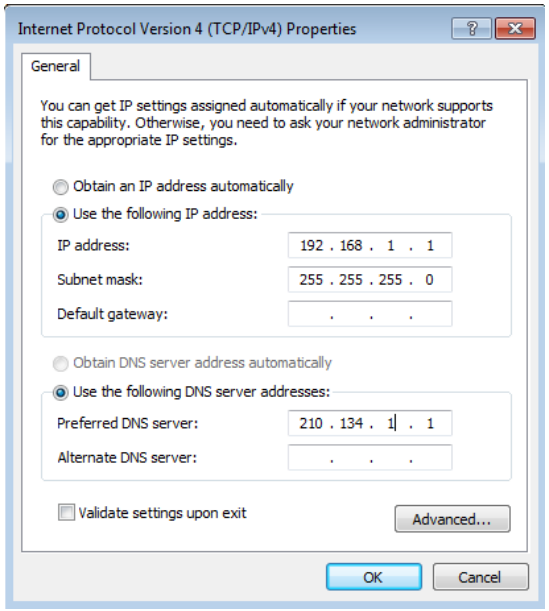
If you are using DeviceXPlorer for the first time, follow the steps below to make the necessary settings and easily check operation.

5.1.1 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".



5.1.2 Installing DeviceXPlorer

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

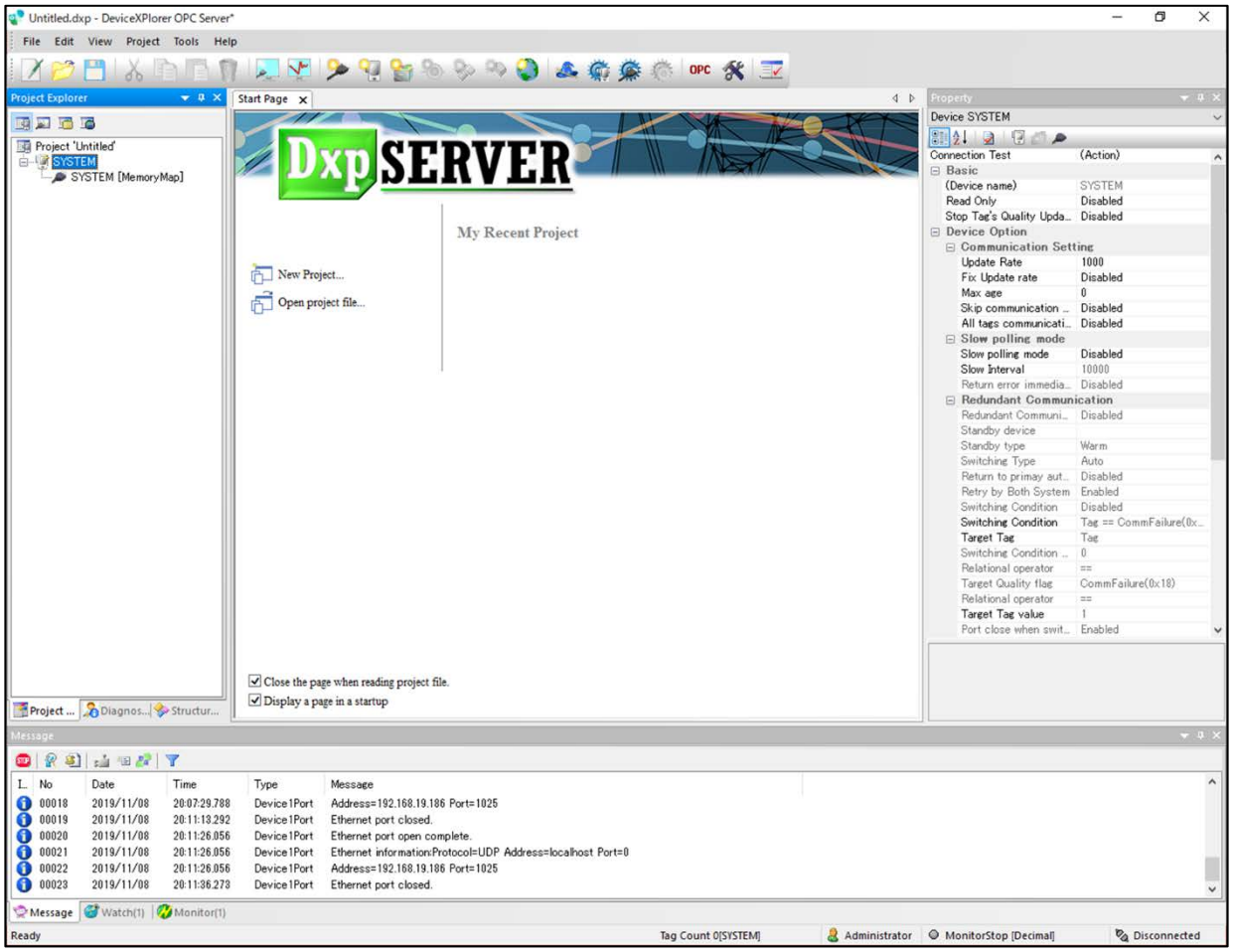
Note
Install DeviceXPlorer using an account with Administrator privileges.

5.1.3 Starting DeviceXPlorer

From the Programs menu, start “DeviceXPlorer OPC Server 6”->”DeviceXPlorer OPC Server 6”.

5.1.4 Start Page

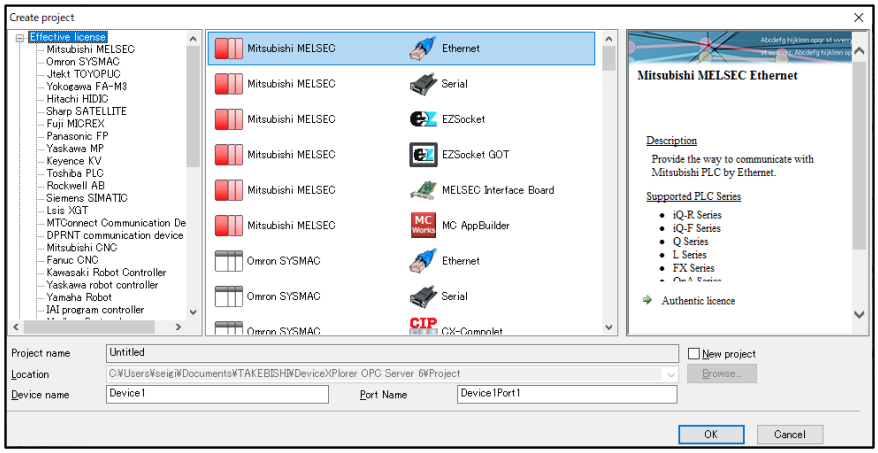
Starting of device Explorer will display a start page as follows.
Please push "New Project" and create a new project.



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.

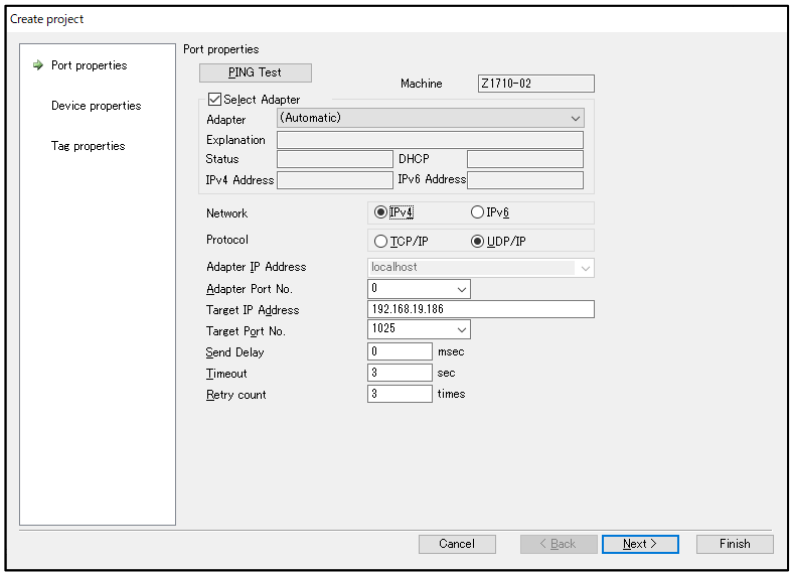
5.1.5 Project Wizard

(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."

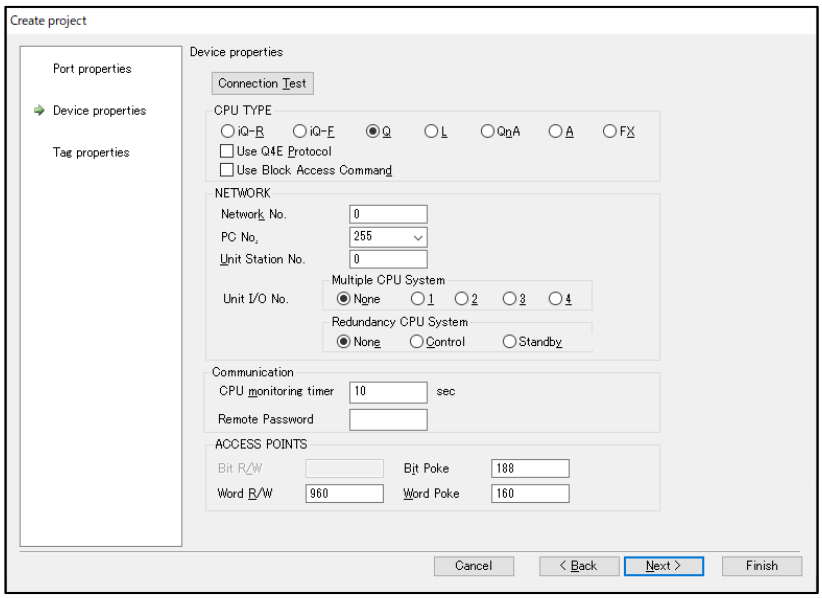


Item	Description
Effective license	The components in which the license is effective are enumerated.
Installed Component	The components which are installed are enumerated.
Project name	Specify project name.
Location	Specify project file path..
Device name	Input Device name for creating Device.
Port name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

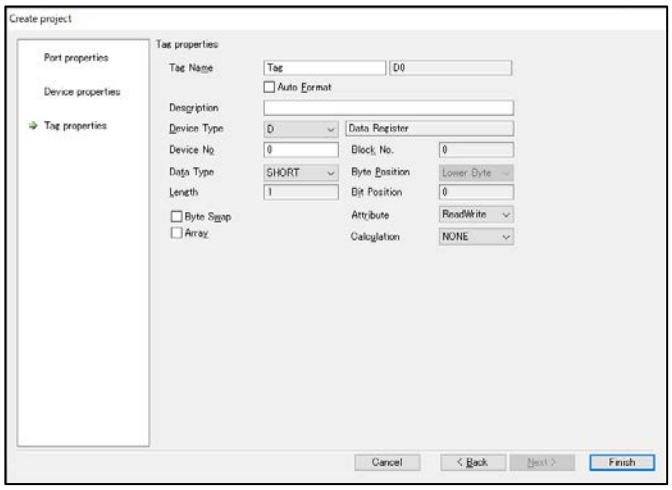
(2) Specify the port information..You can confirm the existence of the IP address you entered by clicking the "PING" button. For details, refer to 5.3.



(3) Specify the Device information.. You can check the communication with the PLC by clicking the "Connection test" button . For details, refer to5.4.

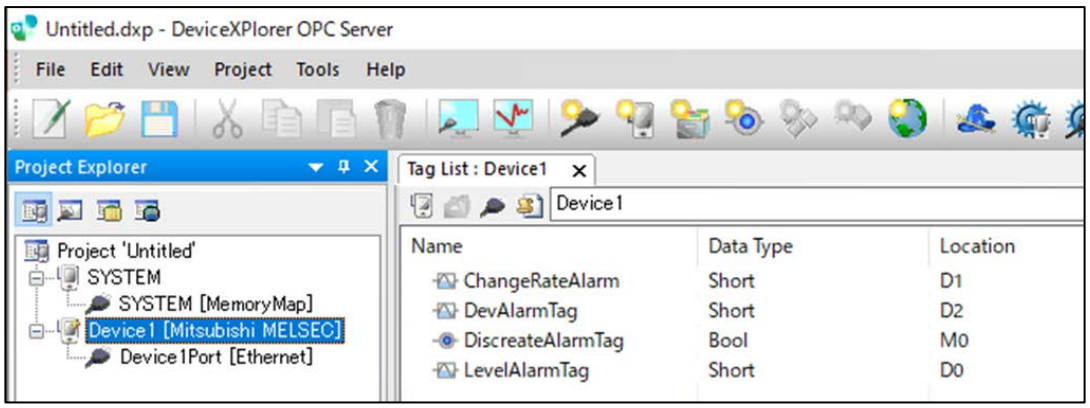


(4) Specify the Tag information.. For details, refer to 5.6.



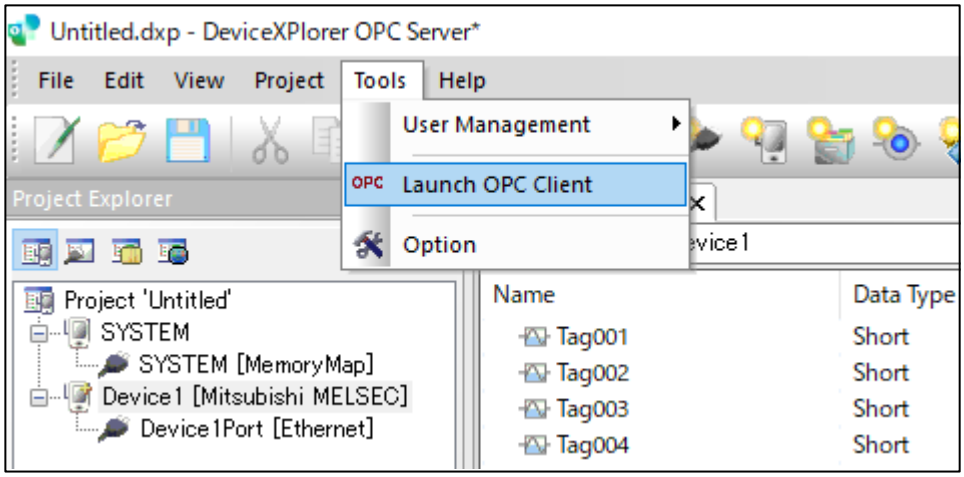
(5) Above, Port settings and Device settings indispensable for communication with partner apparatus, and Tag settings were completed.

The rest should perform addition of a tag and various option setup according to a system.

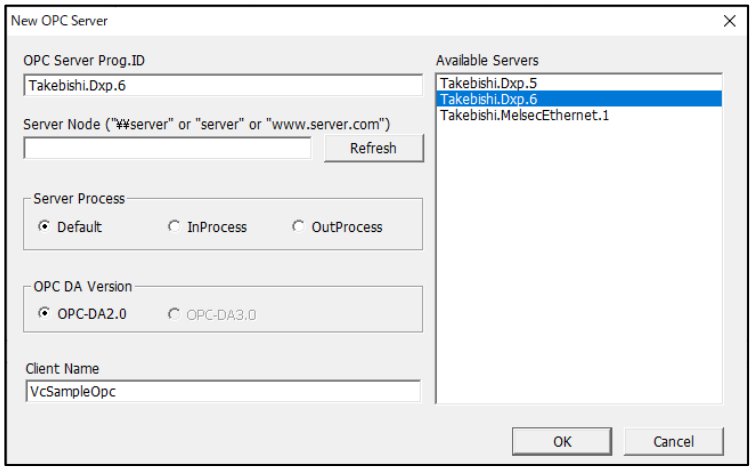


5.1.6 Accessing DeviceXPlorer from a Client

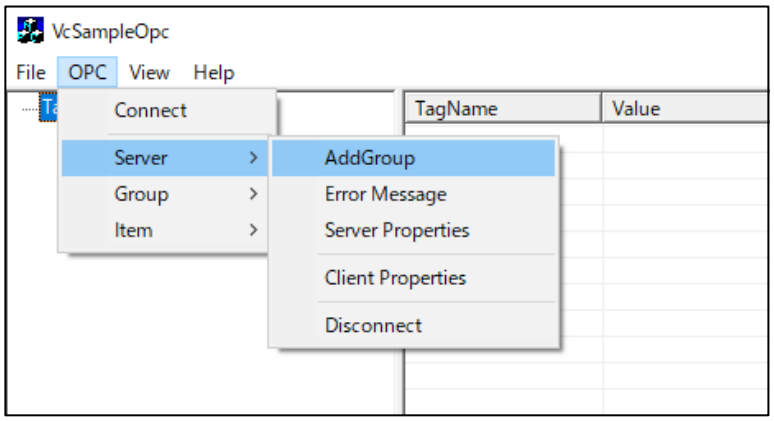
From the Tools menu, select “Start OPC Test Client”.

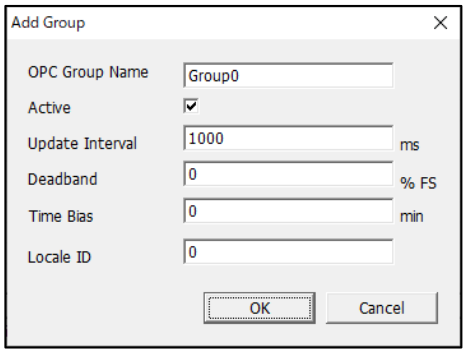


In the OPC Test Client menu, select “Connect” and connect to “Takebishi.Dxp.6”.

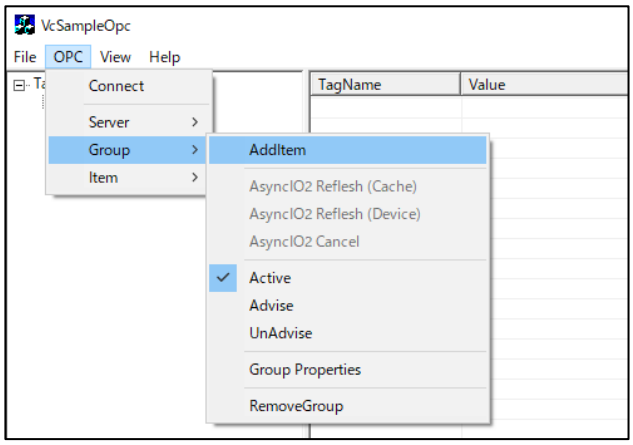


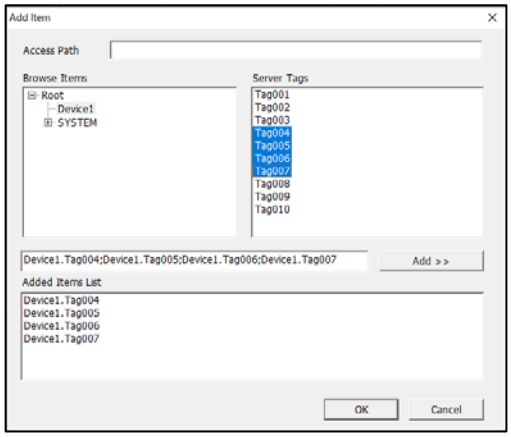
Next, select “Server” and “AddGroup” in “OPC” menu, and add group.



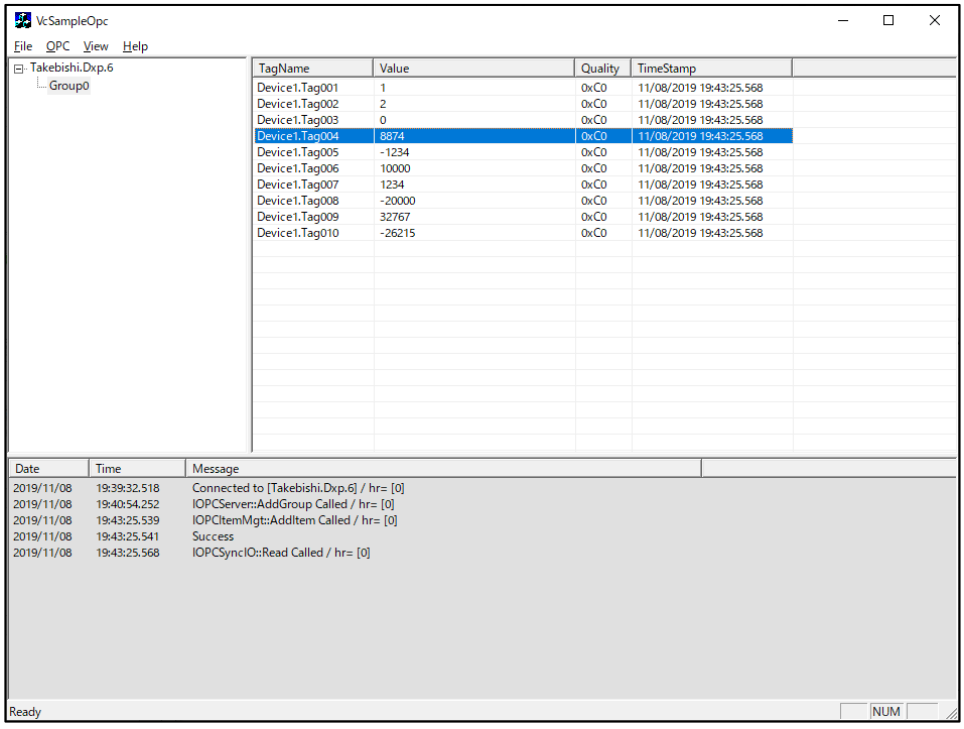


Next, select “Group” and “AddItem” in “OPC” menu, and add tags.

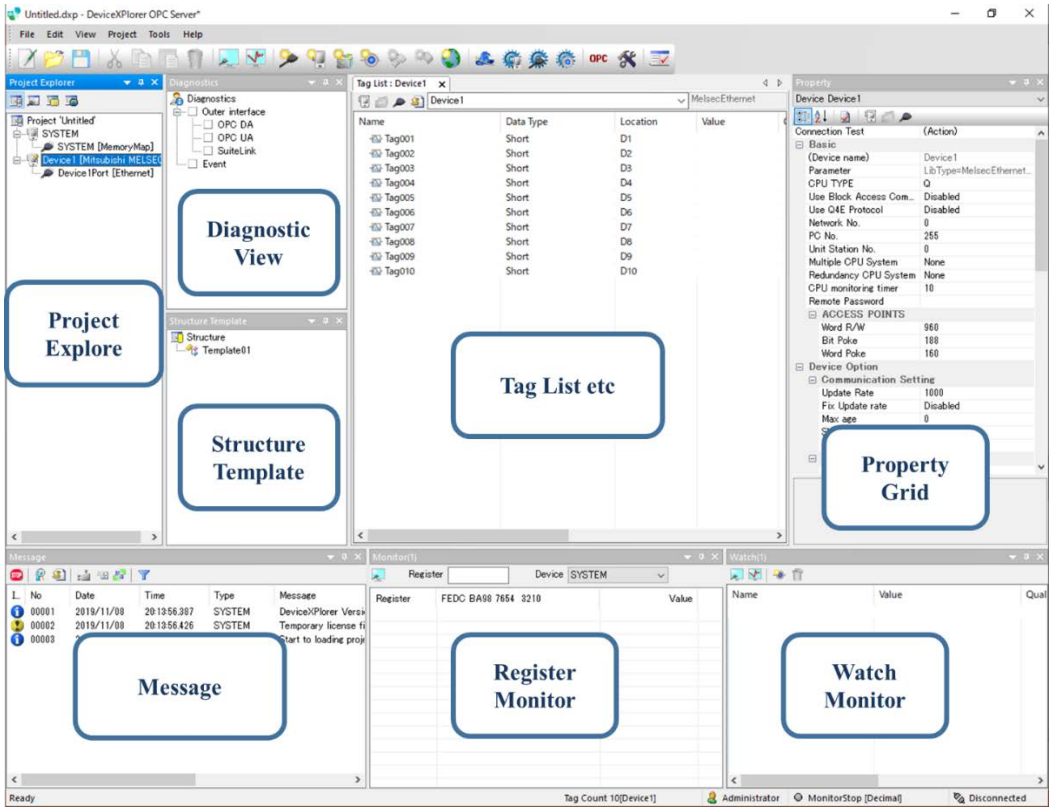




Check that the client successfully obtained data. (If “0xC0” appears under Quality, communication is functioning normally.)



5.2 Screen Layout

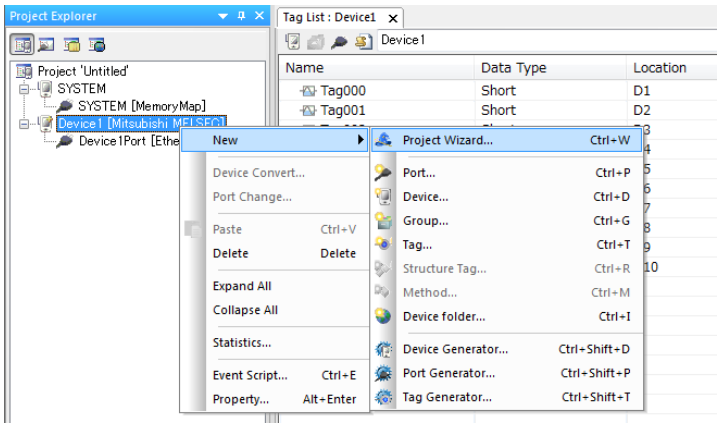


[Project Explorer]

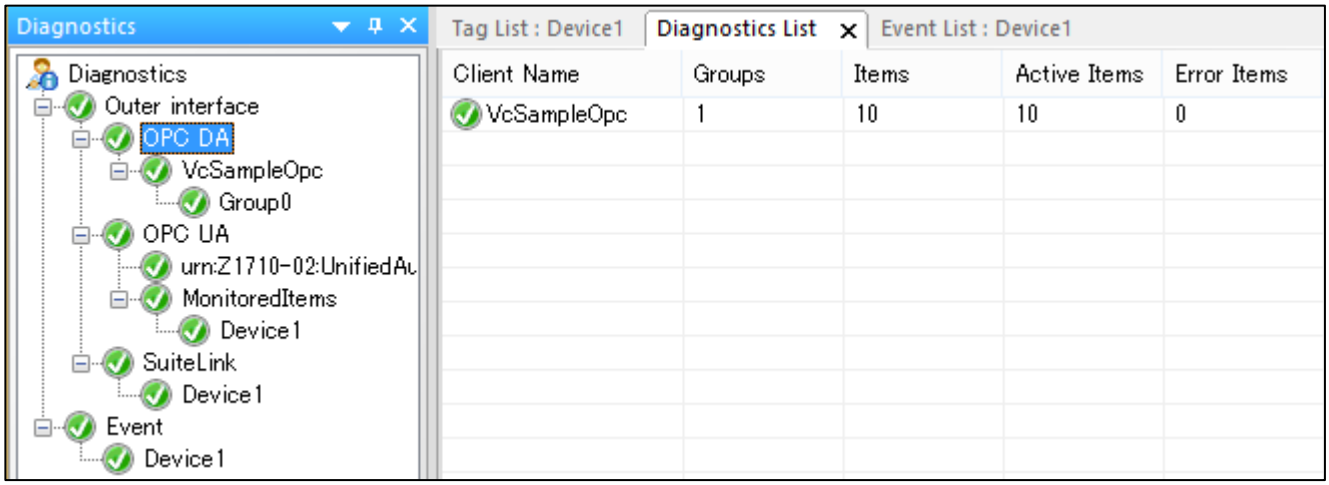
Project Explorer allows you to visually monitor the setting status of ports, devices, and groups, and allows you to create or edit them. Using the toolbar, you can also sort by device, port, or protocol (PLC).

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.

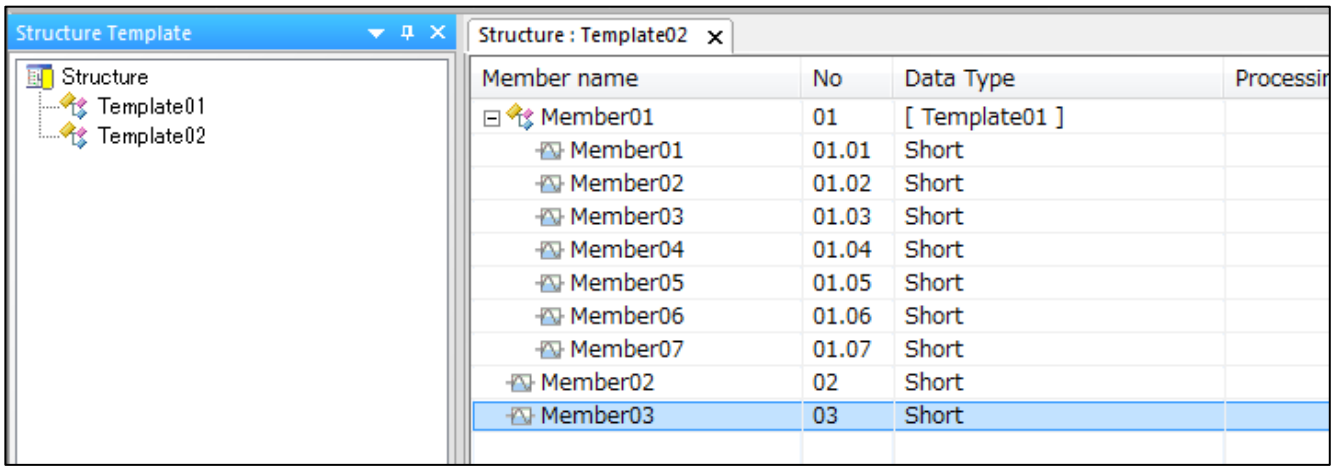
Right-click Project Explorer to view the pop-up menu of related functions.



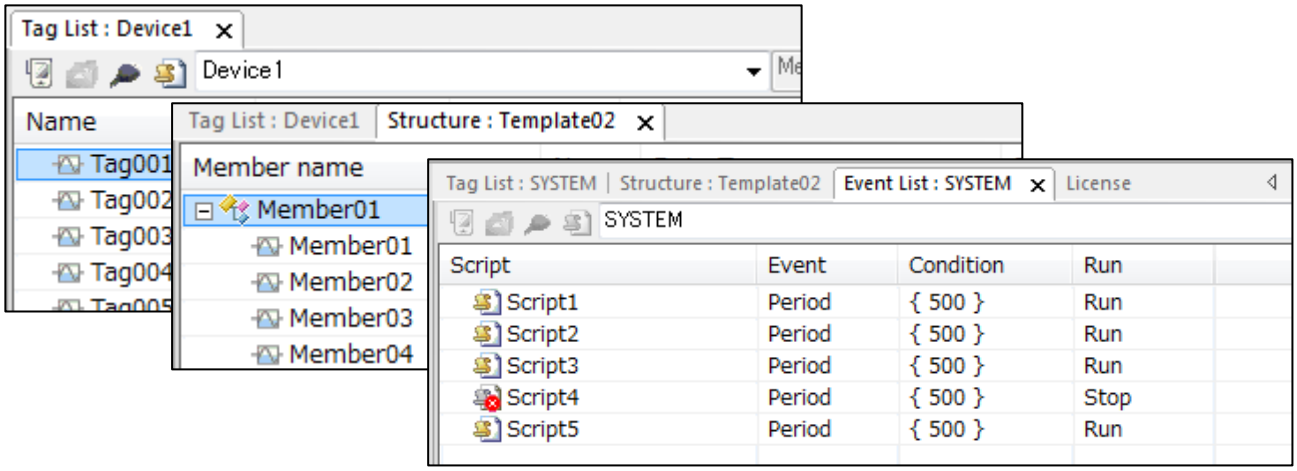
[Diagnostic View]
Diagnostic View allows you to check information such as the access status of each interface and the current value of each tag.



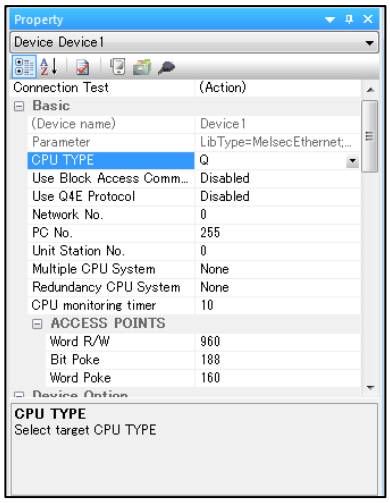
[Structure Template]
Structure Template allows you to check information of structure defined.



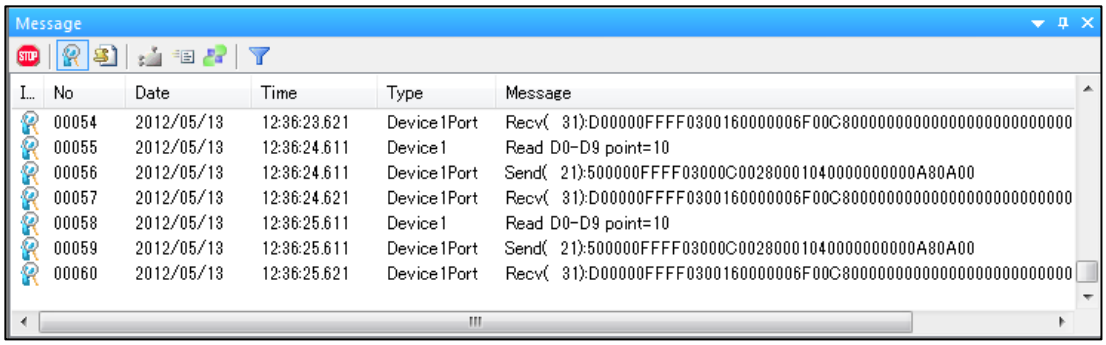
[Tag List etc.]
The property information of Tag List, Structure Template Tag List, Script List etc. are shown.



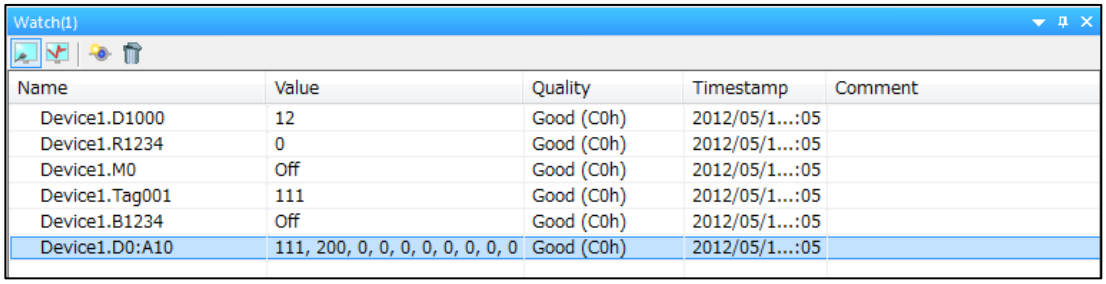
[Property Grid]
This grid shows information of selecting Object. And you can configure object.



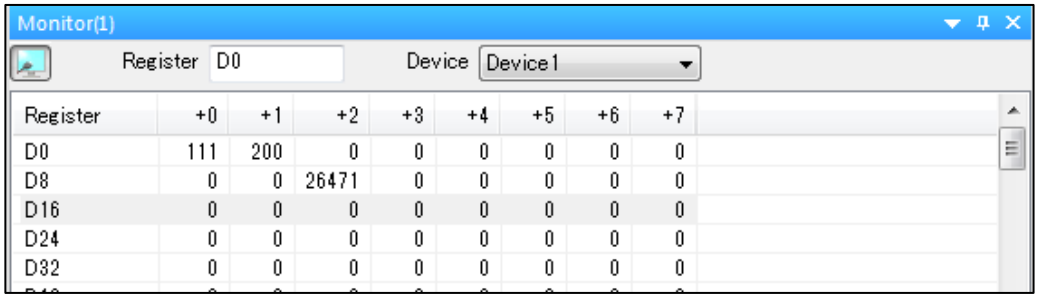
[Message View/Watch View]
This shows messages such as communication error information and diagnostic information.



[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.



5.2.1 Menus

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

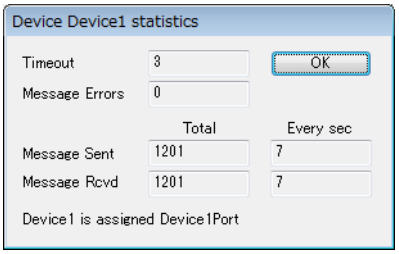
[File Menu]

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	INFO	Shows general information messages.
	WARNING	Shows warning messages.
	ERROR	Shows error messages.
	FATAL ERROR	Shows fatal error messages.
	Script log	Shows Script messages.
	Communication Data	Traces the status of communication with the PLC.
	OPC Interface Access	Traces the access status of the OPC interface.
	SuiteLink Trace	Traces the access status of the SuiteLink interface.
	DxpLink Trace	Traces the access status of the DxpLink interface.
	All on	Enables all of the above 6 messages.
	All off	Disables all of the above 6 messages.
Window	Project Explorer	Shows Project Explorer.
	Message View	Shows Message 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
Pause Message Display		Temporarily pauses the output of trace information.
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.

[Help Menu]

Item		Description
Manuals	User's Guide (Server Edition)	Shows Server Edition.
	User's Guide (PLC Communication Edition)	Shows PLC Communication Edition.
	OPC-DA 3.0 Specification (Custom Interface)	Shows the OPC DA interface specifications.
	OPC-DA 2.05A Specification (Custom Interface)	
	OPC-DA 2.0 Specification (Automation Interface)	
	OPC-AE 1.10 Specification (Custom Interface)	Shows the OPC AE interface specifications.
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.

5.2.2 Toolbar

Toolbar and shortcut key are assigned to following as.

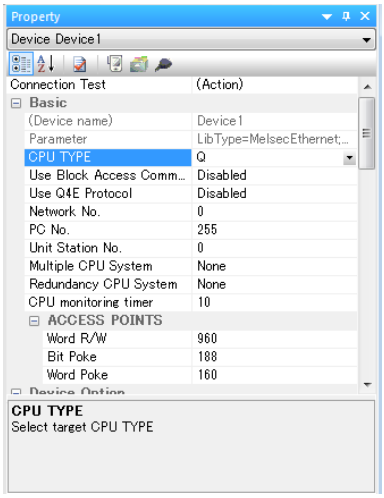
FUNCTION	ICON	SHORTCUT
New		Ctrl + N
Open		Ctrl + O
Save As		Ctrl + S
Cut		Ctrl + X
Copy		Ctrl + C
Past		Ctrl + V
Delete		DEL
Device Monitor		F3
Cache Monitor		F4
Create Port		Ctrl + P
Create Device		Ctrl + D
Create Group		Ctrl + G
Create Tag		Ctrl + T
Create Structure Tag		Ctrl + M
Create Method		Ctrl + 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		
Option		
Project Property		Alt + F7

5.2.3 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

5.2.4 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.

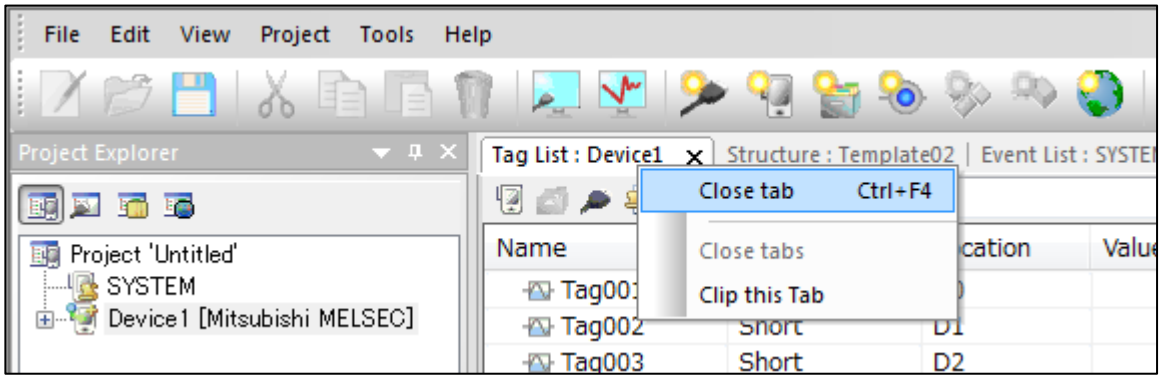


Property Grid icons are assigned to following as.

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.

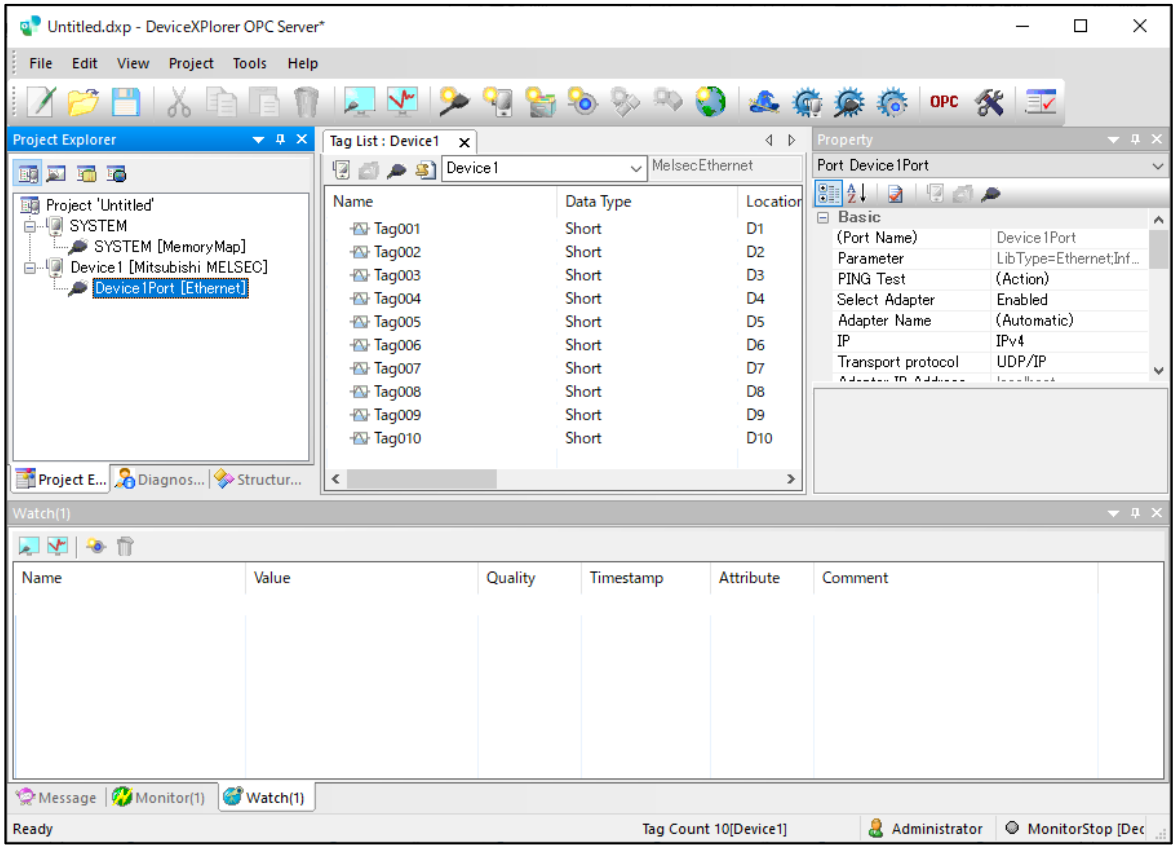
5.2.5 Clip of List View

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.



5.3 Ports

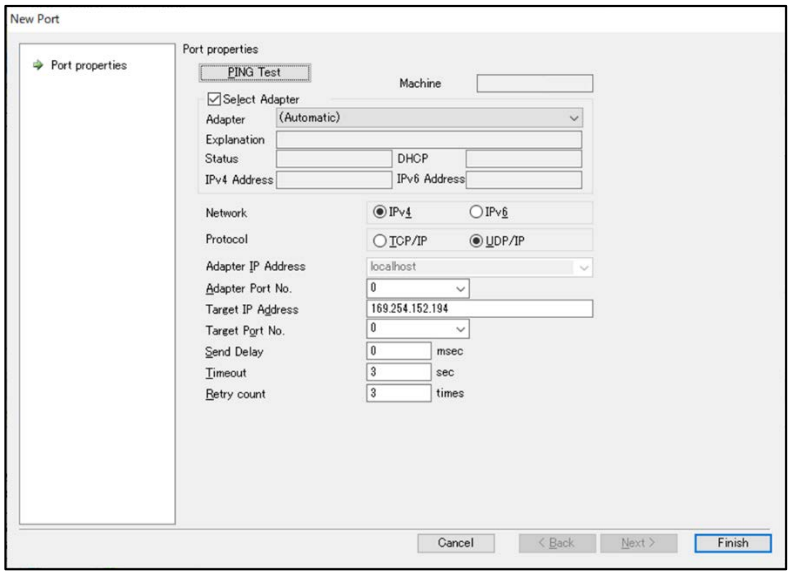
Select Project View -> Ports to view port information. Port information shows a dialog that varies depending on the type of communication port, such as Ethernet or serial.



Important

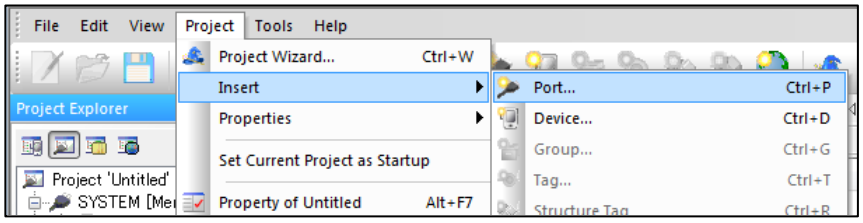
- You can create up to 255 ports.
- A Port name is a name on management and is not related to the name space accessed from a client.

The following are displayed in Port Settings of Ethernet. Setting properties differ for every communication port. Please refer to 5.3.4 if you use Ethernet port or refer to 5.3.5 if you user Serial port. For other ports, refer to the "User's Guide (PLC Communication Edition)".

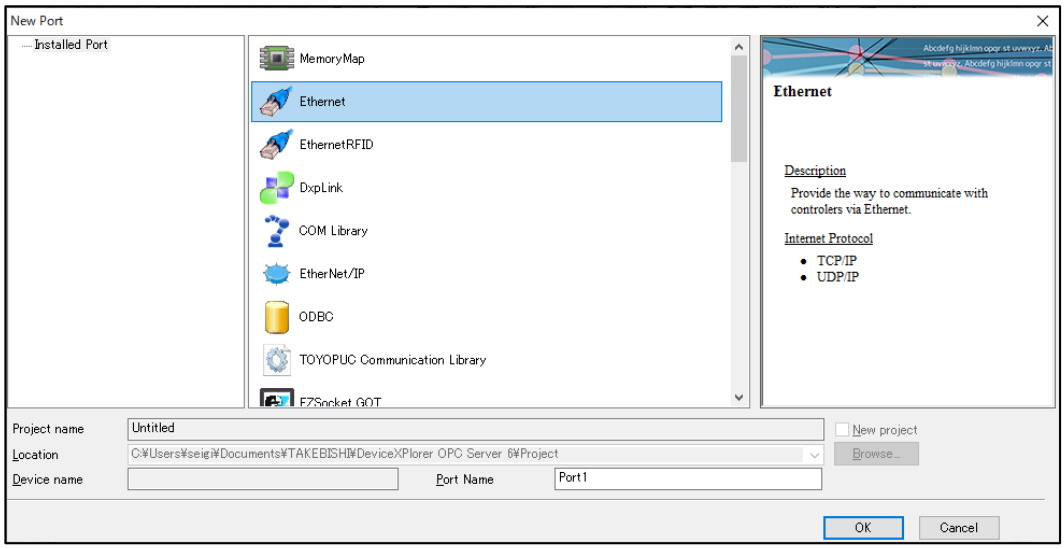


5.3.1 Creating New Ports

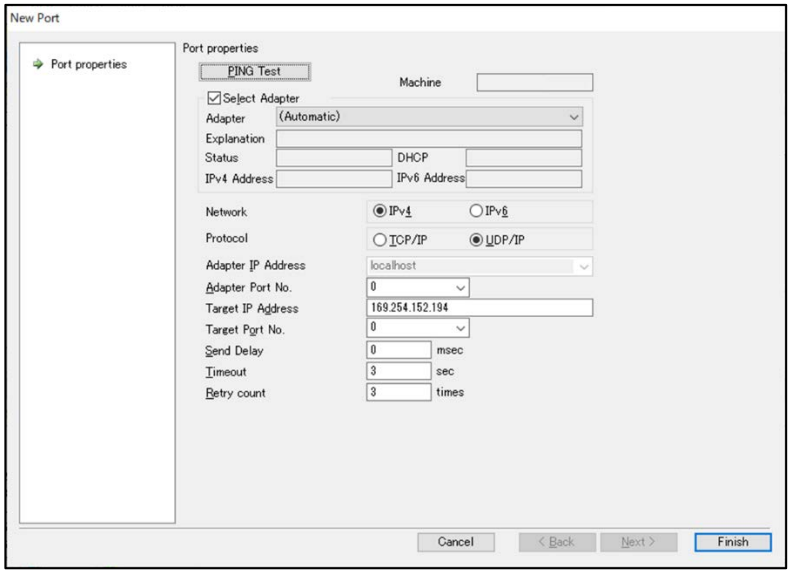
From the Projects menu, select Insert-> Port.



Select the port type and input the Port name.

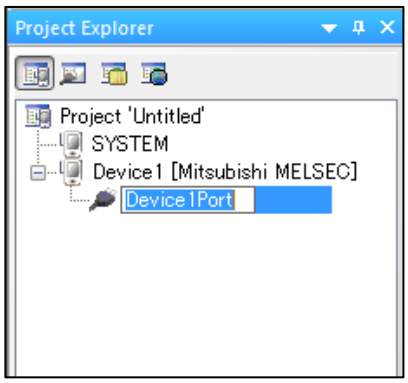


The port dialog will appear. Enter the communication parameters and click “Finish”.



5.3.2 Changing the Port Name

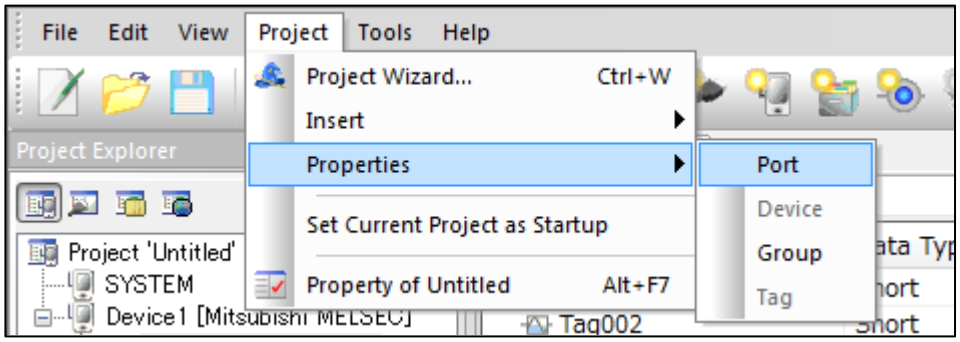
You can change the port name by either clicking on the port name again once it is selected, or by using the F2 key to enable in-place activation.
* You cannot change to a port name that already exists.



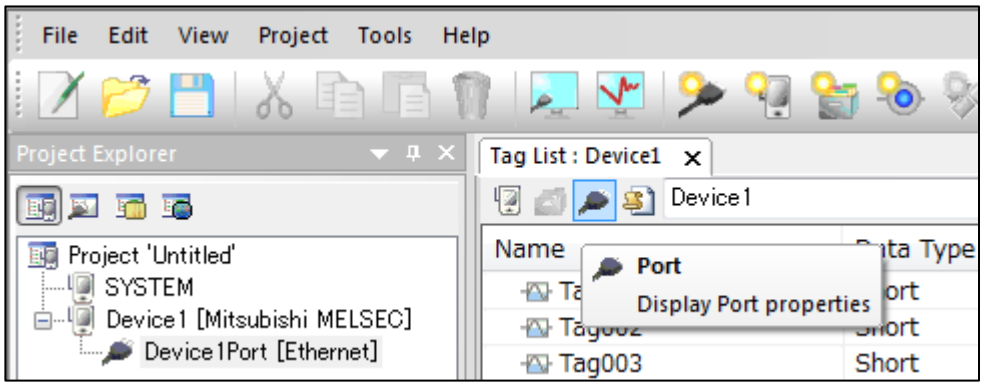
5.3.3 Port Settings

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".

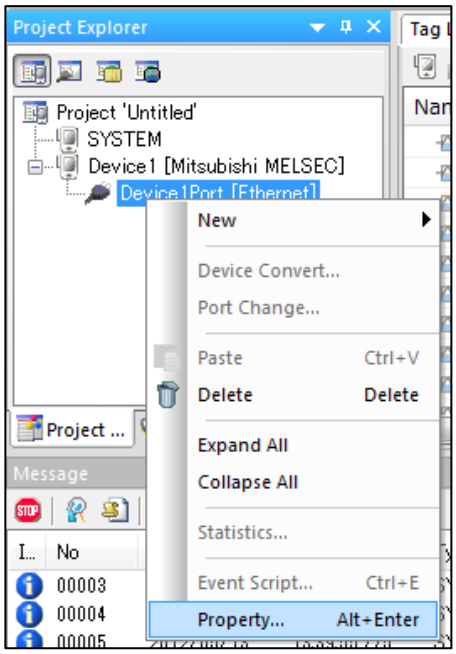
[Project menu]



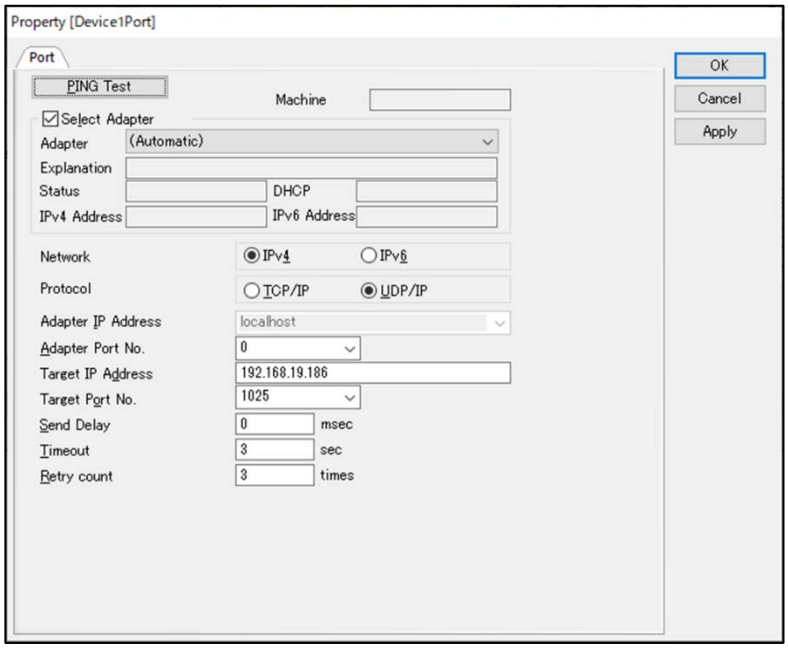
[Port Jump button]



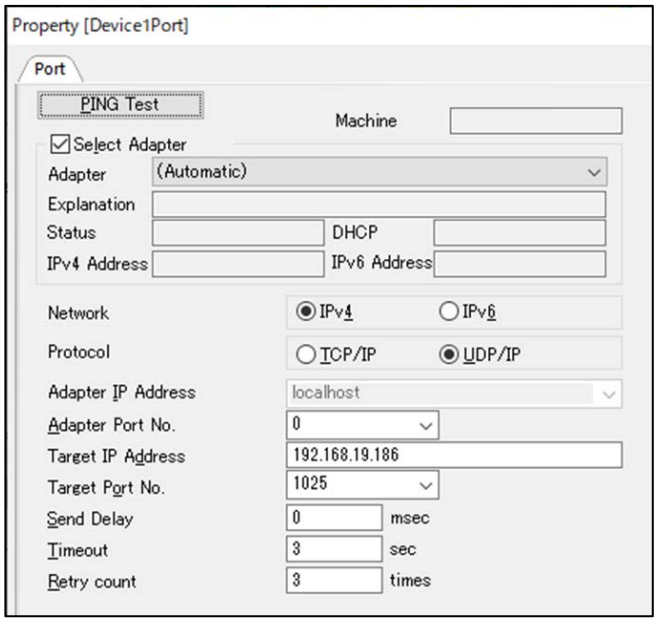
[Right Click]



The port dialog will appear. Enter the communication parameters and click “OK”.
Please refer to 5.3.4 if you use Ethernet port or refer to 5.3.5 if you user Serial port. For other ports, refer to the “User’s Guide (PLC Communication Edition)”.



5.3.4 Setting Ethernet Ports



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.

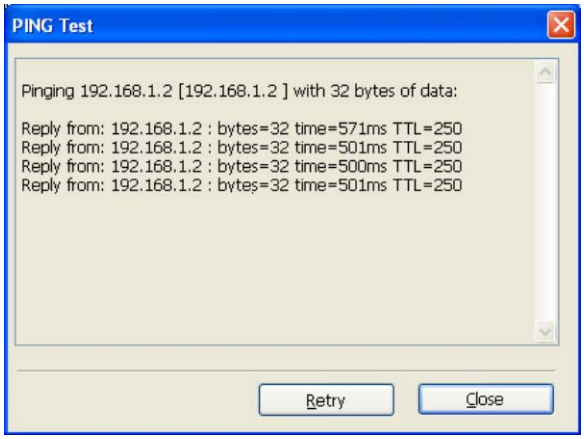
Note

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.

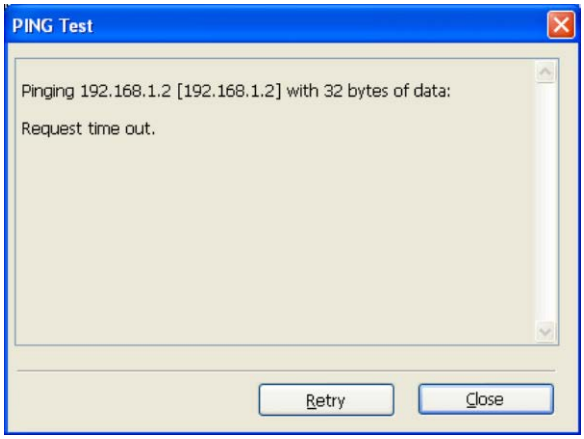
[PING]

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 …" >



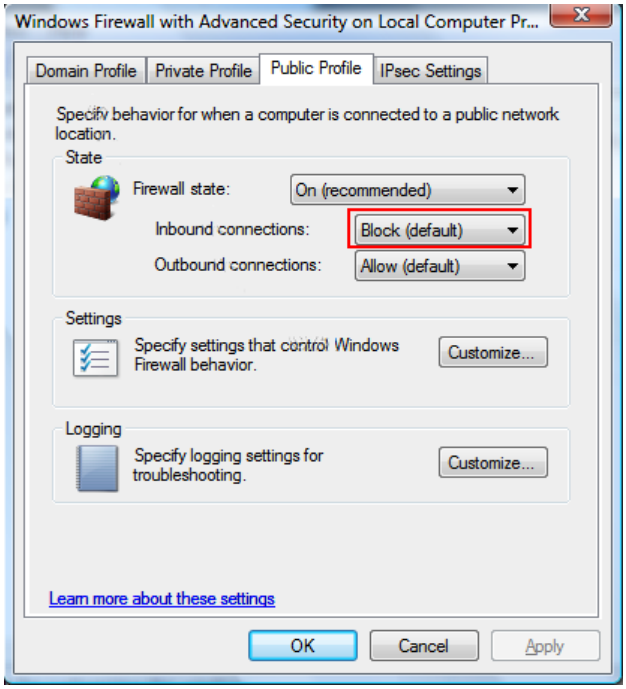
< Failed: "Reply Timeout …" >

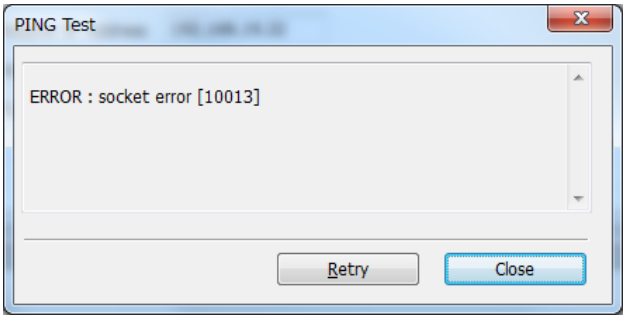


Important

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]"

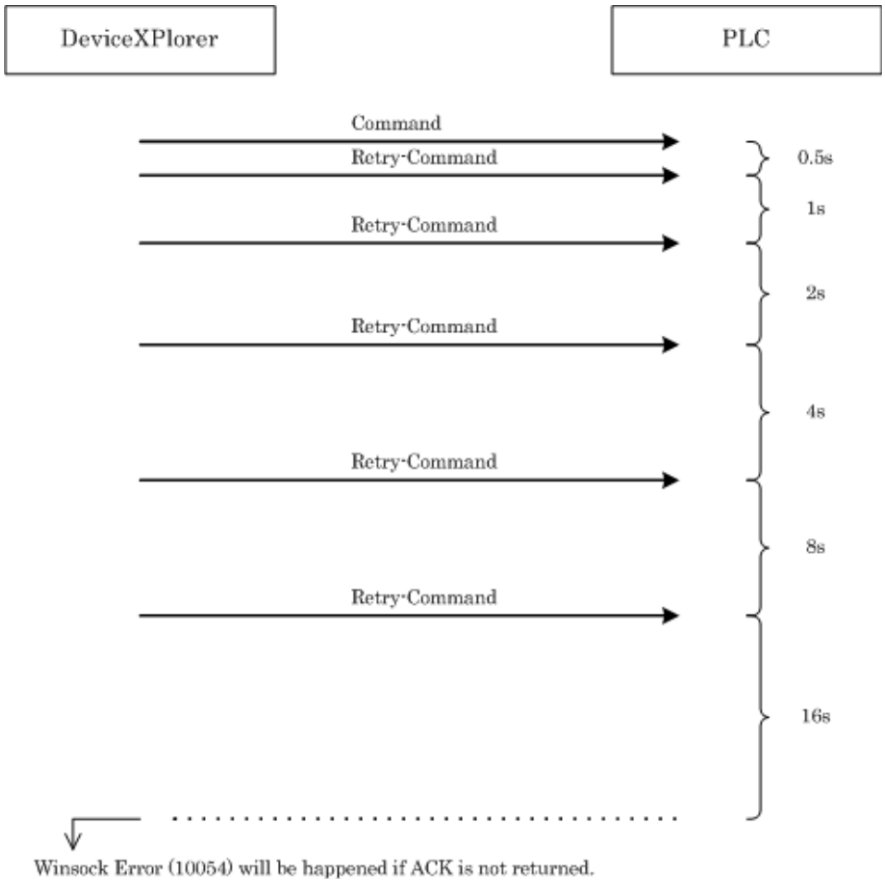




5.3.4.1 Timeout Settings for TCP / IP

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.

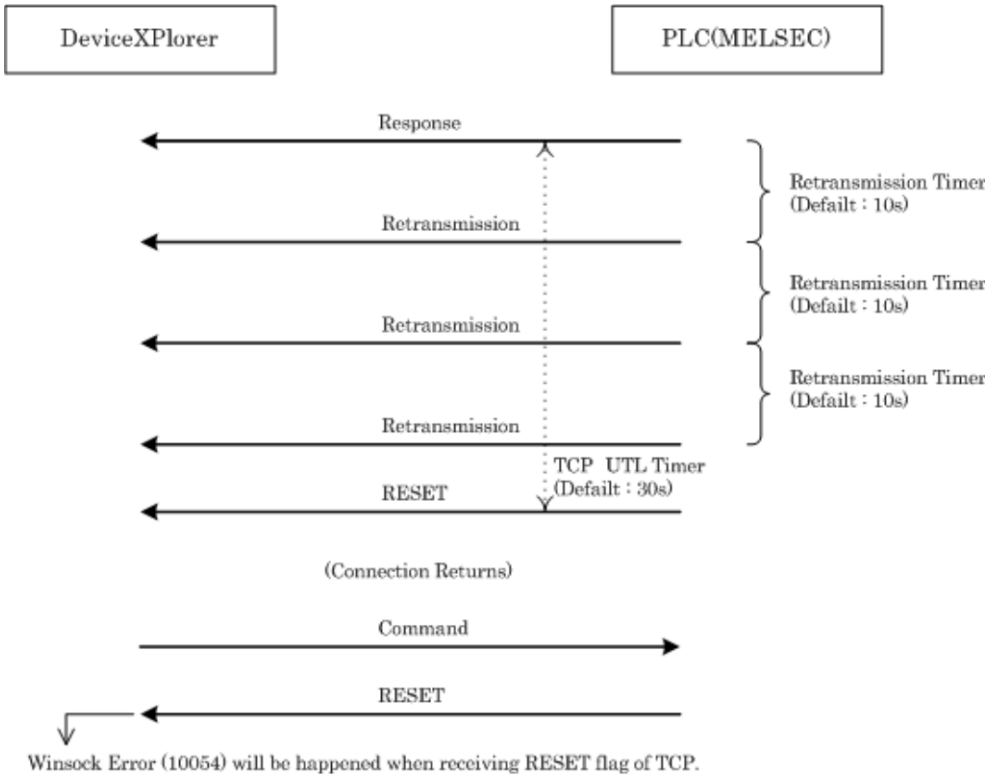
[Sending Commands from PC]
After a PC has sent a command, if there is no acknowledgement returned from the PLC, the command is resent at the TCP level at the intervals shown below.



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)]

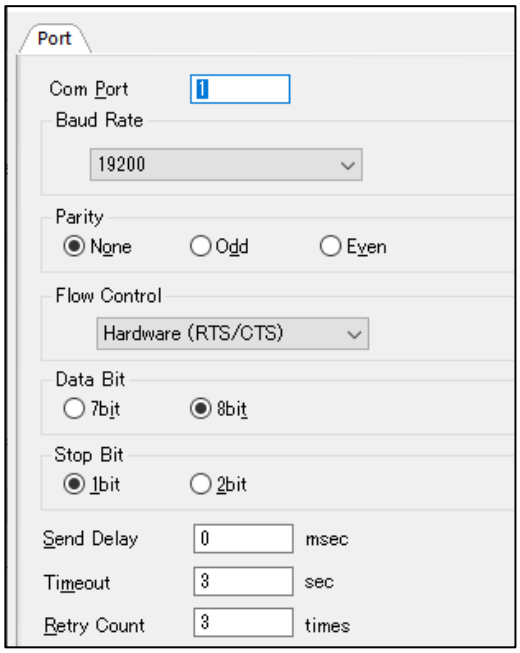
In the MELSEC-Q series, if there is no acknowledgement from the PC after the PLC has sent a response, retransmission occurs at the TCP level in accordance with the TCP retransmission timer (default: 10 seconds). When the TCP ULP timer value is exceeded, a RESET flag is sent.



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".

5.3.5 Serial Port Settings



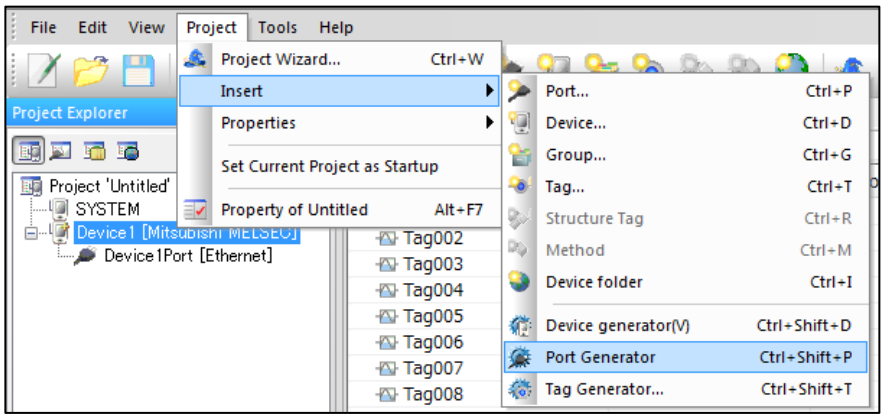
Item	Description
Com Port	Set COM port number between 1 and 256.
Baud rate	Select baud rate for communication.
Parity	Set the parity bit, on or off.
Flow control	Select "Hardware (RTS / CTS)" to perform RTS / CTS flow control.
Data length	Specify the data length.
Stop bit	Set the stop bit.
Send delay	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

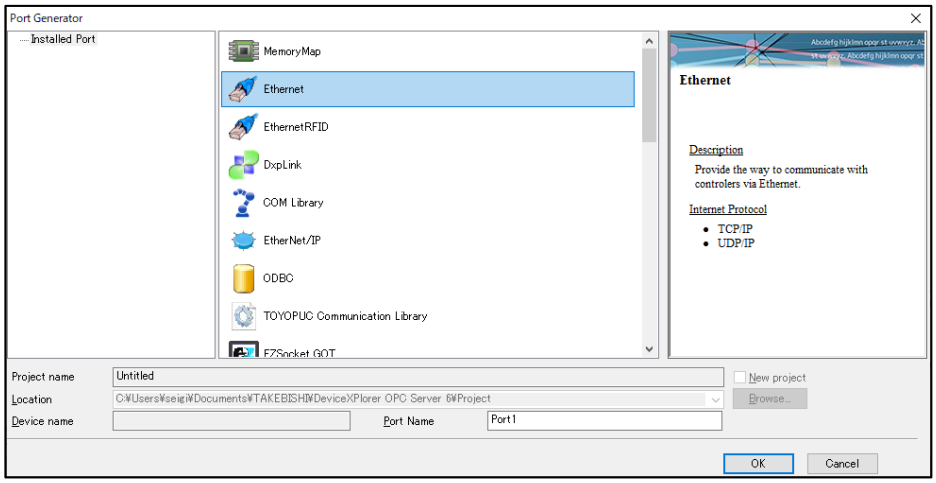
In binary communication, communication is 8 bits / character, so the data length must be set to "8".

5.3.6 Port Generator

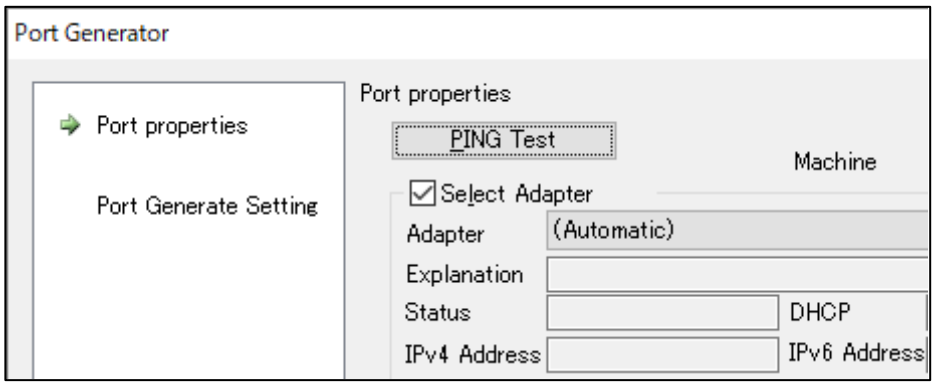
The package generation of the port setup can be carried out.
It can start from a project menu to the Insert ->Port Generator.



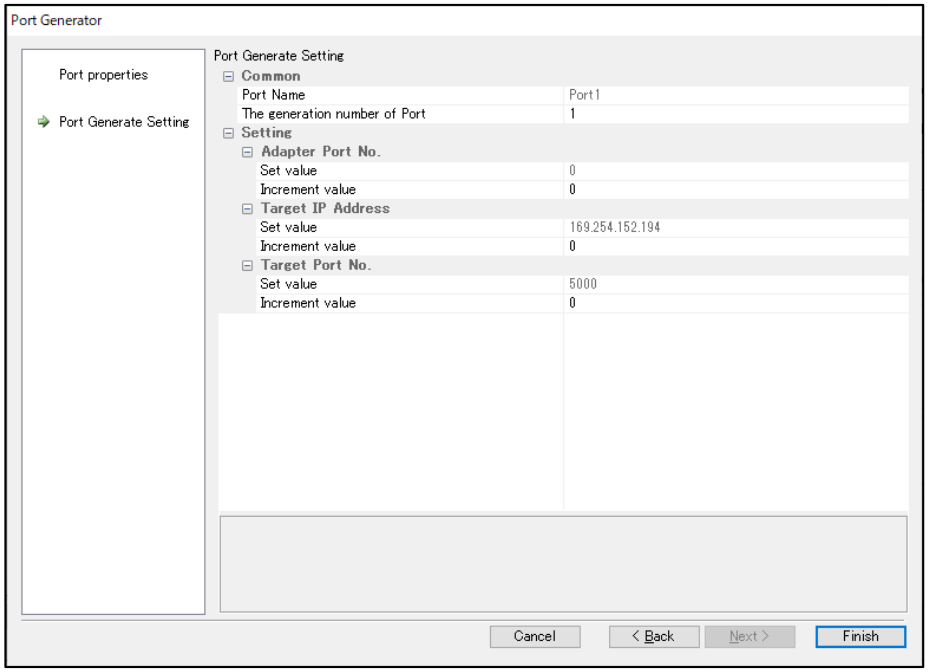
Select the port type and input the Port name.



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



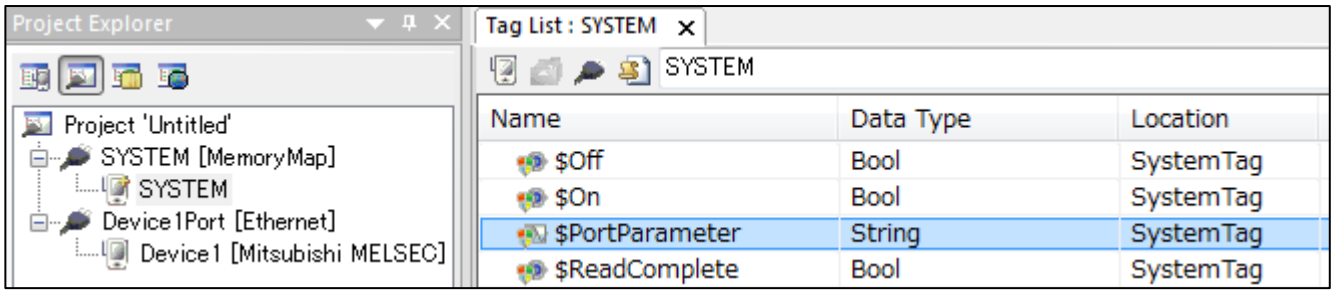
Enter the generation number of Port and Increment value and click “Finish”.



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.

5.3.7 Dynamic Port

To create port dynamic or to change port properties, SYSTEM device has "$PortParameter".

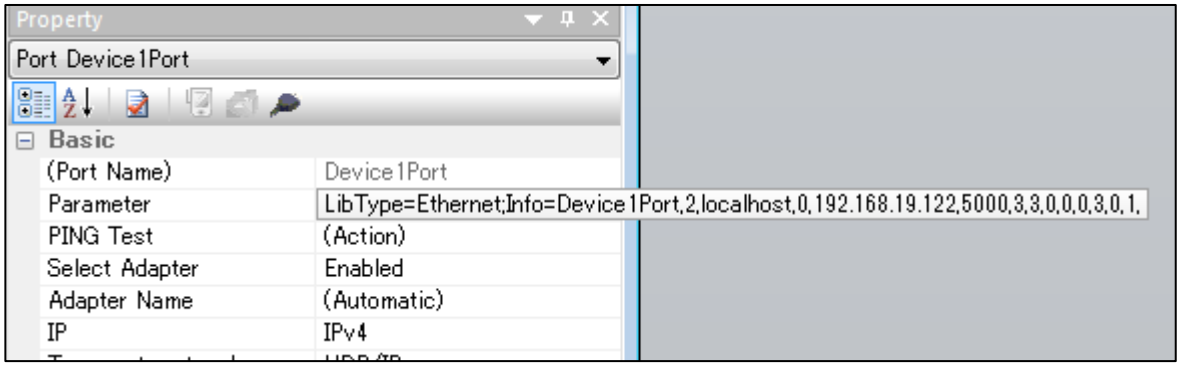


Name	Type	Contents	Data Type	Attribution
$PortParameter	Global System Tag	The parameter system tag for port generation.	STRING	R/W

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

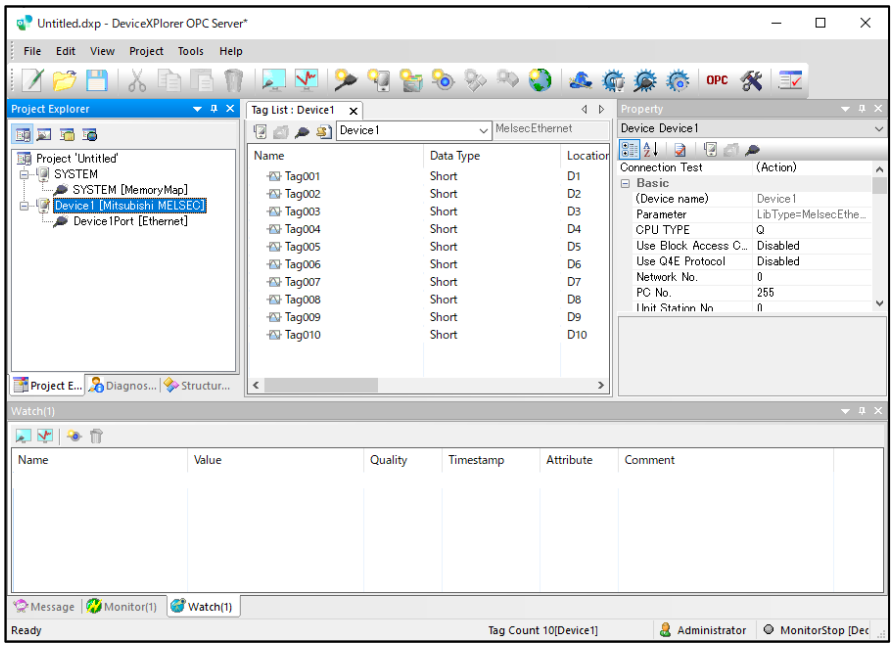
```
LibType=Ethernet;Info=Device1Port,2,localhost,0,192.168.19.122,5000,3,3,0,0,0,3,0,1,
```

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



5.4 Devices

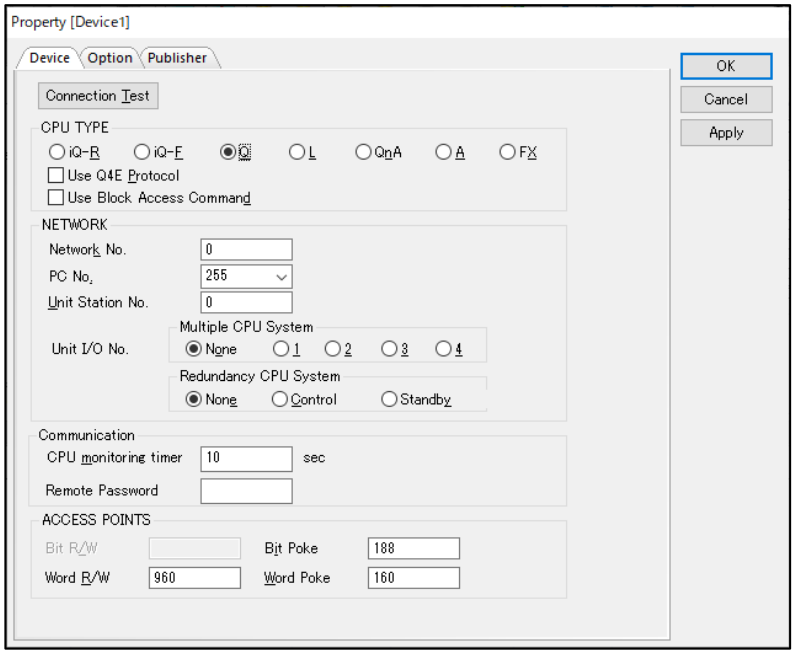
Device has a properties to communicate with target.
Select Project View -> Devices to view device information.



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.

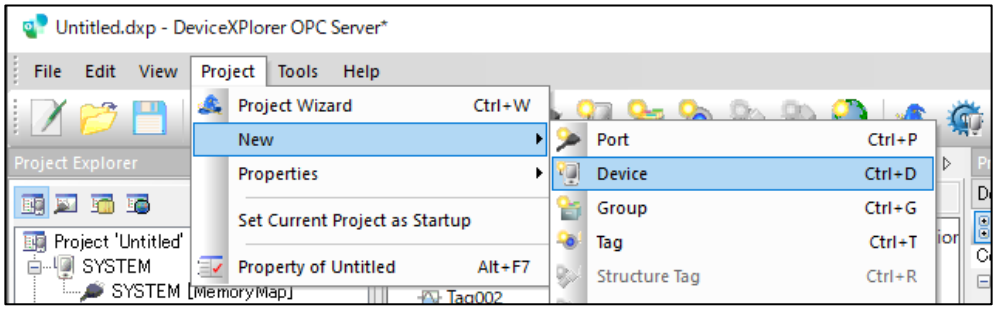


5.4.1 Creating a New Device

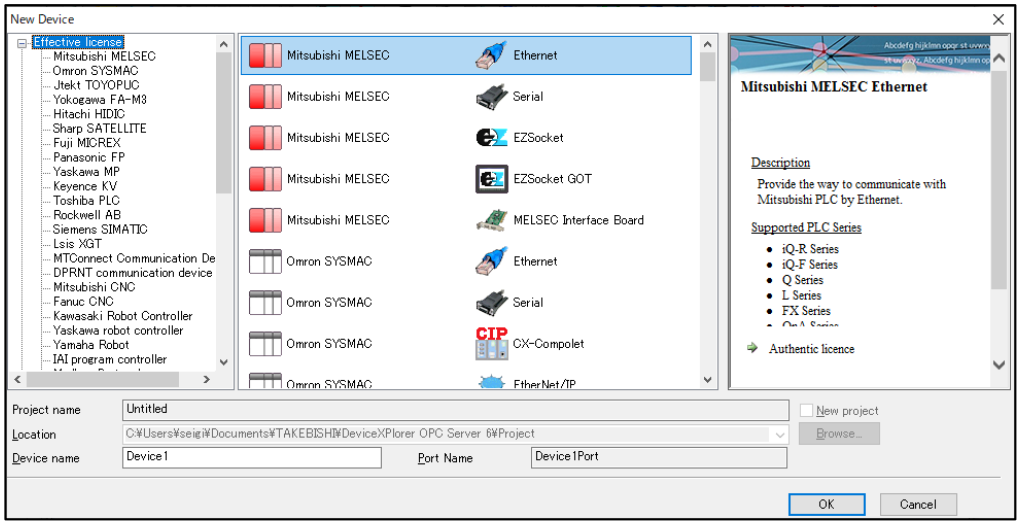
Setup steps differ by the case whether select a port or not.

5.4.1.1 Not select a Port

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

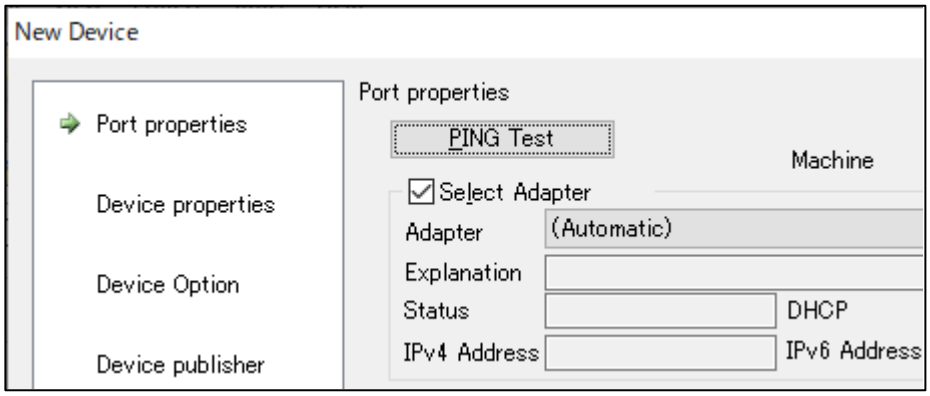


Select the Device and Port type and input Device name and Port name.

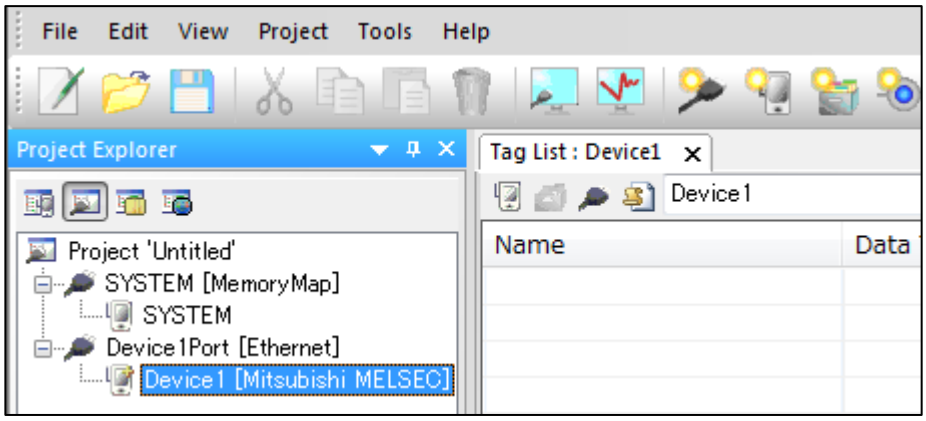


Item	Description
Device Name	Specify Device name for creation.
Port Name	Specify Port name for creation.

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

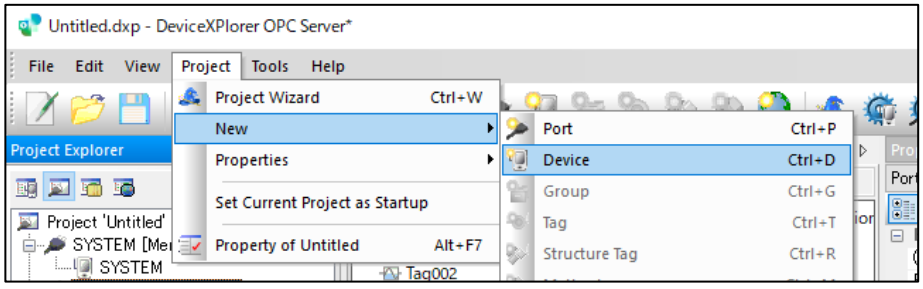


Create Port and Device by completing settings.

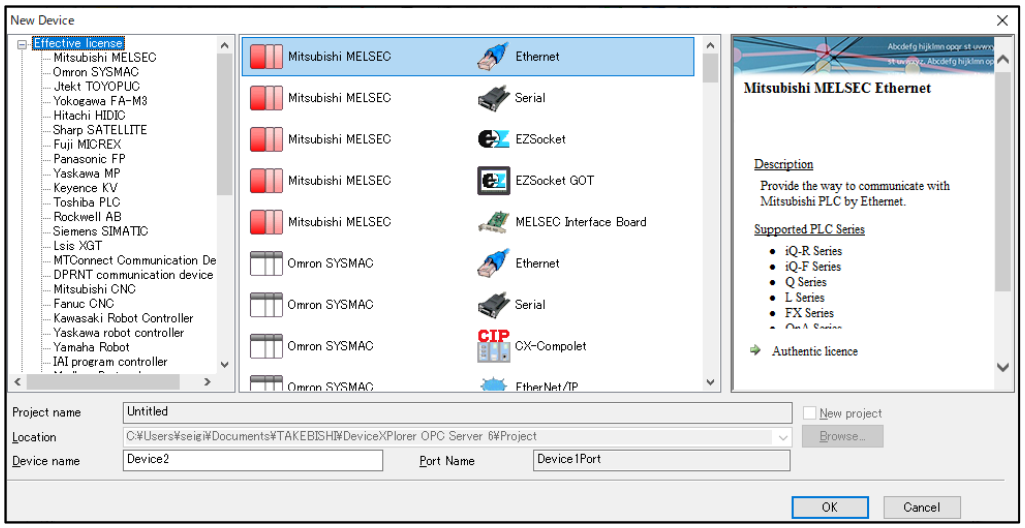


5.4.1.2 Select a Port

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

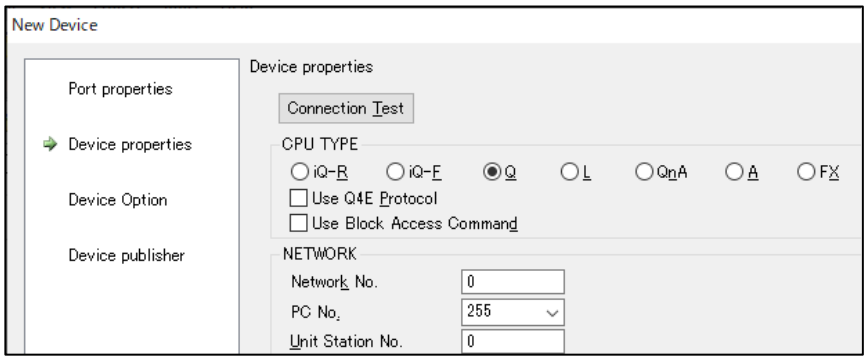


Select the Device type and input Device name.

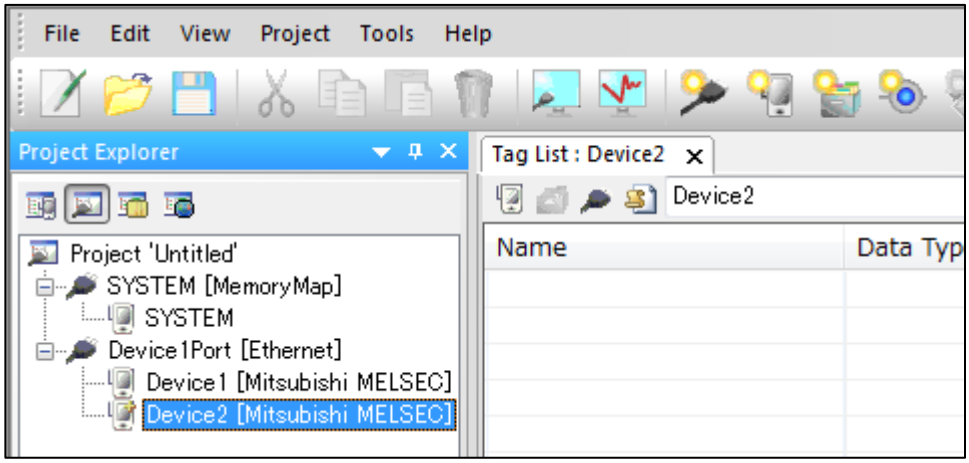


Item	Description
Device Name	Specify Device name for creation.
Port Name	Read-Only.

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

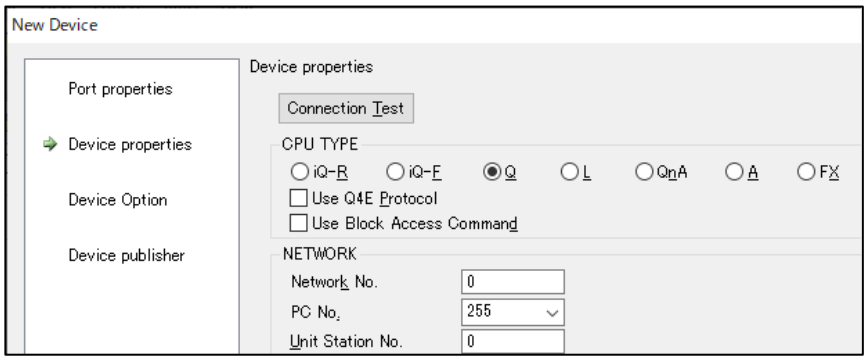


Create Device by completing settings.

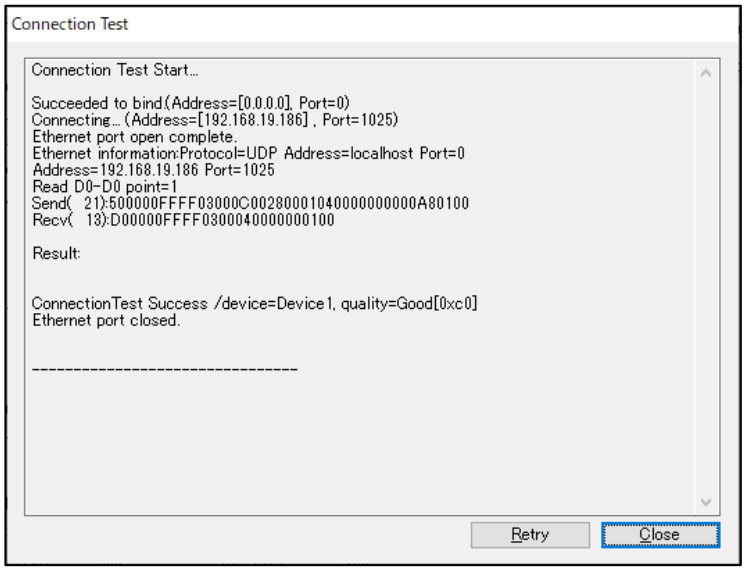


5.4.2 Connection Test

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.



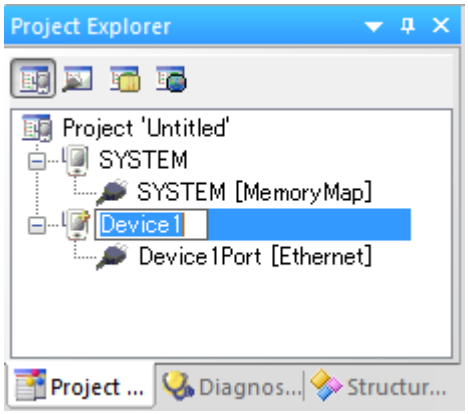
Item	Description
Result	Success: Connection Test Success Result: ConnectionTest Success /device=Device1, quality=Good[0xc0] Ethernet port closed. Error: Connection Test Error Result: ConnectionTest Error /device=Device1, quality=Comm Failure[0x18] Ethernet port closed.
Retry	Retry Connection Test.
Close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

5.4.3 Changing the Device Name

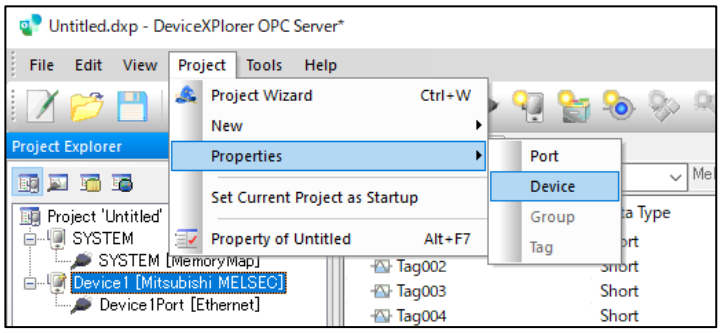
You can change the device name by either clicking on the device name again once it is selected, or by using the F2 key to enable in-place activation.
* You cannot change to a device name that already exists.



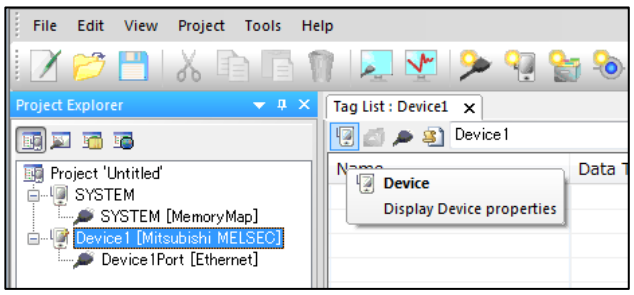
5.4.4 Device Settings

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”.

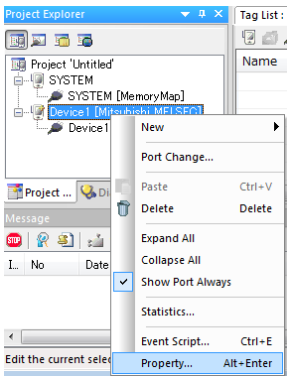
[Project menu]



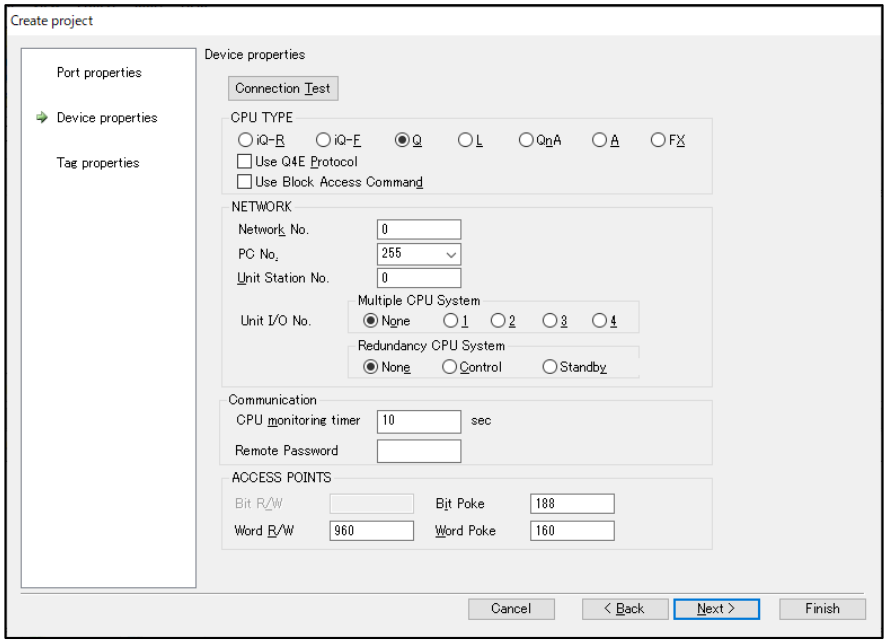
[Port Jump button]



[Right-Click]

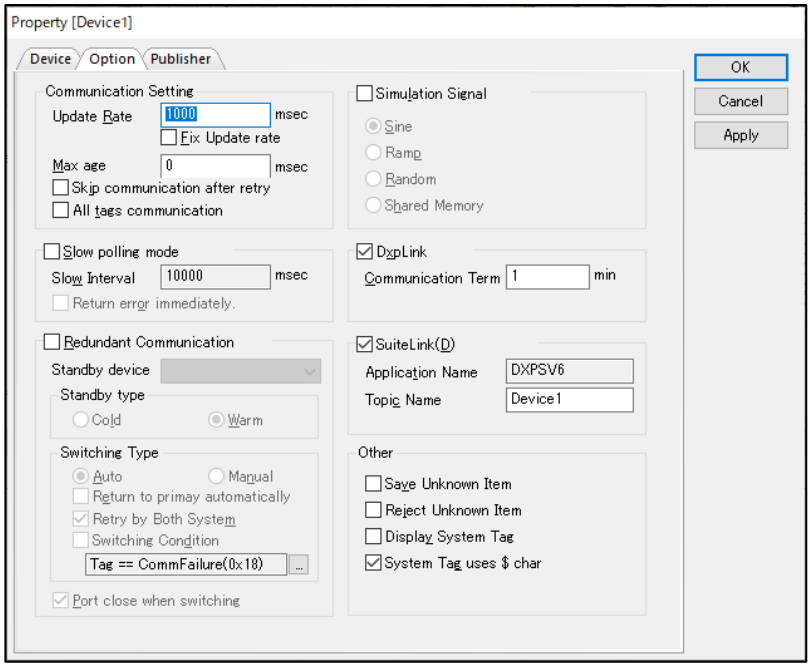


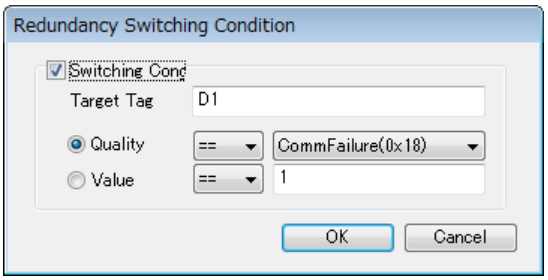
The following screens are displayed in MELSEC Ethernet.
For detail, refer to the "User's Guide (PLC Communication Edition)".



5.4.5 Device Options

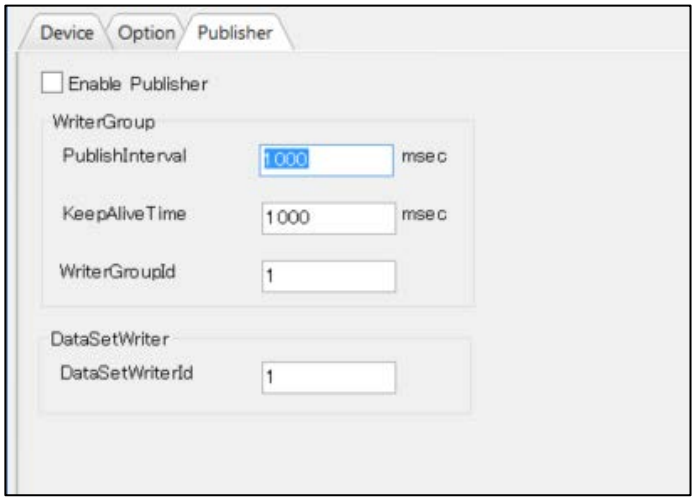
This tab allows you to set proprietary device information that does not depend on the PLC, such as the simulation mode and redundant communication.



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.

5.4.6 Device Publisher

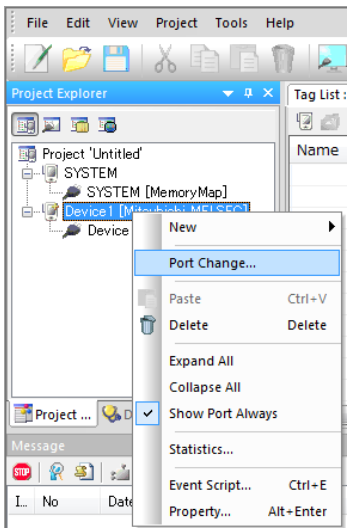


Item		Description
Enable Publisher		Enable publishing.
WriterGroup		-
	PublishInterval	Set the publishInterval. The shortest cycle is 50ms.
	KeepAliveTime	Set the KeepAliveTime.
	WriterGroupId	Set the WriterGroupId. Set WriterGroupId that does not suffer from other devices.
DataSetWriter		-
	DataSetWriterId	Set the DataSetWriterId.

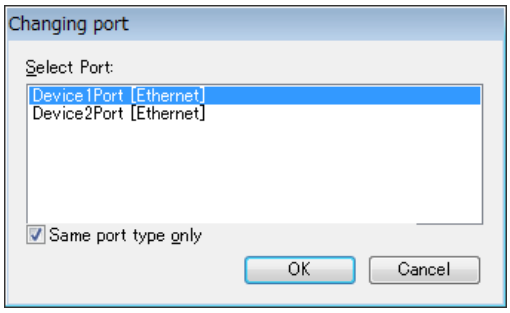
*Please contact us if you wish to use a publisher.

5.4.7 Changing Ports

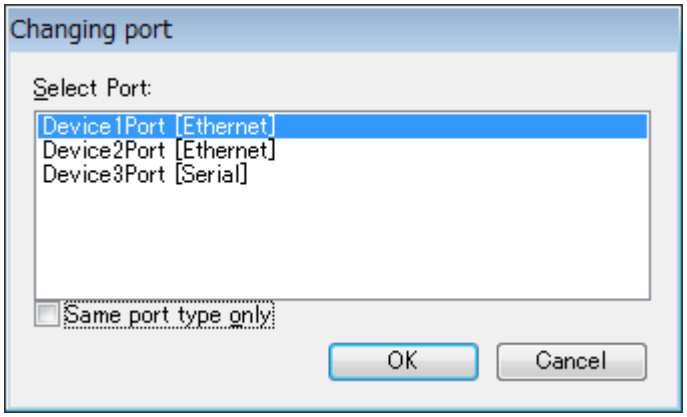
To change a port assigned to a device, select a device node in the Project Explorer, show the pop-up menu, and click “Port Change…”.



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



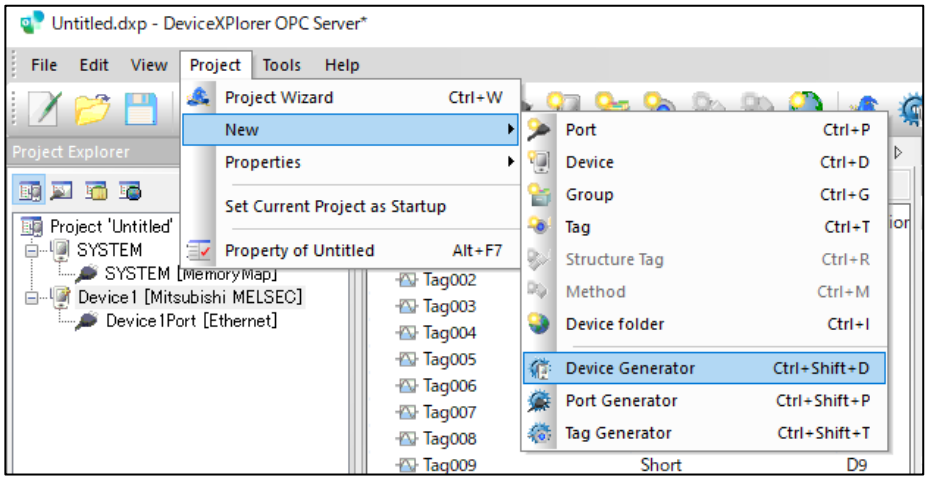
If there is no check in the "Same type of port" checkbox, all the ports will be shown and it will be possible to change to a port of a different type.



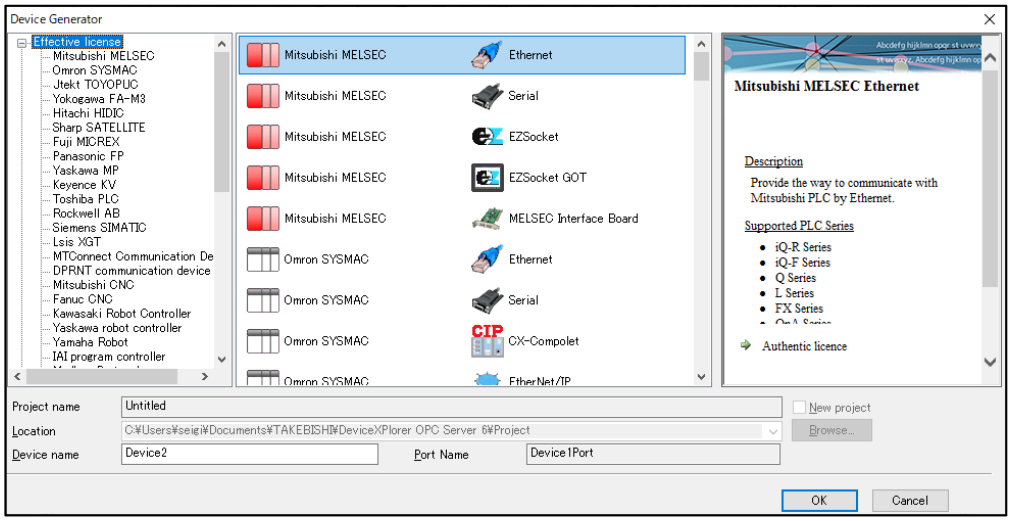
Important
If you change to a port of a different type, the information set for the device and the tag will be lost.

5.4.8 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.



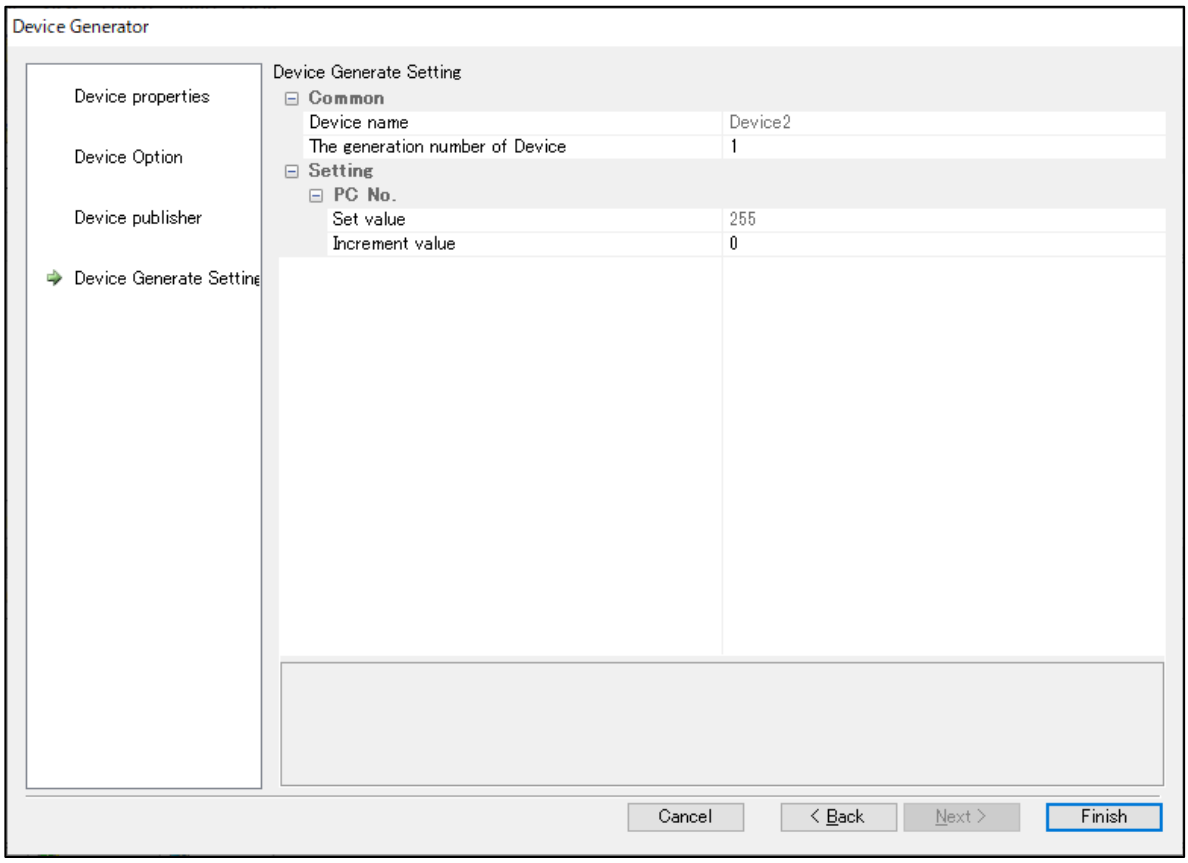
Select the device and port type and input the Device name and Port name.



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)

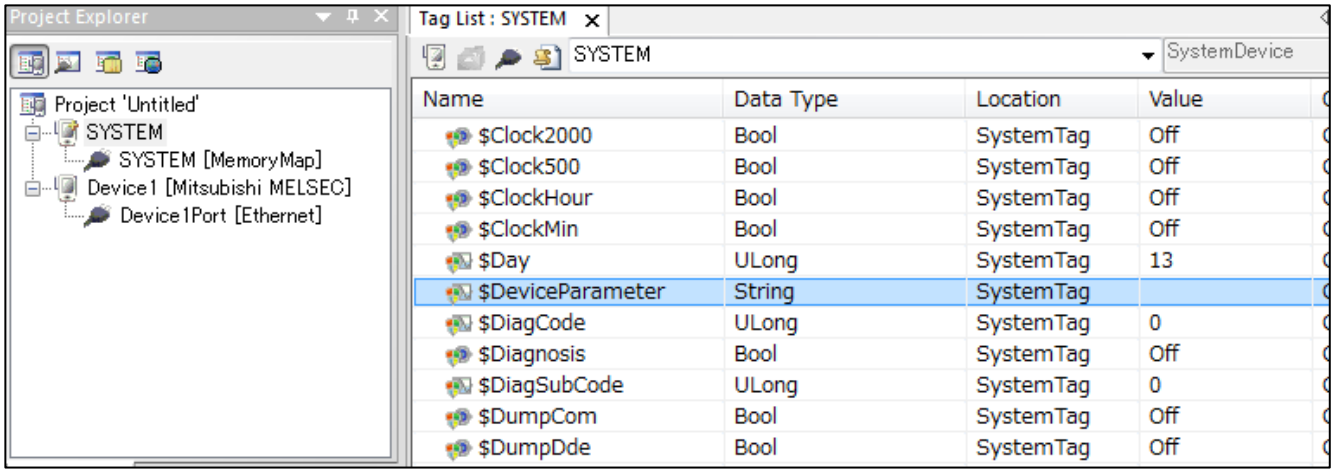
Enter the generation number of Device and Increment value and click “Finish”.



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.

5.4.9 Dynamic Device

To create device dynamic or to change device properties, SYSTEM device has "$DeviceParameter".

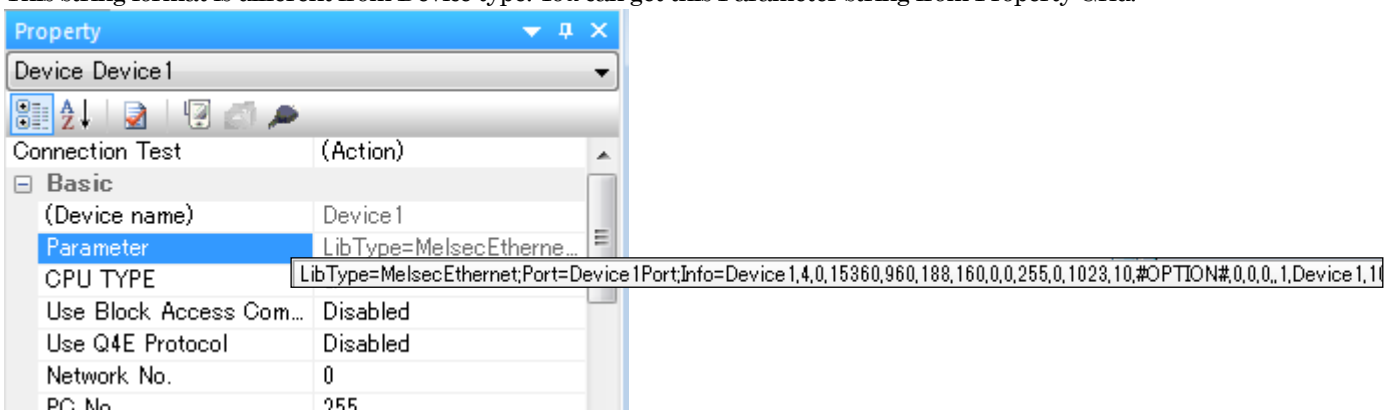


Name	Type	Contents	Data Type	Attribution
$DeviceParameter	Global System Tag	The parameter system tag for device generation.	STRING	R/W

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

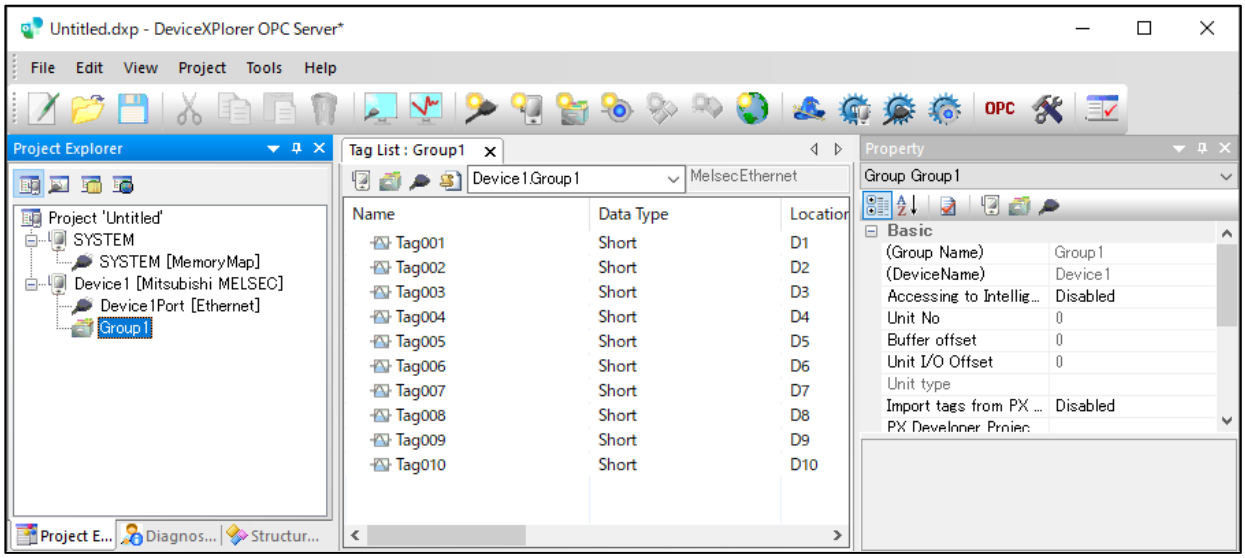
```
LibType=MelsecEthernet;Port=Device1Port;Info=Device1,4,0,15360,960,188,160,0,0,255,0,1023,10,
#OPTION#,0,0,0,,1,Device1,1000,0,0,1,1,0,0,0,0,0,10000,0,1,1,1,1,0,D1,0,0,24,0,1,1,1,1
```

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



5.5 Groups

A Group is a unit which manages tags.
Select Project View -> Groups to view group information.

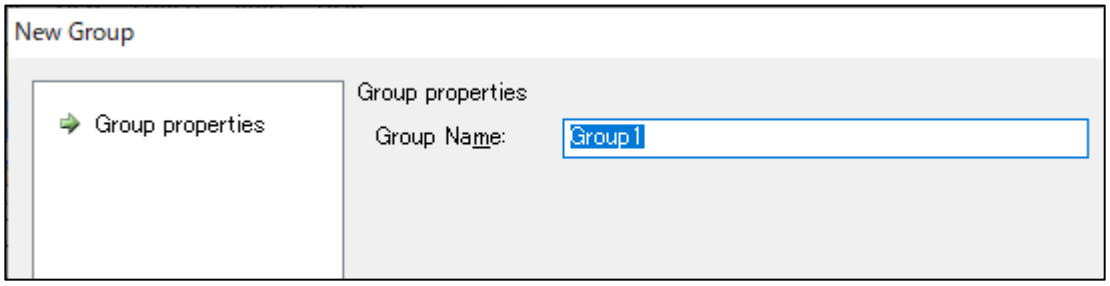


Important

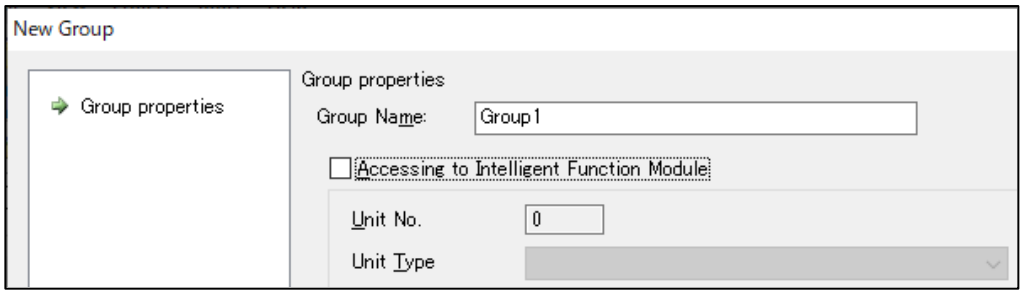
- A Group name is concerned with the name space accessed from a client. As shown in the above, To access Tag001 in Group1 of
Device1, it becomes the name space "Device1.Group1.Tag001."
-There is no restriction on the number of groups.

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)”.

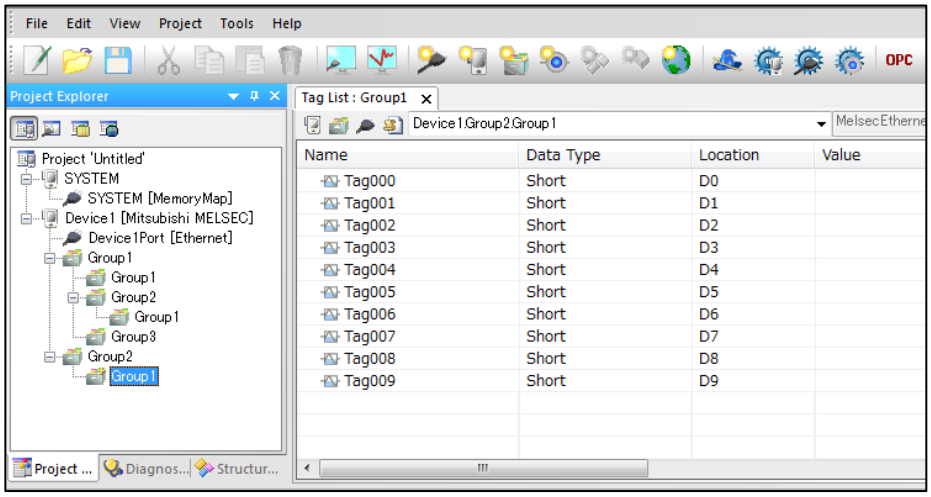
[Only Group name] Shared memory etc.



[Some properties] MelsecEthernet etc.

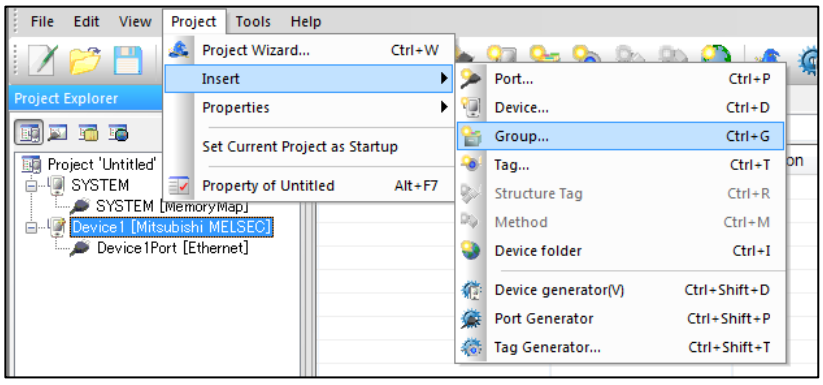


Group can be freely created by a layered structure to a Device subordinate.

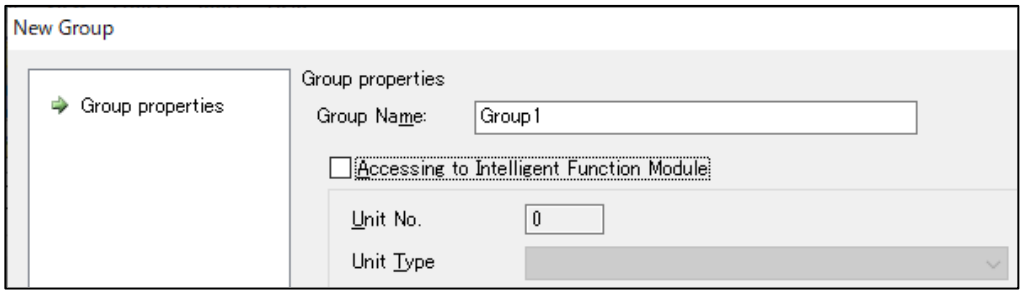


5.5.1 Creating a New Group

From the Projects menu, select Insert-> Group.

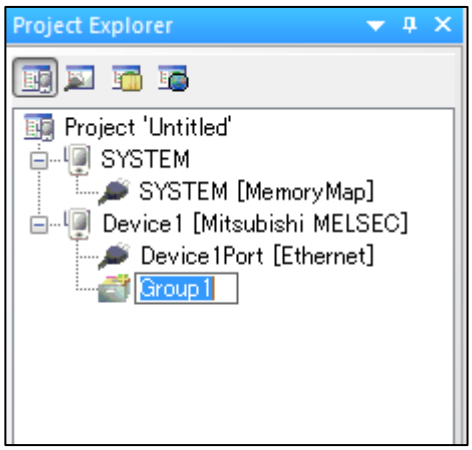


The Group dialog will appear. Enter the Group name and other properties and click “Finish”.



5.5.2 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.

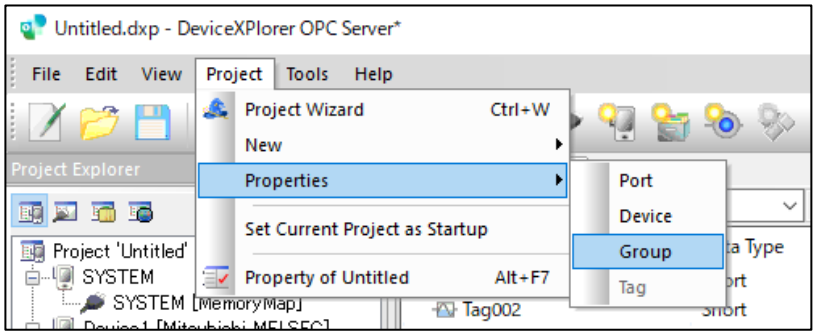


Note
It is necessary to set a name that is not duplicated within the same layer.

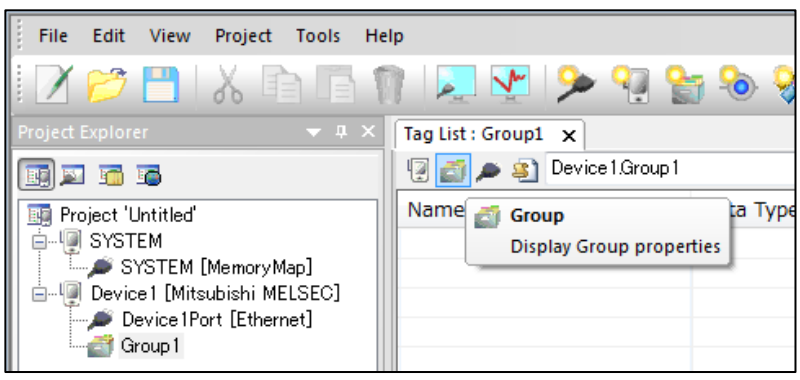
5.5.3 Group Settings

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".

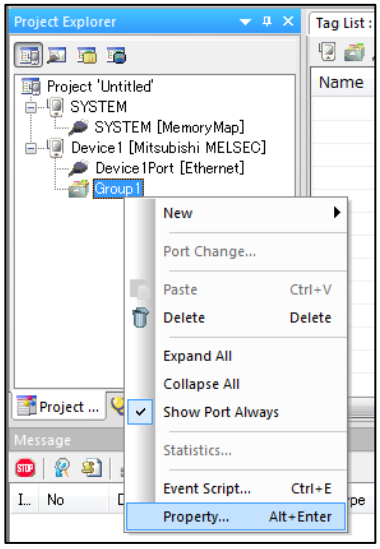
[Project menu]



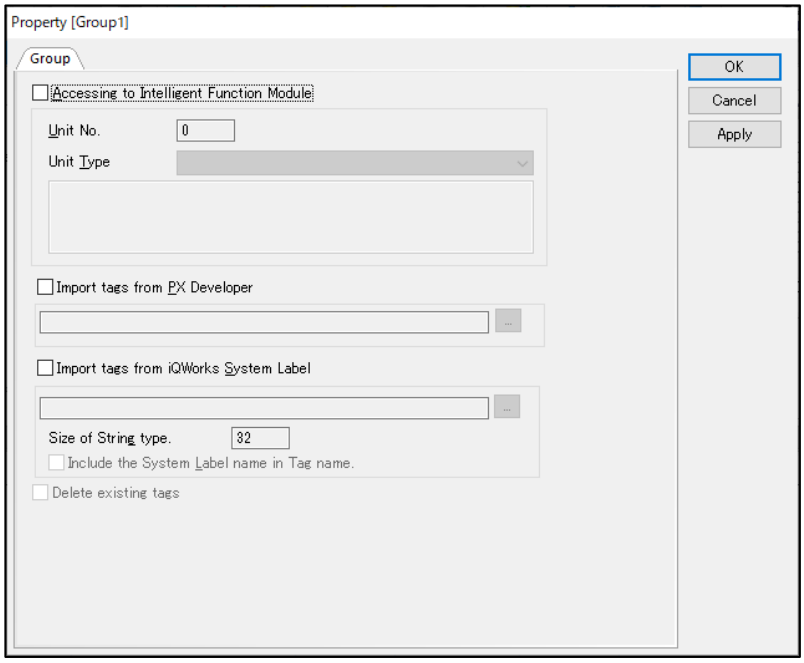
[Group Jump button]



[Right Click]



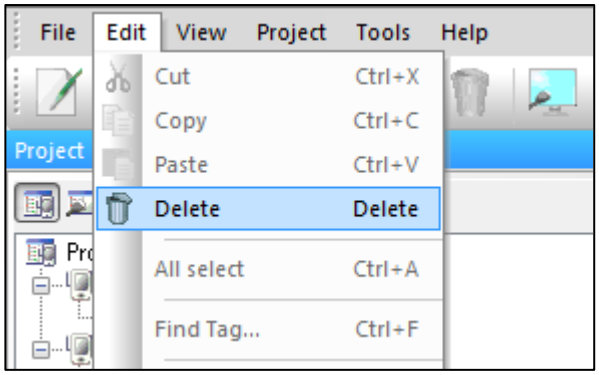
The group dialog will appear. Enter the communication parameters and click “OK”.
For detail, refer to the “User’s Guide (PLC Communication Edition)”.

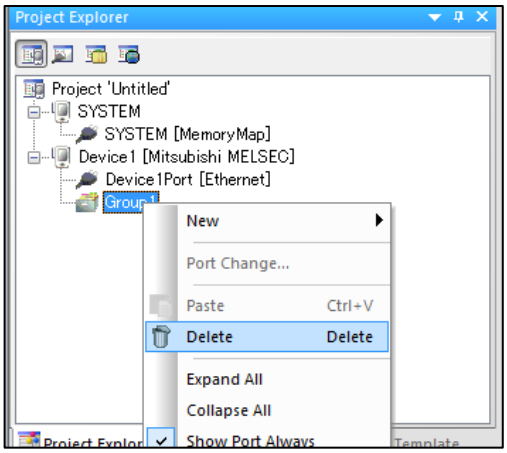


5.5.4 Deleting Group

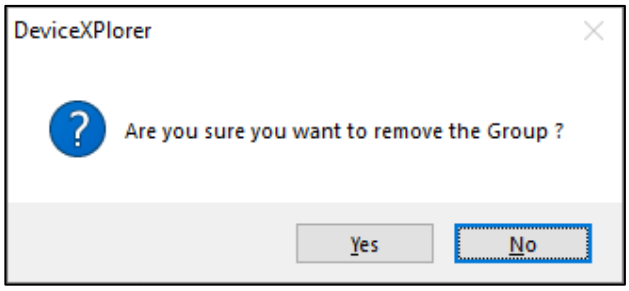
To delete Group

- You select a Group name 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 group.

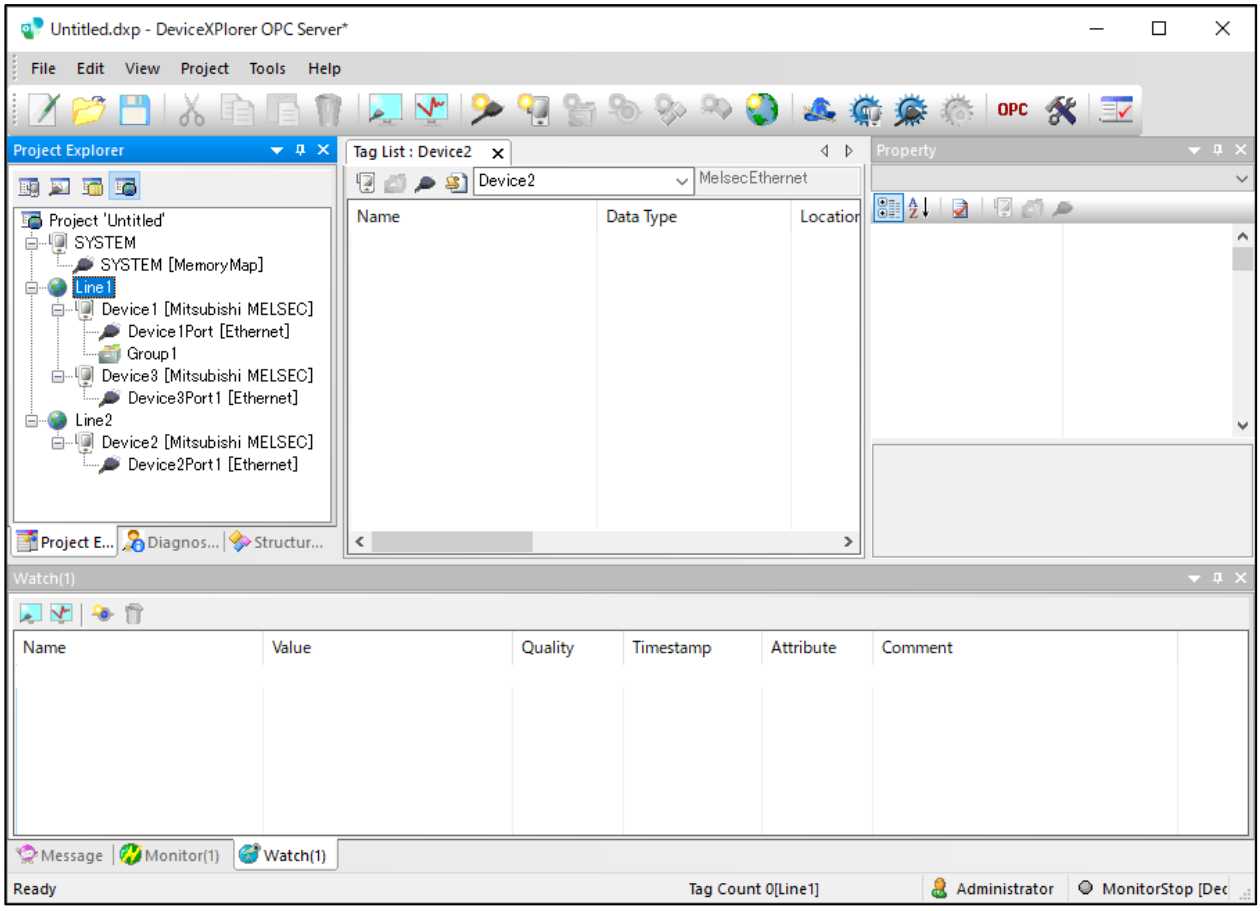


Important

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

5.6 Device Folder

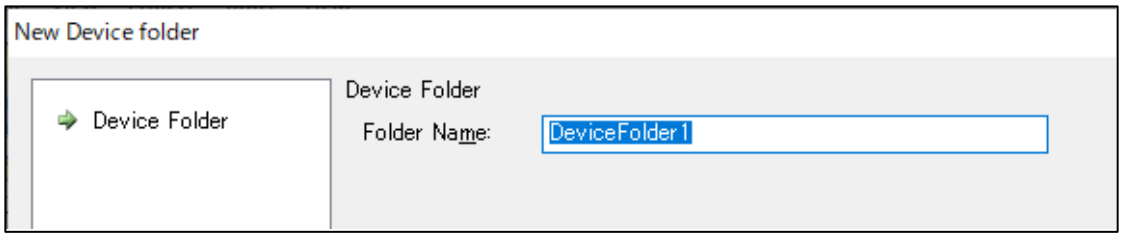
A Device Folder is a unit which manages devices.



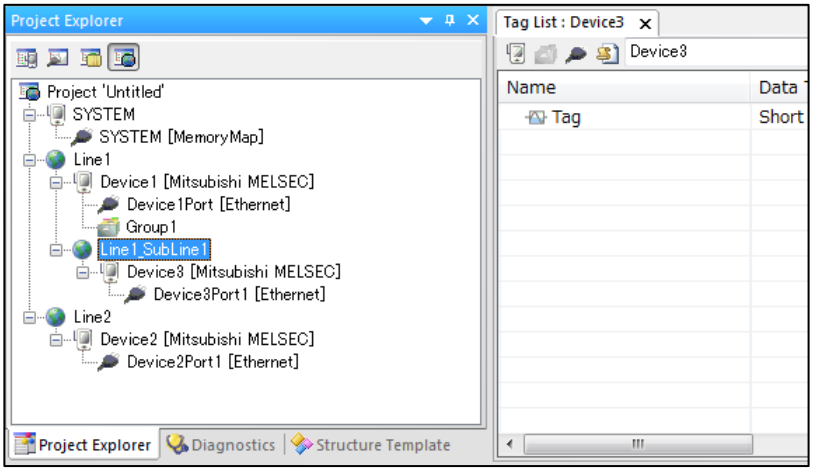
Important

- 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.

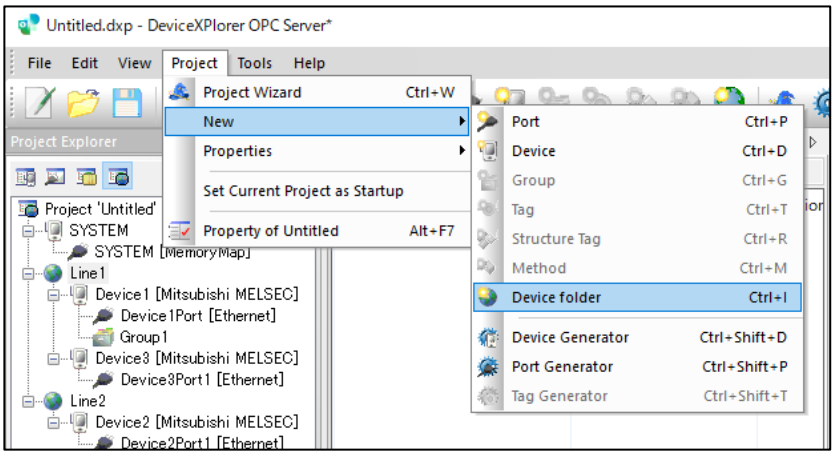


Important

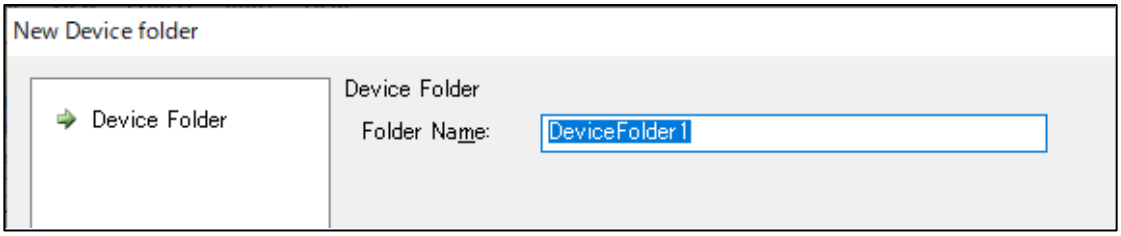
-There is no restriction on the number of Device Folders.

5.6.1 Creating a New Device Folder

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

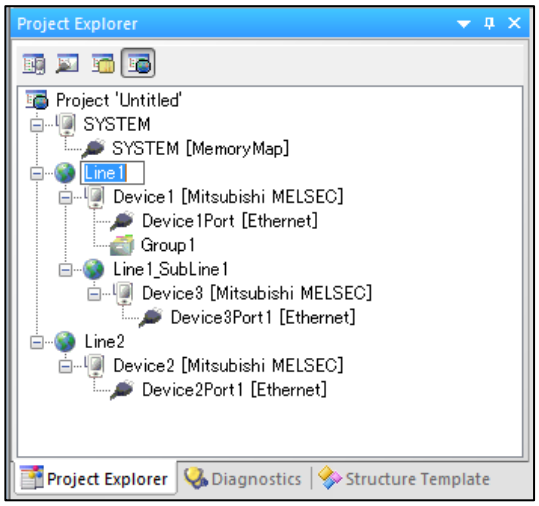


The Device Folder dialog will appear. Enter the Device Folder name and click “Finish”.



5.6.2 Changing the Group Name

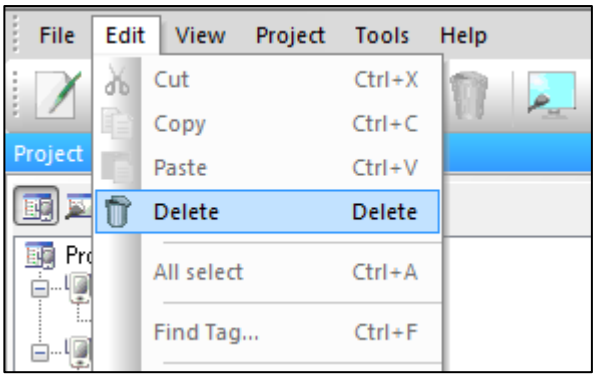
You can change the group name by either clicking on the group name again once it is selected, or by using the F2 key to enable in-place activation.

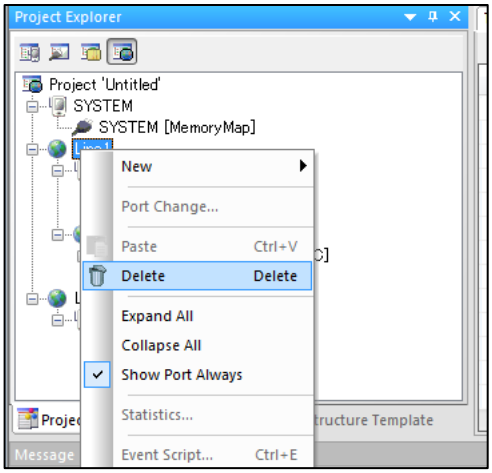


Note
It is necessary to set a name that is not duplicated within the same layer.

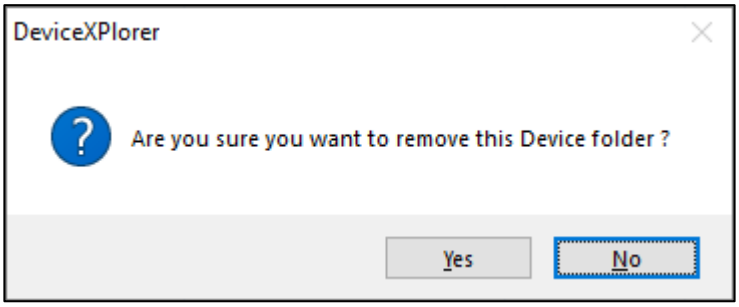
5.6.3 Deleting Device Folder

To delete Device Folder
- You select a Device Folder in Project Explore and select Edit->Delete.
- You Right-Click and select “Delete”.





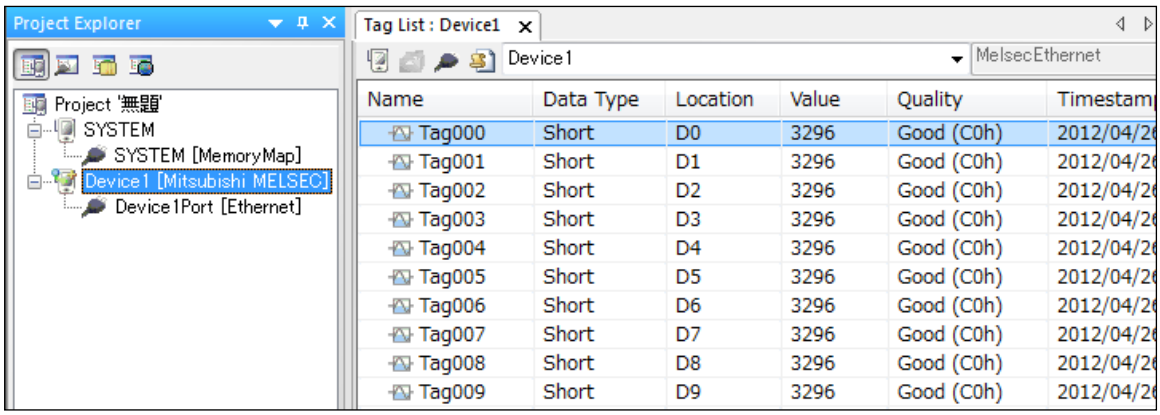
If you select delete, DeviceXPlorer delete the group and tags which exist under the Device Folder.



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

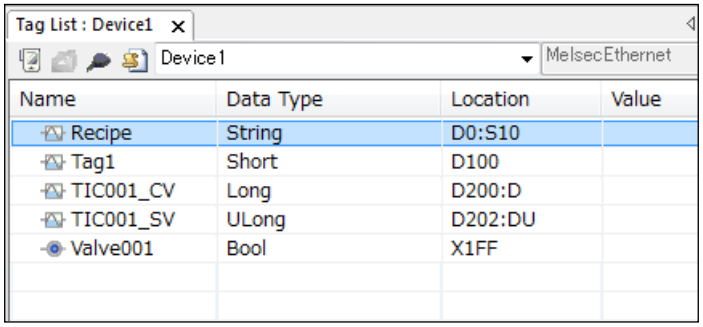
5.7 Tags

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

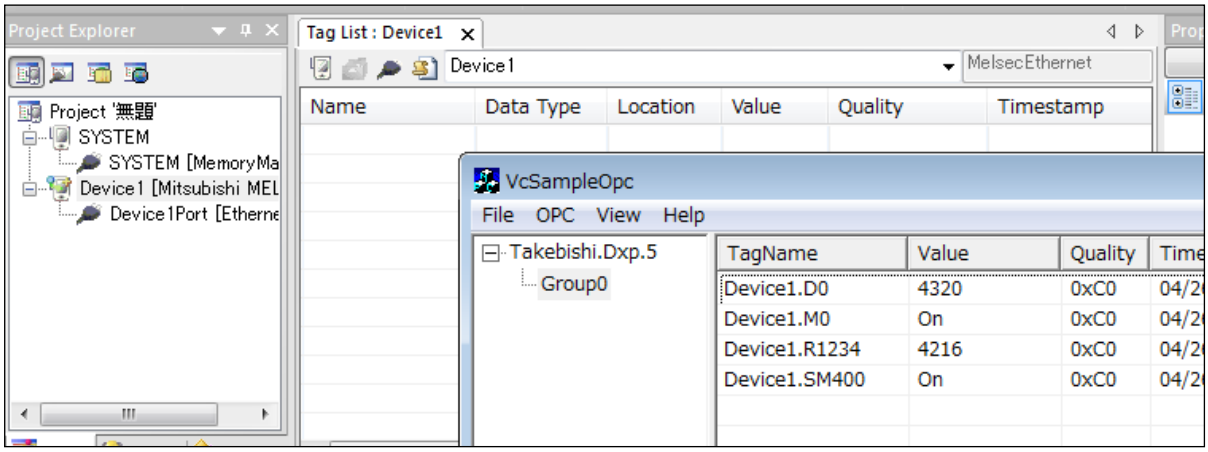


5.7.1 Dynamic Tags and Static Tags

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.



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”.

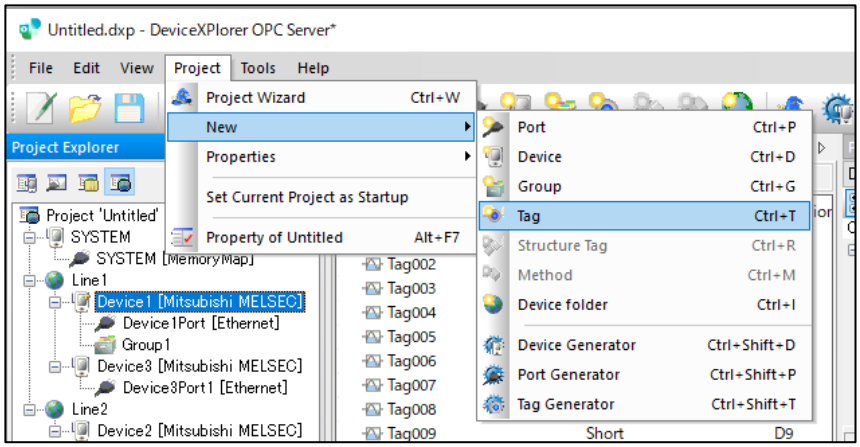


Important

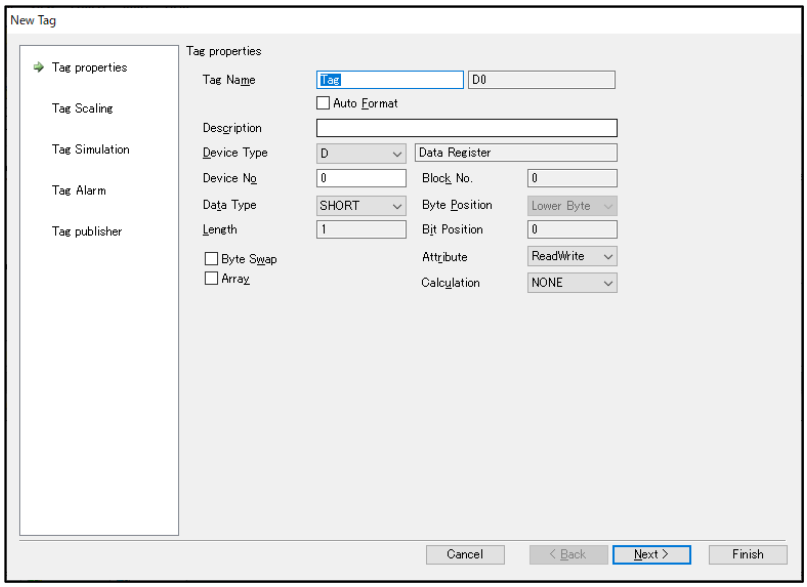
DeviceXPlorer has no restriction on the number of tags, so it is necessary to ensure that the CPU performance and memory are high enough to accommodate the system (number of tags) you are using. If the number of PLCs is more than 50 or if the total number of tags is over 100,000, we recommend that you distribute the load to the DeviceXPlorer across several PCs.

5.7.2 Creating a New Tag

From the Projects menu, select New -> Tag.

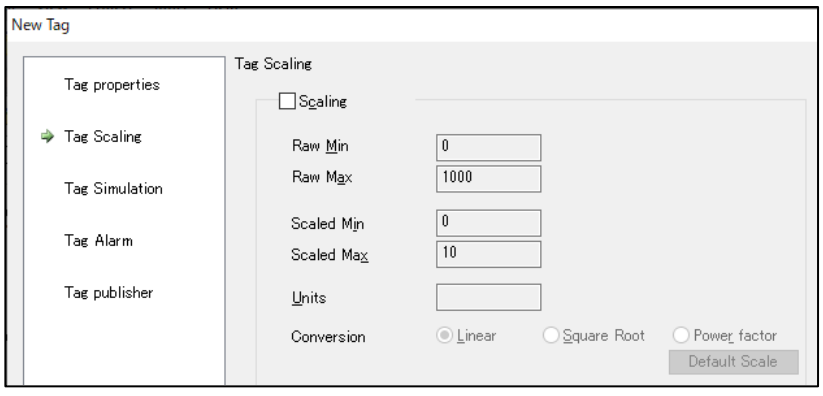


Set the tag parameters and click “Next”.

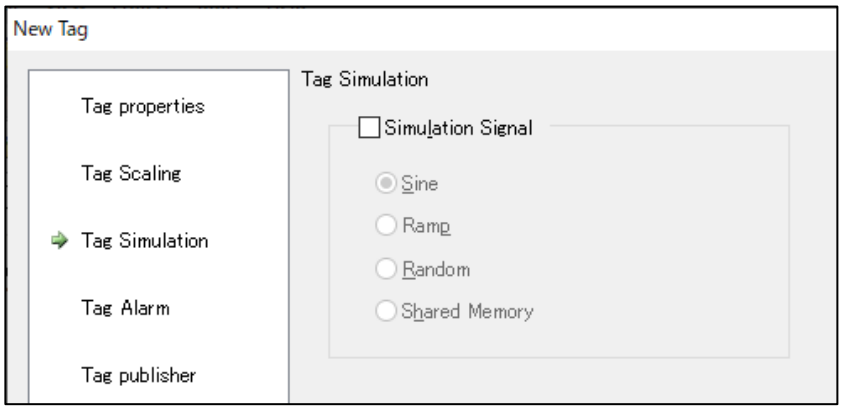


To perform scale conversion, place a check in the box for “Scale Conversion” ,“Simulation” and “Alarm “set the parameters.

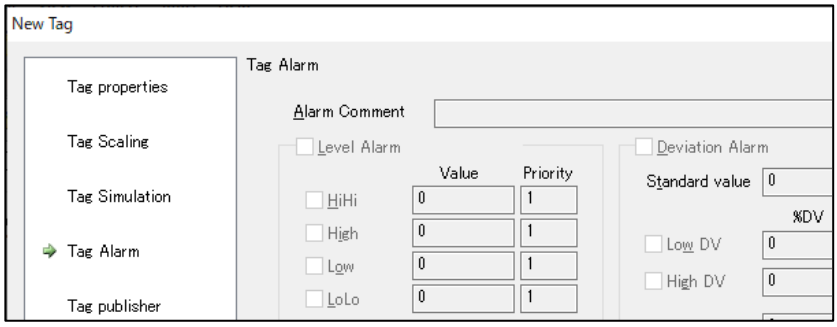
< Scale Conversion>



<Simulation>

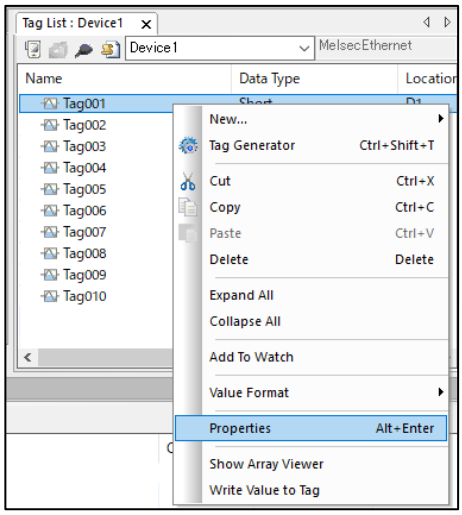


<Alarm>

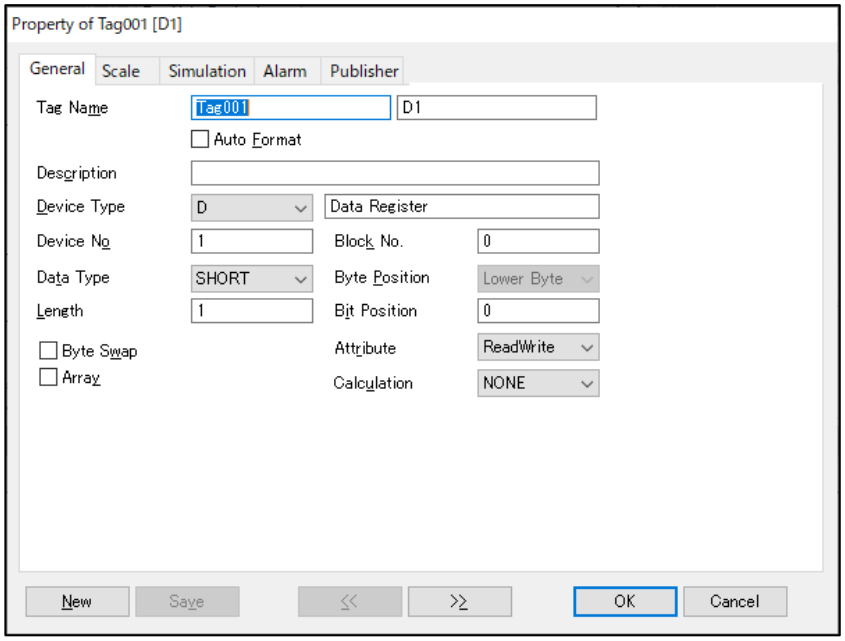


5.7.3 Changing a Tag Name

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



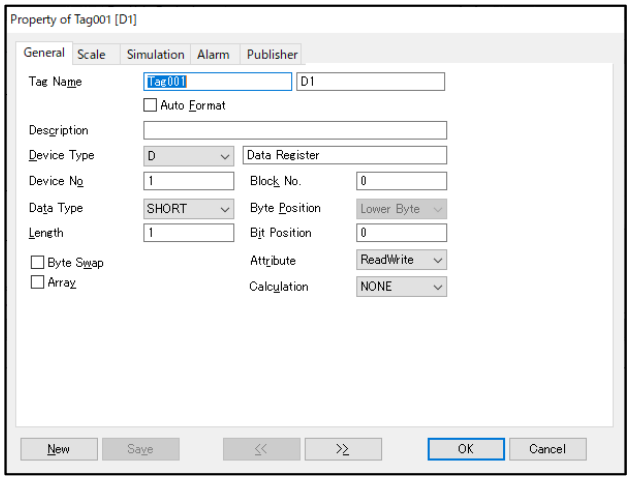
This will allow you to change the set tag name to a name of your choice.



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

5.7.4 Properties (General)

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



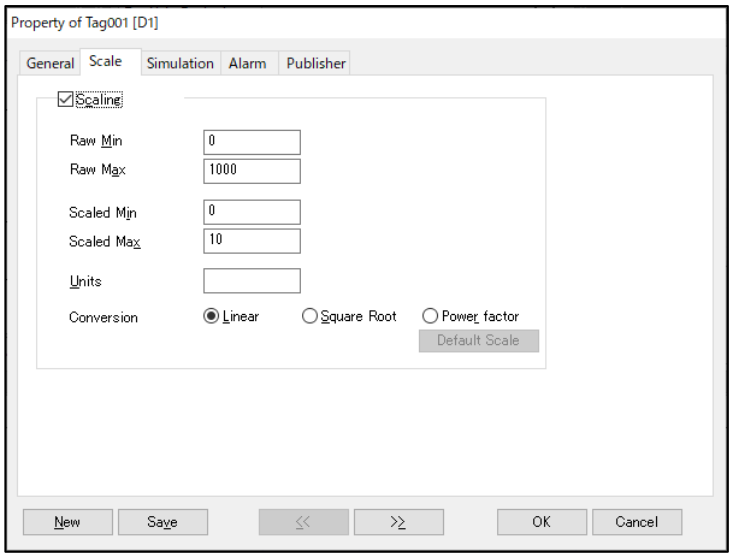
The dialog settings for MELSEC Ethernet are as shown in the table below.
The settings vary for each PLC; for details, see the User's Guide (PLC Communication Edition).

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.

5.7.5 Properties (Scale Settings)

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



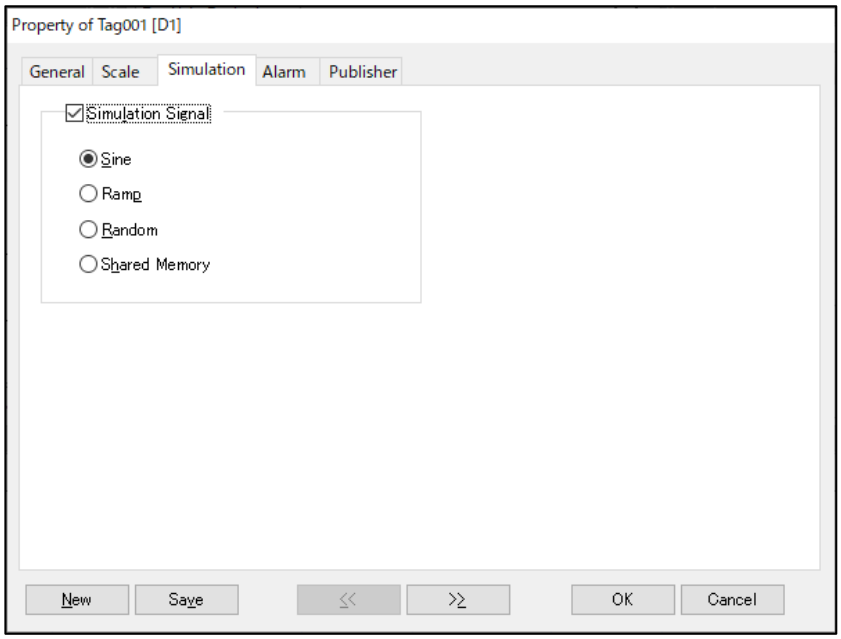
Item	Description
Enable scale conversion	Place a check in this box to implement scale conversion processing.
Pre-conversion – Min. (M)	Enter the minimum value of the value range that can be obtained from the PLC.
Pre-conversion – Max. (A)	Enter the maximum value of the value range that can be obtained 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).

5.7.6 Properties (Simulation)

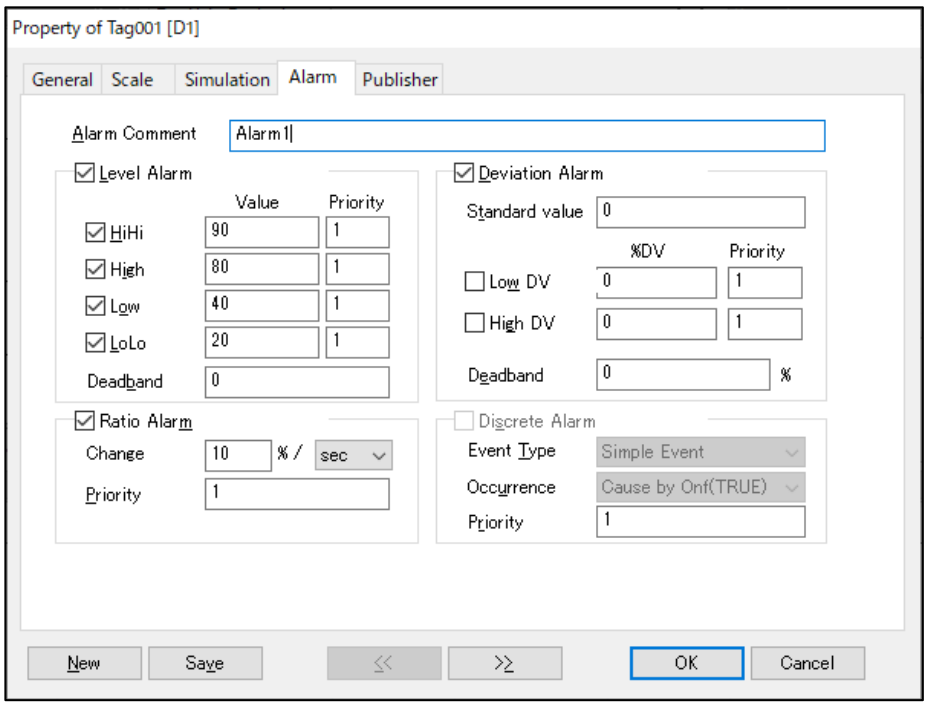
Enable simulation settings if you want to return values of your choice without communicating with the PLC.



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.

5.7.7 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.



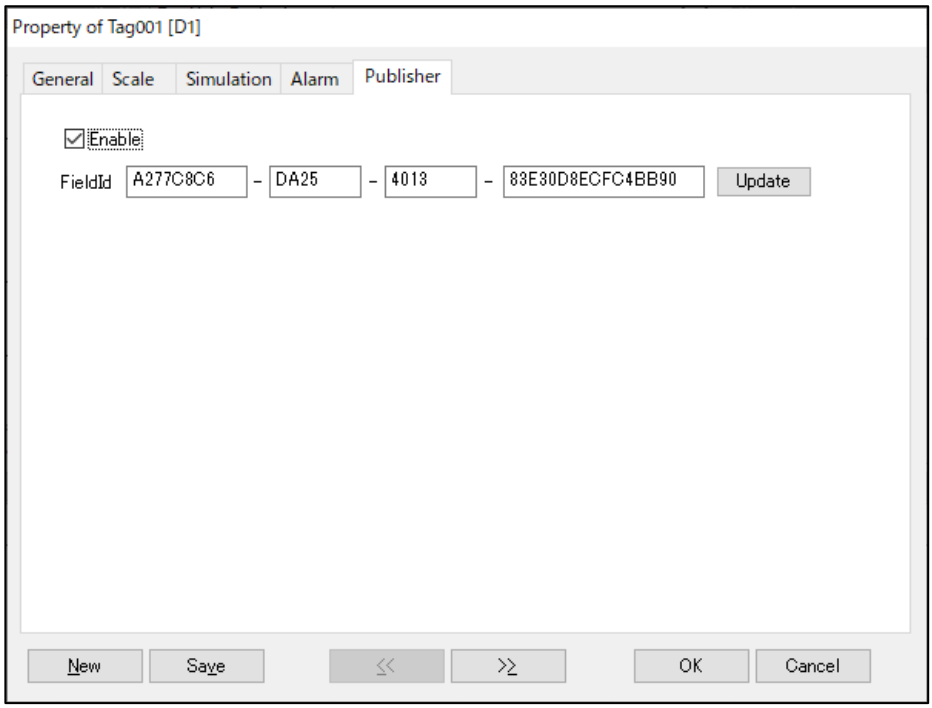
Item		Description
Alarm Comment		Specify the Alarm comment.
Level Alarm	LoLo/Low/High/HiHi	Select Enable or Disable.
	Value	Specify the threshold value.
	Priority	Specify the priority.
	Deadband	Set the deadband of level Alarm. If a current value returns more than deadband, the alarm restores.
Ratio 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 .

5.7.8 Properties (Publisher)

Specify the Publisher setting.



Item	Description
Enable	Select Enable or Disable.
FieldId	The subscriber specifies this FieldId and gets the value. Click the Update button if you need to change it.

5.7.9 Saving Settings

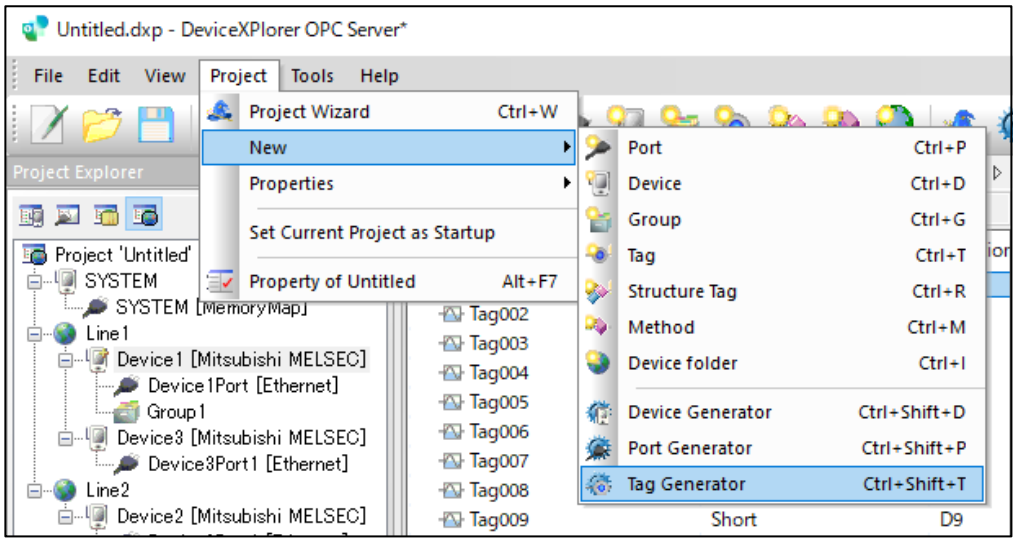
Save settings by clicking the "Save" button or clicking the "OK" button and closing the dialog.

5.7.10 Adding Tags Continuously

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

5.7.11 Tag Generator

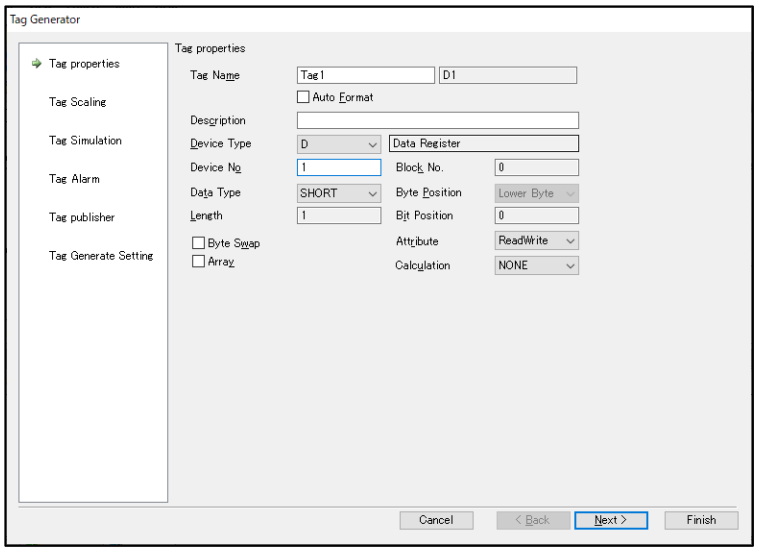
Using the tag generator allows you to generate tags of a particular format at once.



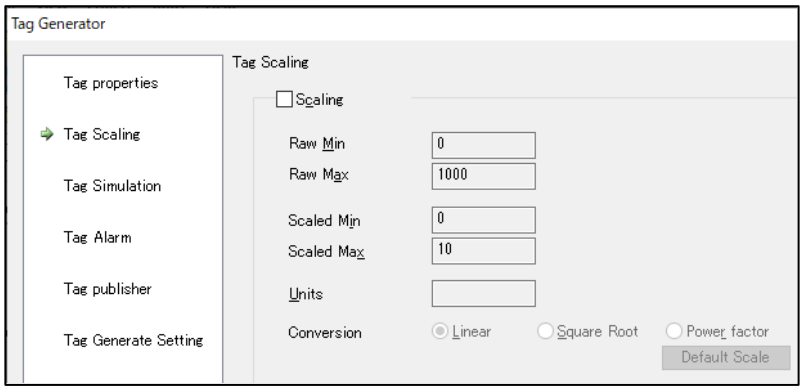
Note
The number included in the tag name and the device number are automatically incremented.

[Operation]

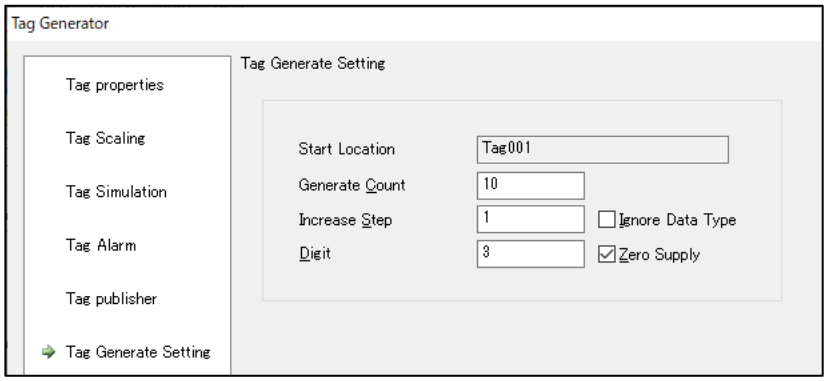
(1) Set the base tag information.



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



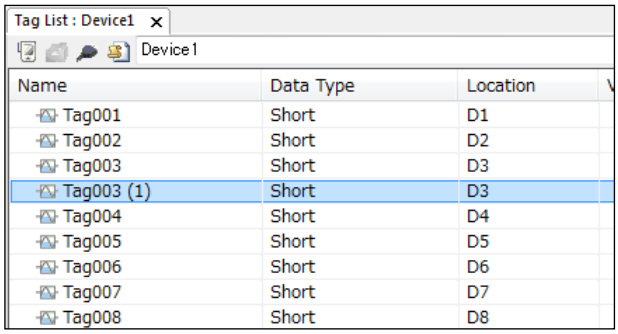
(3) Set the number of tags to be generated at once and the increase step.



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.



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

5.7.12 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 = “ABCDEFGHIJ”
: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”.

5.7.12.1 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".

5.7.12.2 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.

5.7.12.3 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.

5.7.12.4 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.

5.7.12.5 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.

5.7.12.6 LongLong Type (:I, :IU)

The register value is treated as a 64-bit signed or unsigned integer.
Example: In "D1:I", D1 – D4 are accessed as a 64-bit signed integer.

5.7.12.7 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".

5.7.12.8 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".

5.7.12.9 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.

5.7.12.10 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.

5.7.12.11 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.

5.7.12.12 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 “ABCDEFGHIJ” 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.

5.7.12.13 File Batch Write(:F999999)

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.

5.7.12.14 Text Format Binary Type (:B99999)

If you specify “B:99999” 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.

5.7.12.15 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.

5.7.12.16 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.

5.7.12.17 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

5.7.12.18 Bit Inverse

If you specify ":V" after the tag name, DeviceXPlorer inverse Boolean value.
Example:If M0="False", accessing "M0:V" obtains the "True".

5.7.12.19 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).

5.7.12.20 Simulation mode (Sin)

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

5.7.12.21 Simulation mode (Shared Memory)

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

5.7.12.22 Simulation mode (Ramp)

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

5.7.12.23 Simulation mode (Random)

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

5.7.13 System Tags

5.7.13.1 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	Attribution	Note
Version information	$Version	VT_BSTR	R	Ex: "Version 4.0.0.1"
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 (ERROR)	$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	$PortParamter	VT_BSTR	R/W	Dynamic Port. Refer to 5.3.7.
Device Parameter	$DeviceParameter	VT_BSTR	R/W	Dynamic Device. Refer to 5.4.9.
Reset Statistical Information	$ResetStatisticalTags	VT_BOOL	R/W	By writing TRUE, DeviceXPlorer resets the follow system tags. $ElapseMin,$ElapseMax,$Elapse,$Error Counts,$SendCount,$RecievedCounts

5.7.13.2 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	Attribution	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”. Refer to 5.4.2
Primary device Communication Status	$PrimaryStatus	VT_BOOL	R/W	Primary device Communicating: TRUE, Stopped: FALSE
Standby device	$SecondaryStatus	VT_BOOL	R/W	Standby device

Communication Status				Communicating: TRUE, Stopped: FALSE

5.7.13.3 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	Attribution	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

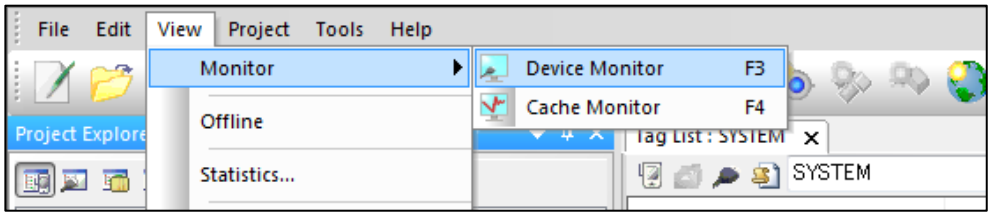
In the time of client unit communication invalidity (default configuration), only the following 4 connection tags can be available.

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

5.7.14 Tag Monitor

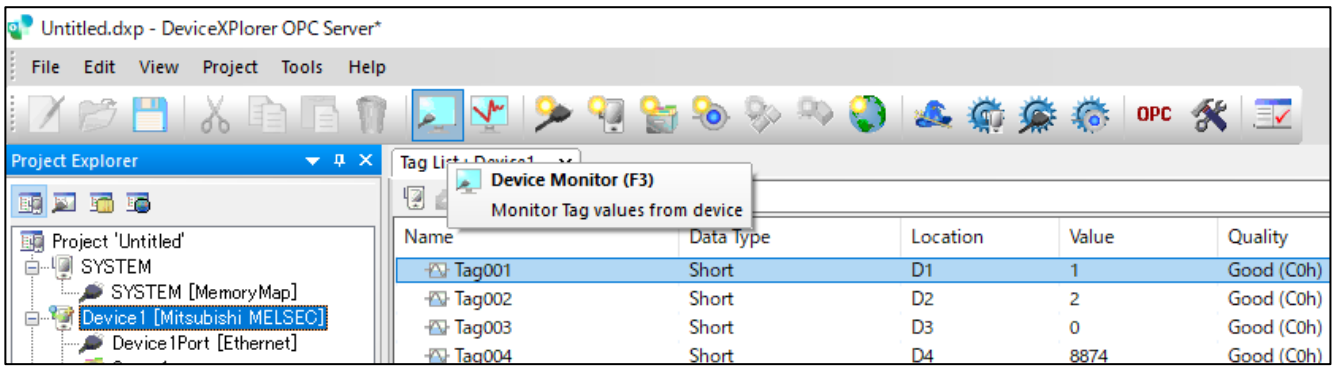
The current value of static tags can be monitored from the DeviceXPlorer screen regardless of whether or not there is access from a client.

[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.

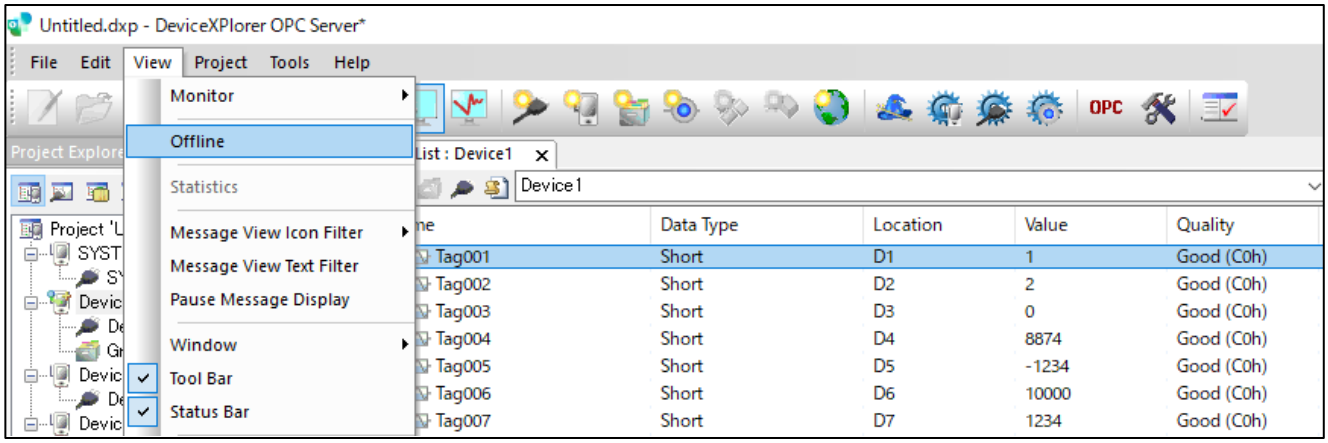
When tag monitoring is performed, the current values and information such as the quality flags appear on the screen.



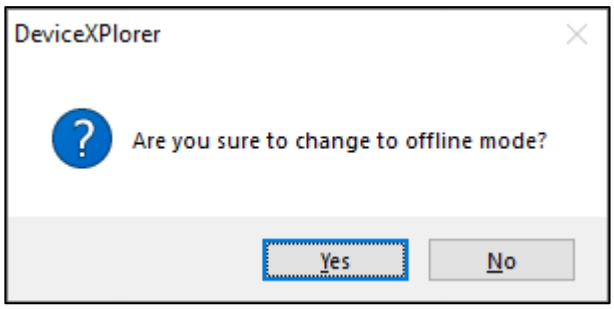
Note
The update interval when the device monitor function is used is value of Monitor Rate in "Common Properties, General".

5.7.15 Offline Mode

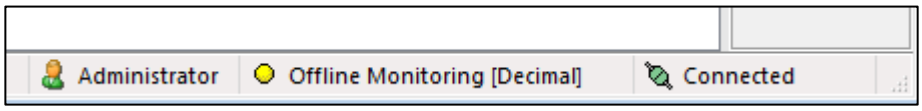
If DeviceXPlorer shifts to off-line mode, it will stop performing communication with target modules entirely, and will process with the cache value which it has in the inside.
If DeviceXPlorer shifts to this mode, the processing to a client etc. will also be affected.



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



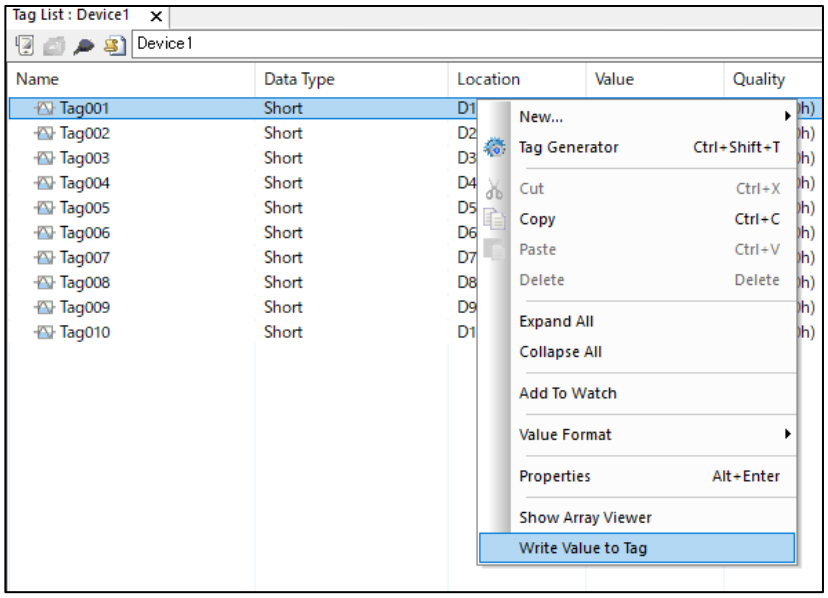
During Offline Mode execution, it is displayed on a lower right status bar as “Offline Monitoring”.



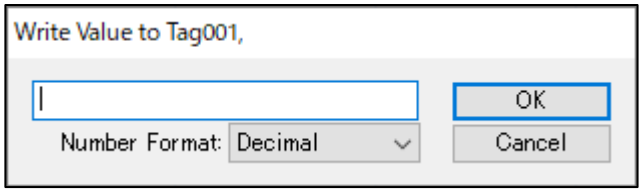
Important
If DeviceXPlorer shifts to Offline Mode, it stops performing communications processing entirely with target modules. When you shift during system running, be careful enough.

5.7.16 Write Tag Value

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



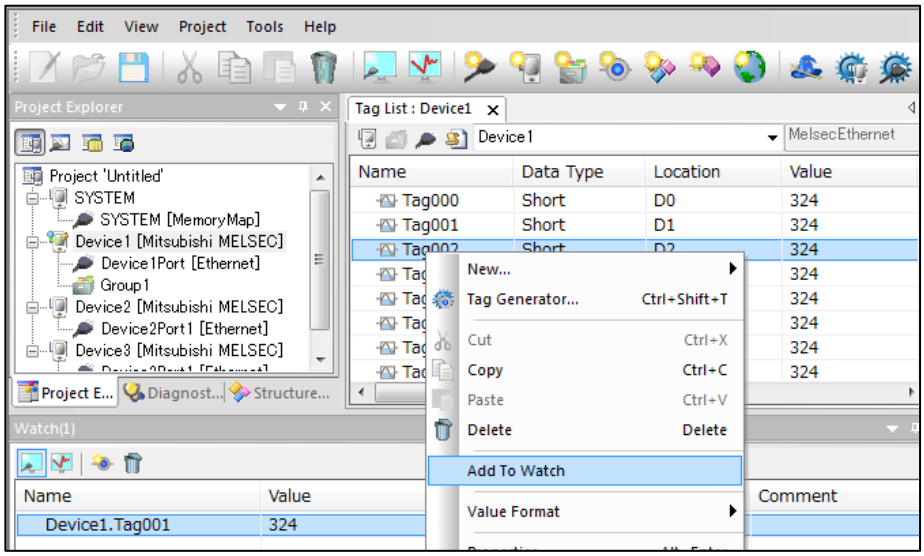
Since the follows dialog is displayed, You input a write value and click “OK”. DeviceXPlorer will write the value to target module.



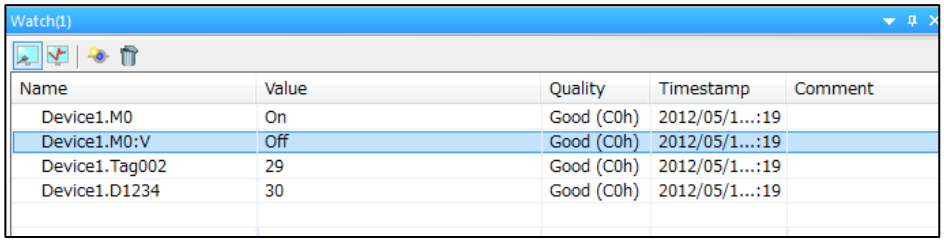
Note
"GUI->Writing Tag value From GUI" needs to be "Permit Writing" of Project Properties(5.13.6).

5.7.17 Watch

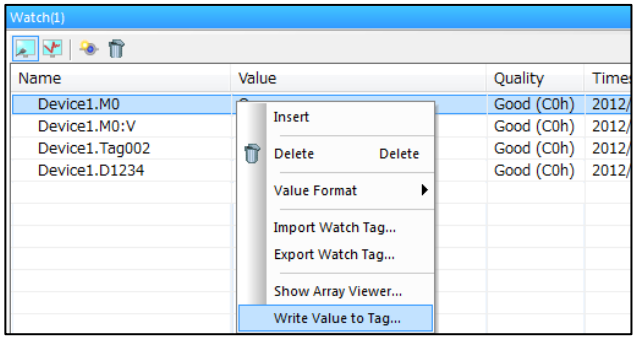
The watch function allows you to dynamically monitor the current values of tags if you drag & drop the tags you want to monitor from the Tag View.



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

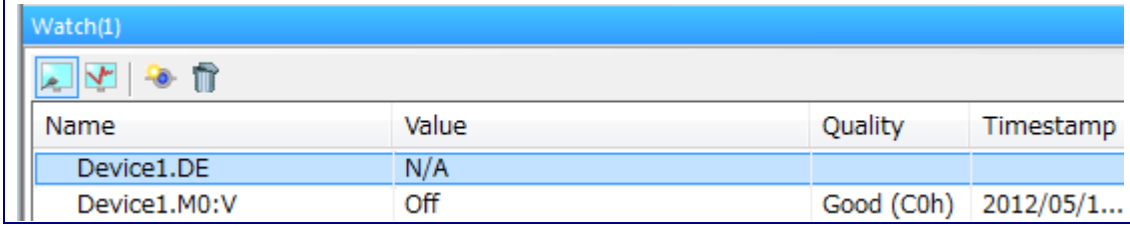


You can also save the tag list registered for the watch function in a file using the pop-up menu.



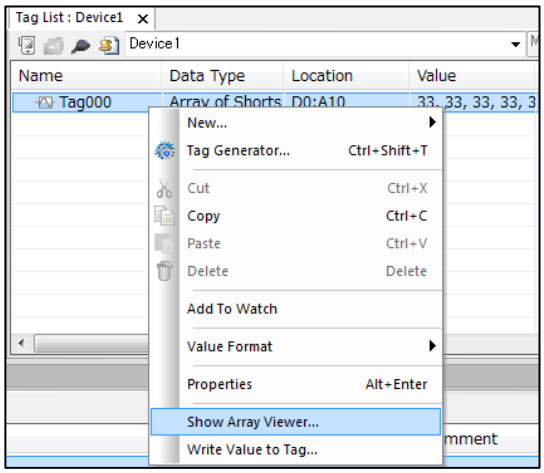
Note

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

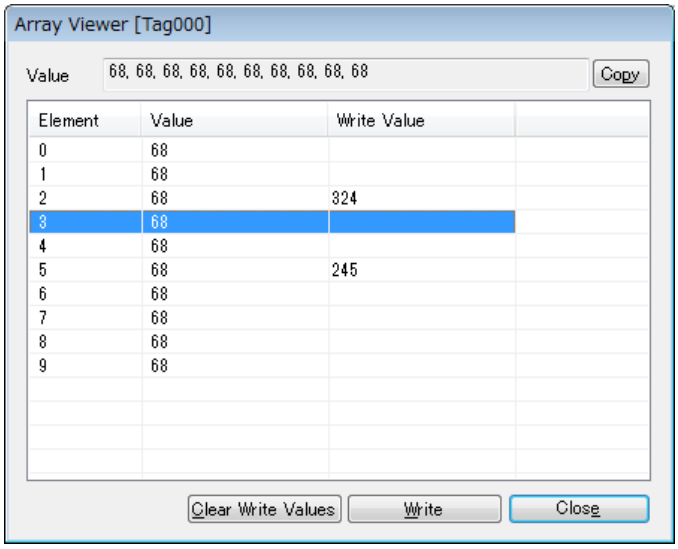


5.7.18 Array Viewer

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.



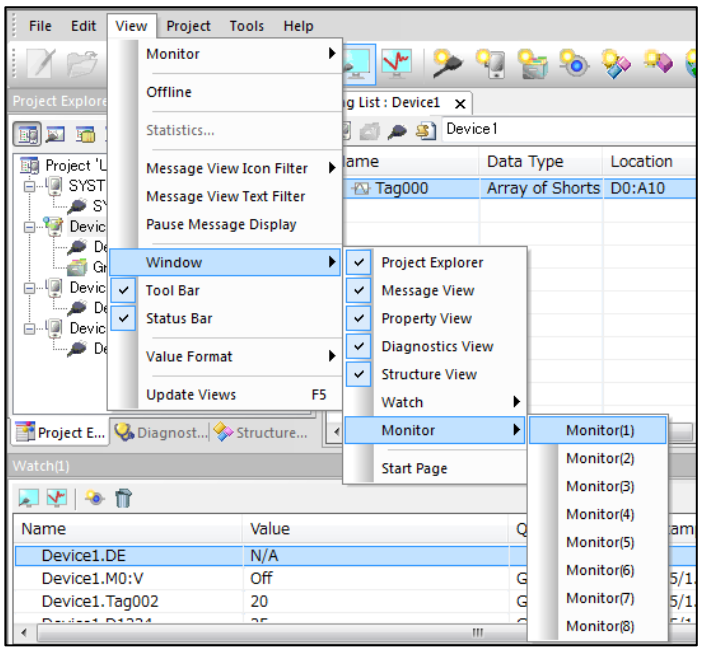
You can check the tag value and write value by “Write”.
If you write the value, please change the value and push “Write” button.



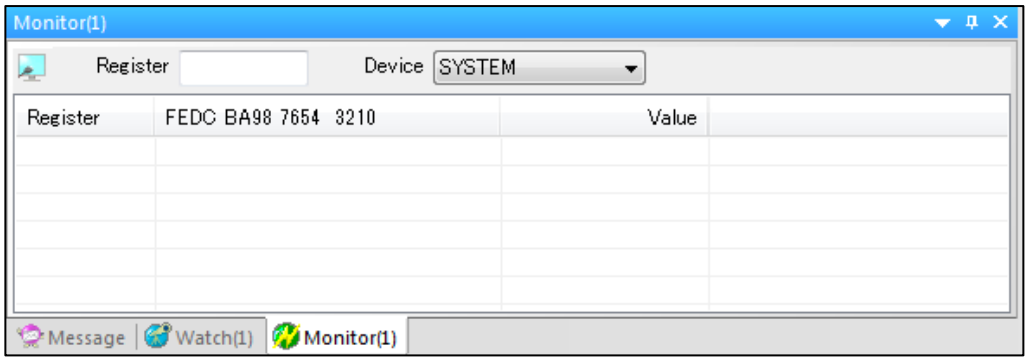
Item	Description
Value / Copy	Show the tag value. Copy button copies the values to clipboard.
Element	The number of the element.
Value	Show the target tag’s value.
Write Value	Edit each writing values.
Clear Write Values	Clear all writing values.
Write	Write the value to the target tag.

5.7.19 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.

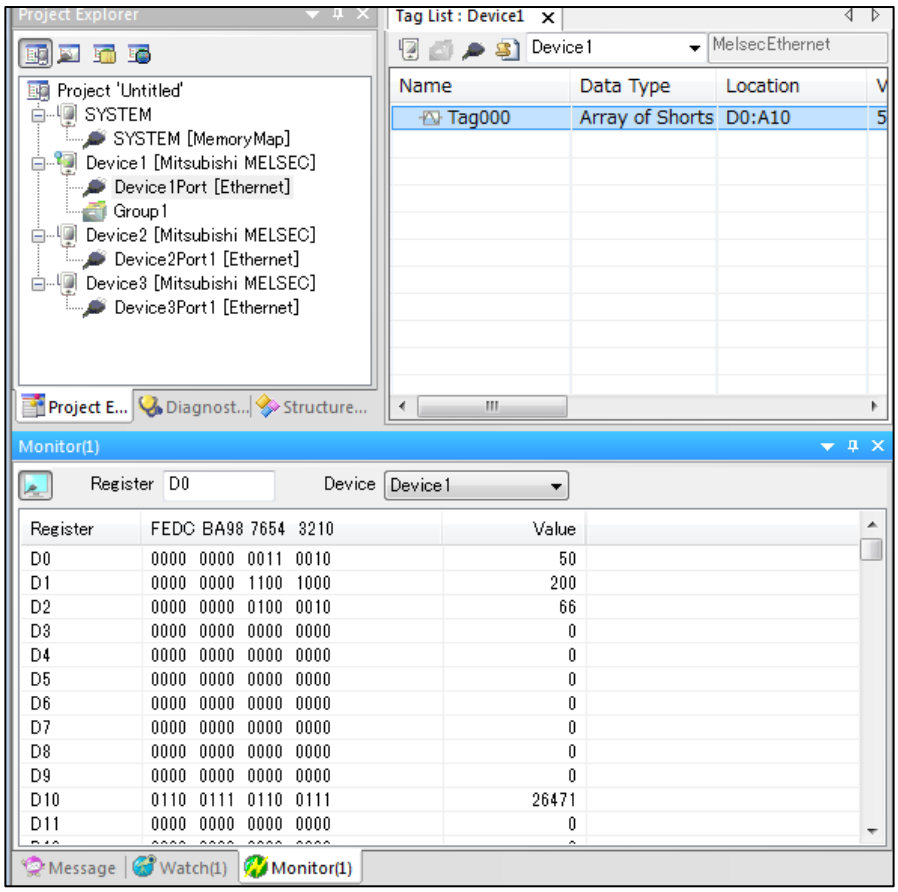


Item	Description
Device	Specify the target Device
Register	Specify register(memory name) in the target module.

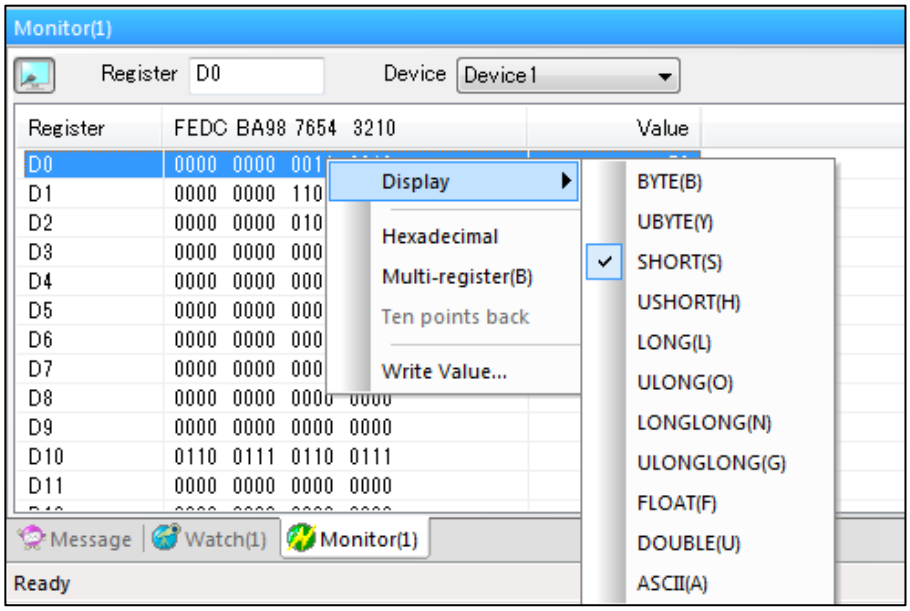
Important

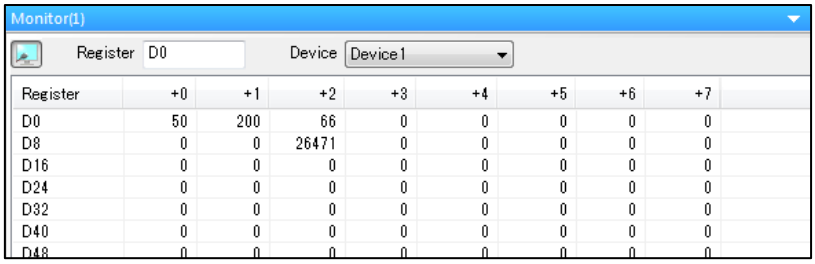
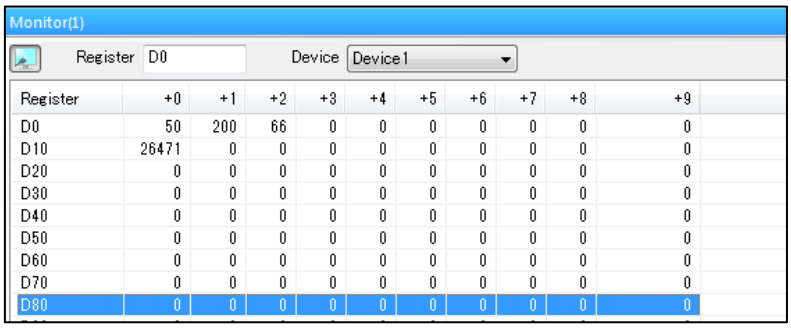
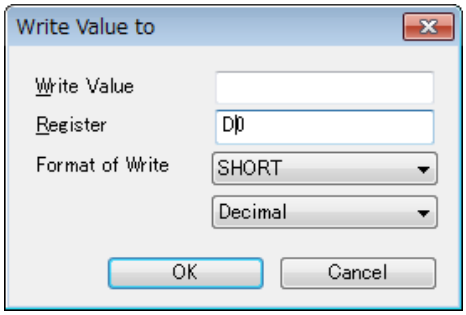
Register Monitor becomes a monitor function for PLC drivers.
The models (for example, ODBC Client, DDE Client etc.) which don't have register area cannot use this function.

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



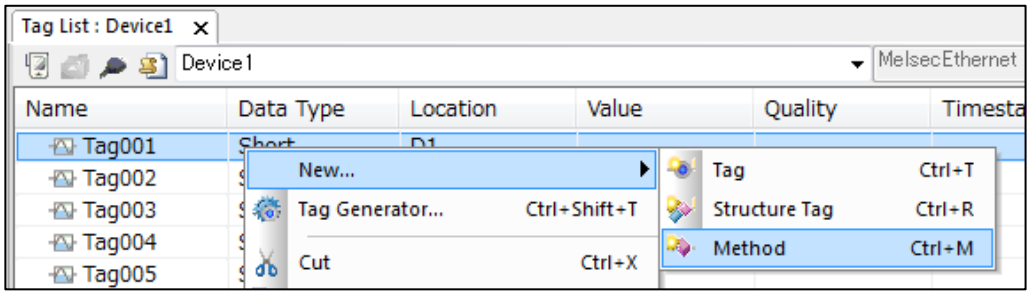
Item	Description
Display	Select data type for display.
Hexadecimal	Display values in hexadecimal.
Multi-register	Many registers are shown as below. (This is 8 points per line)  Monitor(1) Register D0 Device Device1 Register +0 +1 +2 +3 +4 +5 +6 +7 D0 50 200 66 0 0 0 0 0 D8 0 0 26471 0 0 0 0 0 D16 0 0 0 0 0 0 0 0 D24 0 0 0 0 0 0 0 0 D32 0 0 0 0 0 0 0 0 D40 0 0 0 0 0 0 0 0 D48 0 0 0 0 0 0 0 0
Ten points back	Many registers are shown as below. (This is 10 points per line)  Monitor(1) Register D0 Device Device1 Register +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 D0 50 200 66 0 0 0 0 0 0 0 D10 26471 0 0 0 0 0 0 0 0 0 D20 0 0 0 0 0 0 0 0 0 0 D30 0 0 0 0 0 0 0 0 0 0 D40 0 0 0 0 0 0 0 0 0 0 D50 0 0 0 0 0 0 0 0 0 0 D60 0 0 0 0 0 0 0 0 0 0 D70 0 0 0 0 0 0 0 0 0 0 D80 0 0 0 0 0 0 0 0 0 0
Write Value	Input a Write Value and write Register. 

5.8 Method Tag

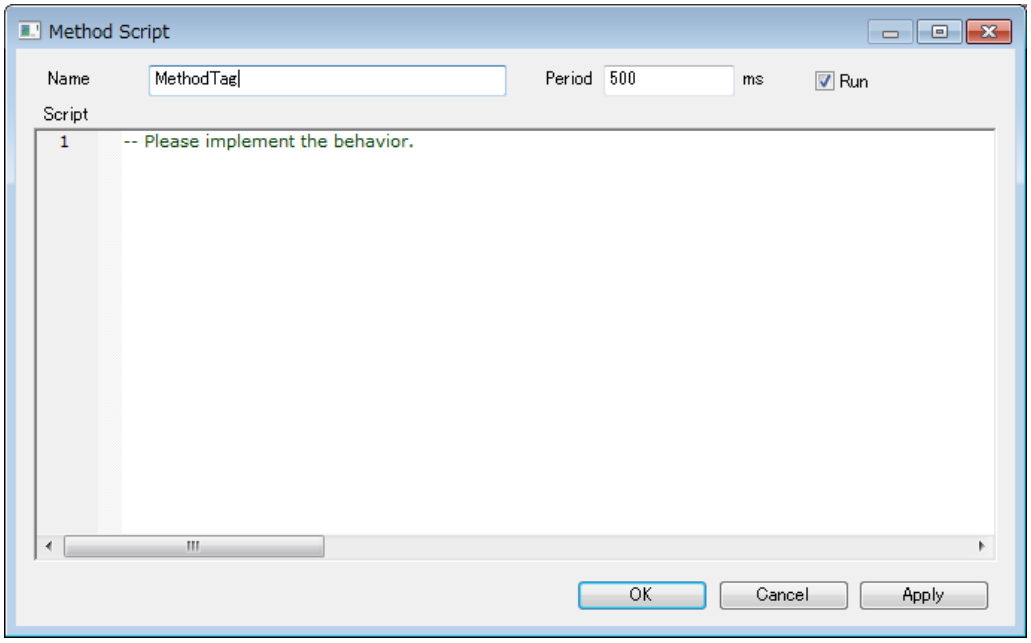
You can create a method tag with script processing.
If you write value is to a method tag, DeviceXPlorer executes the script logic of Method Tag.

5.8.1 Creating a New Method Tag

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



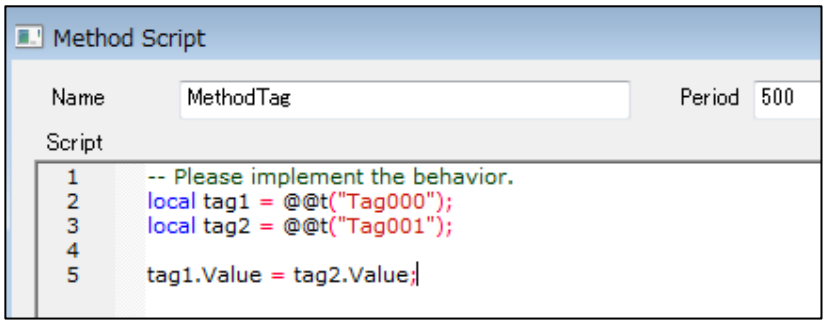
Edit script.

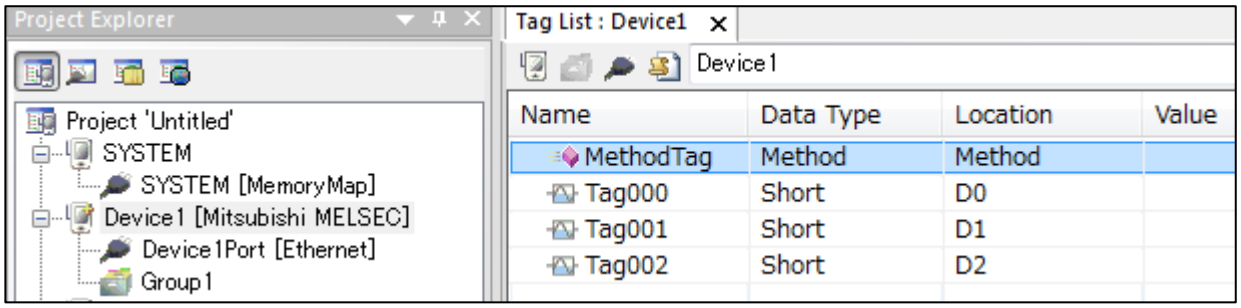


Item	Description
Name	Specify the Method Tag name.
Period	Specify the update rate for refresh tag's cache value which used in script.
Run	If you execute script, Check the box.
Script	Programing. Refers to *****

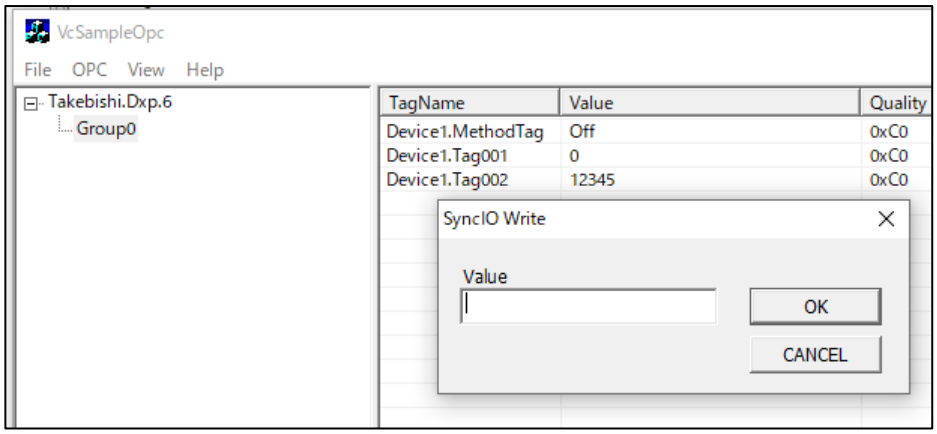
5.8.2 Execute Method Tag

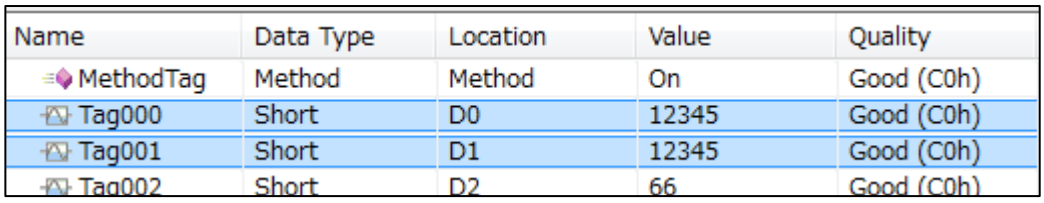
You create a Method tag like follows.



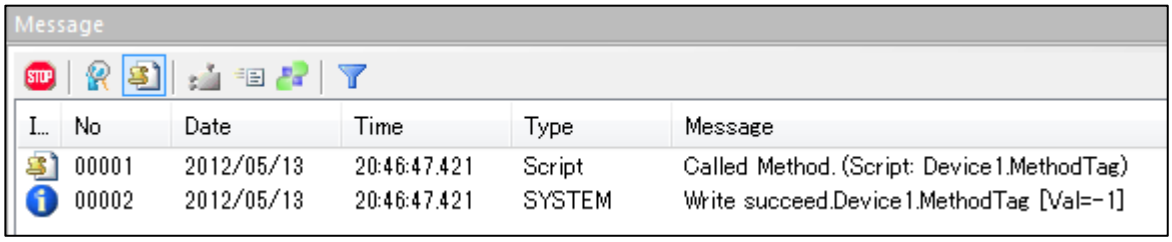


If a value is written in from a client to this tag, the script will be performed.





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



5.9 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 2type Structure Tag and Structure Template.

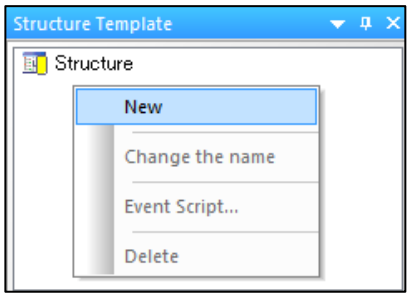
Item	Description
Structure Template	Definition of structure which has some members.
Structure Tag	Tag which created based on Structure Template.

Important

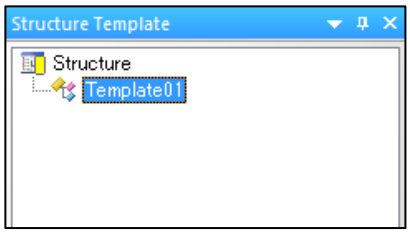
Tag settings in ODBC Client Connection don't support Structure Tag.

5.9.1 Define Structure Template

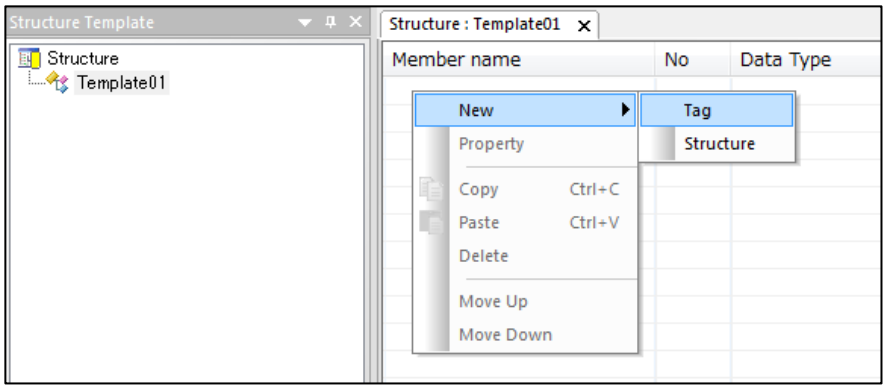
To create a Structure Tag, You define a Structure Template is at first.
You select "New" in Structure Template View.



"Template01" is created as a Structure Template.

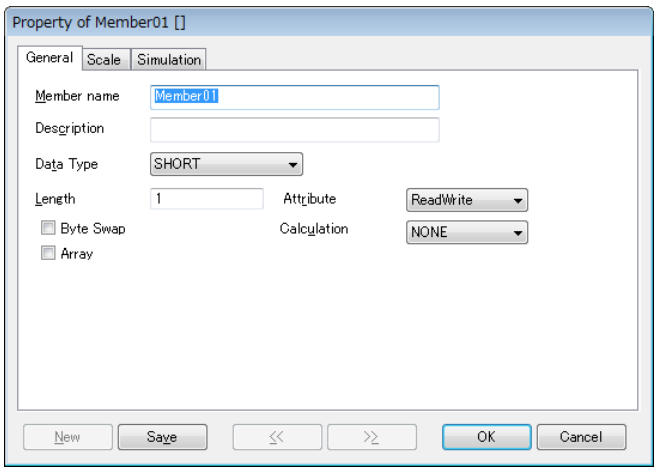


By clicking "Template01", Member list is shown. You select "Tag"/"Structure" from right-click menu of the component.



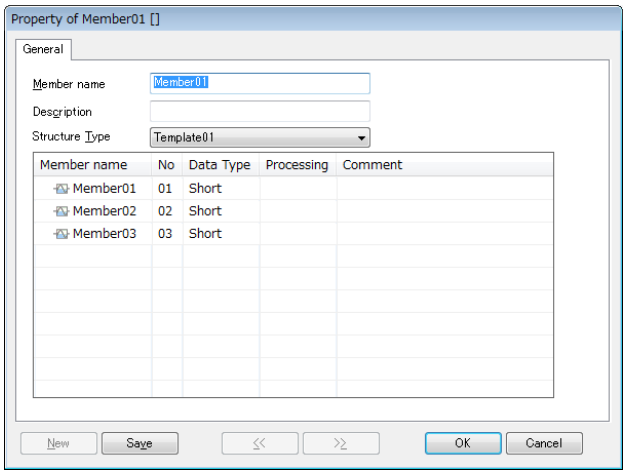
You configure the member properties .

<Tag>



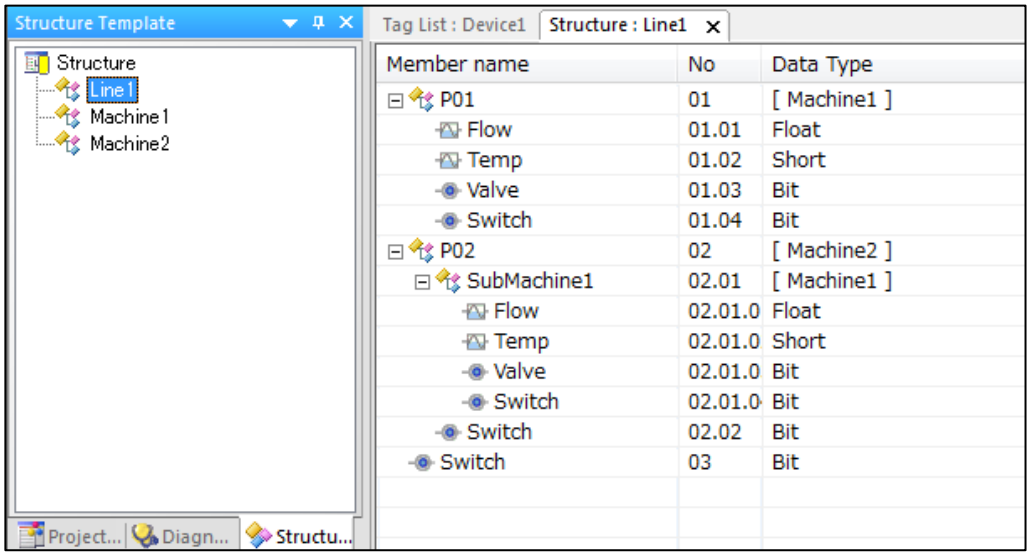
Item	Description
Member name	Specify a member name.
Description	Allows you to add a comment to the member.
Data type	Select a data type.
Size	Specify the size if you have specified a character string or an array.
Attribution	Specify a member attribution.
System calculation	Select a calculation..
Byte 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>

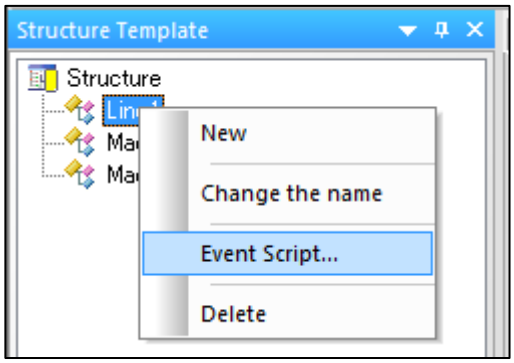


Item	Description
Member name	Specify a member name.
Description	Allows you to add a comment to the member.
Structure Type	Select a Structure Template which defined already.
Member List	Show members of selected Structure Template.

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.



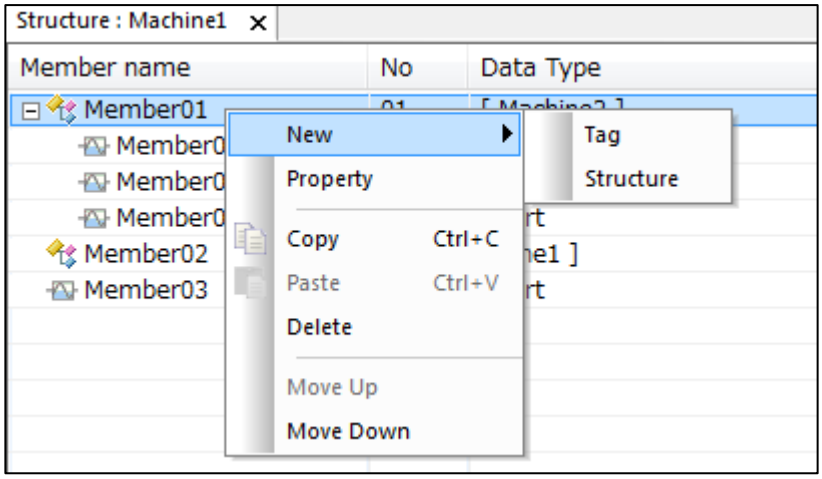
Item	Description
New	Create a new Structure Template.
Change the name	Change the selecting Structure Template name.
Event Script	Create a new Event Script of Structure Template.
Delete	Delete the Structure Template. If selected Structure Template is referred from another Structure Template, This structure 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..



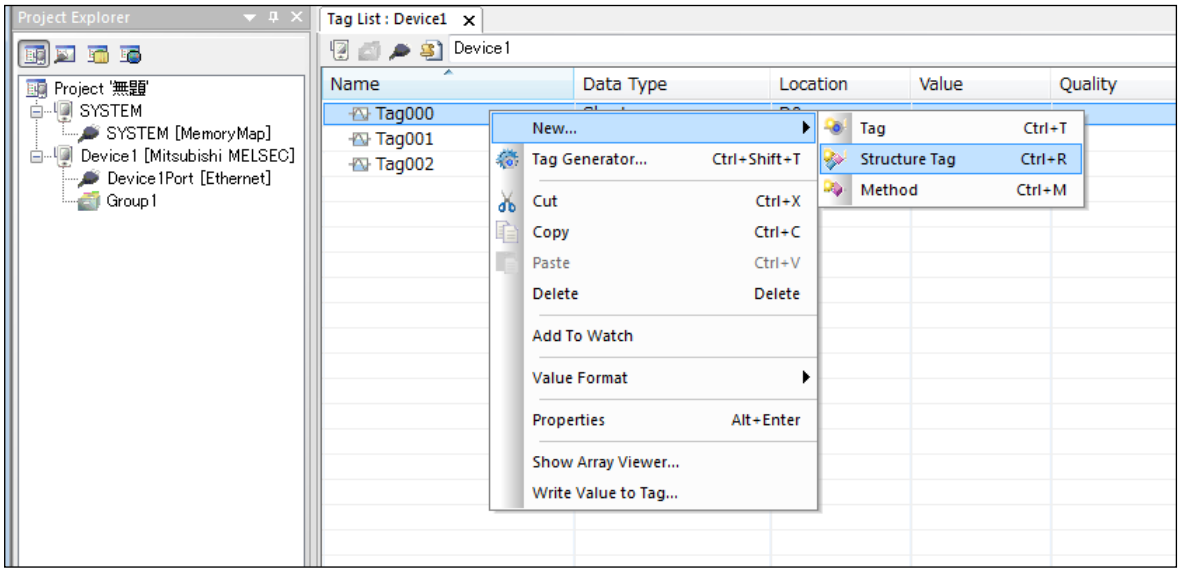
Item	Description
New	Create a new Structure Template member Tag or Structure.
Property	Configure the member properties.
Copy	Copy the selected member.
Paste	Paste the copied member.
Delete	Delete the selected member.
Move Up	The row of the member is moved upwards.
Move Down	The row of the member is moved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

Important

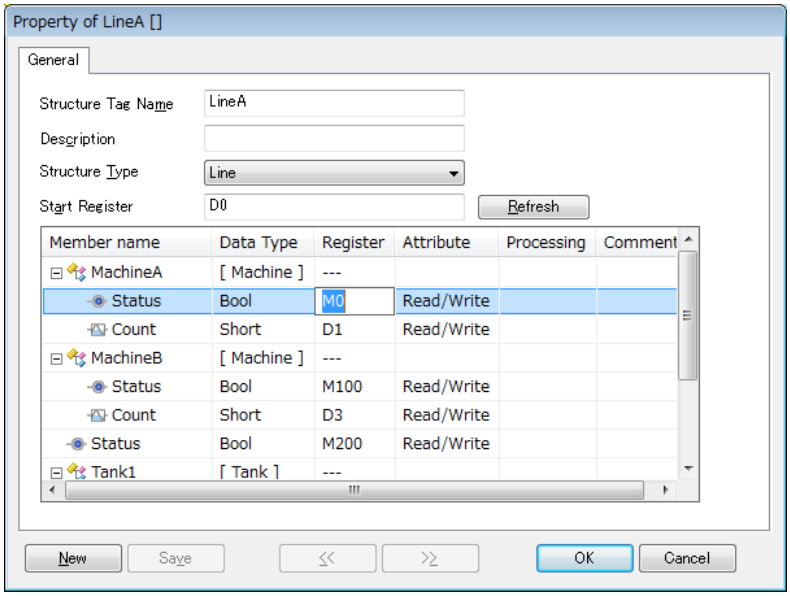
You can't delete the selected Structure Template member, if Structure Tag which created based on this Template is communicating.

5.9.2 Creating a New Structure Tag

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



Input the member properties.

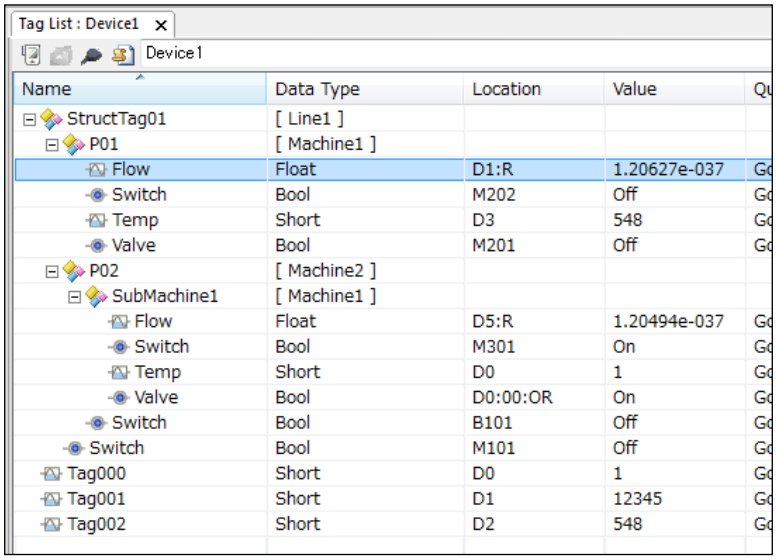


Item	Description
Structure Tag Name	Specify the Structure Tag name.
Description	Allows you to add a comment to the Structure Tag.
Structure Type	Select a Structure Template which defined already.
Start Register	If you specify start register and push "Refresh" button, based on the start register, structure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

Important

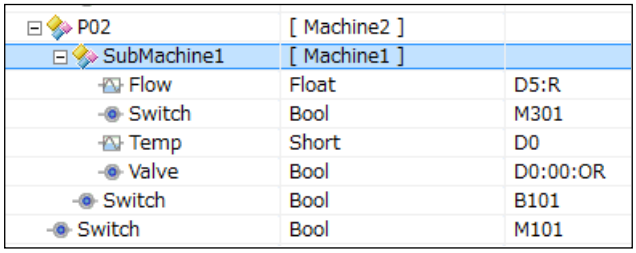
A register's contents of a setting change to a character string with extended specification by the register who assigns. For example, if "D0" is inputted into a Float type member and a setup is saved to him, it will be registered as "D0:R."

You can monitor Structure Tags in Tag List.

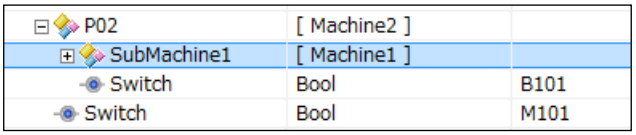


You can expand members by click [+], collapse by [-].

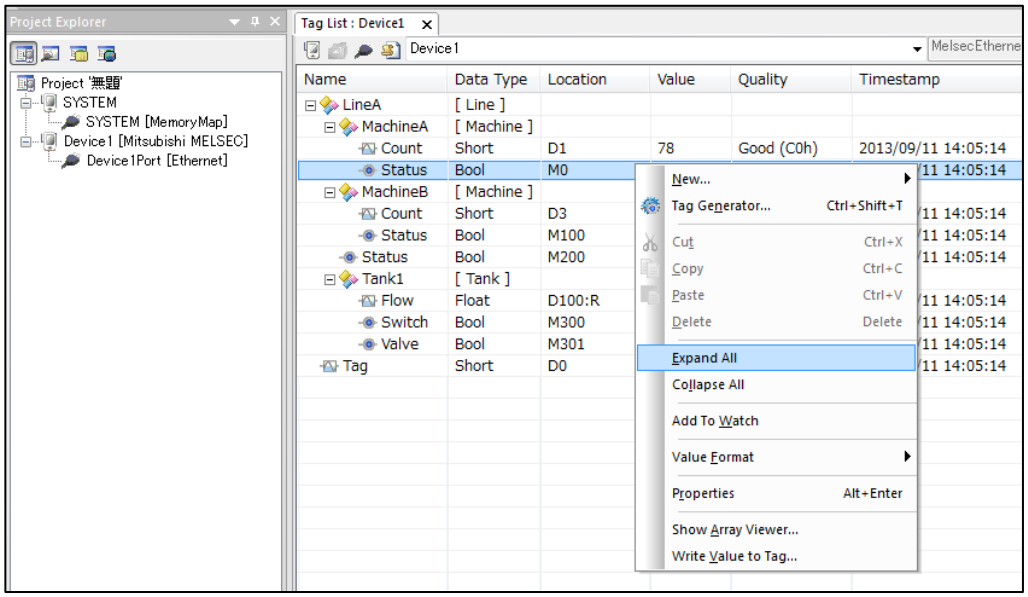
<Expand members>



<Collapse members>

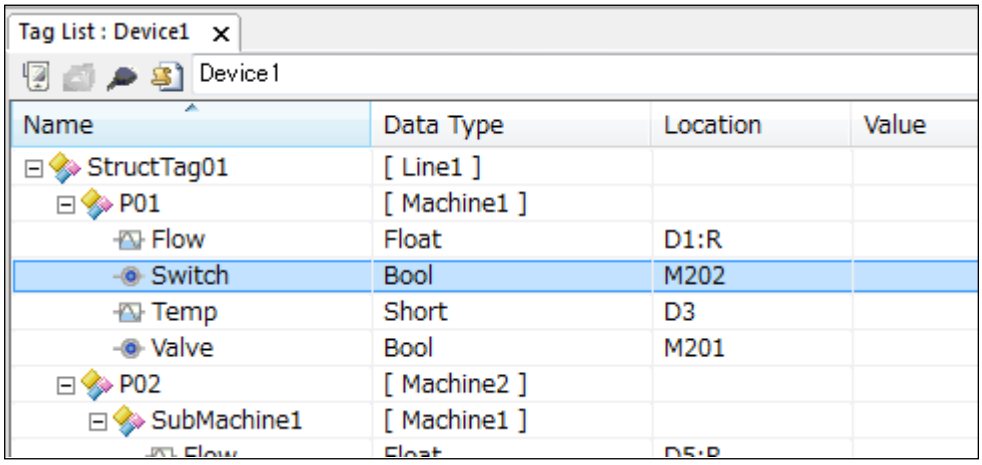


Also, you can expand and collapse all structure tag's members by right-click menu. “Expand All” and “Collapse All”.

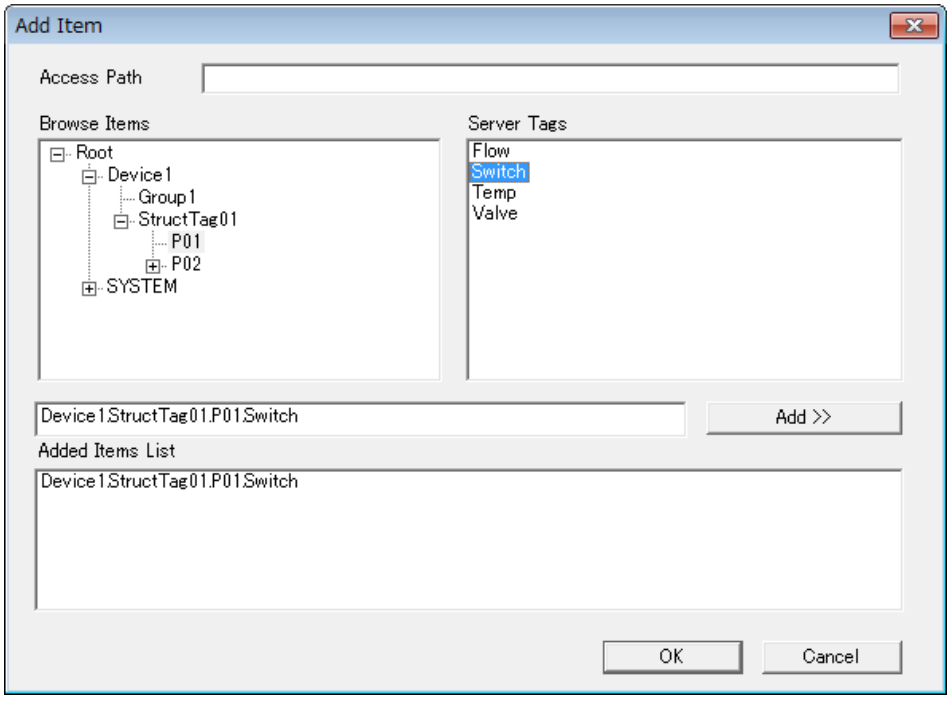


5.9.3 How to access Structure Tags from Client

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."



It is as follows if it browses from a client.



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.

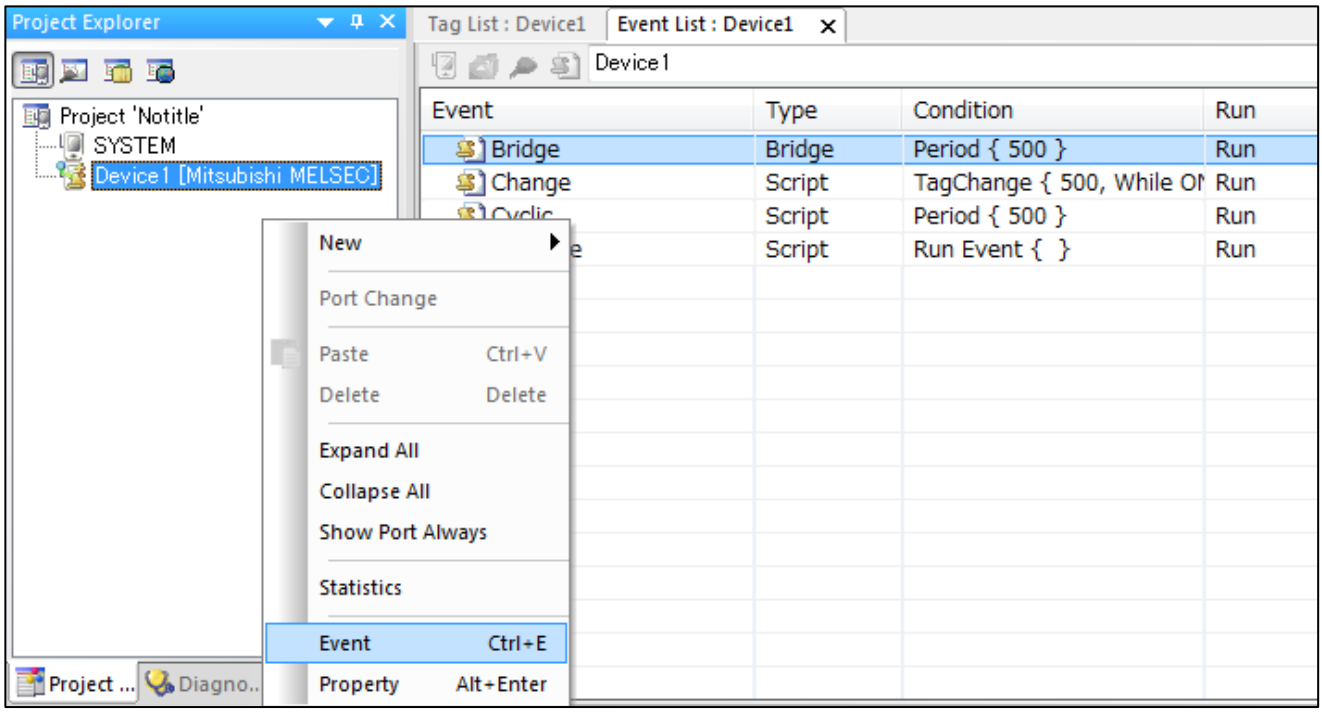
5.10 Bridge

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

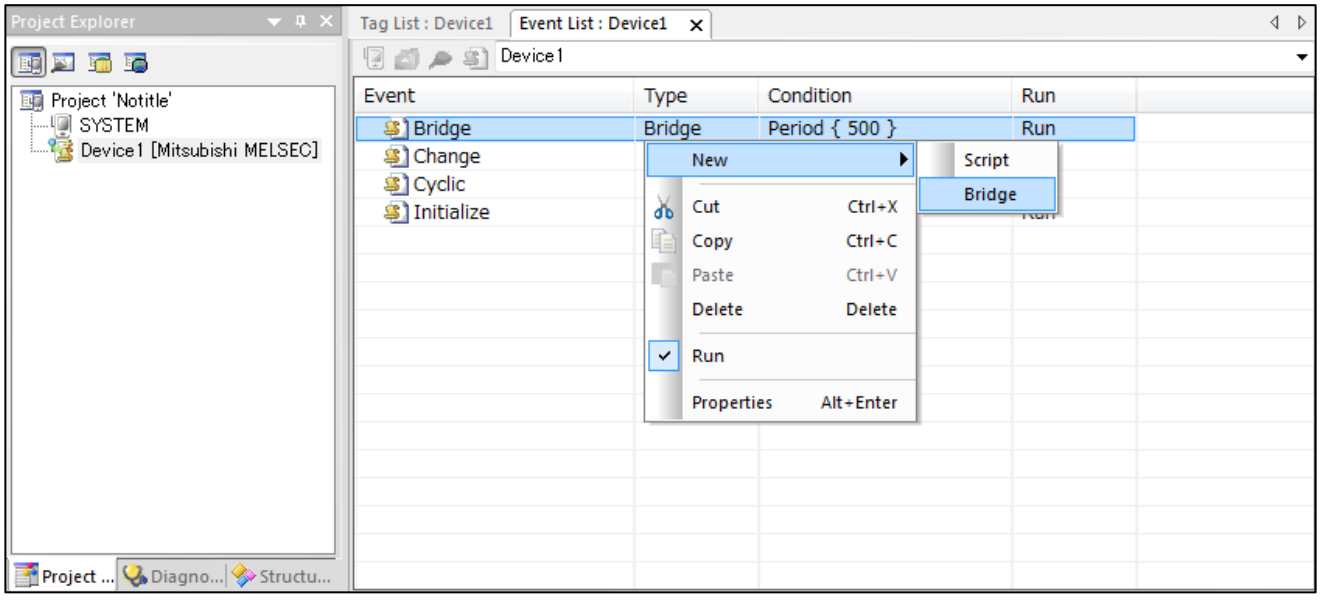
Important
- There is no upper limit to the number of bridge settings, but please note that the communication load will increase. - Up to 32 items can be set in one bridge setting. - Please note that executable number of scripts (including bridges) is limited depending on the edition of DeviceXPlorer.

5.10.1 Bridge Setting

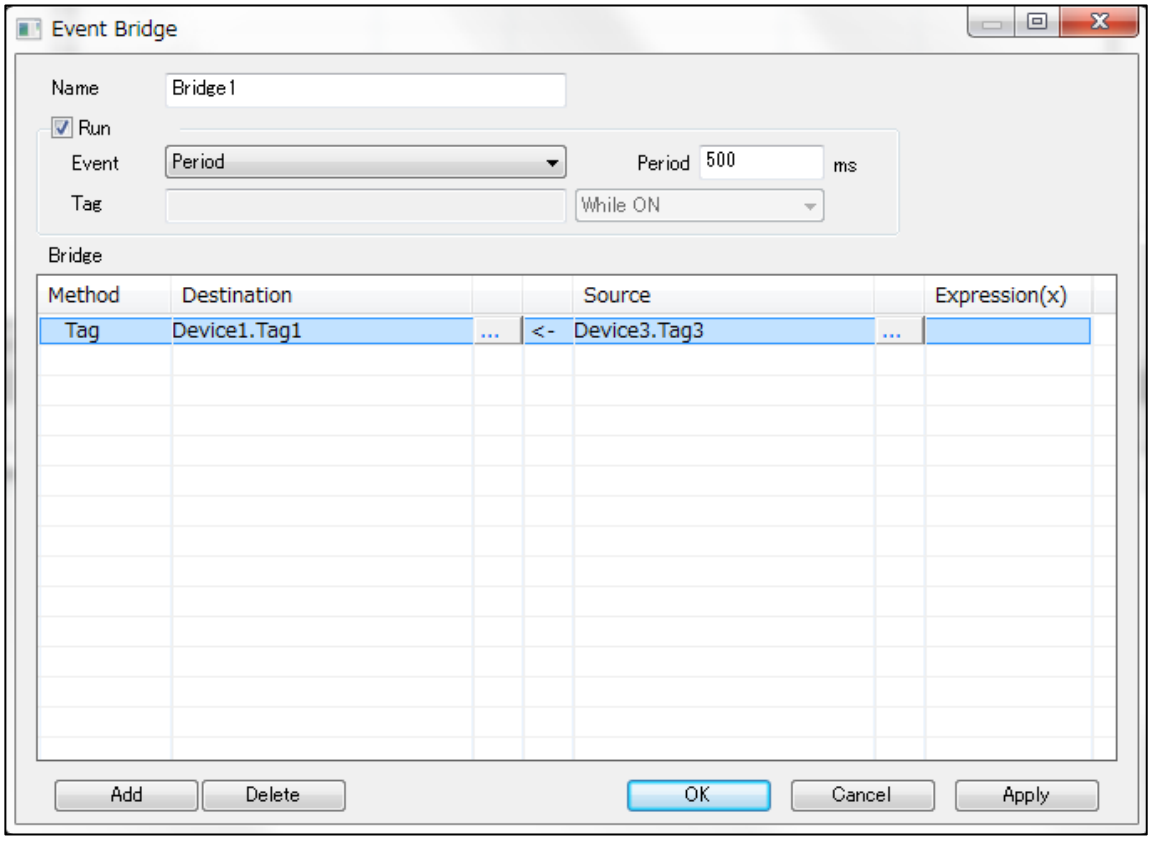
Tag, Group and Device have bridge function.
Event list will be displayed when select [Event] from right-click menu for Device and Group.



Events will be displayed in event list.
You can create, modify, delete and run/stop event from right-click menu.



Input bridge setting.



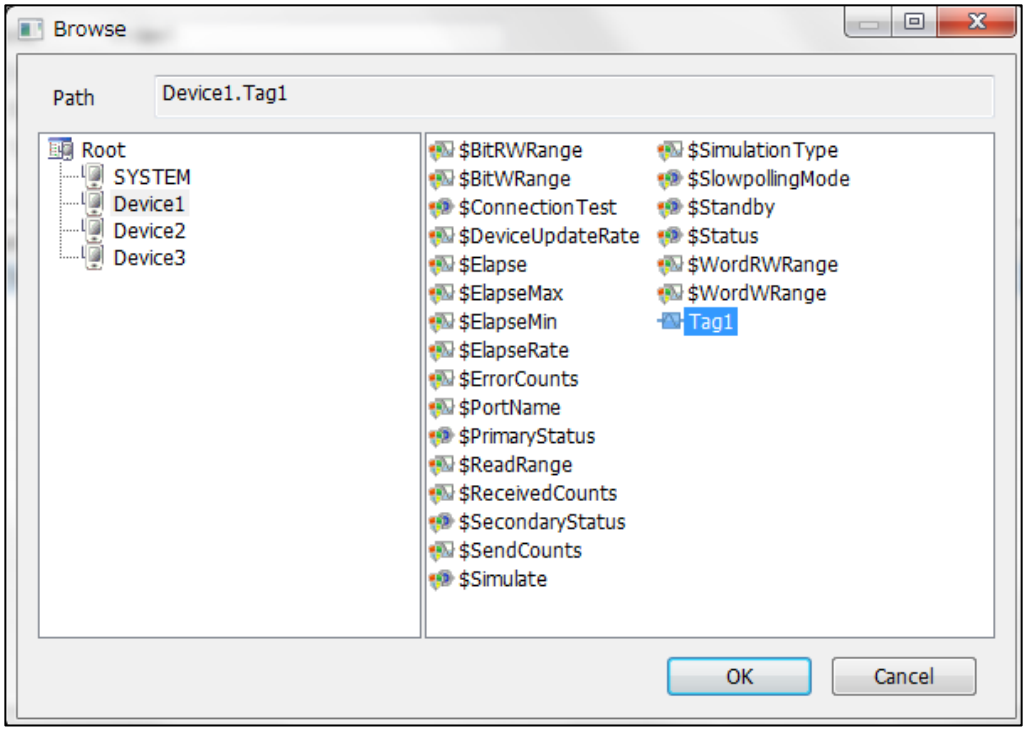
Item		Information
Name		Setting bridge name.
Run		Specify run state of bridge. If no check, bridge is not performed
	Event	Specify event condition to execute bridge. [Period] - - - Bridge is performed the specified cycle (fixed cycle). [Tag Change] - - - Bridge is performed when the value of the tag 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. (*)
	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.

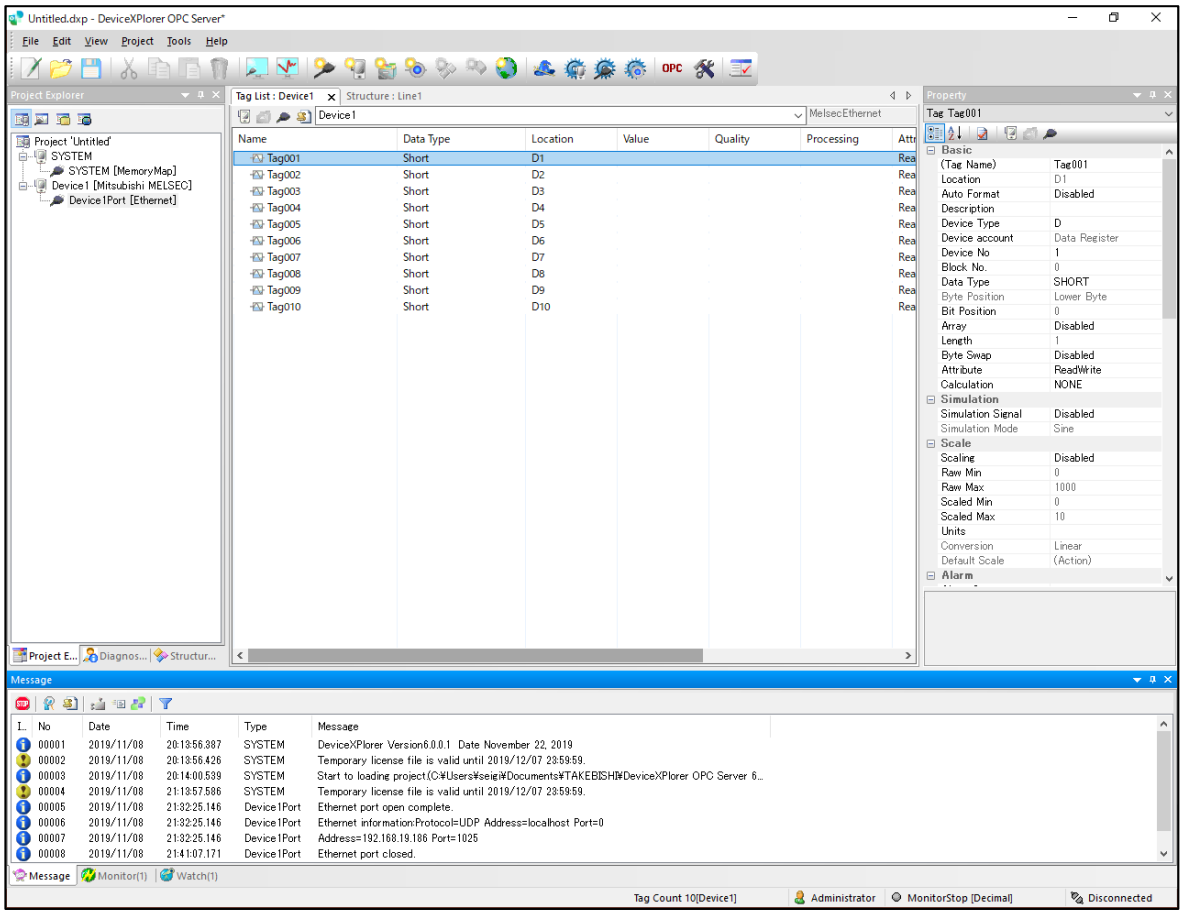
* You can choose Tag, Device and Group, by using Browse dialog function.



5.11 Messages

5.11.1 Message

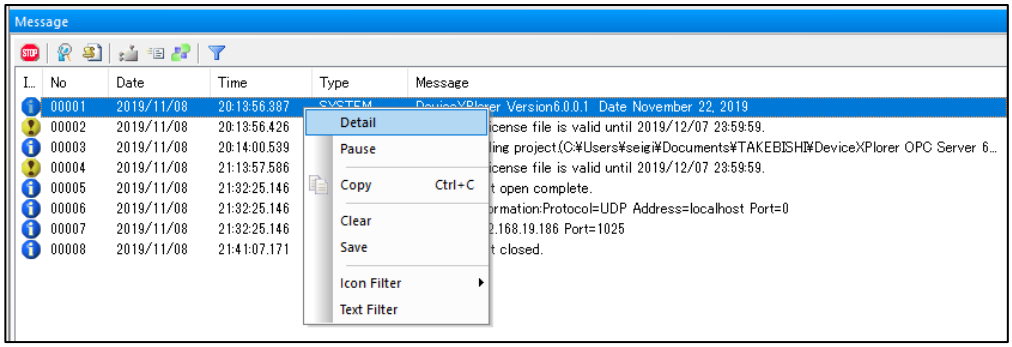
DeviceXPlorer divides messages into four levels of importance for processing (information, warning, error, and fatal error).



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

Icon	Category	Description
	Information	Information message for communication parameters, etc.
	Warning	Warning level message
	Error	Error level message
	Fatal error	Fatal error message
	Script Log	Execute script message
	DxpLink Log	DxpLink communication message
	OPC Trace	OPC Interface 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.

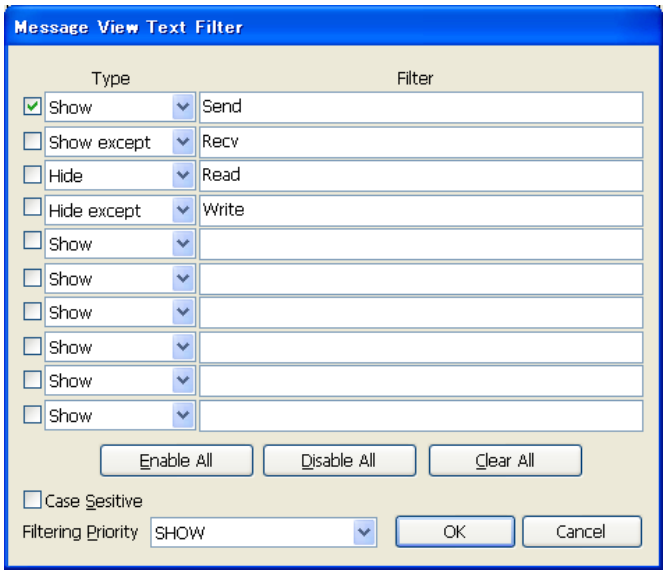
Important
The function that saves log file doesn't be suspended, if you configure "Auto Message Save" as "Save".

[Clear]
Messages are cleared from Message View.

[Icon Filter]

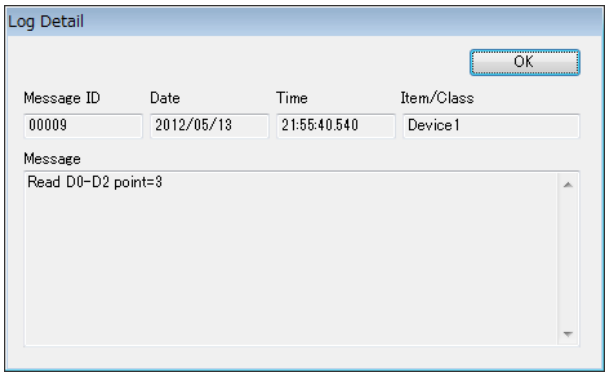
Item	Description
Fatal Error	Show fatal error message.
Error	Show error message such as communication error.
Warning	Show warning message.
Info	Show information message.
Communication Data	Show communication data message with PLC.
OPC Interface Access	Show OPC interface access message with OPC Client.
SuiteLink Trace	Show SuiteLink interface access message with SuiteLink Client.
DxpLink Trace	Show DxpLink interface access message.

[Text Filter]

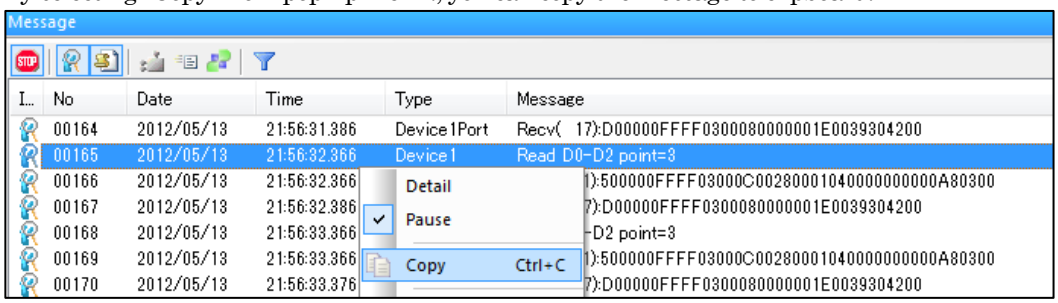


Item		Description
Check Box		Enable the filter.
Type	Show	Show message including the filter message.
	Show except	Show except message including the filter message.
	Hide	Hide message including the filter message.
	Hide except	Hide except message including the filter message.
Filter		Input filter message.
Case Sensitive		Case sensitive check.
Filtering Priority		Specify filtering priority.

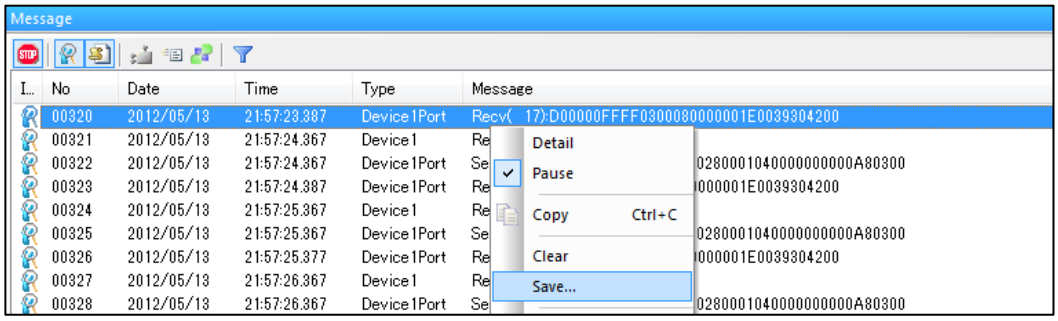
By double-clicking the message line, you can view a detailed dialog. It is possible to copy error messages.



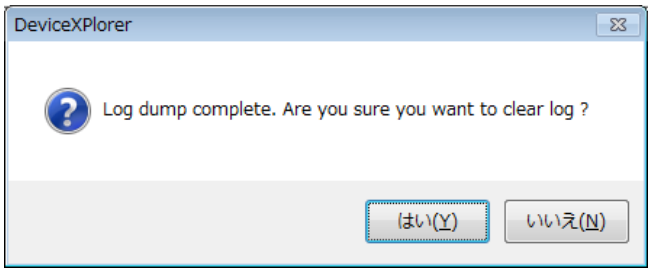
By selecting "Copy" from pop-up menu., you can copy the message to clipboard.



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

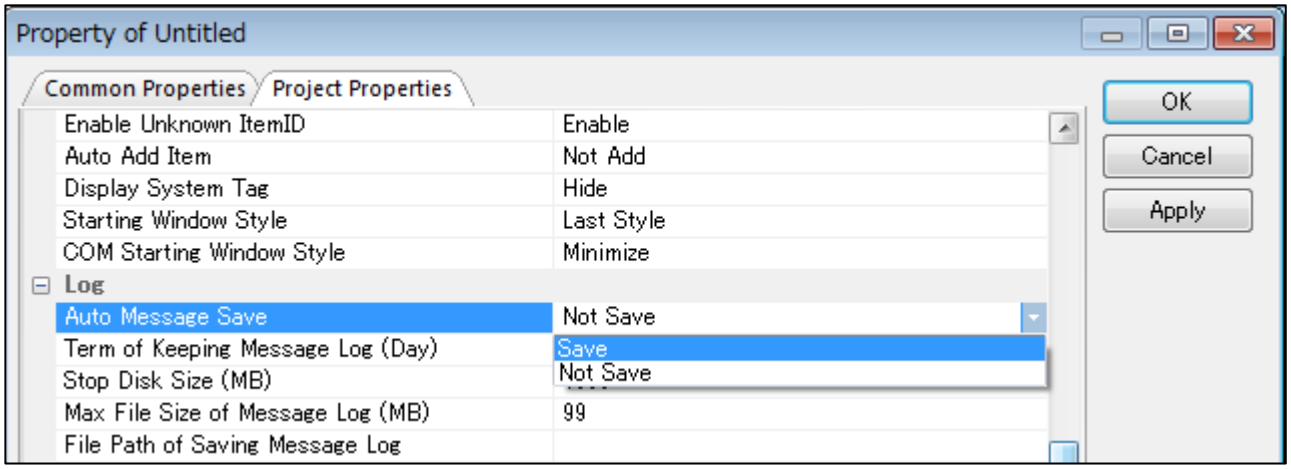


It is checked after CSV file preservation whether messages are cleared or not.



5.11.2 How to save messages automatically

You configure “Auto Message Save” as “Save” in Project Properties (5.13.7).

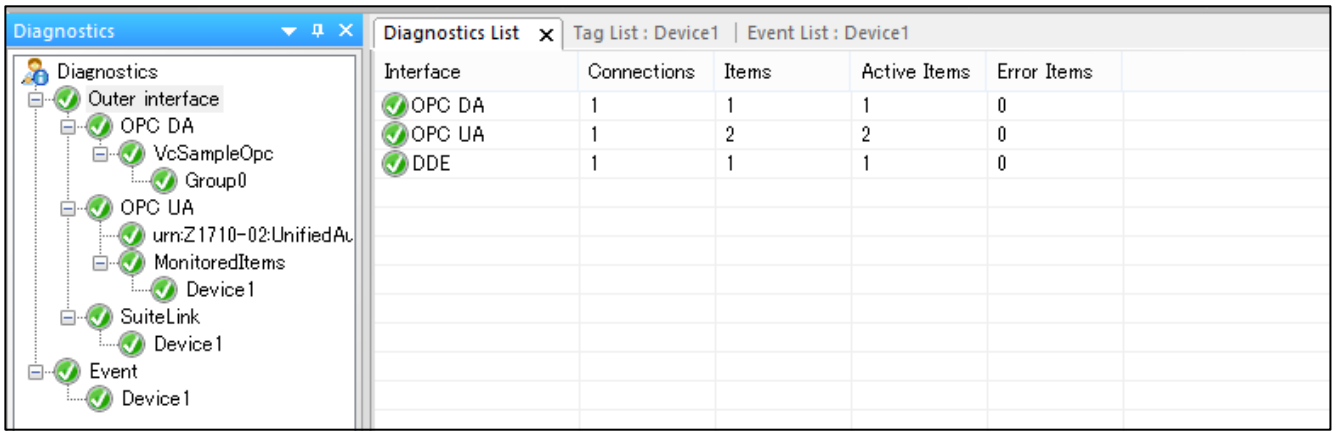


5.12 Diagnosis

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.

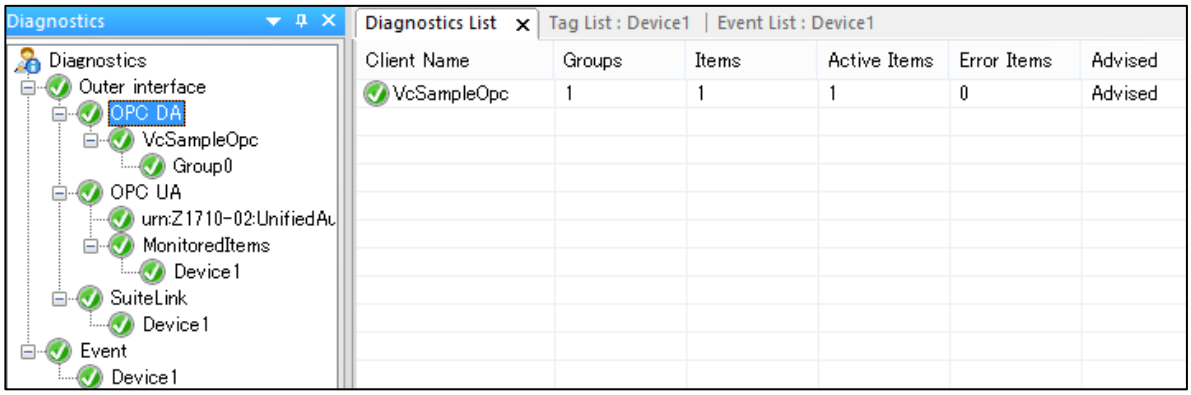
5.12.1 Interface Diagnosis



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

5.12.2 Client Diagnosis

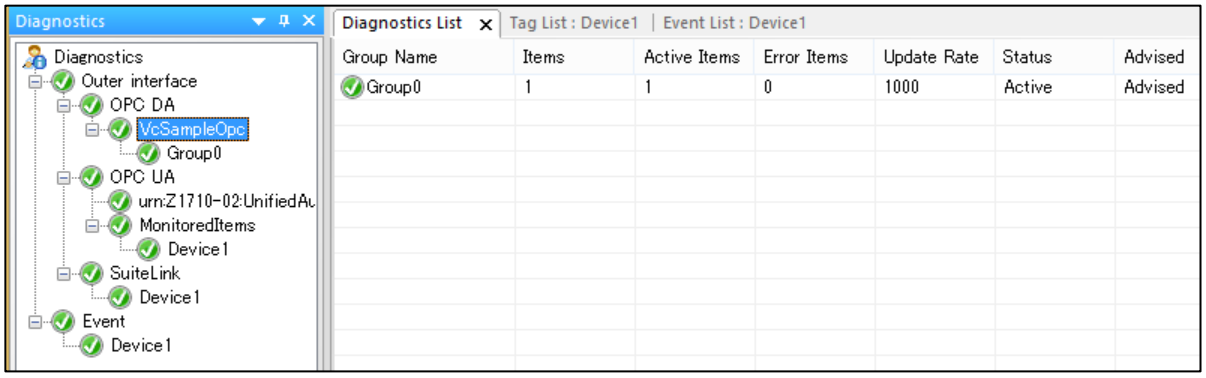
This shows information for the connected clients.



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.

5.12.3 Group Diagnosis

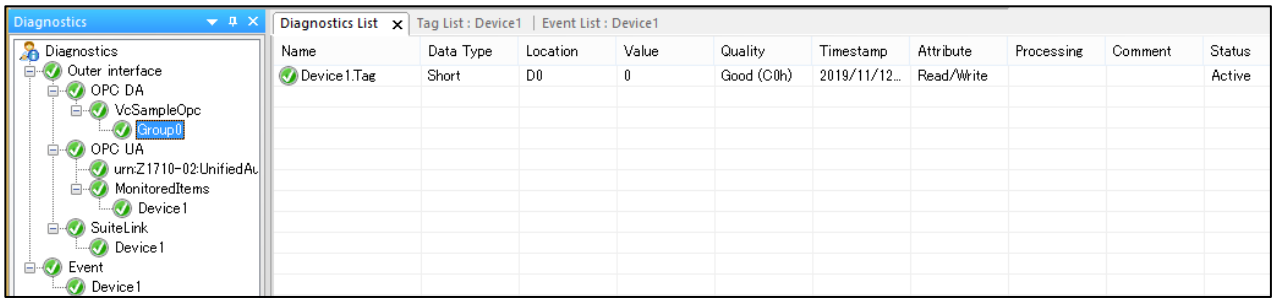
This shows the information of the group to which the client is registered.



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.

5.12.4 Tag Diagnosis

This shows information on the tags included in the group.

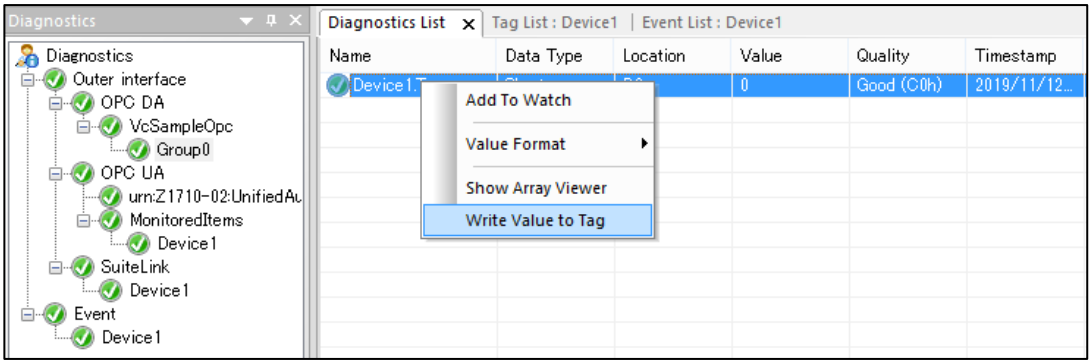


Item	Description
Name	Tag name
Data Type	Tag data type
Location	Physical address to refer
Value	The most recent cache value
Quality	Quality of tag
Time stamp	Time stamp
Attribute	Attribute of tag
Processing	Existence of scale setting
Comment	Comment of tag
Status	Active / InActive status of tag

Undernoted icons show the diagnosis condition of tag.

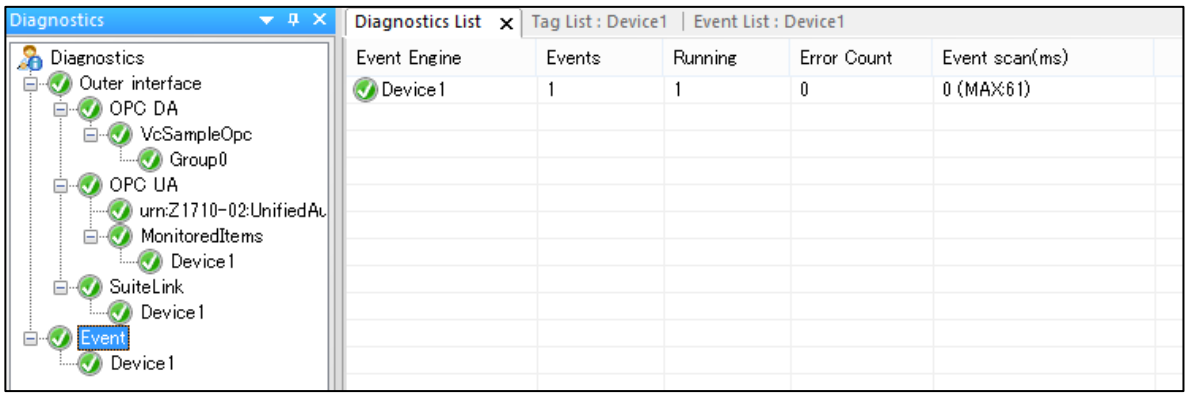
Icon	Description
	Tag quality is Good.
	Tag quality is Uncertain.
	Tag quality is Bad. Communication error is occurring.
	Tag quality is Waiting for initial data.
	Tag condition is not active status.

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.



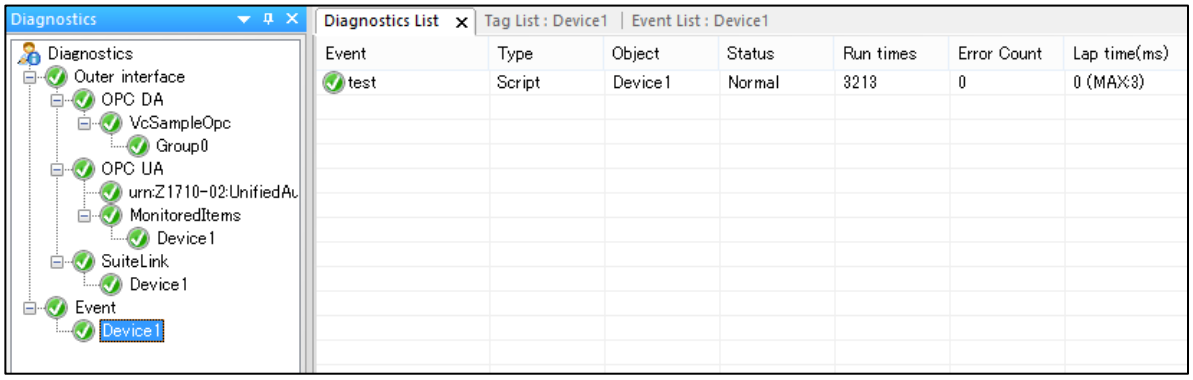
5.12.5 Event Diagnosis

This shows information on the events.



Item	Description
Event Engine	Event Engine name
Events	The number of registered Events
Running	The number of executing Event
Error Count	The count of error
Event scan(ms)	The scan time of Event.(Max scan time)

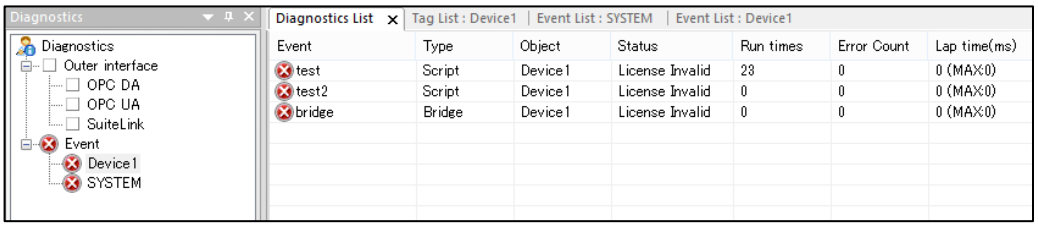
If you click an Event Engine name, the every event status is shown.



Item	Description
Event	The name of Event
Type	The type of Event
Object	The object name to which an Event belongs
Status	Status of Event
Run times	Execute times of Event
Error count	The count of error
Lap time	The lap time of Event.(Max scan time)

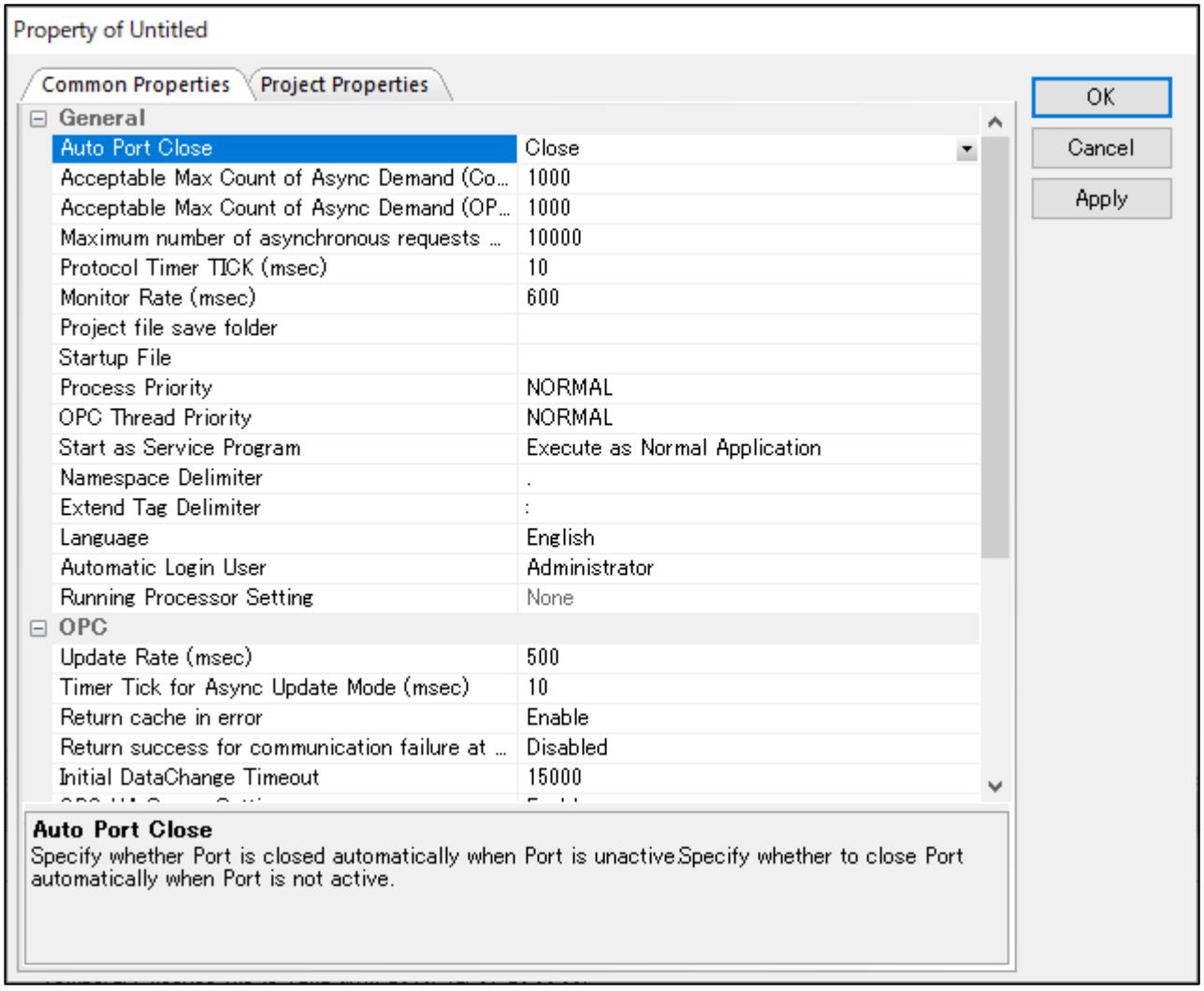
Important

When you use two or more events without the license of enterprise, “License Invalid” occurs in Event Diagnosis.



5.13 Projects

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.



Note
DeviceXPlorer settings are saved to a specified folder as files with the extension “*.dxp”.

5.13.1 Common Properties, General

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

Item	Default	Reboot	Description
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.
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).
Acceptable Max Count of Async Demand (SuiteLink)	10,000	Need	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.
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.
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.
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.
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.
Process Priority	Normal	Need	Specifies the process priority. [High] [Low] REALTIME > HIGH > NORMAL > LOW
OPC Thread Priority	NORMAL	Need	Specifies the thread priority. [High] [Low] CRITICAL > HIGHEST > NORMAL > LOWEST > IDLE
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.
Namespace Delimiter	. (Dot)	Need	Specifies the delimiter for Item ID.
Extend Tag Delimiter	: (Cologne)	Need	Specifies the delimiter for extended tag name of dynamic

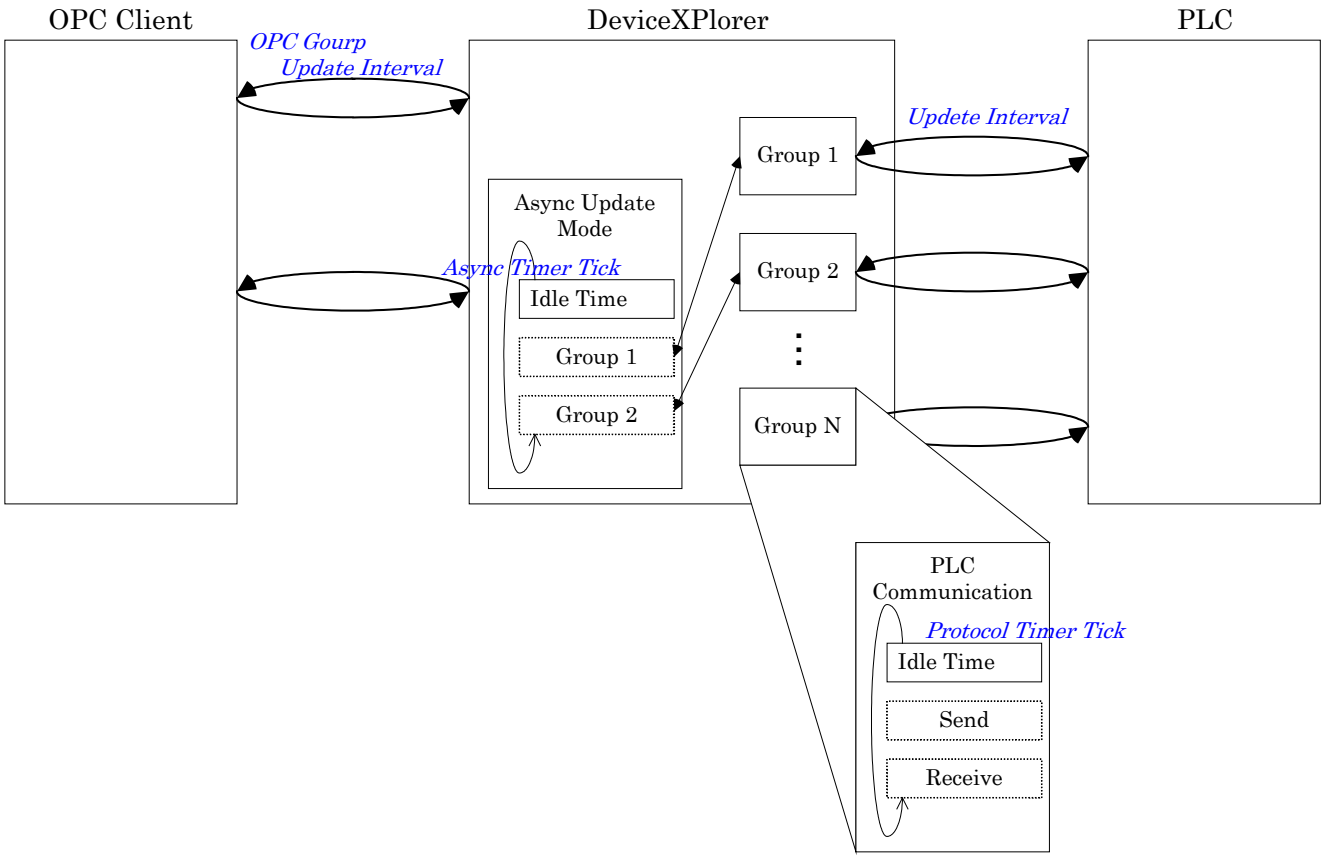
			tag.
Language	English	---	Specifies the language of the application. You can specify “Japanese”, “English” or “Chinese (Simplified)” or “Chinese(Traditional)” or “Korean(Hangul)”.
Auto Login User	Administrator	Need	Select the login user when DeviceXPlorer starts.
Running Processer Setting	(No specification)	---	Specifies running CPU processer of DeviceXPlorer. If this setting is not specified, DeviceXPlorer runs with all CPU processer.

* When connecting with OPC server which runs as windows service, the DCOM configuration is required. About the DCOM configurations, please refer to5.22.

5.13.2 Common Properties, OPC

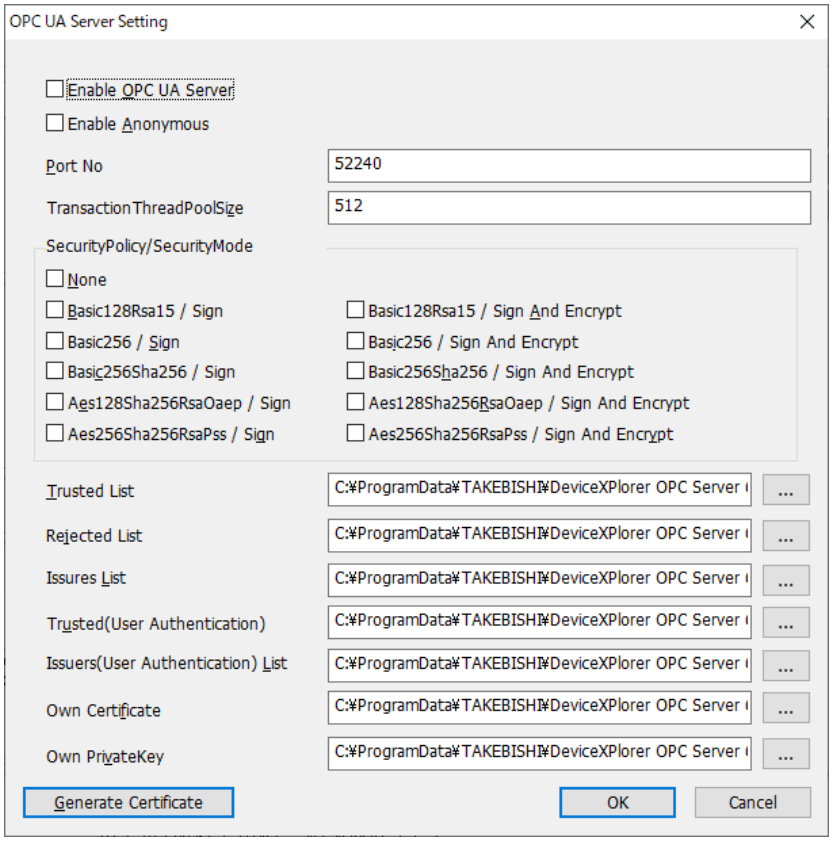
Item	Default	Reboot	Description
OPC Update Rate	500 (ms)	Need	Specifies the interval for notifying an OPC client of data changes. This setting value is used when OPC client specifies OPC UpdateRate as “0”.
Timer Tick for Async Update Mode	10 (ms)	Need	Specifies the interval for notification of data changes 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 UA Server Setting	-	-	Refer to 5.13.2.1
OPC UA Client Setting	-	-	Refer to 5.13.2.2
OPC UA Publisher Settings	-	-	Refer to 5.13.2.4

[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.

5.13.2.1 Common Properties, OPC, OPC UA Server Setting

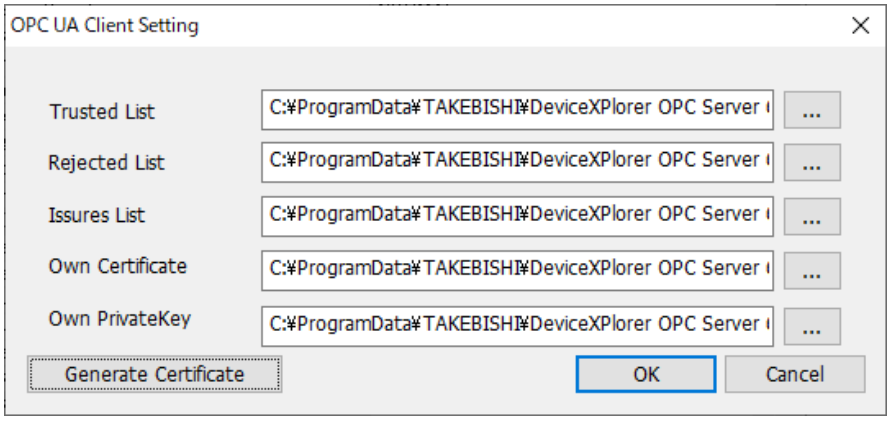


Item	Default	Reboot	Description
Enable UA Server	Disable	Need	Select whether you use OPC UA Server function is, or not.
Enable Anonymous	Disable	Need	Select whether to allow anonymous connections.
Port No	52240	Need	Specify the port number.
Transaction ThreadPoolSize	512	Need	Specify the number of transactions that can be accepted simultaneously.
Security Policy/ Security Mode			
None	Disable	Need	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 ¥TAKEBISHI¥ DeviceXPlorer OPC Server 6¥pkiserver¥tru sted	Need	Specify the directory where trusted certificates are stored.
Rejected List	C:¥ProgramData ¥TAKEBISHI¥ DeviceXPlorer OPC Server 6¥pkiserver¥reje cted	Need	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 ¥TAKEBISHI¥ DeviceXPlorer OPC Server 6¥pkiserver¥issu ers	Need	Specify the directory for storing the certificate and revocation list file (CRL).
Trusted(User Authentication) List*1	C:¥ProgramData ¥TAKEBISHI¥ DeviceXPlorer OPC Server 6¥pkiserver¥use r_trusted	Need	Specify the directory to store the certificate when connecting using an X509 authenticated user.
Issuers(User Authentication) List*1	C:¥ProgramData ¥TAKEBISHI¥ DeviceXPlorer OPC Server 6¥pkiserver¥use r_issuers	Need	Specify the directory for storing the revocation list file (CRL) when connecting using an X509 authenticated user.
Own Certificate	C:¥ProgramData ¥TAKEBISHI¥ DeviceXPlorer OPC Server 6¥pkiserver¥ow n¥certs¥DxpOpc UaServer.der	Need	Specify own certificate path.
Own PrivateKey	C:¥ProgramData ¥TAKEBISHI¥ DeviceXPlorer OPC Server 6¥pkiserver¥ow n¥private¥DxpO pcUaServer.pem	Need	Specify own private key path.
Generate Certificate	—	—	Refer to 5.13.2.3.

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

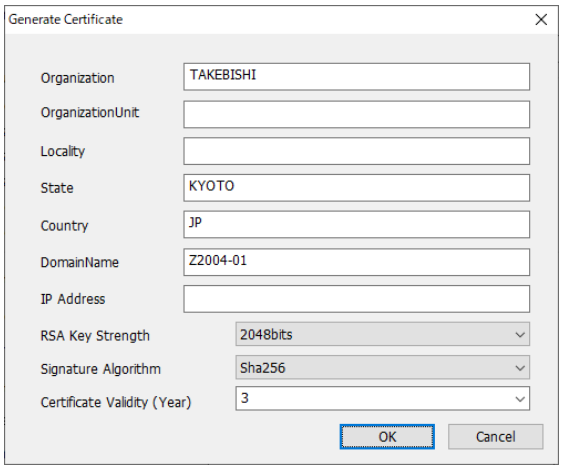
5.13.2.2 Common Properties, OPC, OPC UA Client Setting



Item	Default	Reboot	Description
Trusted List*1	C:¥ProgramData¥TAKEBISHI¥DeviceXPlorer OPC Server 6¥pkiclient¥trusted	Need	Specify the directory where trusted certificates are stored.
Rejected List	C:¥ProgramData¥TAKEBISHI¥DeviceXPlorer OPC Server 6¥pkiclient¥rejected	Need	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¥TAKEBISHI¥DeviceXPlorer OPC Server 6¥pkiclient¥issuers¥certs	Need	Specify the directory for storing the certificate and revocation list file (CRL).
Own Certificate	C:¥ProgramData¥TAKEBISHI¥DeviceXPlorer OPC Server 6¥pkiclient¥own¥certs¥DxpOpcUaClient.der	Need	Specify own certificate path.
Own PrivateKey	C:¥ProgramData¥TAKEBISHI¥DeviceXPlorer OPC Server 6¥pkiclient¥own¥private¥DxpOpcUaClient.pem	Need	Specify own private key path.
Generate Certificate	—	Need	Refer to 5.13.2.3.

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

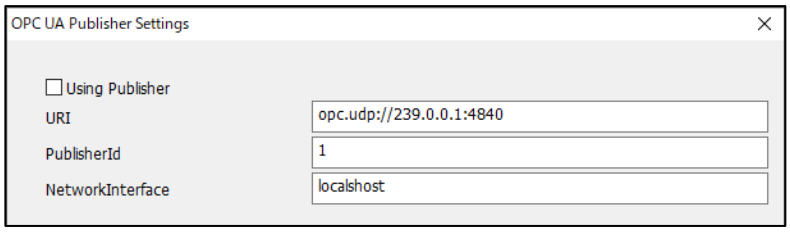
5.13.2.3 Common Properties, OPC, OPC UA Server/Client Setting, Generate Certificate



Item	Description
Organization	Specify the organization.
OrganizationUnit	Specify the organization unit.
Locality	Specify the locality.
State	Specify the state.
Country	Specify the country.
DomainName	Specify the domain name.
IP Address	Specify the IP address.
RSA Key Strength	Select the RSA Key Strength.
Signature Algorithm	Select the Signature Algorithm.
Certificate Validity (Year)	Specify the Certificate Validity.

* A certificate is generated when a certificate does not exist when the application starts.

5.13.2.4 Common Properties, OPC, OPC UA Publisher Settings



Item	Default	Reboot	Description
Using Publisher	Disable	Need	Set whether to use the publisher.
URI	opc.udp://239.0.0.1:4840	---	Specify the multicast address.
PublisherId	1	---	Specify the PublisherId.
NetworkInterface	localhost	---	Specify the interface for publishing to the specified multicast address.

* Please contact us if you wish to use the OPC UA publisher settings.

5.13.3 Common Properties, SuiteLink

Item	Default	Reboot	Description
SuiteLink Application Name	DXP	Need	Specifies the SuiteLink application name.
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.

5.13.4 Common Properties, DxpLink

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

5.13.5 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".
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.
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".
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.
Shared Tag Memory	Use	Need	The shared tag memory can be used for sharing data among applications and as a work register. If the shared tag memory is not used, it reduces memory consumption. You can specify "Use" or "Not Use".
Points of Shared Tag Memory	100 (K points)	Need	If the shared tag memory is used, this specifies the amount of data (device points) for each shared tag. By changing the number of tag points for each data type, you can reduce memory consumption. Between 1K and 100K points can be set.
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.

5.13.6 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”.
Put in Task Tray when minimized	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

5.13.7 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	0 (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".
Output Information Message	Output	---	Outputs information messages. You can specify "Output" or "Not Output".
Output Warning Message	Output	---	Outputs warning messages. You can specify "Output" or "Not Output".
Output ERROR Message	Output	---	Outputs error messages. You can specify "Output" or "Not Output".
Output FATAL ERROR Message	Output	---	Outputs fatal error messages. You can specify "Output" or "Not Output".
Dump Script Log	No Output	---	Outputs Script Log. You can specify "Output" or "Not Output".
Output Communication Message	No Output	---	Traces communication messages. You can specify "Output" or "Not Output".
Output DxpLink	No Output	---	Outputs DxpLink Interface. You can specify "Output" or "Not Output".
Output SuiteLink	No Output	---	Outputs SuiteLink Interface. You can specify "Output" or "Not Output".
Output OPC Interface Access Message	No Output	---	Traces the access status of the OPC interface. You can specify "Output" or "Not Output".
Detail OPC Message	Enable	---	Specifies whether to save OPC Interface Detail Message automatically. If this setting is "Enable", OPC Interface Detail Message is outputted to log file and Message View. You can specify "Enable" or "Disable".

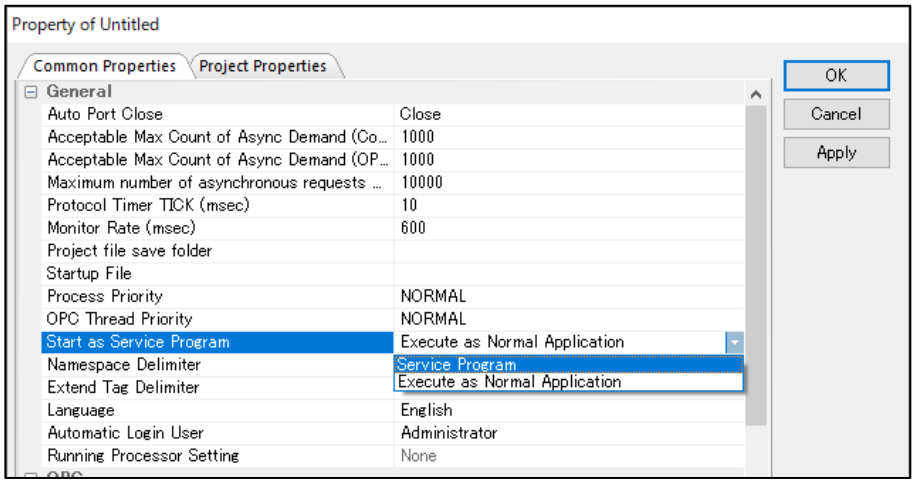
5.14 Service Activation

DeviceXPlorer can be started as a Windows service program.

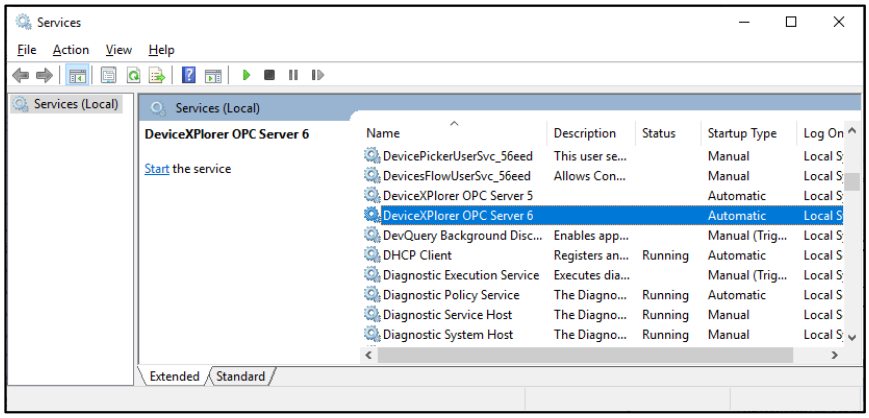
Important
Please configure after start as “Run as administrator” when UAC is ON in on Windows Vista or later OS. DeviceXPlorer runs on the service session (session 0) of Windows when you use service in Windows Vista or later. SuiteLink clients running on other sessions such as desktop cannot connect 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.

5.14.1 Registering DeviceXPlorer as a service program

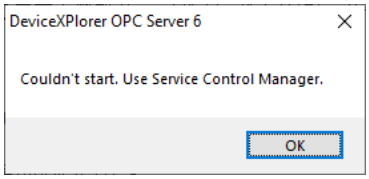
(1) Go to Tools -> Options, specify “Startup file”, and set “Start as Service Program” to “Service Program”.



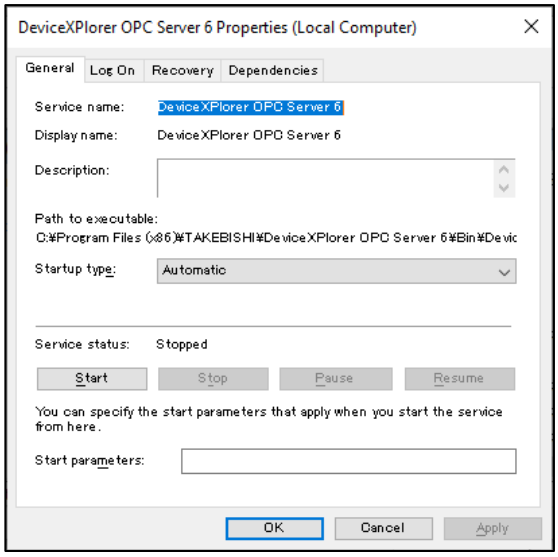
(2) When you exit DeviceXPlorer, it will be registered as a service program.
You can check running status of “DeviceXPlorer OPC Server 6” as service program from "Services" of "Administration Tool" in Control Panel.



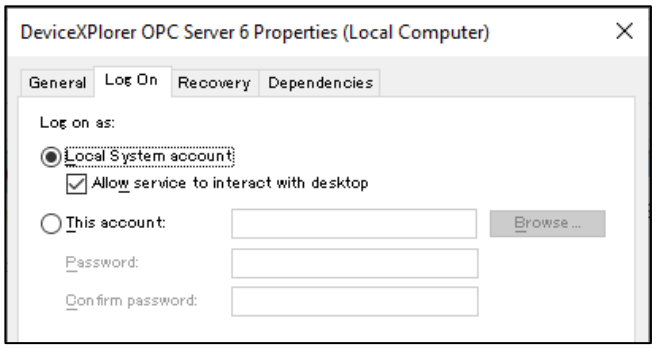
If you don't launch DeviceXPlorer with "Run as administrator" from right-click menu, following error message will show.
This is error and DeviceXPlorer doesn't became service application.



(3) From Control Panel -> Services, open the Properties window for “DeviceXPlorer OPC Server 6”.



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

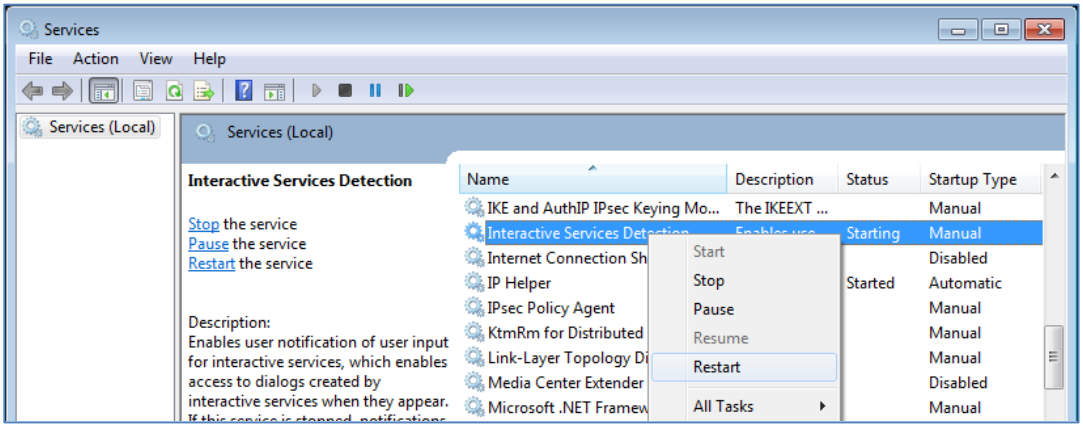


(5) When you reboot the PC, DeviceXPlorer will operate as a service program.

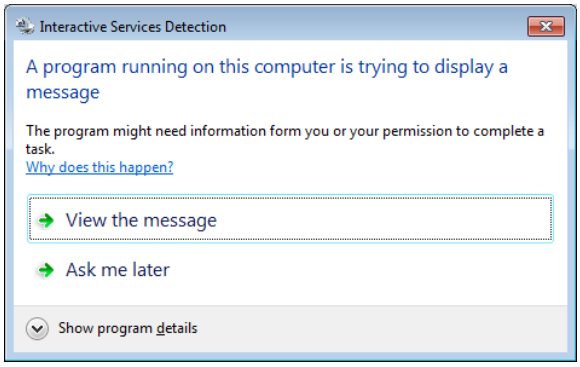
Important
- When connecting with OPC server which runs as windows service, the DCOM configuration is required. About the DCOM configurations, please refer to 5.22. - In Windows10, please clear the check box for "Allow service to interact with desktop".

5.14.2 How to show DeviceXPlorer setting dialog of service program

On Windows Vista or later OS, service program could not be displayed. So, if you would like to show DeviceXPlorer setting, please start or restart "Interactive Services Detection" from right-click menu.



Following dialog will be shown and please click "View the message".

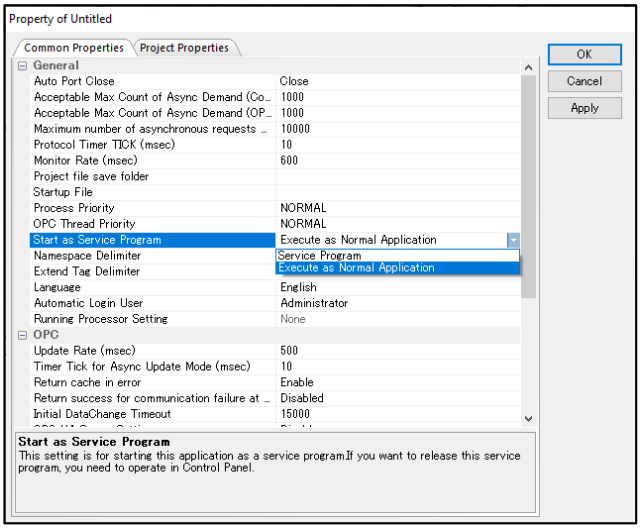


Important

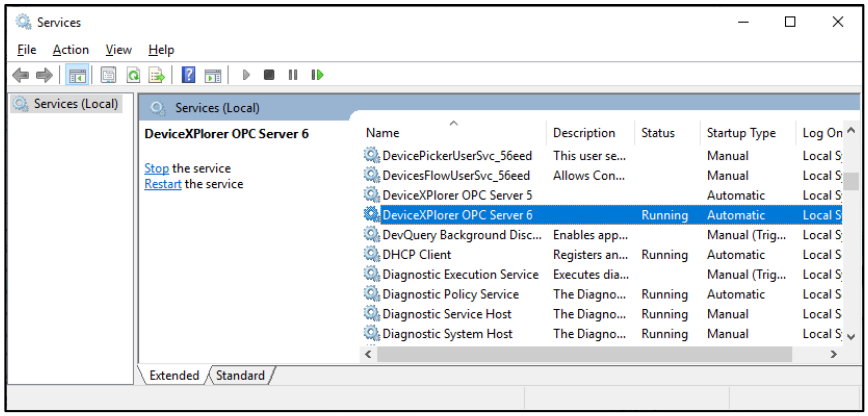
- In Windows10, the service programs don't support to display the window by detection of the interactive service. DeviceXPlorer can't be able to display the screen even if you started the detection of the interactive service by the above procedure.

5.14.3 Unregister from service program

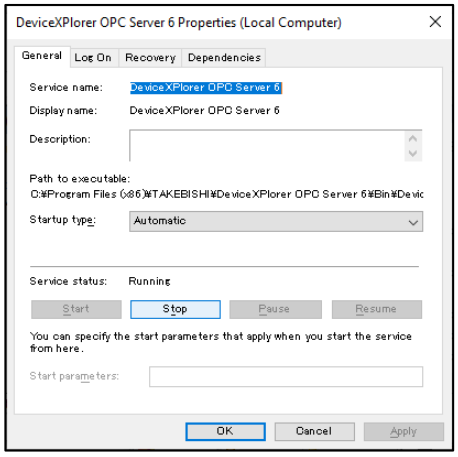
(1) Go to Tools -> Options, set “Start as Service Program” to “Execute as Normal Application”.



(2) From Control Panel -> Services, open the Properties window for “DeviceXPlorer OPC Server 6”.



(3) And stop the service of “DeviceXPlorer OPC Server 6”.



5.14.4 Unregister from service program on Windows 10 later

(1) Stop the service of “DeviceXPlorer OPC Server 6” from "service" of the control panel.
(2) Run the command prompt as an administrator and go to the installation directory.

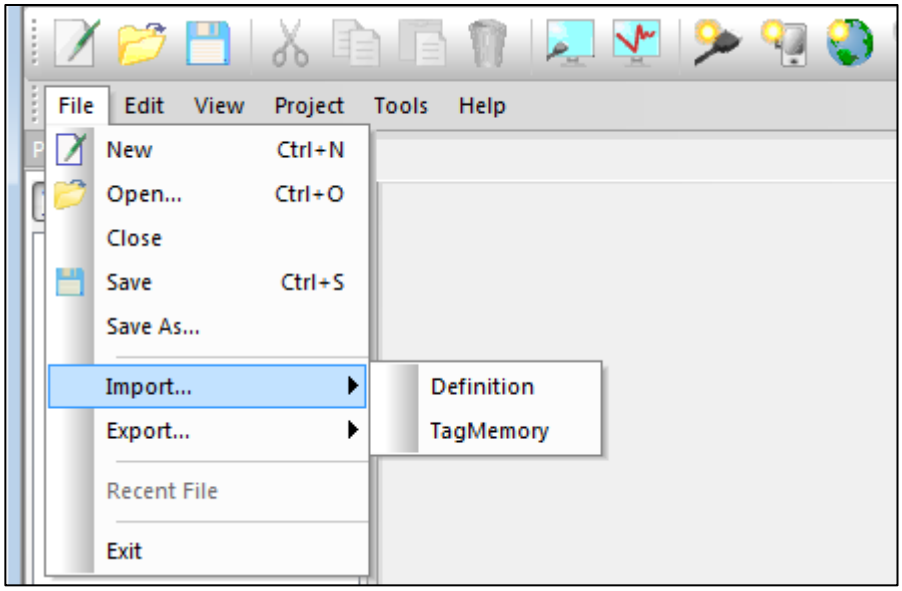
```
C:¥Users¥seigi>cd C:¥Program Files (x86)¥TAKEBISHI¥DeviceXPlorer OPC Server 6¥Bin
```

(3) Enter "DeviceXPlorer / RegServer" and execute it.

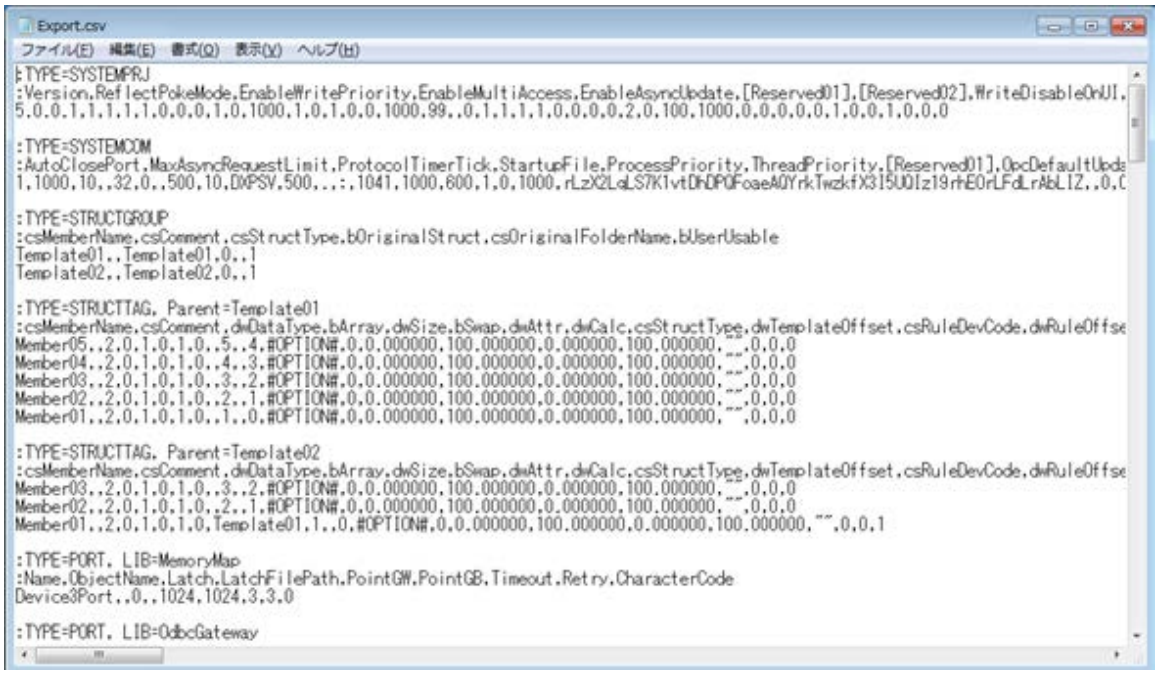
```
C:¥Program Files (x86)¥TAKEBISHI¥DeviceXPlorer OPC Server 6¥Bin>DeviceXPlorer/RegServer
```

5.15 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.



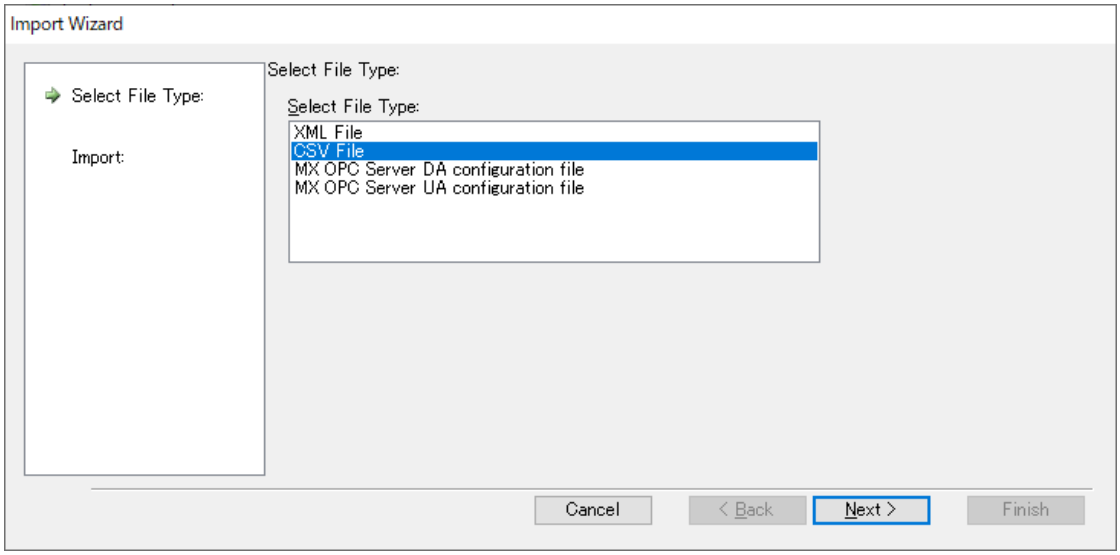
Important
This is can Import and Export file create at Ver.4. But this cannot import CSV file is created by before Ver.3 OPC Server. When CSV file created by Ver.3, please convert at Project Converter[All Program]-[DeviceXPlorer OPC Server 6] - [Project Converter].



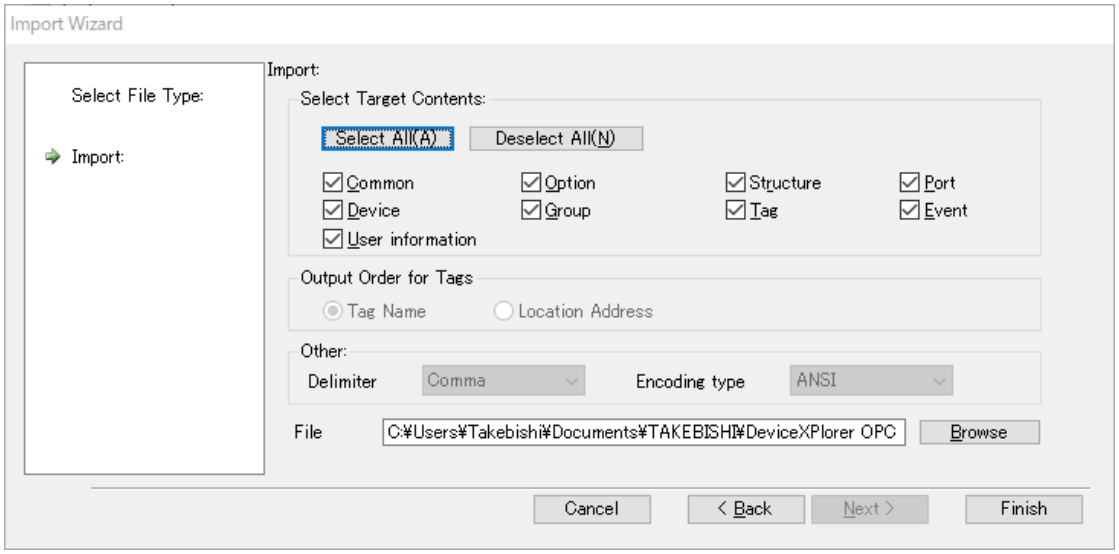
Note
When edit to file , please careful attention.

5.15.1 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.



5.15.1.1 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 Ethenet 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).

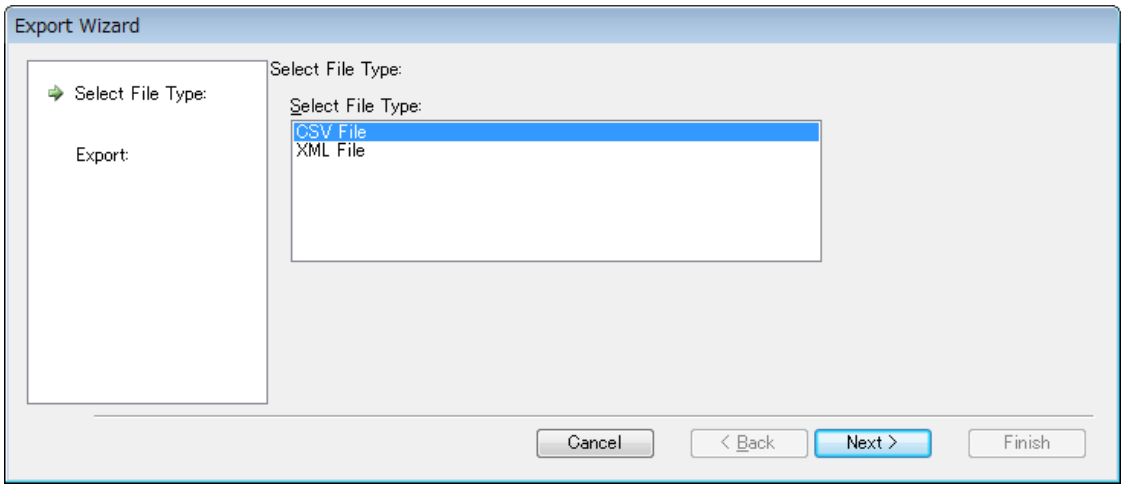
In MX OPC Server, the BOOL array tag is 1 bit per element. (In DxpSERVER , BOOL array tag is 16 bits per element)
When importing a BOOL array tag, convert it to cover all the devices with the BOOL type array tag set in MX OPC Server .

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)

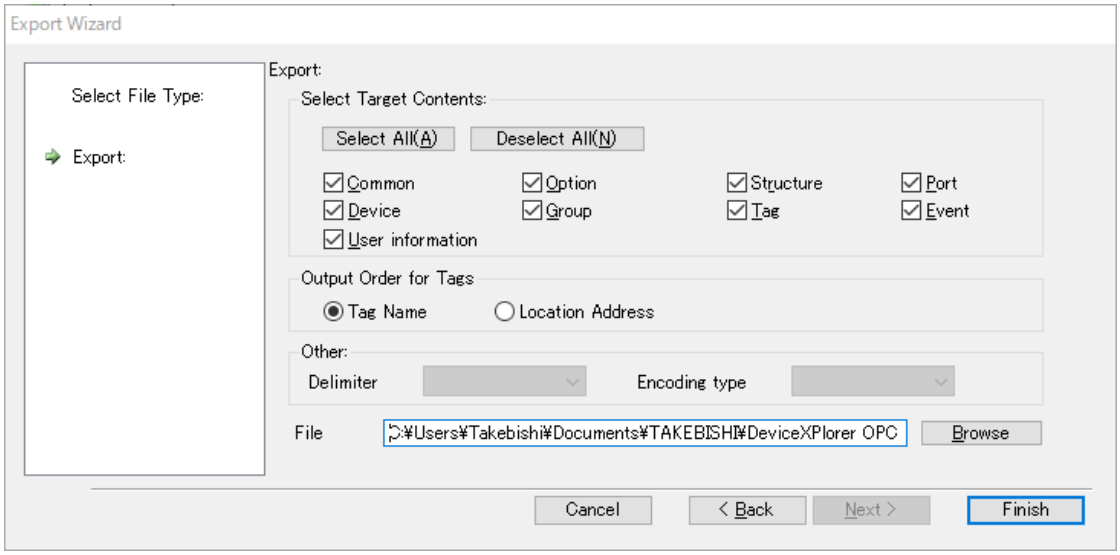
Example 2:
MX OPC Server (before conversion): D0.F Bool Array size 3 Read range: D0.F to D1.1 (3bit)
DxpSERVER (after conversion): D0 Bool Array size 2 Read range: D0.0 to D1.F (32bit)

5.15.2 Export[Definition]

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

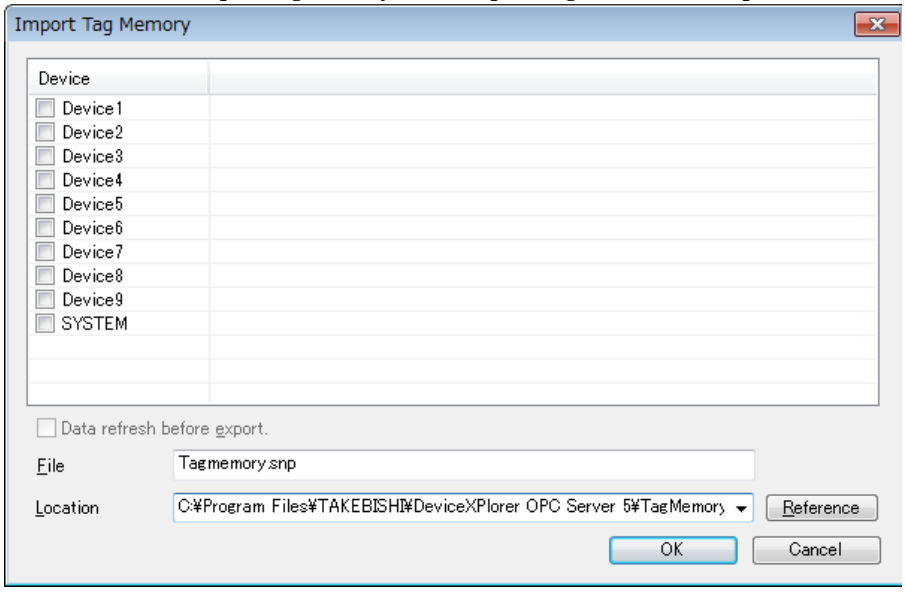


Select target contents and file path to export. And it can specify output order for tags as tag name or location address.

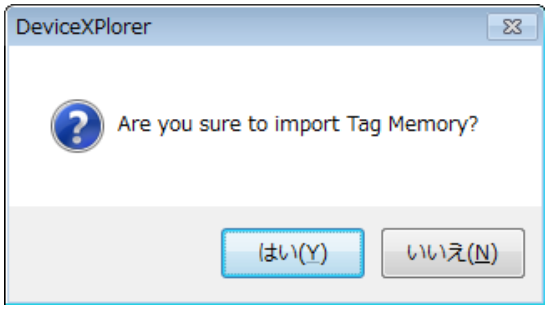


5.15.3 Import[TagMemory]

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

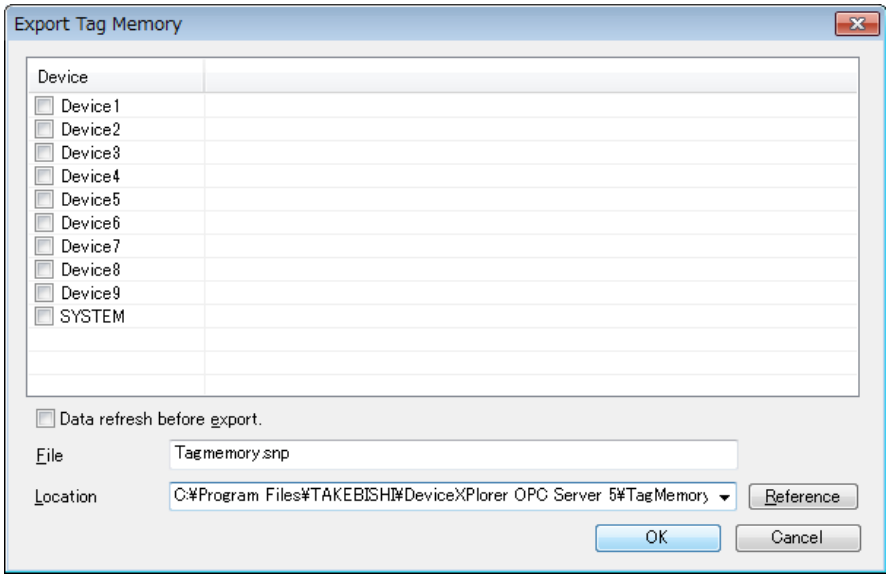


When importing, confirmation will be displayed.



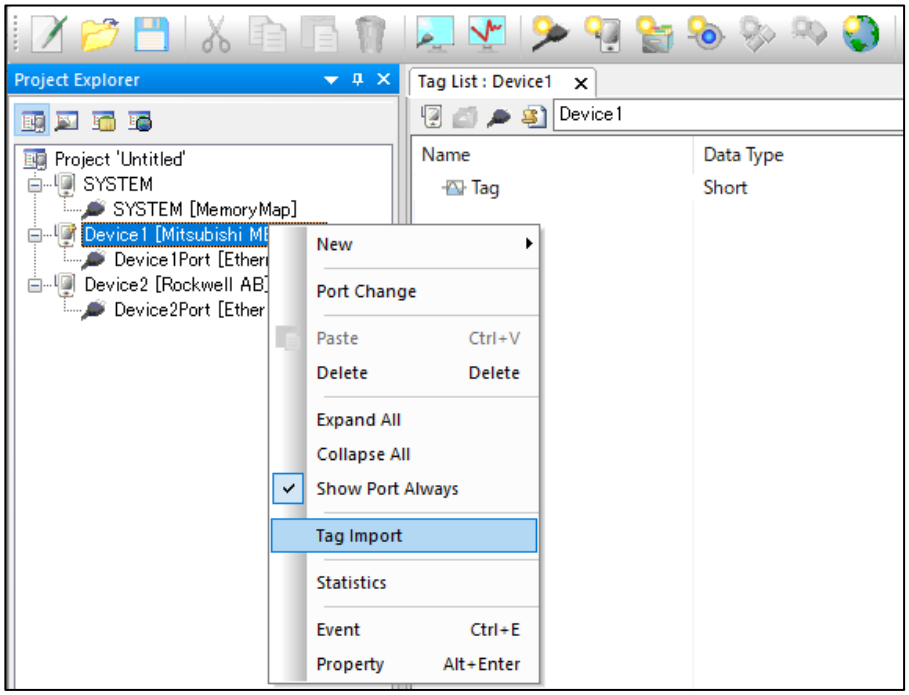
5.15.4 Export[TagMemory]

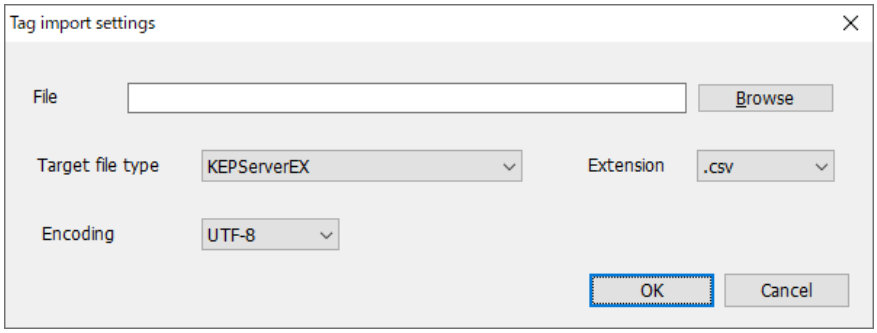
Export TagMemory(cache, timestamp, quality) that DeviceXPlorer has currently.
Select target device.
If [Data refresh before export.] is checked, DeviceXPlorer updates all data to the latest value before exporting TagMemory.



5.15.5 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".

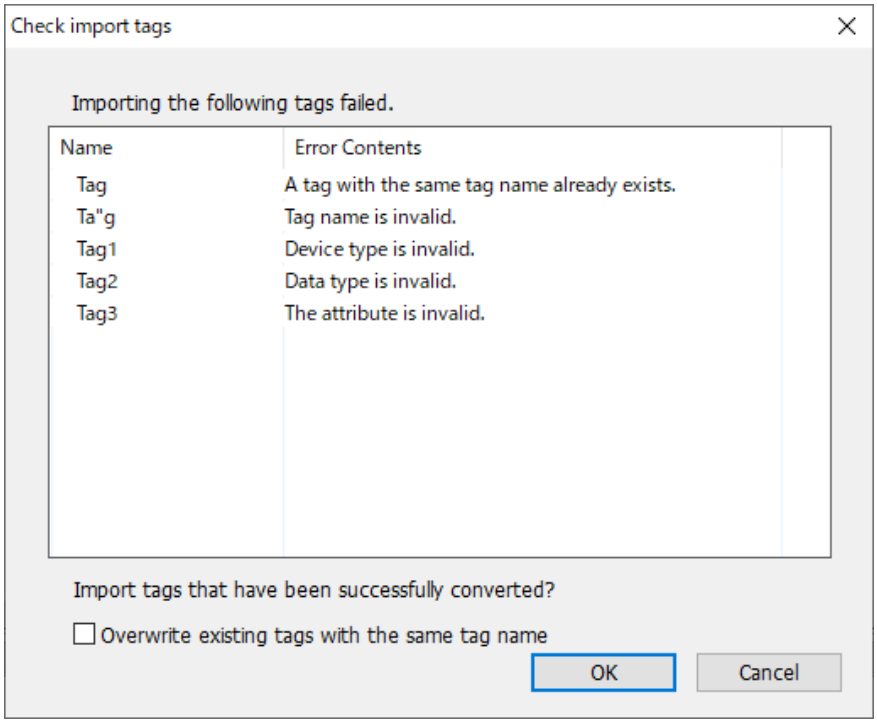




Items	Description
File	Select the tag definition file.
Target file type	Select the type of the target file.
Extension	Select the Extension of the target file.
Encoding	Select the Encoding of the target file.

5.15.5.1 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 "5.15.5.2 Error List" 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]

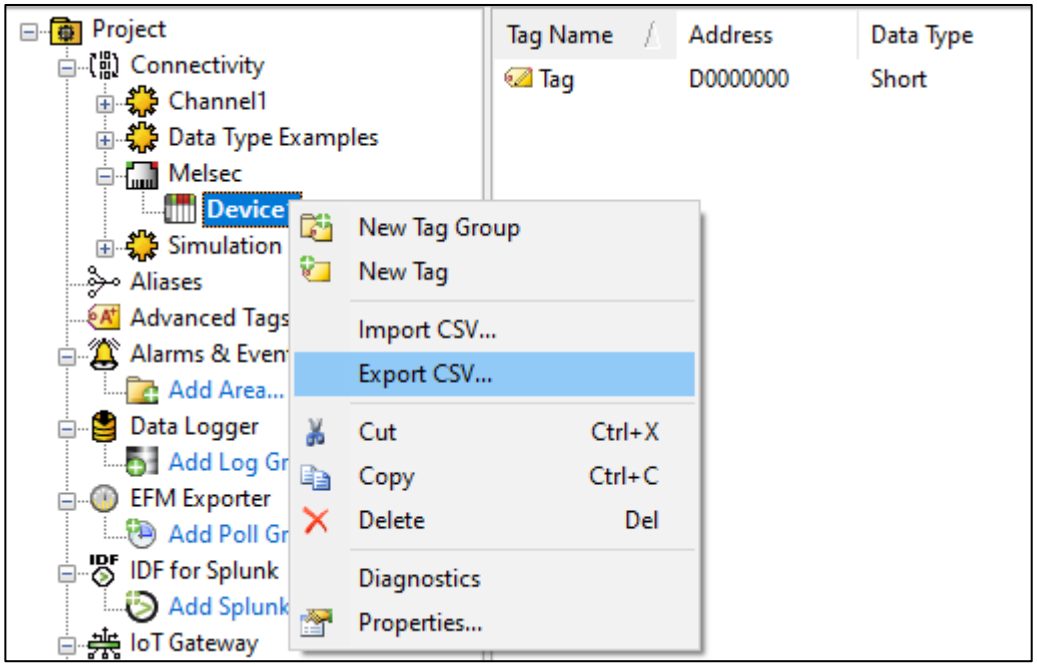
Data Type		Description
KEPServerEX	DeviceXPlorer	
Boolean	BOOL	1 bit *1
Char	BYTE	8 bit signed integers data
Byte	UBYTE	8 bit unsigned integers data
Short	SHORT	16 bit signed integers data
Word	USHORT	16 bit unsigned integers data
Long	LONG	32 bit signed integers data
DWord	ULONG	32 bit unsigned integers 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).

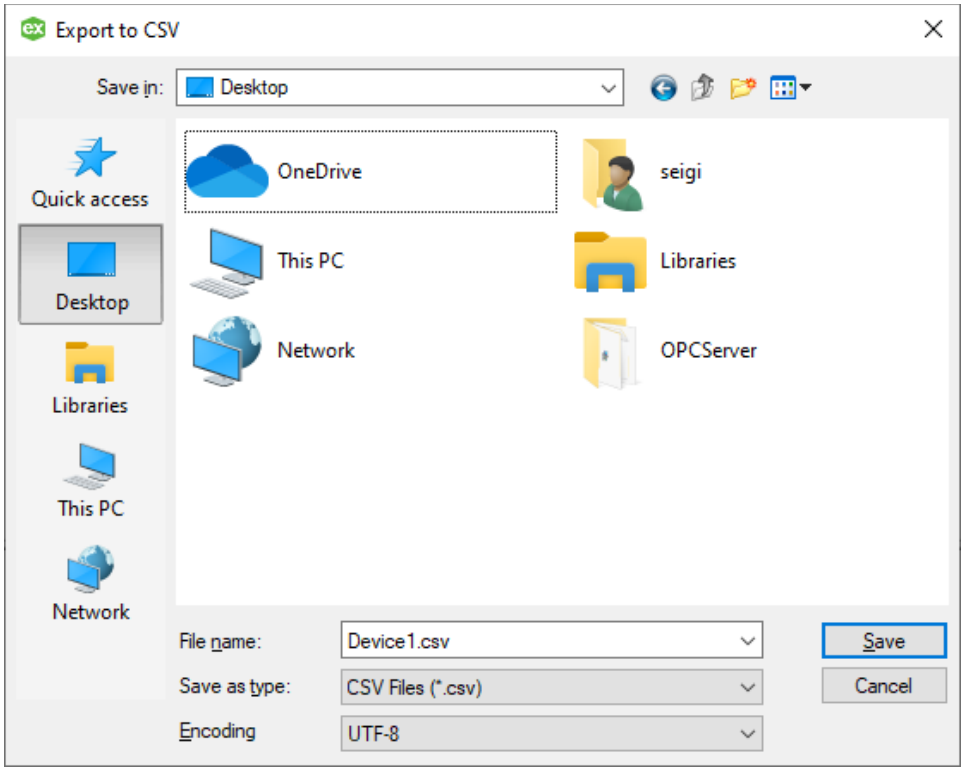
*2 Array tag is not supported.

[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".



5.15.5.2 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.

5.16 Hot Configuration

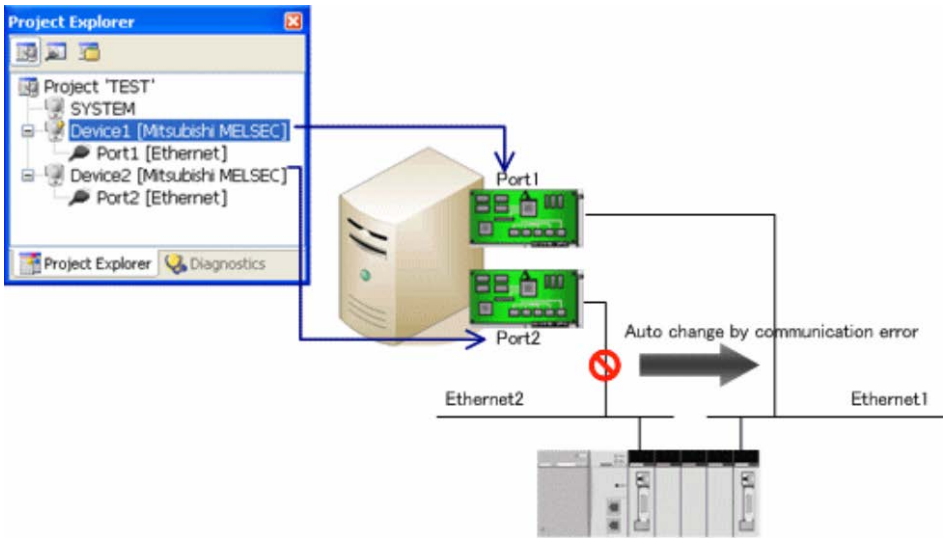
The DeviceXPlorer can change a Device settings and Tag settings also in the connection from a client.
Notes are indicated to be an item reflected in below by Hot Configuration.

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.

5.17 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.

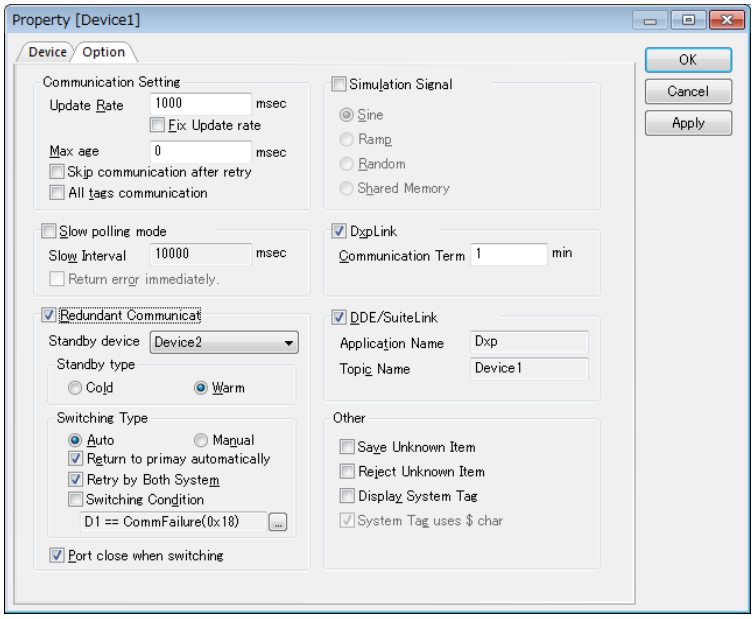


[Settings]

(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 5.4.5.

5.17.1 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.

5.17.2 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.

5.18 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. But if you check "Fix Update Rate" of Device Option, DeviceXPlorer obeys "Update Rate" of Device Option.
OPC UA	
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.

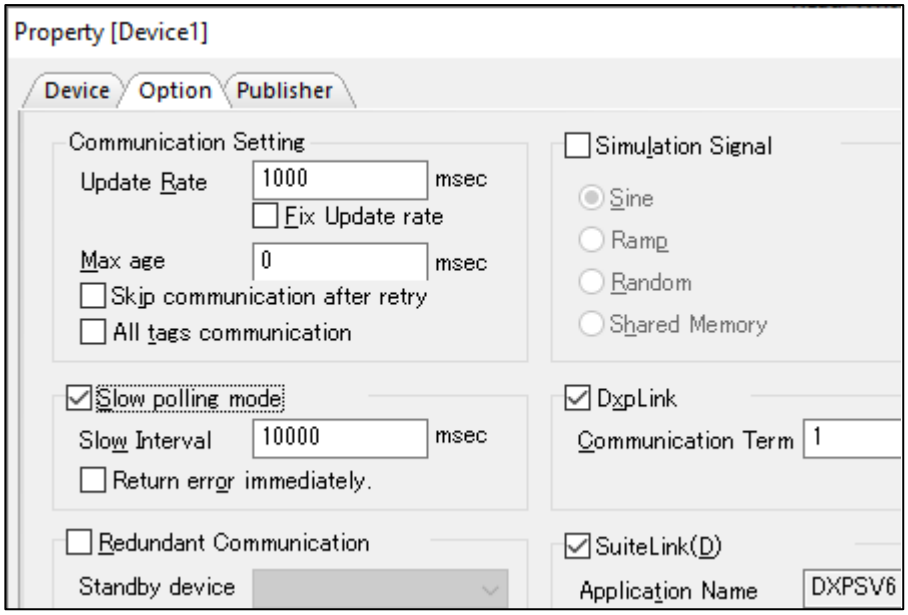
These interfaces perform communications processing a best communication cycle. But to maintain past compatibility, only the OPC DA interface is preparing the mode which does not perform the optimal communications processing. (Client unit communication)
You can configure Client unit communication in "Communication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

5.19 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 5.4.5.

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.

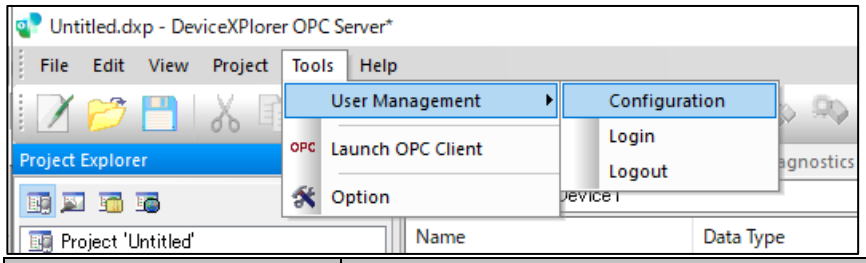
Message	Contents
Entering Slowpolling Mode.	DeviceXPlorer enters under Slow Polling Mode.
Leaving Slowpolling Mode.	DeviceXPlorer leaves from Slow Polling Mode.

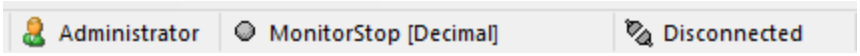
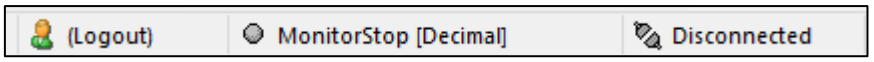
Note
If you'd like to reduce communication time under Slow Polling Mode, please check "Return error immediately".

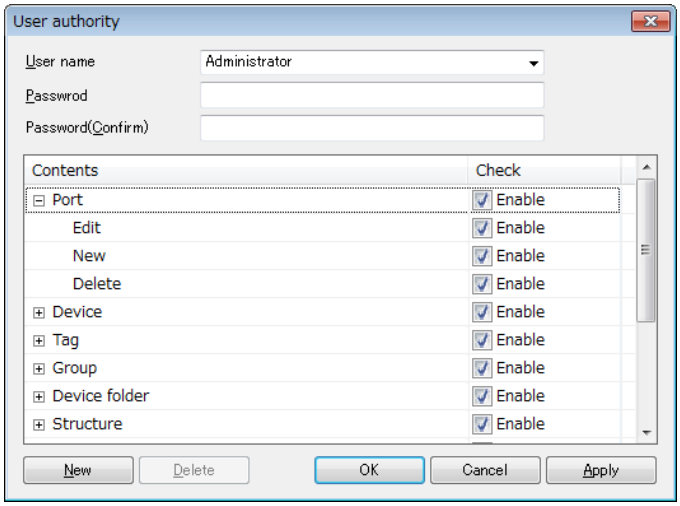
5.20 User Management

5.20.1 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 5.20.2 for explanation of each contents. You cannot modify “Administrator” authority.
New	Create a new user.
Delete	Delete a user. You cannot delete “Administrator”.

5.20.2 Contents List

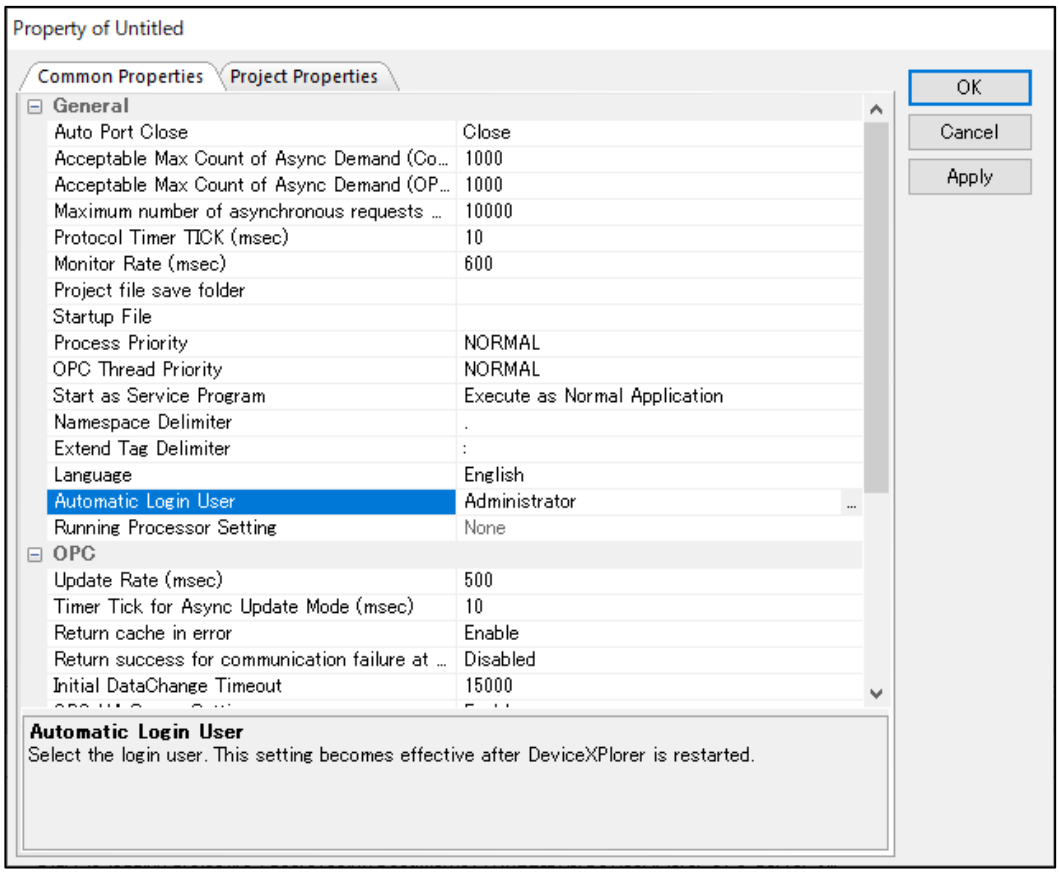
Item		Description
Port	Edit	Enable to configure Port
	New	Enable to create Port
	Delete	Enable to delete Port
Device	Edit	Enable to configure Device
	New	Enable to create Device
	Delete	Enable to delete Device
Tag	Edit	Enable to configure Tag
	New	Enable to create Tag
	Delete	Enable to delete Tag
Group	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

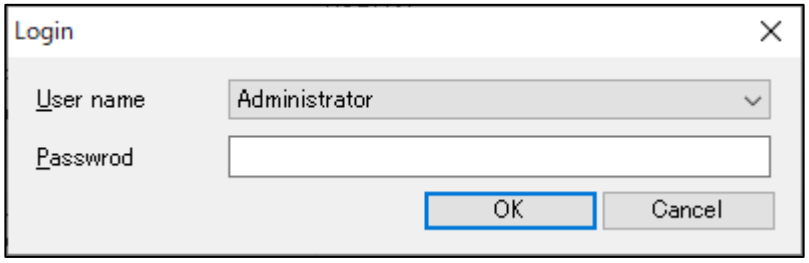
- To configure license, You need to login as “Administrator”
- User authority is invalid to the client at the time of carrying out COM starting. For example, the application which carried out COM starting also in the state of login by the user without the end authority of application carries out the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

5.20.3 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.



Specify User name and password.



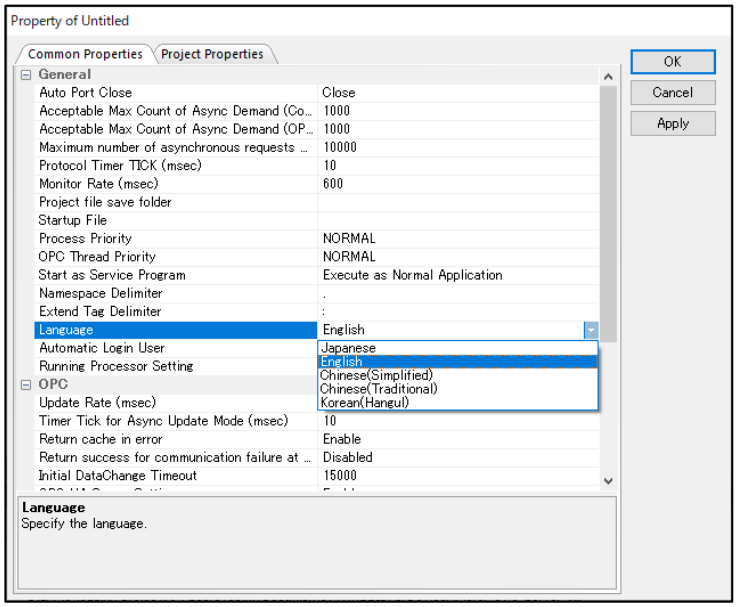
5.21 Change display language dynamically

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

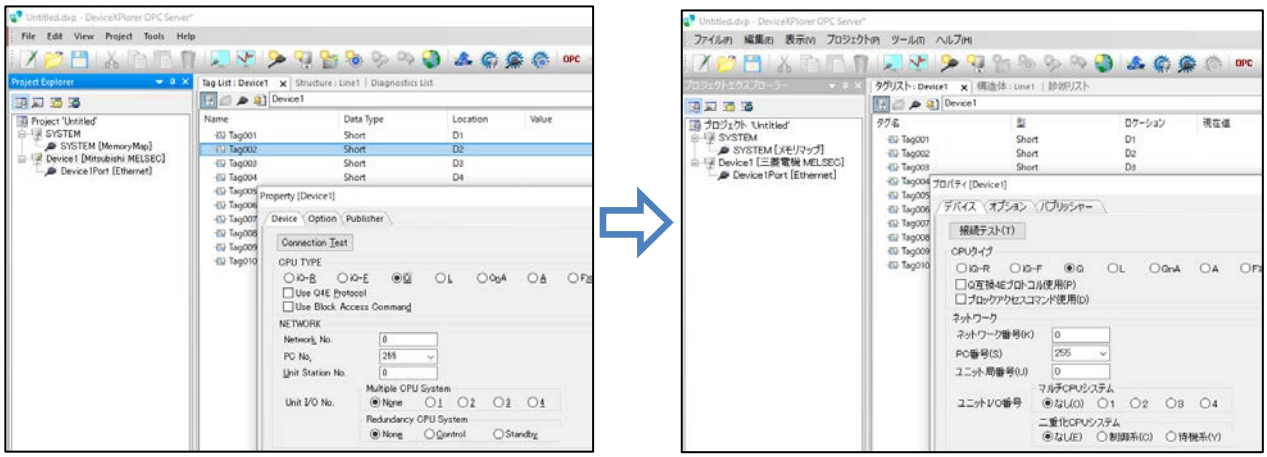
[Settings]

(1) Go to Tools -> Options, specify “Language”, and select other language.

(2) Click “OK”.



(3) DeviceXPlorer changes the language.



5.22 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.

5.22.1 Overview

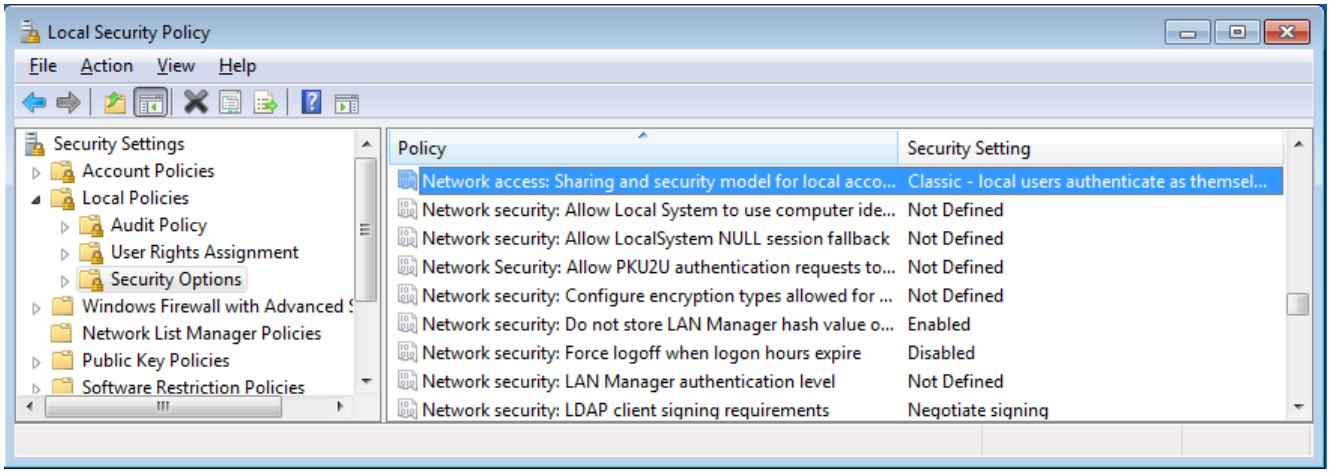
Client user must be authenticated on the machine on which the OPC server is running in order to enable DCOM connection. Authentication of the client machine user is normally done by configuring one user account and one password on both machines running in a work group. From the user account on the client 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.



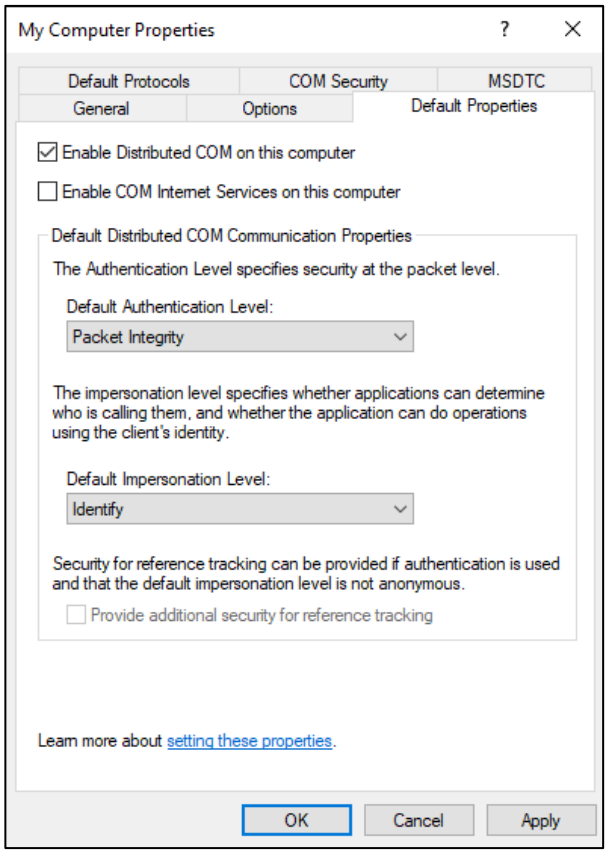
Important
You cannot connect via DCOM if checking the "Windows firewall" in property window of "Local Area Connection" So you should not check this option button. DCOM can be connected by making the firewall effective by setting security.

5.22.2 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.

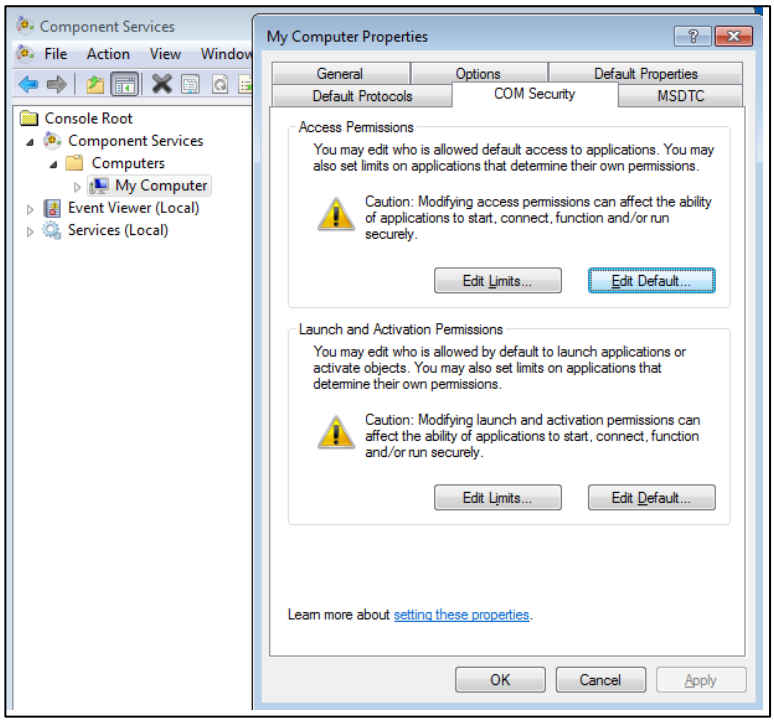
[Default Properties]
Combined utilities start Navigate through "Component Services" -> "Computer" -> "My Computer." Right mouse click on "My Computer," open the Properties page from the pop-up menu, and select the "Default Properties" tab.



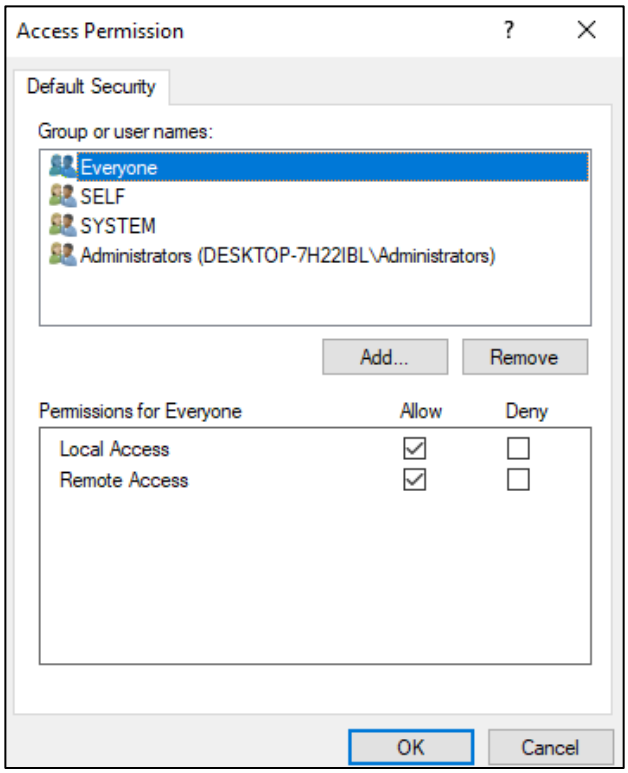
* Check the "Enable Distributed COM on this Computer" check box.
* Set "Default Authentication Level" to "Packet Integrity."
* Set "Default Impersonation Level" to "Identify."

Important
On the client machine, set "Default Authentication Level" to "Packet Integrity."

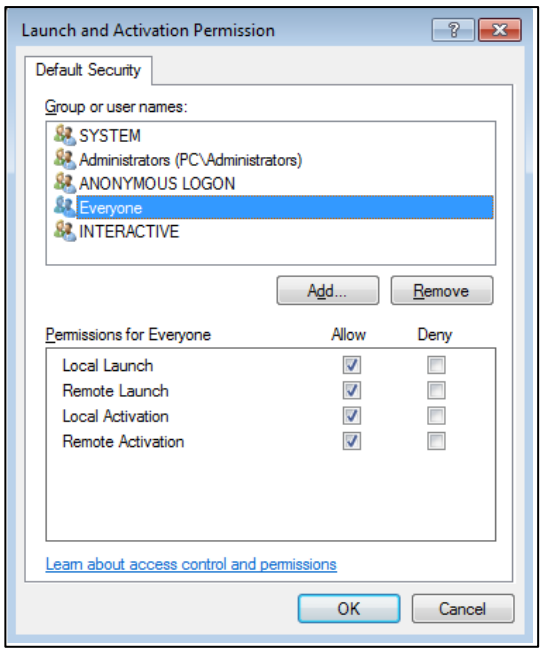
[COM Security]
The change in the DCOM setting is needed.



"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".

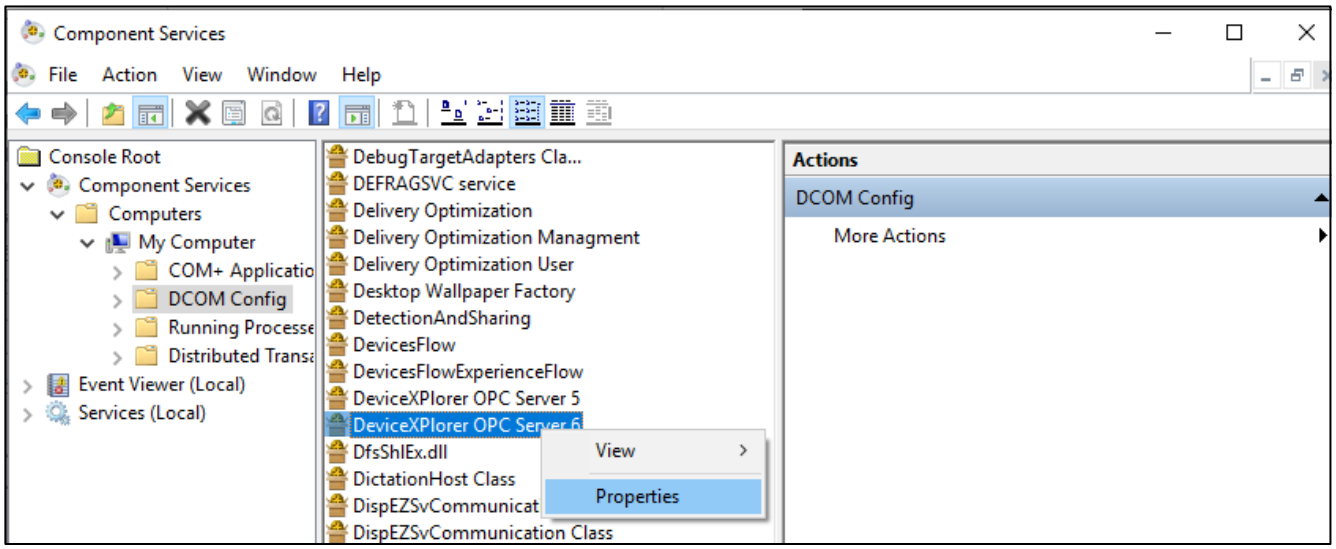


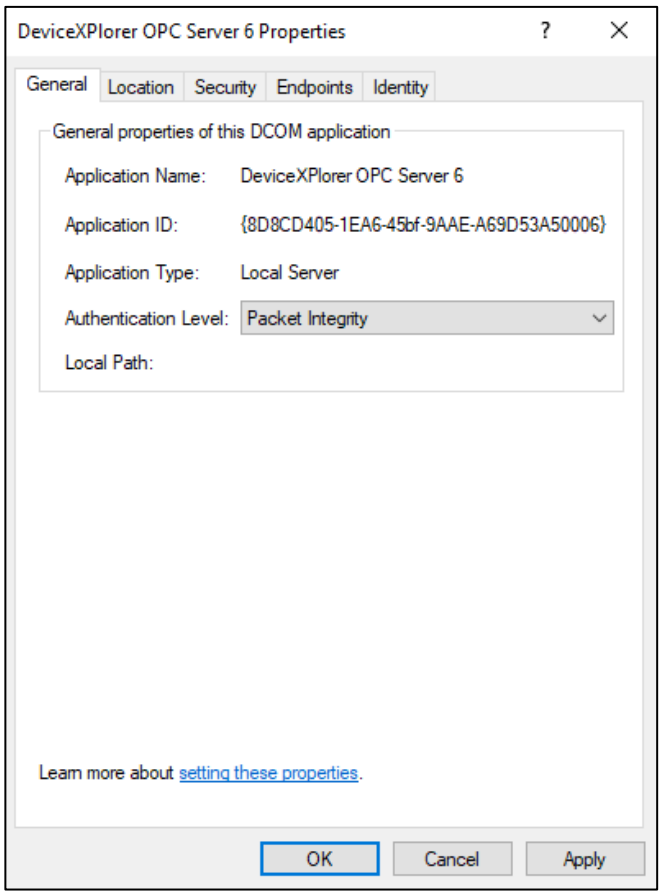
The dialog opens when "Edit limits" button of "Access permit of starting and making actively" is pushed.
"Everyone" and "SYSTEM" is selected, and "Remote Launch" and "Remote Activation" are set to "Allow".

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

[Specific Properties (General)]

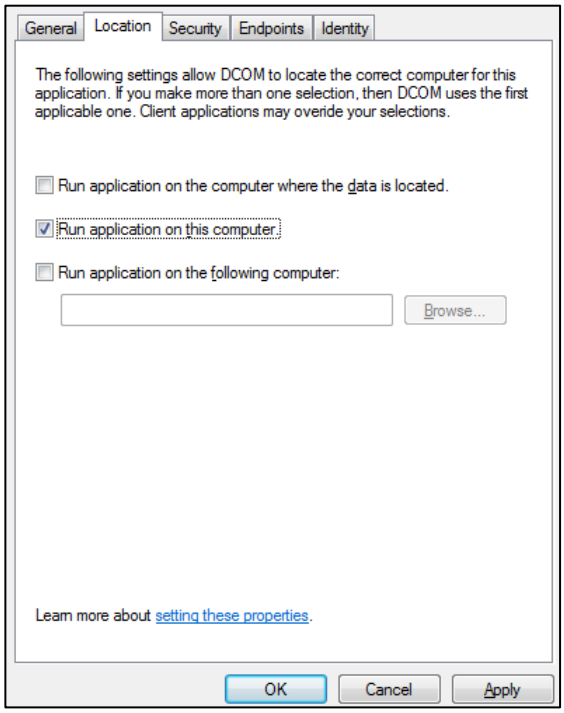
Right mouse click on "DeviceXPlorer OPC Server 6" from "DCOM Config" under "My Computer" ,and select Properties from the pop-up menu.





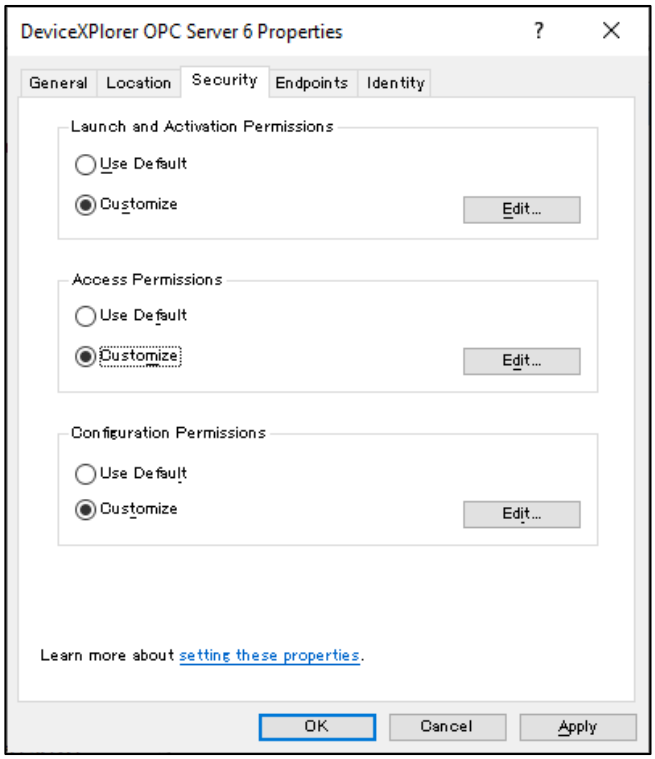
* Set the "Authentication Level" to "Packet Integrity."

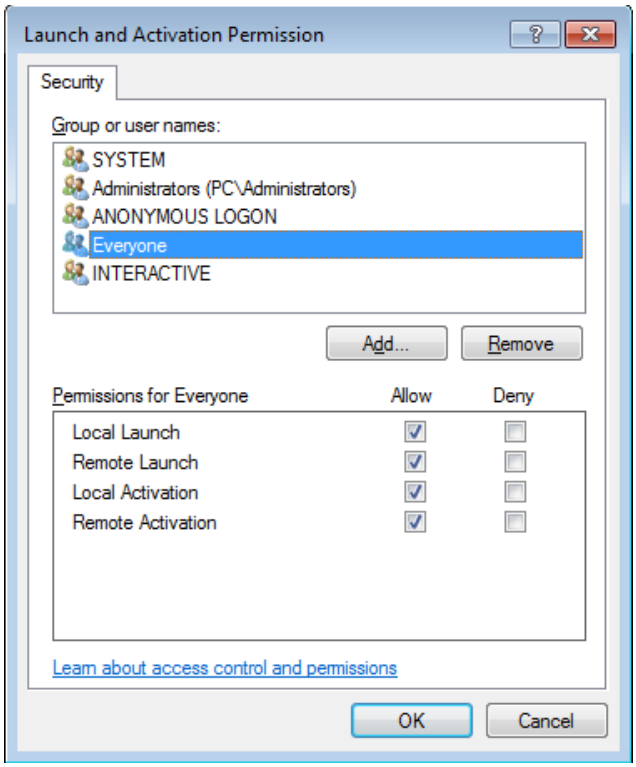
[Specific Properties (Location)]
Select the "Location" tab.

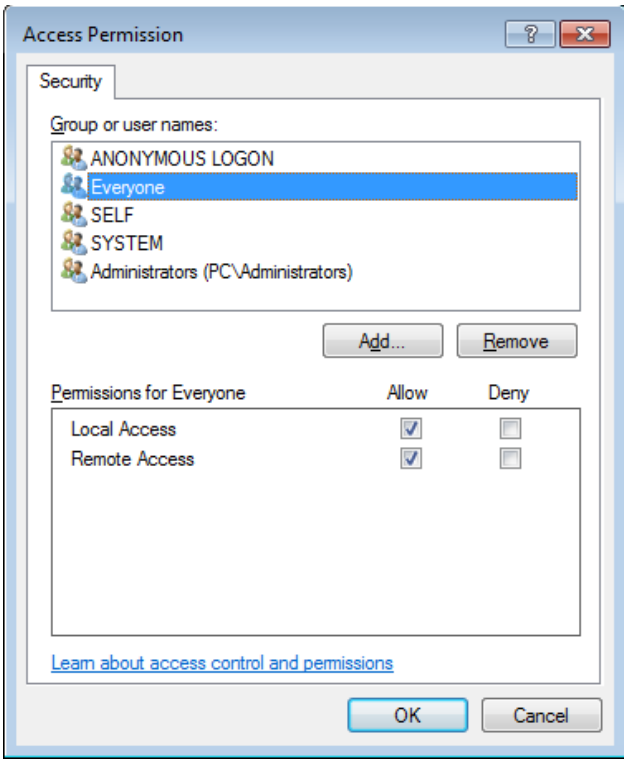


*Check "Run application on this computer."

[Specific Properties (Security)]
Select "Customize" in each field and press "Edit."

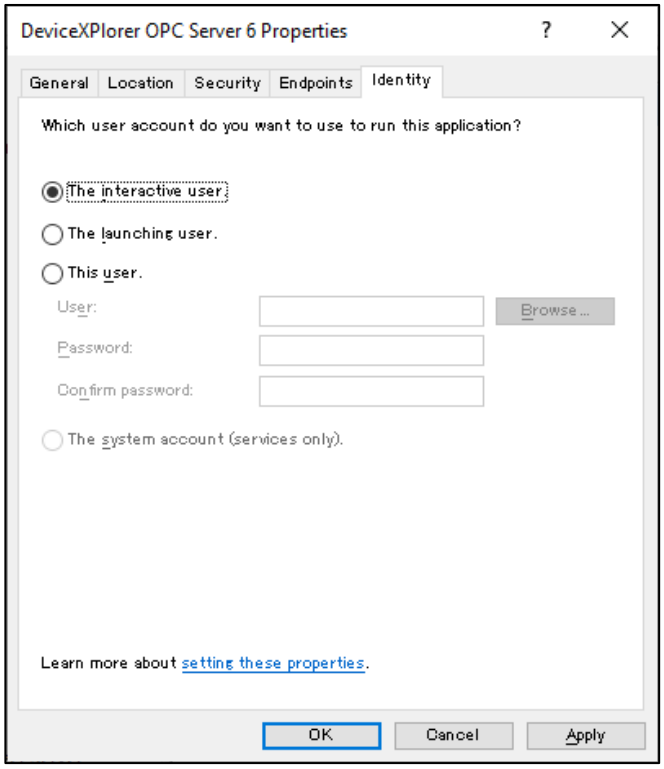






*Add "Everyone" and "SYSTEM" to "launch permissions," "access permissions," and "configuration permissions."

[Specific Properties (ID/Identity)]
Select the "ID" tab, and specify the user account that runs the application.

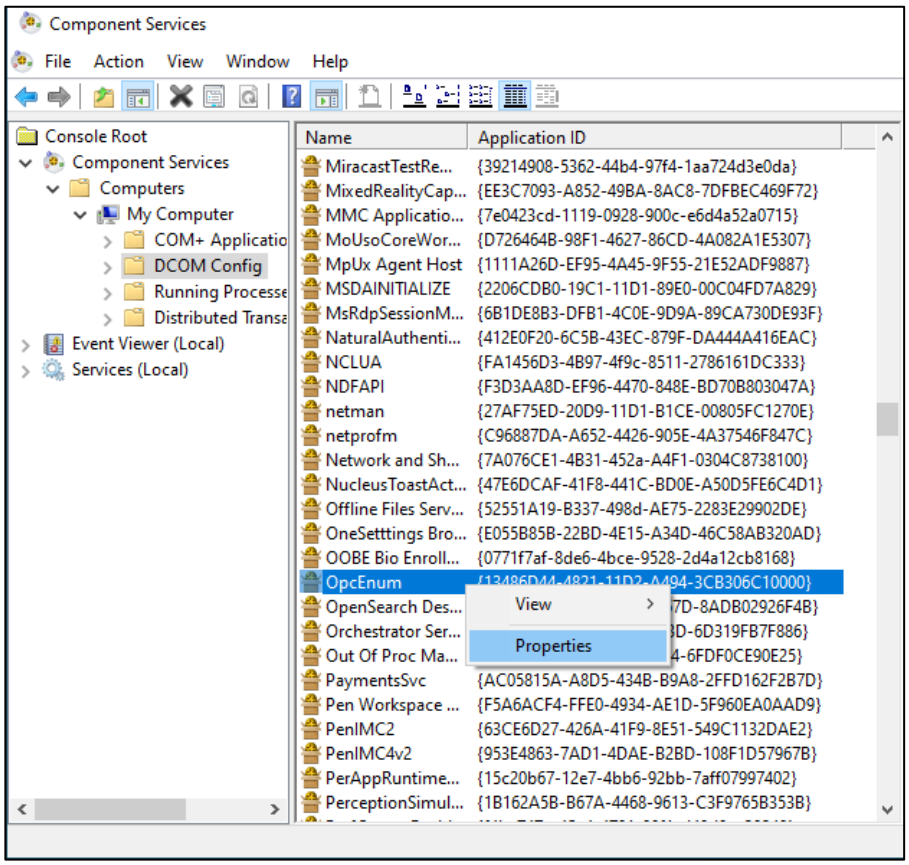


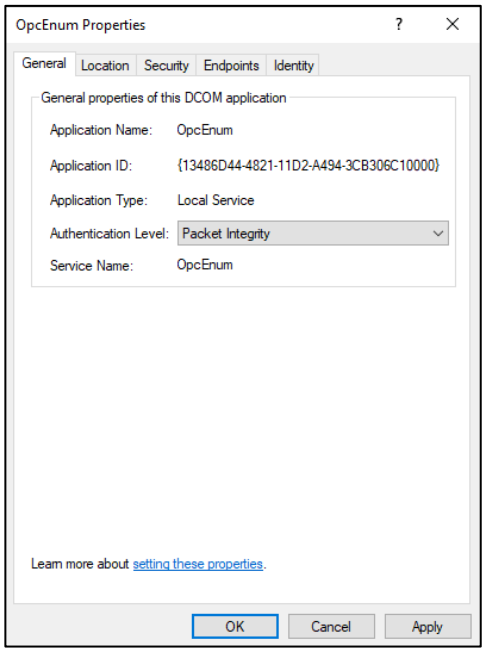
Select "This User" and specify a user that has Administrator rights. Or select "System Account" to run it as a service program. When launching DeviceXPlorer as a service, it becomes a "System Account," and you cannot specify it as an "Interactive User" or "Launching User."

Important
Restart computer with changing DCOM Configuration.

[OPCEnum　Settings]

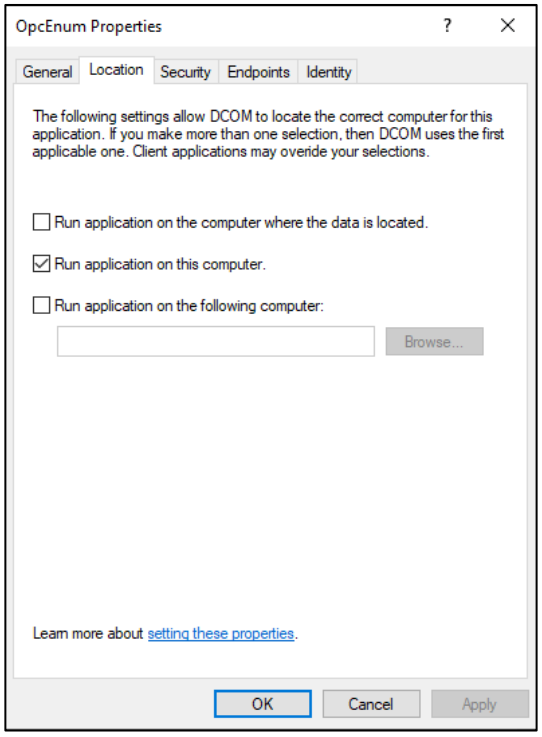
Right mouse click on "OpcEnum" from "DCOM Config" under "My Computer" ,and select Properties from the pop-up menu.





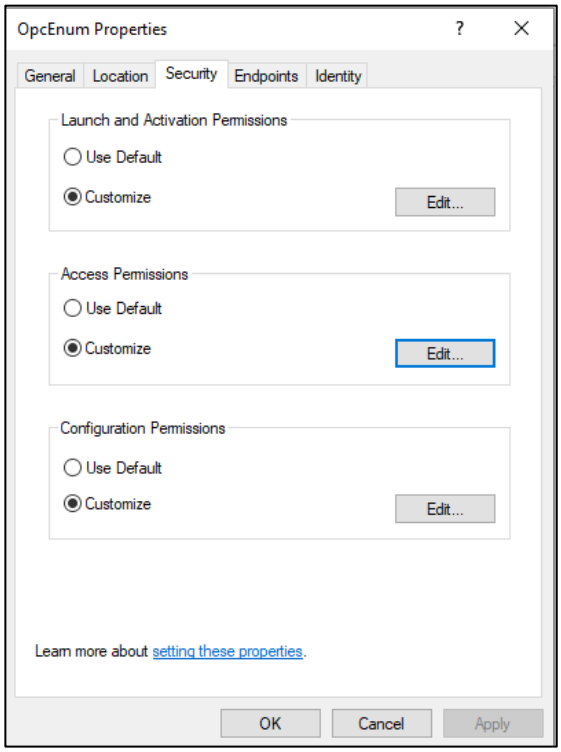
* Set the "Authentication Level" to "Packet Integrity ."

[Specific Properties (Location)]
Select the "Location" tab.

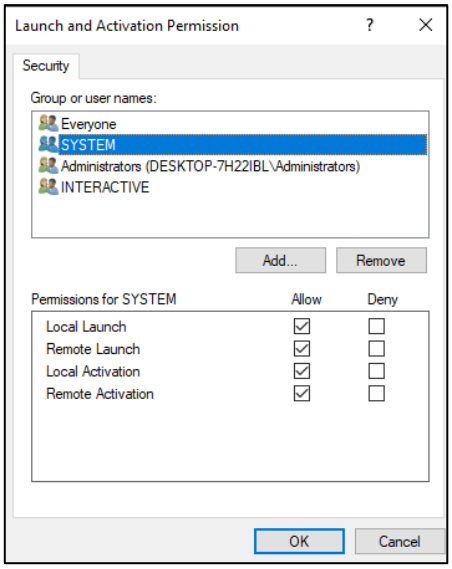


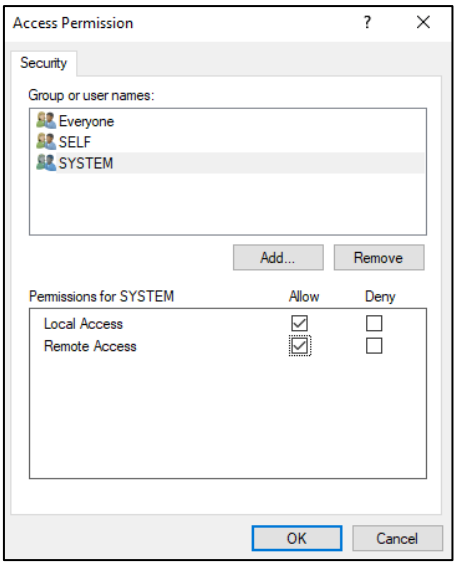
*Check "Run application on this computer."

[Specific Properties (Security)]
Select "Customize" in each field and press "Edit."



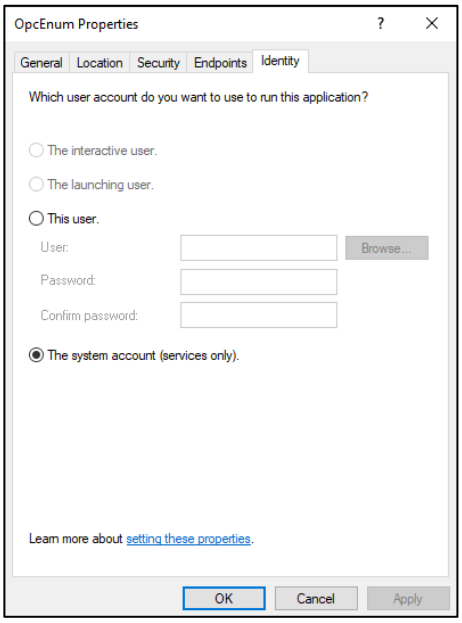
Under "Security," add "SYSTEM" and grant access. (If you cannot connect, add "Evryone" to allow access.)





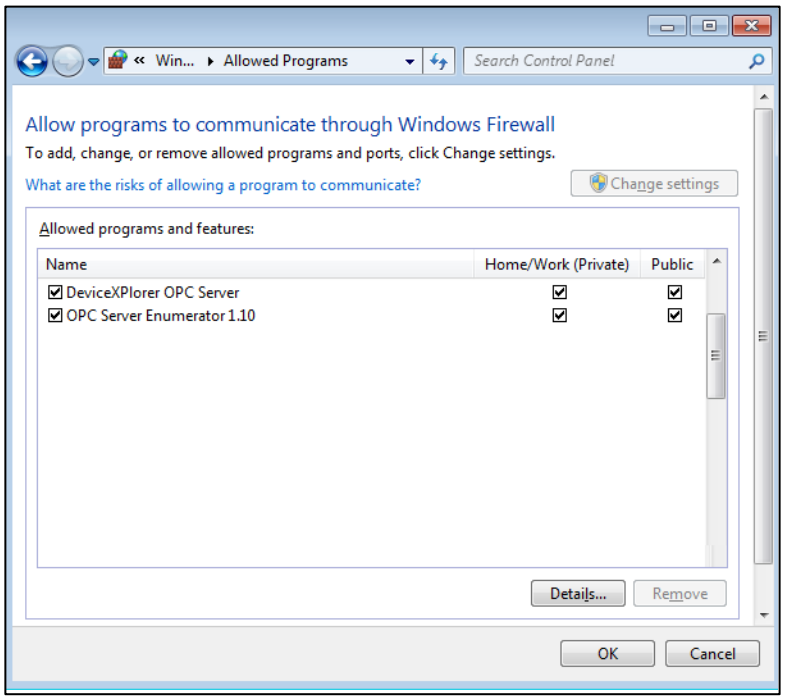
[Specific Properties (ID/Identity)]

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

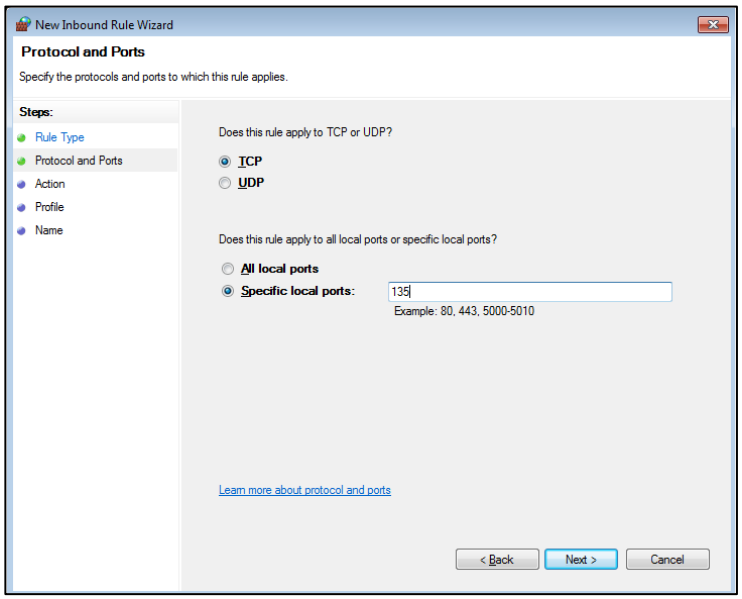


[Firewall(Exceptions])]
The Windows firewall is effectively set by default. It explains the method of connecting DCOM with DeviceXPlorer with the firewall made effective since it the next pages.

The application to pass the firewall can be specified in "Exceptions" tab. And push "Add Programs" button and add "DeviceXPlorer OPC Server 6". 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.



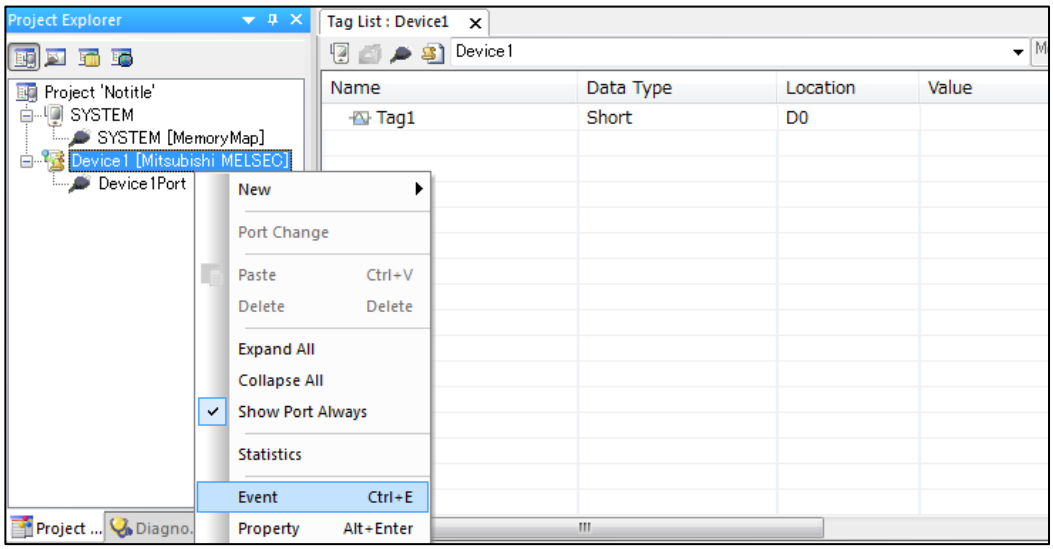
6 Script

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 5.8.

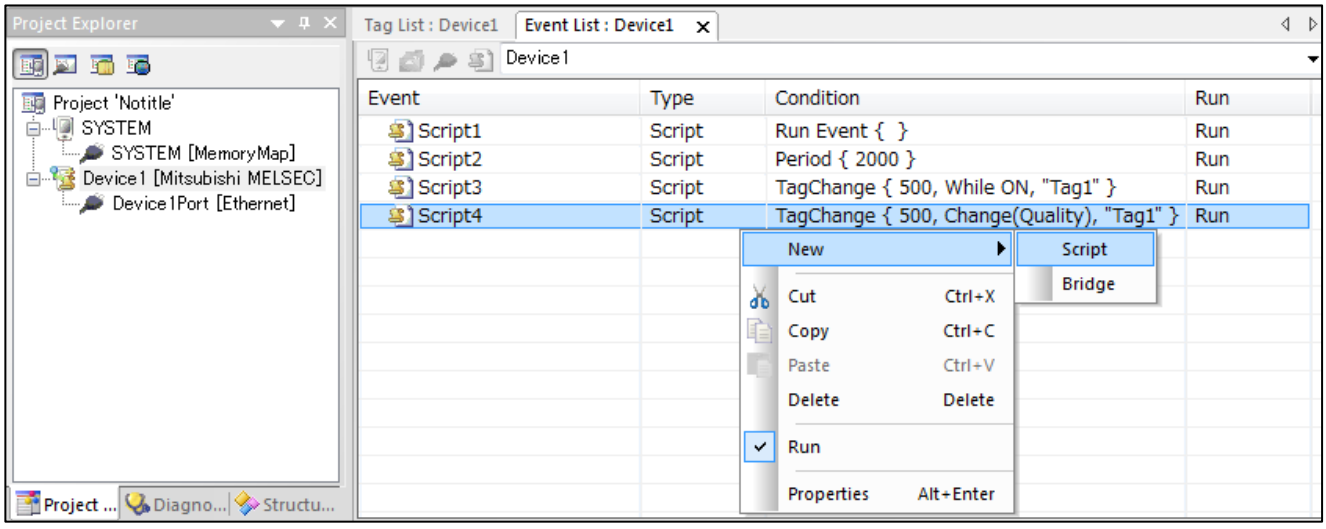
Important
- There is no upper limit to the number of script settings, but please note that the communication load will increase. - Please note that executable number of scripts (including bridges) is limited depending on the edition of DeviceXPlorer.

6.1 Event Script Settings

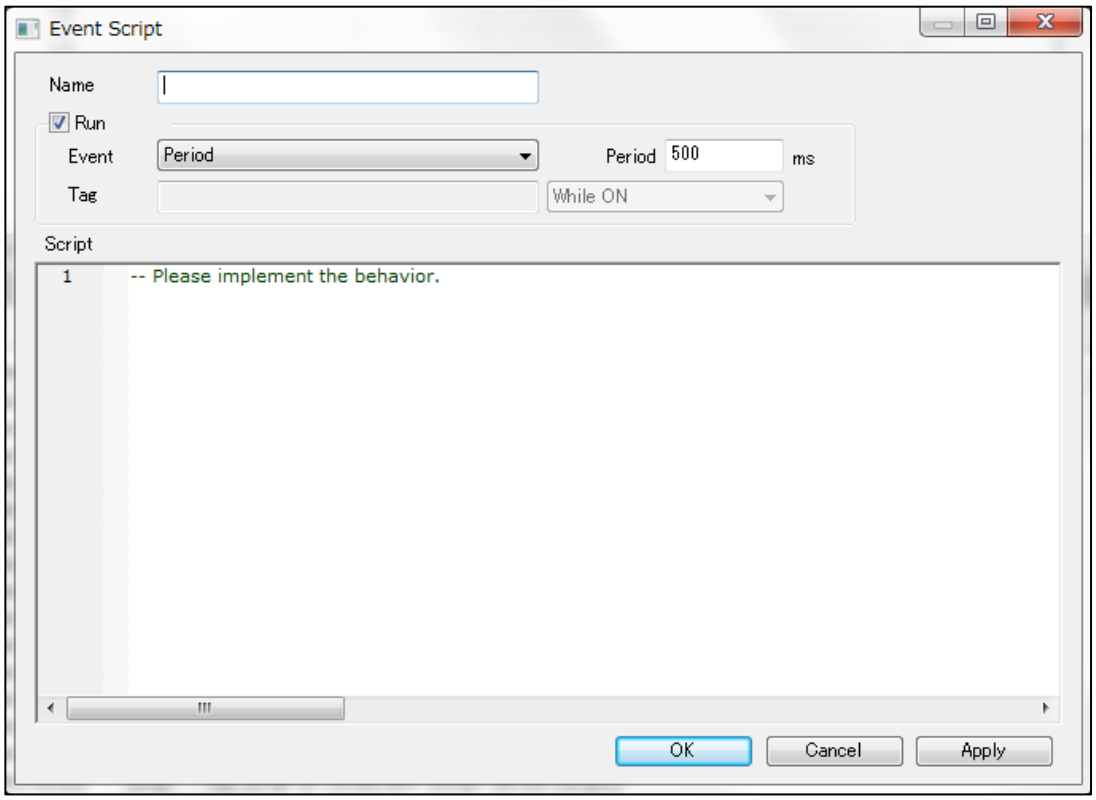
Device, Group and Struct Template have event list.
Event list will be displayed when select [Event] from right-click menu for Device and Group.



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. (*)
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.
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".

* 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.

6.2 Script Specification

Script in DeviceXPlorer uses open script language Lua .
Please refer to the official reference of Lua for the basic specification of Lua.

6.3 Extend Specification

DeviceXPlorer has some extend specifications for Lua.

6.3.1 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.

6.3.2 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

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

6.3.3 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
Time	Time Class

6.3.3.1 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 6.3.8about 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) "*ObjectTypeName*{*ClassName*, *Name*, *Path*, *Location*, *Value*}"
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)

6.3.3.2 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) "*ObjectTypeName*{ *ClassName*, *Name*, *Path*}"
ResolvePath(*name*)	[Argument value] name : Subordinate's relative path (String) [Return value] Fullpath

FindChild(*name*)	[Argument value] name : Subordinate's relative path (String)
	[Return value] Object Access Class(BooleanTag , Device etc), Function
	<Example>
	local group = @@g("Group1");
	local tag = group:FindChild("tag1");

GetParent()	[Return value] Group (Group, Device, Struct)
GetEvent(*name*)	[Argument value] name : event script name (Stirng) [Return value] Event (EventScript)
GetMethod(*name*)	[Argument value] name : method tag name (Stirng) [Return value] Method (Method)

6.3.3.3 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) "*ObjectTypeName*{ *ClassName*, *Name*, *Status*, *Standby*, *Simulate*}"
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.

6.3.3.4 Struct Access Class

Struct Access Class (Struct)

Member	Type	Attribute	Information
TypeName	String	R	Type name

Function	Information
ToString()	[Return value] instance information (String) "*ObjectTypeName*{ *ClassName*, *Name*, *StructTypeName*}"

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

6.3.3.5 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) "*ObjectTypeName*{ *ClassName*, *Name*}"

6.3.3.6 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) "*ObjectTypeName* { *ClassName*, *Name* }"

Run()	Run Event Script manually.
	[Return value] None
	
	<Sample1>
	local device = @d("Device1");
	local event = device:GetEvent("script01");
	event:Run();
	<Sample2>
	local event = @e("Device1.script01");
	event:Run();

6.3.3.7 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) "*ObjectTypeName* { *ClassName*, *Name* }"

Run()	Run Method manually.
	[Return value] None
	
	<Sample1>
	local device = @d("Device1");
	local method = device:GetMethod("Method01");
	method:Run();
	<Sample2>
	local method = @m("Device1.method01");
	method:Run();

6.3.3.8 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> local time = Dxp.GetCurrentTime(); Dxp.LogInfoMessage(time:Format("%Y/%m/%d %H:%M:%S")); Specify format of COleDateTime::Format().
Diff(*Time*)	Get difference of the time of Second [Argument] Time : Time class (Time) [Return value] difference second time (Number)
Year()	[Return value] Year (Number)
Month()	[Return value] Month. 1 to 12 (Number)
Day()	[Return value] Day. 1 to 31 (Number)
Hour()	[Return value] Hour. 0 to 23 (Number)
Min()	[Return value] Minute. 0 to 59 (Number)
Sec()	[Return value] Second. 0 to 59 (Number)
WDay()	[Return value] Week day 1(Sunday) to 7 (Saturday) (Number)

6.3.4 Functions

Under the global table Dxp, public and provide the following function.

Function		Content
Dxp		
	IsTag(*path, name*) IsDevice(*name*) IsGroup(*path, name*) IsStruct(*path, name*) IsPort(*name*) IsMethod(*path, name*) IsEvent(*path, name*)	This is to confirm whether the object exists for each access class. [Argument] path : Path (String) name : Name (String) [Return value] Whether there is a specified type of object(Boolean)
	GetObject(*path, name*)	This will get the object access class. [Argument] path : Path (String) name : Name (String) [Return value] Object access class (BooleanTag, Device etc)
	GetObjectType(*path, name*)	This will get the type of object access class. [Argument] path : Path (String)

		name : Name (String) [Return value] Object type (Number : Dxp.ObjectType.***)
	GetObjectTypeString(*type*)	This will get the name of the class type of object access. [Argument] type : Object Type (Number : Dxp.ObjectType.***) [Return value] Object type name (String)
	GetDelimiter()	This gets the delimiter character. [Return value] Name space delimiter (String)
	MakePath(*fullpath, child*)	Which takes the path with a delimiter character. [Argument] fullpath : Absolute path of the parent (String) child : Child name (String) [Return name] Absolute path of the parent + Name space delimiter + Child name
	GetQualityString(*quality*)	This will get a string of quality flags. [Argument] quality : Quality flag of the tag (Number) [Return value] Quality flag string (String)
	IsGoodQuality(*quality*)	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(*quality*)	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(*quality*)	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(*dst, src*)	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(*tags*)	This is done in a synchronous read.
		[Argument] tags : Table that contains the tag(Tag)
		[Return value] Result Success / Failure (Boolean)
		
		<Example>
		`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`

	SyncWrite(*tags*)	This is done in a synchronous write. [Argument] tags : Table that contains the tag (Tag) [Return value] Result Success / Failure (Boolean)

		<Example>
		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);

	LogMessage(*msg*)	This script outputs the log message. [Argument] msg : Output message (String)
	LogInfoMessage(*msg*)	This will output the info message. [Argument] msg : Output message (String)
	LogWarningMessage(*msg*)	This will output a warning message. [Argument] msg : Outoput message (String)
	LogErrorMessage(*msg*)	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.

6.3.5 Access Macros

DeviceXPlorer provides following macros originally. Access Macros creates the instance of tag, device, group, struct, method and port.

Macro	Content

@(*fullpath*)	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(*fullpath*)	Absolute access macro for tag. This macro creates the instance of the Tag Access Class.
@d(*fullpath*)	Absolute access macro for device. This macro creates the instance of the Device Access Class.
@g(*fullpath*)	Absolute access macro for group. This macro creates the instance of the Group Access Class.
@s(*fullpath*)	Absolute access macro for struct. This macro creates the instance of the Struct Access

	Class.

@m(*fullpath*)	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(*fullpath*)	Absolute access macro for event script. This macro creates the instance of the Event Script Access Class.
@p(*fullpath*)	Absolute access macro for port. This macro creates the instance of the event Port Access Class.
@@ @@(*relativePath*)	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"):ToStringValue()); <Example2> local device = @@; local tag = @@("Tag1"); Dxp.LogInfoMessage(device.Name .. "/" .. tag:ToStringValue());
@@t(*relativePath*)	Relative access macro for tag. This macro creates the instance of the Tag Access Class.
@@g(*relativePath*)	Relative access macro for group. This macro creates the instance of the Group Access Class.
@@s(*relativePath*)	Relative access macro for struct. This macro creates the instance of the Struct Access Class.
@@m(*relativePath*)	Relative access macro for method. This macro creates the instance of the Method Access Class.
@@e(*relativePath*)	Relative access macro for event script. This macro creates the instance of the Event Script Access Class.

6.3.6 Priority of Finding Object

Indicates a preference for objects that can be retrieved from the Object access macro(≒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

6.3.7 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".

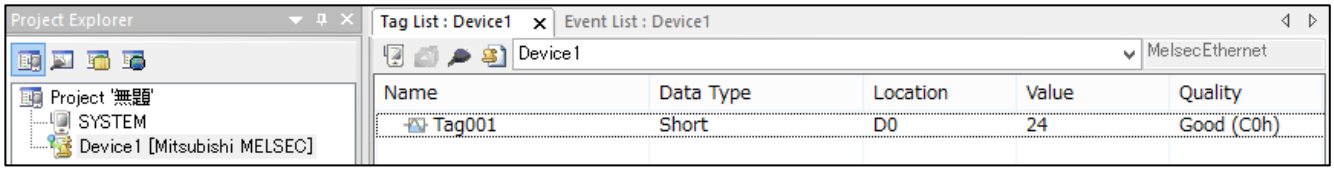
6.3.8 About Script Writing

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.

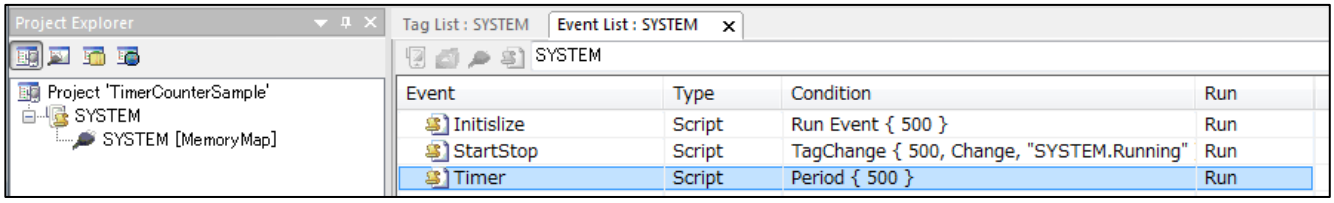
6.4 Samples

6.4.1 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 6.3.7 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;
```

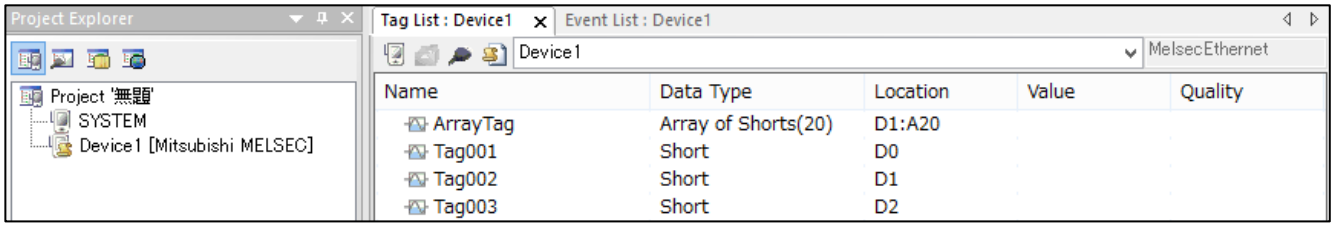
Cycle script(Period) implements the processing that continues always running. You can use global variables and functions which is defined in the Initialize script in this script. You can use "local" keyword to declare the local variable. If you want to access the tags, you can use the Tag Access Macro (@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)

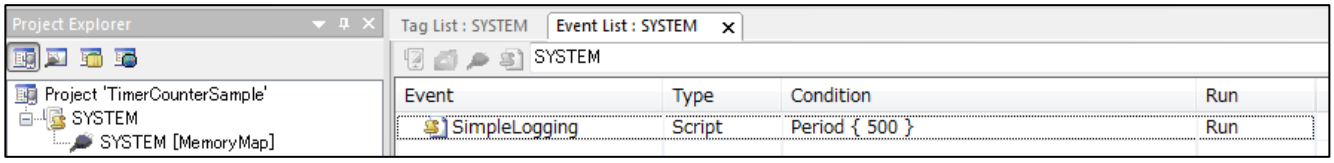
```
1   -- Count up global variable
2   g_counter = g_counter + 1;
3
4   -- Getting tag access object with local variable
5   local tag = @@t( "Tag001" );
6
7   -- Calling the function and writing the value of tag
8   tag.Value = CalcAdd( g_counter, 1000 );
9
10  -- Output message log (WarningLog)
11  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
```

6.4.2 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.



Make event script in the Device1.



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
```

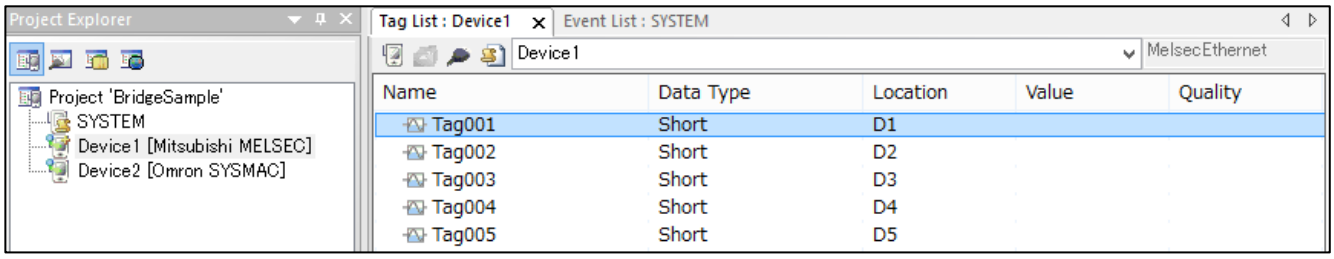
If you change the code to use the tag of array type, you can realize easily the logging about big continuous data.

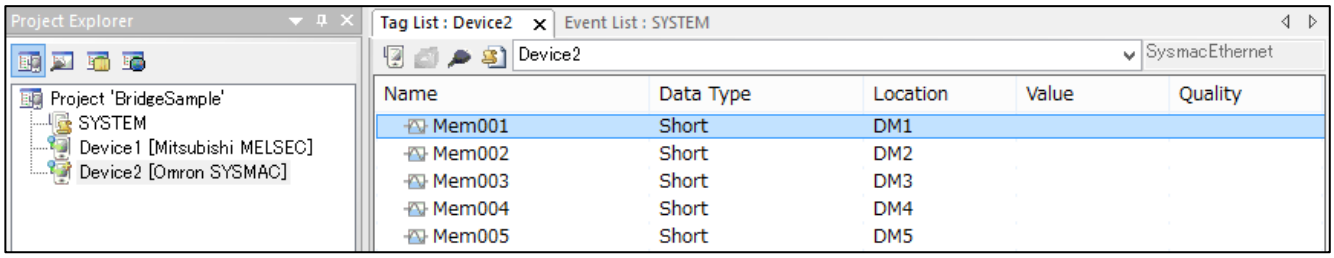
Script Example (SimpleLogging)

```
:
9   -- Make CSV
10  local text = "";
11  text = text .. Dxp.GetCurrentTime():Format();
12  text = text .. "," .. @@t("ArrayTag"):ToStringValue();
13
14  -- Write to file
:
```

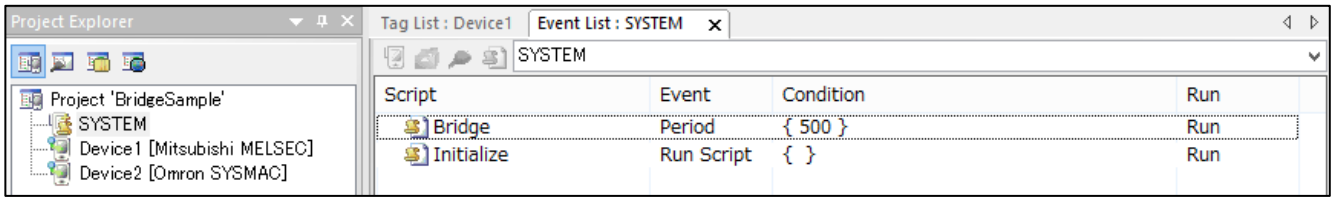
6.4.3 Example which defines the class in the external file (Bridge)

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).





Make the event script in the SYSTEM.



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
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 6¥Bin¥) if it is first time to making the external script file.

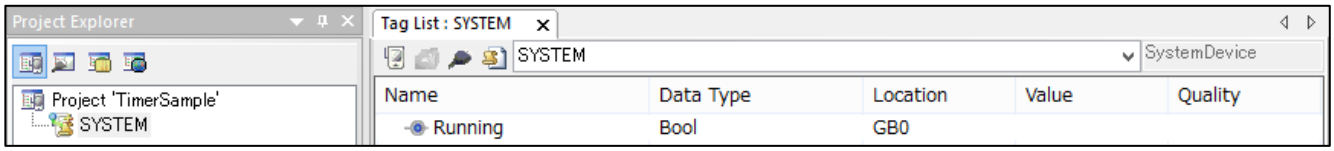
Script Example (<Bin Folder>¥Lua¥BridgeSample.lua)

```
if (LUA_TKB_BRIDGE_SAMPLE_INCLUDE == nil) then
    LUA_TKB_BRIDGE_SAMPLE_INCLUDE = 1;

    -- ********************************************************************
    -- BridgeData class
    -- ********************************************************************
    BridgeData =
    {
        New = function(_DstTagPath, _SrcTagPath)
            return
            {
                SrcTagPath = _SrcTagPath,
                DstTagPath = _DstTagPath,
                SrcTag = nil,
                DstTag = nil,

                -- ////////////////////////////////////////////////
                -- PrepareTag method
                -- ////////////////////////////////////////////////
                PrepareTag = function(this)
                    this.SrcTag = Dxp.GetTag( "", this.SrcTagPath );
                    this.DstTag = Dxp.GetTag( "", this.DstTagPath );

                    local found = true;

                    if ( this.SrcTag == nil) then
                        Dxp.LogErrorMessage( "BridgeData - Not found SrcTag [" ..
this.SrcTagPath .. "]" );
                        found = false;
                    end
                    if (this.DstTag == nil ) then
                        Dxp.LogErrorMessage( "BridgeData - Not found DstTag [" ..
this.DstTagPath .. "]" );
                        found = false;
                    end

                    if ( found == false ) then
                        this.SrcTag = nil;
                        this.DstTag = nil;
                    end

                    return found;
                end,

                -- ////////////////////////////////////////////////
                -- HasValue method
                -- ////////////////////////////////////////////////
                HasValues = function(this)
                    if ( this.SrcTag:HasValue() == false ) then
                        return false;
                    end
                    if ( this.DstTag:HasValue() == false ) then
                        return false;
                    end
                    return true;
                end,

                -- ////////////////////////////////////////////////
                -- BridgeForwared method
```

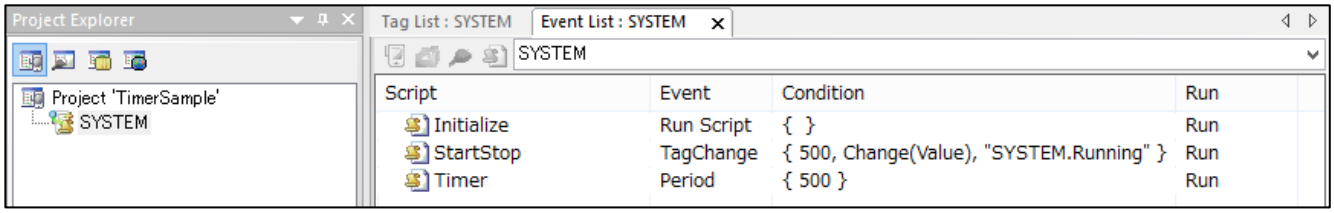
```
60                      -- ////////////////////////////////////////////////
61                      BridgeForward = function(this)
62                          this.DstTag.Value = this.SrcTag.Value;
63                      end,
64
65                      -- ////////////////////////////////////////////////
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
```

6.4.4 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).



Make the event script in the SYSTEM.



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 exript file in the “Lua” folder.
You should make “Lua” folder in the Bin Folder(C:¥Program Files (x86)¥TAKEBISHI¥DeviceXPlorer OPC Server 6¥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
```

7 Troubleshooting

7.1 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.

7.1.1 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.

7.1.2 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	

7.1.3 Serial Error Code

Code	Contents
2	Specified port does not exist.
5	Specified port already open by other process.

7.2 Others

7.2.1 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.

8 Appendix

8.1 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.

8.2 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.

8.3 User Support

TAKEBISHI offers a user support service to customers who have completed user registration. If you have any questions about how to set up or operate DeviceXPlorer, please contact the support service detailed below.

User Support
TEL 075-325-2261/FAX 075-325-2273
fa-support@takebishi.co.jp
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 CD, 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.

8.4 Copyright Information

The copyright information on the third party software currently used by DeviceXPlorer is shown.

8.4.1 Scintilla

License for Scintilla and SciTE
Copyright 1998-2003 by Neil Hodgson <neilh@scintilla.org>

All Rights Reserved
Permission to use, copy, modify, and distribute this software and its
documentation for any purpose and without fee is hereby granted,
provided that the above copyright notice appear in all copies and that
both that copyright notice and this permission notice appear in
supporting documentation.

NEIL HODGSON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS
SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS, IN NO EVENT SHALL NEIL HODGSON BE LIABLE FOR ANY
SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE
OR PERFORMANCE OF THIS SOFTWARE.

8.4.2 Lua

Copyright(C) 1994-2012 Lua.org, PUC-Rio.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

8.4.3 tolua++

Copyright (C) 2009 Ariel Manzur.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

8.4.4 CGridListCtrlEx

Copyright 2012 by Rolf Kristensen, under CPOL (The Code Project Open License).

8.4.5 OpenSSL

LICENSE ISSUES
==============

The OpenSSL toolkit stays under a double license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts.

OpenSSL License

```
/* ====================================================================
 * Copyright (c) 1998-2019 The OpenSSL Project.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 *    software must display the following acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 *    endorse or promote products derived from this software without
 *    prior written permission. For written permission, please contact
 *    openssl-core@openssl.org.
```

```
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 *    nor may "OpenSSL" appear in their names without prior written
 *    permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
 *    acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * ====================================================================
 *
 * This product includes cryptographic software written by Eric Young
 * (eay@cryptsoft.com).  This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).
 *
 */
```

Original SSLeay License

```
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
```

```
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
*      notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*      notice, this list of conditions and the following disclaimer in the
*      documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*      must display the following acknowledgement:
*      "This product includes cryptographic software written by
*       Eric Young (eay@cryptsoft.com)"
*      The word 'cryptographic' can be left out if the rouines from the library
*      being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*      the apps directory (application code) you must include an acknowledgement:
*      "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed.  i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/
```

8.4.6 Libxml2

Except where otherwise noted in the source code (e.g. the files hash.c,
list.c and the trio files, which are covered by a similar licence but
with different Copyright notices) all the files are:

Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is fur-
nished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FIT-
NESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

DeviceXPlorer OPCServer

User's Guide

Server Edition