Right choice for ultimate yield

LSIS strives to maximize customers' profit in gratitude of choosing us for your partner.

Human Machine Interface

XGT InfoU

XGT HMI Software

User's Manual





Safety Instructions

- Read this manual carefully before installing, wiring, operating, servicing or inspecting this equipment.
- Keep this manual within easy reach for quick reference.



Before using the product

Please read this user' manual thoroughly before using the product For a safe and effective use of this product,

- ► Because the purpose of cautions for safety is to prevent any possible accident or danger related to the project in advance by using it safely and correctly, please, comply with those cautions.
- ► Cautions can be divided into 'Warning' and 'Caution' and each has the following meaning:
- **!** Warning If any instruction is violated, serious injure or causality is likely to take place.
- ! Caution If any instruction is violated, minor injure or damage to the product is likely to take place.
- Symbols marked on the product and in the manual have the following meanings:
 - /!\ Symbol to notice that there is some possibility of the occurrence of danger.
 - // Symbol to notice that there is some possibility of electric shock.
- ► After reading the manual, please keep it at the place the product users can always see it.

Cautions when keeping the product

Warning

Please be cautious to keep and retain the enclosed license key.

Any lost license key will not be reissued and program actions may have functions restricted according to the license key.

Cautions when operating the product

Caution

> Please operate this software only in the allowed OS and installation environments.

The entire or part of the program may generate some errors or malfunctions if it is installed and used in any unauthorized OS.

Checklist when any error occurs during the program installation or the initial

execution

Please be aware that some program errors may occur because the vaccine program in the computer may block the program's communication port or service.

Check the history data volume

Accumulation of history data such as alarm history and data history may cause insufficient storage space in the system that leads to the occurrence of error.

Take into account of the system performance during the engineering works

If the user registers and uses too many tags or logs in excessive history data over the system performance, the program speed may become dull or generate malfunctions during the engineering or runtime operation.

Be cautious about logging data loss when adding a tag after deleting it

There is no problem with the logging settings even though a tag set up in the logging group is deleted from the tag database and then it is added again with the same name, however, it is likely to lose some logging data

It is better to use tag edition or CSV file update functions rather than adding a tag after deleting it when changing its properties.

Revision History

Version	Date	Major Change	Revised Page
V 1.0	'07.07	The first edition is published	-
V 1.1	'08.06	Revised to version 1.1	Full revision
V 1.5	'09.03	Revised to version 1.5	Full revision
V 1.5(English)	'09.06	English version 1.5 is published	-

^{*} The manual number is marked on the right side of the manual back cover

^{© 2007} LS Industrial Systems Co., Ltd All rights reserved.

About this manual

Thank you for purchasing LS Industrial System's HMI Software XGT-InfoU.

Before using this program, please thoroughly read and fully understand this manual on the functions, performances and program methods of the purchased product for a better and effect use and distribute it to the final users and persons responsible for maintenance of this product.

The following manual is the one related to this product.

Please, take a look at the following contents of the manual and make an order if necessary.

You may also download it in PDF file at our homepage http://www.lsis.biz/.

The manual list

Manual Title	Contents	Manual Number
Manual on InfoU HMI Package	This software manual explains functions needed for using InfoU products such as Project, Tag Engineering, Screen Edition, Communication and environment Settings, History Data and Report Settings.	10310001053

С	hapter	: 1 XGT-InfoU	
1.1	Specia	al Feature	
1.1	1.1	Open structure	1-1
1.1	1.2	Efficient Project Management	
1.1	1.3	Tag Engineering	1-2
	1.4	I/O Driver Convenience	1-2
	1.5	Better Alarm Function	
	1.6	Convenient Logging Management	
• • • • • • • • • • • • • • • • • • • •	1.7	Define VBS and Action Trigger	
	1.8	Robust Graphic Engine	
	1.9	Crooss Reference	
1.	1.10	Web Server	1-5
1.2	Syste	m requirement	1-6
1.3	Licens	Se	1-6
1.4	Produ	ct Compostion	1-7
1.5	Cautio	on	1-8
Ch	apter.	2 System Compostion	2-1 ~ 2-6
2.1	Softwa	are Component	2-1
2.2		Alone System	
2.3		/Server System	
2.4		Redundancy	
2.5		ndancy System)	
2.6		Server	
0			
Ch	apter.	3 Install and Start	3-1 ~ 3-12
3.1	Install		3-1
3.1	1.1	Arrange Installation	3-1
3.1	1.2	License Contract	
3.1	1.3	Select a Package to Install	3-3
3.1	1.4	Start Installation	3-4
3.1	1.5	Complete Installation	3-5
3.1	1.6	Menu and Shortcut	
	1.7	Reinstall InfoU	
3.1	1.8	Update Program Patch	3-8
3.2	Start a	and Stop	3-11
3.2	2.1	Start InfoU	3-11

3.2.2 Stop InfoU	3-12
Chapter. 4 Engineering	4-1 ~ 4-8
4.1 Screen Composition	4-1
4.2 Menu and InfoU Explorer	4-2
4.2.1 File	4-2
4.2.2 Tool	4-4
4.2.3 Help	
4.2.4 Tool Bar	
4.2.5 Pop-up Window	4-8
Chapter. 5 Run Time	
5.1 Start InfoU Run-Time	5-1
5.2 Shutdown InfoU Run-Time	5-3
5.3 Device Status	5-5
5.4 Simulation Settings	5-6
5.5 Device Data	5-7
Chapter. 6 System Settings	
6.1 Project	6-1
6.1.1 Create New Project	
6.1.2 Project Property	6-3
6.2 Graphic Run-Time Settings	6-6
6.3 Alarm Property	6-8
6.4 Time Synchronization	6-14
6.4.1 Default Screen	6-14
6.4.2 Scenario for the whole system	6-16
Chapter. 7 I/O Driver	7-1 ~ 7-135
7.1 Channel Device Settings	7-1
7.1.1 Channel	7-1
7.1.2 Station	
7.1.3 Delete Channel	7-4
7.2 Driver List	
	7-5
7.3 MASTER-K S Series Computer Link	

7.5	OPC Client	7-13
7.6	MelsecQEnet	7-25
7.7	MdelsecQSerial	7-31
7.8	GLOFA Loader	7-37
7.9	MKEnet	7-40
7.10	GlofaEnet20.	
7.11	GlofaCnet20	
7.12	XGIEnet	
	XGICnet	
	XGKEnet20	
7.15	XGKCnet20	7-75
7.16	XGKLoader	7-80
7.17	XGILoader	7-86
7.18	ModbusTCP	7-92
7.19	ModbusRTU	7-97
7.20	OmronE	7-104
7.21	OmronS	7-110
7.22	NaisS	7-116
7 23	MelserEXSerial	7-122
7.23	MelsecFXSerial	
_	MelsecFXSerial	
7.24		7-128
7.24 Cha	ABEthernetIPapter. 8 Tag Management	7-128
7.24 Cha 8.1	ABEthernetIP apter. 8 Tag Management Screen Composition	7-128 8-1 ~ 8-50
7.24 Cha 8.1	ABEthernetIP	7-128 8-1 ~ 8-50 8-1
7.24 Cha 8.1 8.1	ABEthernetIP	
7.24 Cha 8.1 8.1 8.1 8.2	ABEthernetIP	7-128 8-1 ~ 8-50 8-1 8-1 8-3 8-5
7.24 Cha 8.1 8.1 8.1	ABEthernetIP	8-1 ~ 8-50 8-1 ~ 8-50 8-1 8-1 8-1 8-3 8-5 8-5
7.24 Cha 8.1 8.1 8.1 8.2 8.2	ABEthernetIP	8-1 ~ 8-50 8-1 ~ 8-50 8-1 8-1 8-1 8-3 8-5 8-5 8-7
7.24 Cha 8.1 8.1 8.2 8.2 8.2	ABEthernetIP	8-1 ~ 8-50 8-1 ~ 8-50 8-1 8-1 8-1 8-3 8-5 8-5 8-5 8-7 8-8
7.24 Cha 8.1 8.1 8.2 8.2 8.2 8.2	ABEthernetIP	
7.24 Cha 8.1 8.1 8.2 8.2 8.2 8.3	ABEthernetIP	
7.24 Cha 8.1 8.1 8.2 8.2 8.2 8.3 8.3 8.3 8.3	ABEthemetIP	
7.24 8.1 8.1 8.2 8.2 8.2 8.3 8.3 8.3 8.3	ABEthernetIP	8-1 ~ 8-50 8-1 ~ 8-50 8-1 8-1 8-3 8-5 8-5 8-7 8-8 8-9 8-10 8-10 8-11 8-11
7.24 Cha 8.1 8.1 8.2 8.2 8.2 8.3 8.3 8.3 8.3 8.3	ABEthernetIP	8-1 ~ 8-50 8-1 ~ 8-50 8-1 8-1 8-3 8-5 8-5 8-7 8-8 8-9 8-10 8-10 8-11 8-11 8-13
7.24 8.1 8.1 8.2 8.2 8.2 8.3 8.3 8.3 8.3	ABEthernetIP apter. 8 Tag Management Screen Composition	8-1 ~ 8-50 8-1 ~ 8-50 8-1 8-1 8-3 8-5 8-5 8-7 8-8 8-9 8-10 8-10 8-11 8-11 8-13

8.3	3.9 Digital Alarm	8-15
8.4	Expression	8-16
8.4	4.1 Basic Concept	8-16
8.4	1.2 Operation Interval	8-16
8.4	1.3 Grammar and Insturction	8-17
8.5	Tag Edit	8-19
8.5	5.1 New Tag	8-19
8.5	5.2 Save	8-20
8.5	5.3 Delete	8-21
8.6	CSV Tag In/Outpu	8-23
8.6	S.1 Instroduction	8-23
8.6	6.2 Export to CSV File	8-23
8.6	5.3 Import from CSV File	8-27
8.7	Copy/Paste Tag	8-37
8.7	7.1 Introduction	8-37
8.7	7.2 Copy Tag	8-37
8.7	7.3 Paste Tag	8-38
8.8	Structure Tag	8-39
8.8	3.1 Introduction	8-39
8.8	3.2 Demo	8-40
8.8	3.3 Add Structure Tag Template	8-41
8.8	3.4 Add Structure Tag Item	8-43
8.8	3.5 Add Structure Tag	8-45
8.8	License Information	8-49
8.9	9.1 Introduction	8-49
8.9	9.2 Demo	8-49
Cha	apter. 9 Graphic Editor	9-1 ~ 9-83
9.1	Configuration of Graphic Editor	9-2
9.2	Menu Configuration	9-3
9.3	File	
9.4	Edit	
9.5	View	
9.6	Insert	9-15
9.7	Draw	9-18
9.8	Tool	
9.9	Window	
9.10	Help	9-22
9.11	Insert Image	9-23

9.14 Layer	9.12	Tag Li	nk	9-25
9.15 Object Property	9.13	Librar	y	9-27
9.15 Object Property	9.14	Laver		9-28
9.16 Define Dynamic Properties		•		
9.16.1 Horizontal Size 9.32 9.16.2 Vertical Size 9.33 9.16.3 Horizontal Move 9.33 9.16.4 Vertical Move 9.34 9.16.5 Horizontal Fill 9.35 9.16.6 Vertical Fill 9.36 9.16.6 Vertical Fill 9.36 9.16.8 Blink 9.37 9.16.9 Visible 9.38 9.16.10 Solate 9.38 9.16.11 Display Value 9.40 9.16.12 Display String 9.41 9.16.13 GiF Animation 9.42 9.16.14 GiF Play Time 9.43 9.16.16 GiF Tag Image 9.44 9.16.16 Gick 9.44 9.16.17 Input Value 9.40 9.16.18 Horizontal Drag 9.50 9.16.19 Tortical Drag 9.50 9.17 Runtime Settings 9.50 9.17 Runtime Settings 9.50 9.18 Graphic Script 9.52 9.17.2 Page Settings 9.53 9.18 Graphic Script 9.56 9.18.1 InfoU Graphic Function List 9.56 9.18.1 InfoU Graphic Function Isl 9.79 Chapter 10 Alarm Function 10.7 Chapter 11 Alarm Management 11.1		-		
9.16.2 Vertical Nove 9.33 9.16.3 Horizontal Move 9.33 9.16.4 Vertical Move 9.34 9.16.4 Vertical Move 9.34 9.16.5 Horizontal Fill 9.35 9.16.6 Vertical Fill 9.35 9.16.6 Vertical Fill 9.35 9.16.7 Color 9.36 9.16.8 Blink 9.37 9.16.9 Visible 9.38 9.16.11 Display Value 9.40 9.16.12 Display String 9.41 9.16.13 GiF Alimation 9.41 9.16.15 GiF Alimation 9.42 9.16.14 GiF Play Time 9.43 9.16.15 GiF Tag Image 9.44 9.16.15 Gick 9.44 9.16.16 Gick 9.44 9.16.17 Input Value 9.40 9.16.18 Horizontal Drag 9.50 9.16.19 Vertical Drag 9.50 9.16.19 Vertical Drag 9.50 9.17.1 Graphic Runtime Settings 9.52 9.17.2 Page Settings 9.53 9.18 Graphic Script 9.55 9.18.1 InfoU Graphic Function List 9.56 9.18.2 InfoU Explanation of InfoU Graphic Function 9.61 9.19 Tag Variable 9.79 Chapter 10 Alarm Function 10.7 Chapter 11 Alarm Management 11.1				
9.16.3 Horizontal Move 9.34 9.16.4 Vertical Move 9.34 9.16.5 Horizontal Fill 9.35 9.16.6 Vertical Fill 9.35 9.16.6 Vertical Fill 9.35 9.16.6 Vertical Fill 9.35 9.16.7 Color 9.36 9.16.8 Blink 9.37 9.16.9 Visible 9.38 9.16.10 Rotate 9.38 9.16.11 Display Value 9.40 9.16.12 Display String 9.41 9.16.13 GIF Animation 9.42 9.16.14 GIF Play Time 9.43 9.16.16 Glick 9.44 9.16.17 Input Value 9.44 9.16.18 Horizontal Drag 9.50 9.16.19 Vertical Drag 9.50 9.16.19 Vertical Drag 9.50 9.16.19 Vertical Drag 9.50 9.17.1 Graphic Runtime Settings 9.52 9.17.2 Page Settings 9.52 9.17.1 InfoU Graphic Function List 9.55 9.18.1 InfoU Graphic Function Graphic Function 9.61 9.19 Tag Variable 9.79 Chapter: 10 Alarm Function 10.7 Chapter: 11 Alarm Management 11.1 11.1 Alarm Action during the Runtime				
9.16.4 Vertical Move 9.34 9.16.5 Horizontal Fill 9.35 9.16.6 Vertical Fill 9.35 9.16.7 Color 9.36 9.16.8 Blink 9.37 9.16.9 Visible 9.38 9.16.11 Display Value 9.40 9.16.12 Display Value 9.40 9.16.12 Display String 9.41 9.16.13 GiF Alimation 9.42 9.16.14 GiF Play Time 9.43 9.16.15 GiF Tag Image 9.44 9.16.15 Gib Gib Cick 9.49 9.16.19 Visible 9.49 9.16.19 Graphic Burding 9.50 9.16.19 Vertical Drag 9.50 9.16.19 Vertical Drag 9.50 9.16.18 Horizontal Drag 9.50 9.17.1 Graphic Runtime Settings 9.51 9.17.1 Graphic Runtime Settings 9.52 9.17.2 Page Settings 9.53 9.18 Graphic Script 9.55 9.18.1 InfoU Graphic Function List 9.56 9.18.2 InfoU Explanation of InfoU Graphic Function 9.61 9.19 Tag Variable 9.79 Chapter 10 Alarm Function 10.7 Chapter 11 Alarm Management 11.1 Alarm Occurrence 11.1	_			
9.16.5 Horizontal Fill 9-35 9.16.6 Vertical Fill 9-35 9.16.7 Color 9-36 9.16.8 Blink 9-37 9.16.9 Visible 9-38 9.16.10 Display Value 9-40 9.16.11 Display String 9-41 9.16.12 Display String 9-41 9.16.13 GIF Play Time 9-43 9.16.15 GIF Tag Image 9-44 9.16.16 Click 9-44 9.16.17 Input Value 9-49 9.16.18 Horizontal Drag 9-50 9.16.19 Vertical Drag 9-50 9.16.19 Vertical Drag 9-51 9.17 Page Settings 9-52 9.17.2 Page Settings 9-53 9.18 InfoU Graphic Function List 9-55 9.18.1 InfoU Explanation of InfoU Graphic Function 9-61 9.19 Tag Variable 9-79 Chapter. 10 Alarm Function 10.1 Alarm Management 10.2 Chapter. 11 Alarm Management 11.1 11.1 Alarm Occurrence 11.1				
9.16.6 Vertical Fill				
9.16.7 Color				
9.16.9 Visible				
9.16.10 Rotate 9.39 9.16.11 Display Value 9.40 9.16.12 Display Value 9.40 9.16.12 Display Salue 9.41 9.16.13 GIF Animation 9.42 9.16.14 GIF Play Time 9.43 9.16.15 GIF Tag Image 9.44 9.16.16 Click 9.44 9.16.17 Input Value 9.49 9.16.18 Horizontal Drag 9.50 9.16.19 Vertical Drag 9.50 9.17.1 Graphic Runtime Settings 9.52 9.17.1 Graphic Runtime Settings 9.52 9.17.2 Page Settings 9.53 9.18 Graphic Script 9.55 9.18.1 InfoU Graphic Function List 9.56 9.18.2 InfoU Explanation of InfoU Graphic Function 9.61 9.19 Tag Variable 9.79 Chapter. 10 Alarm Function 10.5 Chapter. 11 Alarm Management 10.5 Chapter. 11 Alarm Management 11.1 11.1 Alarm Action during the Runtime 11.1 11.1 Alarm Action during the Runtime 11.1	9.16	6.8	Blink	9-37
9.16.11 Display Value	9.16	5.9	Visible	9-38
9.16.12 Display String	9.16	5.10	Rotate	9-39
9.16.13 GIF Animation 9.42 9.16.14 GIF Play Time 9.43 9.16.15 GIF Tag Image 9.44 9.16.15 Input Value 9.44 9.16.17 Input Value 9.49 9.16.18 Horizontal Drag 9.50 9.16.19 Vertical Drag 9.51 9.17 Runtime Settings 9.52 9.17.1 Graphic Runtime Settings 9.52 9.17.2 Page Settings 9.53 9.18 Graphic Script 9.55 9.18.1 InfoU Graphic Function List 9.56 9.18.2 InfoU Explanation of InfoU Graphic Function 9.61 9.19 Tag Variable 9.79 Chapter: 10 Alarm Function 10.5 10.3 Alarm Delay 10.7 Chapter: 11 Alarm Management 11.1 11.1 Alarm Action during the Runtime 11.1 11.1 Alarm Action during the Runtime 11.1	9.16	3.11	Display Value	9-40
9.16.14 GIF Play Time 9-43 9.16.15 GIF Tag Image 9-44 9.16.16 Click 9-44 9.16.17 Input Value 9-49 9.16.18 Horizontal Drag 9-50 9.16.19 Vertical Drag 9-51 9.17 Runtime Settings 9-52 9.17.1 Graphic Runtime Settings 9-52 9.17.2 Page Settings 9-53 9.18 Graphic Script 9-55 9.18.1 InfoU Graphic Function List 9-56 9.18.2 InfoU Explanation of InfoU Graphic Function 9-61 9.19 Tag Variable 9-79 Chapter. 10 Alarm Function 10-1 10-2 Digital Alarm 10-5 10-3 Alarm Delay 10-7 Chapter. 11 Alarm Management 11-1 11-1 Alarm Action during the Runtime 11-1 11-1 11-1 Alarm Occurrence 11-1 11	9.16	5.12	Display String	9-41
9.16.15 GIF Tag Image	9.16	5.13	GIF Animation	9-42
9.16.16 Click 9.44 9.16.17 Input Value 9.49 9.16.18 Horizontal Drag 9.50 9.16.19 Vertical Drag 9.51 9.17 Runtime Settings 9.52 9.17.1 Graphic Runtime Settings 9.52 9.17.2 Page Settings 9.53 9.18 Graphic Script 9.55 9.18.1 InfoU Graphic Function List 9.56 9.18.2 InfoU Explanation of InfoU Graphic Function 9.61 9.19 Tag Variable 9.79 Chapter: 10 Alarm Function 10.1 10.2 Digital Alarm 10.1 10.2 Digital Alarm 10.7 Chapter: 11 Alarm Management 10.7 Chapter: 11 Alarm Management 11.1 11.1 Alarm Action during the Runtime 11.1	9.16	5.14	GIF Play Time	9-43
9.16.17 Input Value 9-49 9.16.18 Horizontal Drag 9-50 9.16.19 Vertical Drag 9-51 9.17 Runtime Settings 9-52 9.17.1 Graphic Runtime Settings 9-52 9.17.2 Page Settings 9-53 9.18 Graphic Script 9-55 9.18.1 InfoU Graphic Function List 9-56 9.18.2 InfoU Explanation of InfoU Graphic Function 9-61 9.19 Tag Variable 9-79 Chapter. 10 Alarm Function 10-1 10.2 Digital Alarm 10-5 10.3 Alarm Delay 10-7 Chapter. 11 Alarm Management 11-1 11.1 Alarm Action during the Runtime 11-1 11.1 Alarm Occurrence 11-1	9.16	3.15	GIF Tag Image	9-44
9.16.18 Horizontal Drag 9-50 9.16.19 Vertical Drag 9-51 9.17 Runtime Settings 9-52 9.17.1 Graphic Runtime Settings 9-52 9.17.2 Page Settings 9-53 9.18 Graphic Script 9-55 9.18.1 InfoU Graphic Function List 9-56 9.18.2 InfoU Explanation of InfoU Graphic Function 9-61 9.19 Tag Variable 9-79 Chapter. 10 Alarm Function 10-1 10.2 Digital Alarm 10-5 10.3 Alarm Delay 10-7 Chapter. 11 Alarm Management 11-1 11.1 Alarm Action during the Runtime 11-1 11.1.1 Alarm Occurrence 11-1	9.16	5.16	Click	9-44
9.16.19 Vertical Drag 9-51 9.17 Runtime Settings 9-52 9.17.1 Graphic Runtime Settings 9-52 9.17.2 Page Settings 9-53 9.18 Graphic Script 9-55 9.18.1 InfoU Graphic Function List 9-56 9.18.2 InfoU Explanation of InfoU Graphic Function 9-61 9.19 Tag Variable 9-79 Chapter. 10 Alarm Function .10-1 10.2 Digital Alarm 10-5 10.3 Alarm Delay 10-7 Chapter. 11 Alarm Management .11-1 11.1 Alarm Occurrence .11-1	9.16	5.17	Input Value	9-49
9.17 Runtime Settings 9-52 9.17.1 Graphic Runtime Settings 9-52 9.17.2 Page Settings 9-53 9.18 Graphic Script 9-55 9.18.1 InfoU Graphic Function List 9-56 9.18.2 InfoU Explanation of InfoU Graphic Function 9-61 9.19 Tag Variable 9-79 Chapter. 10 Alarm Function 10-1 ~ 10-7 10.1 Analog Alarm 10-5 10.3 Alarm Delay 10-7 Chapter. 11 Alarm Management 11-1 ~ 11-9 11.1 Alarm Action during the Runtime 11-1 11.1.1 Alarm Occurrence 11-1	9.16	5.18	Horizontal Drag	9-50
9.17.1 Graphic Runtime Settings 9-52 9.17.2 Page Settings 9-53 9.18 Graphic Script 9-55 9.18.1 InfoU Graphic Function List 9-56 9.18.2 InfoU Explanation of InfoU Graphic Function 9-61 9.19 Tag Variable 9-79 Chapter. 10 Alarm Function 10-1 10.2 Digital Alarm 10-5 10.3 Alarm Delay 10-7 Chapter. 11 Alarm Management 11-1 11.1 Alarm Occurrence 11-1	9.16	5.19	Vertical Drag	9-51
9.17.2 Page Settings 9-53 9.18 Graphic Script 9-55 9.18.1 InfoU Graphic Function List 9-56 9.18.2 InfoU Explanation of InfoU Graphic Function 9-61 9.19 Tag Variable 9-79 Chapter. 10 Alarm Function 10-1 10.2 Digital Alarm 10-5 10.3 Alarm Delay 10-7 Chapter. 11 Alarm Management 11-1 ~ 11-9 11.1 Alarm Occurrence 11-1	9.17	Runtir	ne Settings	9-52
9.18 Graphic Script 9-55 9.18.1 InfoU Graphic Function List 9-56 9.18.2 InfoU Explanation of InfoU Graphic Function 9-61 9.19 Tag Variable 9-79 Chapter. 10 Alarm Function 10-1 ~ 10-7 10.1 Analog Alarm 10-1 10.2 Digital Alarm 10-5 10.3 Alarm Delay 10-7 Chapter. 11 Alarm Management 11-1 ~ 11-9 11.1 Alarm Action during the Runtime 11-1 11.1.1 Alarm Occurrence 11-1	9.17	7.1	Graphic Runtime Settings	9-52
9.18.1 InfoU Graphic Function List	9.17	7.2	Page Settings	9-53
9.18.2 InfoU Explanation of InfoU Graphic Function 9-61 9.19 Tag Variable 9-79 Chapter. 10 Alarm Function 10-1 ~ 10-7 10.1 Analog Alarm 10-1 10.2 Digital Alarm 10-5 10.3 Alarm Delay 10-7 Chapter. 11 Alarm Management 11-1 ~ 11-9 11.1 Alarm Action during the Runtime 11-1 11.1.1 Alarm Occurrence 11-1	9.18	Graph	iic Script	9-55
9.18.2 InfoU Explanation of InfoU Graphic Function 9-61 9.19 Tag Variable 9-79 Chapter. 10 Alarm Function 10-1 ~ 10-7 10.1 Analog Alarm 10-1 10.2 Digital Alarm 10-5 10.3 Alarm Delay 10-7 Chapter. 11 Alarm Management 11-1 ~ 11-9 11.1 Alarm Action during the Runtime 11-1 11.1.1 Alarm Occurrence 11-1	9.18	3.1	InfoU Graphic Function List	9-56
Chapter. 10 Alarm Function 10-1 ~ 10-7 10.1 Analog Alarm 10-1 10.2 Digital Alarm 10-5 10.3 Alarm Delay 10-7 Chapter. 11 Alarm Management 11-1 ~ 11-9 11.1 Alarm Action during the Runtime 11-1 11.1.1 Alarm Occurrence 11-1			•	
Chapter. 10 Alarm Function 10-1 ~ 10-7 10.1 Analog Alarm 10-1 10.2 Digital Alarm 10-5 10.3 Alarm Delay 10-7 Chapter. 11 Alarm Management 11-1 ~ 11-9 11.1 Alarm Action during the Runtime 11-1 11.1.1 Alarm Occurrence 11-1	9.19	Tag V	ariable	9-79
10.1 Analog Alarm 10-1 10.2 Digital Alarm 10-5 10.3 Alarm Delay 10-7 Chapter. 11 Alarm Management 11-1 ~ 11-9 11.1 Alarm Action during the Runtime 11-1 11.1.1 Alarm Occurrence 11-1				
10.2 Digital Alarm	Cha	pter.	10 Alarm Function	10-1 ~ 10-7
10.3 Alarm Delay	10.1	Analo	g Alarm	10-1
10.3 Alarm Delay	10.2	Digita	Alarm	10-5
Chapter. 11 Alarm Management 11-1 ~ 11-9 11.1 Alarm Action during the Runtime		-		
11.1 Alarm Action during the Runtime	10.3	Alam	Delay	10-7
11.1.1 Alarm Occurrence	Cha	pter.	11 Alarm Management	11-1 ~ 11-9
	11.1	Alarm	Action during the Runtime	11-1
	11 1	l 1	Alarm Occurrence	11_1

11.1.3	Cancel Alarm	11-2
11.2 Alam	m Group Settings	11-3
11.2.1	Demo	
11.2.1	Demo	
Chapter	: 12 Logging Management	12-1 ~ 12-9
12.1 Logg	ging Type	12-1
12.2 Logg	ging Settings	12-3
12.2.1	Add, Edit, Delete	12-4
12.2.2	General	12-5
12.2.3	Cycle	12-6
12.2.4	Statistics	12-7
12.2.5	Accumulation	12-7
12.2.6	String	12-8
12.2.7	Event	12-8
12.2.8	Selective Log	12-9
Chantar	. 42 Comint	
Chapter	: 13 Script	13-1~ 13-16
13.1 Start		13-1
13.2 Edito	or Screen	13-2
13.3 How	to Use	13-5
	ime Debug	
TO: T TRAIN	D 3334g	
Chapter	: 14 Function	14-1 ~ 14-66
14.1 Fund	ction Inquire	14-1
14.2 Fund	ction Classification	14-2
14.2.1	List by Item	14-2
14.2.2	List by alphabetical order	
14.3 Fund	ction Description and Example	14-10
14.3.1	Project Function	14-10
14.3.2	Screen Function	
14.3.3	Tag Function	
14.3.4	Script Function	
14.3.5	Event Function	
14.3.6	Log Function	
14.3.7	Network Function	
14.3.8		
17.0.0	System Function	14-30
14.3.9	System Function	
	·	14-33

Chapte 17.1 Con 17.1.1 17.1.2 17.1.3 17.2 Setti 17.2.1 17.2.2 17.2.3 17.3 Rec 17.3.1 17.3.2 17.3.3	r. 17 Recipe	
17.1 Con 17.1.1 17.1.2 17.1.3 17.2 Setti 17.2.1 17.2.2 17.2.3 17.3 Rec 17.3.1 17.3.2	r. 17 Recipe Inponent Model Unit Data Ings Model Information Unit information Data Information Data Information Function Recipe Viewer OCX	
17.1 Con 17.1.1 17.1.2 17.1.3 17.2 Setti 17.2.1 17.2.2 17.2.3 17.3 Rec 17.3.1	r. 17 Recipe nponent Model Unit Data ings Model Information Unit information Data Information Data Information Function	
Chapter 17.1 Con 17.1.1 17.1.2 17.1.3 17.2 Setti 17.2.1 17.2.2 17.2.3 17.3 Rec	r. 17 Recipe Inponent Model Unit Data ings Model Information Unit information Data Information Data Information Data Information	
Chapte 17.1 Con 17.1.1 17.1.2 17.1.3 17.2 Setti 17.2.1 17.2.2 17.2.3	r. 17 Recipe Inponent Model Unit Data ings Model Information Unit information Data Information	
Chapte 17.1 Con 17.1.1 17.1.2 17.1.3 17.2 Setti 17.2.1 17.2.2	r. 17 Recipe nponent Model Unit Data ings Model Information Unit information	
Chapte 17.1 Con 17.1.1 17.1.2 17.1.3 17.2 Setti 17.2.1	r. 17 Recipe	
Chapter 17.1 Con 17.1.1 17.1.2 17.1.3 17.2 Setti	r. 17 Recipe	
17.1 Con 17.1.1 17.1.2 17.1.3	r. 17 Recipe	
17.1 Con 17.1.1 17.1.2	r. 17 Recipe nponent Model Unit	17-1 ~ 17-8 17-1 17-1
Chapte 17.1 Con 17.1.1	r. 17 Recipenponent	17-1 ~ 17-8
Chapte	r. 17 Recipe	17-1 ~ 17-8
Chapte	r. 17 Recipe	17-1 ~ 17-8
16.1 Setti	Ing 00/00/1	10-1
40.4 0-44	ing Screen	16.1
Chapte	r. 16 User Settings	16-1 ~ 16-2
13.3 FION	uay Settings	10-14
	day Settings	
15.2.3 15.2.4	Tag Value Change	
15.2.2	Schedule	
15.2.1	General Information	15-3
15.2 Editi	ion Screen	15-3
15.1 Start	t	15-1
Chapte	r. 15 Action Trigger	
14.4 App	lication Example of OLE Automation	14-66
14.3.16	I/O Drive Function	
	Report Function History Data Function	
14.3.14 14.3.15		
14.3.13 14.3.14 14.3.15	Lloor Function	
14.3.14	External Device Function	14-53

18.1.1	Demo	18-1
18.1.2	Database Connection Settings	18-2
18.2 Data	abase Operation	18-4
18.2.1	Operation Type	18-4
18.2.2	Demo	18-4
18.2.3	Database Operation Settings	18-5
18.3 Data	abase Runtime	18-9
18.3.1	Function	18-10
18.3.2	Internal Function	18-10
Chapter	r. 19 Report	
19.1 Start	t	
19.2 Repo	ort Menu	19-2
19.3 New	<i>1</i>	19-3
19.3.1	Menu	19-3
19.3.2	Edit	19-5
19.3.3	Copy/Move	19-8
19.4 Layo	out	19-9
19.4.1	Report Layout	19-9
19.4.2	Sheet Property	19-11
19.5 Repo	ort Open	19-16
19.5.1	Edit	19-16
19.5.2	Report View	
19.5.3	Refresh Auto Output Information	
19.5.4 10.5.5	View History Information How to Use during the Runtime	
19.5.5	now to use during the Ruhume	19-20
Chapter	r. 20 Utility	20-1 ~ 20-18
20.1 Unit.		20-1
20.2 Alarr	m Message Registration	20-3
20.3 On/C	Off Display Message Registration	20-5
20.4 LSIS	S Program Registration	20-7
20.5 LSP	PLC Tag Import	20-11
20.6 Appl	lication Registration	20-16
Chapter	r. 21 Cross Reference	21-1 ~ 21-7
21.1 Fund	ction and Property	21-1
21.2 Engi	ineering Function	21-2
14 L S	5 Industrial Systems	

21.3 Runtii	me Function	21-7
Chapter.	22 Graphic Runtime	22-1 ~ 22-13
22.1 Run		22-1
22.2 Menu	ı Composition	22-2
22.3 File		22-3
22.4 View		22-5
	ation	
·		
22.7 Cross	s Reference Function	22-10
Chapter.	23 Graphic Object	23-1 ~ 23-83
23.1 How t	to Use Graphic Object	23-1
23.1.1	Object Insert	23-1
23.1.2	Script Use	23-3
23.2 Alarm	ı Viewer	23-4
23.2.1	Prerequisite and Environment	23-4
23.2.2	Screen Configuration	
23.2.3	Real-time Function	23-6
23.2.4	Settings	
23.2.5	Filtering	
23.2.6	Other Functions	
23.2.7	How to Use Script	23-17
23.3 Trend	d Viewer	23-23
23.3.1	Prerequisite and Environment	23-23
23.3.2	Screen Configuration	
23.3.3	Settings	
23.3.4	Logging Search	
23.3.5	Others	
23.3.6	How to Use Script	
23.4 List T	rend Viewer	23-50
23.4.1	Prerequisite and Environment	23-50
23.4.2	Screen Configuration	
23.4.3	Settings	
23.4.4	How to Use Script	23-61
23.5 Recip	pe Viewer	23-67
23.5.1	Prerequisite and Environment	23-67
23.5.2	Screen Configuration	23-67
23.5.3	Real-Time Function	23-69
23 5 4	How to Use Script	23-73

23.6 Gaug	e Control	23-76
23.6.1	Prerequisite and Environment	23-76
23.6.2	Screen Configuration	23-76
23.6.3	Property Settings	23-78
23.7 Pictu	reLamp Control	23-80
23.7.1	Prerequisite and Environment	23-80
23.7.2	Screen Configuration	23-80
23.7.3	Function Description	23-81
23.7 How	to Use External ActiveX	23-82
23.7.1	How to Use	23-82
Chapter	24 Web	24-1 ~ 24-26
24.1 InfoU	Web Navigator Automatic Setup	24-1
24.1.1	Prerequisite	24-1
24.1.2	InfoU Web Navigator Run	24-2
24.2 InfoU	Web Navigator	24-13
24.2.1	Login Screen	24-13
24.2.2	Prerequisite	24-13
24.2.3	Description of Login Screen Menu	24-14
24.2.4	Downloading InfoU ActiveX	24-15
24.2.5	Main Screen	24-18
24.2.6	Menu	24-18
24.3 Web	HMI Server	24-24
24.3.1	Introduction	24-24
24.3.2	Operation Screen	24-24
24.3.3	Server Settings	24-25
Chapter	25 Project DB Migration	25-1 ~ 25-6
	Data Migrator	25-2
25.2 Data	Migration Information	25-3
Chapter	26 Redundancy and Client Server	26-1 ~ 26-21
26.1 Syste	m Environment Settings	26-1
26.1.1	Cancel Network Security	26-1
26.2 Redu	ndancy Settings	26-6
26.2.1	Procedure	26-6
26.2.2	Setting Screen	26-6
26.2.3	Realtime Run	26-12

26.2 Clie	ent-Server Settings	26-14
26.3.1	Procedure	
26.3.2	Server Project Settings	
26.3.3	Client Project Settings	26-17
Appen	dix 1. Visual Basic Script Help	A1-1~A1-58
_		
Appen	dix 2. LSIS InfoU OPC Server Help	A2-1~A2-33
A2.1 Introduction		A2-1
A 2.2 Install Program		A2-1
2.2.1	Install	A2-1
2.2.2	Start Installation	A2-3
2.2.3	Complete Installation	A2-4
2.2.4	Modify / Repair / Remove Program	A2-5
A 2.3 Co	nfiguration of Default Screen	A2-10
2.3.1	Configuration	A2-10
2.3.2	Monitoring	A2-13
A 2.4 OPC Project Configuration		A2-15
2.4.1	New OPC Project	A2-15
2.4.2	Open OPC Project / Modify	
A 2.5 Run Monitoring		A2-18
2.5.1	Connect LS InfoU OPC Server / Start Monitoring	A2-18
2.5.2	Disconnect LS InfoU OPC Server / Stop Monitoring	
A 2.6 DC	COM Configuration	A2-20
2.6.1	Firewall Configuration	A2-20
2.6.2	DCOM Configuration	

Chapter 1 XGT-InfoU

As a PC HMI(Human Machine Interface) software package for collecting real-time data on the device, setting up a data transmission system and playing a role of a central observer, InfoU allows the operator to monitor/analyze the overall operational status of the device in real time.

This software provides various useful functions that can run in the latest operation systems such as Windows XP, Windows 2000 and Windows Vista and enhances the reliability, fast processing speed and convenient use of the automatic system.

Data management function of this monitoring software helps the user to collect, store and calculate data on the device's operation status and to respond to requests for data. Its monitoring function also provides real-time data along with graphs configured by the operator on the display screen.

1.1 **Special Feature**

1.1.1 **Open Structure**

Its open structure with excellent extension and compatibility makes it possible to provide stable communication and effective interface with other systems.

- (1) Supports industrial standard interface such as 10LE DB, OPC Server/Client
- (2) Easy access to DBMS (Data Base Management System) Its database connection or work function ensures easy input or output by using tag values of universal databases such as MS, SQL or Oracle and efficient data sharing with MES or ERP system.
- (3) It can be referred by other applications with OLE automation.
- (4) It supports edition and compatibility of MS office software especially PowerPoint of the graphic editor for easy graphic works
- (5) It also provides ActiveX component graphic objects and allows the user to use other company's components for his/her graphic works.

1.1.2 **Efficient Project Management**

InfoU project is treated as the unit subjected to engineering and real-time operation management and under each project unit, sub-paths such as tag, logging, alarm, graph, report and script are created and managed.

Therefore, it becomes much easier to move or copy a certain project to other system, drive or path.

The user can easily manage numerous projects while easily copying and transferring contents between projects and efficiently conducting engineering works.

In addition, the user can selectively execute only the modules he/she wants since each function works individually in its own execution unit,

1.1.3 Tag Engineering

It provides efficient engineering and convenient operation by providing two perspectives of tag management criteria.

(1) View by device

It is very convenient to use since device set-up and tag work can be executed at the same time as a treetype tag explorer according to device configuration.

(2) View by group

Logical tag groups are displayed in a window explorer type to allow the user to manage tags according to their purposes from the point of operator's view.

(3) Import/Export CSV

The user can output, add, edit or delete a large amount of tags by using CSV (Comma Separated Values)type files editable with Excel or general text editors (notepad).

(4) Expression

Not only numeric expression but also logical expression can be used with memory tag values

(5) LS PLC Tag Sharing

GLOFA-LS Industrial System's PLC-shares its program tool GMWIN and engineering files to enable automatic input to InfoU tags.

1.1.4 I/O Driver Convenience

(1) Tag Simulator

Even though there is no device in place actually or it is before communication service is open, it can provide various rules of data to tags and significantly reduce debugging time since it can run the same applications with those in the site.

Analog Simulator: Random, sign wave, chopping wave, saw tooth wave, control function Digital Simulator: Random, change, control function

(2) Create auto block

Even though the user does not set up additional communication blocks, it automatically creates blocks according to tag's I/O address setting when runtime runs.

(3) Automatically register system tag when adding a station

When registering a station, it also registers its communication status automatically to allow the user to check the station status and its communication performance.

1.1.5 **Better Alarm Function**

Its improved alarm function using alarm group delivers more efficient problem solving.

(1) Manage alarm group

It can manage alarm groups of the tag designated by the user to execute alarm sound, printing, sending mails and SMS by the level of the alarm generated in the relevant tag.

(2) Batch assignment of alarm rank

If any alarm rank is not assigned in the tag setting, the user can assign a certain rank uniformly to all of the alarms in a group by using this batch assignment function.

(3) Various analog alarms

Since Limit alarm, Gap alarm and Variation alarm occur for one tag at the same time, each process can be closely and thoroughly monitored.

(4) Change dynamic setting

Dynamic change in alarm's limit value, gap value, variation value and other setting values enables more flexible action processing.

1.1.6 Convenient Logging Management

The user can create and manage a data logging group for major data that need history management according to his/her purpose and assign one tag data to multiple logging groups.

(1) Type of logging group

Divided into Cycle, Statistics, Accumulation and String

(2) Backup

It is possible to backup all of the collected groups and backup files can be inquired from trend and list trend.

(3) Save in excel file

Logging data inquired from trend and list trend can be immediately saved in excel files.

(4) Inquire logging data

Data can be inquired from trend, report and list trend and API, which can be inquired from script, is provided

(5) Automatically calculated statistics/integrated data

Statistics/integrated data are created without the use of additional script or application.

(6) Data to inquire according to type of history model

Cycle: Analog (current value), digital (current status)

Statistics: Analog (current value, min value, min time, max value, max time, average value) Digital (ON count, running time, change count)

Accumulation: Analog (change amount, current value)

Chapter 1. XGT-InfoU

(7) Report

Daily report, monthly report, temporary report and etc can be generated. The user can easily select an Excel form and register data in the form. Data of logging group can be inquired and automatically printed from Report and this Report function provides copy of report model, drag & drop report tag, clipboard copy/paste,

1.1.7 Define VBS and Action Trigger

It supports Action Trigger and VB Script for preparing robust and flexible applications

(1) Script Editor

It supports VB script, uses external OLE individuals such as ActiveX and registers and uses DLL's API.

(2) Transmit tag name with parameter value.

It is possible to implement effective applications by assigning various action triggers for script execution and transmitted tag name to script function with parameter value.

(3) Action Trigger

It integrates execution action conditions of script, recipe and database work together to manage them and trigger actions.

(4) Changing tag value, generating alarm and action by schedule can be set up.

Also efficient management is possible by assigning one action to multiple tasks.

(5) Provide recipe

Recipe is used when the same production/process line generates various products according to the prescribed vales and this function is applicable to various fields including flexible manufacturing system and lighting control.

(6) Execute recipe

A recipe is provided by its exclusive graphic object and can be executed and controlled on the run time graphic screen.

In addition, execution can be defined from Action Trigger. Providing API available in Script ensures control at more diverse situations.

1.1.8 Robust Graphic Engine

(1) Symbol Library

It supports more than 1000 symbol library images and its drag & drop provides a more convenient usage way.

(2) Support various types of graphic files

BMP, GIF, JPEG, PNG, TIFF, ICON, WMF, EMF, Animation GIF and other various graphic files are supported.

(3) Engineering Template

A certain screen can be more frequently and conveniently reused with its template object and plant screens and transmitted parameter values.

(4) Provide exclusive graphic objects for stable extension:

Alarm view (real-time, history), trend (real-time, history), recipe view, list trend (real-time, history) and gauge control are provided for convenient use.

(5) Provide accurate and fast screen mobility

The user can display part of the screen or the entire screen or move to the work area he/she want with Zoom, panning or navigation function to more accurately and fast identify the relevant process.

(6) Layer

Hide/UnHide and Lock/UnLock functions can be used after selecting a graphic object.

(7) Built-in Graphic Script

When it is insufficient with the screen settings only because of complicate dynamic components of the graphic screen, various functions can be used from the built-in scripts and internal functions of the graphic editor.

(8) ActiveX Control

Direct control over the Company's unique control functions such as Alarm Viewer and Trend and other company's ActiveX enables efficient and high-functional graphic works. It supports virtual keyboard and window to input various values for panel PC.

1.1.9 Cross Reference

(1) Simple correction

With Cross Reference, the user can trace the interrelation and immediately correct engineering error without making unnecessary effort to find each error in the tag and the drawing to change or delete it.

(2) Accident Analysis during the run time

When alarm occurs during the run time, the user can use Cross Reference in the alarm screen and the alarm pop-up screen to move to the relevant plant screen or Trend and analyze the accident fast and conveniently.

1.1.10 Web Server

(1) Simple Web Conversion

In the InfoU Web server, existing projects can be automatically converted to the web through the Wizard function so that even a user who does not have any expertise on web server or website configuration can easily configure a website.

(2) Monitoring system without restriction to place

InfoU is possible to control monitoring only with web browser and monitor history data through Trend as well as real-time tag data without additional installation or restriction to place.

(3) Configuration with various types

The user can create a new web site by configuring a site he/she wants and form a link connection by adding virtual folders to existing web sites.

1.2 System Requirement

At least the following environment is required to execute this program;

- (1) IBM-PC compatible computer with Intel Pentium 3 or higher
- (2) Memory of 512MB or higher
- (3) VGA graphic card and monitor with basic resolution 1024 X 768 or higher
- (4) CD-ROM drive (8X or higher)
- (5) MS Windows 2000/XP/2003
- (6) Excel (MS Office 2000/XP/2003)
- (7) Hard disc space required of 1G or more
- (8) Port of USB 1.0 or higher

1.3 License

- (1) USB Lock Key
 - (a) The use of the project may be restircted depending on the following conditions(1-hour use).
 - (b) Restriction to the number of I/O tags: When more number of I/O tags is using than the designated number to the key
 - (c) Project option: Even though there is the following correlation among project options; stand alone < server project < redundancy < web option, it restructs to the number of concurrent users and clients if any upper-level project option is selected with a lower-level option key.
 - (d) Web version: Restrict to the number of concurrent users
- (2) If the user uses excessive functions than the assigned ones to the lock key, demo mode will run for one hour.
 - Example of use 1> If you run more I/O tgas than 500 tags with the lock key for 500 tags, yout program will end after one hour of run time and engineering tool execution.
 - Example of use 2> If you set up the web project with the lock key for a signal system, the program will end after one-hour run time.

1.4 **Product Composition**

Classification	Production Name	I/O Tag	Other
	InfoU-RC-75	75 tags	
	InfoU-RC-150	150 tags	
Developer Version (Engineering+Real-time	InfoU-RC-500	500 tags	
operation)	InfoU-RC-1K	1000 tags	
	InfoU-RC-2K	2000 tags	
	InfoU-RC-MAX	Unlimited tags	
	InfoU-RT-75	75 tags	
	InfoU-RT-150	150 tags	
Executor Version	InfoU-RT-500	500 tags	
(Real-time operation)	InfoU-RT-1K	1000 tags	
	InfoU-RT-2K	2000 tags	
	InfoU-RT-MAX	Unlimited tags	
	InfoU-OP-WEB2	Unlimited tags	Concurrent users: 2 Users
Web Version	InfoU-OP-WEB5	Unlimited tags	Concurrent users: 5 Users
(Engineering + Real-time	InfoU-OP-WEB10	Unlimited tags	Concurrent users: 10 Users
operation + Web)	InfoU-OP-WEBMAX	Unlimited tags	Concurrent users: Unlimited Users

1.5 Caution

- (1) Please, be cautious to keep and retain the enclosed license key.
 - Any lost license key will not be reissued and some program operations and functions may be restricted depending on the license key.
- (2) This software shall be operated only in the authorized OS and installation environment. The entire or part of the program may generate some errors or malfunctions if it is installed and used in any unauthorized OS.
- (3) Checklist when any error occurs during the program installation or the initial execution

 Please, be aware that some program errors may occur because the vaccine program in your computer may block the program's communication port or service.
- (4) Check your history data volume
 - Accumulation of history data such as alarm history and data history may cause insufficient storage space in the system and lead to the occurrence of error.
- (5) When carrying out engineering works, take into account the system performance

 If you register and use too many tags or log in excessive history data over the system performance, the program may become significantly slower or generate malfunctions during the engineering or run-time operation.
- (6) Be cautious when preparing a script
 - When preparing a script, excessive system load or malfunction may occur due to the script which is omitted from unlimited loop by logical error or the one that requires long processing time.
- (7) Be cautious about logging data loss when adding a tag after deleting it
 - There is no problem with the logging setting even though you delete a tag, which is set up in the logging group, from the tag database and then add it again with the same name, however, you may loss some logging data.
 - It is better to work with tag edition or CSV file update function rather than adding a tag after deleting it when changing tag attributes,

Chapter 2 **System Composition**

2.1 **Software Component**

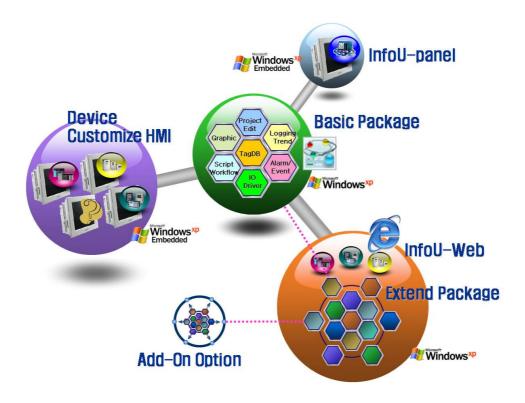


InfoU software is made up of the components as seen in the figure above and the structure of this software allows the user to add other various components to its basic ones and those newly added components are internally connected through data communication and API to act dependently on each other.

In the engineering environment, integral engineering tool InfoUD and graphic editor InfoUG are provided in additional execution files and Report is provided as an additional Excel function.

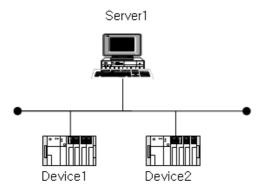
In the run time environment, the components above and I/O drivers are activated by additional execution files and play a each one's role depending on the engineering setting environment.

Chapter 2 System Composition



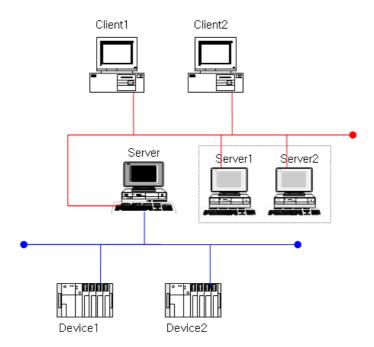
Various types of software packages can be provided by re-composing InfoU's functions and options as seen in the figure above.

2.2 **Stand Alone System**



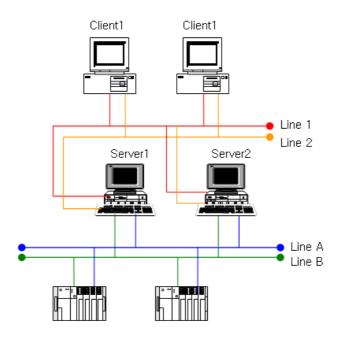
Only with this stand alone system, the user can monitor and control the data status of multiple devices.

2.3 **Client/Server System**



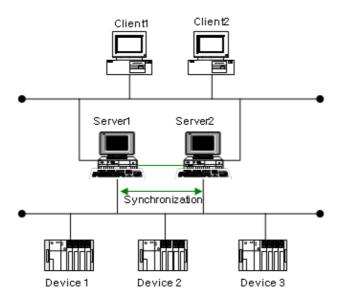
- Remote/multiple monitoring and controlling the system is available.
- Time synchronization between server and client is provided.

2.4 **Line Redundancy**



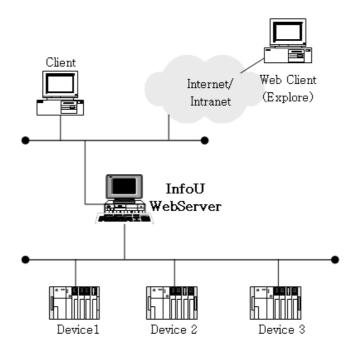
- Line redundancy with the device port such as PLC is available.
- Ethernet port redundancy of Server 1, Server 2 and Client
- An additional switcher is needed for signal communication between system and device, .

2.5 **Redundancy System**



- When the Run Server is failed, the Standby Server will be activated through auto switching to provide a reliable service.
- Time synchronization among Server 1, Server 2 and Client
- Monitoring to check for any error of the partner server is available through RS-232C communication line.
- Project duplication: A redundancy project can be automatically duplicated to prevent error and to provide convenience to the user.
- For signal communication between system and device, an additional switcher is needed.

Web Server 2.6



- Without any need to conduct additional engineering works for the projects in the InfoU server, real-time monitoring and controlling is available in the Web Client.
- Without installing any additional client program, monitoring is available only through MS Explore web browser.
- Data such as Trend (real-time history), Alarm (real-time history) can be monitored simultaneously as if the user is in the site.

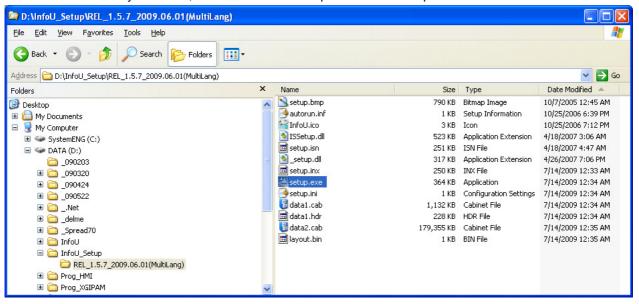
Chapter 3 Install and Start

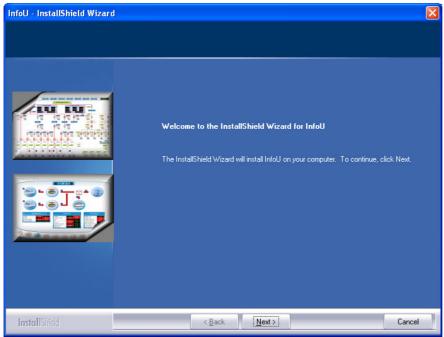
3.1 Install

3.1.1 **Arrange installation**

Once the setup CD are inserted in the computer's CD/DVD input device, the installation screen automatically appears in the user's computer as seen in the splash screen below.

If it is not automatically executed, execute the CD's setup.exe file in the explorer.





Press [Next(N)] button.

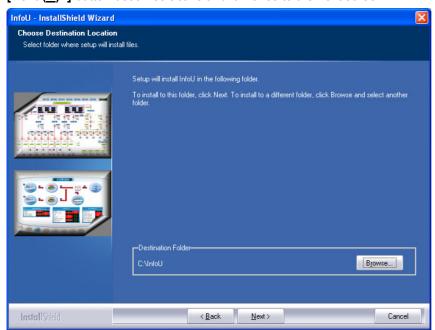
Chapter 3 Install and Start

If you want to change the previous screen's setting, press [<Back (B)] button or, if you stop installation, press [Cancel] button. Then, the installation work will stop.

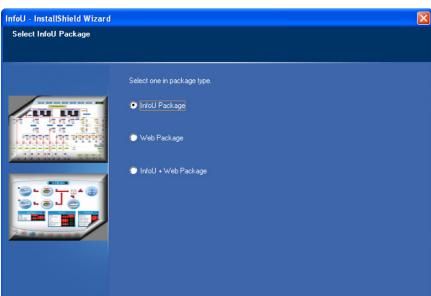
3.1.2 **License Contract**



In case that the printer is already set up in the system, the terms and conditions of the license is printed upon the click on [Print (P)]. Only after you press the radio buttone of "I agree on the terms and condtions of this license", **[Next (N)]** button becomes active and it moves to the next screen.



Then, you may select a path to install.



< <u>B</u>ack

3.1.3 Select a Package to Install

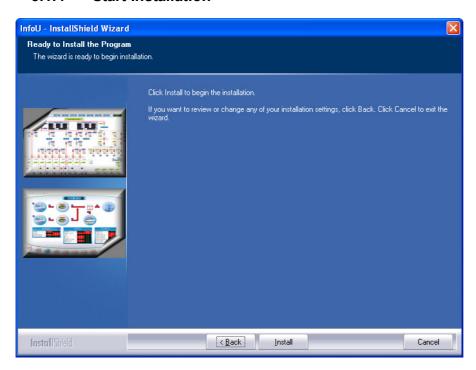
There are three types of packages you may install.

InstallShield

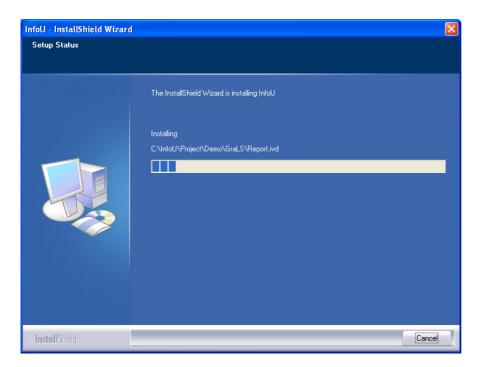
- InfoU Package Package that consists of a wider range of applications by using InfoU LS Industrial System's universal HMI software package.
- Web Package InfoU's extend package that allows the user to monitor and operate the screen functions presented by InfoU with Internet browser.
- InfoU + Web Package Package that allows the user to configure No 2 and No 3 packages in one computer.

Click [Next (N)>] button after selecting one,

3.1.4 **Start Installation**



After completing settings, press [Install (I)] button to start installation.



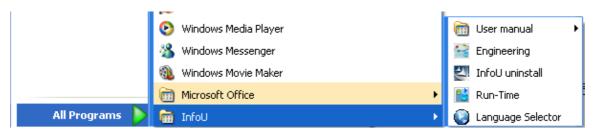
3.1.5 **Complete Installation**



Once installation is completed, the computer shall be restarted for the program's normal execution.

Select the radio button "Yes, I will restart the computer now" and press [Complete] button, the system will restart after few minutes. Once the system restarts, all of the installation works are completed.

3.1.6 Menu and Shortcut

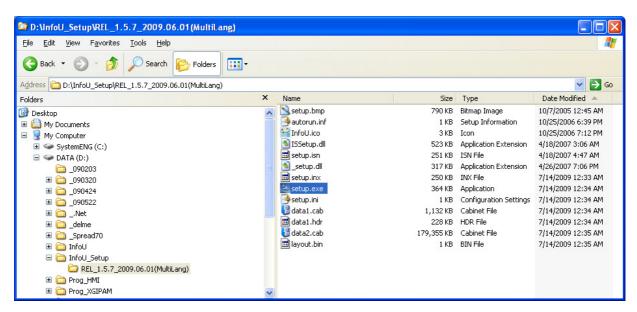


If installation is normally completed, [User Manual] [Run-Time] [Engineering] and [Language Selector] menus are created in the new "InfoU" program list.

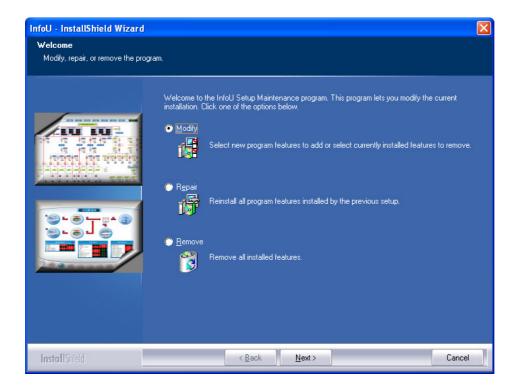


Shortcut icons for Runtime and Engineering are created in the desk top.

3.1.7 Reinstall InfoU



If you want to modify, restore or remove the installed program, execute the CD's setup.exe file in the explorer.



- Modify: Change the installed package.
- Restore: Reinstall the previously installed package.
- Remove: Remove the installed InfoU.

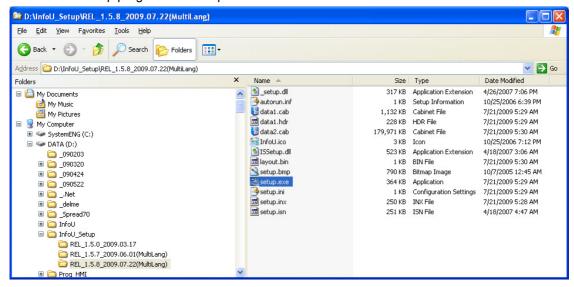
3.1.8 **Update Program Patch**

To upgrade the existing InfoU version, the following two patch ways are available.

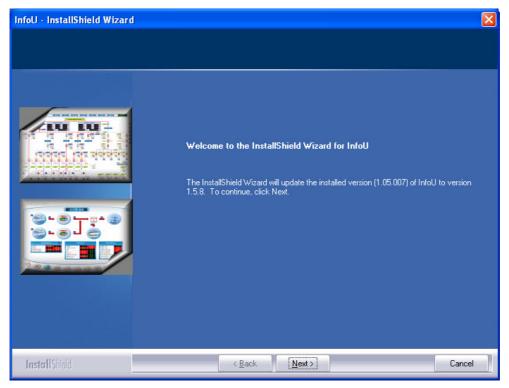
- Reinstall after removing: First remove the installed InfoU and reinstall it as a Full Setup program.
- Install patch: If installed previous version the patch is installed.
- Install update patch without removing the existing program.

To download the latest version of Full Setup and Patch Program, visit LS Industrial System's product homepage www.lsis.biz.

Execute the update program patch Execute the Setup program at the explorer.

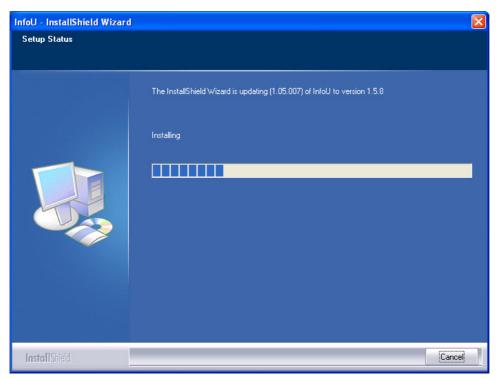


The following shows that the version update program is normally in progress.

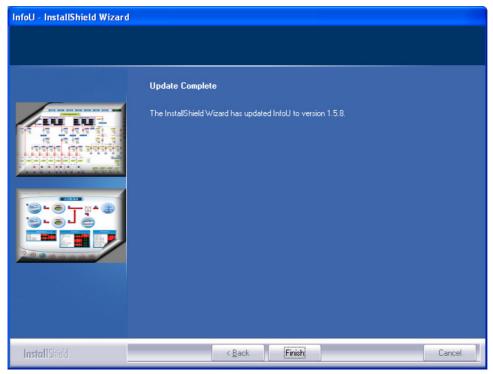


Click [Next] to execute update patch

Chapter 3 Install and Start



Progress is displayed.



This screen shows that update patch is successful completed.

3.2 Start and Stop

3.2.1 Start InfoU

- How to execute engineering (InfoUD)

 Select [Engineering] from the program menu to execute InfoUD- InfoU's integral engineering tool.

 You may also click and execute the shortcut icons generated during the program installation.
- How to execute run time (InfoUD)
 Select [File]-[Runtime]-[Start InfoU Run-Time] from the engineering (InfoUD) program menu or select the tool bar icon to execute InfoUR, InfoU's Run-Time, along with other Runtime component programs.

At this time, the default project that starts to run is the project on which the user worked most recently or the demo project included when installing.

You may also click and execute the shortcut icons generated during the program installation.





Chapter 3 Install and Start



3.2.2 Stop InfoU

- How to stop engineering (InfoUD)
 - Select [File]-[Exit] from the menu to end InfoU.

At this time, only the engineering (InfoUD) program will end without having influence on the run time in effect.

How to stop run time (InfoUD) Select [File]-[Runtime]-[Stop InfoU Run-Time] from the engineering (InfoUD) program menu or select the tool bar icon to end Runtime.





When the engineering (InfoUD) program is not in effect, you can also end the project when ending the graphic runtime.

You may select project end menu from the tray icons to end.

[See Chapter 5: Runtime]

Chapter 4 Engineering

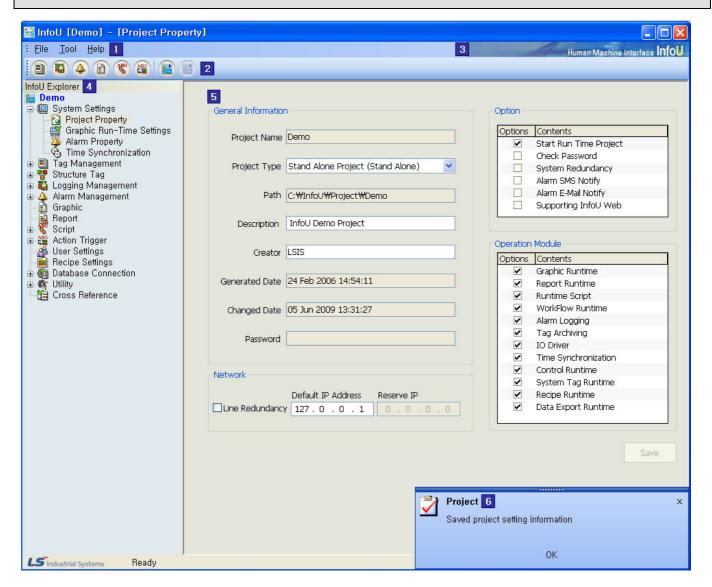
As a tool used for configuring a monitoring and control system and developing an application, InfoU Engineering (InfoUD) allows the user to define system environments needed for executing Run-Time and set up device configuration, tag, logging and alarm. Engineering settings are managed on a project basis.

The screen below is the one that appears when the program starts executing and it is made up of menu, tool bar, tool (application) explorer and splash screen. If the program is executed, the default project will be loaded and the project information screen will appear. The project on which the user worked most recently or the project that is currently running is designated as the default project.

If Close Project is executed or Open Project fails, the project list screen will appear.

Since this chapter covers only the default screen, menu and tool bar, please refer to [Help] if you need more information on each application function.

4.1 Screen Composition

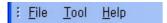


Chapter 4 Engineering

The main screen consists of the following;

- (1) Menu bar: The user can select a menu from the menu bar made up of functions that can be used for the application
- (2) Tool bar: The user can execute a frequently used application program with one click on the tool bar.
- (3) Identity Zone: A zone to display the identity of the InfoU program.
- (4) InfoU Explorer: Icons displayed in a tree type to select an application program
- (5) Main View: An area in which the application program is executed.
- (6) Pop-up Window: Notice is displayed on the system's status or the user event for application programs.

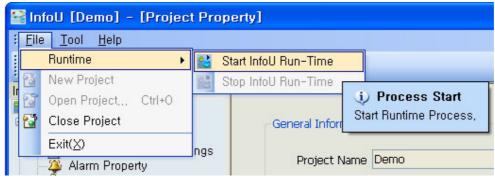
4.2 Menu and InfoU Explorer



Function menus to be used in InfoUD Engineering are provided.

4.2.1 File

Basic menus such as Start InfoU Run-Time/Stop InfoU Run-Time, Open Project/Close Project, Print and other menu are provided.



- (1) Runtime: The user can start or stop InfoU Run-Time.
- (2) Open Project: The user can open an InfoU project.
- (3) Close Project: The user can close the current project.
- (4) Exit: The screen displayed in the current menu view will end and a new empty screen will be displayed.

Caution

* If any project is running in real-time, other project can not be selected even after the project is closed in the

(a) Project Folder and File

Once a new project is create in InfoU, the following folders and files will be managed.

The following folders are created having the same names with projects.

Folder Name	Description			
Archive-AlarmLog	Path to save alarm event logging files			
Archive-TagLog	Path to save data logging files			
Config	Path for configuration files needed for running			
CrossRef	Path for files related to cross reference			
GraLS	Path for graphic screen files			
I/O Driver	Path for driver configuration files			
Log	Path to save application logs			
Pacakage	cakage Path for server package files			
Report	eport Path for configuration files related to report			
Report-Form	eport-Form Path for report form files			
Report-Data	eport-Data Path to save report creation files			
Script	Path for script files			
TagDB	gDB Path to save all of the engineered setup information database			
Wav Path for sound file to generate salarm sound				

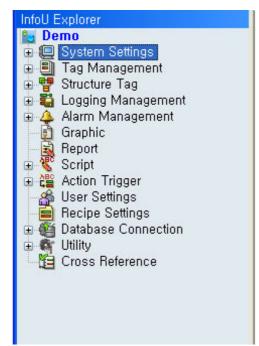
(b) File type related to project

Item	Extension	Description		
Project	.lsp	Project information files		
	.adb	Alarm event logging database files		
Alarm Logging	.rec	Real-time alarm saving files		
Tag Logging	.db	Logging database		
Config	.ini	Configuration files needed for running		
Database	.dbf	Engineered actual database files		
Dalabase	.cdx	Engineered setup database files		
Graphic	.ivd	Graphic file		

4.2.2 Tool

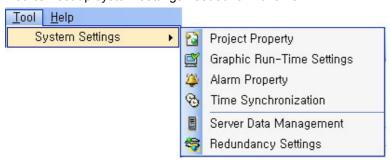
You can execute various functions that need for your engineering works. Tool menu and InfoU Explorer perform the same functions.

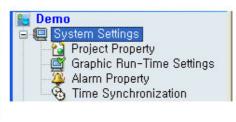




(1) System Settings

You can set up system settings needed for Runtime.

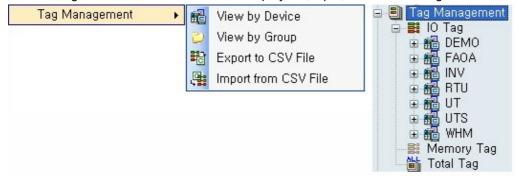




- (a) Project Property: The user can modify project information and options.
- (b) Graphic Run-Time Settings: The user can set up the splash screen and style of the graphic runtime screen.
- (c) Alarm Property: The user can set up overall settings related to activating an alarm
- (d) Time Synchronization: The user can set up time synchronization property between system and device.

(2) Tag Management

You can organize devices to be used for the projects, input, edit or delete tags.



- (a) View by Device: A list of devices is displayed for the user to set up a tag corresponding to the relevant device.
- (b) View by Group: A list of tag groups is displayed for the user to set up tags by group.
- (c) Export to CSV File: The user can save the current project tag in a form of CSV file.
- (d) Import from CSV File: The user can select a CSV file to input, edit or delete a large amount of tags.

(3) Structure Tag

A Structure Tag template and item list is displayed.



(a) Structure Tag: The user can input, edit or delete structure tags.

(4) Logging Management

You can set up and manage logging groups of the tag data needed for the user's history management. Logging groups can be inquired from Trend, Report, List Trend and Script.



- (a) Cycle: The user can set up regular cycle logging groups
- (b) Statistics: The user can set up statistics logging groups
- (c) Accumulation: The user can set up accumulation logging groups
- (d) String: The user can set up string logging groups

Chapter 4 Engineering

(5) Alarm Management

You can manage alarm event information by setting up alarm groups.



(a) Alarm Management: The user can input, edit or delete alarm groups.

(6) Graphic

A list of graphic files currently used for the project is displayed.



(a) Graphic: The user can click a file from the InfoU Graphic file list in the project "GraLS" folder and edit it in the graphic editor.

(7) Report

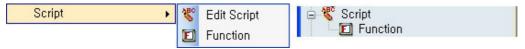
A report list is displayed.



(a) Report: The user can execute Excel and prepare a report in Excel sheet by double clicking an item from the report list.

(8) Script

You can edit a script and inquire an internal function list.



- (a) Edit Script: The user can edit a script document.
- (b) Function: A list of the internal functions presented in InfoU is displayed.

(9) Action Trigger

With this function, you can execute Edit Action Trigger and Holiday Settings for executing script, recipe and database.



- (a) Edit Action Trigger: The user can add, modify or delete action trigger.
- (b) Holiday Settings: The user can add, modify or delete holiday settings.

(10) User Settings

The user can set up user-related information and use these settings when receiving the authorization to control them in the alarm action and graphic.



(a) User Settings: The user can set up his/her personal information such as his/her authorization to control or contact information.

(11) Recipe Settings

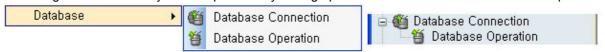
The user can set up control over designated patterns by inputting items, units and data of a recipe.



(a) Recipe Settings: The user can add, modify or delete a recipe from the displayed recipe list.

(12) Database

Sharing data with other systems is possible by setting up Database Connection and Database Operation.



- (a) Database Connection: The user can manage settings to connect with external database.
- (b) Database Operation: The user can define the data input and output relationship with external database.

(13) Utility

It supports convenient engineering works by registering and setting up various components.



- (a) Unit Message Registration: Various industrial units registered in this menu are very helpful to select a unit easily during the tag edit.
- (b) Alarm Message Registration: If the user registers alarm messages that will be displayed in case of any alarm event and select those during the tag edit, he/she can inquire those messages when alarm actually takes place.
- (c) On/Off Display Registration: The user can register display methods according to the status of digital value to use them during the tag edit.
- (d) LS IS Program Registration: The user can register LS Industrial System's PLC Programs and execute them in the engineering screen and also input a tag automatically.
- (e) Application Registration: The user can register external programs and execute them in the engineering screen.

(14) Cross Reference

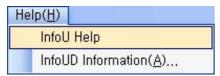
The Cross reference relationship in the project between tag and object is displayed to allow the user to identify any engineering error.



(a) Cross Reference: The cross reference screen is displayed.

Chapter 4 Engineering

4.2.3 Help



- (a) InfoU Help: The Help screen is displayed.
- (b) InfoU Information: Information on the InfoU version is displayed.

4.2.4 Tool Bar



Tool Bar	Name			
	Tag Management			
3	Logging Management			
4	Alarm Management			
	Graphic			
	Script			
ABC	Action Trigger			
	Start InfoU Run-Time			
	Stop InfoU Run-Time			

4.2.5 Pop-up Window



- (a) Notice to inform the system status and user event for application programs is displayed.
- (b) Notice is displayed in the messenger-style window.
- (c) This window disappears few minutes later after it pops up.

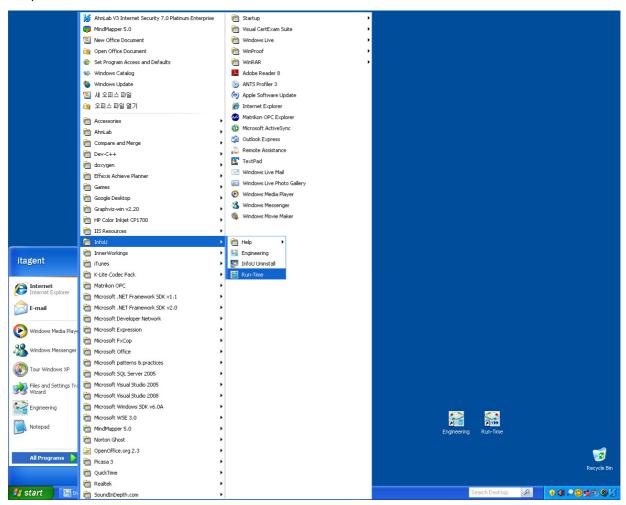
Chapter 5 Run Time

With this function, the user can execute the project registered for engineering settings.

After establishing a communication link between system and device according to the setting information on the channel and station, the user can monitor the real-time site status with the attained data in the plant screen. Also, the user can perform various functions such as history logging and report printing by generating alarms of the major monitoring points.

Start InfoU Run-Time 5.1

To start InfoU Run Time, click Start in the window screen to find a program directory or click the Run Time icon in the desk top.

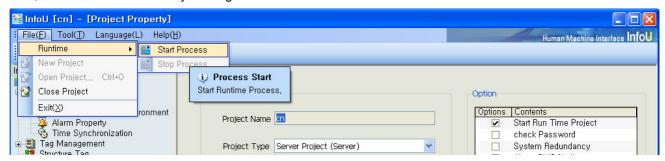


Caution

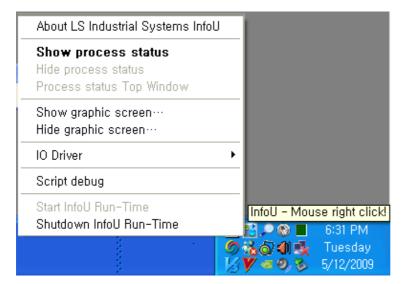
- * Once "Start Runtime Process" is selected in the Project Property, Run Time begins executing and a project designated as Default starts.
- * If the above option is not set up, it will remain in standby mode after Runtime Process is executed.

Chapter 5 Run time

Also, the user can execute it by clicking File →Runtime->Start Process in order.



For Runtime, Start Process Mode or Standby Mode is retained when starting the program depending on the project settings.



Tray icons of the system are displayed as seen in the figure above and the user may use the following menus by clicking one of them.

(1) Standby Mode Menu

- (a) Start InfoU Run-Time: Runtime Process starts when a project is executed. When any project is not open, select a project to execute.
- (b) Shutdown InfoU Run-Time: Both of the standby mode and InfoU Run-time end.

(2) Operation Mode Menu

- (a) Shutdown InfoU Run-Time: The project being operated in real time ends.
- (b) Show process status: The monitoring screen for process status is displayed.
- (c) Show graphic viewer: The graphic monitoring screen is displayed as an upper-level screen.
- (c) Hide process status: The graphic monitoring screen is hidden. Even though it is not displayed, it is being executed in the background.
- (e) I/O Driver --> Device Status: The status of a device connected and set up is displayed.
- (f) I/O Driver --> Simulation settings: The simulation setting screen for the set tag is displayed to generate data on the device values.

(g) Device Data: A screen is displayed to allow the user to inquire device data.



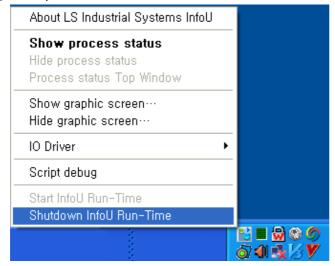
Once the Runtime starts, the following splash screen appears to show the process status.



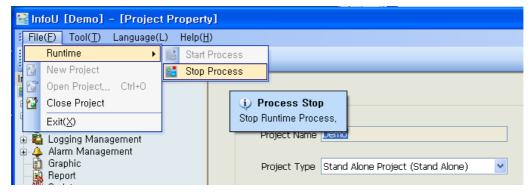
Shutdown InfoU Run-Time 5.2

All of the functions operated during Runtime will stop when this function is selected. This function is executed as follow:

(1) To stop InfoU Runtime Process, select Shutdown InfoU Run-Time from Runtime.



(2) Click Runtime --> Shutdown InfoU Run-Time to execute.



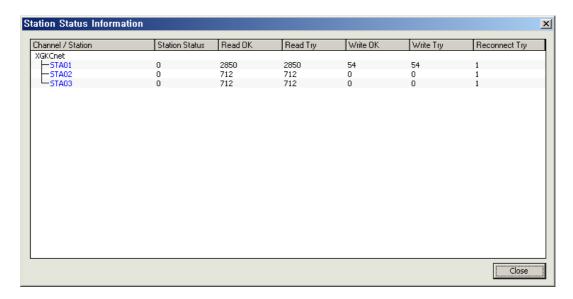
(3) Once InfoU Run-time ends, the following splash screen appears to show the process status.



5.3 **Device Status**

This function is executed as follow:

- (1) Select [IO Driver] --> [Device Status] from Run-Time Tray icon Menu to execute.
- (2) Select [Work] --> [IO Station Status] from Graphic Run-Time Program Menu to execute.
- (3) The user may use ShowNetworkStatus() from Graphic Script Function to execute.
- (4) Devices in blue color in the device list indicate that they are in normal status and ones in red color mean that they are in abnormal status in communication.

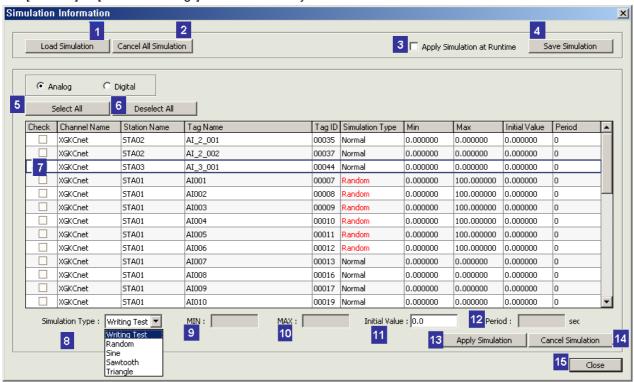


- (a) Channel/Station: The communication configuration and status of the device is displayed.
- (b) Station OK: It shows the communication status of the device and 0 means normal, 1 means abnormal.
- (c) Read OK: The number of cases in which device values have been successfully read since the project started is displayed.
- (d) Read Try: The number of cases in which reading device values has been attempted since the project started is displayed.
- (e) Write OK: The number of cases in which device values has been successfully written since the project started is displayed.
- (f) Write Try: The number of cases in which writing device values has been attempted since the project started is displayed.
- (d) Reconnect Try: The number of cases in which connecting with the device has been attempted is displayed.

5.4 Simulation settings

As a function to generate virtual device values, the user can simulates virtual device values with this function before connecting a device to identify any error of the engineering work in advance.

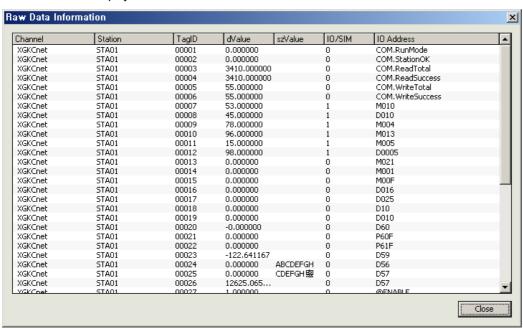
Select [IO Driver] -> [Simulation settings] from Run-Time Tray icon Menu to execute.



- (1) Load Simulation: The saved simulation files are loaded to set up simulation.
- (2) Cancel All Simulation: All of the simulation settings that have been set up until now are cancelled.
- (3) Apply Simulation at Runtime: When starting Runtime, the user can apply simulation immediately to set up.
- (4) Save Simulation: The current simulation information is saved in a file.
- (5) Select All: All of the tags displayed in the tag list window are selected.
- (6) Deselect All: All of the selected tags in the tag list window are cancelled.
- (7) Tags are displayed depending on the selection either analog or digital. Multiple selections are available.
- (8) Simulation Type: Select a rule to generate simulation data.
 - Analog: The user can select one among Writing Test, Random, Sine, Sawtooth and Triangle,
 - Digital: The user can select one among Writing Test, Random and Change,
- (9) MIN: The user can set up the lowest value of the range of change when generating data.
- (10) MAX: The user can set up the highest value of the range of change when generating data.
- (11) Initial Value: The user can input the initial value when applying the simulation rule.
- (12) Period: Input a period of one cycle of the simulation type that has a certain cycle (Sine, Sawtooth and Triangle).
- (13) Apply Simulation: The user can apply simulation to the selected tags.
- (14) Cancel Simulation: The user can cancel applying simulation to the selected tags.
- (15) Close: The current close ends. Even after the screen ends, simulation data keep operating.

5.5 **Device Data**

The user may monitor raw data values attained from the communication with the device. Simulation data are displayed when device values are simulated.



Chapter 6 System settings

In this step, you may create a project and manage basic environment settings for Runtime.

6.1 **Project**

As the first step needed for Runtime, the user can create a project and manage configuration settings and options in this step.

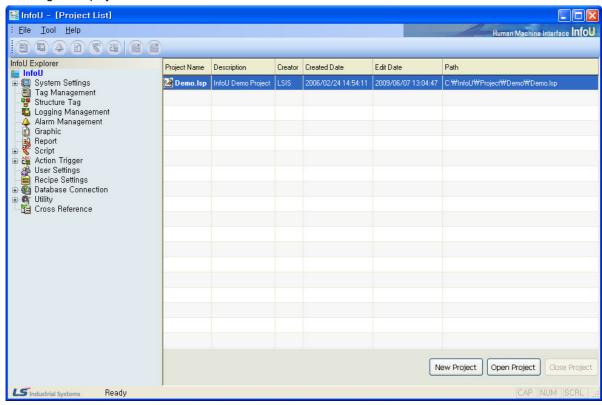
Each project becomes an unit subjected to Engineering and Run-time management and the user can move, copy or use projects to other system, drive or path since they are created and managed as a batch and under each project unit, sub-paths such as tag, logging, alarm, graph, report and script are created and managed.

It can easily manage numerous projects while providing easy copying and transferring contents between projects and efficient engineering works.

6.1.1 **Create New Project**

A function New Project can be executed after closing the previous project.

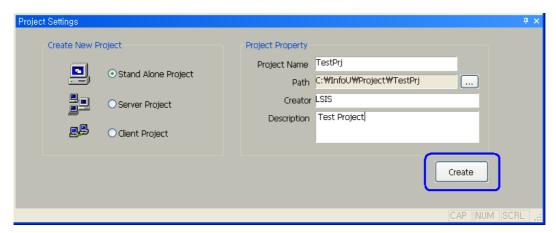
Once the project is successfully closed, the latest project list is displayed. Click [New Project] button to perform the work of creating a new project.



Caution

^{*} If any project is currently operated, a new project is not created.

Chapter 6 System Settings



(1) Project Type:

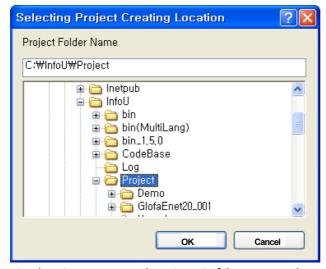
- Stand Alone Project: A stand alone project is created.
- Server Project:project that is activated by the server mode of the client to server system is created.
- Client Project: A project that is activated by the client mode of the client to server system is created.

(2) Project Name:

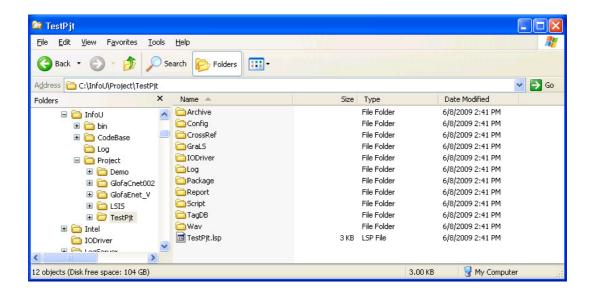
Input a project name.

(3) Path:

- It refers to a path where a new project is created. If [...] button is selected, the following screen will appear to allow the user to select a path.
- A new folder with the same name as the project is created in the subfolder of the folder designated by the user to create a new project.



- (4) Creator: Input a name or a department of the person who creates the project.
- (5) Description: Input detailed information.
- (6) Create: A new project is created. After the new project is created, the project will open.
- (7) Once the new project is created, a new project file with a project folder and extension ".lsp" will be created.



6.1.2 **Project Property**

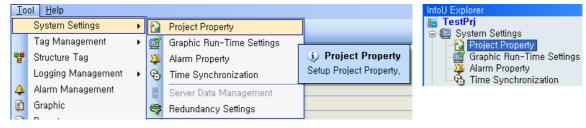
You may set up general properties of the project and basic configurations of the network. Also, you may set up project options and execution modules suitable for your purpose.

* Project options can be set up depending on the authorized scope by the system license. Refer to the following example:

If the license key is of developer or executor version and "Web Function Support" is selected as an option, the demo mode will operated for one hour resulted from the violation of license.

The screen will be displayed as follow:

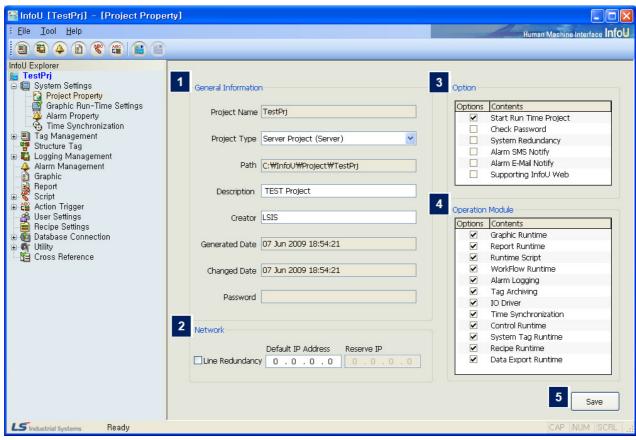
-. If the engineering program is executed, the Project Property screen of the basic project is displayed



- -. Select Project Property from InfoU Explorer or
- -. Select route item (project name) from InfoU Explorer.



Chapter 6 System Settings



(1) General Information

- (a) Project Name: The name the user gives when creating a project is displayed and it is not editable.
- (b) Project Type: The selected project type is not editable.
- (c) Path: The location of the project is displayed.
- (d) Description: The user can give additional information on the project name.
- (e) Creator: The name of the first creator is displayed.
- (f) Generated Date: The date when the project is created for the first time is displayed.
- (g) Changed Date: The date when the project is latestly changed is displayed.
- (h) Password: If the option "Check password when the program opens" is selected, the user needs to input the password.

(2) Network

- (a) Line Redundancy: If Line Redundancy was set up in the client and server project, this function is selected.
- (b) Default IP Address: The user needs to write the IP Address he/she will primarily use.
- (c) Reserve IP: The user needs to write the IP Address he/she will use in emergency.

(3) Option

- (a) Start Run Time Project: It is automatically executed in a runtime mode while InfoUR is executed. If it is not set up, the standby mode is executed. When creating a project, the user may select Default.
- (b) Check Password: The user needs to input the password to begin the engineering work when opening the project.
- (c) System Redundancy: When starting the server, the user needs to set up whether to start it as redundancy.
- (d) Alarm SMS Notify: The user needs to set up whether to provide the alarm message service by SMS or not when any alarm occurs.
- (e) Alarm E-Mail Notify: The user needs to set up whether to provide the alarm message service by e-mail or not when any alarm occurs
- (f) Supporting InfoU Web:As an option, the user needs to decide whether to provide the Web service support or not.

(4) Operation Module

- (a) The user can limit program functions by selecting functions to execute from the process list.
- (b) This function is intended to restrict unnecessary functions for tests when using a low-capacity hardware.

Caution

- * Select All is a default setting and if the user limits functions voluntarily, unexpected results may be prompted because the program settings may effect on the interrelation between functions.
 - (c) Graphic Runtime: If it is not selected, only data service is executed as background without the plant monitoring control.
 - (d) Report Runtime: The function to automatically output or save the report is restricted.
 - (e) Runtime Script: The function to execute the scrip is restricted.
 - (f) WorkFlow Runtime: The function to designate the scrip operation is restricted
 - (g) Alarm Logging: The function to save alarm event history is restricted
 - (h) Tag Archive: The function to save history data is restricted
 - (i) IO Driver: The function to communicate with the device is restricted
 - (j) Time Synchronization: The function for time synchronization is restricted.
 - (k) Control Manager: All of the functions set up not only in the screen but also in the scrip are restricted.
 - (I) System Tag Runtime: The function to generate system tag data is restricted.
 - (m) Recipe: The function for recipe is restricted.
 - (n) Data Export Runtime: The function for data export is restricted.

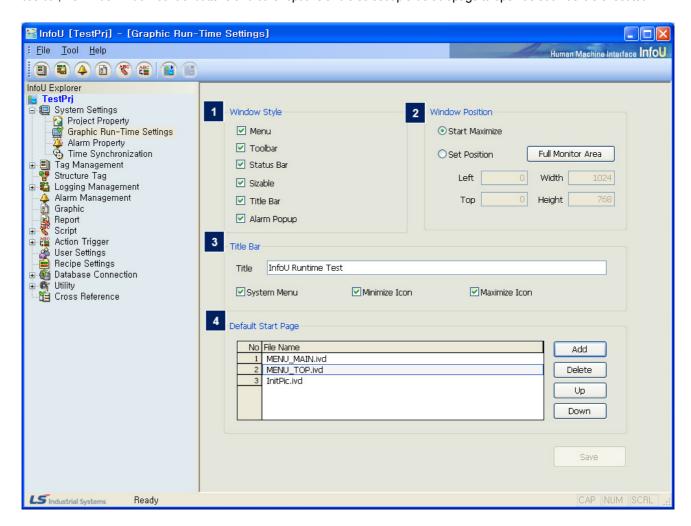
(5) Save

- (a) The selected options are saved.
- (b) The options selected during Runtime are not applied, therefore, the user needs to restart the program after closing it to apply the selected options.

6.2 Graphic Run-Time Settings

In the graphic execution screen, the user can set up environment for the graphic run-time (InfoUX.exe).

For environment settings, the user can set up menu, view/hide toolbar, initial execution position, name to be displayed in the title bar, view/hide window control buttons and other options and also set up a default page to open as soon as it is executed.



- (1) Window Style
 - (a) Menu: Select whether to view the basic menu.
 - (b) Toolbar: Select whether to view the toolbar.
 - (c) Status Bar: Select whether to view the status bar(d) Sizable: Select whether to allow the operator to change the screen size.
 - (e) Title Bar: Select whether to display the title window located in the top of the window.
 - (f) Alarm Popup: Select whether to display the latest alarm in the pop-up window when any alarm occurs during RunTime

(2) Window Position

- (a) Start Maximize: Select whether to execute RunTime in the maximum screen.
- (b) Set Position: The user can designate the location and size of the screen during Run Time.
- Click "Full Monitor Area" button, the area coordinates for the multiple monitors are displayed. The coordinates reference
- (0, 0) is for the main monitor.

(3) Title Bar

- (a) Title: Input the title of the screen.
- (b) System Menu: Select whether to display system menus in icon types in the left top of the screen.
- (c) Minimize Icon: Select whether to display minimized buttons in the right top of the screen
- (d) Maximize Icon: Select whether to display maximized buttons in the right top of the screen

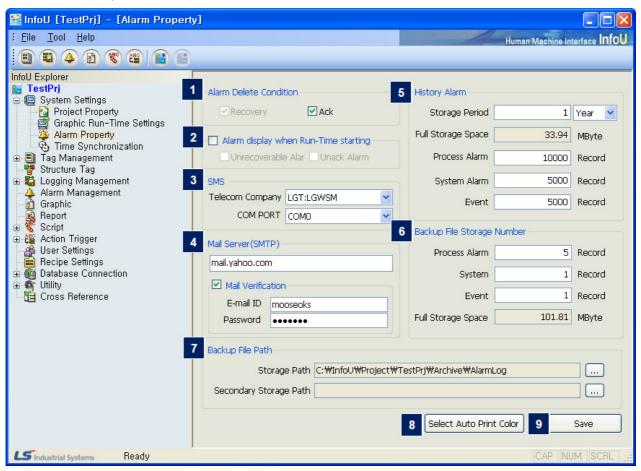
(4) Default Start Page

This is for selecting screens to be displayed at first when executing Run Time. If you press Add button, all of the written pages for the current page will be listed up and you may select one to be used as a splash screen.

6.3 Alarm Property

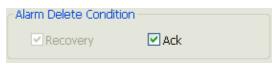
You may set up items needed for handling and operating alarms. You may set up the number of alarm events to save and the criteria to handle real-time alarms (Criteria to delete or designate auto printing and color, etc) and also set up common options (SMS, mail) needed to handle alarms.

Select Alarm Property from InfoU Explorer.



(1) Alarm Delete Condition

Set up a time to delete alarm events from the real-time alarm memory.



(a) Recovery

When the alarm is recovered, it is deleted form the real-time alarm memory (Default)

(b) Ack

Only when the alarm is acknowledged, it is deleted from the real-time alarm memory.

(2) Alarm display when Run-Time Starting

Set up whether to cancel the real-time alarm when restarting. If not set up, the real-time alarm is reset when the system restarts and the number of alarm begins from 0.



(a) Unrecoverable Alarm

When starting the program, information on the unrecovered alarm is loaded in the memory.

(b) Unack Alarm

When starting the program, information on the unacknowledged alarm is loaded in the memory.

(3) SMS

Set up details on the SMS message service to send alarm information when it occurs.



(a) Telecom Company

Select a telecom company used for sending SMS

- Currently, only LG Telecom is available.
- (b) SMS COM Port

If the SMS sending terminal is installed, set up COM Port.

(4) Mail Server (SMTP)

A mail server (SMTP) shall be set up to send a mail related to alarm information to the user. Set up a mail server address.

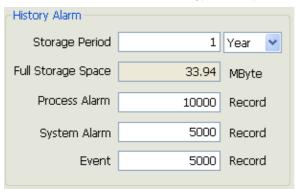


Mail Verification: If an account is needed to connect with the mail server, input the mail ID and password.

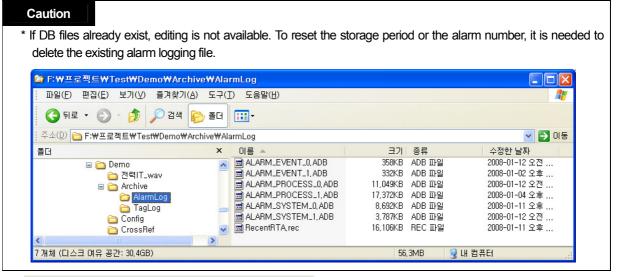
Chapter 6 System Settings

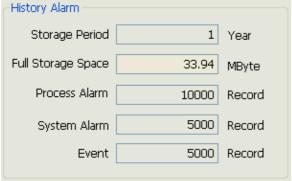
(5) History Alarm

The number of alarms for each alarm type is set up to save them in the history DB.



- (a) Storage Period: It is available to save them in annual, monthly or daily base and select up to 3 years, 36months or 1000days.
- (b) Full Storage Space: The total storage space calculated with the number of the alarms to be saved according to the user's setting is displayed.
- (c) Process Alarm: Input the number of process history alarms to be saved.
- (d) System Alarm: input the number of system history alarms to be saved.
- (e) Event: Input the number of event history alarms to be saved.





(6) Backup File Storage Number

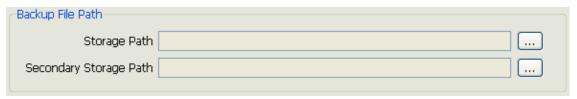
You may set up the number of backup DB files on history alarms.



- (a) Process Alarm: The number to backup DB files on process alarms
- (b) System Alarm: The number to backup DB files on system alarms
- (c) Event: The number to backup DB files on event alarms
- (d) Full Storage Space: The storage space needed to maintain the selected number of backup files is displayed.

(7) Backup File Path

You may a path to save backup files.



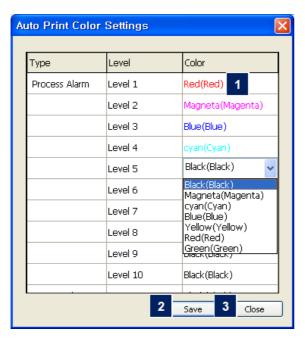
- (a) Storage Path: A path to save backup files is set up.
- (b) Secondary Storage Path: To be prepared for the event that it fails to access to the storage path, set up a secondary storage path to access.
- (c) Path Search: A button to search folders to input each storage path and when this button is pressed, a folder search window will appear.

Chapter 6 System Settings

(8) Select Auto Print Color

You may set up print colors by levels and types when auto printing alarms.

Click "Select Auto Print Color" button.



(a) Auto Print Color Settings

Black is the default color and six colors such as pink, dark green, blue, yellow, red and green are available.

(b) Save

Once color setting is completed, click 'Save' button to save the selected colors.

(c) Close

The window for Auto Print Color Settings is closed.

(9) Save

Alarm settings are saved.

(10) SMSOption-Notepad

If any new modem control device is used in addition to the mandatory SMS terminal, additional instruction for users is presented.

(a) SMSOption.ini File

There is SMSOption.ini file in the bin folder (ex>C:\InfoU\bin) of the InfoU installation folder (If there is no corresponding file, the file is automaticalled created when InfoU starts)

The file contains the following contents:

```
SMSOption - Notepad
File Edit Format View Help
[SMS]
Čount=2
1 = LGT : LGWSM
2=SKT:BSM-856
[LGT:LGWSM]
MSG=AT$SMSMO0=[ToTel],[FromTel],[Opt1],[Msq]
OptCount=1
MsgType=Text
Opt1=4098
Timeout_msec=30000
SuccessMsg=$SMSMOACKO
FailMsg=$SMSMONAKO
Retry=3
[SKT:BSM-856]
MSG=AT$SMSMO=[ToTel],[FromTel],[Opt1],[Opt2],[Opt3],[Opt4],[Msg]
OptCount=4
MsgType=Hex
Opt1=4098
opt2=
opt3=
Opt4=
Timeout_msec=30000
SuccessMsg=$006
FailMsg=$007
Retry=3
```

(b) Edit

[SMS] Item

Specify the count number when adding the SMS terminal number to be displayed in the setting screen.

- 1) 1=LGT: LGWSM (Exist as many as the above count from 1. When setting SMS telecom companyterminal, contents to be displayed in the combo box)
- 2) 2=SKT: BSM-856
- 3) 3=telecom company: Terminal (write SMS terminal number and the telecom company name and terminal name to be newly added.)

[Telecom Company: Terminal]:Set up sending and receiving options for the telecom company and terminal model (see the terminal manual)

4) MSG: Write message types to send/receive to/from the SMS terminal.

[ToTel]: Receiver's telephone number

[FromTel]: Sender's telephone number

[Opt1]:If there is any option, write it.

- 5) OptCount:Write the number of options
- 6) MsgType: Select either Text or Hex (Be caution because specifications are different between
- 7) Opt1,...: Option character to be entered in each option part.
- 8) Timeout msec: Time to wait response from the telecom company
- 9) SuccessMsg: Message code to receive from the terminal when sending is successful.
- 10) FailMsg: Message code to receive from the terminal when sending is failed.
- 11) Retry:Retry number when sending is failed.

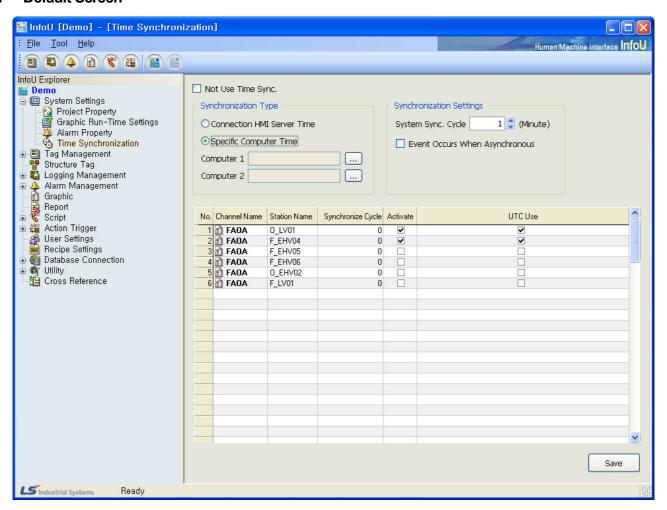
6.4 **Time Synchronization**

To prevent any improper time from being applied to all of the processes used in the system due to any wrong computer time during the real-time operation of the current project, time synchronization is executed by periodically obtaining time from the computer on which the project is executed or other computer.

This can be applied to the stand alone system provided in the InfoU system and other server/client systems and it also provides time synchronization with the I/O device station connected with the InfoU system.

For time in the system, basically UTC (Universal Time Coordinated) is used.

Default Screen 6.4.1



In the engineering main screen, select Time Synchronization from subclauses under System Settings of InfoU Explorer to display the setting screen above. Time Synchronization is selected as a default value he initial setting screen.

(1) General Settings

- (a) Not Use Time Sync
 - 1) If selected: Time synchronization is not used between the computer in which InfoU system is installed and I/O device.
- (b) Synchronization Type
 - 1) Connection HMI Server Time: The computer in which the current InfoU project is operated is designated
 - 2) Specific Computer Time: The computer on the network is designated.
- (c) System Sync. Cycle (Unit: sec)
 - 1) Synchronize time between the entity that attempts time synchronization in the specified interval by second and the computer
- (d) Event Occurs When Asynchronous
 - 1) If synchronization fails, it generates this information as an event after it carries out time synchronization with a local system.
- (e) System Sync. Cycle (Unit: sec)
 - 1) The interval of the computer's time synchronization can be set up by the user.
- (f) Synchronization of the device that provides time synchronization
 - 1) Whether time synchronization is activated or not
 - 2) Even though a device provides time synchronization, it does not perform it if it is not activated. Time synchronization cycle (Unit: sec), Time unit (UTC/ Local Time)
 - 3) For time synchronization, UTC is used as basic time and if not specified, local time can be used.

(2) Device's Time Synchronization

Device's time synchronization is available only if full time synchronization is activated. Device's time synchronization does not apply to the entire devices but only to the stations that provide time synchronization while providing the following list to allow the user to select.

No.	Channel Name	Station Name	Synchronize Cycle	Activate	UTC Use
1	🛐 FAOA	O_LV01	0	~	✓
2	🚺 FAOA	F_EHV04	0	~	✓
3	🛐 FAOA	F_EHV05	0		
4	🚺 FAOA	F_EHV06	0		
5	🚺 FAOA	O_EHV02	0		
6	🛐 FAOA	F_LV01	0		

Also, for device's time synchronization, it can be set up and executed separately depending on whether each device is activated or not even through the user selects a certain time. The user can select either UTC or local time since a certain time display way exist for each station of the device rather than using UTC in the entire stations such as in full time synchronization

6.4.2 Scenario for the whole system

(1) Standalone

A time synchronization way used for projects that are executed independently.

(a) Connection HMI Time

Time is synchronized with the time of its own local system.

(b) Specific Computer Time

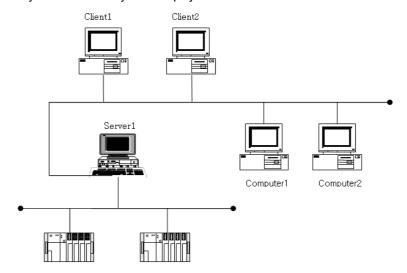
- 1) Time synchronization is performed with the computer assigned as Computer 1.
- 2) If time synchronization fails in Computer 1, it is carried out with the time of Computer 2 after an event occurs (if set up).
- 3) If time synchronization fails both in Computer 1 and Computer 2, it is carried out with the time of local system.

(c) Notice

If time synchronization is carried out with local system due to the failure in time synchronization, time synchronization is attempted with the very first setting information.

(2) Server/Client-Single System

A time synchronization way used for projects that form a server and client relation to execute.



(a) Connection HMI Time

- 1) Server: Time is synchronized with the time of its own local system.
- 2) Client: Time is synchronized with the time of the server connected with the project that is being executed.

(b) Specific Computer Time

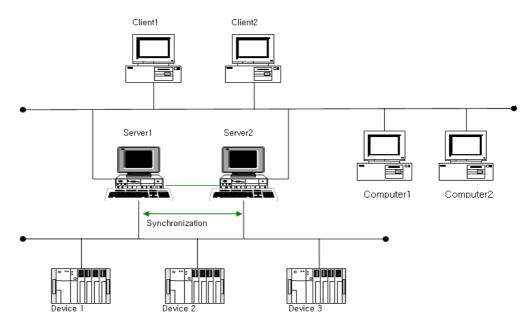
1) If all of the time synchronization fails, it is carried out with the time of the local system.

(c) Notice

If time synchronization is carried out with local system due to the failure in time synchronization, time synchronization is attempted with the very first setting information.

(3) Server/Client-Redundancy System

A time synchronization way for projects that form a server and client relation and server redundancy



(a) Connection HMI Time

- 1) Server(Running)
- Server 1 brings the information on whether it is in running or standby when it runs first and also brings the information on its partner-Server 2.
- If Server 1 is running, Server 1 uses its local time and record s this information on the log.
- If Server 1 is in standby, Server 1 synchronizes its time with that of Server2,
- 2) Server (If redundancy is abstained)
- If Server 1 is running and Server 2 is in standby and Server 2's running is abstained by Server 1, Serve 2 uses local time and Server1 obtains time information from Server 2 to time synchronize after receiving information on the change in the system's running status
- 3) Client:
- The client synchronizes time between the project under executing and the currently connected Server 1.
- If Server 1 is abstained to Server 2, redundancy system, due to any failure in the network or the system, the client that receives this information changes automatically from Server 1 to Server 2 for time synchronization and an event for the changed information on synchronization occurs.

(b) Specific Computer Time

1) Server/Client

- It is synchronized with the time of Computer 1 connected with the network.

-

- If time synchronization fails in Computer 1, it is carried out with the time of Computer 2 after an event occurs (if set up).
- If time synchronization fails both in Computer 1 and Computer 2, it is carried out with the time of local system.

(c) Notice

If time synchronization is carried out with local system due to the failure in time synchronization, time synchronization is attempted with the very first setting information.

This software is to collect data from a device and transfer to the upper process.

The user can add or set up Channel and Station in Tag Management.

7.1 **Channel Device Settings**

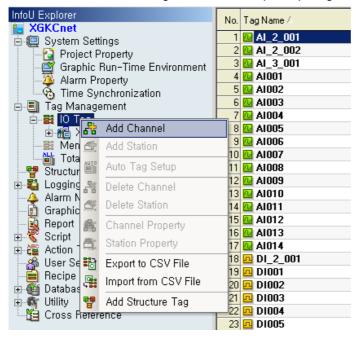
The user can add, delete or modify Channel and Station through pop-up menu if each device is displayed in a tree structure. This section briefly explains how to edit each item.

7.1.1 Channel

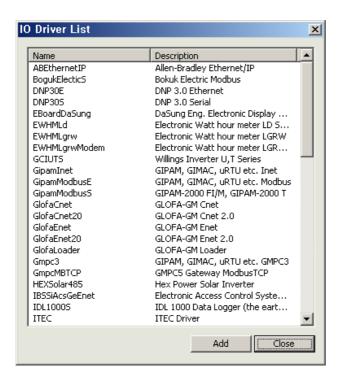
Channel refers to a channel to be communicated and selected among supporting I/O Drivers. Also, it basically provides a "memory" type of channels to classify input/out tags and memory tags in InfoU.

Setting

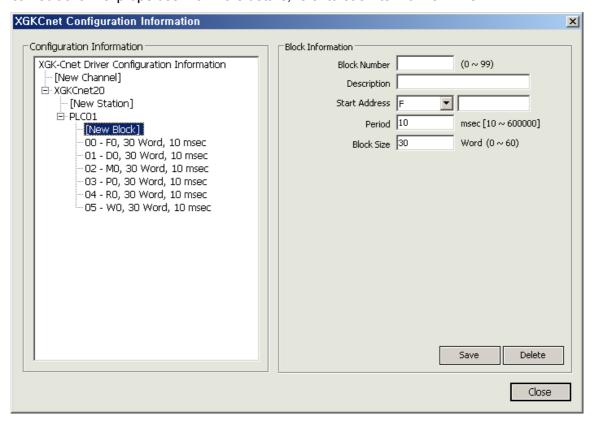
To select a channel, click the right mouse on the input/output tag list and select "Add Channel" menu.



If any additional menu is selected, a list of available I/O drivers is displayed.

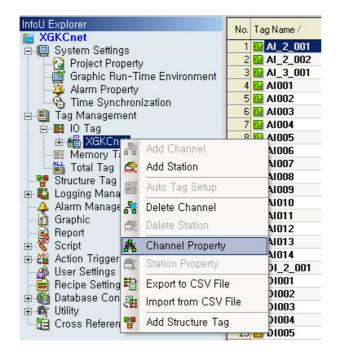


Select an item the user wants to add and click "Add" menu to show a screen in which the user can edit channel properties. For more details, refer to each item of I/O Driver.



Edit

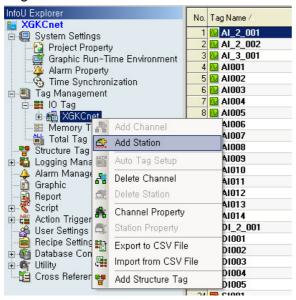
To edit Channel, click the right mouse on a relevant channel to select a menu.



7.1.2 Station

A station indicates a device such as PLC or inverter that belongs to the upper channel. Menu is similar to that of Channel. For more information on setting, refer to each item of I/O Driver.

Settings



7.1.3 Delete Channel

A channel can be deleted only after deleting all of the tags and stations that belong to the lower level. This also applies to stations. That is, deleting is possible only if there is no item in the lower level. If the user wants to delete a channel, delete Tag -> Station -> Channel in order.

7.2 **Driver List**

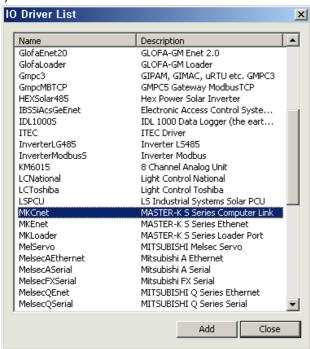
The following I/O drivers are supported (2009. May, it is scheduled to update this list).

Number	IO Driver Name	Description	Protocol	Manufacture	
1	GlofaEnet	GLOFA-G	TCP/UDP	LS Industrial Systems	
2	GlofaCnet	GLOFA-G	RS232/422/485	LS Industrial Systems	
3	XGKEnet	XGK Enet	TCP/UDP	LS Industrial Systems	
4	XGKCnet	XGK Cnet	RS232/422/485	LS Industrial Systems	
5	MKCnet	MASTER-K S Series Computer Link	RS232/422/485	LS Industrial Systems	
6	MKLoader	MASTER-K S Series Loader Port	RS232	LS Industrial Systems	
7	PMU	PMU Series	RS232	LS Industrial Systems	
8	GipamInet	GIPAM, GIMAC, uRTU etc. Inet	INET	LS Industrial Systems	
9	Gmpc3	GIPAM, GIMAC, uRTU etc. GMPC3	RS232	LS Industrial Systems	
10	DNP30E	DNP 3.0 Ethernet	TCP/UDP	DNP	
11	GipamModbusS	GIPAM-2000 FI/M, GIPAM-2000 T	Modbus(Serial)	LS Industrial Systems	
12	GipamModbusE	GIPAM, GIMAC, uRTU etc. Modbus	Modbus(Ethernet)	LS Industrial Systems	
13	InverterLG485	Inverter LS485	LG485	LS Industrial Systems	
14	InverterModbusS	Inverter Modbu	Modbus(Serial	LS Industrial Systems	
15	OPCClient	OPC Client	OP	OPC	
16	ITEC	ITEC Drivert	ITEC	LS Industrial Systems	
17	HEXSolar485t	Hex Power Solar Invertert	RS422/48	HEX POWE	
18	EWHMLgrw	Electronic Watt hour meter LGRW	RS422/48	LS Industrial Systems	
19	EBoardDaSung	DaSung Eng. Electronic Display Boards	RS422/48	DaSun	
20	EWHMLd	Electronic Watt hour meter LD Series	RS422/485	LS Industrial Systems	
21	WDC3000S	WDC3000 Serial	RS232	LS Industrial Systems	
22	WDC3000E	WDC3000 Ethernet	TCP/UD	LS Industrial Systems	
23	EWHMLgrwModem	Electronic Watt hour meter LGRW Modem	RS422/48	LS Industrial Systems	
24	P2EJson	P2E Gateway	TCP/UD	NaMoo I&C	
25	MelsecQEnet	MITSUBISHI Q Series Ethernet	TCP/UD	MITSUBISHI	
26	MelsecQSerial	MITSUBISHI Q Series Serial	RS232/422/485	MITSUBISHI	
27	RFID1356S	LS Industrial Systems RFID 13.56MHz	RS232/422/485	LS Industrial Systems	
28	GlofaLoader	GLOFA-GM Loader	RS232/422/485	LS Industrial Systems	
29	P3000DCS	P3000 DCS	TCP/UD	LS Industrial System	

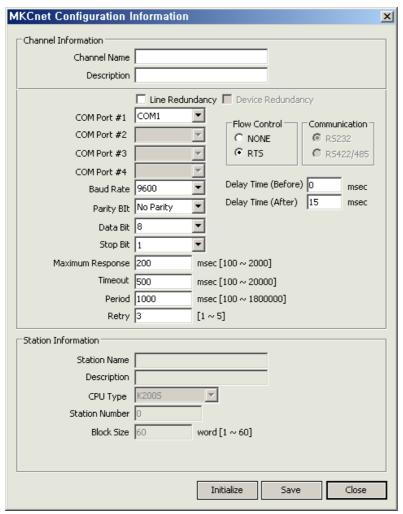
Number	IO Driver Name	Description	Protocol	Manufacture
30	LCToshiba	Light Control Toshiba	RS232/422/485	Toshib
31	LCNational	Light Control National	RS232/422/485	National
32	MKEnet	MASTER-K S Series Ethenet	TCP/UD	LS Industrial Systems
33	GCIUTS	Willings Inverter U,T Series	RS232/422/485	Willingsl
34	PVCB12CH	Junction Box RTU 12 Channel	RS232/42	Willingsl
35	PVCB16CH	Junction Box RTU 16 Channel	RS232/422/485	Mobitronl
36	IBSSiAcsGeEnet	Electronic Access Control Systems GE Secure Perfect v6.1.1 Ethernet	TCP/UD	SecureTechl
37	KM6015	8 Channel Analog Unit	RS422/485	Kisan System
38	LSPCU	LS Industrial Systems Solar PCU	RS422/485	LS Industrial Systems
39	GlofaEnet20	GLOFA-GM Enet 2.0	TCP/UDP	LS Industrial Systems
40	BogukElecticS	Bokuk Electric Modbus	RS232/422/485	Bokuk Electric
41	GlofaCnet20	GLOFA-GM Cnet 2.0	RS232/422/485	LS Industrial Systems
42	XGKEnet20	XGK Enet 2.0	TCP/UDP	LS Industrial Systems
43	XGKCnet20	XGK Cnet 2.0	RS232/422/485	LS Industrial Systems
44	XGIEnet	XGI Enet	TCP/UDP	LS Industrial Systems
45	XGICnet	XGI Cnet	RS232/422/485	LS Industrial Systems
46	ModbusTCP	ModbusTCP	TCP/UDP	Modbus
47	ModbusRTU	ModbusRTU	RS232/422/485	Modbus
48	OmronS	Omron CJ/CS Serial (Host Link)	RS232/422/485	Omron
49	OmronE	Omron Ethernet (FINS)	TCP/UDP	Omron
50	NaisS	Panasonic Nais (FP Series) Serial	RS232/422/485	Panasonic
51	MelsecFXSerial	Mitsubishi FX Serial	RS232/422/485	MITSUBISHI
52	ABEthernetIP	Allen-Bradley Ethemet/IP	TCP/UDP	Allen-Bradley
53	XGKLoader	XGK Loader	RS232	LS Industrial Systems
54	XGILoader	XGI Loader	RS232	LS Industrial Systems
55	IDL1000S	IDL 1000 Data Logger (the earth of the Heat Monitoring)	RS232/422/485	KM Data
56	MelsecASerial	Mitsubishi A Serial	RS232/422/485	MITSUBISHI
57	MelsecAEthernet	Mitsubishi A Ethernet	TCP/UDP	MITSUBISHI
58	GmpcMBTCP	GMPC5 Gateway ModbusTCP	TCP/UDP	LS Industrial Systems

7.3 **MASTER-K S Series Computer Link**

(1) Add Channel

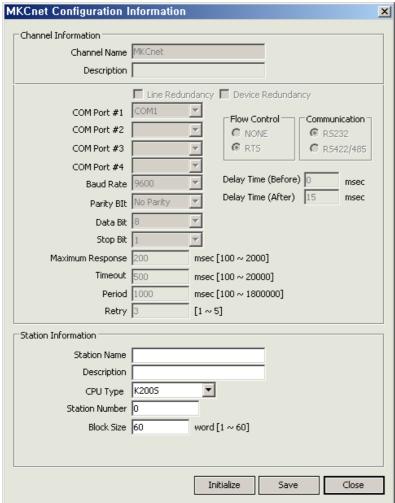


Select "MKCnet" from the I/O Driver and press "Add".



- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) Line Redundancy: Check $\sqrt{}$ in the box to use Line Redundancy.
- (d) COM Port #1: Select a communication port.
- (e) COM Port #2: Select a communication port.
- (f) COM Port #3: Select a communication port.
- (g) COM Port #4: Select a communication port.
- (h) Baud Rate: Select a communication speed.
- (i) Parity Bit: Select a parity bit.
- (j) Stop Bit: Select a stop bit.
- (k) Maximum Response: It refers to a certain time period that will take to receive a response after requesting data.
- (I) Time Out: It refers to a certain time period for which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (m) Period: It refers to an interval to request data.
- (n) Retry: Set up the number of times to retry when communication fails.
- (o) Flow Control: Select either None or RTS
- (p) Delay Time (Before): Set up a communication delay time. It can be set up only when using RTS control.
- (q) Delay Time (After): Set up a communication delay time. It can be set up only when using RTS control.

(2) Add Station.



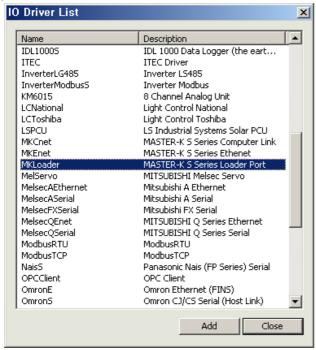
- (a) Station Name: Input a station name.
- (b) Description: Input some information on the station.
- (c) CPU Type: Select a CPU type.
- (d) Station Number: Input the number of the station.
- (e) Block Size: Input a block size of PLC communication.

(3) I/O Address

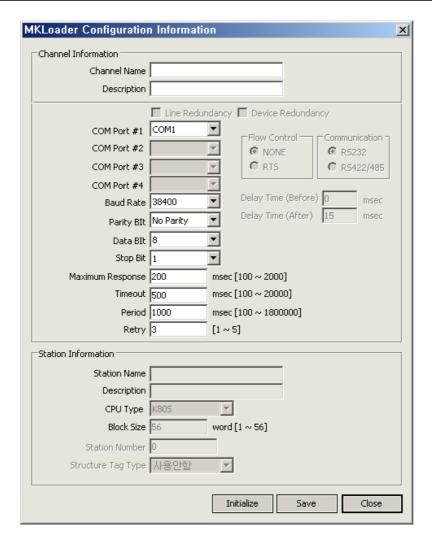
- (a) Devices to be supported
 - : P, M, K, F, L, D (Device T and C are not supported.)
- (b) I/O Address Type
 - : P000 (Analog Tag: The last three digits are reserved to show the word area), P00F (Digital tag: The last one digit shows the bit area.) The remaining area is for the word area.

7.4 **MASTER-K S Series Loader Port**

(1) Add Channel

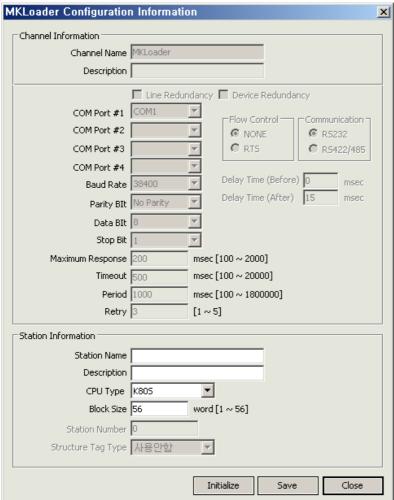


Select "MKLoade" from the I/O Driver list and press "Add."



- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) Line Redundancy: Check $\sqrt{}$ in the box to use Line Redundancy.
- (d) COM Port #1: Select a communication port.
- (e) COM Port #2: Select a communication port.
- (f) COM Port #3: Select a communication port.
- (g) COM Port #4: Select a communication port.
- (h) Baud Rate: Select a communication speed.
- (i) Parity Bit: Select a parity bit.
- (j) Stop Bit: Select a stop bit.
- (k) Maximum Response: It refers to a certain time period that will take to receive a response after requesting data.
- (I) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (m) Period: It refers to an interval to request data.
- (n) Retry: Set up the number of times to retry when communication fails.

(2) Add Station



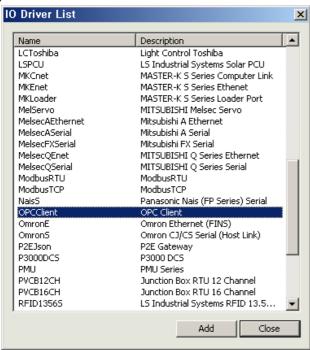
- (a) Station Name: Input a station name.
- (b) Description: Input some information on the station.
- (c) CPU Type: Select a CPU type.
- (d) Block Size: Input a block size of PLC communication.

(3) I/O Address

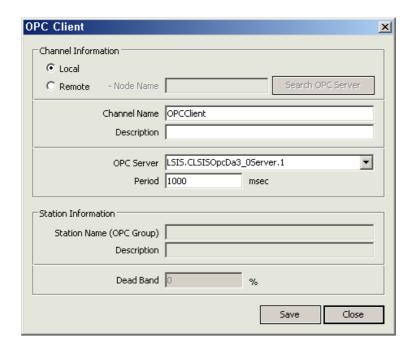
- (a) Devices to be supported
 - : P, M, K, F, L, D (Device T and C are not supported.)
- (b) I/O Address Type
 - : P000 (Analog Tag: The last three digits are reserved to show the word area), P00F (Digital tag: The last one digit shows the bit area.) The remaining area is for the word area.

7.5 OPC Client

(1) Add Channel



Select"OPCClient"from the I/O driver list and press "Add."

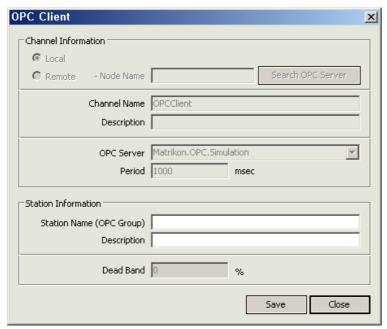


- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) OPC Server: Select an OPC server name. If selecting a local radio button, OPC server lists that are installed on the local

PC will be displayed in the combo box list. If the user presses "Remote OPC server inquire" button after selecting a remote radio button and inputting a node name, the OPC server lists that are installed on the remote PC will be displayed in the combo box list.

- (n) Period: Select an interval to update data.
- (e) Local/Remote: Identify whether the OPC server to be used is either on the local PC or on the remote PC.
- (f) Node Name: Input the remote PC's name. (It is available only when "Remote" is selected).
- (g) Search OPC Server: It searches the OPC server lists installed on the remote PC. The search results will appear in the combo box list of "OPC server name". (It shall be enabled only when "Remote" is selected).

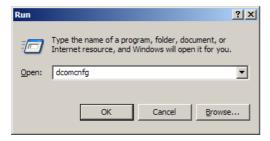
(2) Add Station



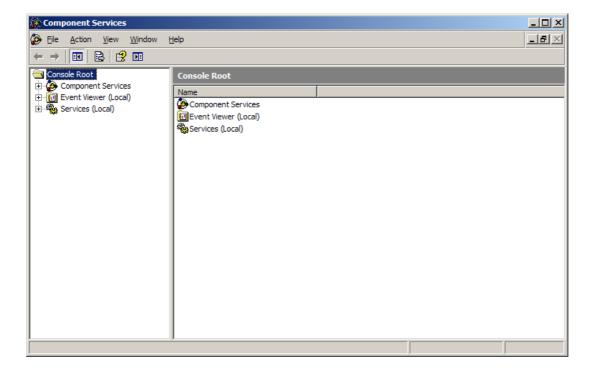
- (a) Station Name (OPC Group): Input a station name.
- (b) Description: Input some information on the station.

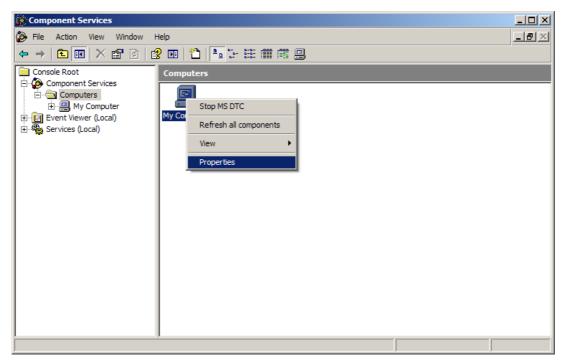
(3) DCOM Settings

- (a) Default
- The user needs to set up DOM to use the OPC servers installed on the remote PC.
- DCOM is set up as the following procedure and the local PC and the remote PC are also applied the same settings as DCOM.
- The following description is about DCOM settings in the window XP and for the DCOM settings in other OS, any similar type is used.
- The same account shall be registered for both of the local PC and remote PC and the same password shall be used for both of them.
- The local PC and remote PC shall be accessed with the same account.
- The firewall in the remote PC shall be cancelled to log in to the system.
- To use the OPC servers installed in the remote PC, basic OPC files shall be installed in the local PC. (Install the OPC Core Components Redistributable program first. Visit www.opcfoundation.org)
- (b) Component Service Settings

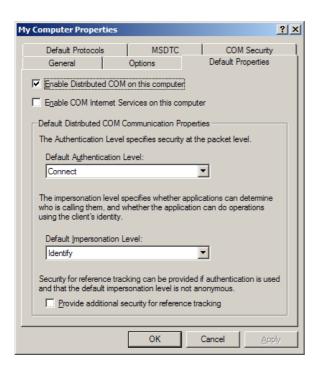


- Input "dcomcnfg" in the window and press OK.
- The following "Component Services" window will appear.

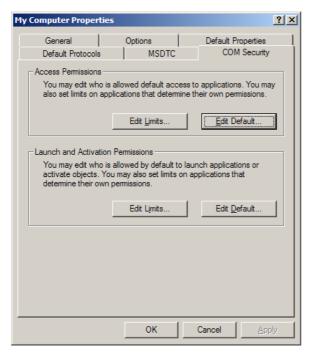




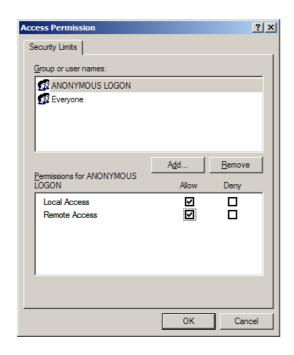
- Find "My Computer" and execute "Properties" menu.
- The following "My Computer Properties" window will appear.



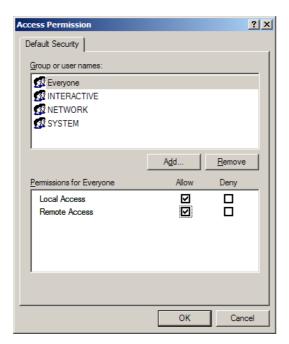
- Select "Default Properties" tap.
- Check √ in the checkbox of "Enable Distributed COM on this computer".
- Select "Connect" for "Default Authentication Level" item.
- Select "ID" for "Default Impersonation Level" item.



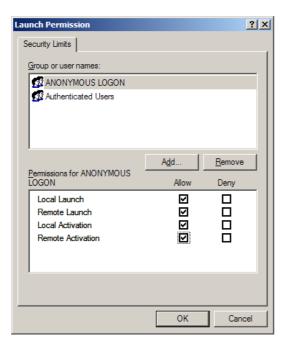
- Select"COM Security" tap.
- To set up "Access Permissions", press "Edit Limits" and "Edit Default" to set up as follow:
 - Edit Limits; Check √ in both of the boxes under "Allow" for "Local Access" and "Remote Access" to permit ANONYMOUS LOGON and Authenticated Users.



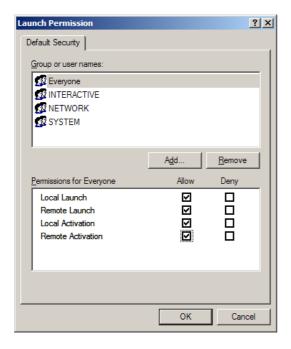
 Edit Default: - Edit Limits; Check ¡îin both of the boxes under "Allow" for "Local Access" and "Remote Access" to permit INTERACTIVE, NETWORK and SYSTEM users to log in.



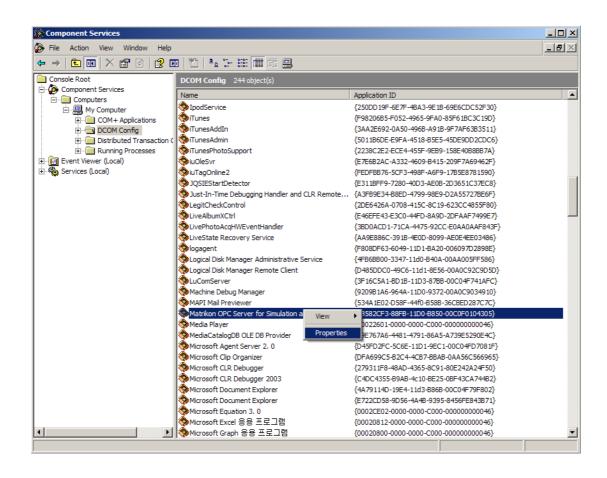
- To set up "authorization to launch and activate", press "Edit Default" and set up as follow:
 - Edit Limits: Check $\sqrt{\ }$ in all of the boxes under "Allow" for "Local Launch", "Remote Launch", "Local Activation" and "Remote Local Activation" to permit Authenticated Users to launch and activate.



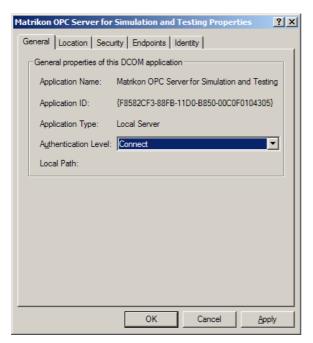
 Edit Default: Check √ in all of the boxes under" Allow" for "Local Launch", "Remote Launch", "Local Activation" and "Remote Local Activation" to permit INTERACTIVE, NETWORK and SYSTEM users to launch and activate.



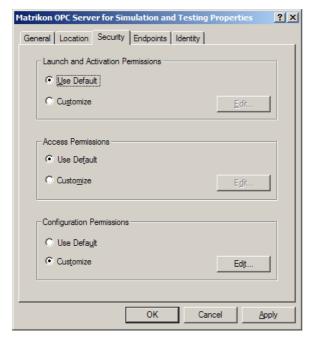
- In the next step, DOM settings are carried out on the OPC server to be used.
- The user needs to set up the OPC server he/she wants to use according to the following procedure:



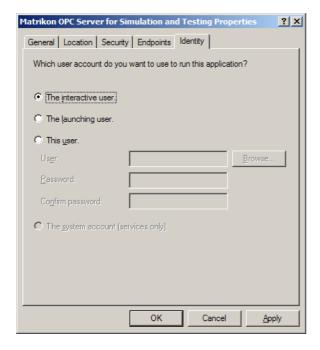
• Find the OPC server the user wants to use and execute "Properties" menu.



• Select "Connect" for "Authentication Level" from "General" tap.

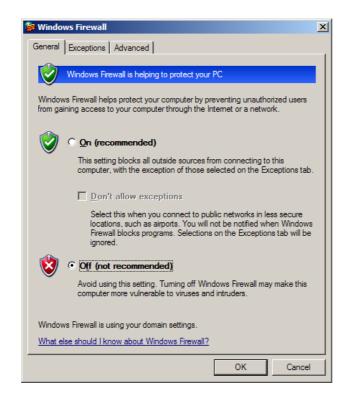


- Select "Security" tap.
- Select "Use Default" for "Launch and Activation Permissions."
- Select "Use Default" for "Access Permissions."

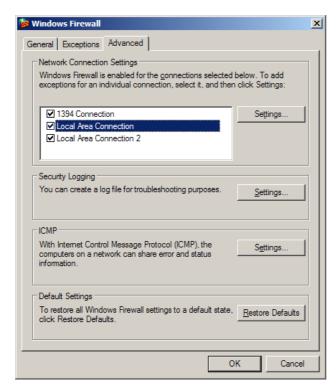


- Select "Identity" tap.
- Select "The Interactive User."

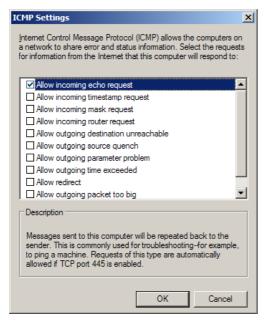
(C) Firewall Settings



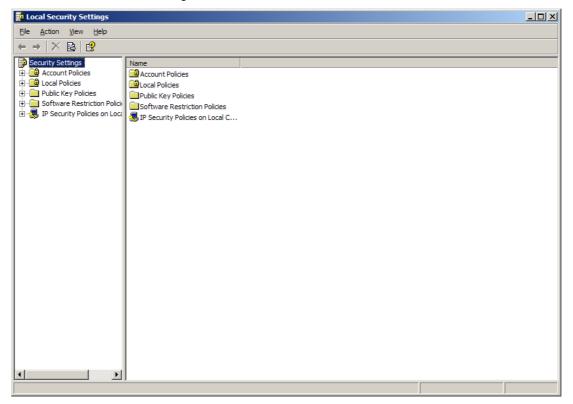
- The above Windows Firewall window is displayed.
- Select "Off (not recommended)".
- Select "Advanced" tap.



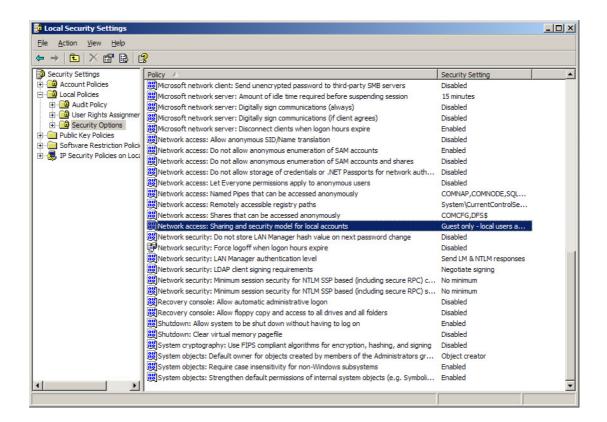
- Click "Settings" button for ICMP.

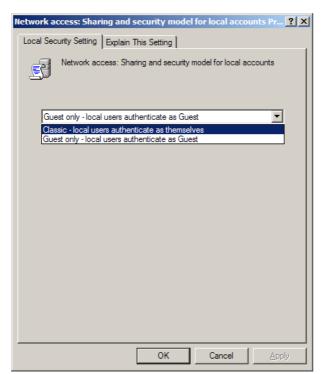


- Check "Allow incoming echo request".
- (d) Security Settings
- Execute Control Board → Management Tool → Local Policies.



■ Local Policies → Security Options → "Network Access: Execute "Sharing and security model for local accounts".

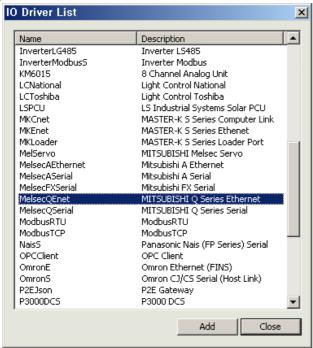




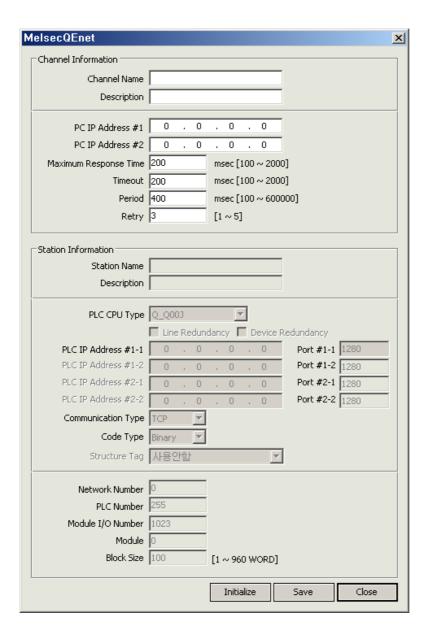
Select"Classic – Local users authenticate as themselves".

7.6 MelsecQEnet

(1) Add Channel

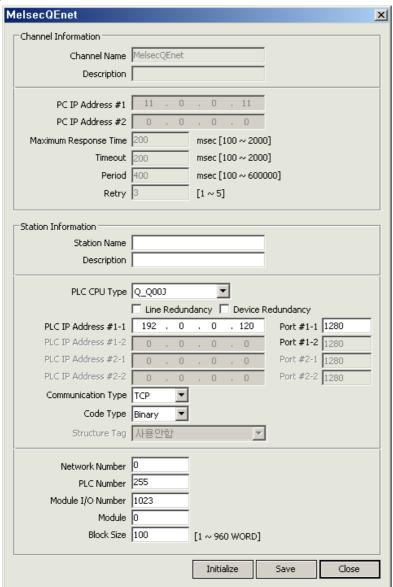


Select "MelsecQEnet" from the I/O Driver list and press "Add.".



- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) Serve IP: Input PC's IP Address.
- (d) Server IP Address #2: If Line Redundancy will be used, input the second IP Address to be used.
- (e) Maximum Response: It refers to a certain time period that will take to receive a response after requesting data.
- (f) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (g) Period: It refers to an interval to request data.
- (h) Retry: Set up the number of times to retry when communication fails.

(2) Add Station



- (a) Station Name: Input a station name.
- (b) Description: Input some information on the station.
- (c) PLC CPU Type: Select a PLC CPU type.
- (d) Line Redundancy: Check √in the box to use Line Redundancy.
- (e) Device Redundancy: Check √in the box to use Device Redundancy.
- (f) PLC IP Address #1-1: Input PLC's IP Address.
- (g) PLC IP Address #1-2: Input PLC's IP Address. Input the address when using Device Redundancy.
- (h) PLC IP Address #2-1: Input PLC's IP Address. Input the address when using Device Redundancy.
- (i) PLC IP Address #2-2: Input PLC's IP Address. Input the address using Line Redundancy along with Device Redundancy.
- (j) Port #1-1: Input the number of the corresponding port to the IP Address.
- (k) Port #1-2: Input the number of the corresponding port to the IP Address
- (I) Port #2-1: Input the number of the corresponding port to the IP Address.
- (m) Port #2-2: Input the number of the corresponding port to the IP Address.
- (n) Communication Type: Select either TCP or UDP.

- (o) Protocol Type: Select either ASCII or Binary.
- (p) Network Number: Input the number of the network.
- (q) PLC Number: Input the number of PLC.
- (r) Module I/O Number: Input the number of I/O.
- (s) Module Prefix Number: Input the prefix number of the module.
- (t) Block Size: Input a block size of PLC communication.

(3) I/O Address

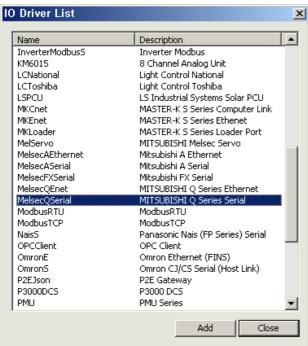
(a) I/O Address Map

(a) I/O Address Map							
			Address Range				
Device Type	Bit	Word	Q02(H), Q06H, Q12H, Q25H, Q12PH, Q25PH, Q2A, Q2A-S1, Q2AS, Q2AS- S1, Q2ASH, Q2ASH-1, Q3A, Q4A, Q4AR	Q00J, Q00, Q01	Decimal	Hexdecimal	Example(InfoU Address)
SM	•	-	000000~00204 7	000000~00102 3	•	-	SM0 , SM10, SM197
SD	-	•	000000~00204 7	000000~00102 3	•	-	SD1, SD2047
Х	•	-	000000~001FF F	000000~0007F F	-	•	X0~XF, X10~X1F, X1FFF
Υ	•	-	000000~001FF F	000000~0007F F	-	•	Y0~YF, Y10~Y1F, YFFF
М	•	-	000000~00819 1	000000~00819 1	•	-	M0, M1F
L	•	-	000000~00819 1	000000~00204 7	•	-	L0~L11, L15, L100
F	•	-	000000~00204 7	000000~00102 3	•	-	F0~F17, F9, F2000
V	•	-	000000~00204 7	000000~00102 3	•	-	V1000, V2047
В	•	-	000000~001FF F	000000~0007F F	-	•	B1F, B1000, B1FFF
D	-	•	000000~01228 7	000000~01113 5	•	-	D0, D1000, D10000, D12287
W	-	•	000000~001FF F	000000~0007F F	-	•	W0, WF, W1F, W1FFF
TS	•	-			•	-	TS0, TS12, TS1000, TS2000
тс	•	-	000000~00204	000000~00051	•	-	TC1, TC17, TC200, TC2047
TN	-	•	7	1	•	-	TN0, TN2047
SS	•	-			•	-	SS0, SS2047
SC	•	-			•	-	SC0~SS2047
SN	-	•			•	-	SN0~SN2047
CS	•	-	000000~00102	000000~00051	•	-	CS0~CS1023
CC	•	-	3	1	•	-	CC0~CC1023
CN	-	•		<u> </u>	•	-	CN0~CC1023

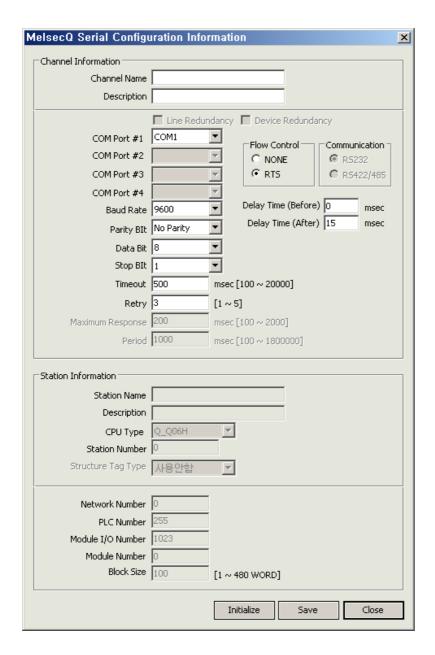
SB	•	-	000000~0007F F	000000~0003F F	-	•	SB0~SB7FF
SW	-	•	000000~0007F F	000000~0003F F	1	•	SW0~SW7FF
s	•	-	000000~00819	000000~00204 7	•	-	S0~S8191 (Q00J,Q00,Q01 Not accessible)
DX	•	-	000000~001FF F	000000~0007F F	1	•	DX0~DX1FFF
DY	•	-	000000~001FF F	000000~0007F F	1	•	DY0~DY1FFF
Z	1	•	000000~00001 5	000000~00000 9	•	1	Z0~Z15
R	-	•	000000~03276 7	000000~03276 7	•	-	R0~R32767
ZR	-	•	000000~0FE7F F	000000~00FF FF	-	•	ZR0~ZRFE7FF

7.7 MdelsecQSerial

(1) Add Channel



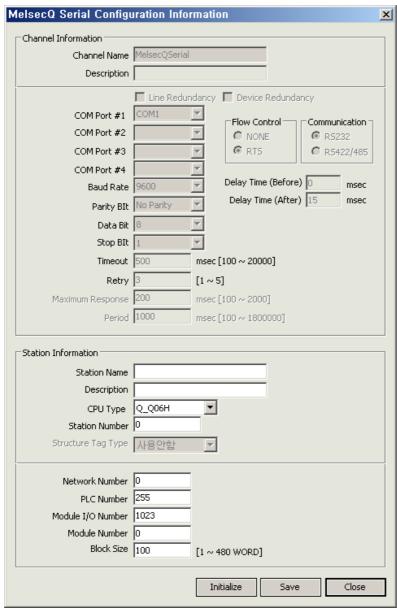
Select"MelsecQSerial"from the I/O driver list and press "Add."



- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) Line Redundancy: Check $\sqrt{}$ in the box to use Line Redundancy.
- (d) COM Port #1: Select a communication port.
- (e) COM Port #2: Select a communication port.
- (f) COM Port #3: Select a communication port.
- (g) COM Port #4: Select a communication port.
- (h) Baud Rate: Select a communication speed.
- (i) Parity Bit: Select a parity bit.
- (j) Data Bit: Select a data bit.
- (k) Stop Bit: Select a stop bit.
- (I) Maximum Response: It refers to a certain time period that will take to receive a response after requesting data.

- (m) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (n) Period: It refers to an interval to request data.
- (o) Retry: Set up the number of times to retry when communication fails.
- (p) Flow Control: Select either None or RTS
- (q) Delay Time (Before): Set up a communication delay time. It can be set up only when using RTS control.
- (r) Delay Time (After): Set up a communication delay time. It can be set up only when using RTS control.

(2) Add Station



- (a) Station Name: Input a station name.
- (b) Description: Input some information on the station.
- (c) CPU Type: Select a CPU type.
- (d) Station Number: Input the station number.
- (e) Network Number: Input the network number.

- (f) PLC Number: Input the PLC number.
- (g) Module I/O Number: Input the module I/O number.
- (h) Module Prefix Number: Input the prefix number of the module.
- (i) Block Size: Input a block size of PLC communication.

(3) I/O Address

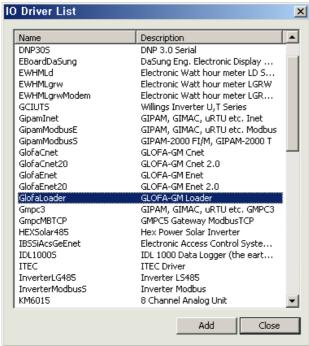
(a) I/O Address Map

a) I/O Address Map										
			Address Range							
Device Type	Bit	Word	Q02(H), Q06H, Q12H, Q25H, Q12PH, Q25PH, Q2A, Q2A-S1, Q2AS, Q2AS- S1, Q2ASH, Q2ASH-1, Q3A, Q4A, Q4AR	Q00J, Q00, Q01	Decimal	Hexdecimal	Example(InfoU Address)			
SM	•		000000~00204 7	000000~001023	•		SM0 , SM10, SM197			
SD		•	000000~00204 7	000000~001023	•		SD1, SD2047			
Х	•		000000~001FF F	000000~0007FF		•	X0~XF, X10~X1F, X1FFF			
Υ	•		000000~001FF F	000000~0007FF		•	Y0~YF, Y10~Y1F, YFFF			
M	•		000000~00819 1	000000~008191	•		M0, M1F			
L	•		000000~00819 1	000000~002047	•		L0~L11, L15, L100			
F	•		000000~00204 7	000000~001023	•		F0~F17, F9, F2000			
V	•		000000~00204 7	000000~001023	•		V1000, V2047			
В	•		000000~001FF F	000000~0007FF		•	B1F, B1000, B1FFF			
D		•	000000~01228 7	000000~011135	•		D0, D1000, D10000, D12287			
W		•	000000~001FF F	000000~0007FF		•	W0, WF, W1F, W1FFF			
TS	•				•		TS0, TS12, TS1000, TS2000			
тс	•		000000~00204	000000 000511	•		TC1, TC17, TC200, TC2047			
TN		•	7	000000~000511	•		TN0, TN2047			
SS	•				•		SS0, SS2047			
SC	•				•		SC0~SS2047			
SN		•			•		SN0~SN2047			
CS	•	•			•		CS0~CS1023			
CC	•		000000~00102	000000~000511	•		CC0~CC1023			
CN		•	3	200000 000011	•		CN0~CC1023			
SB	•		000000~0007F	000000~0003FF		•	SB0~SB7FF			
OD	_		000000 -0007 F	JUUUUU JUUUSEE		•	וויםטי טפט			

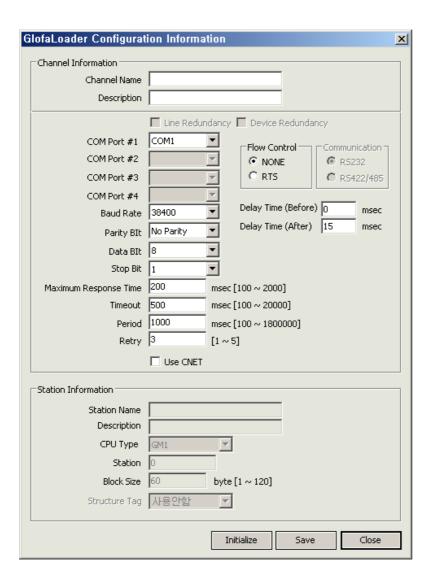
			F				
SW		•	000000~0007F F	000000~0003FF		•	SW0~SW7FF
S	•		000000~00819	000000~002047	•		S0~S8191 (Q00J,Q00,Q01 Not accessible)
DX	•		000000~001FF F	000000~0007FF		•	DX0~DX1FFF
DY	•		000000~001FF F	000000~0007FF		•	DY0~DY1FFF
Z		•	000000~00001 5	000000~000009	•		Z0~Z15
R		•	000000~03276 7	000000~032767	•		R0~R32767
ZR		•	000000~0FE7F F	000000~00FFFF		•	ZR0~ZRFE7FF

7.8 GLOFA Loader

(1) Add Channel

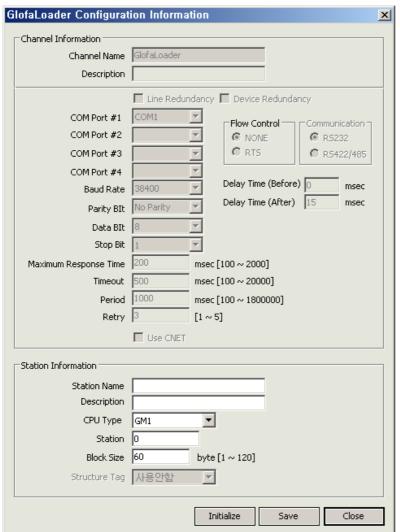


• Select "GLOFALoader" from the I/O driver list and press "Add."



- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) COM Port #1: Select a communication port.
- (d) Baud Rate: Select a communication speed.
- (e) Parity Bit: Select a parity bit.
- (f) Data Bit: Select a data bit.
- (g) Stop Bit: Select a stop bit.
- (h) Maximum Response: It refers to a certain time period that will take to receive a response after requesting data.
- (i) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (j) Period: It refers to an interval to request data.
- (k) Retry: Set up the number of times to retry when communication fails.

(2) Add Station



- (a) Station Name: Input a station name.
- (b) Description: Input some information on the station.
- (c) CPU Type: Select a CPU type.
- (d) Station Number: Input the station number.
- (e) Block Size: Input a block size of PLC communication.

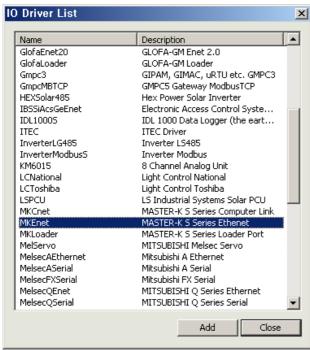
(3) I/O Address

I/O Address Type

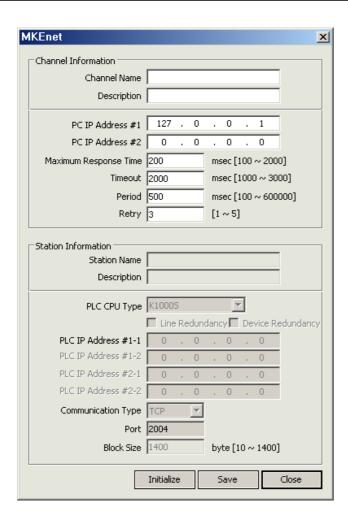
- : %MX0, %MB0, %MW0, %MD0, %ML0
- : %MB0.0, %MW0.0, %MD0.0, %ML0.0

7.9 **MKEnet**

(1) Add Channel

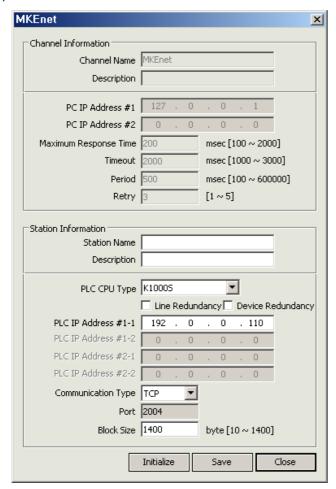


Select"MKEnet"from the I/O driver list and press "Add."



- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) Serve IP: Input PC's IP Address.
- (d) Server IP Address #2: If Line Redundancy will be used, input the second IP Address to be used.
- (e) Maximum Response: It refers to a certain time period that will take to receive a response after requesting data.
- (f) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (g) Period: It refers to an interval to request data.
- (h) Retry: Set up the number of times to retry when communication fails.

(2) Add Station



- (a) Station Name: Input a station name.
- (b) Description: Input some information on the station.
- (c) PLC CPU Type: Select a PLC CPU type.
- (d) Line Redundancy: Check √in the box to use Line Redundancy.
- (e) Device Redundancy: Check √in the box to use Device Redundancy.
- (f) PLC IP Address #1-1: Input PLC's IP Address.
- (g) PLC IP Address #1-2: Input PLC's IP Address. Input the address when using Device Redundancy.
- (h) PLC IP Address #2-1: Input PLC's IP Address. Input the address when using Device Redundancy.
- (i) PLC IP Address #2-2: Input PLC's IP Address. Input the address when using Line Redundancy along with Device Redundancy.
- (n) Communication Type: Select either TCP or UDP.
- (k) Port: The port number is automatically inputted according to the selected communication type.
- (I) Block Size: Input a block size of PLC communication.

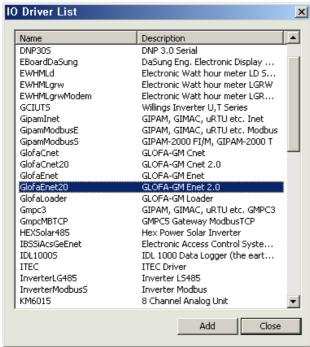
(3) I/O Address

- (a) Devices to be supported
 - : P, M, K, F, L, D (Device T and C are not supported.)
- (b) I/O Address Type

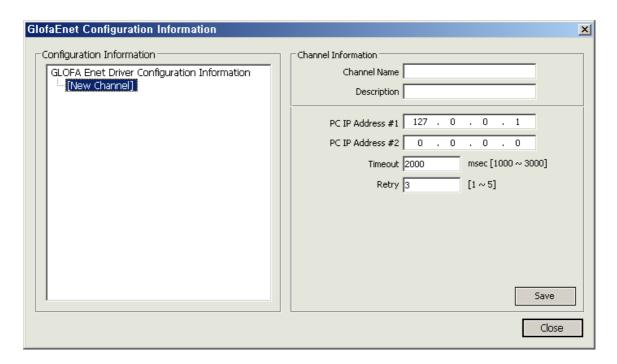
: P000 (Analog Tag: The last three digits are reserved to show the word area), P00F (Digital tag: The last one digit shows the bit area.) The remaining area is for the word area.

7.10 GlofaEnet20

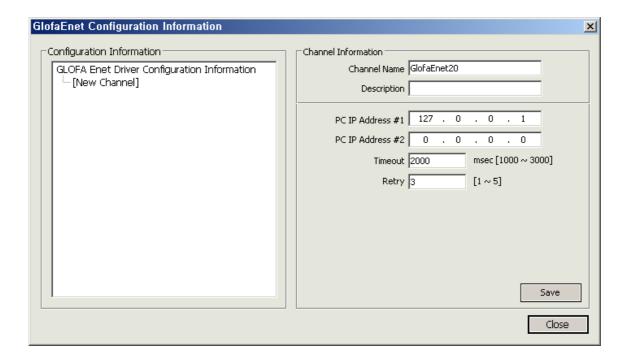
(1) Add Channel



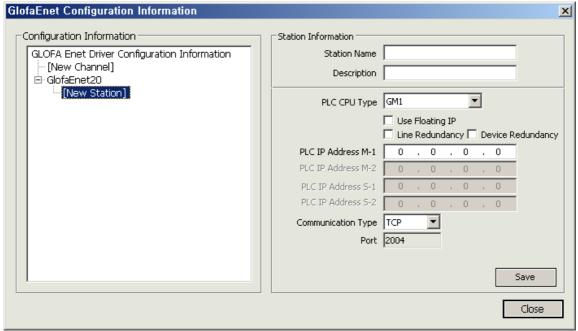
• Select "GlofaEnet20" from the I/O driver list and press "Add."



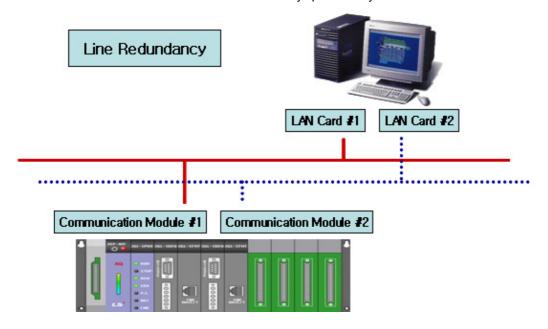
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) Serve IP: Input PC's IP Address.
- (d) Server IP Address #2: If Line Redundancy will be used, input the second IP Address to be used.
- (e) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (f) Retry: Set up the number of times to retry when communication fails.
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.



(2) Add Station

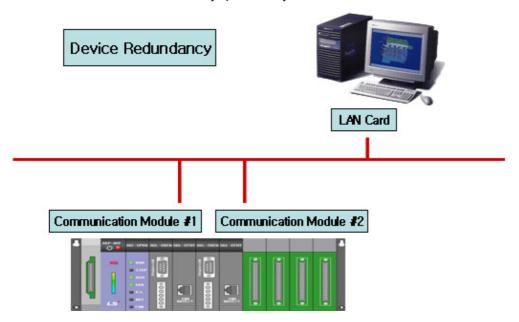


- (a) Selec [New Station] from "Configuration Information" tree.
- (b) Station Name: Input a station name.
- (c) Description: Input some information on the station.
- (d) PLC CPU Type: Select a PLC CPU type.
- (e) Line Redundancy: Check √ in the box to use Line Redundancy. It is used when the network is intended to be separated and communicated as seen in the figure below by installing two LAN cards on the computer and two Enet communication modules on PLC. It is a redundancy option for any failure in the network line.

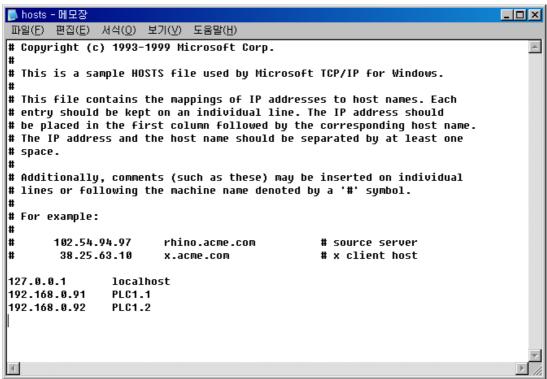


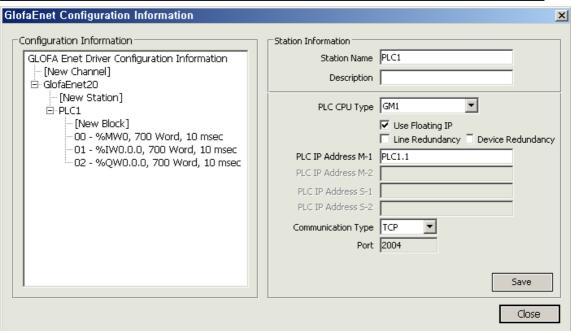
(f) Device Redundancy: Check √ in the box to use Device Redundancy. It is used when the communication module is

separated as seen in the figure below by installing one LAN card on the computer and two Enet communication modules on PLC. It is a redundancy option for any failure in the PLC communication module.

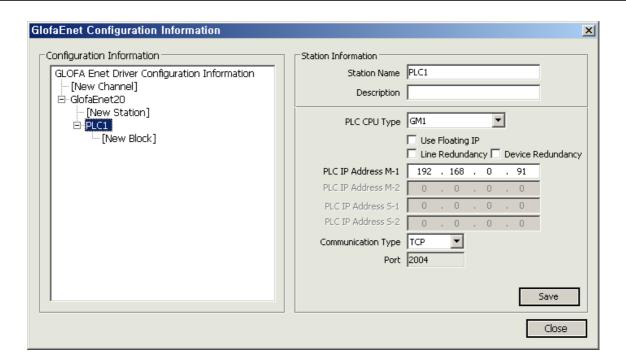


- (g) PLC IP Address #1-1: Input PLC's IP Address.
- (h) PLC IP Address #1-2: Input PLC's IP Address. Input the address when using Device Redundancy.
- (i) PLC IP Address #2-1: Input PLC's IP Address. Input the address when using Device Redundancy.
- (j) PLC IP Address #2-2: Input PLC's IP Address. Input the address when using Line Redundancy along with Device Redundancy.
- (k) Communication Type: Select either TCP or UDP.
- (I) Port: The port number is automatically inputted according to the selected communication type.
- (m) Flexible IP Support: Check the following if the user wants to use a flexible IP. A flexible IP uses hosts files to communicate and they are located in C:\WINDOWS\system32\drivers\etc. Once they are saved as follow, set up a flexible IP according to the inst

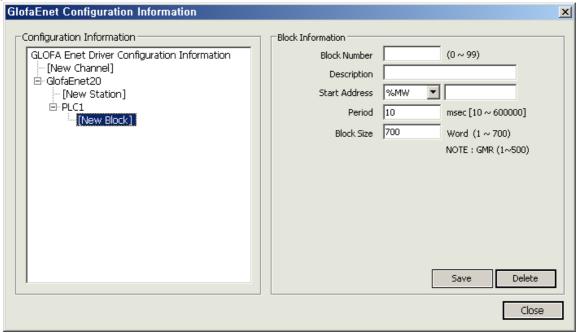




(n) Save: If 'Save' button is pressed, Station information will be saved and the saved information will add to the left "Configuration Information" tree.

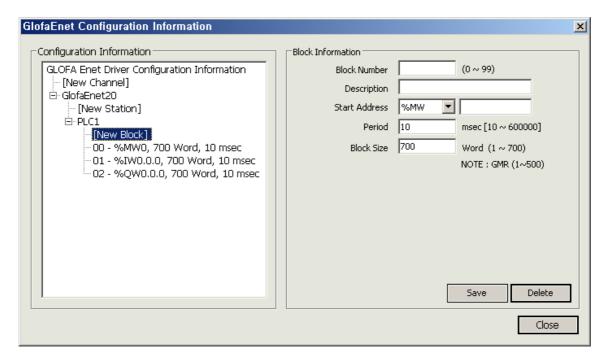


(3) Add Block



- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: Input the Block's Start Address. There are three kinds and each address is designated as the following ways respectively:
 - -. Right example: %MW0, %MW20, %IW0.0.0, %QW1.0.0
 - -. Wrong example: %MW0.0.0, %IW0, %QW0

- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Block Size: Input a block size of the relevant block (unit: Word (2 byte)).
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.

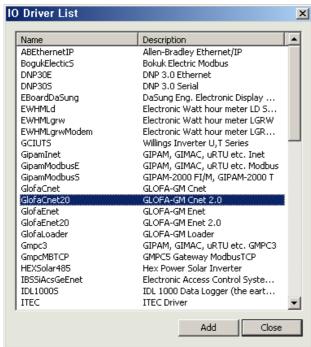


(4) I/O Address

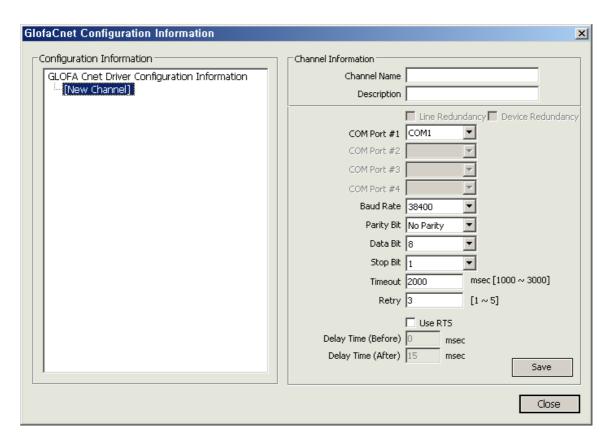
- (a) Type
 - -Analog: %MX0, %MB0, %MW0, %MD0, %ML0, %IW0.0.0, %QW0.0.0
 - -.Digital: %MB0.0, %MW0.0, %MD0.0, %ML0.0, %IX0.0.0, %QX0.0.0
- (b) Available devices
 - -. I, M, Q

7.11 GlofaCnet20

(1) Add Channel



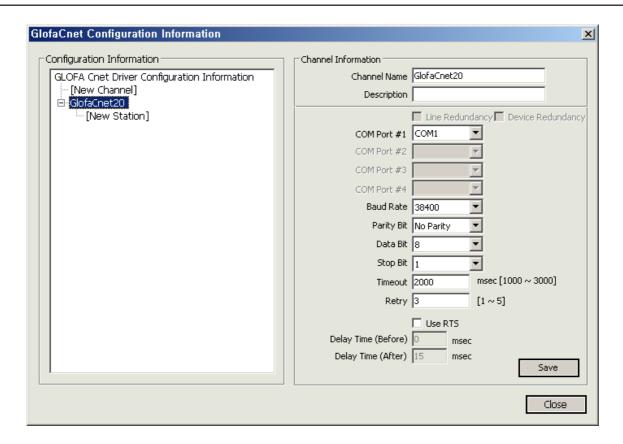
Select"GlofaCnet20"from the I/O driver list and press "Add."



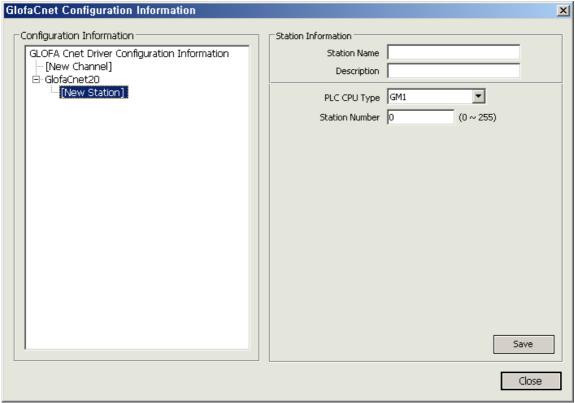
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) COM Port #1: Select a communication port.
- (d) Baud Rate: Select a communication speed.
- (e) Parity Bit: Select a parity bit.
- (f) Data Bit: Select a data bit.
- (g) Stop Bit: Select a stop bit.
- (h) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process.

The time period to be set will be a base to judge communication errors.

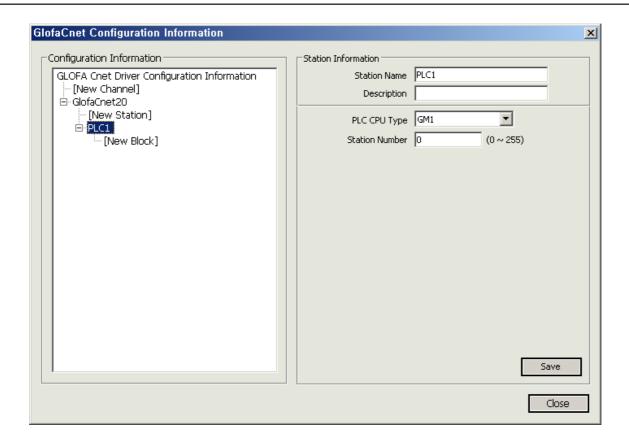
- (i) Retry: Set up the number of times to retry when communication fails.
- (j) Use RIS: Check $\sqrt{}$ in the box only when using Channel 485 and 422.
- (k) Delay Time (Before): Information used only in Channel 485 and 422. The delay time right before requesting Data to PLC.
- (I) Delay Time (After): Information used only in Channel 485 and 422. The delay time right after requesting Data to PLC.
- (m) Save: If 'Save' button is pressed, Channel information will be saved and the saved information will add to the left. "Configuration Information" tree.



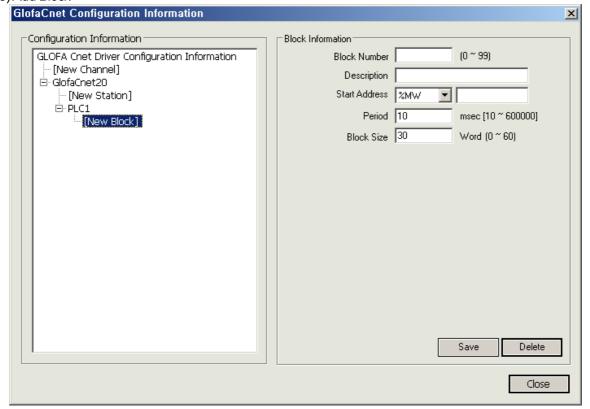
(2) Add Station



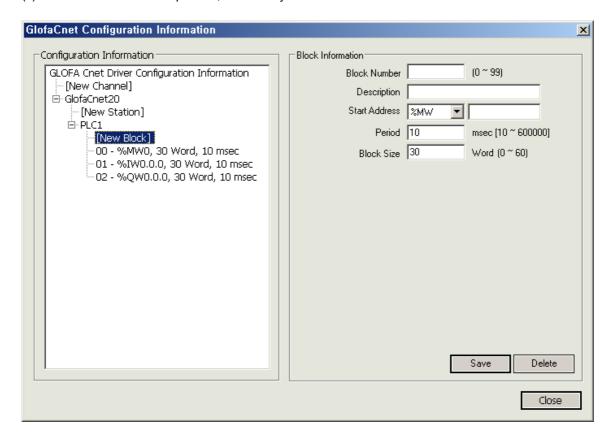
- (a) Select [New Station] from "Configuration Information" tree.
- (b) Station Name: Input a station name.
- (c) Description: Input some information on the station.
- (d) PLC CPU Type: Select a PLC CPU type.
- (e) Station Number: Input the prefix number of PLC Cnet Module.
- (f) Save: If 'Save' button is pressed, Station information will be saved and the saved information will add to the left "Configuration Information" tree.



(3) Add Block



- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: Input the Block's Start Address. There are three kinds and each address is designated as the following ways respectively:
 - -. Right example: %MW0, %MW20, %IW0.0.0, %QW1.0.0
 - -. Wrong example: %MW0.0.0, %IW0, %QW0
- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Block Size: Input a block size of the relevant block (unit: Word (2 byte)).
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.

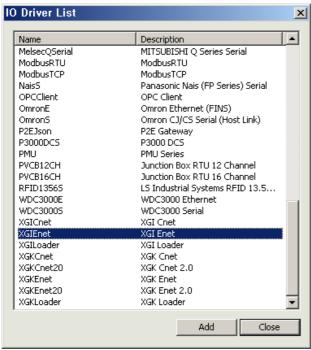


(4) I/O Address

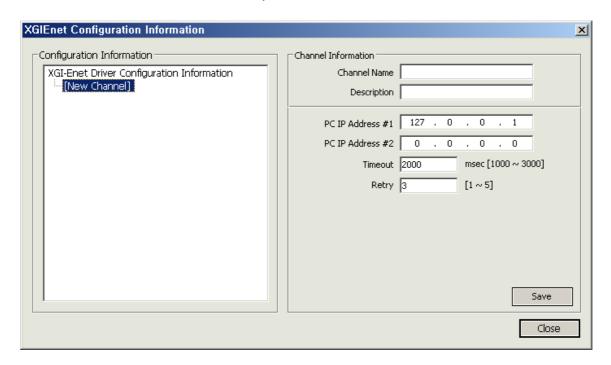
- (a) Type
 - -Analog: %MX0, %MB0, %MW0, %MD0, %ML0, %IW0.0.0, %QW0.0.0
 - -.Digital: %MB0.0, %MW0.0, %MD0.0, %ML0.0, %IX0.0.0, %QX0.0.0
- (b) Available devices
 - -. I, M, Q

7.12 **XGIEnet**

(1) Add Channel



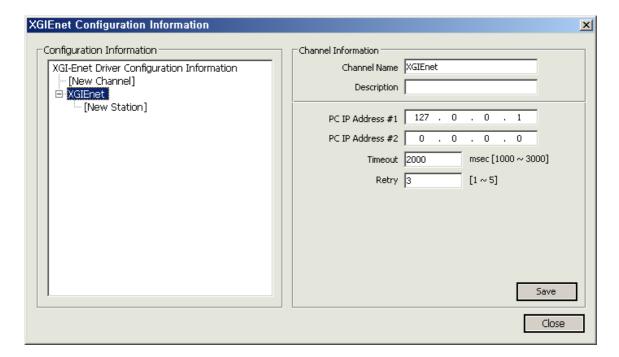
Select"XGIEnet"from the I/O driver list and press "Add."



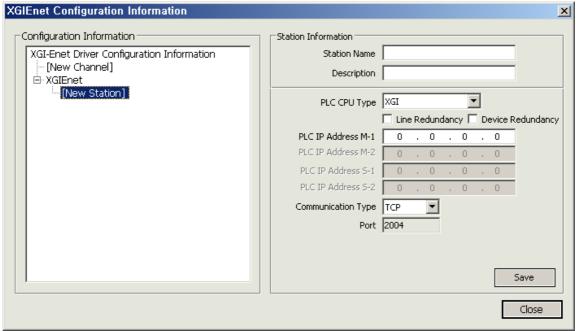
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) Serve IP: Input PC's IP Address.
- (d) Server IP Address #2: If Line Redundancy will be used, input the second IP Address to be used.
- (e) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process.

The time period to be set will be a base to judge communication errors.

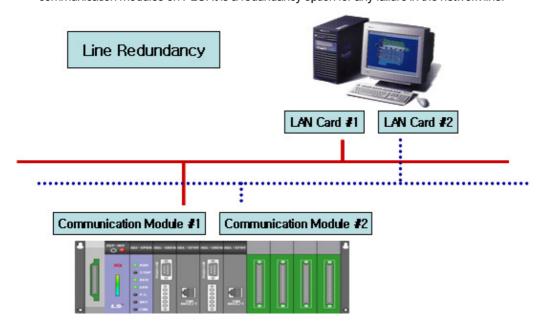
- (f) Retry: Set up the number of times to retry when communication fails.
- (g) Save: If 'Save' button is pressed, Channel information will be saved and the saved information will add to the left "Configuration Information" tree.



(2) Add Station

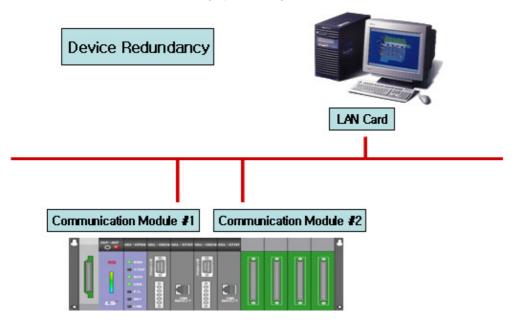


- (a) Select [New Station] from "Configuration Information" tree.
- (b) Station Name: Input a station name.
- (c) Description: Input some information on the station.
- (d) PLC CPU Type: Select a PLC CPU type.
- (e) Line Redundancy: Check √ in the box to use Line Redundancy. It is used when the network is intended to be separated and communicated as seen in the figure below by installing two LAN cards on the computer and two Enet communication modules on PLC. It is a redundancy option for any failure in the network line.

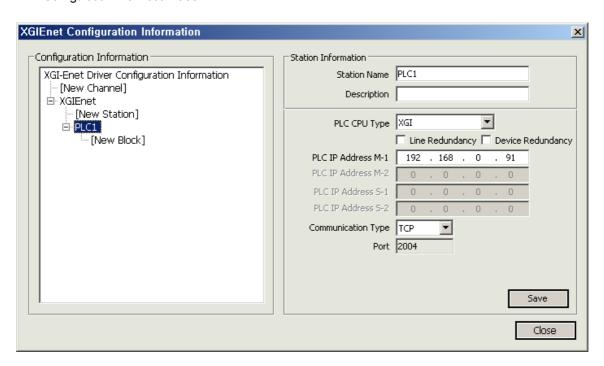


(f) Device Redundancy: Check $\sqrt{\ }$ in the box to use Device Redundancy. It is used when the communication module is separated as seen in the figure below by installing one LAN card on the computer and two Enet communication

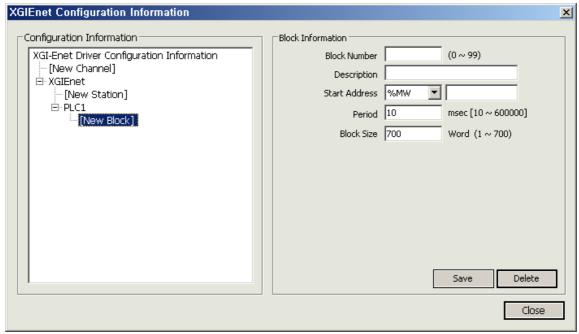
modules on PLC. It is a redundancy option for any failure in the PLC communication module.



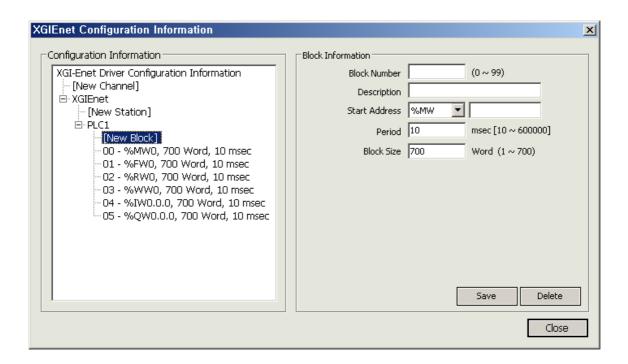
- (g) PLC IP Address #1-1: Input PLC's IP Address.
- (h) PLC IP Address #1-2: Input PLC's IP Address. Input the address when using Device Redundancy.
- (i) PLC IP Address #2-1: Input PLC's IP Address. Input the address when using Device Redundancy.
- (j) PLC IP Address #2-2: Input PLC's IP Address. Input the address when using Line Redundancy along with Device Redundancy.
- (k) Communication Type: Select either TCP or UDP.
- (I) Port: The port number is automatically inputted according to the selected communication type.
- (m) Save: If 'Save' button is pressed, Station information will be saved and the saved information will add to the left "Configuration Information" tree.



(3) Add Block



- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: Input the Block's Start Address. There are six kinds and each address is designated as the following ways respectively:
 - -. Right example: %MW0, %MW20, %IW0.0.0, %QW1.0.0
 - -. Wrong example: %MW0.0.0, %IW0, %QW0
- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Block Size: Input a block size of the relevant block (unit: Word (2 byte)).
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.

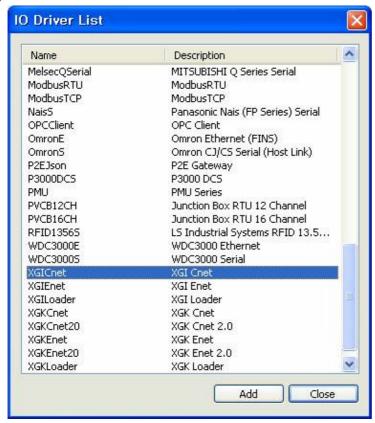


(4) I/O Address

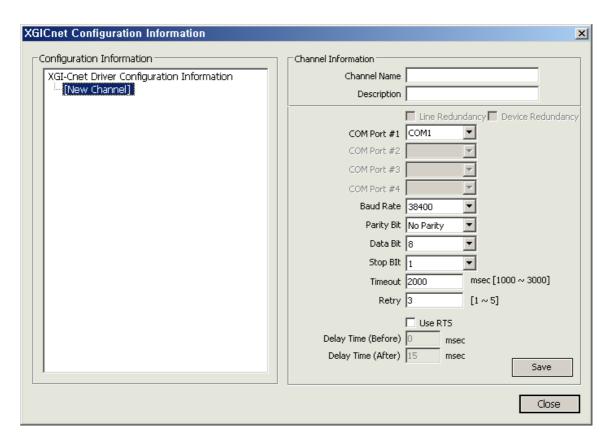
- (a) Type
 - -Analog: %MX0, %MB0, %MW0, %MD0, %ML0, %IW0.0.0, %QW0.0.0
 - -.Digital: %MB0.0, %MW0.0, %MD0.0, %ML0.0, %IX0.0.0, %QX0.0.0
- (b) Available devices
 - -. F, I, M, Q, R, W

7.13 **XGICnet**

(1) Add Channel



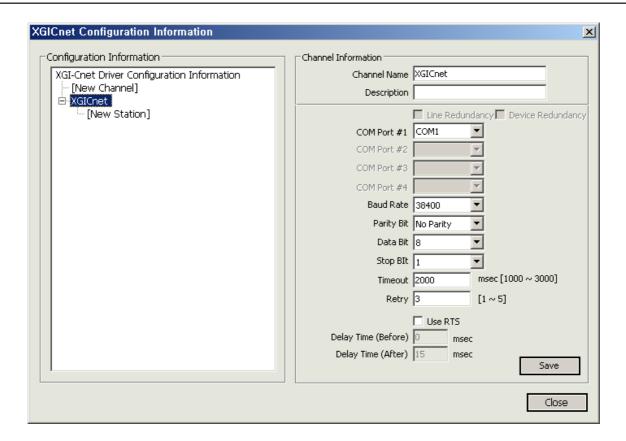
Select "XGICnet" from the I/O driver list and press "Add."



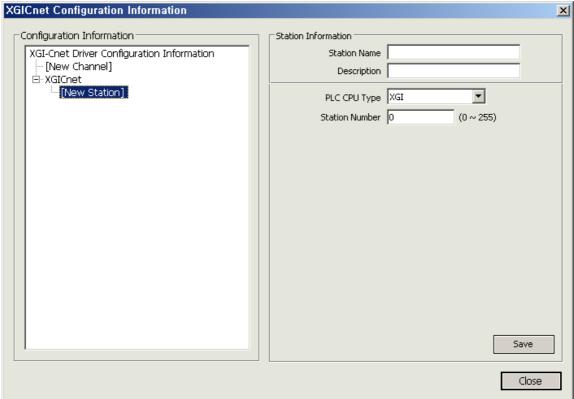
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) COM Port #1: Select a communication port.
- (d) Baud Rate: Select a communication speed.
- (e) Parity Bit: Select a parity bit.
- (f) Data Bit: Select a data bit.
- (g) Stop Bit: Select a stop bit.
- (h) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process.

The time period to be set will be a base to judge communication errors.

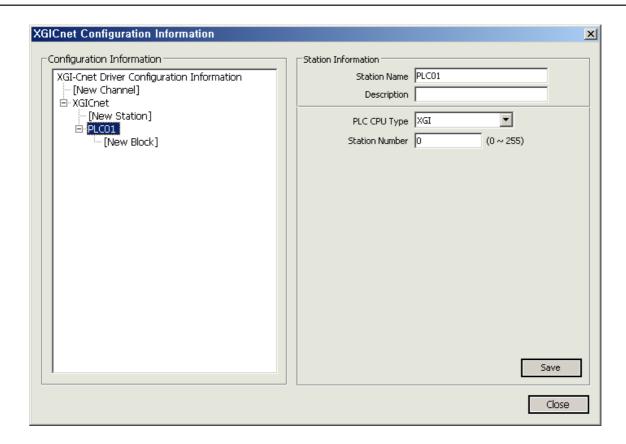
- (i) Retry: Set up the number of times to retry when communication fails.
- (j) Use RIS: Check $\sqrt{}$ in the box only when using Channel 485 and 422.
- (k) Delay Time (Before): Information used only in Channel 485 and 422. The delay time right before requesting Data to PLC.
- (I) Delay Time (After): Information used only in Channel 485 and 422. The delay time right after requesting Data to PLC.
- (m) Save: If 'Save' button is pressed, Channel information will be saved and the saved information will add to the left "Configuration Information" tree.



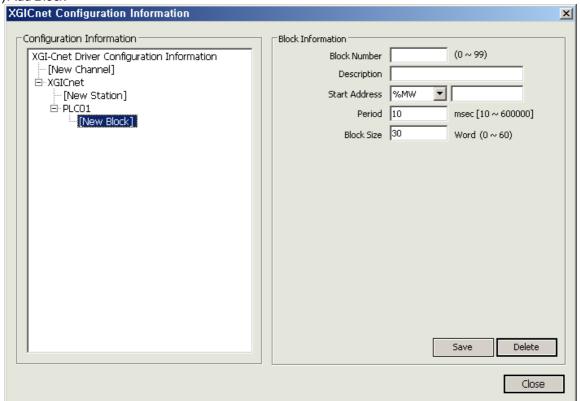
(2) Add Station



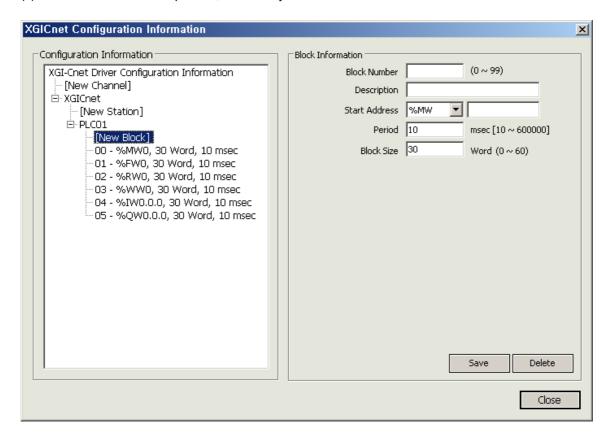
- (a) Select [New Station] from "Configuration Information" tree.
- (b) Station Name: Input a station name.
- (c) Description: Input some information on the station.
- (d) PLC CPU Type: Select a PLC CPU type.
- (e) Station Number: Input the prefix number of PLC Cnet Module.
- (f) Save: If 'Save' button is pressed, Station information will be saved and the saved information will add to the left "Configuration Information" tree.



(3) Add Block



- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: Input the Block's Start Address. There are six kinds and each address is designated as the following ways respectively:
 - -. Right example: %MW0, %MW20, %IW0.0.0, %QW1.0.0
 - -. Wrong example: %MW0.0.0, %IW0, %QW0
- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Block Size: Input a block size of the relevant block (unit: Word (2 byte)).
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.

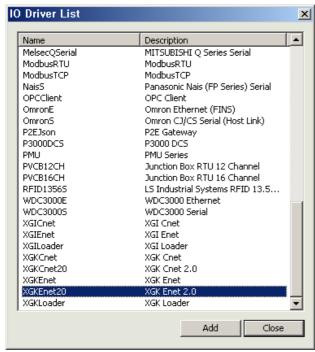


(4) I/O Address

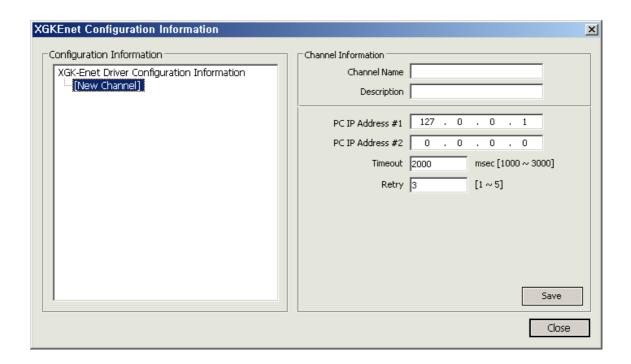
- (a) Type
 - -Analog: %MX0, %MB0, %MW0, %MD0, %ML0, %IW0.0.0, %QW0.0.0
 - -.Digital: %MB0.0, %MW0.0, %MD0.0, %ML0.0, %IX0.0.0, %QX0.0.0
- (b) Available devices
 - -. F, I, M, Q, R, W

7.14 XGKEnet20

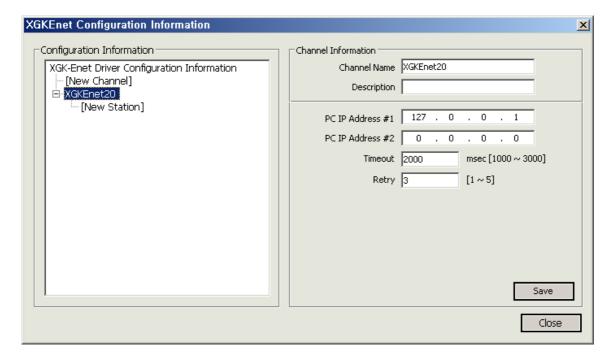
(1) Add Channel



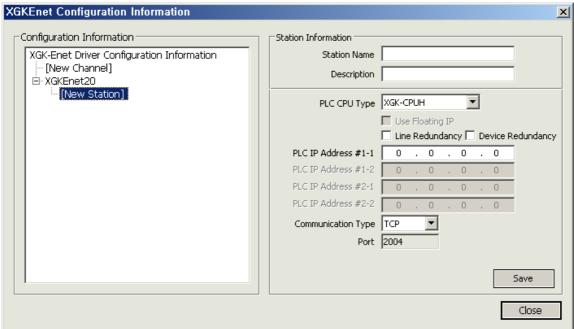
Select "XGKEnet20" from the I/O driver list and press "Add."



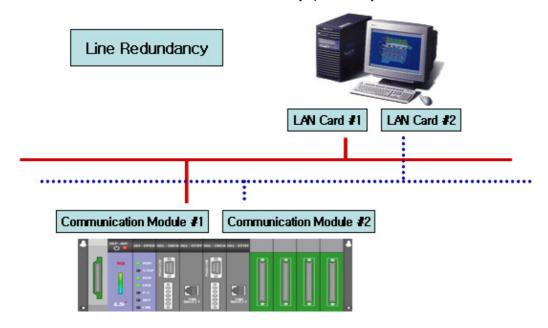
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) Serve IP: Input PC's IP Address.
- (d) Server IP Address #2: If Line Redundancy will be used, input the second IP Address to be used.
- (e) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (f) Retry: Set up the number of times to retry when communication fails.
- (g) Save: If 'Save' button is pressed, Station information will be saved and the saved information will add to the left "Configuration Information" tree.



(2) Add Station

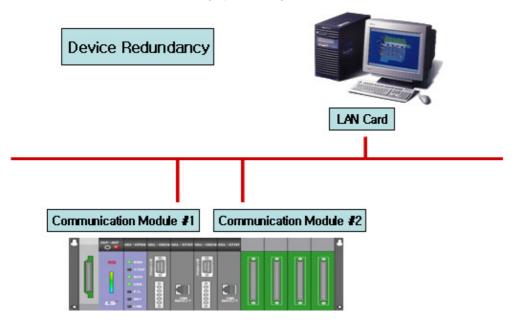


- (a) Select [New Station] from "Configuration Information" tree.
- (b) Station Name: Input a station name.
- (c) Description: Input some information on the station.
- (d) PLC CPU Type: Select a PLC CPU type.
- (e) Line Redundancy: Check √ in the box to use Line Redundancy. It is used when the network is intended to be separated and communicated as seen in the figure below by installing two LAN cards on the computer and two Enet communication modules on PLC. It is a redundancy option for any failure in the network line.

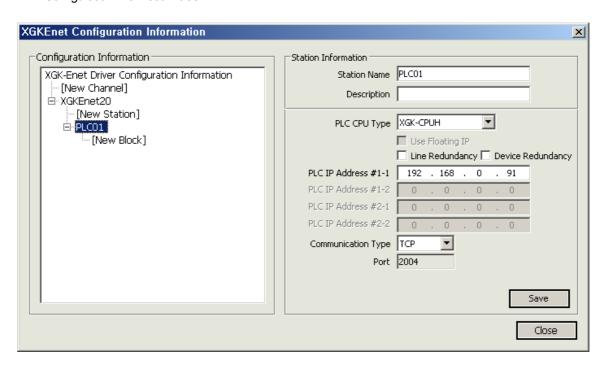


(f) Device Redundancy: Check $\sqrt{\ }$ in the box to use Device Redundancy. It is used when the communication module is separated as seen in the figure below by installing one LAN card on the computer and two Enet communication

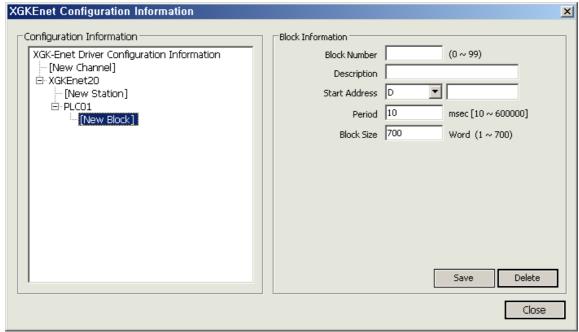
modules on PLC. It is a redundancy option for any failure in the PLC communication module.



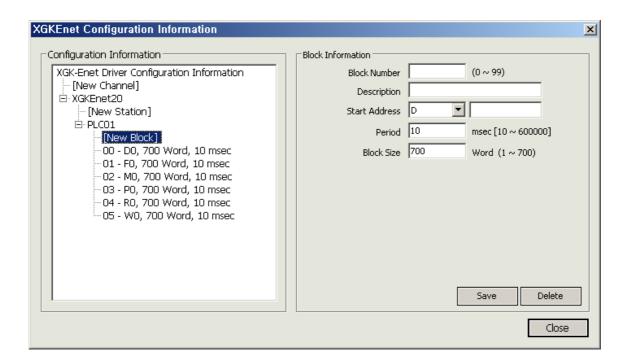
- (g) PLC IP Address #1-1: Input PLC's IP Address.
- (h) PLC IP Address #1-2: Input PLC's IP Address. Input the address when using Device Redundancy.
- (i) PLC IP Address #2-1: Input PLC's IP Address. Input the address when using Device Redundancy.
- (j) PLC IP Address #2-2: Input PLC's IP Address. Input the address when using Line Redundancy along with Device Redundancy.
- (k) Communication Type: Select either TCP or UDP.
- (I) Port: The port number is automatically inputted according to the selected communication type.
- (m) Save: If 'Save' button is pressed, Station information will be saved and the saved information will add to the left "Configuration Information" tree.



(3) Add Block



- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: Input the Block's Start Address. There are six kinds and each address is designated as the following ways respectively:
 - -. Right example: DO, F20, M10, P30, R20, W20
 - -. Wrong example: M0, 0.0, F11A
- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Block Size: Input a block size of the relevant block (unit: Word (2 byte)).
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.

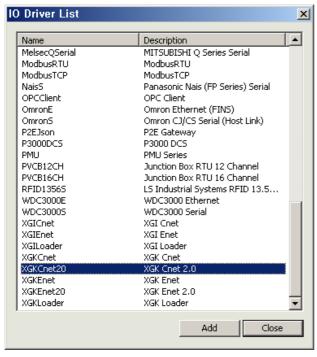


(4) I/O Address

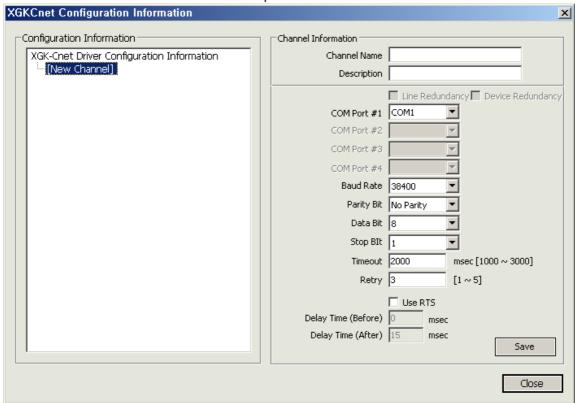
- (a) Type
 - -Analog: D20, F10, M30, PO, R20
 - -Digital: D20A.F105, M30F, P01, R20F (The last letter indicates the bit location of the corresponding word value. Available from O to F)
- (b) Available devices
 - -. D, F, M, P, R, W

7.15 XGKCnet20

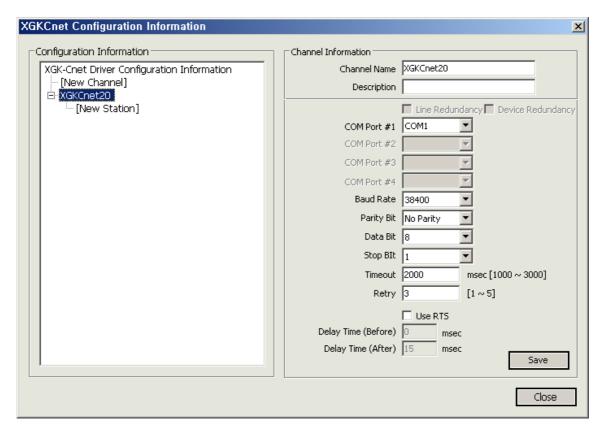
(1) Add Channel



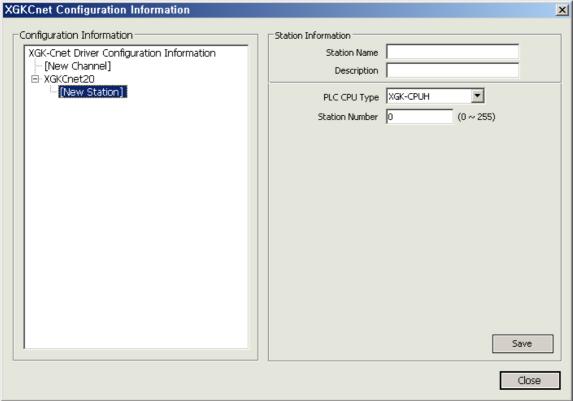
Select"XGKCnet20"from the I/O driver list and press "Add."



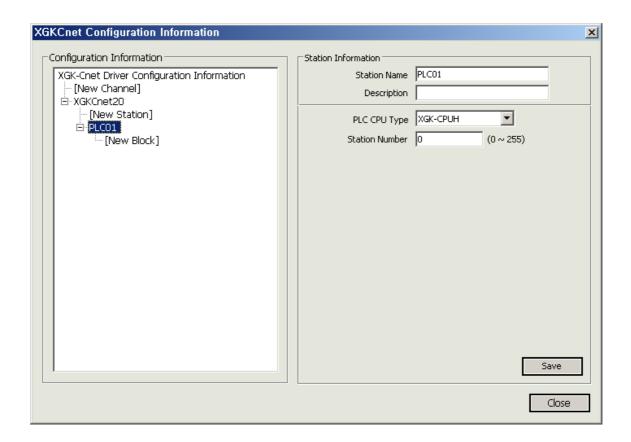
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) COM Port #1: Select a communication port.
- (d) Baud Rate: Select a communication speed.
- (e) Parity Bit: Select a parity bit.
- (f) Data Bit: Select a data bit.
- (g) Stop Bit: Select a stop bit.
- (h) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (i) Retry: Set up the number of times to retry when communication fails.
- (j) Use RIS: Check $\sqrt{}$ in the box only when using Channel 485 and 422.
- (k) Delay Time (Before): Information used only in Channel 485 and 422. The delay time right before requesting Data to PLC.
- (I) Delay Time (After): Information used only in Channel 485 and 422. The delay time right after requesting Data to PLC.
- (m) Save: If 'Save' button is pressed, Channel information will be saved and the saved information will add to the left "Configuration Information" tree.



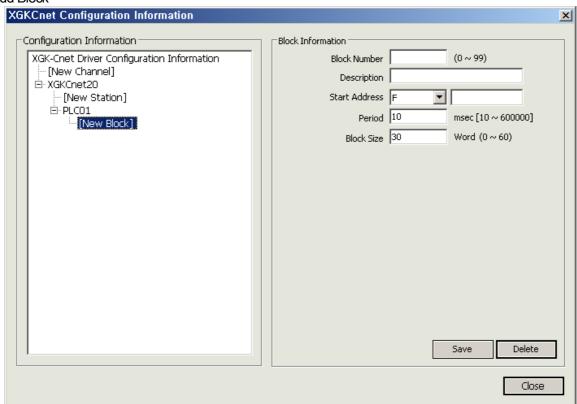
(2) Add Station



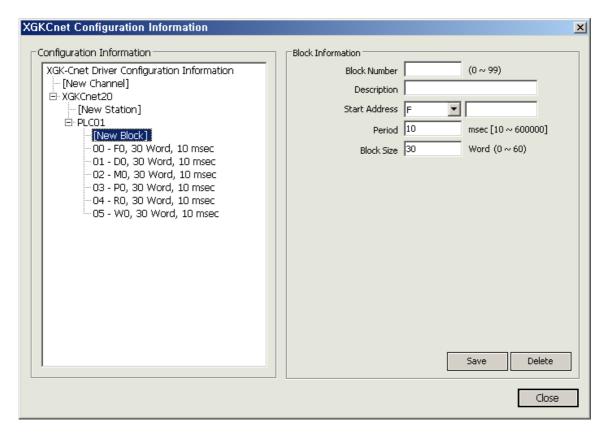
- (a) Station Name: Input a station name.
- (b) Description: Input some information on the station.
- (c) PLC CPU Type: Select a PLC CPU type.
- (d) Station Number: Input the prefix number of PLC Cnet Module.
- (f) Save: If 'Save' button is pressed, station information will be saved and the saved information will add to the left "Configuration Information" tree.



(3) Add Block



- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: Input the Block's Start Address. There are six kinds and each address is designated as the following ways respectively:
 - -. Right example: DO, F20, M10, P30, R20, W20
 - -. Wrong example: M0, 0.0, F11A
- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Block Size: Input a block size of the relevant block (unit: Word (2 byte)).
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.

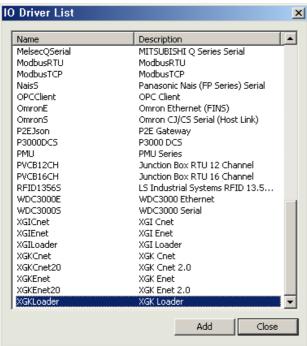


(4) I/O Address

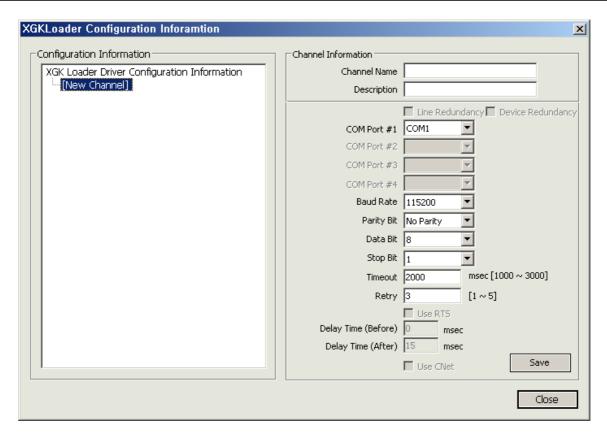
- (a) Type
 - -Analog: D20, F10, M30, PO, R20
 - -Digital: D20A.F105, M30F, P01, R20F (The last letter indicates the bit location of the corresponding word value. Available from O to F)
- (b) Available devices
 - -. D, F, M, P, R, W

7.16 XGKLoader

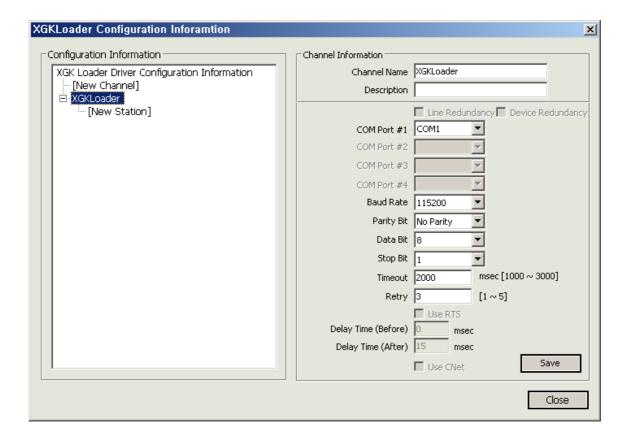
(1) Add Channel



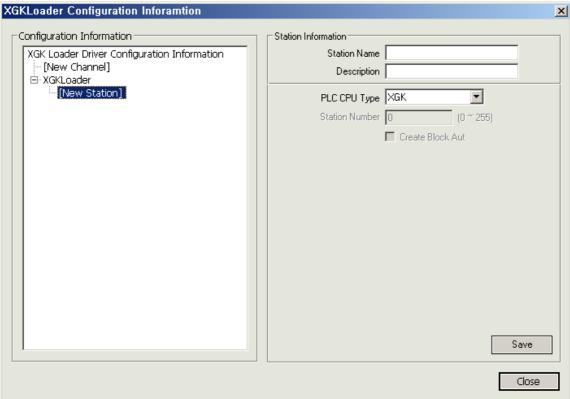
Select"XGKLoader"from the I/O driver list and press "Add."



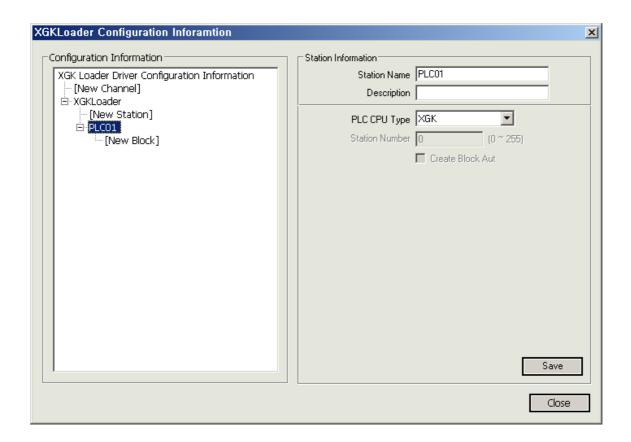
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) COM Port #1: Select a communication port.
- (d) Baud Rate: Select a communication speed.
- (e) Parity Bit: Select a parity bit.
- (f) Data Bit: Select a data bit.
- (g) Stop Bit: Select a stop bit.
- (h) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (i) Retry: Set up the number of times to retry when communication fails.
- (j) Save: If 'Save' button is pressed, Channel information will be saved and the saved information will add to the left "Configuration Information" tree.



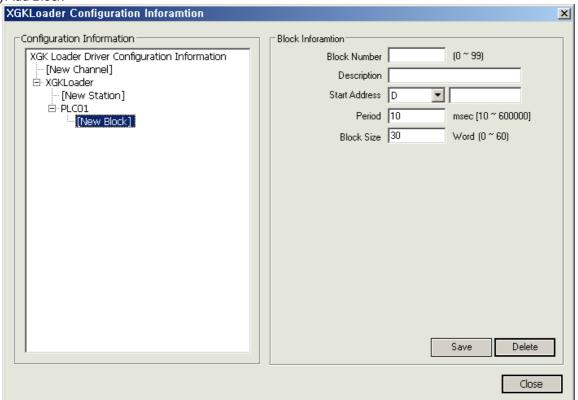
(2) Add Station



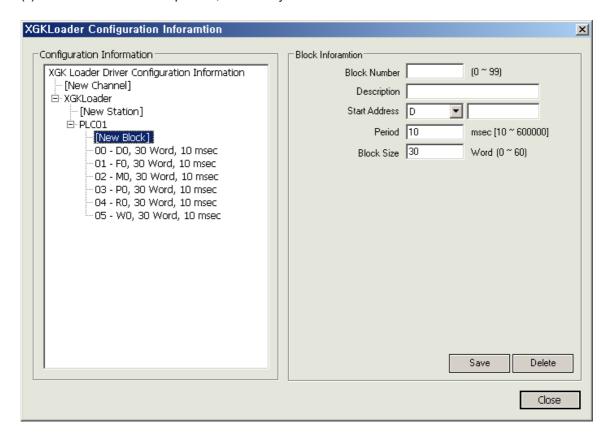
- (a) Select [New Station] from "Configuration Information" tree.
- (b) Station Name: Input a station name.
- (c) Description: Input some information on the station.
- (d) PLC CPU Type: Select a PLC CPU type.
- (e) Save: If 'Save' button is pressed, Station information will be saved and the saved information will add to the left "Configuration Information" tree.



(3) Add Block



- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: Input the Block's Start Address. There are six kinds and each address is designated as the following ways respectively:
 - -. Right example: DO, F20, M10, P30, R20, W20
 - -. Wrong example: M0, 0.0, F11A
- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Block Size: Input a block size of the relevant block (unit: Word (2 byte)).
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.

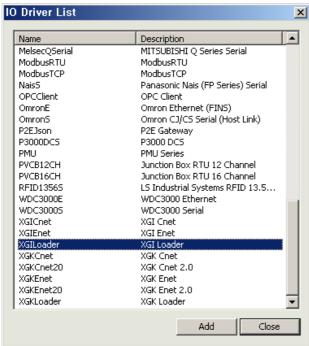


(4) I/O Address

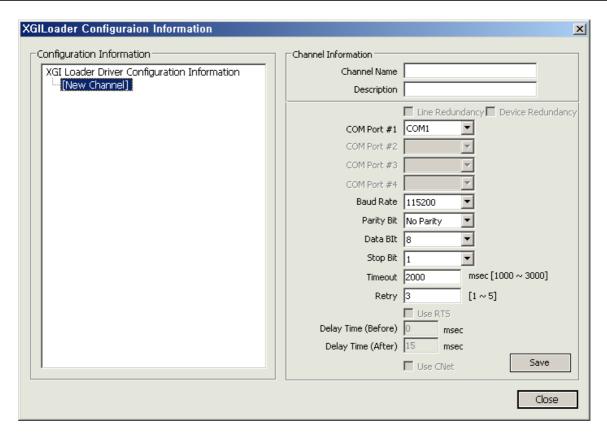
- (a) Type
 - -Analog: D20, F10, M30, PO, R20
 - -Digital: D20A.F105, M30F, P01, R20F (The last letter indicates the bit location of the corresponding word value. Available from O to F)
- (b) Available devices
 - -. D, F, M, P, R, W

7.17 XGILoader

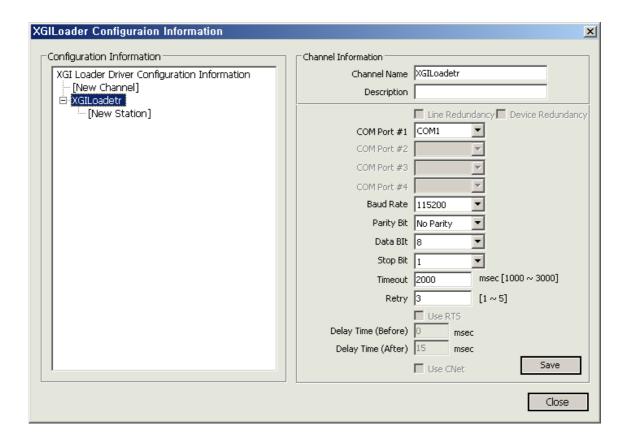
(1) Add Channel



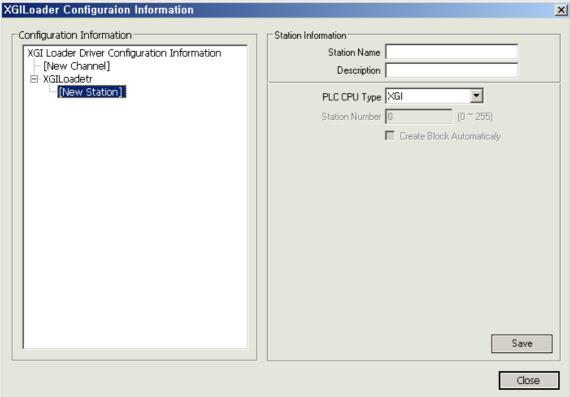
• SELECT "XGILoader" from the I/O driver list and press "Add."



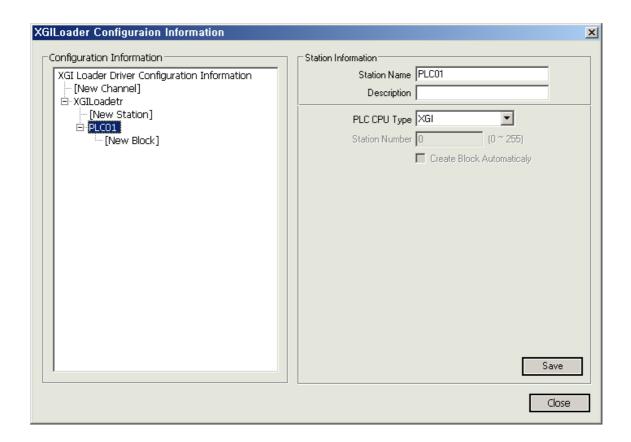
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) COM Port #1: Select a communication port.
- (d) Baud Rate: Select a communication speed.
- (e) Parity Bit: Select a parity bit.
- (f) Data Bit: Select a data bit.
- (g) Stop Bit: Select a stop bit.
- (h) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (i) Retry: Set up the number of times to retry when communication fails.
- (j) Save: If 'Save' button is pressed, Channel information will be saved and the saved information will add to the left "Configuration Information" tree.



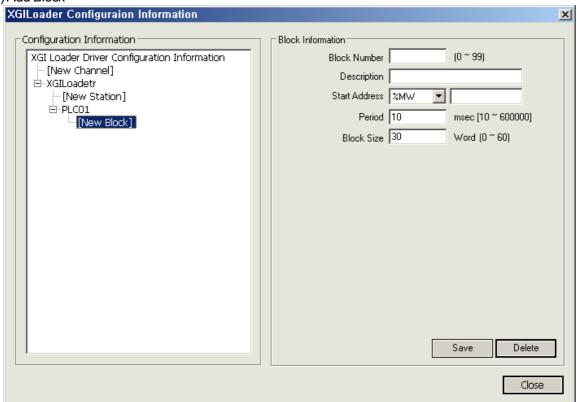
(2) Add Station



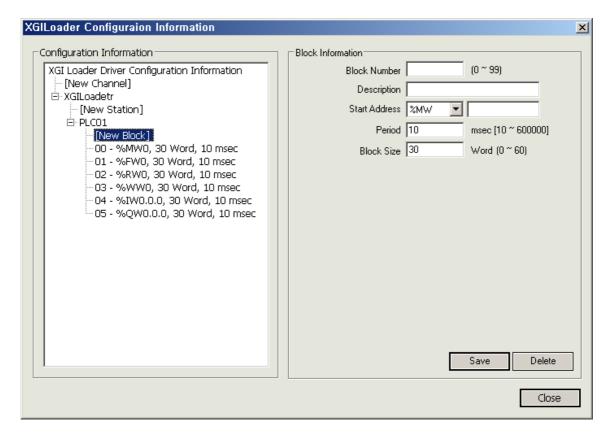
- (a) Select [New Station] from "Configuration Information" tree.
- (b) Station Name: Input a station name.
- (c) Description: Input some information on the station.
- (d) PLC CPU Type: Select a PLC CPU type.
- (e) Save: If 'Save' button is pressed, Station information will be saved and the saved information will add to the left "Configuration Information" tree.



(3) Add Block



- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: Input the Block's Start Address. There are six kinds and each address is designated as the following ways respectively:
 - -. Right example: %MW0, %MW20, %IW0.0.0, %QW1.0.0
 - -. Wrong example: %MW0.0.0, %IW0, %QW0
- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Block Size: Input a block size of the relevant block (unit: Word (2 byte)).
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.

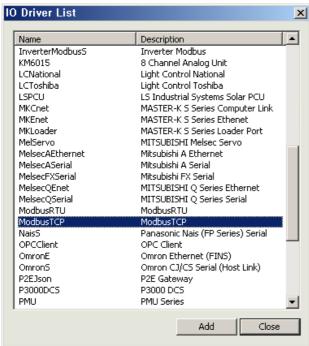


(4) I/O Address

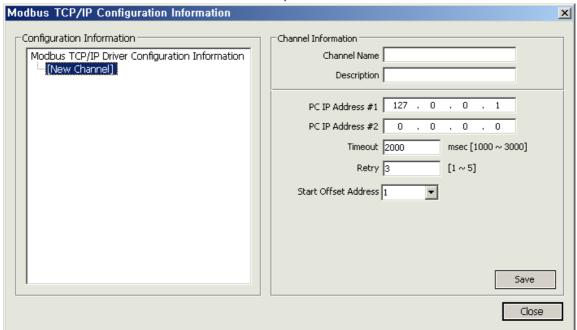
- (a) Type
 - -Analog: %MX0, %MB0, %MW0, %MD0, %ML0, %IW0.0.0, %QW0.0.0
 - -.Digital: %MB0.0, %MW0.0, %MD0.0, %ML0.0, %IX0.0.0, %QX0.0.0
- (b) Available devices
 - -. F, I, M, Q, R, W

7.18 **ModbusTCP**

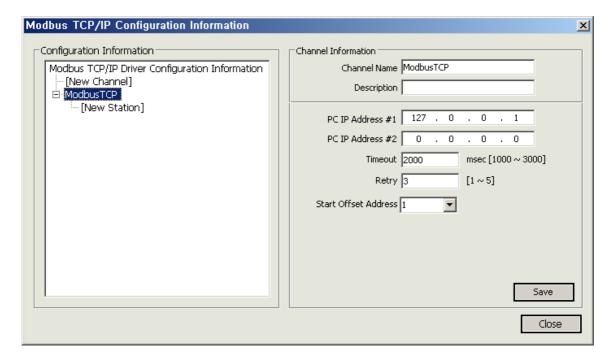
(1) Add Channel



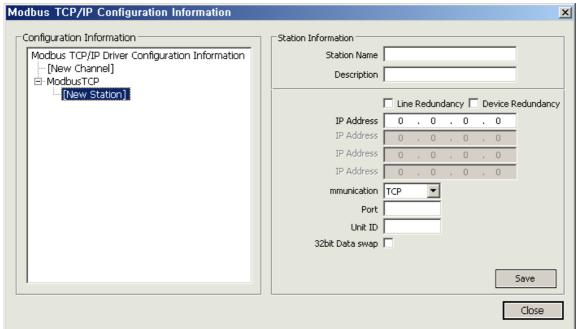
SELECT "ModbusTCP" from the I/O driver list and press "Add."



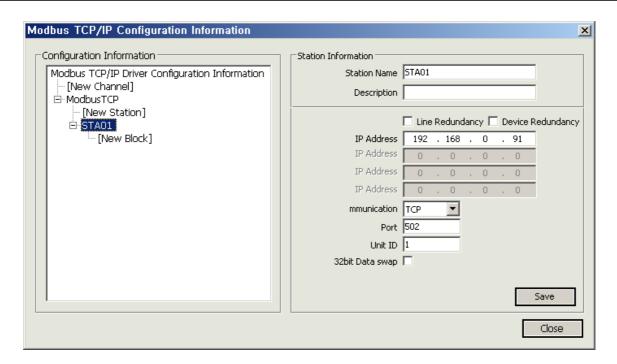
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel
- (c) Serve IP: Input PC's IP Address.
- (d) Server IP Address #2: If Line Redundancy will be used, input the second IP Address to be used.
- (e) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (f) Retry: Set up the number of times to retry when communication fails.
- (g) Start Offset Address: Select either 0 or 1. It sets up whether to start from either 0 or 1.
- (h) Save: If 'Save' button is pressed, Channel information will be saved and the saved information will add to the left "Configuration Information" tree.



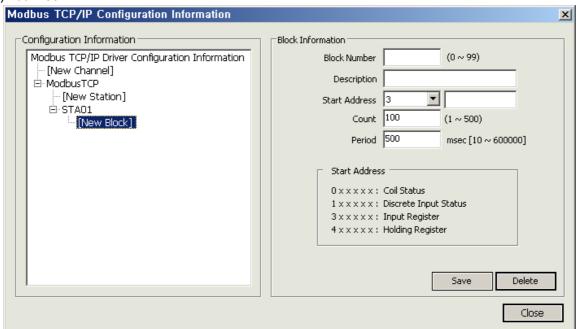
(2) Add Station



- (a) Select [New Station] from "Configuration Information" tree.
- (b) Station Name: Input a station name.
- (c) Description: Input some information on the station.
- (d) Line Redundancy: Check √in the box to use Line Redundancy.
- (e) Device Redundancy: Check √in the box to use Device Redundancy.
- (f) PLC IP Address #1-1: Input PLC's IP Address.
- (g) PLC IP Address #1-2: Input PLC's IP Address. Input the address when using Device Redundancy.
- (h) PLC IP Address #2-1: Input PLC's IP Address. Input the address when using Device Redundancy.
- (i) PLC IP Address #2-2: Input PLC's IP Address. Input the address when using Line Redundancy along with Device Redundancy.
- (j) Communication Type: Select either TCP or UDP.
- (k) Port: The port number is automatically inputted according to the selected communication type.
- (I) Unit ID: Input the ID of the Modbus TCP device the user wants to communicate.
- (m) 32bit Data Swap: 32bit can be expressed in the combination of [Upper level word + Lower level word] or [Lower level word + Upper level word] and for this, it is available to use swap of upper level word and lower level word.
- (n) Save: If 'Save' button is pressed, Station information will be saved and the saved information will add to the left "Configuration Information" tree.

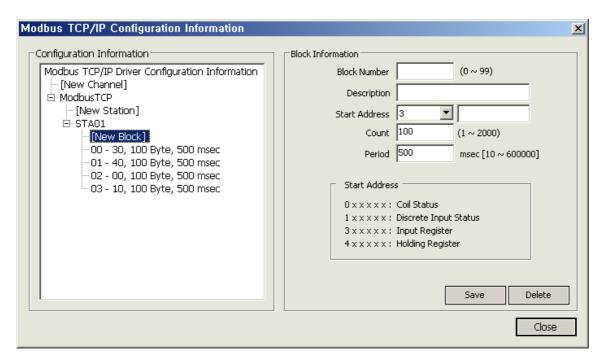


(3) Add Block



- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: Input the Block's Start Address. There are four kinds and each address is designated as the following ways respectively:
 - -. Address delimiter O(Coil Status), 1(Discrete Input Status), 3(Input Register), 4(Holding Register)
 - -. Its address is $0\sim65536$ and if "Start Offset Address" is set up to start from 1 in the configuration information , it will be $1\sim65536$

- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Count: A number to read for each corresponding delimiter.
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.



(4) I/O Address

(a) Type

- Analog: Delimiter (3 or 4) \pm 0~65535 (if "Start Offset Address" is set up to start from 1 in the configuration information, it will be 1~65536)

Example 1) If the user wants to read Oth(0 of 0~65535) of No. 3 (Input Register: Read only),

Set up 300000 or 30 if 0 is selected for "Start Offset Address"

Set up 300001 or 31 if 1 is selected for "Start Offset Address"

Example 2) If the user wants to read 17th(17 of 0~65535) of No. 4 (Holding Register),

Set up 400017 or 417 if 0 is selected for "Start Offset Address"

Set up 400018 or 418 if 1 is selected for "Start Offset Address"

- Digital: Delimiter (0 or 1, or 3 or 4) +0~65535 (if "Start Offset Address" is set up to start from 1 in the configuration information, it will be 1~65536)

Example 1) If the user wants to read Oth(0 of 0~65535) of No. 0 (Coil Status: Read/Write),

Set up 000000 or 00 if 0 is selected for "Start Offset Address"

Set up 000001 or 01 if 1 is selected for "Start Offset Address"

Example 2) If the user wants to read 17th(17 of 0~65535) of No. 1 (Discrete Input Status: Read only),

Set up 100017 or 117 if 0 is selected for "Start Offset Address"

Set up 100018 or 118 if 1 is selected for "Start Offset Address"

Example 3) If the user wants to read 10th bit of 17th(17 of 0~65535) of No. 3 (Input Register: Read only), Set up 300017.A or 317.A if 0 is selected for "Start Offset Address"

Set up 300018.A or 318.A if 1 is selected for "Start Offset Address"

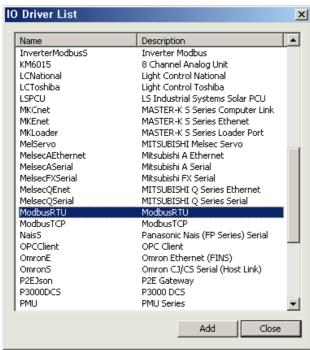
Example 4) If the user wants to read 15th bit of 0th(0 of 0~65535) of No. 4 (Holding Register),

Set up 300000.F or 30.F if 0 is selected for "Start Offset Address"

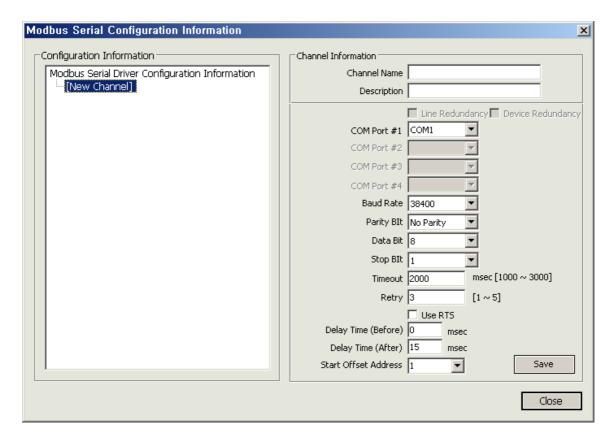
Set up 300001.F or 31.F if 1 is selected for "Start Offset Address"

7.19 ModbusRTU

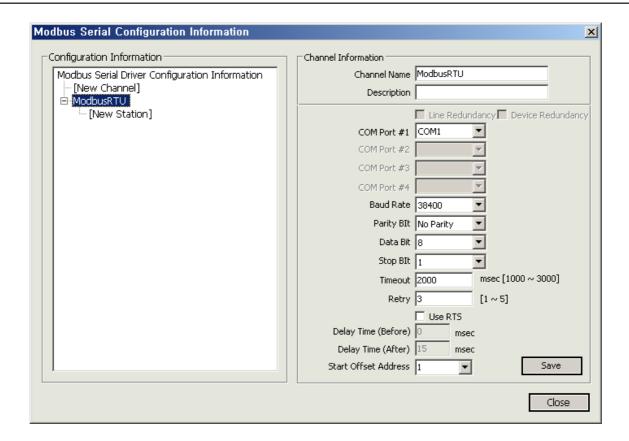
(1) Add Channel



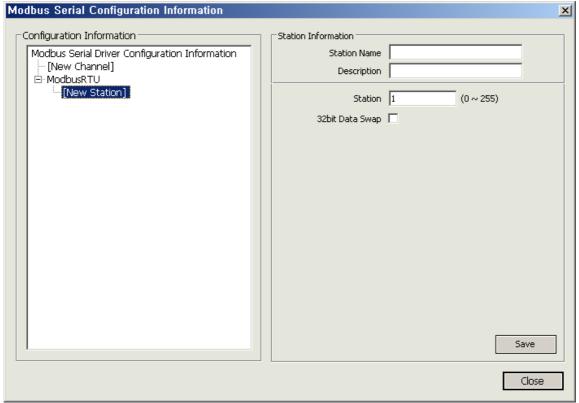
• SELECT "ModbusRTU" from the I/O driver list and press "Add."



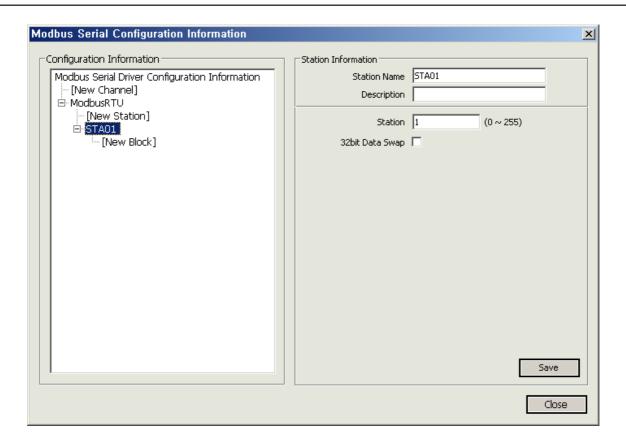
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) COM Port #1: Select a communication port.
- (d) Baud Rate: Select a communication speed.
- (e) Parity Bit: Select a parity bit.
- (f) Data Bit: Select a data bit.
- (g) Stop Bit: Select a stop bit.
- (h) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (i) Retry: Set up the number of times to retry when communication fails.
- (j) Use RIS: Check $\sqrt{ }$ in the box only when using Channel 485 and 422.
- (k) Delay Time (Before): Information used only in Channel 485 and 422. The delay time right before requesting Data to PLC
- (I) Delay Time (After): Information used only in Channel 485 and 422. The delay time right after requesting Data to PLC
- (m) Start Offset Address: Select either 0 or 1. It sets up whether to start from either 0 or 1.
- (n) Save: If 'Save' button is pressed, Channel information will be saved and the saved information will add to the left "Configuration Information" tree.



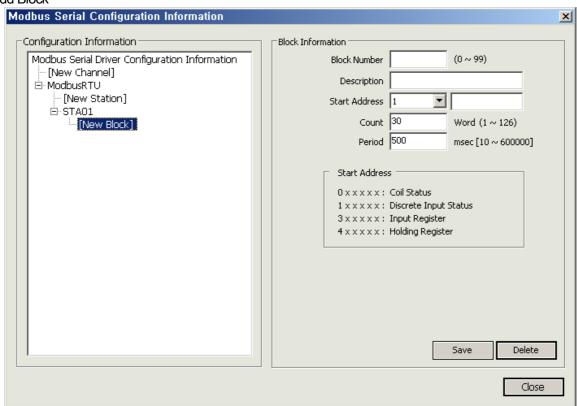
(2) Add Station



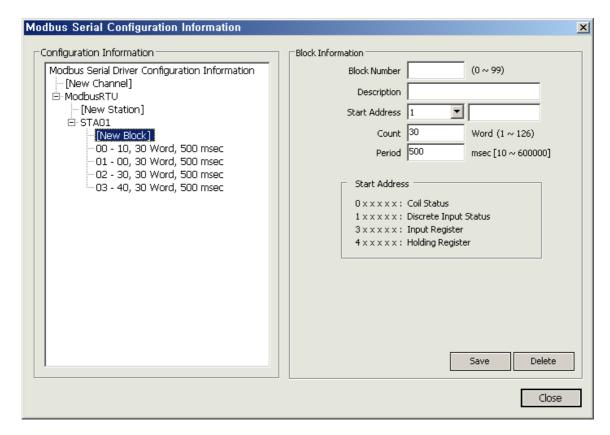
- (a) Select [New Station] from "Configuration Information" tree.
- (b) Station Name: Input a station name.
- (c) Description: Input some information on the station.
- (d) PLC CPU Type: Select a PLC CPU type.
- (e) Station Number: Input the prefix number of PLC Cnet Module.
- (f) 32bit Data Swap: 32bit can be expressed in the combination of [Upper level word + Lower level word] or [Lower level word + Upper level word] and for this, it is available to use swap of upper level word and lower level word.
- (g) Save: If 'Save' button is pressed, Station information will be saved and the saved information will add to the left "Configuration Information" tree.



(3) Add Block



- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: Input the Block's Start Address. There are four kinds and each address is designated as the following ways respectively:
 - -. Address delimiter 0(Coil Status), 1(Discrete Input Status), 3(Input Register), 4(Holding Register)
 - -. Its address is $0\sim65536$ and if "Start Offset Address" is set up to start from 1 in the configuration information , it will be $1\sim65536$
- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Count: A number to read for each corresponding delimiter.
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.



(4) I/O Address

- (a) Type
 - Analog: Delimiter (3 or 4) $+0\sim65535$ (if "Start Offset Address" is set up to start from 1 in the configuration information, it will be $1\sim65536$)

Example 1) If the user wants to read Oth(0 of 0~65535) of No. 3 (Input Register: Read only), Set up 300000 or 30 if 0 is selected for "Start Offset Address"

Out are 000000 or 00 if 4 is a standard of other office the days of

Set up 300001 or 31 if 1 is selected for "Start Offset Address"

Example 2) If the user wants to read 17th(17 of 0~65535) of No. 4 (Holding Register), Set up 400017 or 417 if 0 is selected for "Start Offset Address" Set up 400018 or 418 if 1 is selected for "Start Offset Address"

- Digital: Delimiter (0 or 1, or 3 or 4) +0~65535 (if "Start Offset Address" is set up to start from 1 in the configuration information , it will be 1~65536)

Example 1) If the user wants to read Oth(0 of 0~65535) of No. 0 (Coil Status : Read/Write), Set up 000000 or 00 if 0 is selected for "Start Offset Address" Set up 000001 or 01 if 1 is selected for "Start Offset Address"

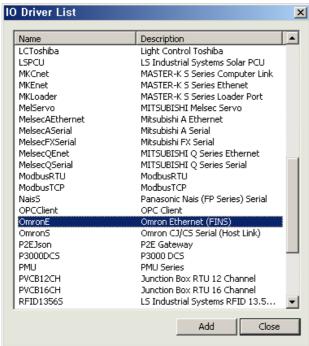
Example 2) If the user wants to read 17th(17 of 0~65535) of No. 1 (Discrete Input Status: Read only), Set up 100017 or 117 if 0 is selected for "Start Offset Address" Set up 100018 or 118 if 1 is selected for "Start Offset Address"

Example 3) If the user wants to read 10th bit of 17th(17 of 0~65535) of No. 3 (Input Register: Read only), Set up 300017.A or 317.A if 0 is selected for "Start Offset Address" Set up 300018.A or 318.A if 1 is selected for "Start Offset Address"

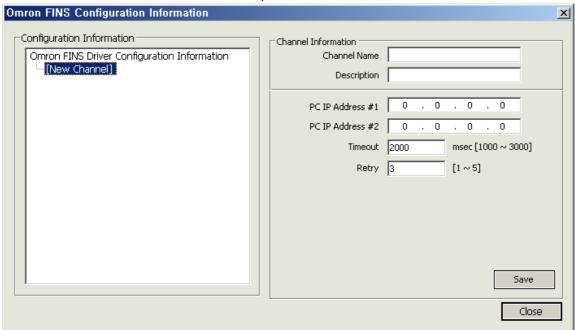
Example 4) If the user wants to read 15th bit of 0th(0 of 0~65535) of No. 4 (Holding Register), Set up 300000.F or 30.F if 0 is selected for "Start Offset Address" Set up 300001.F or 31.F if 1 is selected for "Start Offset Address"

7.20 **OmronE**

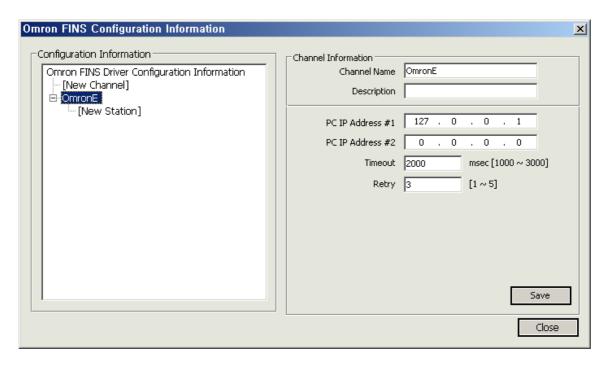
(1) Add Channel



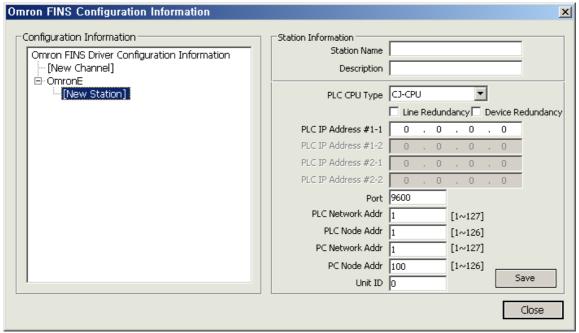
SELECT "OmronE" from the I/O driver list and press "Add."



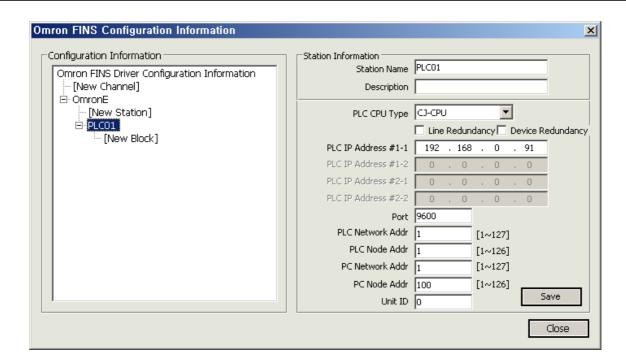
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) Serve IP: Input PC's IP Address.
- (d) Server IP Address #2: If Line Redundancy will be used, input the second IP Address to be used.
- (e) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (f) Retry: Set up the number of times to retry when communication fails.
- (g) Start Offset Address: Select either 0 or 1. It sets up whether to start from either 0 or 1.
- (h) Save: If 'Save' button is pressed, Channel information will be saved and the saved information will add to the left "Configuration Information" tree.



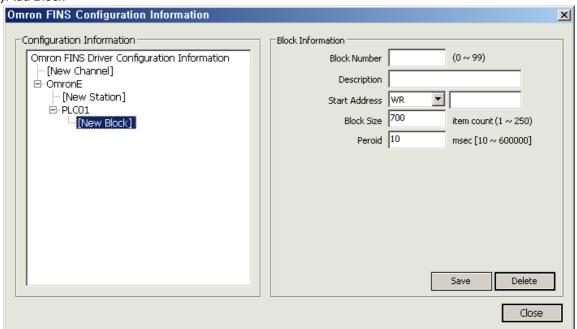
(2) Add Station



- (a) Select [New Station] from "Configuration Information" tree.
- (b) Station Name: Input a station name.
- (c) Description: Input some information on the station.
- (d) Line Redundancy: Check √in the box to use Line Redundancy.
- (e) Device Redundancy: Check √in the box to use Device Redundancy.
- (f) PLC IP Address #1-1: Input PLC's IP Address.
- (g) PLC IP Address #1-2: Input PLC's IP Address. Input the address when using Device Redundancy.
- (h) PLC IP Address #2-1: Input PLC's IP Address. Input the address when using Device Redundancy.
- (i) PLC IP Address #2-2: Input PLC's IP Address. Input the address when using Line Redundancy along with Device Redundancy.
- (j) Communication Type: Select either TCP or UDP.
- (k) Port:For FINS, a default value 9600 is set before it leaves the factory but it is allowed to change PLC setting values if other ports are preferred.
- (m) PC Node Addr: Input the network address of the Omron PLC device to be communicated with. Default '1'
- (n) PC Node Addr: Input the node address of the Omron PLC device to be communicated with. Default '1'
- (o) PC Node Addr: Input the network address of the PC (InfoU PC) being communicated with. Default '1'
- (p) PC Node Addr: Input the node address of the PC (InfoU PC) being communicated with. Default '100'
- (q) Unit ID: Select one among 0 through F. (Prefix)
- (r) Save: If 'Save' button is pressed, Station information will be saved and the saved information will add to the left "Configuration Information" tree.



(3) Add Block



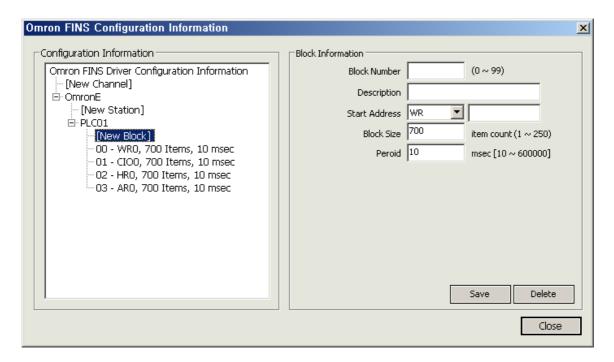
- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: Input the Block's Start Address.
 - -.Start Address : Memory Type (CIO,WR,HR,AR,TF,CF,EM0)+ Start Address
 - Block Size : A number of items to be loaded
 Example 1) Start Address: C1010, Block Size: If it is 100, 100 words will be loaded starting from the 10th word of CIO memory (starting from 0)

Example 2) Start Address: TF20, Block Size: If it is 20, 20 status bits will be loaded starting from the 20th status bit of TF memory (starting from 0)

« CIO,WR,HR,AR,TIM,CNT,DM,EM,EM0,EM1,EM2,EM3,EM4,EM5,EM6,EM7,EM8,EM9,EMA,EMB,EMCAND EM CURR, DR ARE DEVICES IN WORD UNIT AND IR IS A DEVICE IN DWORD UNIT.

TKF, TKS, TF and CF are devices in Bit unit.

- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Block Size: A number to read for each corresponding delimiter.
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.



(4) I/O Address

(a) Type

-. Example of Analog Input Address

CIO 100 : Load the value of the 100th word in CIO Area EM7 20000 : Load the value of the 20000th Word in EM7 Area

-. Example of Digital Input Address

CIO 90A : Load the 10(A)th bit value of the 90th word in CIO Area EM7_07 : Load the 7th bit value of the 0th word in EM7 Area

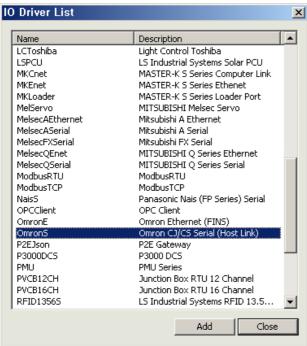
TF_123 : Load the completion flag value of the timer in Timer Flag Area

-. Omron PLC Memory Device Map

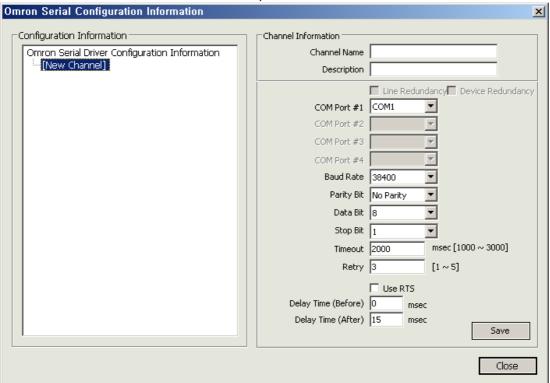
		iviap				
Memory Code	Description	Data Type	CS/CJ CPU Memory Address	CV CPU Memory Address	Remark	Address Example
CIO	CIO Area	WORD	0~6143	0~2555		Al: CIO_xxxx (xxxx: 0~6143) CIO_100 DI: CIO_xxxx0~F(xxxx: 0~6143) CIO_100A
WR	Work Area		0~511			AI: WR_xxxx (xxxx: 0~511) WR_100 DI: WR_xxxx0~F(xxxx: 0~511) WR_0F
HR	Holding Bit Area					AI: HR_xxxx (xxxx: 0~511) HR_234 DI: HR_xxxx0~F(xxxx: 0~511) HR 19
AR	Auxiliary Bit Area		0~959	0~959	0~447 (Read Only), 448~959 (Read/ Write)	AI: AR_xxxx (xxxx: 0~511) AR_800 DI: AR_xxxx0~F(xxxx: 0~511) AR_9001
TF	Timer Area	Completi	0~4095	0~2047 or 0~1023	Read Only	DI : TF_xxxx(xxxx : 0~4095), TF_2047
CF	Counter Area	on Flag(BIT)				DI : CF_xxxx(xxxx : 0~4095), CF_2048
TIM	Timer Area PV		0~4095	0~2047 or		AI: TIM_xxxx (xxxx: 0~4095) TIM_1000
CNT	Counter Area PV		0~4095	0~1023		AI: CNT_xxxx (xxxx: 0~4095) CNT_2000
DM	DM Area			0.00707		AI: DM_xxxxx (xxxxx: 0~32767) DM_10000 DI: DM_xxxxx0~F(xxxx: 0~32767) DM 900E
EM0~EM7	EM Area	WORD	0~32767	0~32767		AI: EM0_xxxxx (xxxxx: 0~32767) EM0_9000 DI: EM0_xxxxx0~F(xxxxx: 0~32767) EM0_7000A
EM7~EMC						AI:EMA_xxxxx (xxxxx:0~32767) EMA_9000 DI:EMA_xxxxx0~F(xxxxx: 0~32767) EMA_7000A
EM						AI: EM_xxxxx (xxxxx: 0~32767) EM_9000 DI: EM_xxxxx0~F(xxxxx: 0~32767) EM_7000A
EM_CURR			O of 0th (Current Bank No)		Read	AI:EM_0
TKF	Task Flag BIT	BIT	0~31		Only	DI : TKF_xxxx(xxxx : 0~31), TKF_21
TKS	Task Flag Status	DI1	0 01	_		DI : TKS_xxxx(xxxx : 0~31), TKS_1
IR	Index Register	DWORD		<u> </u>		Al: IR_xxxx (xxxx: 0~15) IR_5
DR	Data Register	WORD	0~15	0~15		AI: DR_xxxxx (xxxxx: 0~15) DR_9 DI: DR_xxxxx0~F(xxxxx: 0~15) DR_1A

7.21 OmronS

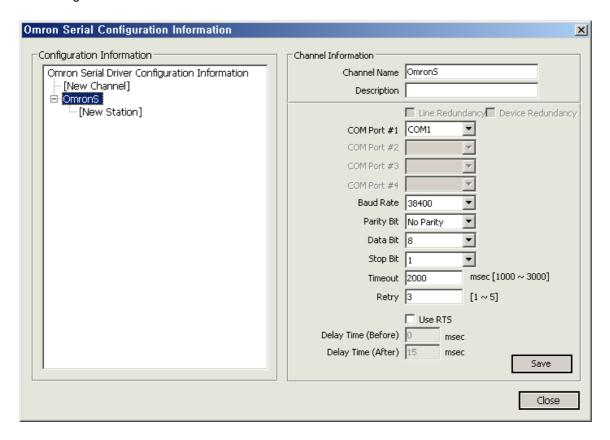
(1) Add Channel



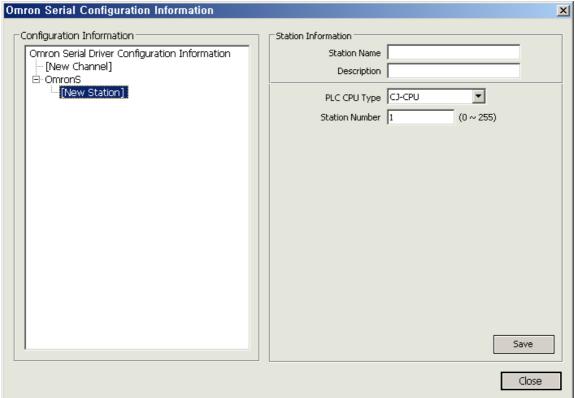
SELECT "OmronS" from the I/O driver list and press "Add."



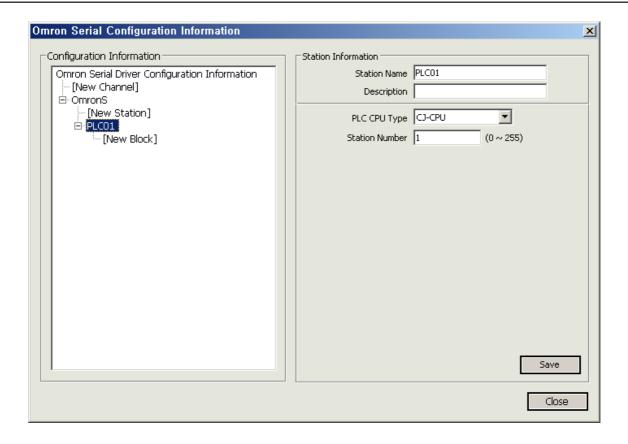
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) COM Port #1: Select a serial port of PC.
- (d) Baud Rate: Select a communication speed.
- (e) Parity Bit: Select a parity bit.
- (f) Data Bit: Select a data bit.
- (g) Stop Bit: Select a stop bit.
- (h) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (i) Period: It refers to an interval to request data.
- (j) Use RIS: Check $\sqrt{}$ in the box only when using Channel 485 and 422.
- (k) Delay Time (Before): Information used only in Channel 485 and 422. The delay time right before requesting Data to PLC
- (I) Delay Time (After): Information used only in Channel 485 and 422. The delay time right after requesting Data to PLC
- (m) Save: If 'Save' button is pressed, Channel information will be saved and the saved information will add to the left "Configuration Information" tree.



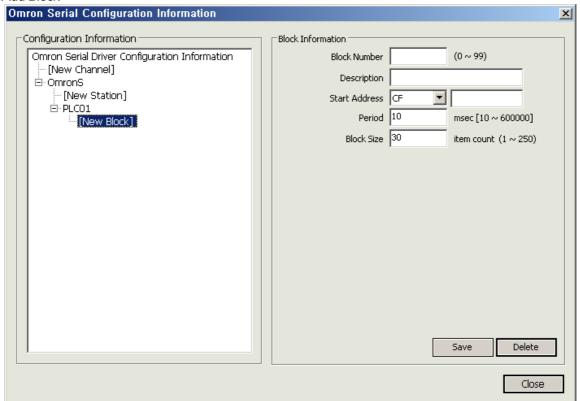
(2) Add Station



- (a) Select [New Station] from "Configuration Information" tree.
- (b) Station Name: Input a station name.
- (c) Description: Input some information on the station.
- (d) PLC CPU Type: Select a PLC CPU type.
- (e) Station Number: Input the number of PLC Cnet Module.
- (f) Save: If 'Save' button is pressed, Station information will be saved and the saved information will add to the left "Configuration Information" tree.



(3) Add Block



- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: Input the Block's Start Address.
 - -.Start Address : Memory Type (CIO,WR,HR,AR,TF,CF,EM0)+ Start Address
 - -. Block Size : A number of items to be loaded

Example 1) Start Address: C1010, Block Size: If it is 100, 100 words will be loaded starting from the 10th word of CIO memory (starting from 0)

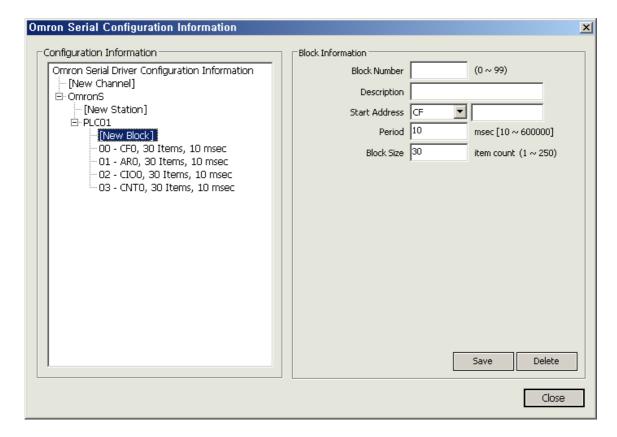
Example 2) Start Address: TF20, Block Size: If it is 20, 20 status bits will be loaded starting from the 20th status bit of TF memory (starting from 0)

Caution) CIO,LR,HR,AR,TIM,CNT,DM,EM,EM0,EM1,EM2,EM3,EM4,EM5,EM6,EM7,EM8,EM9,EMA,EMB AND EMC ARE DEVICES IN WORD UNIT.

Caution) TF and CF are devices in Bit unit.

Caution) If there are more than 30 devices in WORD unit and more than 120 devices in Bit unit, they can be read with multi frames.

- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Block Size: A number to read for each corresponding delimiter.
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.



(4) I/O Address

(a) Type

-. Example of Analog Input Address

CIO_100 : Load the value of the 100th word in CIO Area
EM7_20000 : Load the value of the 20000th Word in EM7 Area

-. Example of Digital Input Address

CIO_90A : Load the 10(A)th bit value of the 90th word in CIO Area EM7_07 : Load the 7th bit value of the 0th word in EM7 Area

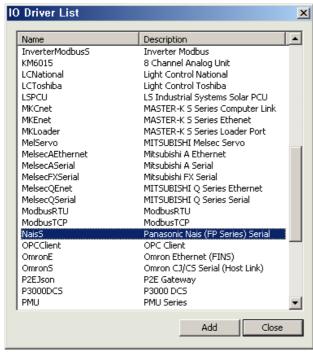
TF_123 : Load the completion flag value of the timer in Timer Flag Area

-. Omron PLC Memory Device Map

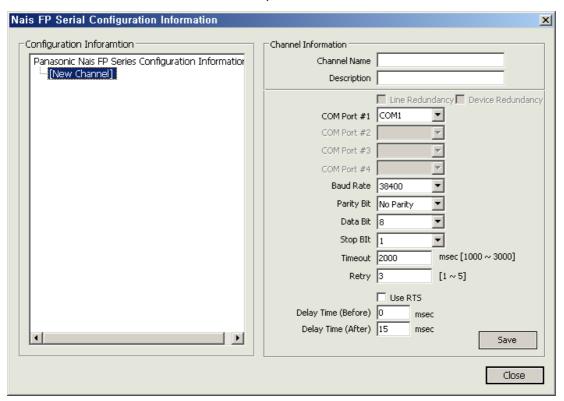
Smith Lo Memory Bevice Map									
Memory COde	Description	Data Type	CS/CJ CPU Memory Address	CV CPU Memory Address	Remarks	Address Example			
CIO	CIO Area		0~6143	0~2555		AI: CIO_xxxx (xxxx: 0~6143) CIO_100 DI: CIO_xxxx0~F(xxxx: 0~6143) CIO_100A			
LR	CIO1000~ CIO1199	WORD	0~199	_		AI:WR_xxxx (xxxx:0~511) WR_100 DI:WR_xxxx0~F(xxxx:0~511) WR_0F			
HR	Holding Bit Area		0~511			AI: HR_xxxx (xxxx: 0~511) HR_234 DI: HR_xxxx0~F(xxxx: 0~511) HR_19			
AR	Auxiliary Bit Area		0~959	0~959	0~447(읽 기전용), 448~959 (읽기/쓰 기)	AI: AR_xxxx (xxxx: 0~511) AR_800 DI: AR_xxxx0~F(xxxx: 0~511) AR_9001			
TF	Timer Area	Completi	0~2047	0~2047 or 0~1023	읽기전용	DI: TF_xxxx(xxxx: 0~4095), TF_2047			
CF	Counter Area	on Flag(BIT)				DI : CF_xxxx(xxxx : 0~4095), CF_2048			
TIM	Timer Area PV		0~2047	0~2047 or 0~1023		AI: TIM_xxxx (xxxx : 0~4095) TIM_1000			
CNT	Counter Area PV					AI: CNT_xxxx (xxxx: 0~4095) CNT_2000			
DM	DM Area		0~9999	0~9999		AI: DM_xxxxx (xxxxx: 0~32767) DM_10000 DI: DM_xxxxx0~F(xxxxx: 0~32767) DM_900E			
EM0~EM7	EM Area	WORD		0 3333		AI: EM0_xxxxx (xxxxx: 0~32767) EM0_9000 DI: EM0_xxxxx0~F(xxxxx: 0~32767) EM0_7000A			
EM7~EMC				_		AI: EMA_xxxxx (xxxxx: 0~32767) EMA_9000 DI: EMA_xxxxx0~F(xxxxx: 0~32767) EMA_7000A			
EM				0~9999		AI: EM_xxxxx (xxxxx: 0~32767) EM_9000 DI: EM_xxxxx0~F(xxxxx: 0~32767) EM_7000A			

7.22 NaisS

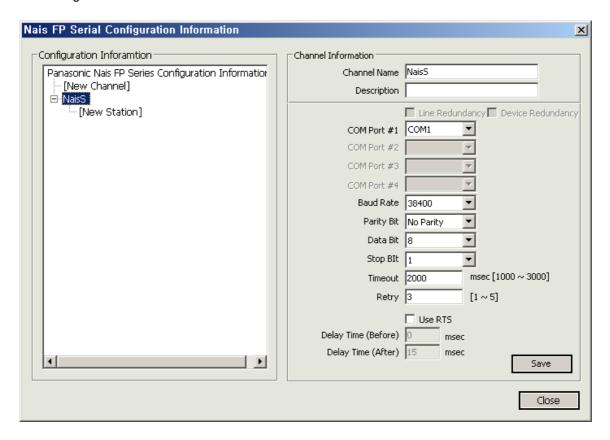
(1) Add Channel



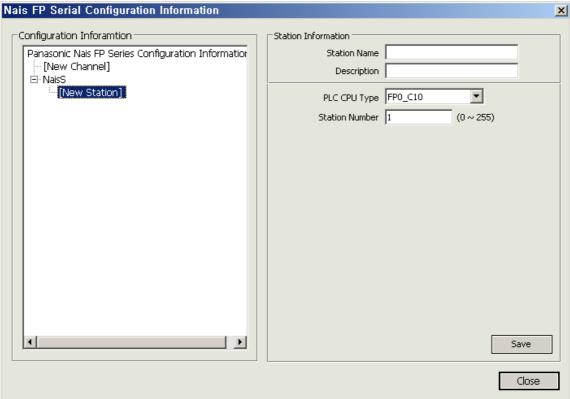
SELECT "OmronS" from the I/O driver list and press "Add."



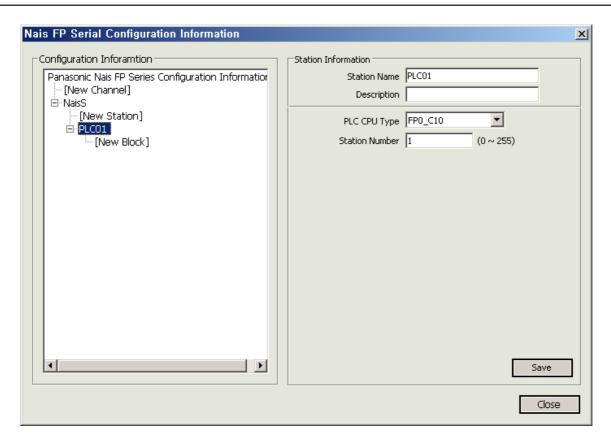
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) COM Port #1: Select a serial port of PC.
- (d) Baud Rate: Select a communication speed.
- (e) Parity Bit: Select a parity bit.
- (f) Data Bit: Select a data bit.
- (g) Stop Bit: Select a stop bit.
- (h) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (i) Retry: Set up the number of times to retry when communication fails.
- (j) Use RIS: Check $\sqrt{}$ in the box only when using Channel 485 and 422.
- (k) Delay Time (Before): Information used only in Channel 485 and 422. The delay time right before requesting Data to PLC
- (I) Delay Time (After): Information used only in Channel 485 and 422. The delay time right after requesting Data to PLC
- (m) Save: If 'Save' button is pressed, Channel information will be saved and the saved information will add to the left "Configuration Information" tree.



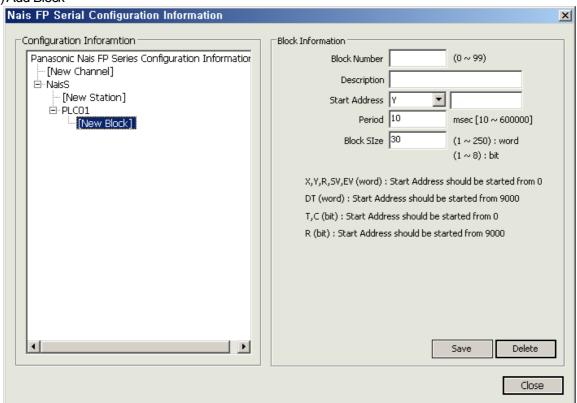
(2) Add Station



- (a) Station Name: Input a station name.
- (b) Description: Input some information on the station.
- (c) PLC CPU Type: Select a PLC CPU type.
- (e) Station Number: Input the number of PLC Cnet Module.
- (e) Save: If 'Save' button is pressed, Station information will be saved and the saved information will add to the left "Configuration Information" tree.



(3) Add Block



- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: Input the Block's Start Address.
 - -. For X,Y,R,DT,SV,EV,IX and IY, use reading in WORD unit.
 - Ex 1) X, Y and R are memories in Bit unit but, for them, reading in Word unit is also used. For X input, 208 points can be used for X0 through X12F.

Start Address: X1, Block Size: If 5 is inputted, it will load data of X0 through X5F.

Ex 2) EV and SV are memories in Word unit. Therefore, reading in Word unit is used.

Start Address: EV0, Block Size: If 20 is inputted, it will load data of EV0 through EV19.

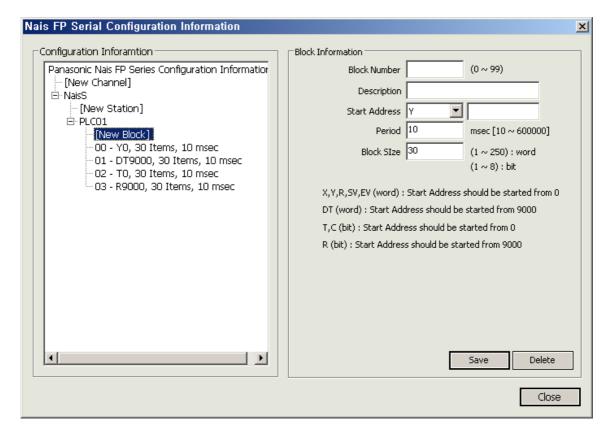
Ex 3) IX and IV are momories in Word unit but, they exist in one word that is, there are only 1X0 and 1Y0.

Start Address: 1X0, Block Size: select 1. For IY, also select 1.

Ex 4) DT are memories in Word unit but, they exist only DT9000 through DT9001.

Start Address: DT9000, Block Size: If 10 is inputted, it will load data of DT9000 through DT9009.

- -. For more than R9000 ,T and C use reading in Bit unit. Their Start Address are in Bit unit and one number shall be selected 1 through 8 for Block Size.
 - Ex 1) Start Address: T0, Block Size: If 8 is inputted, it will load data of T0 through T7.
 - Ex 2) Start Address: R9000, Block Size: If 8 is inputted, it will load data of R9000 through R9007.
- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Block Size: A number to read for each corresponding delimiter.
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.



(4) I/O Address

(a) Type

-. Example of Analog Input Address

DT100 : Load the value of the 100th word in DT area.

SV10 : Load the setting value of T10 in Timer area.

SV111 : Load the setting value of C111 in Counter area.

-. Example of Digital Input Address

X10F : Load the 175th contact value equal to 10*16+Fh (15) = 160+15 in X area. R501 : Load the 801st contact value 50*16+1h (1) = 800+1 in Internal Relay R area.

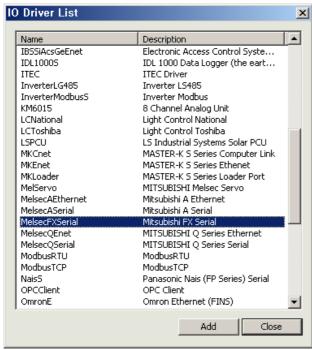
R9001 : Load the contact value of Special Internal Relay R9001. DT1280F : Load the F (15th) bit value of Data Register 1280 words.

-. Nais PLC Memory Device Map

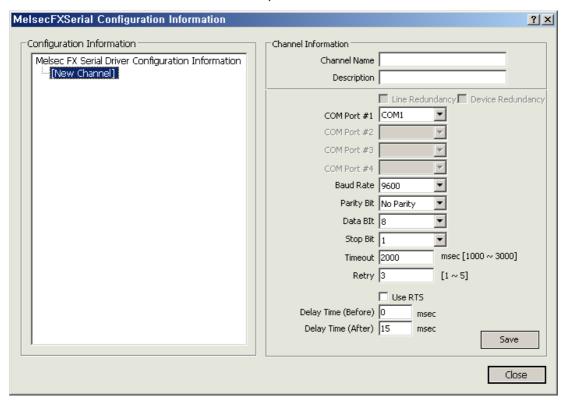
Memory Code	Description	Data Type	C10/C14/C16 CPU Memory Address	C32/SL1 CPU Memory Address	Address Example		
Х	Extenal Input		208 Point(X	(0~X12F)	DI : Xxxxx0~F(xxxx : 0~12) X12A		
Y	Extenal Output		208 Point(Y	′0~Y12F)	DI : Yxxxx0~F(xxxx : 0~12) Y0F		
R	Intenal Relay	BIT	1008 Point(F Special Internal I (R9000~I	Relay 64 Point	DI: Rxxxx0~F(xxxx:0~62) R19F (Special) DI: Rxxxx (xxxx: 9000~9063) R9010		
Т	Timer		T0~T	⁻ 99	DI: Txxxx (xxxx: 0~99) T90		
С	Counter		C100~(C143	DI : Cxxxx(xxxx : 100~143), C108		
DT	Data Register		1660 Word (DT0~DT1659)	6144 Word (DT0~DT6143)	AI: DTxxxx (xxxx: 0~1659 or 0~6143) DT1000 DI: DTxxxx0~F (xxxx: 0~1659 or 0~6143), DT108A		
SV	Timer/ Counter Setting Area	WORD	144 Word(S\ Timer : 0~99, Cou	,	AI: SVxxxx (xxxx: 0~143) SV1 or SV100		
EV	Timer/ Counter Value Area	WORD	144 Word(E\ Timer : 0~99, Cou	,	AI : EVxxxx (xxxx : 0~143) EV40 or EV129		
IX	Index Register		1 Word	(IX0)	AI : IX0 (1 Word Only) DI : IX_0(0~F) IX0F		
IY	Index Register		1 Word	(IY0)	AI : IY0 (1 Word Only) DI : IY_0(0~F)		

7.23 MelsecFXSerial

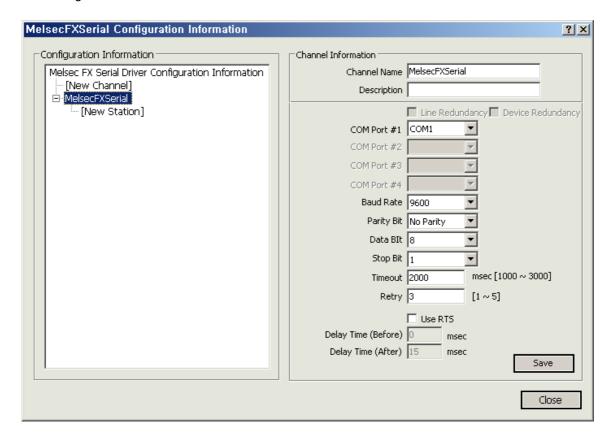
(1) Add Channel



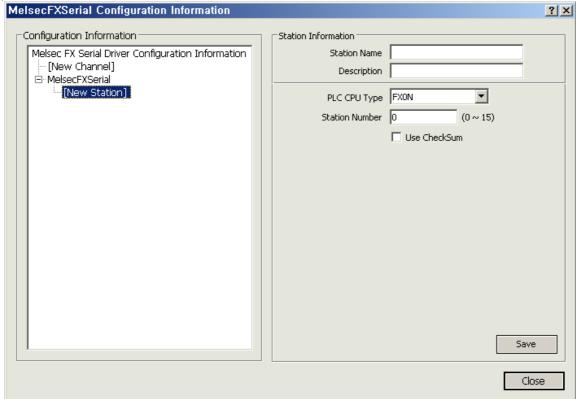
SELECT "OmronS" from the I/O driver list and press "Add."



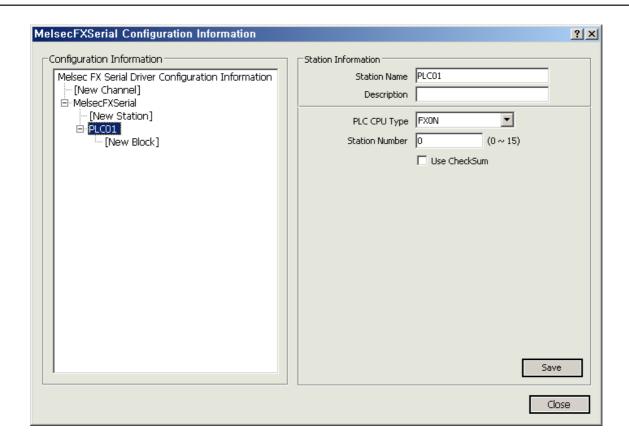
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) Baud Rate: Select a communication speed.
- (d) Parity Bit: Select a parity bit.
- (e) Data Bit: Select a data bit.
- (f) Stop Bit: Select a stop bit.
- (g) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (h) Retry: Set up the number of times to retry when communication fails.
- (j) Use RIS: Check $\sqrt{}$ in the box only when using Channel 485 and 422.
- (k) Delay Time (Before): Information used only in Channel 485 and 422. The delay time right before requesting Data to PLC
- (I) Delay Time (After): Information used only in Channel 485 and 422. The delay time right after requesting Data to PLC
- (I) Save: If 'Save' button is pressed, Channel information will be saved and the saved information will add to the left "Configuration Information" tree.



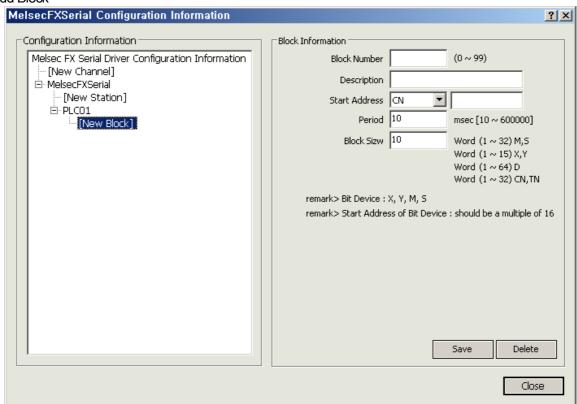
(2) Add Station



- (a) Select [New Station] from "Configuration Information" tree.
- (b) Station Name: Input a station name.
- (c) Description: Input some information on the station.
- (d) PLC CPU Type: Select a PLC CPU type.
- (e) Station Number: Input the number of PLC Cnet Module.
- (I) Save: If 'Save' button is pressed, Station information will be saved and the saved information will add to the left "Configuration Information" tree.



(3) Add Block



- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: Input the Block's Start Address.
 - -X, Y, M and S are device memories in Bit unit and load reading in word unit.
 - Ex 1) X and Y are Bit memories in octal unit. Their addresses are expressed in X10 after X7. "Start Address" shall be multiples of 8 because of they are octal. That is, XO (Oth bit), X10 (8th bit), X20 (18th bit), X30 (24th bit)...

In addtion, Block Size is in Word unit. If the user wants to load 16 points, he/she shall input 1.

If X0 is set up for "Start Address" and 2 is inputted for "Block Size," this system will load 32Bit of X0~X7, X10~X17, X20~X27 and X30~X37.

Ex 2) M and S are Bit memories in decimal unit and their addresses are expressed in M10 after M9. "Start Address" shall be multiples of 8 because of they are decimal. That is, MO (Oth bit), M8 (8th bit), M16 (18th bit), M24 (24th bit)...

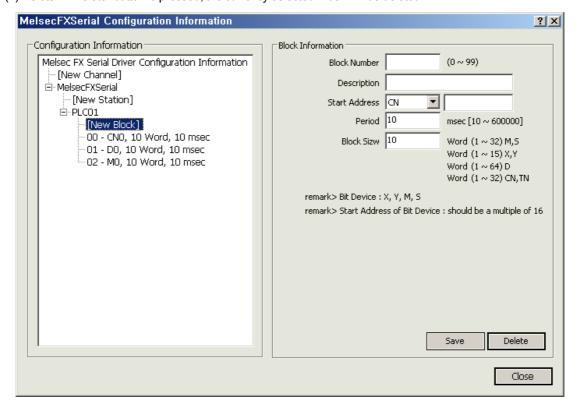
In addition, Block Size is in Word unit. If the user wants to load 16 points, he/she shall input 1.

If M0 is set up for "Start Address" and 2 is inputted for "Block Size," this system will load 32Bit of M0 through M31.

-D. T.C. CN and TN are device memories in Word unit.

If D10 is set up for "Start Address" and 10 is inputted for "Block Size," this system will load 10 word values D10 through D19.

- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Block Size: A number to read for each corresponding delimiter.
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.



(4) I/O Address

(a) Type

-. Example of Analog Input Address

D100 : Load the value of the 100th word (starting from O) in D area
TN6 : Load the 6th current value (starting from O) in Timer area
CN10 : Load the 10th current value (starting from O) in Counter area

-. Example of Digital Input Address

X7 : Load the 7th bit value (starting from O) of Input Relay
 Y10 : Load the 10th bit value (starting from O) of Output Relay

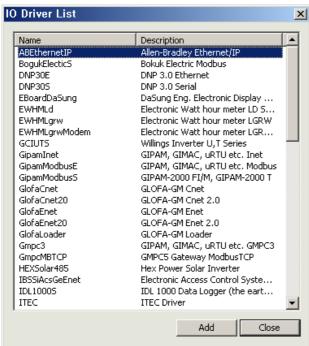
M10 : Load the 100th bit value (starting from O) of Supplementary Relay

-. MelsecFX Serial PLC Memory Device Map

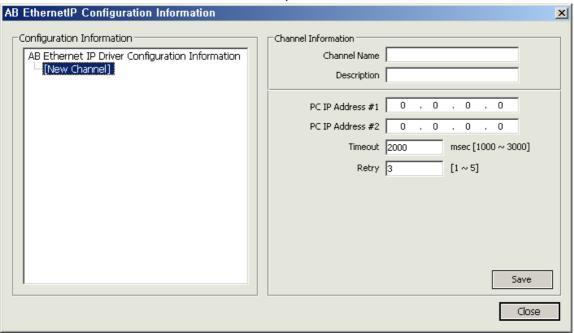
Device Type			Device Number Range (Default)					Expression			Address
Bit	Word	Device Code	FX1S	FX0N	FX2, FX2C	FX1NC, FX1N	FX2N, FX2N C	Hex Decimal	Decimal	Remarks	Example
•		Х	X0000 ~ X0017	X0000 ~ X0177	X0000~ X0337	X0000~ X0177	X0000 ~ X0267	•		Input Relay	X0,X07, X007,X 0007
•		Y	Y0000 ~ Y0015	Y0000 ~ Y0177	Y0000~ Y0337	Y0000~ Y0177	Y0000 ~ Y0267	•		Output Relay	Y0,Y10, Y110,Y 0110
•		М	M0000~M0511		M0000-	-M1535	M0000 ~ M3071		•	Sub Relay	M0,M04 ,M111, M1234
•		S	S0000~S127		S0000~S0999			•	State	S1,S50, S200,S 0005	
•		M	M8000~M8254		M8000~M8255				•	Special Sub Relay	-
	•	TN	TN000~TN063		TN000~TN255			•	Timer Value	TN0,TN 01,TN0 01	
	•	CN	CN000~CN031, CN235~CN254		CN000~CN255				•	Counter Value	CN6,C N10,CN 100
	•	D	D0000~D0255		D0000~ D0999	D0000~ D7999			•	Data Register	D1,D10, D100,D 1000
	•	D	D1000 - ~ D2499		D1000~ D2999	-		_	•	File Register	-
	•	D	-		D6000~ D7999				•	RAM File Register	-
	•	D	D8000~D8255		D8	D8000~D8255			•	Special Data Register	-

7.24 **ABEthernetIP**

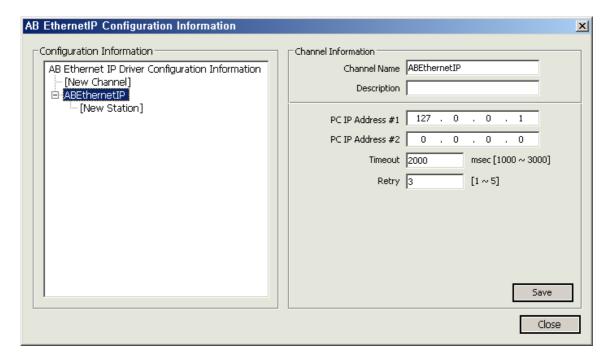
(1) Add Channel



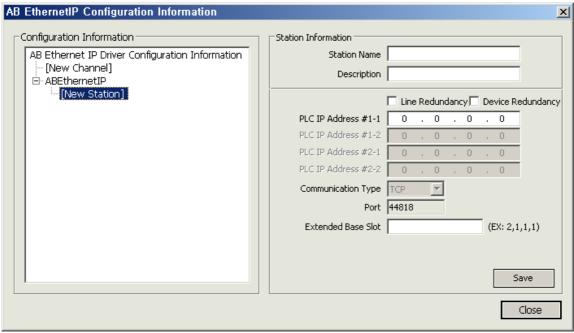
SELECT "ABEthemetIP" from the I/O driver list and press "Add."



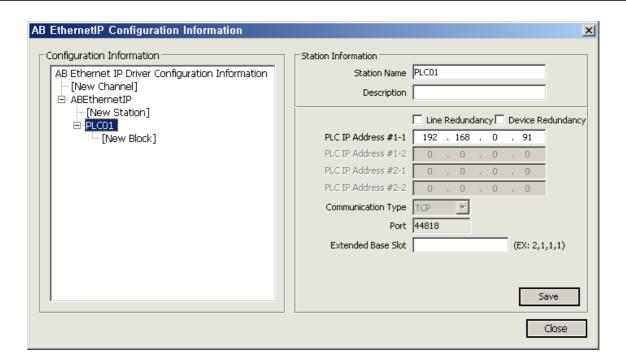
- (a) Channel Name: Input a channel name.
- (b) Description: Input some information on the channel.
- (c) Serve IP: Input PC's IP Address.
- (d) Server IP Address #2: If Line Redundancy will be used, input the second IP Address to be used.
- (e) Time Out: It refers to a certain time period during which any response to the request for data is not made and after passing such a time period, the system will declare timeout to move on to the next process. The time period to be set will be a base to judge communication errors.
- (f) Retry: Set up the number of times to retry when communication fails.
- (g) Start Offset Address: Select either 0 or 1. It sets up whether to start from either 0 or 1.
- (h) Save: If 'Save' button is pressed, Channel information will be saved and the saved information will add to the left "Configuration Information" tree.



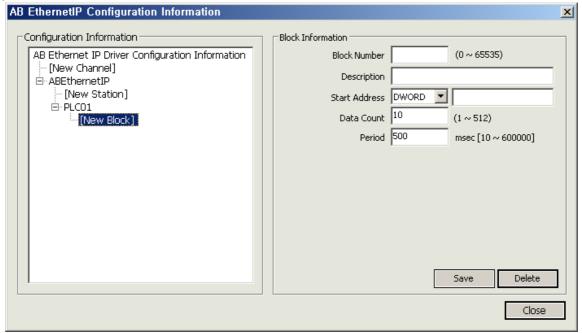
(2) Add Station



- (a) Select [New Station] from "Configuration Information" tree.
- (b) Station Name: Input a station name.
- (c) Description: Input some information on the station.
- (d) Line Redundancy: Check √in the box to use Line Redundancy.
- (e) Device Redundancy: Check √in the box to use Device Redundancy.
- (f) PLC IP Address #1-1: Input PLC's IP Address.
- (g) PLC IP Address #1-2: Input PLC's IP Address. Input the address when using Device Redundancy.
- (h) PLC IP Address #2-1: Input PLC's IP Address. Input the address when using Device Redundancy.
- (i) PLC IP Address #2-2: Input PLC's IP Address. Input the address when using Line Redundancy along with Device Redundancy.
- (j) Communication Type: TCP. Fixed
- (I) Port: For EthernetIP, 44818 Port is fixed.
- (m) Extended Base Slot: Input the number of the extended base slot if the slot on which EtherNet/IP module is mounted is attached to an extended base.



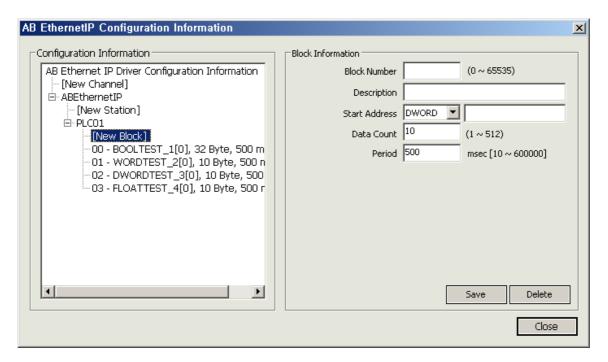
(3) Add Block



- (a) Select [New Block] from "Configuration Information" tree.
- (b) Block Number: This number is a unique code of the block. The user needs to designate a different code to each block.
- (c) Description: Input some information on the block.
- (d) Start Address: For 'Start Address' of EtherNEt/IP's block, the tag name registered on PLC shall be used as its address.
 - If 10 INT-type data are assigned to Array 0 through 10 with INR_A_TAG names on PLC, they shall be have Start Addressess of WORDL type INT_A_TAG[0] and data number 100.
 - If 10 BOOL-type data are assigned to Array 0 through 10 with BOOL_A_TAG names on PLC, they shall be have

Start Addressess of BOOL type BOOL A TAG[0] and data number 100. 'Start Address' of a BOOL-type tag shall be always multiples of 32.

- (e) Period: Input an interval to collect data of the relevant block (unit: msec).
- (f) Data Count: A number to read for each corresponding delimiter.
- (g) Save: If 'Save' button is pressed, Block information will be saved and the saved information will add to the left "Configuration Information" tree.
- (h) Delete: If "Delete" button is pressed, the currently selected Block will be deleted.

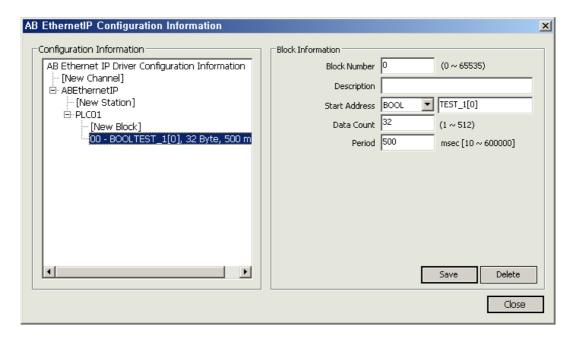


(4) I/O Address

- (a) Type
 - Tag Name
 - Tag Name[Array Number]
 - Tag Name[Array Number].0 ~ Tag Name[Array Number]. 15
 - Tag Name[Array Number].0 ~ Tag Name[Array Number]. 31
 - Remote1:1:1. Data.0~ Remote1:1:1. Data.31

-. Block setting and I/O address input for BOOL Array Type

If 32 BOOL-type tags starting from Array TEST_1[0] are set on RS Logix 5000, the following setting shall be made on InfoU. If it loads the value of PLC's BOOL Array-type TEST_2[32], the following values shall be entered for the relevant channel block.



* [Caution] Start Address shall be multiples of 32.

Ex) TEST_1[0], TEST_1[32], TEST_1[512]

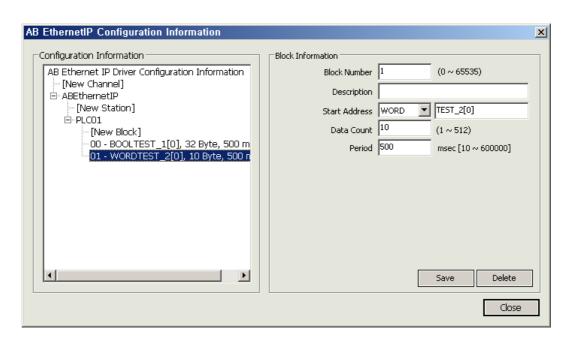
Input the following tag name [Array Number] when inputing database on InfoU.

Tag Type : Digital Tag BOOL Type

Address : TEST_1[0], TEST_1[1], TEST_1[2], ... TEST_1[31]

-. Block setting and I/O address input for INT Array Type

If 10 INT-type tags starting from Array TEST_2[0] are set on RS Logix 5000, the following setting shall be made on InfoU. If it loads the value of PLC's INT Array-type TEST_2[10], the following values shall be entered for the relevant channel block.



Input the following tag name [Array Number] when inputting database on InfoU.

Tag Type : Analog

Data Type : INT16, UINT16

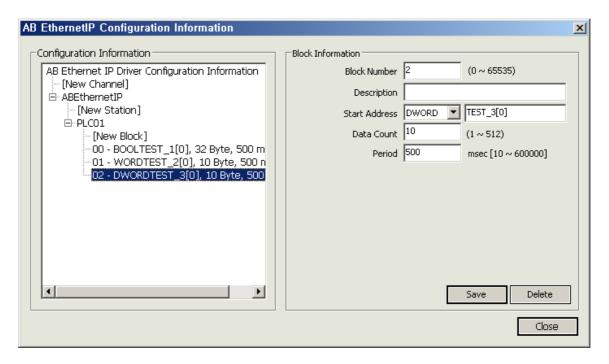
Address : TEST_2[0], TEST_2[1], ... TEST_2[9]

Tag Type : Digital

Address : TEST_2[0].0, TEST_2[0].1, ... TEST_2[0].15

-. Block setting and I/O address input for DINT Array Type

If 10 DINT-type tags starting from Array TEST_3[0] are set on RS Logix 5000, the following setting shall be made on InfoU. If it loads the value of PLC's DINT Array-type TEST_3[10], the following values shall be entered for the relevant channel block.



Input the following tag name [Array Number] when inputting database on InfoU.

Tag Type : Analog

Data Type : INT32, UINT32

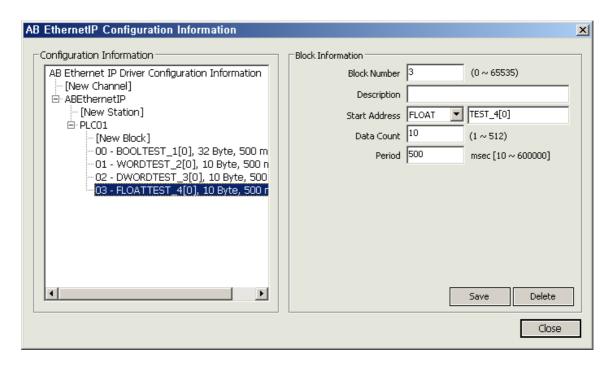
Address : TEST_3[0], TEST_3[1], ... TEST_3[9]

Tag Type : Digital

Address : TEST_3[0].0, TEST_3[0].1, ... TEST_3[0].31

-. Block setting and I/O address input for REAL Array Type

If 10 real-type tags starting from Array TEST_4[0] are set on RS Logix 5000, the following setting shall be made on InfoU. If it loads the value of PLC's ReaL Array-type TEST_4[10], the following values shall be entered for the relevant channel block.



Input the following tag name [Array Number] when inputting database on InfoU.

Tag Type : Analog Data Type : Float

Address : TEST_4[0], TEST_4[1], ... TEST_4[9]

Chapter 8 Tag Management

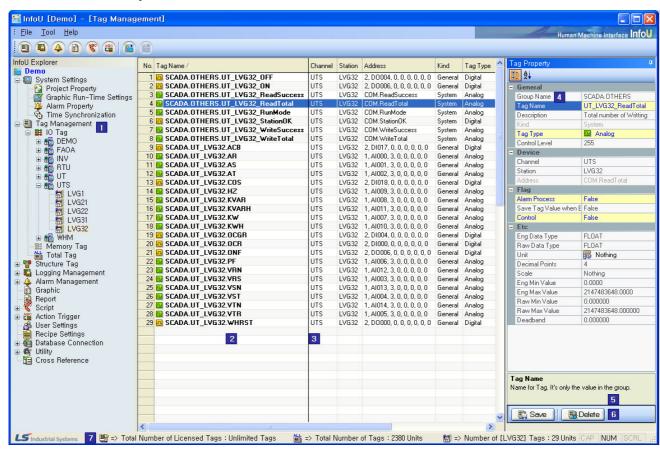
8.1 Screen Composition

This function is to register and manage the tag, a basic element of HMI system. All of the functions supported and operated in HMI system depend on the tag registered in the tag DB.

The tag can be classified into I/O tag that will be handled with signals received from the device in the field and memory tag that will be virtually registered for needs of the upper level system.

InfoU clearly shows information on the tag in the property window at a glance to allow the user to perform tag engineering works conveniently. In/output of CSV files is also supported to enable to register and manage a large amount of tag data quick and handy.

8.1.1 Screen Composition



Chapter 8 Tag Engineering

The tag DB screen consists of the following;

(1) Tag Structure Tree

Tag components form a tree-type structure to add more tags efficiently.

In addition, for more effective tag management, the following two types of view are supported simultaneously: "View by device" through which tags are arranged in a physical structure, showing how devices are actually connected and "View by group" through which tags are arranged in a logical structure, showing which structure the user wants to form.

Basically "View by device" is presented as default value but, the user can change to "View by group" by clicking the right button of the mouse.

(2) Tag List

Once a group is chosen in the virtual tag structure tree, a list of tags that belong to the selected group will appear and the user can arranged the list in order he/she wants by clicking one of column titles.

(3) Tag Frame Anchor

If the list is so long that other data can not be seen, this function is used to fix a frame in a location where the user wants to see and use other data.

To use this function, locate the mouse cursor on the tag frame anchor and the mouse cursor will be changed to a lock shape. At this time, the user can move the mouse to a location he/she wants by dragging and dropping the frame anchor.

As default, the frame is fixed in the boxes for tag name and other property names.

(4) Tag Property

This screen shows the tag properties selected from the list.

This property window shows the entire properties in one screen to allow the user to access to those in a very convenient way.

(5) Tag Property Help

When a focus moves from one tag property to the other tag property, a brief explanation is automatically displayed on the corresponding property.

When editing, this 'help' function helps the user to edit the relevant tag property easily without popping up other additional help screen.

(6) Save and Delete

The user can save or delete information related to tags.

But, if a tag is selected and some of its properties are changed, the tag will have the changed properties unless the tag name is not changed. Contrary to this, a new tag will be added if the tag name is changed, This is helpful when the user wants to create multiple tags with similar properties because the user can do

so only by change their names to make many additional tags for efficient engineering performance.

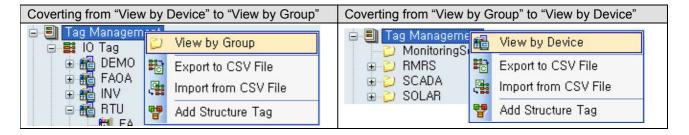
(7) License Information

This screen briefly shows information on the current license.

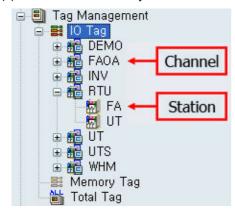
It displays the validity of the license and the number of available tags under the license. Also, it shows the total number of tags and the tag number of devices or stations selected by the current user.

8.1.2 **Convert Tag View**

"Convert Tag View" presents a tree-type tag configuration to allow the user to manage tags more easily. Click the right side of the mouse on Tag Management to show a pop-up menu. Through this menu, "View by Group" can be converted to "View by Device" and vice versa. As default, "View by Device" is seen first.



(1) Tree structure of View by Device



A tree structure of View by Device is a method used to show how tags are physically connected to devices. It consists of the following sub trees:

(a) IO Tag

A tag actually connected to I/O point of the device in the field. As its sub items, there are channel and station.

(b) Channel

It refers to the type of the channel to be communicated and the selected channel appears among supporting IO drivers.

When selecting, a list of tags that belong to the relevant channel is displayed.

(c) Station

It refers to an actual device such as PLC or Inverter as sub-concept of the channel. In selecting, a list of tags that belong to the relevant station is displayed.

(d) Memory Tag

A tag virtually created to operate in the upper-level system. Actually, it is not collected from the device in the field. When this is selected, a list of the entire memory tags is seen.

(e) Total Tag

It is an Item for managing the entire registered tags.

It refers to the sum of IO tags and memory tags.

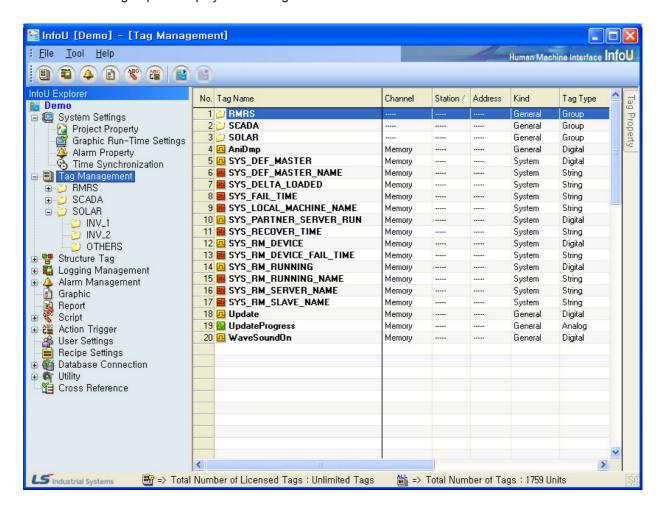
Chapter 8 Tag Engineering

(2)Tree structure of View by Group



Contrary to View by Device, the tree structure of View by Group shows tags by logically organizing groups. Each group may have sub groups and also enable to create logical groups.

If one group is selected, the tags that belong to the selected group along with the sub groups of the selected group are displayed in the tag list screen.



8.2 **Channel Device Settings**

The user can add, delete or modify channels and stations through pop-up menu if each device is displayed in a tree structure. This section briefly explains how to edit each item.

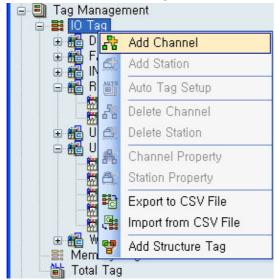
For properties and detailed instructions to use each channel and station, see "I/O Driver" manual.

8.2.1 Channel

It refers to a channel to be communicated and selected among supporting I/O Drivers. Also, it basically provides a "memory" type of channels to classify into I/O tags and memory tags in InfoU.

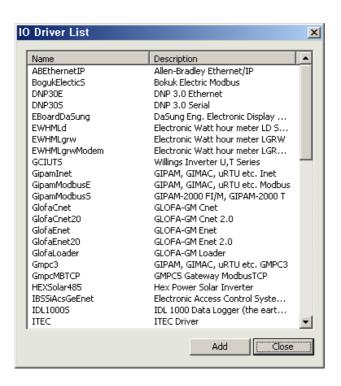
(1) Setting

To add a channel, click the right mouse on the I/O tag list and select "Add Channel" menu.

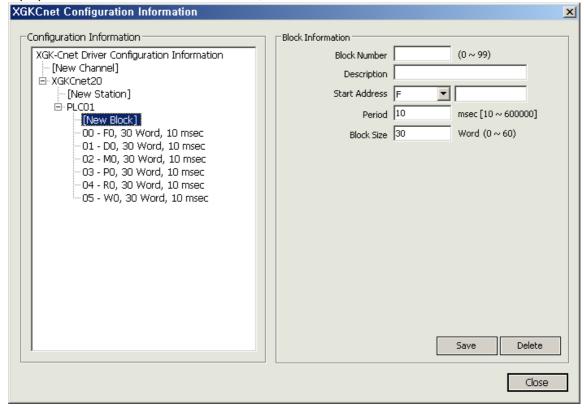


If "Add Channel" menu is selected, a list of available I/O drivers is displayed.

Chapter 8 Tag Engineering

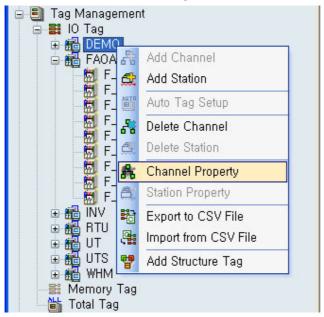


To add a channel, select one and click "Add" menu to show a screen in which the user can edit channel properties of the channel. For more details, refer to each item of I/O Driver.



(2) Edit

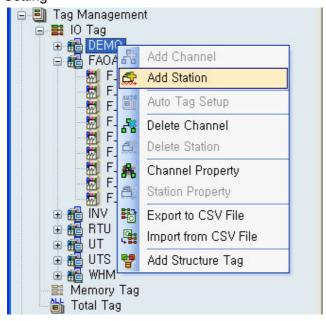
To edit a channel, click the right mouse on the relevant channel to select a menu.



8.2.2 Station

A station refers to an actual device such as PLC or inverter, which belongs to the upper-level channel. Its way to unfold is similar to that of the channel. For more details, refer to "I/O Driver" manual.

(1) Setting



8.2.3 Delete Channel

Deleting a channel is possible only after deleting all of the tags and stations that belong to the lower level. This also applies to stations. That is, deleting is possible only if there is no item in the lower level. If the user wants to delete a channel, delete Tag -> Station -> Channel in order.

8.3 **Tag Property**

Default tags in the InfoU system have very various properties.

However, each tag's properties are limited depending on which category among the followings the tag belongs to and which combination it has:

- Channel (IO tag/Memory Tag)
- Tag type (Analog / Digital / String)

Also, each tag may or may not define its detailed properties depending on activation/inactivation options such as alarm process and expression.

InfoU shows available items to be set according to the channel and tag type and also displays explanation on the property to be inputted in the" Help" screen automatically to allow the user to define the tag easily.

The following figure is an example of the tag property screen.

Each property belongs to a small group such as [General], [Device], [Flag] and [Etc] according to its characteristics and each group can be unfolded or hid by clicking the mouse on '- 'or '+ ' icon next to the small group title.

Also, the items that can not editable by the user such as [Kind] are displayed in gray color. Especially, important properties for engineering are highlighted with blue letters and yellow background.

The changed properties are displayed in bold to easily distinguish them from unchanged ones.



Each property is explained as follows:

8.3.1 General

As the most basic properties, they can be commonly used for all kinds of tags.

Property Name	Description				
Group Name	Name for managing a tag as a logical group				
Tag Name	Tag name. Each tag name has its own unique value in the group.				
	The length of a tag name can not exceed 95 Korean letters/190 alphabet letters.				
	Here, the length of a tag name means the entire name including its group name.				
	When defining a tag name, the following rules are applied.				
	Length is limited to 95 Korean letters/190 alphabet letters.				
	Any tag name can not start with number nor include blank.				
	With some exception, special letters can not be used.				
	• Allowed letters : # \$ [] _ { }				
Description Explanation on the relevant tag.					
	Its length is limited to 127 Korean letters/254 alphabet letters.				
IZ: a al	Vind of the velocient to a				
Kind	Kind of the relevant tag.				
	Tags can be divided into general tags created by the user during the tag engineering				
T T	and system tags needed for managing device status.				
Tag Type	Tags can be divided into Analog, Digital, String and Group according to the				
	characteristics of tag value.				
Control Level	Each tag has a control level from 1 to 255 depending on its importance from the				
	aspect of tag control.				
	Level 1 has the first priority.				

8.3.2 Device

A collection of tag's properties related to I/O. The channel is set up for I/O tags and memory tags while the station and I/O address are activated only for I/O tags.

Property Name	Description
Channel	A channel is set up to communication with tags.
	The user may select a channel that has I/O driver kind and memory type
	registered in the InfoU system.
Station	As sub-concept of the channel, an actual device such as PLC or Inverter
	can be set up.
Address	Input an actual tag's communication address in the relevant station.
	How to set up an address varies in each device. For details, refer to
	"I/O Driver" manual.

8.3.3 Flag

A collection to set up tag's property and operation to be performed during the runtime

Property Name	Description				
Alarm Process	The user may decide whether to perform alarm process for a certain tag value. If this property is activated, properties to set detailed alarm characteristics are presented.				
Write the initial value	The user may decide whether to apply the initial value to the site device when starting Runtime. This property is valid only for I/O tags.				
Save Tag Value when Exit	The user may decide whether to save the last value of the tag in the database when ending Runtime. If this property is activated, the tag value is saved when ending the project and this saved value is used as the initial value when starting the project.				
Control	The user may decide whether to enable tag control. If this property is cancelled, control can not be executed even a control order is made.				
Show Real Data Reverse	The user may decide whether to reverse the digital value downloaded from the site. This property belongs to I/O tag and it can be set up only when the tag type is digital.				
Memory Tag Expression	In case of memorial tags, the user may decide whether to perform the expression.				

8.3.4 Analog Value

A collection of properties any tag can have only when its type is analog.

Property Name	Description					
Eng Data Type	Type for tag's industrial value seen from the upper-level system.					
	The following val	The following values are available to be selected.				
	• INT_8	: 8bit inte	egral nur	mber with sign		
	• INT_16	: 16bit in	tegral nu	ımber with sigr	1	
	• INT_32	: 32bit in	tegral nu	umber with sigr	1	
	• UNSIGNED IN	Γ_8	: 8bit int	tegral number v	vithout sign	
	UNSIGNED INT 16 : 16bit integral number without sign					
	UNSIGNED INT_32 : 32bit integral number without sign					
	• FLOAT : 4bit actual number					
	• DOUBLE		: 8bit ac	tual number		
Raw Data Type	Type of tag values collected by the device					
	The following values are available to be selected.					
	• INT 8 • INT 16 • INT 32 • IN				• INT_64	
	• UINT 8 • UINT 16 • UINT 32 • UINT 64				• UINT_64	
	• FLOAT	• DOUB	LE			

	•BCD_8 •BCD_16 •BCD_32 •BCD_64				
	• UBCD_8 • UBCD_16 • UBCD_32 • UBCD_64				
Initial Value	The tag's initial value when a project starts. But, be cautious that if the 'save' option is activated when the project ends, the initial value will disregarded and the last value of the saved tag during the ending will be an actual initial value.				
Unit	Unit for industrial value.				
Decimal Points	If data type is [Double] or [Float], decimal points are displayed. As default, four digits are set up at first but the user may extend up to 10 digits.				
Scale	Rule used to convert tag's device value to industrial value. The following two ways are used. • Ratio / Bias : A way to use slope/section Industrial Value = Ratio X Device Value + Bias • Ratio : A way to convert by using the min/max value of the device value and industrial value				
Ratio	A value by which the device value is multiplied. It becomes activated when 'Ratio/Bias' is selected for scale.				
Bias	A value to be added to the sum which is multiplied by ratio for the device value. It becomes activated when 'Ratio/Bias' is selected for scale.				
Eng Min Value	Tag's min value for industrial value.				
Eng Max Value	Tag's max value for industrial value.				
Raw Min Value	Minimum value for tag's device value.				
Raw Max Value	Maximum value for tag's device value.				
Deadband	If the change in values is fine, the user may set a deadband to prevent unnecessary processing. Default value is 0.				
Memory Operation Interval	In case of a memory tag, it is an interval (sec) to perform expression. In case that "0" is set, it operates in an event way and if higher than 0 is inputted, it performs in a regular interval. To perform the relevant tag expression automatically according to the input of tag value, this value should be set up as "0." When a regular interval is operated, the expression is processing only for the corresponding tag.				
Memory Expression	An expression for the memory tag is set up.				

8.3.5 Digital Value

A collection of properties any tag can have only when its type is digital.

Property Name	Description
Eng Data Type	Type for tag's industrial value seen from the upper-level system.
	In case of digital tag, it is fixed as "bool" value.
Raw Data Type	Type of tag values collected by the device
	In case of digital tag, it is fixed as "bool" value.
Initial Value	The tag's initial value when a project starts.
	But, be cautious that if the 'save' option is activated when the project ends, the initial
	value will disregarded and the last value of the saved tag during the ending will be an
	actual initial value.
On String	String displayed when tag value is ON.
Off String	String displayed when tag value is OFF.
Memory Operation	In case of a memory tag, it is an interval (sec) to perform expression.
Interval	In case that "0" is set, it operates in an event way and if higher than 0 is inputted, it
	performs in a regular interval.
	To perform the relevant tag expression automatically according to the input of tag value,
	this value should be set up as "0."
	When a regular interval is operated, the expression is processing only for the
	corresponding tag.
Memory Expression	An expression for the memory tag is set up.

8.3.6 String Value

A collection of properties any tag can have only when its type is string.

Property Name	Description			
Eng Data Type	Type for tag's industrial value seen from the upper-level system.			
	In case of a string tag, it is fixed as "string" value.			
Raw Data Type	Type of tag values collected by the device			
	In case of a digital tag, it is fixed as "bool" value.			
Initial Value	The tag's initial value when a project starts.			
	But, be cautious that if the 'save' option is activated when the project ends, the initial			
	value will disregarded and the last value of the saved tag during the ending will be an			
	actual initial value.			
String Length	Set up string length. Up to 256 can be set up.			

8.3.7 Alarm General

A collection of alarm's common properties

For more explanation on alarm occurrence/recovery/processing, refer to Section "Alarm function" in the manual.

Property Name	Description				
Alarm Deadband	If the change in data values is fine in the border of alarm setting value, the user may set				
	an alarm deadband to prevent unnecessary alarm occurrence.				
Alarm Level	Each alarm has a level from 1 to 10 depending on its importance.				
Occurrence Display	String to be displayed when alarm occurs.				
String					
Recovery Display	String to be displayed when alarm is recovered.				
String					
User Definition	String to be defined by the user conveniently.				
String					
Delay Time (sec)	Delay time when alarm occurs. When any event occurs to prompt an alarm, it does not				
	occur immediately. Rather, it occurs after waiting as much as the delay time.				
	For details, refer to alarm functions.				

8.3.8 Analog Alarm

(1) Watch Alarm

A collection of properties related to setting a watch alarm.

Property Name	Description				
HH Alarm	HH alarm setting value. This value shall not exceed the maximum industrial value				
	and it shall be higher than HI value.				
HI Alarm	HI alarm setting value. This value shall be lower than HH setting value and higher				
	than LO setting value.				
LO Alarm	LO alarm setting value. This value shall be lower than HI setting value and higher				
	than LL setting value.				
LL Alarm	LL alarm setting value. This value shall be lower than LO setting value and higher				
	than the minimum industrial value.				

(2) Change Alarm

A collection of properties related to setting a change alarm.

Property Name	Description
Change Value	A criterion value to prompt a change alarm
	When the current tag value is compared to its past value and the value is changed
	significantly higher than the change value, an alarm will occur.

(3) Distance Alarm

A collection of properties related to setting a distance alarm.

Property Name	Description			
Distance Value	Type of the criterion value to prompt a distance alarm. The user may select either			
Туре	absolute value or relative value.			
Criterion Value	If an absolute value is set up, the absolute value becomes the criterion value for the			
	occurrence of a distance alarm.			
Criterion Tag	If a relative value is set up, the relative value becomes the criterion value for the			
Name	occurrence of a distance alarm.			
Main Distance	Main distance alarm value.			
Value				
Sub Distance	Sub distance alarm value.			
Value				

8.3.8 Digital Alarm

A collection of properties related to setting digital alarm.

Property Name	Description		Remark		
Alarm Kind	A rule for alarm of	A rule for alarm occurrence is set according to value in case of digital tag. The			
	following alarm k	following alarm kinds are in place.			
	• On : It is prompted when tag value is "On".				
	• Off	• Off : It is prompted when tag value is "Off".			
	• On → Off	: It is prompted when tag value is changed from On to Off.			
	• Off → On	: It is prompted when tag value is changed from Off to On.			
	When changing: It is prompted when tag value is changed.				

8.4 Expression

8.4.1 Basic Concept

In InfoU, memory tags can be calculated by using expression such as operation, conditional statement or variable.

In addition, it is used that adding a formula to the expression property of the subject tag that can be a result of the calculation in InfoU.

A data processing engine compiles all of expressions that belong to the tag when the project starts and sets up each operation group and operation priority to calculate.

It can be explained as follows:

- TagA, a memory tag, tries to operate with the value that equals to the value of Tag B, an I/O tag, plus one.
- TagC, a memory tag, uses the value that equals to the value of Tag D, an I/O tag, plus one.

The assumption above can be expressed as the following formula:

- TagA = TagB + 1
- TagC = TagA + TagD

If the items above are processed for Tag DB engineering, the following results can be delivered: Here, the subjects of the expression results become TagA and TagC and a formula are inserted to expression property of each tag.

Tag Name	Operation Interval	Expression	Remark
TagA	0	TagB + 1	Memory Tag
TagB	X	X	I/O tag
TagC	0	TagA + TagD	Memory Tag
TagD	Х	X	I/O tag

Two formulas above can be form one collection that has one causal relationship and memory tag operations can be carried out by finding out operation orders of the above collection internally in the data processing engine.

But, if tag operations are cross-referred to each other or they cycle along the route, any operation can not be performed.

- TagA = TagB + 1
- TagB = TagA + 1

8.4.2 Operation Interval

Any number other than "0" can be inputted for operation interval in the property window because the interval is inputted on a second base.

But, the operation interval can be divided into an event way and a regular way in terms of the performance of expression.

If "0" is set up, it works in an event way.

(1) Event Way

An expression that is formed by a causal relationship from the basic concept of the above expression follows this way.

The user shall input operation formula systematically as he/she inputs mathematical functions, then the internal engine decides the order of performing the operation formulas and carries out the operation in an artificial intelligence way.

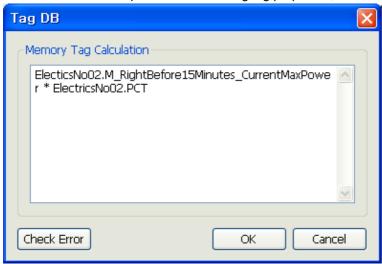
But, as stated above, special attention should be paid when cross-reference or cycling across the route is conducted.

(2) Regular Interval Way (sec)

Unlike in the event way, expressions that belong to the system are performed in a certain interval. There is no interactive relation and it is impossible to establish a priority for execution between each other in case expressions sharing the same interval. It is used when other tag values are referred regularly.

8.4.3 Grammar and Instruction

If the user clicks an expression item among tag properties, the following screen will appear.



If an expression is inputted and [Check Error] button is selected, inspection for error is performed. Once compiling is completed and [Ok] button is selected, the relevant expression shows tag properties and if [Cancel] is selected, the conversation box will end.

The following grammar can be used in the expression.

Category	Keyword	
Reserved word	IF, THEN, ELSE	
Operator	= , +, -, *, /, %, **	
Bit	8, , ~, ^, <<, >>	
Function	SIN, COS, TAN, SQRT, LOG, LOG10, EXP, ASIN, ACOS, ATAN, ABS, MAX, MIN, AVG, STRCAT	
Delimiter	(,)	

^{*} Reserved words and functions shall be inputted with capital letters.

^{*} IF~THEN~ELSE shall always exit in pairs.

[Example of using grammar]

1 1 00 1	
Expression	Description
IF (A + B > 10) THEN 4 ELSE 5	If the sum of Tag A and Tag B is higher than 10, return 4 and if that
	is lower than 10, return 5.
ABS(A)	Return A's absolute value
IF (A & B) THEN C ELSE D	If the sum of A and B is true, return the value of C, if not, return the
	value of D.

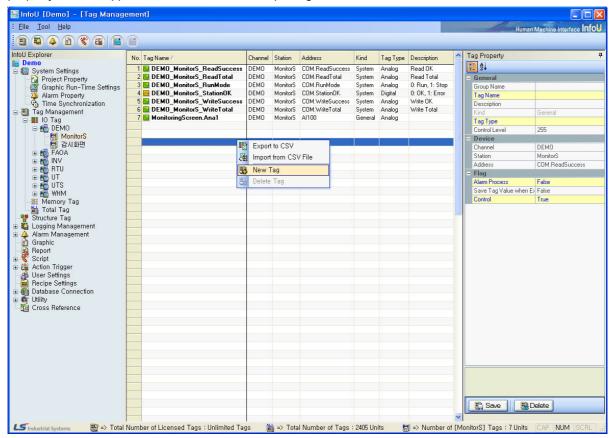
[Example of actual use]

Tag	Expression
Current Effective	Electricity No01. for 15 minutes right before_Current Maximum Power +
Power per 15	Electricity No02. for 15 minutes right before_Current Maximum Power +
minutes	Electricity No01.for 15 minutes right before_Current Maximum Power
Current Electric	WDC. Current Power * Calibrating Constant .PCT
Energy 2	
Current Electric	WDC.Current Power * Calibrating Constant .PCT+Calibrating
Energy	Constant .Correction Value
Current Power	(100 * MEM.Current Electric Energy) / 150000
_Ratio	
	IF ((WDC.Demand Time < 10) (WDC.Demand Time > 895)) THEN
	(MEM.Current Average Power) ELSE (WDC.Current Power *Calibrating
Current Power	Constant .PCT*900+Calibrating Constant .Correction Value) / (WDC.Demand
	Time)
	Electric Energy = MEM.Current Electric Energy / 4
Integration Pulse	WDC.Current Power *5000/4000
	IF(WDC.Demand Time < 60) THEN (MEM.Current Power) ELSE
Estimated Power	(WDC.Estimated Power * Calibrating Constant .PCT+Calibrating
	Constant .Correction Value)
	IF ((WDC.Demand Time < 61) (MEM. Estimated Power <= MEM.Current
Load Power	Electric Energy)) THEN (MEM.Current Power) ELSE ((MEM.Estimated Power
	- MEM.Current Electric Energy) * 900 / MEM.Demand Time _Remaining Time)
Criterion Power	WDC.Criterion Power * Calibrating Constant .PCT
PCT	(Calibrating Constant .PT * Calibrating Constant .CT)/(Calibrating
1 01	Constant .PT_2 * Calibrating Constant .CT_2)

8.5 Tag Edit

8.5.1 New Tag

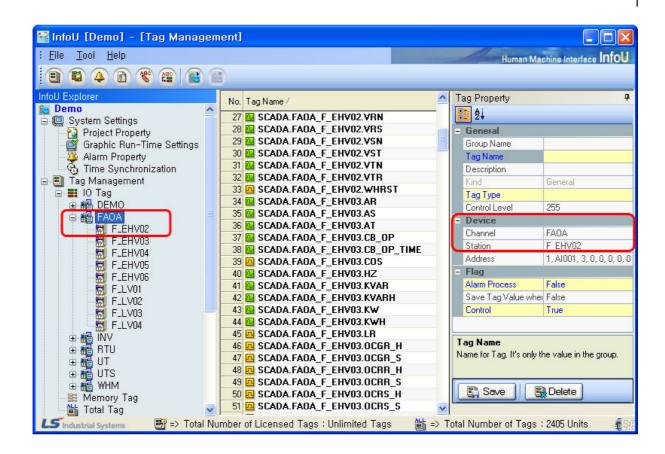
If any empty list is selected in the tag list screen or [New Tag] is selected from the pop-up menu, an empty property window appears to allow the user to input tags.



Here, important properties for engineering are highlighted with blue letters and yellow background and unchangeable items are shown in gray. .

And, in View Device Tag, channels and stations in the current locations are basically inherited and displayed according to the locations to execute [New Tag] menu. Of course, these values can be editable.

In View by Device, group names can be inherited. At this time, the inherited group name can be changed. If group names are changed, existing group names remain as they are and new group names are created.



8.5.2 Save

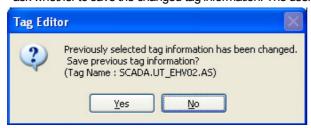
After editing tag properties and pressing "save" button, the following screen will appear.

If any existing tag is selected and edited, and then renamed, a new tag with a new name is added rather than modifying the said tag if [Save] button is pressed.



If any error occurs during the saving, an error notice window will appear rather than the "Save Confirmation Notice" window.

If moving to other tag record after editing tag properties without saving them, the following conversation box will appear to ask whether to save the changed tag information: The user may select [Yes] to save it or [No] to cancel the changed one.



8.5.3 Delete

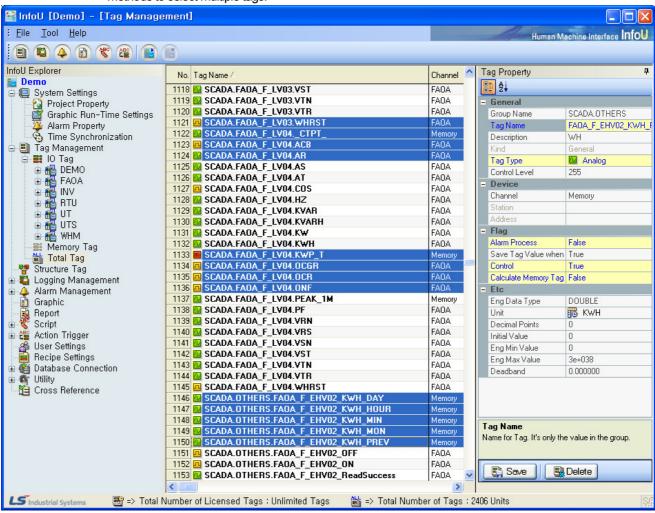
If [Delete] button is pressed after selecting a tag from the tag list, the following conversation box will appear to ask whether to delete the selected tag.



To delete tags, individual deleting and multiple deleting are available.

- Individual Deleting: Deleting is performed after one tag is selected from the tag list.
- Multiple Deleting: Deleting is performed after multiple tags are selected from the tag list...

Mouse dragging, Shift Key + Selecting with mouse, Ctrl Key + mouse and etc are available as methods to select multiple tags.



Unlike other applications, some steps exist for deleting tags.

Since a tag is created in order of Channel->Station->Tag, the reverse order is needed to delete it.

Therefore, any station or tag under the channel can not be deleted at a time and they shall be proceeded in order of deleting Tag→ Station → Channel to be deleted.

If a device that has lower-level tags is attempted to be deleted, the following error message is prompted for the user.



It is also true in View by Group so that any tag group with lower-level tags can not be deleted immediately. To delete a group, first delete tags under the group and then, delete the group.



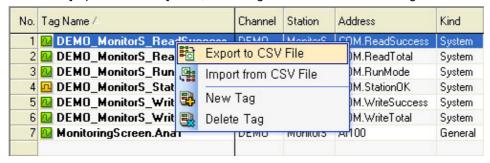
8.6 CST Tag In/Output

8.6.1 Introduction

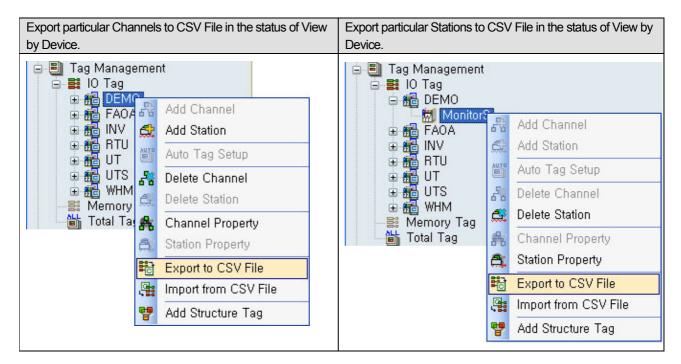
It needs significant time and efforts to engineer individual tags with the tag DB when conducting a large amount of tag work. To make this easy, this system provides an environment where CSV can be performed with the tag DB and the tag DB can be worked with CSV by using CSV files.

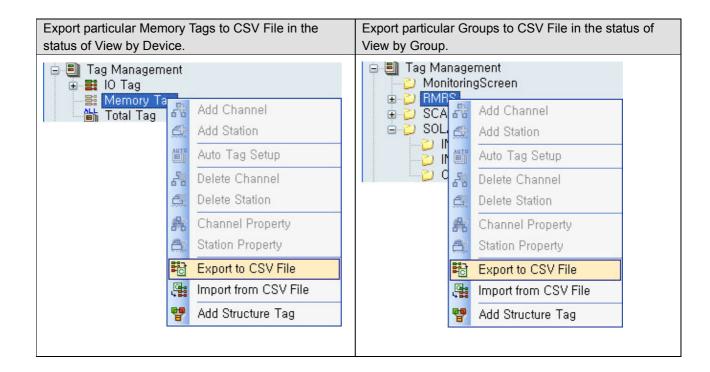
8.6.2 Export to CSV File

To execute [Export to CSV File] menu, click the right side of the mouse in the tag list screen to display the following menu.



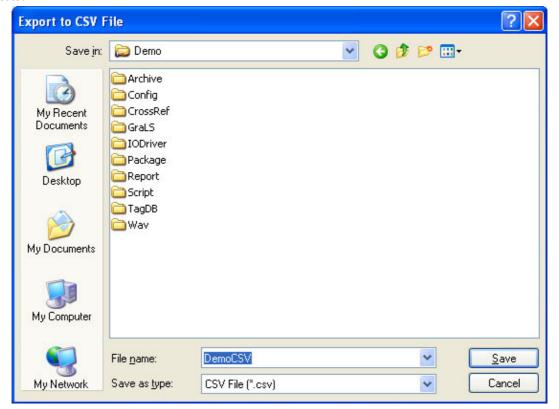
Also, it is also possible to export a channel, station, memory tag or group tag to CSV file. At this time, rather than exporting the entire tag DB, only selected tag DB can be exported to CSV files.

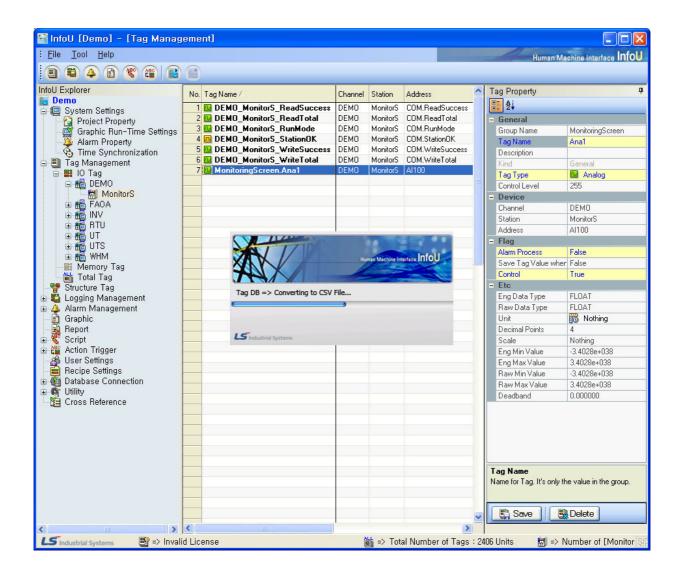




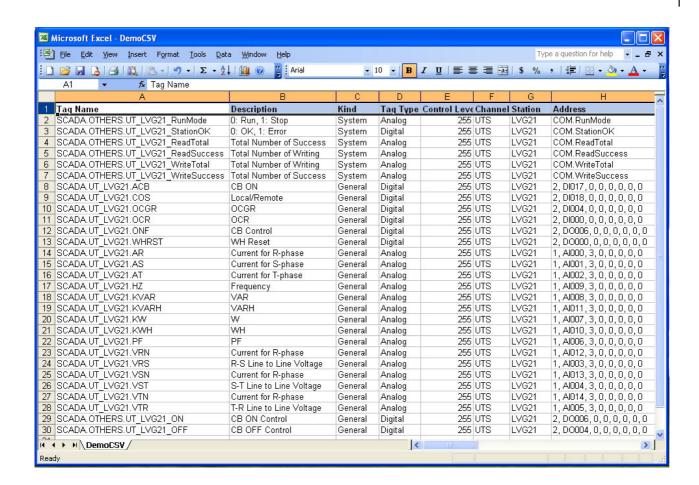
When selecting a menu, [Export to CSV File] conversation box appears as seen in the figure above.

If [Save] button is pressed after a location where the user wants to save the file is selected, exporting from [Tag DB] to CSV file is executed.





If CSV saving starts, this process is displayed in the splash in order of Convert->Save->Complete



The above figure shows that CSV file is executed after completing [Export to CSV File]. A new file created by [Export to CSV File] can be easily operated by the user in a text-file type and it is also conveniently edited.

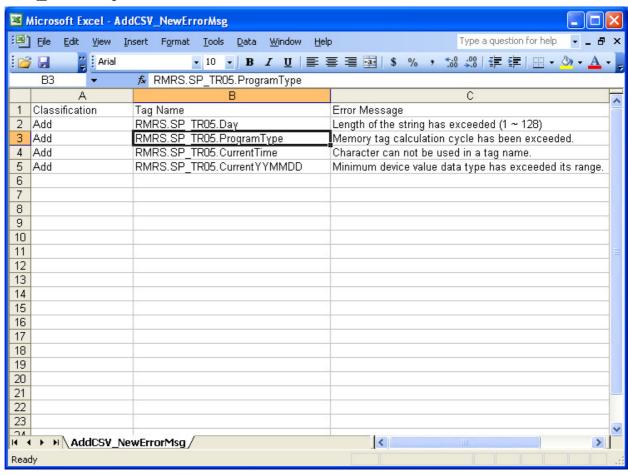
8.6.3 Import from CSV File

This is to bring any existing CSV tag or modified CSV tag after [Export to CSV File] to the Tag DB to edit or delete them. [Import from CSV File] consists of three menus as follows:

1. [CSV File Add] 2. [CSV File Update] 3. [CSV File Delete].

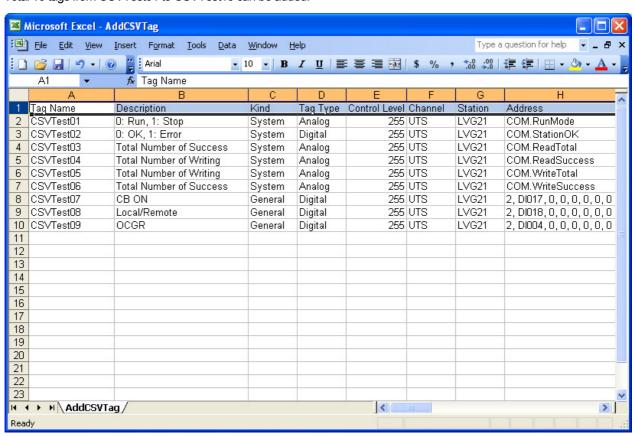


Error data created during "Import from CSV File" and "Export to CSV File" are saved in [CSV File Name]_NewErrorMsg.csv in the same folder as follows:



[Add CSV File]

The screen in the following figure is the one to set up a tag in CSV file to add it. Total 10 tags from CSVTest01 to CSVTest10 can be added.

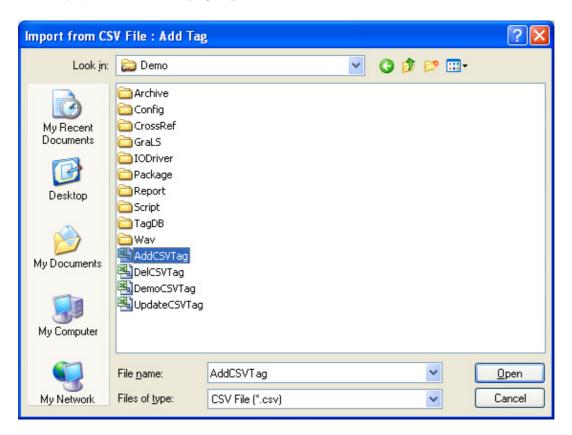


After saving the CSV files above, close the text file.

Then, select [Import from CSV File] menu to bring CSY tags to the Tag DB. After selecting Add CSV Tag as below, click Apply button.



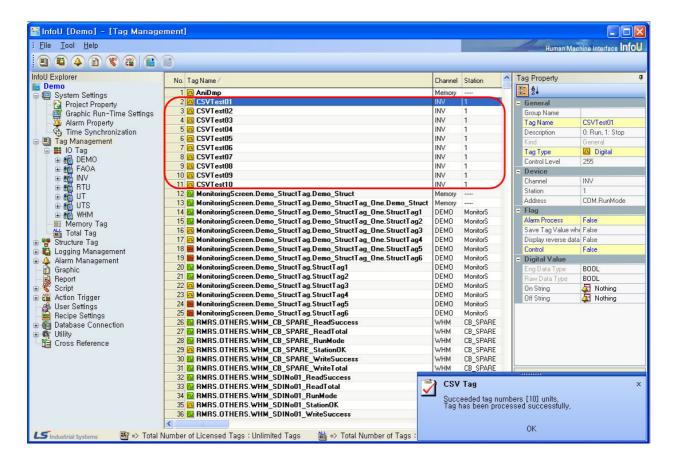
Conversation box [Import from CSV File: Add CSV Tag] is displayed. If [Open] button is clicked after a file to be imported is selected, [Import from CSV file to [Tag DB]] is executed.



This splash screen below shows the selected tags from CSV Test 01 to 10 CSV Test 10 in CSV files are completely added.

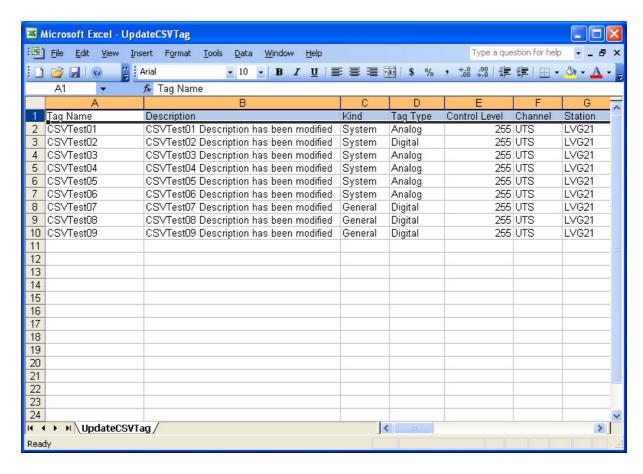


The screen in the following figure is the Tag DB screen that appears after [Import from CSV File] is executed and adding tags is completed. After adding tags is completed, the number of successful tags is displayed in the notice window. As seen in the figure, total 10 tags have been successfully added.

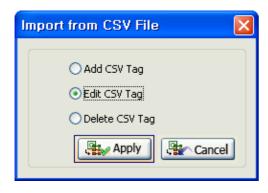


[Edit CSV File]

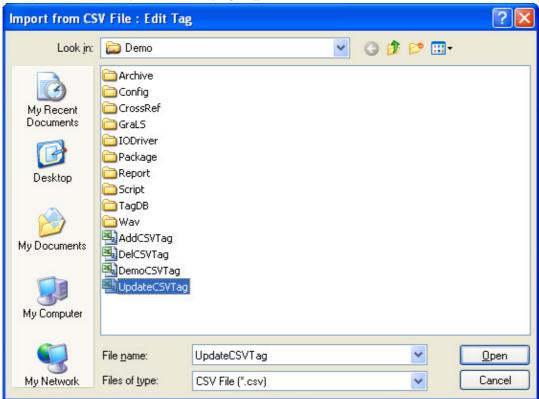
The screen in the following figure is the one to set up a tag in CSV file to update it. Descriptions on tags from CST Test 01 to CST Test 10 are updated.



Then, select [Import from CSV File] menu to bring CSY tags to the Tag DB. After selecting Add CSV Tag as below, click Apply button.



Conversation box [Import from CSV File]: Edit CSV Tag] is displayed. If [Open] button is clicked after a file to be updated is selected, [Edit CSV file to [Tag DB]] is executed.

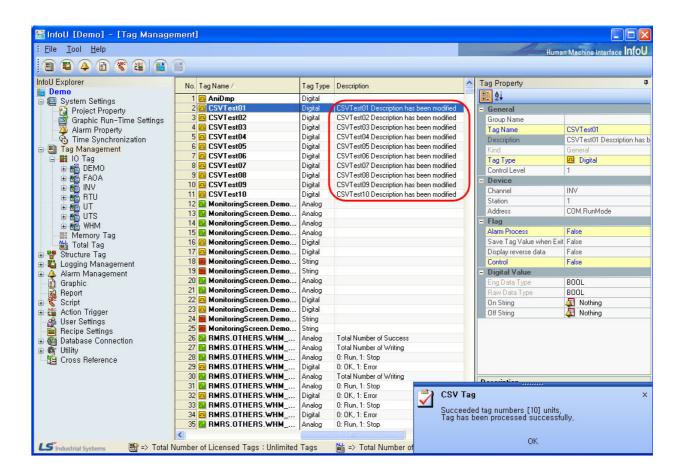


Monitoring screen set up in CSV file. The following screen is a splash screen that shows that editing tags from Ana1 monitoring screen to Ana 3 is completed.



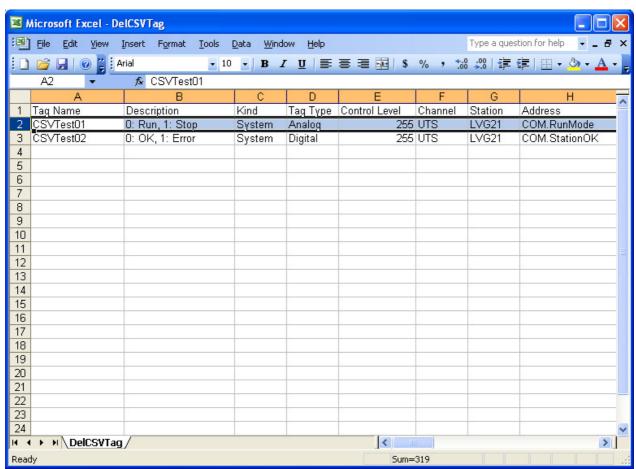
Chapter 8 Tag Management

The screen below is a tag DB screen that appears after [Import from CSV File] is executed and tag updating is completed. As seen in the part outlined by a red line, descriptions on 10 tags have been modified.



[Delete CSV File]

The screen in the following figure is the one to set up a tag in CSV file to delete it. Total 2 tags from CSVTest01 to CSVTest12 can be deleted.

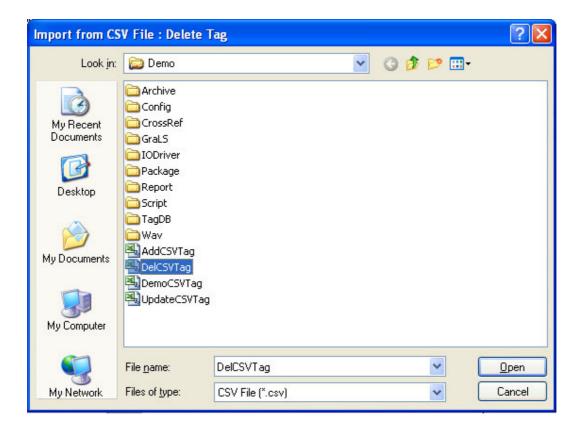


After setting to delete CSV tags as seen in the figure above, save the CSV file and close it.

Then, select [Import from CSV File] menu to delete CSY tags from the Tag DB. After selecting 'Delete CSV Tag 'as below, click Apply button



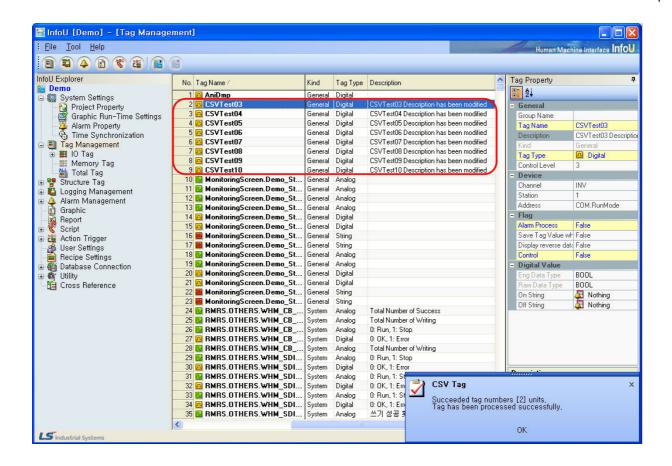
Conversation box [Import from CSV File]: Delete CSV Tag] is displayed. If [Open] button is clicked after a file to be deleted is selected, [Import from CSV file to [Tag DB]] is executed.



This splash screen below shows the selected tags from CSV Test 01 to 10 CSV Test 10 in CSV file have been completely deleted.



The screen below is a tag DB screen that appears after [Import from CSV File] is executed and deleting tags is completed. As seen in the figure below, two tags have been deleted.



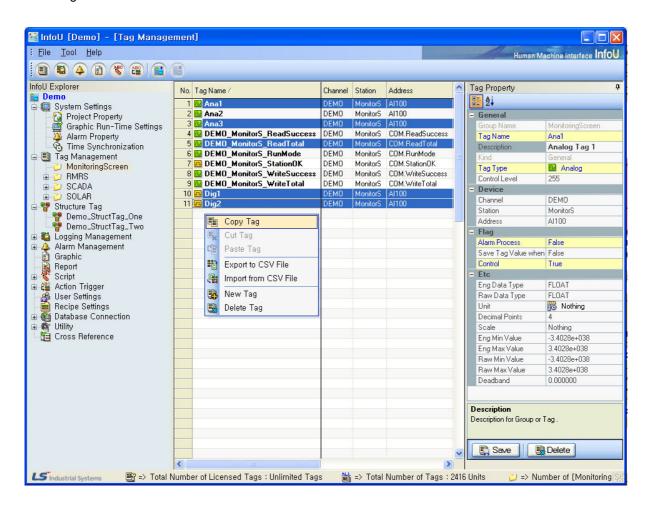
8.7 Copy/Paste Tag

8.7.1 Introduction

It needs significant time and efforts to engineer individual tags with the tag DB when conducting a large amount of tag work. Existing tags can be easily utilized if [Copy Tag] is used to paste the copied tag to the location the user wants with [Paste Tag], [Copy Tag] and [Paste Tag] presents a better environment to the user who wants to use existing tags for his/her tag works.

8.7.2 Copy Tag

Select [Copy Tag] menu from the tree structure of View by Group or click the right side of the mouse on the menu in the tag list screen to execute the menu.



For copying tags, individual copying and multiple copying are available.

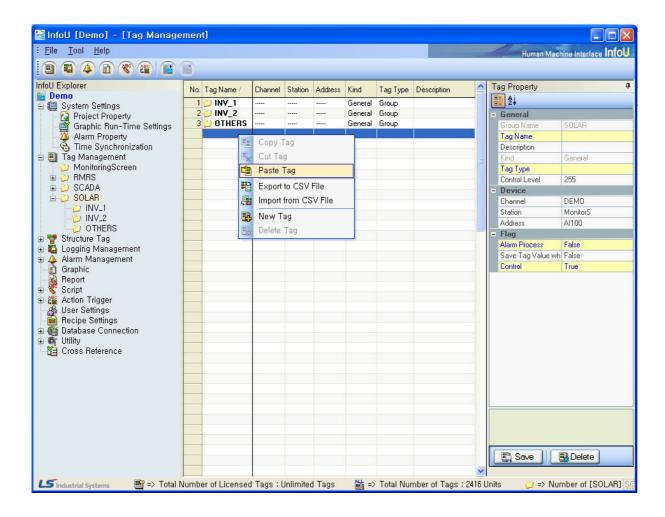
- Individual Copying: Copying is performed after one tag is selected from the tag list.
- Multiple Copying: Copying is performed after multiple tags are selected from the tag list.

Mouse dragging, Shift Key + Selecting with mouse, Ctrl Key + mouse and etc are available as methods to select multiple tags.

8.7.3 Paste Tag

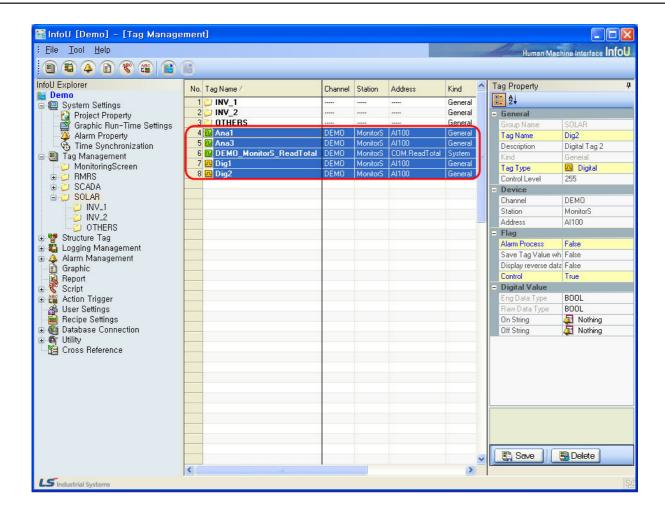
Select [Paste Tag] menu from the tree structure of View by Group or click the right side of the mouse on the menu in the tag list screen to execute the menu.

But, when executing [Paste Tag] menu, [Copy Tag] menu shall be preceded.



The screen below is a tag DB screen that appears after [Paste Tag] is executed and pasting tag is completed.

As seen in the figure, the 5 tags selected when executing [Copy Tag] menu have been added to the place the user wants.



8.8 **Structure Tag**

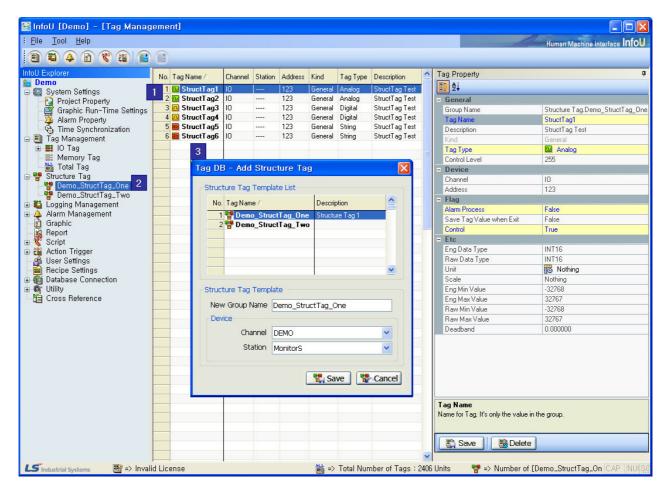
8.8.1 Introduction

This function allows the user to write tag groups that are frequently used in advance and to add tags in a group base for more convenient use of tags.

This function is mainly used when adding tag groups to the tag DB repeatedly, which express mainly physical devices or logical groups.

If specific tags are set up to register them as structure tag Templates, actual tags in a Template unit can be created and added to the tag DB.

8.8.2 Demo



Structure tags use the same screen configuration as that of tag management. By selecting a structure tag in Tool Explorer, the user may see relevant information in the template list screen and the tag property screen. Terms used in "Structure Tag" are defined as follows:

Structure Tag Item

Even though structure tag items have almost similar properties with general tags, they are not actual tags and they are copied to the Tag DB to be used. Structure tag items are included only in the structure tag Template and they can be used as general tags after being converted to general tags during the instantiating when structure tags are added.

Structure Tag Template

It means a name of representative tags that make one group. In actual, they are converted into one group during the instantiating when structure tags are added.

Add Structure Tag

It means a process in which the set up structure tag template is added to the structure tag and it is copied to the tag DB. This copying process is also called the instantiating process.

8.8.3 Add Structure Tag Template

To add a structure tag template, double click a structure tag item in the Tool Explorer and select [Add Structure Tag Template] by clicking the right side of the mouse. Also, click the right side of the mouse in the template list screen and select [Add Structure Tag Template] from the pop-up menu.

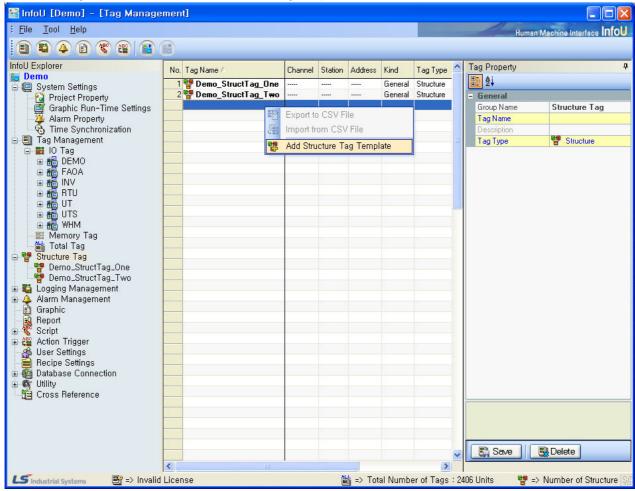
Double click "Structure Tag"



Click the right side of the mouse on "Structure Tag"



Click the right side of the mouse in the "Structure Tag Template List Screen"



If a menu is selected in the way above, a screen to define the template of the structure tag appears as seen the figure above.

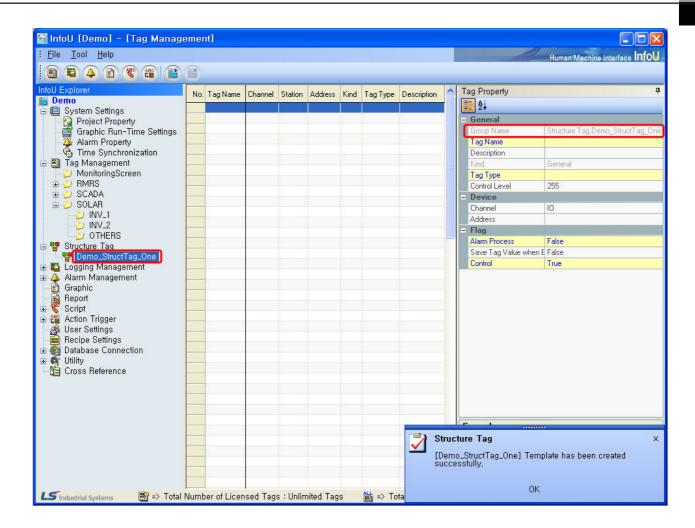


- (a) Name
 - It refers to the name of the template and the user may add structure tag items for registered templates.
- Description
- The user may add brief explanation on the template.
- (c)
 - A new structure tag template is created with the inputted information.
- (d) Cancel
 - Cancel the inputted information and close the [Structure Tag Template] window

Input the name and description of the relevant structure tag template in the conversation box [Structure Tag Template]. When the name of the structure tag template is inputted, "Save" button becomes activated.



The screen below is a tag DB screen that appears after [Structure Tag Template] is executed and adding the structure tag template is completed. The structure tag template is added and a newly created [Demo_StructTag] template appears on the structure tag tree. A message is displayed like follows: "the relevant structure tag template has been successfully created "in the notice window.



8.8.4 Add Structure Tag Item

It is a function to add a structure tag item that belongs to the structure tag template. Like in the way to register a general tag to a group in the tag DB, a structure tag item can be added by using the tag property window.

For details on how to set up a tag property window and how to edit a tag, refer to "Tag Management" manual.

The structure tag item is similar to a general tag but, some differences exist between them.

The differences are as follows:

(1) Tag Type

Unlike other common tags, structure tag items can not be set up as [Group].

(2) Channel

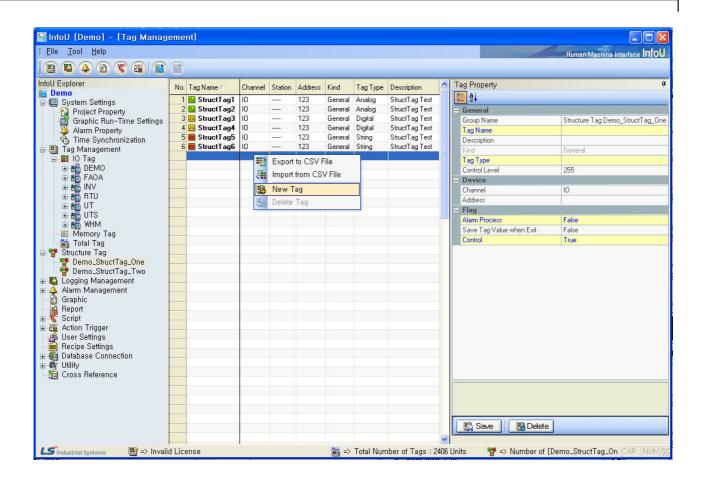
For channels, only [I/O tag] and [Memory tag] are selectable. If [I/O Tag] is selected, inputting is available at the point of time when the structure tag is actually inputted to the tag DB, that is, a point of time when the structure tag is inputted to become instantiated. If [Memory Tag] is selected, the relevant tag shall be set up as a memory tag.

(3) Station

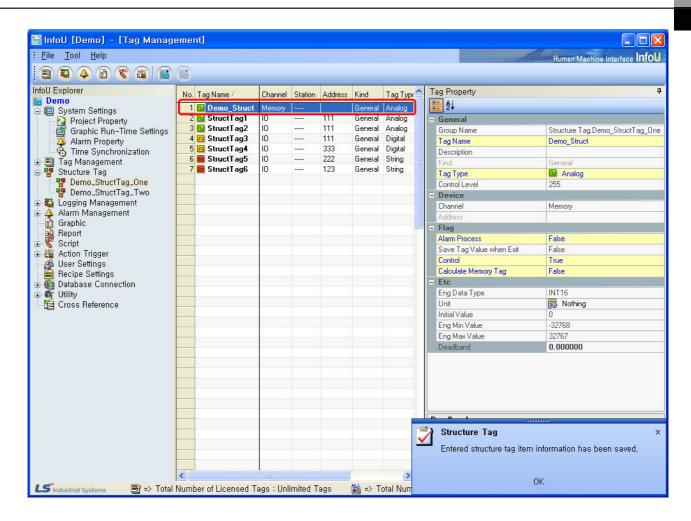
Similar to the channel, in the station, it is possible to set up the relevant templates collectively at the moment when they are added to the structure tag and instantiated.

For adding a structure tag item, if any empty list is selected in the tag list screen or [New Tag] is selected from the popup menu, an empty tag property window appears to allow the user to fill out the window.

Chapter 8 Tag Engineering



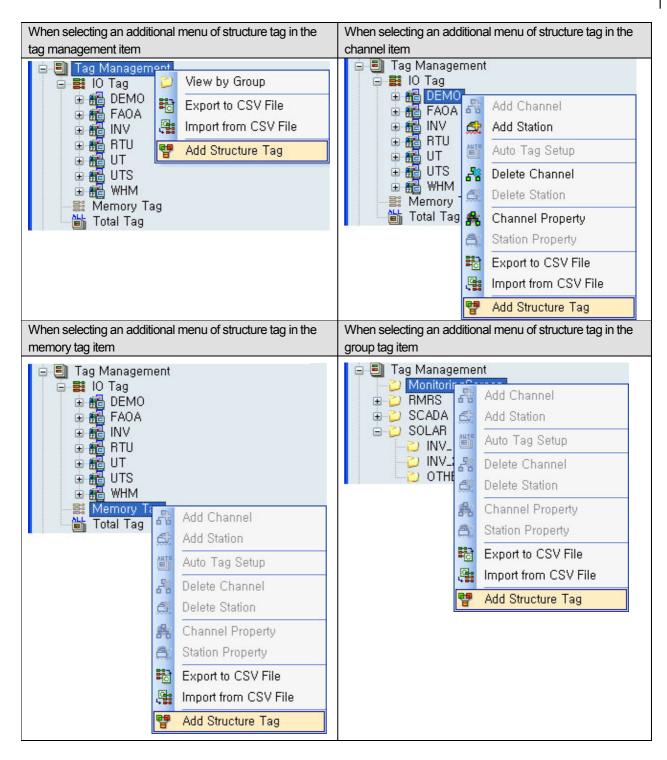
The screen below is a tag DB screen that appears after "Add Structure Tag Item" is executed and tag updating is completed. As seen in the figure, one structure tag item is added and message saying that "The information on the inputted structure tag item has been save" appears in the notice window.



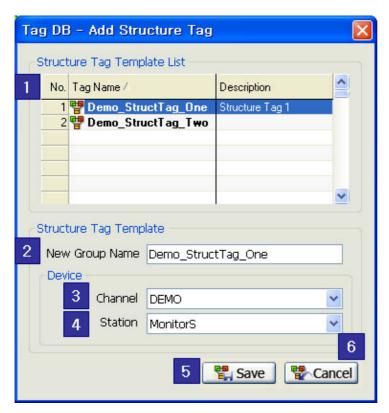
8.8.5 Add Structure Tag

It is a work to copy the set up structure tag to the tag DB. A structure tag can be added only in a template unit and if added, it is created as a new group in the tag DB. To make a pop-up menu appear, click the right side of the mouse on the tag management item or its low-level item. Through this menu, [Add Structure Tag] can be executed.

Chapter 8 Tag Engineering



If a menu is selected in one of the ways above, "Add Structure Tag" window appears as seen in the following figure



(1) Structure Tag Template List

The currently created structure tag template appears in the list.

(2) New Group Name

Input the name of the group tag to be used when adding a structure tag template

(3) Channel

The channel list set up in the current project is presented. If a channel is selected, the entire tags other than [memory] tags among structure tag item properties are assigned collectively to the same channel to be inputted. It is conveniently used when assigning structure tag templates to devices.

(4) Station

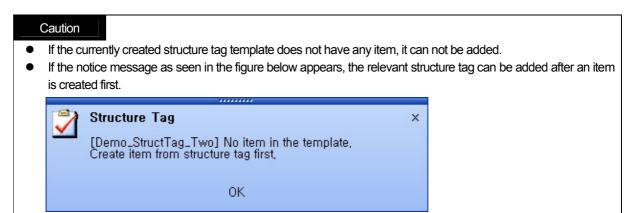
A station is selected to be assigned to the channel. It is collectively applied as in the channel.

(5) Save

A structure tag template is instantiated to copy it to the tag DB.

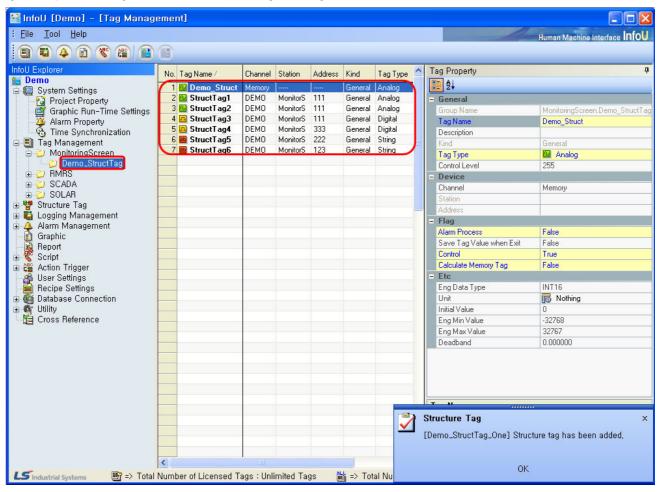
(6) Cancel

Adding structure tag is cancelled and the window is closed.



Chapter 8 Tag Engineering

To add a structure tag, press "Save" button after setting up each item to create a relevant group [Demo StructTag] as seen in the screen below. Copy a structure tag item under the relevant group to the tag DB to make the list screen appear. The structure tag item copied to the tag DB can be used as if it is a general tag.

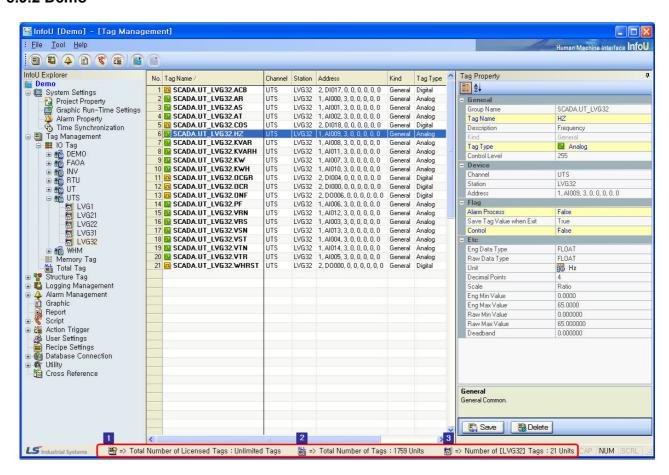


8.9 **License Information**

8.9.1 Introduction

By showing brief information on the currently used license in the status bar, it helps the user to identify the number of tags easily.

8.9.2 Demo



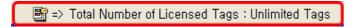
The status bar shows the following information:

(1) License Tag

It shows the status of the currently used license. It shows whether the license is valid or not and how many license tags can be used. If a valid license is used, it shows the available number of tags under the license

Chapter 8 Tag Engineering

When valid license is used:



When invalid license is used:



- (2) Total number of tags

 Total number of tags saved in the current DB is displayed.
- (3) Group, device, station tags selected by the user

 The number of tags that are being used in the item selected by the user through [Tool Explorer]=>[Tag Management].

Caution

• If any invalid license is used, the following warning message will be displayed after one hour and the program will end.



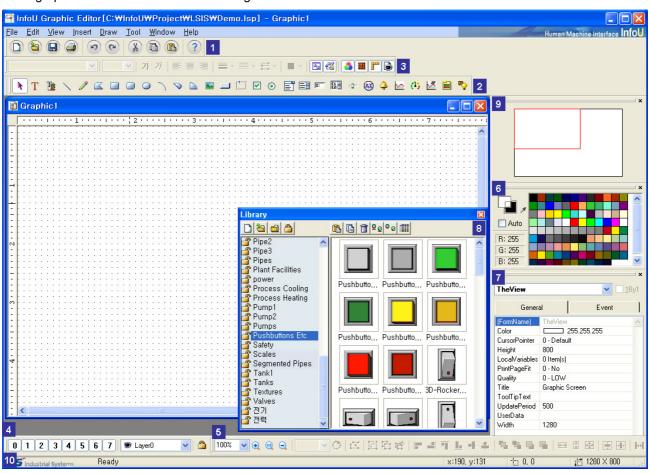
• If the available tag number under the license is exceeded, those tags processed exceeding the available tag number will not be saved there fore special caution should be taken.

This graphic editor provides the following various functions for editing the monitoring screen as a program used to edit the process screen:

- (1) Graphic drawing tools
- (2) Various graphic libraries.
- (3) Support of ActiveX container and its server, ActiveX control.
- (4) OLE automation interface for execution and manipulation from other application programs.
- (5) Visual basic script for robust applications
- (6) Exclusive ActiveX control and availability of using ordinary ActiveX
- (7) Function to inquire tag information used in the object and screen
- (8) Function to save tag information used in the object and screen in the cross-reference database
- (9) Function to prohibit or allow control according to the user's authorization
- (10) Function to edit diverse and easy-to-use objects
- (11) Support of various dynamic properties and user's action functions to display dynamic information through animation based on the site data
- (12) Support of template screen

9.1 Configuration of Graphic Editor

This graphic editor consists of the following menus and tool collections.



As a system development tool, this InfoU graphic editor defines and sets up system components needed to perform projects such as defining dynamic properties of screen configuration, setting up execution environments and setting up page environments. The figure above shows an example of the InfoU graphic editor screen when a basic page is created and the main tool collection needed for the screen configuration is as follows:

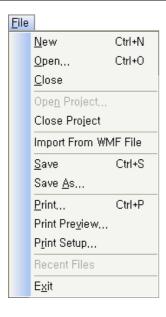
- (1) Standard tools: Command icons the most frequently used in the graphic editor
- (2) Object tools: Object icons needed when writing a page.
- (3) Format tools: Icons needed to define font or line prosperities
- (4) Layer tools: Tools that enable to perform layer organization works useful for editing.
- (5) Draw tools: Icons related to Zoom, Group/Regroup and Align.
- (6) Color tools: A tool window that designate the color of page or object conveniently
- (7) Object Property Window: A window to define each object's properties or define events with scripts
- (8) Library: A window to register/delete/edit symbols in the library
- (9) Navigation: Tools to ensure easy moving to a work area the user wants to go.
- (10) Status tools: Tools to show major properties such as object coordinates or Help

9.2 **Menu Configuration**

Upper-level Menu	Menu	Description
File	New	Create a new page
	Open	Open one among existing documents
	Close	Close the current document.
	Open Project	Open one among existing projects.
	Close Project	Close the current project.
	Import From WMF File	Import WMF file in the currently activated document.
	Save	Save the activated document with the same name.
	Save As	Save the activated document with the different name.
	Print	Print document.
	Print Preview	Display the contents to be printed in the screen.
	Print Setup	Set up print options
	Recent Files	Open documents that have been worked recently.
	Exit	Exit the graphic editor.
	Undo	Cancel the previously executed command.
Edit	Redo	Re-input the cancelled command.
	Cut	Delete a certain range of data from the document and move it to the clipboard.
	Сору	Copy a certain range of data to the clipboard
	Paste	Paste the copied data to the document from the clipboard.
	Delete	Delete the selected data.
	Select All	Select the entire objects in the page.
	Find/Replace String	Find a certain string used for the object in the page and replace the found string with other string.
	Find/Replace Tag Link	Find a certain tag used for the object in the page and replace the found tag with other tag
	Tag LinkMS Excel Export	Send tags mapped in the page to Excel file.
View	Toolbars	Show or hide standard, object, format and draw tools.
	Status Bar	Show or hide status tools.
	Library	Show or hide the symbol library dialog box.
	Color	Show or hide color tools.
	Navigation	Show or hide navigation tools for moving to other screen.
	Layer	Show or hide layer tools.
	Dynamic Properties	Show or hide dynamic properties of the object and the screen for action settings.
	Object Properties	Show or hide the property screen for editing object types or shapes.

Upper-level Menu	Menu	Description
View	Ruler	Show or hide the ruler.
	Snap to Grid	Setup or cancel snapping to grid.
	Grid Settings	Designate grid intervals and colors by displaying the grid setting dialog box.
	Zoom	Zoom in/out the page with setting rates.
	Full Screen	Convert the page into a full screen size.
Insert	Figure	Draw various figure objects.
	Window Control	Draw various window control objects.
	Text	Input strings.
	Tag Value	Create an object with tag values.
	Image	Insert image files.
	ActiveX	Insert OCX objects.
	Alarm Viewer	Insert alarm viewer objects.
	Trend Viewer	Insert trend viewer objects.
	Gauge	Insert gauge viewer objects.
İ	List Trend	Insert list trend objects.
	Recipe	Insert recipe objects.
	Picture Lamp	Insert picture lamp objects.
	Group	Create one group object with several objects.
	Ungroup	Cancel the group object.
	Regroup	Recover the control function over the objects that have been
	Order	grouped and then cancelled to re-create the group. Bring the select object to the very front or sent it to the very end or
	Cidei	bring it to the front or send it to the rear.
Draw	Edit Points	Change the shapes of multiple curves or looped curves freely.
	Open/Close	Convert multiple curves to looped curves and looped curves to multiple curves.
	Align	Align, assign and arrange the selected objects.
	Make Same Size	Make the sizes of the selected objects equivalent.
	Rotate	Rotate the selected objects freely to their original locations or
	Run Mode	90°rotation to the right Preview the operation screen of the runtime.
	Objects	Change the order of the object list and set up the tap sequence for
	Objecte	keyboard run.
Tool	Declutter Settings	Set up a range of enlarge rate for each layer to be displayed in
	Graphic Runtime Settings	each screen during the runtime. Set up the execution environment of graphic runtime.
	Option	Set up the environment such as number of cancellation or mode to
	·	select multiple options.
	Cascade	Arrange windows in a cascade form.
Windows	Tile	Arrange windows in a tile form.
	Arrange Icons	Arrange windows to the location to designate various tool bar defaults.
Llok	User Manual	Show the Help items of the graphic editor.
Help	About Graphic Editor	Show version information of the graphic editor.

9.3 File

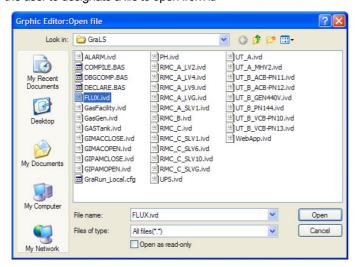


(1) New

A new graphic file is created.

(2) Open

This command is used to open one of the existing documents. If this command is made, a dialog box appears to allow the user to designate a file to open from it.



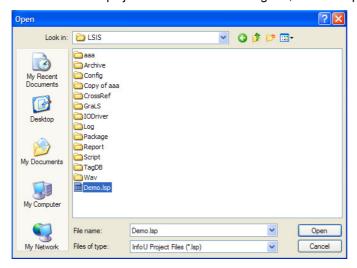
(3) Close

The currently activated document is closed with this command. Before the document is closed, it needs to be saved. If it

is closed without saving, the changed contents that have been made before the latest saving will disappear. When it is closed, a dialog box appears to allow the user to designate a name and save it with the name.

(4) Open Project

If this command is made, a dialog box appears to allow the user to open one of the projects that have been already written. If one of the projects is selected in the dialog box, the current project automatically saved and then closed.

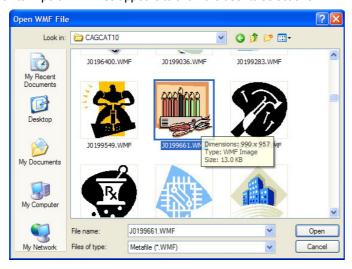


(5) Close Project

The current project is closed.

(6) Import From WMF File

A WMF file is inserted to the current work screen in the graphic editor's own format type. If this command is made, a dialog box to import WMF files appears to allow the user to select one.

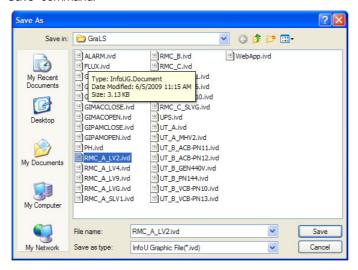


(7) Save

This command is used when saving the activated document with the same name as the current one. If the page document is saved for the first time, the "Save" dialog box appears to allow the user to designate its name. If the user wants to save the document with other name, he/she may choose "Save As" command.

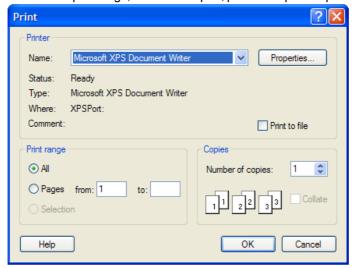
(8) Save As

This command is used when saving the activated document with a different name from the current one. the "Save" dialog box appears to allow the user to designate its name. If the user wants to save it with the current name, he/she may use "Save" command.



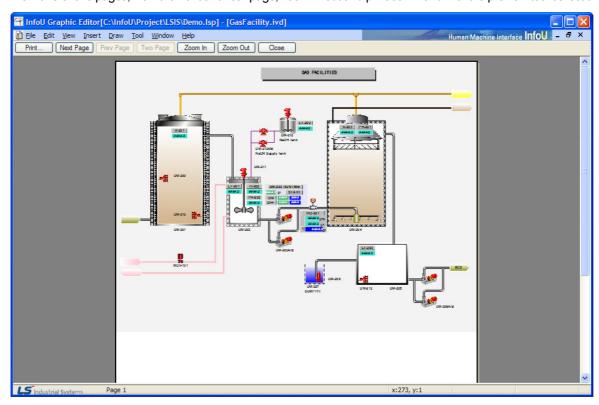
(9) Print

This command is used when printing document. If this command is executed, the "print" dialog box appears to allow the user to select a print range, number of copies, printer and printer option.



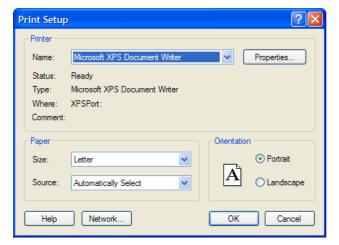
(10) Print Preview

This command allows to user to preview the content to be printed in an actual output format in advance. If this command is performed, a preview window is displayed. The user may perform various functions such as view one or two pages, view the front and rear page, zoom in/out and print command with the preview tool collection.



(11) Print Setup

Letter size and orientation for printing documents can be selective.

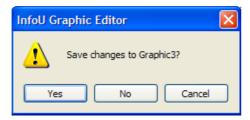


(12) Recent Files

The user can open the documents that have been worked recently in this graphic editor. Up to 10 recent documents can be displayed. The user can select one among the displayed files.

(13) Exit

The graphic editor is ended. If any changed content exists without being saved, a dialog box appears to ask the user whether to save it or not. At this time, press "Yes" if he/she wants to save it otherwise press "No." If "Cancel" is selected, this 'Exit' command is cancelled.



9.4 Edit



(1) Undo

This command is used when the user wants to cancel the contents that have been recently inputted and return to the previous status. Up to 20 events of inputting can be cancelled.

(2) Redo

This command is used when re-performing the cancelled contents.

(3) Cut

This command is used to delete the selected data from the document and copy them to the clipboard. If no data is selected currently, this command can not be used. If this "Cut" command is performed, the previous data in the clipboard is replaced.

(4) Copy

This command is used to copy the selected data to the clipboard. If no data is selected currently, this command can not be used. If this "Copy" command is performed, the previous data in the clipboard is replaced.

(5) Paste

This command is used to copy data in the clipboard and insert them to the document. If the clipboard is empty, this command can not be used.

(6) Delete

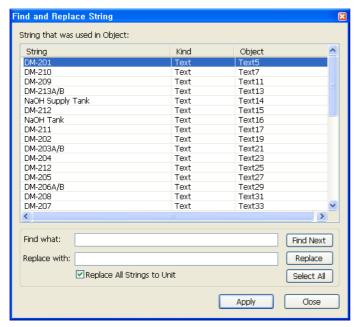
This command is used to delete the selected data. If no data is selected currently, this command can not be used. If important data are about to be deleted, a dialog box appears to ask the user whether he/she really wants to delete them, At this time, he/she may press "Yes" to delete them or he/she may press "No" to cancel the 'Delete' command.

(7) Select All

The entire objects in the page are selected.

(8) Find/Replace String

This command is used to find a string that was used throughout the objects in the page and replace the found one with other one. When this command is executed, the "Find and Replace String" dialog box appears.



(a) Find what

Input the string to replace with.

(b) Replace with

Click "Replace" button to display the planned strings in the list.

(c) Apply

Click "Apply" button to replace the strings with the ones displayed in the list.

(9) Find/Replace Tag Link

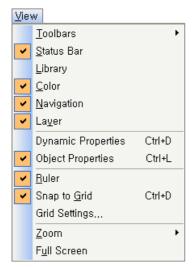
This command is used when finding mapped tags among the entire objects or the selected objects in the page or replacing the found tags with others. When this command is executed, the "Find and Replace Tag Link" dialog box appears. For detailed explanation, refer to the "Tag Link" provision.

(10)Excel Export

This command is used when exporting the mapped tags in the page to excel.



9.5 View



(1) Toolbars

There are [Standard], [Object], [Format] and [Draw] tools and they are either shown or hid with various related tools in the tool window upon clicking on each one.

(a) Standard

It is used when showing or hiding standard tools. Standard tools are command icons the most frequently used in the graphic editor. When it is shown, the check mark appears in front of the menu.



(b) Object Bar

It is used when showing or hiding object tools. Object bar has object icons needed to write pages. When it is shown, the check mark appears in front of the menu.

For details, refer to the 'Insert' menu.



(c) Format Bar

It is used when showing or hiding format tools. This format bar has various icons including font, size, bold, underline, italic, arrange to the left, center and right, line thickness, style and arrow type. When it is shown, the check mark appears in front of the menu.



(d) Draw Bar

It is used when showing or hiding draw tools. This draw bar has various icons related to functions to arrange objects in the page and icons for Group/Regroup/Ungroup, Zoon in/out and Rotate. When it is shown, the check mark appears in front of the menu. For details, refer to the 'Draw' menu.



(2) Status Bar

It is used when showing or hiding status tools. 'Help' messages or coordinates are displayed in the status bar. When it is shown, the check mark appears in front of the menu.

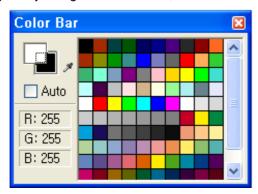


(3) Library

It is used when showing or hiding the library dialog box. In addition to the library provided by the system, the user may register/delete/edit his/her own library. For detailed explanation, refer to the "Library" provision.

(4) Color

It is used when showing or hiding color tools. This color bar has a part to set up view colors, background colors and transparency setting. When it is shown, the check mark appears in front of the menu.



(5) Navigation

It is used when showing or hiding navigation tools. If the current work area is small, this navigation bar allows the user to move pictures to the work area he/she wants. When it is shown, the check mark appears in front of the menu.



(6) Layer

It is used when showing or hiding layer tools. This lay bar can either show or hid the relevant layer and it can also lock it. For detailed explanation, refer to the "Layer" provision.



(7) Dynamic Properties

It is used when setting up dynamic properties and action functions of the relevant object. First, select an object and execute this command to show a dialog box to set up its dynamic properties. The displayed dialog box allows the user to set up dynamic properties and action functions of the relevant object. For detailed explanation, refer to the "Dynamic Property Definition" provision.

(a) Horizontal Size

A function to change the object's size in the horizontal direction according to tag values

(b) Vertical Size

A function to change the object's size in the vertical direction according to tag values

(c) Horizontal Move

A function to move the object in the horizontal direction according to the designated tag value

(d) Vertical Move

A function to move the object in the vertical direction according to the designated tag value

(e) Horizontal Fill

A function to fill the object in the horizontal direction according to the designated tag value

(f) Vertical Fill

A function to fill the object in the vertical direction according to the designated tag value

(g) Color

A function to change the object in the prescribed color according to the inputted tag value

(h) Blink

A function to blink the object in the specified interval according to the tag status

(i) Visible

A function either to show or hide the object according to the input tag value

(i) Rotate

A function to rotate the object by the designated angle degree according to the input tag value

(k) Display Value

It is used to create an object marked with its tag value.

(I) Display String

A function to change the object with the prescribed string according to the inputted tag value

(m) Click

A control function to execute the defined action when clicking on the object with the mouse or keyboard.

(n) Input Value

A function to select an object and input a value in the predefined dialog box with the mouse or keyboard

(o) Horizontal Drag

A function to drag the object in the horizontal direction with the mouse to convert coordinates values to tag values

(p) Vertical Drag

A function to drag the object in the vertical direction with the mouse to convert coordinates values to tag values

(8) Object Properties

If an object is selected and [View] [Object Properties] is clicked, object properties of the selected object and the event list window appear. For detailed explanation, refer to the "Object Properties" provision.

(9) Ruler

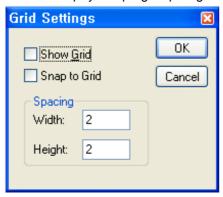
It is used when showing or hiding the rulers on the top and the left side in the current page

(10) Snap to Grid

It is used to set up or cancel the 'Snap to Grid' command. If 'Snap to Grid.' is set up, objects move to adjust to the grid coordinates. If cancelled, objects can move freely.

(11) Grid Settings

It is used to designate grid spacing and whether to show grid. If this command is performed, the 'Grid Settings' window is displayed. Input grid spacing and check in the box of 'Show Grid,' grid appears in the screen.



(12) Zoom

To make a more delicate edition in the screen, use zoom in or out. From 400% to 25% zooming can be designated.

(13) Full Screen

It is used when displaying the current page in a full screen size.

9.6 Insert



(1) Figure

(a) Line

It is used to draw a line. If this command is executed, the edit mode is changed to the line draw mode.

(b) Free Line

It is used to draw a free curve. If this command is executed, the edit mode is changed to the free line draw mode.

(c) Poly Line

It is used when drawing multiple curves or looped curves. If this command is executed, the edit mode is changed to the free draw mode.

(d) Rectangular

It is used to draw a rectangular. If this command is executed, the edit mode is changed to the rectangular draw mode.

(e) RoundRect

It is used to draw a round square. If this command is executed, the edit mode is changed to the round square draw mode.

(f) Arc

It is used when drawing a circle or an oval. If this command is executed, the edit mode is changed to the arc draw mode.

(q) Chord

It is used to draw a circular arc. If this command is executed, the edit mode is changed to the chord draw mode.

(h) Pie

It is used to draw a fan shape. If this command is executed, the edit mode is changed to the pie draw mode

(i) String

It is used to draw a string. If this command is executed, the edit mode is changed to the string draw mode.

(2) Window Control

(a) Button

It is used to draw a button. If this command is executed, the edit mode is changed to the button draw mode.

(b) Group Box

It is used to draw a group box. If this command is executed, the edit mode is changed to the group box draw mode.

(c) Check Box

It is used to draw a check box. If this command is executed, the edit mode is changed to the check box draw mode.

(d) Radio Button

It is used to draw a radio button. If this command is executed, the edit mode is changed to the radio button draw mode.

(e) Combo Box

It is used to draw a combo box. If this command is executed, the edit mode is changed to the combo box draw mode.

(f) List Box

It is used to draw a list box. If this command is executed, the edit mode is changed to the list box draw mode.

(g) Edit Box

It is used to draw an editor box. If this command is executed, the edit mode is changed to the edit box draw mode.

(h) Multiple Edit Box

It is used to draw a multiple editor box. If this command is executed, the edit mode is changed to the multiple edit box draw mode.

(i) Slider

It is used to draw a slider. If this command is executed, the edit mode is changed to the slider draw mode.

(3) Text

It is used to input text. If this command is executed, the edit mode is changed to the text input mode.

(4) Tag Value

It is used to create an object marked with its tag value. If this command is executed, the edit mode is changed to the object input with tag value mode.

(5) Image

It is used to insert various types of image files. If this command is executed, the edit mode is changed to the image insert mode and the 'Open' dialog box appears. Select an image file to insert in the displayed dialog box. For detailed explanation, refer to the "Image Insert" provision.

(6) ActiveX

It is used to insert Active X object. If this command is executed, the edit mode is changed to the Active X insert mode and the 'Open' dialog box appears. Select an Active X object to insert in the displayed dialog box.



(7) Alarm Viewer

It is used to insert Active Viewer object. If this command is executed, the edit mode is changed to the Active Viewer object insert mode.

(8) Trend Viewer

It is used to insert Trend Viewer object. If this command is executed, the edit mode is changed to the Trend Viewer object insert mode.

(9) Gauge

It is used to insert Gauge object. If this command is executed, the edit mode is changed to the Gauge object insert mode.

(10) List Trend

It is used to insert List Trend object. If this command is executed, the edit mode is changed to the List Trend object insert mode.

(11) Recipe

It is used to insert Recipe object. If this command is executed, the edit mode is changed to the Recipe object insert mode.

(12) Picture Lamp

It is used to insert Picture Lamp object. If this command is executed, the edit mode is changed to the Picture Lamp object insert mode.

9.7 **Draw**



(1) Group

If this command is executed while more than two objects are selected, those selected objects are bound into one. During the period between the time when selecting 'Group' and the time when canceling this 'Group,' the selected objects move together as a group.

(2) Ungroup

It is used to cancel the selected objects as a group and release them into individual objects. If this command is executed, control functions defined for the group disappear. These control functions for the group can be recovered with the command 'Regroup',

(3) Regroup

To edit some of the objects in a group with control functions, cancel the 'Group' command and edit them. After finishing editing, use this command to recover those control functions. Regrouping is not allowed if other objects are newly added or some of them are deleted. This command is available only when original objects are edited.

(4) Order

It is used to decide the order of the overlapped objects. There are [Bring to the Very Front], [Send to the Very Rear], [Bring to Front] and [Send to Back],

(5) Edit Points

It is used to change shapes of free curve, multiple curve and looped curve objects. If this command is executed, a black rectangular mark appears on each vertex. The user may change shapes by selecting the displayed mark and moving it with the mouse.

(6) Open/Close

It is used to convert from multiple curves to looped curves and looped curves to multiple curves. If this command is executed, multiple curves are converted to looped curves or looped curves are converted to multiple curves.

(7) Alian

As a function to arrange more than two objects at the same position vertically and horizontally, distribute or sort them, there are [Align to left], [Align to center], [Align to right], [Align to top], [Align to middle], [Align to bottom], [Same horizontal spacing], [Same vertical spacing], [Vertical center] and [Horizontal center].

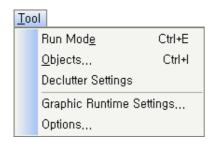
(8) Make Same Size

It is used to assign the same size to all of the selected objects.

(9) Rotate

It is used to rotate objects in the direction the user wants. [Free Rotate] allows the user to rotate them freely and [Rotate 90° to the right] allows the user to rotate them by 90°. [Original Status] sends the direction of objects to their own status.

9.8 **Tool**

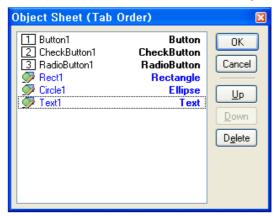


(1) Run Mode

If this command is executed, a new window is created and runtime begins to operate as in the Runtime screen. The user may test during the editing by loading its actual operation status to check whether it is properly operated.

(2) Objects

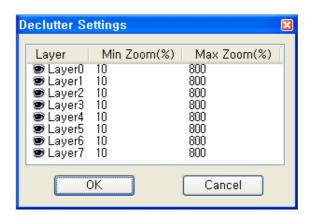
All of the activated objects are displayed in the screen. The user may set up the order among objects or rename or delete objects in the Object Sheet dialog box and he/she may also organize a sequence to move one object to the other one to ensure operation is enabled without mouse during the Runtime.

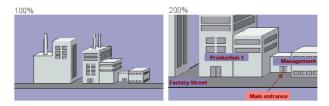


(3) Declutter Settings

During the Runtime, layers (the entire objects in the layers) can be shown or hid depending on the zoom in/ out rate. Therefore, overall objects can be seen in the zoom out screen while closer and more detailed objects can be displayed in the zoom in screen. The user can assign the minimum and maximum zoom rate unique to each layer. Each layer can be seen in the assigned range and the rate of its zoom in/ out are set in percentage.

For detailed explanation on how to assign to layers, refer to the "Layer" provision.



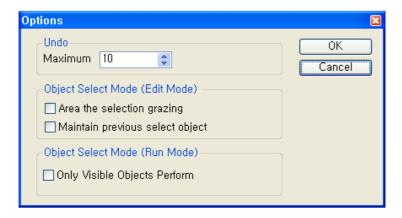


(4) Graphic Runtime Settings

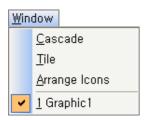
It is used to edit settings during the Graphic Runtime such as window style, execution location, title bar or start page. If this command is executed, a "Graphic Runtime Environment Setting" dialog box appears. For detailed explanation, refer to the "Execution Settings" provision.

(5) Options

The Graphic Editor/Runtime environments can be set up such 'the maximum number to undo' or 'Object Select Mode.'



9.9 **Window**



(1) Cascade

Windows are arranged in a cascade style.

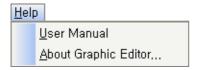
(2) Tile

Windows are arranged in a tile style.

(3) Arrange Icons

Windows are arranged to the location to designate various tool bar defaults.

9.10 Help

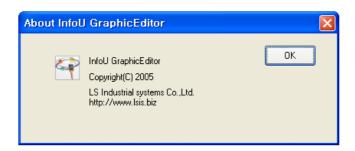


(1) User Manual

It is used to show 'Help' provisions. Step-by-step commands and instructions on the displayed provisions are presented. A content button is clicked in the Help screen, the corresponding provision screen appears.

(2) About Graphic Editor

It is used to show the version information of the graphic editor. If this command is executed, version information and copyright message appear.



9.11 Insert Image

A graphic file format refers to the type of a file that contains graphic images. A graphic file can either lower or extend its image capacity and have a different file format depending on how the graphic images are compressed and saved. This InfoU Graphic Editor supports general-purpose graphic file formats.

(1) Insert Image

If the user selects [Insert] [Insert] [Insert] [Insert] [Insert] [Insert] Insert] [Insert] [I screen page, the 'Open' dialog box appears. The user may select an image file to insert in the displayed dialog box. The image file types that are supported by the Graphic Editor are BMP, JPG, GIF, PNG, TIF, WMF, EMF and etc. The user may also use the clip board to insert images by copying images from external APP and pasting them. Image names and formats can be checked and methods to save images, transparency degree and colors can be designated in 'Image Object Properties.' For GIF animation image objects, a time cycle of each frame can be set up.



(2) Difference between Bit Map and Vector

(a) Bit Map

Bit map refers to a way to divide one image into several points when saving images. With this way, the user may use various colors and produce vivid colors and flexible pictures like photograph. However, it has some limitations to process delicate lines or shapes and the resolution becomes rough or deteriorated by zooming in/out. Because information is processed in pixel unit, file size is relatively large.

(b) Vector

Vector refers to a way to divide one image into several lines and shapes when saving images. This way is proper when expressing simple shapes and objects and it enables to express delicate lines or forms effectively. Zooming in/out does not have any influence on the resolution. However, available colors are limited and flexible works like photograph can not created in this way.

(3) Characteristics of each graphic file format

(a) BMP

As a format to save uncompressed bit map images, this file format is used basically in Windows and OS/2. Logo pictures displayed when the window starts or ends or background pictures in the wallpaper are all BMP-type files. Since BMP saves image data inefficiently, a file with larger capacity than the actually needed one is created. Both RGB mode and Indexed mode are supported and BMP-type files can be compressed in RLE compression method. Layers and alpha channels are not supported.

(b) JPG

It is not the file format that can be used only in particular software or platforms rather, as a data encoding method in which data files can be downloaded only after their size is lowered, this method has significant bigger merit than compression loss and is the one used the most when saving image files among other image saving methods. As one of the graphic file formats used the most on the Internet, its compression rate is the most excellent method. To compress image files, this method deletes overlapped information in the frame to compress color static image data. JPG can be applied to any picture regardless of image contents or resolution and it can also make it possible to use sequential decoding and gradual decoding that gradually decodes the entire picture. Unlike GIF format, JPEG maintains all of color information of RGB image. However, even though it has a high compression rate because it conducts loss compression, the quality becomes significantly deteriorated as compression increases. About 11.2MB is required when saving 24 bit color images, however, the file size lowers to 1.4MB if it is compressed in ratio of 8 to 1. This format is considered almost as a standard format on the Internet and file size is very low about 1/10 of that of other formats when saving the same number of colors under similar definition.

(c) GIF

As a graphic format developed by CompuServe, a leading PC computer company in the USA, it can be used in various device types for smooth interchange of graphic information between different device types. GIF files lower image capacity by reducing the number of colors so that this file format has remarkable compression rates to obtain high resolution images by reducing important transfer time. When saving, this method sorts 256 colors mostly used in the overall image to create a color index and the entire images are expressed in those 256 colors. Therefore, the file's capacity remarkably decreases. However, as its biggest shortcoming, the available number of colors may vary depending on the version of this format so that up to 256 colors are supported in version 87a. This way is rarely used for image saving today because a format with 256 colors is the smallest one. Therefore, it is mainly used when producing website icons. Gif 89a Format is mostly used for creating moving animation or files with transparent background.

(d) PNG

As an abbreviation word of Portable Network Graphics, it is introduced to support interface functions and increase the compression rate as PEG does. Like GIF, this image format lowers the number of colors to compress images but, unlike GIF which limits up to 256 colors, it can save 16,000,000 colors and provides more excellent compression rates of 10~30% than those of GIF. Basically, it supports true colors, uses no-loss compression and enables to express original images as they are without image deterioration on the web. PNG provides improved functions in the alpha channel related to image's transparency and supports 256 gray colors and 256 colors as well as true colors. This format can be called an emerging web image format after GIF and JPEG. PNG is currently used in Netscape and Explorer.

(e) TIF

TIFF means an image file format with a tag and it is co-developed by Aldus Corp and Microsoft Corp. This file format is used when files are exchanged between application and computer platform and even though most files are saved without being compressed under this format, it reduces file capacity most significantly by using a "no-loss compression method." Basically, it does not depend on OS so that it can describe resolution and compression methods and as a format to preserve image data from black and white to various colors, it is used to preserve dotexpression images. Most of scanner software or graphic software support this TIFF type and it is mainly used for

scanning, saving or sending image files expressed in gray scale. It seems passable to think this format is used for image exchange in Macintosh and Window.

(f) WMF

As an abbreviation word of Windows Meta File, this is the format to support the vector method -drawing programs. As a common format of drawing programs, it is supported by Microsoft Office as clip arts. Not only it can enlarge into any size freely without lowering resolution but also this format is supported by many useful programs. This format also expresses graphic information displayed in the window screen with GDI Function.

(g) EMF

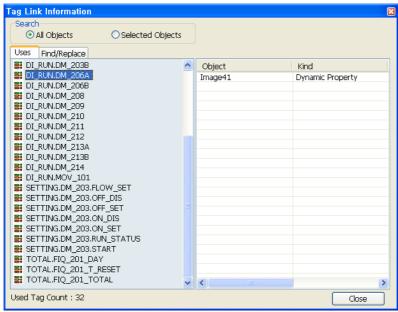
As an abbreviation word of Enhanced Meta File, this can provide enhanced 32bit definition even though it is the same as WMF.

9.12 Tag Link

This command is used when finding mapped tags among the entire objects or the selected objects in the page or replacing the found tags with others. The user may define a range to be searched by selecting either 'All Objects' or 'Selected Objects.'

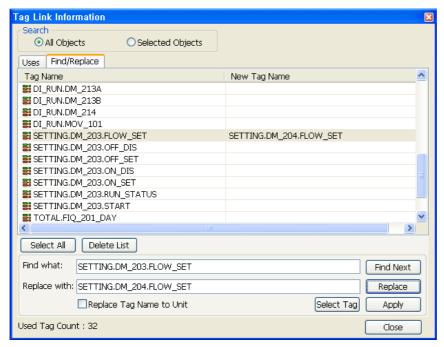
(1) Uses

The left area under 'Uses' tap shows a tag list organized through selecting objects. The right area shows objects used in each tag and a list of dynamic properties.



(2) Find/Replace

The left area under 'Find/Replace' tab shows the names of the tags organized through selecting objects and it is used when tag link is re-organized for all of the selected objects. Preview is available through single or multiple selection and it is also possible to copy with the clipboard to edit in a Excel file and paste.



(a) Find What

When replacing the selected tag name, directly input a name of the new tag or click 'Select Tag' button and then input it in the 'Select Tag' dialog box.

(b) Select All

All of the tag lists are selected.

(c) Delete List

The selected tag lists are cancelled.

(d) Replace

Click 'Replace' button to check and inspect the planned replace.

(e) Apply

Click 'Apply' button to link the select tag link as seen in the displayed list.

9.13 Library

In addition to the library basically provided by the system, the user may register/delete/edit his/her own library. One of the strong points of this "Library Object" command is that several objects can be selected at the same time to be saved in the library list and at this time, the properties set up in those objects are also saved.

(1) Use Symbol Objects

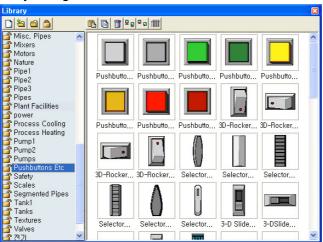
Any object registered in the library can be inserted in the screen. The user may drag any object he/she wants to edit in the library and drop to the page he/she is editing with the left button of the mouse or the user may copy relevant symbols and paste them to the edit page to insert objects. Then, these objects can be reformed to a desired shape through editing.

(2) Register Symbol Objects

In addition to the library basically provided by the system, the user may add his/her own library. To register an object, the user can select an object in the edit page and drag and drop it to the Library dialog box while pressing 'Alt 'key or the user can copy the relevant symbol and paste it to a desire location in the Library dialog box.

When creating a new group, the user needs to input a group name. Library data are saved in location 'InfoU Installation Path\bin\PicLibs' in file type 'Group Name.dat.'

(3) Library Dialog box



(a) New Name

It is used to create a new library group. If the name of an image to be registered in the library is inputted, the image with the new name is added to the library group.

(b) Open

It is used to open one of existing library groups. The opened library group is added to the list.

(c) Close

The relevant library group is closed. The closed library group is deleted from the list.

(d) Lock

If the relevant library group is locked, any symbol is not added nor deleted from the relevant library group.

(e) Copy/Paste/Delete

It is available to copy, paste or delete symbol objects from the library dialog box.

(f) Preview

The user may preview symbol objects in icon, name or icon & name type.

9.14 Layer

By using layers such as Show/Hide, Lock and Declutter function in the graphic editor, the user may simplify complicate process screen works into individual object processes. The contents in the process screen can be divided up to eight layers and Layer0 objects are located on the lowest layer while Layer7 objects are located on the top.

(1) Layer Settings

In default, all of the layers can be seen and Layer O is activated. Objects can be always added to the activated layer but, they can be quickly moved to other layer. The user may change to assign objects to layers by using the Object Property screen. If the user needs to move multiple objects in different layers to one layer, he/she may use multiple-selection to move them quickly. To change active layers, the user may use the Layer combo box.



(2) Layer Lock/Unlock

The layer lock function makes selecting and editing objects either possible or impossible. By using Lock button, the user may lock the relevant layer and use it as background image.

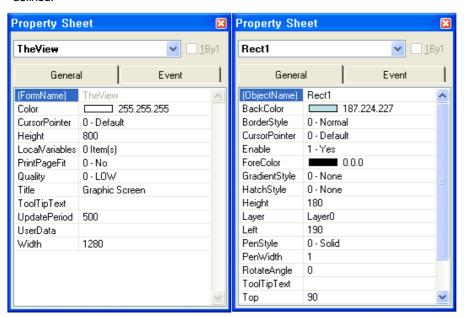
(3) Layer Show/Hide

By using the Layer Palette, the user may show or hide the relevant layer individually. That is, it is possible to show/hide the objects assigned to the relevant layer.

9.15 **Object Property**

(1) General

Static characteristics of each object such as page and object size, shape, location and color can be defined.



(a) Page Property

Property	Description
(FormName)	Name of page object
Color	Page background color
CursorPointer	Cursor shape when the mouse is on an object
Height	Page height
Width	Page width
PrintPageFit	Designate a rate to align print paper and page
Quality	Designate the quality level of an object to be drawn
Tag	Save data defined by the user
Title	Page name
ToolTipText	Tooltip when the mouse is on an object
UpdatePeriod	Screen update period during Runtime
LocalVariables	Designate variables used in the page

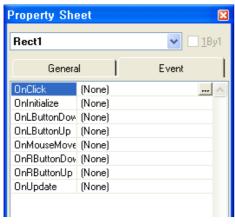
(b) Object Property

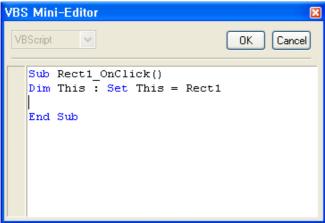
Property	Description
(ObjectName)	Object name
Left	Left location of an object
Тор	Top location of an object
Width	Object width
Height	Object height
CursorPointer	Cursor shape when the mouse is on an object
ToolTipText	Tooltip when the mouse is on an object
Enable	Designate to assign event handling of an object
Visible	Designate whether to show an object
Tag	Save data defined by the user
BackColor	Object background color
ForeColor	Object line or letter color
Font	Designate font name, style, size
Layer	Designate layer where an object exists
RotateAngle	Designate a rotation angle of an object
PenWidth	Designate line width
HatchStyle	Designate a hatch pattern to fill
BorderStyle	Normal, 3D, Sunken or Raised.
GradientStyle	Designate a gradient pattern to fill
PenStyle	Designate a line style
Arrowhead	Designate arrow direction
ArrowheadHeight	Designate arrow size
Text	Designate text to be displayed
Alignment	Designate text alignment
Angle1	Designate the start angle
Angle2	Designate the end angle
BorderColor	Object border color
ImageName	Name and format type of the displayed image
ImageInclude	Designate a type to save images

(2) Event Property

The user may use scripts when designating complicate dynamic properties defined by the user or executing functions other than the predefined movements. 'Script Engine' provided by Microsoft Corp is used and it is very similar to visual basic programming.

If 'Event' tab is selected in the Object Sheet and an event to be edited is selected, VBS Mini-Editor, a scripter editor, appears.





Event	Description
OnInitialize	Event that takes place when the relevant page is open(For setting-up)
OnTerminate	Event that takes place when the relevant page is closed(For clean up)
OnClick	Event that takes place when the user clicks on an object
OnDblClick	Event that takes place when the user double clicks on an object
OnLButtonDown	Event that takes place when the user clicks the left of the mouse on an object
OnRButtonDown	Event that takes place when the user clicks the right of the mouse on an object
OnMouseMove	Event that takes place when the user moves the mouse
OnLButtonUp	Event that takes place when the user moves up an object with the left mouse
OnRButtonUp	Event that takes place when the user moves up an object with the right mouse
OnUpdate	Event that takes place when screen updates regularly.

9.16 **Define Dynamic Properties**

Dynamic Properties are used when setting up dynamic properties and action functions of the relevant object. First, select an object and execute this command to show a dialog box to set up its dynamic properties. The displayed dialog box allows the user to set up dynamic properties and action functions of the relevant object.

9.16.1 **Horizontal Size**

A function to change the object's size in the horizontal direction according to tag values

(1) Tag Name

Input a tag name or expression that will be a baseline of horizontal size. A tag name can be designated with 'Search' (...) on the right.

(2) Direction

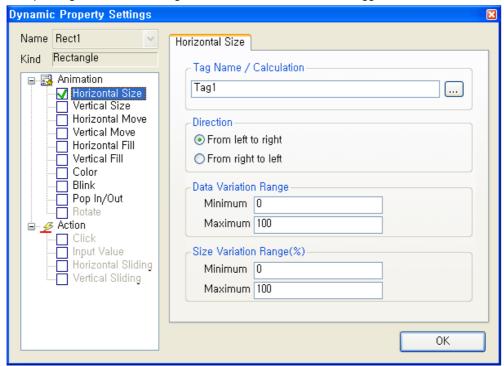
Set up a base location and direction of change in size.

(3) Data Variation Range

Set up a range of data to be referred when changed. The minimum value can not be bigger than the maximum one.

(4) Size Variation Range

Set up a range of size to be changed. The minimum size can not be bigger than the maximum one.



9.16.2 **Vertical Size**

A function to change the object's size in the vertical direction according to tag values

(1) Tag Name

Input a tag name or expression that will be a baseline of vertical size. A tag name can be designated with 'Search' (...).

(2) Direction

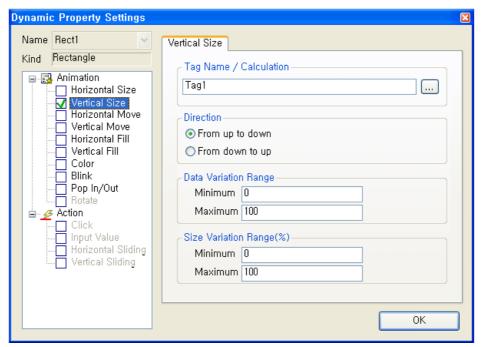
Set up a base location and direction of change in size.

(3) Data Variation Range

Set up a range of data to be referred when changed. The minimum value can not be bigger than the maximum one.

(4) Size Variation Range

Set up a range of size to be changed. The minimum value can not be bigger than the maximum one.



9.16.3 **Horizontal Move**

A function to move the object in the horizontal direction according to the designated tag value

(1) Tag Name

Input a tag name or expression that will be a baseline of horizontal move. A tag name can be designated with 'Search' (...) on the right.

(2) Direction

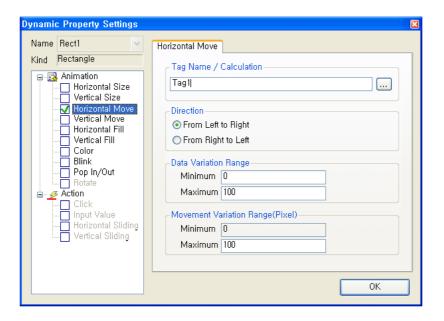
Set up a base location and direction of change in move.

(3) Data Variation Range

Set up a range of data to be referred when changed. The minimum value can not be bigger than the maximum

(4) Move Variation Range

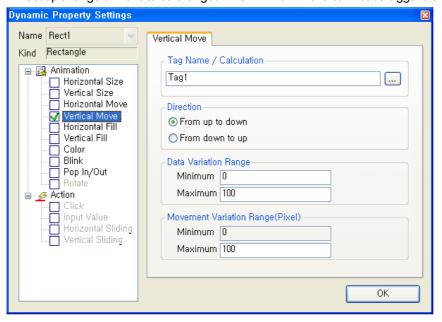
Set up a range of move to be changed. The minimum move can not be bigger than the maximum one.



9.16.4 Vertical Move

A function to move the object in the vertical direction according to the designated tag value

- (1) Tag Name
 - Input a tag name or expression that will be a baseline of vertical move. A tag name can be designated with 'Search' (...) on the right.
- (2) Direction
 - Set up a base location and direction of change in move.
- (3) Data Variation Range
 - Set up a range of data to be referred when changed. The minimum value can not be bigger than the maximum one.
- (4) Move Variation Range
 - Set up a range of move to be changed. The minimum move can not be bigger than the maximum one.



9.16.5 **Horizontal Fill**

A function to fill in the object in the horizontal direction according to the designated tag value

(1) Tag Name

Input a tag name or expression that will be a baseline of horizontal fill. A tag name can be designated with 'Search' (...) on the right.

(2) Direction

Set up a base location and direction of change in fill.

(3) Data Variation Range

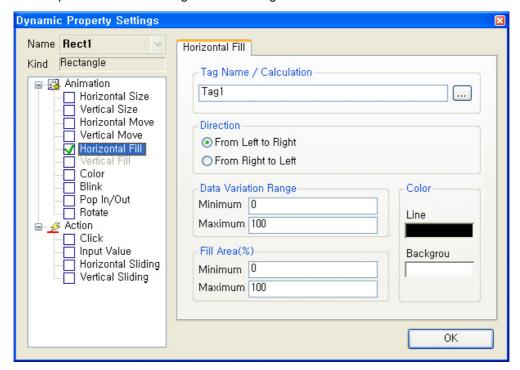
Set up a range of data to be referred when changed. The minimum value can not be bigger than the maximum one.

(4) Fill Area (%)

Set up an area to fill when changed. The start area can not be bigger than the maximum one.

(5) Color

Set up colors of line and background when filling.



9.16.6 **Vertical Fill**

A function to fill in the object in the vertical direction according to the designated tag value

(1) Tag Name

Input a tag name or expression that will be a baseline of vertical fill. A tag name can be designated with 'Search' (...) on the right.

(2) Direction

Set up a base location and direction of change in fill.

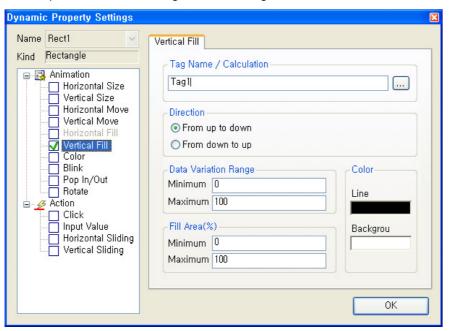
(3) Data Variation Range

Set up a range of data to be referred when changed. The minimum value can not be bigger than the maximum one.

(4) Fill Area (%)

Set up an area to fill when changed. The start area can not be bigger than the maximum one.

Set up colors of line and background when filling.

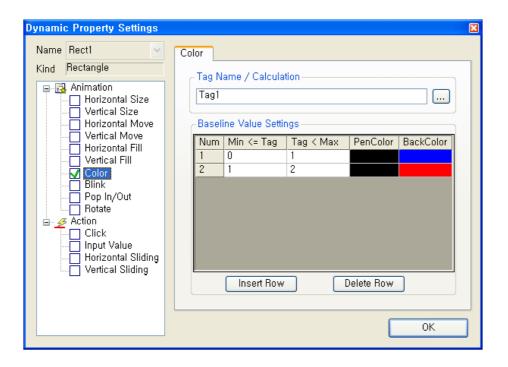


9.16.7 Color

A function to change the object in the prescribed color according to the inputted tag value

- (1) Tag Name
 - Input a tag name or expression that will be a baseline of color change. A tag name can be designated with 'Search' (...) on the right.
- (2) Baseline Value Settings

When colors are changed as new rows are inserted to grids, input the minimum and maximum value of the inputted tag or expression and set up colors of line and background to display under the relevant condition.

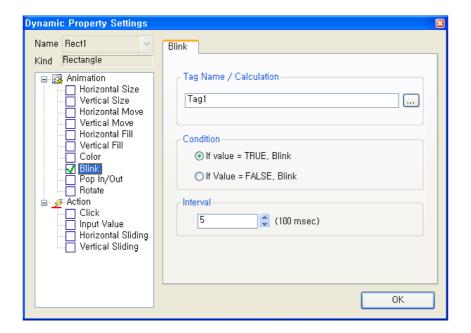


9.16.8 Blink

A function to blink the object in the specified interval according to the tag status

- (1) Tag Name
 - Input a tag name or expression that will be a baseline of blink. A tag name can be designated with 'Search' (...) on the right.
- (2) Condition
 - (a) True: If the tag value designated above is true, blink the object. If false, do not blink it.
 - (b) False: If the tag value designated above is false, blink the object. If true, do not blink it.
- (3) Interval

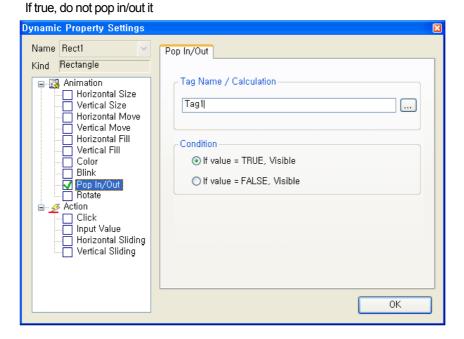
Select a blink interval. Its unit is 100msec. That is, if 1 is inputted, it blinks every 100msec and if 10 is inputted, it blinks every second. Input one value between 1 and 999.



9.16.9 Visible

A function either to show or hide the object according to the input tag value

- (1) Tag Name
 - Input a tag name or expression that will be a baseline of pop in/out. A tag name can be designated with 'Search' (...) on the right.
- (2) Condition
 - (a) True: If the tag value designated above is true, pop in/out the object. If false, do not pop in/out it
 - (b) False: If the tag value designated above is false, pop in/out the object.



9.16.10 Rotate

A function to rotate the object by the designated angle degree according to the input tag value

(1) Tag Name

Input a tag name or expression that will be a baseline of rotation. A tag name can be designated with 'Search' (...) on the right.

(2) Direction

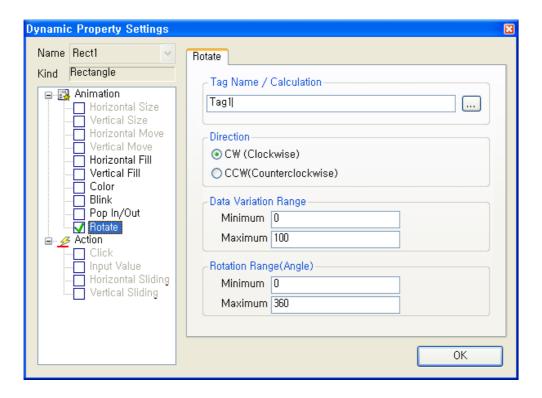
Set up a rotation direction when rotation is changed.

(3) Data Variation Range

Set up a range of data to be referred when changed. The minimum value can not be bigger than the maximum one.

(4) Rotation Range (Angle)

Set up a range of rotation to be changed. The minimum angle can not be bigger than the maximum one.



9.16.11 Display Value

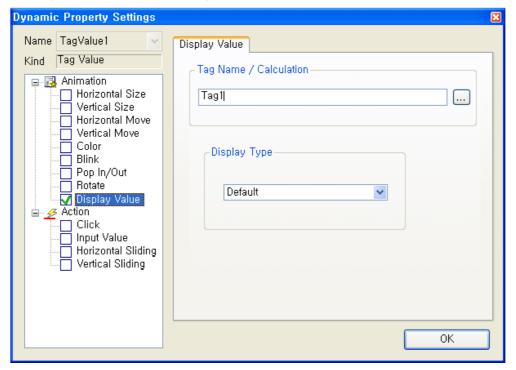
It is used to create an object marked with its tag value.

(1) Tag Name

Input a tag name or expression. A tag name can be designated with 'Search' (...) on the right.

(2) Display Type

A shape marked with values can be previewed in the screen.



9.16.12 Display String

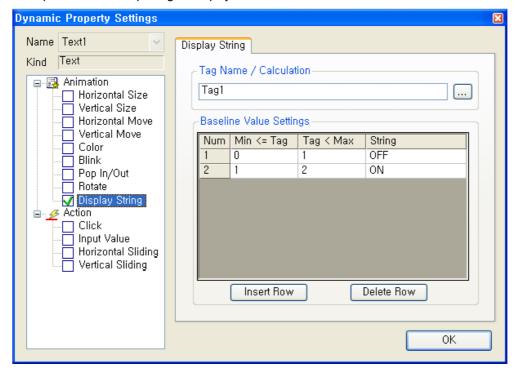
A function to change the object to the prescribed string according to the inputted tag value

(1) Tag Name

Input a tag name or expression that will be a baseline to change strings. A tag name can be designated with 'Search' (...) on the right.

(2) Baseline Value Settings

When strings are changed as new rows are inserted to grids, input the minimum and maximum value of the inputted tag or expression and set up strings to display under the relevant condition.



9.16.13 GIF Animation

A function to execute GIF animation according to the inputted tag values

(1) Tag Name

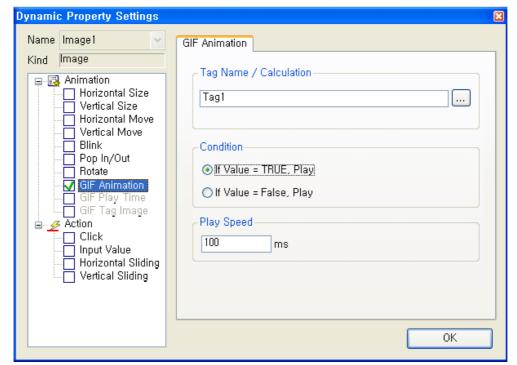
Input a tag name or expression that will be a baseline to execute GIF animation. A tag name can be designated with 'Search' (...) on the right.

(2) Condition

- (a) True: If the tag value designated above is true, execute GIF animation. If false, stop executing GIF animation.
- (b) False: If the tag value designated above is false, execute GIF animation. If true, stop executing GIF animation.

(3) Play Speed

Designate a play speed between frames.



9.16.14 GIF Play Time

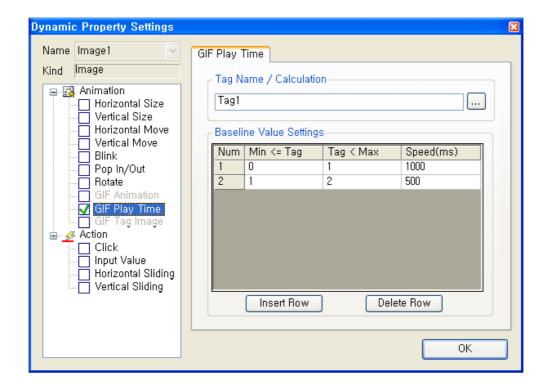
A function to designate a play speed of each GIF animation frame according to the inputted tag value.

(1) Tag Name

Input a tag name or expression that will be a baseline of GIF play time. A tag name can be designated with 'Search' (...) on the right.

(2) Baseline Value Settings

When strings are changed as new rows are inserted to grids, input the minimum and maximum value of the tag or expression that will be a baseline of conversion and set up a play speed under the relevant condition.



9.16.15 GIF Tag Image

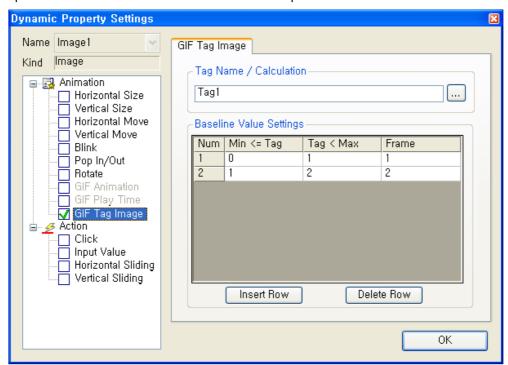
A function to show each GIF animation frame according to the inputted tag value.

(1) Tag Name

Input a tag name or expression that will be a baseline of showing each GIF animation frame. A tag name can be designated with 'Search' (...) on the right.

(2) Baseline Value Settings

When strings are changed as new rows are inserted to grids, input the minimum and maximum value of the tag or expression that will be a baseline of conversion and set up a frame to show under the relevant condition.

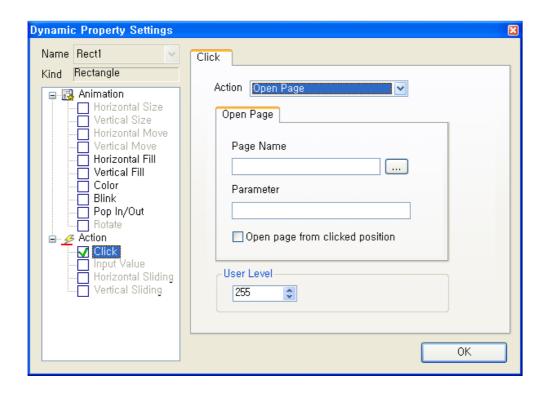


9.16.16 Click

A control function to execute the defined actions when clicking the object with the mouse or keyboard

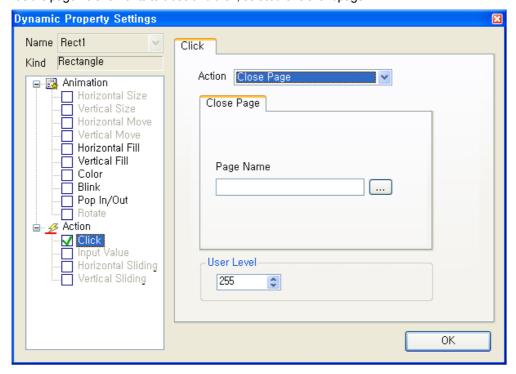
(1) Open Page

The user may directly input a name of the page he/she wants to open or use 'Search' (...) button to move the folder that has the page he/she wants to open and select the relevant page. When opening the page, the user can deliver parameter and open the page from the clicked position.

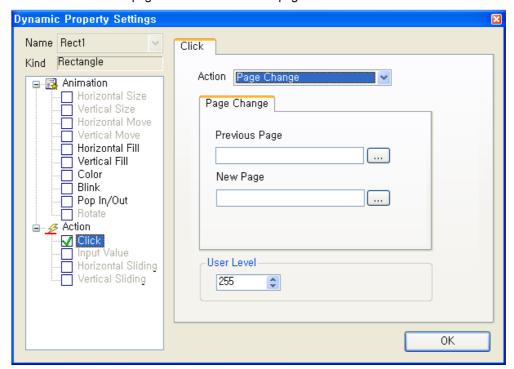


(2) Close Page

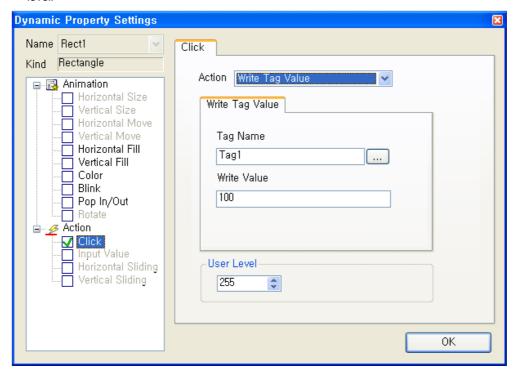
The user may directly input a name of the page he/she wants to close or use 'Search' (...) button to move the folder that has the page he/she wants to close and then, select the relevant page.



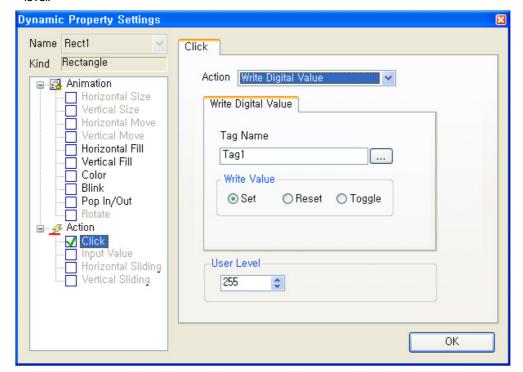
- (3) Page Change
- (a) Previous Page: The user may directly input a name of the page to be closed or use 'Search' (...) button to move the folder that has the page and select the relevant page.
- (b) New Page: The user may directly input a name of the page to be newly displayed or use 'Search' (...) button to move the folder that has the page and select the relevant page.



- (4) Write Tag Value
- (a) Tag Name: Directly input a tag name or use 'Search' (...) on the right to designate a tag name.
- (b) Write Value: Input a value to write.
- (c) User Level: Select one between level 1 and level 255 to limit the user's authority and the lower number has the higher level.



- (5) Write Digital Value
- (a) Tag Name: Directly input a digital tag name or use 'Search' (...) on the right to designate a tag
- (b) Write Value: 'Set' designates the value of the relevant digital tag as 1. 'Reset' designates the value as 0. 'Toggle' compares the value of the current digital tag and designates the value as 1 if the result is 0 and the value as 0 if the
- (c) User Level: Select one between level 1 and level 255 to limit the user's authority and the lower number has the higher level.



9.16.17 Input Value

A function to select an object and input a value in the predefined dialog box with the mouse or Keyboard.

(1) Tag Name

Input a name of the tag to which the inputted value will apply. A tag name can be designated with 'Search' (...) on the

(2) Input Type

The user may choose either 'Number' or 'String' to decide how to input a value.

(3) Input Value Range

Designate the minimum and maximum value to input. The minimum value can not be bigger than the maximum one.

(4) Display Type

Designate a type of the window screen to be displayed in the screen.

(5) Title

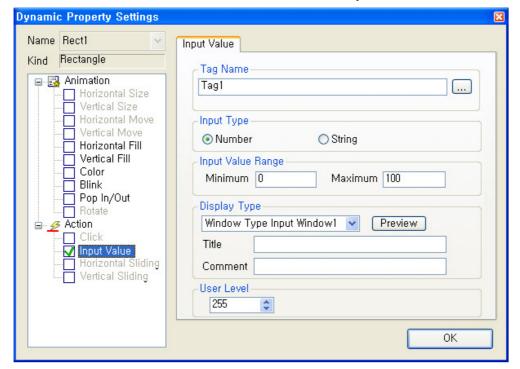
Input a title of the screen.

(6) Comment

Input comment on the window.

(7) User Level

Select one between level 1 and level 255 to limit the user's authority and the lower number has the higher level



9.16.18 Horizontal Drag

A function to drag an object in the horizontal direction with the mouse to convert coordinate values to tag values.

(1) Tag Name

Input a tag name to which the value of horizontal sliding will be inserted. A tag name can be designated with 'Search' (...) on the right.

(2) Direction

Set up a base location and direction of horizontal sliding.

(3) Data Variation Range

Set up a range of data to be referred during the horizontal sliding. The minimum value can not be bigger than the maximum one.

(4) Drag Variation Range (Pixel)

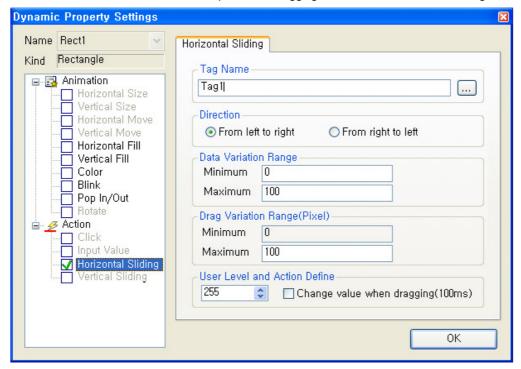
Set up a range of move during the horizontal sliding. The minimum move can not be bigger than the maximum one.

(5) User Leve

Select one between level 1 and level 255 to limit the user's authority and the lower number has the higher level

(6) Change value when dragging (100ms)

Check $\sqrt{}$ in the box if the user wants to operate the dragging value while the mouse is moving.



9.16.19 Vertical Drag

A function to slide an object in the vertical direction with the mouse to convert coordinate values to tag values.

(1) Tag Name

Input a tag name to which the value of vertical sliding will be inserted. A tag name can be designated with 'Search' (...) on the right.

(2) Direction

Set up a base location and direction of vertical sliding.

(3) Data Variation Range

Set up a range of data to be referred during the vertical sliding. The minimum value can not be bigger than the maximum one.

(4) Drag Variation Range (Pixel)

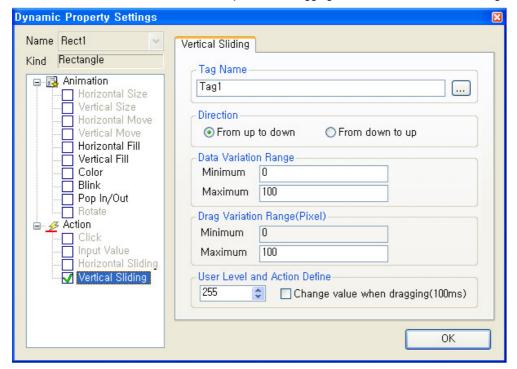
Set up a range of move during the vertical sliding. The minimum move can not be bigger than the maximum one.

(5) User Level

Select one between level 1 and level 255 to limit the user's authority and the lower number has the higher level

(6) Change value when dragging (100ms)

Check $\sqrt{}$ in the box if the user wants to operate the dragging value while the mouse is moving.

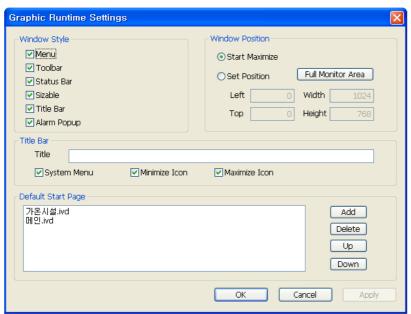


9.17 Runtime Settings

9.17.1 Graphic Runtime Settings

'Runtime settings' is to set up the environment for the graphic runtime (InfoUX.exe).

To set up the environment, the user can set up menu, view/hide toolbar, initial execution position, name to be displayed in the title bar, view/hide window control buttons and other options and also set up a default start page to open simultaneously along with execution.



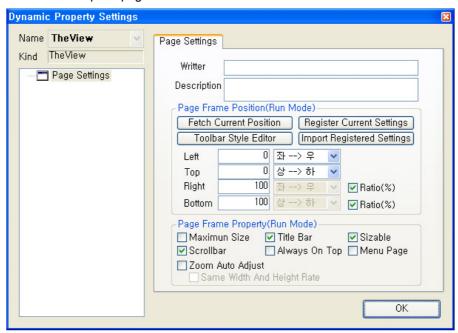
- (1) Window Style
 - (a) Menu: Select whether to view the basic menu.
 - (b) Toolbar: Select whether to view the toolbar.
 - (c) Status Bar: Select whether to view the status bar
 - (d) Sizable: Select whether to allow the operator to change the screen size.
 - (e) Title Bar: Select whether to display the title window located on the top of the window.
- (2) Window Position
 - (a) Start Maximize: Select whether to execute RunTime in the maximum screen.
 - (b) Set Position: The user can designate the location and size of the screen during Run Time.

 Click 'Full Monitor Area' button, the area coordinates for the multiple monitors are displayed. The coordinates reference (0, 0) is for the main monitor.
- (3) Title Bar
 - (a) Title: Input the title of the screen.
 - (b) System Menu: Select whether to display system menus in icon types on the left top of the screen.
 - (c) Minimize Icon: Select whether to display minimized buttons on the right top of the screen
 - (d) Maximize Icon: Select whether to display maximized buttons on the right top of the screen
- (4) Default Start Page

This is for selecting screens to be displayed at first when executing Run Time. If 'Add' button is pressed, all of the written pages for the current page are listed up and the user may select one to be used as a start page.

9.17.2 Page Settings

The page settings are used to designate a view (size and position, etc) when each graphic screen is displayed. In order to set up the page environment, click [View] [Object property] of the menu in the screen to display the 'page settings' screen. The screen to set up the page environment is as follows:



(1) Writer and Description

Input the writer of the relevant graphic file and describe on the file. (Optional and informative)

(2) Fetch Current Position

Calculate the location and size during the editing in the graphic editor and display them on the location/size input box.

(3) Toolbar Style Editor

It provides the editing window to set up the page in a simple toolbar style.

(4) Register Current Settings

Register and save the current setting value in the page setting window.

(5) Import Registered Settings

Import the registered and saved page setting value to the page setting window.

(6) Left (Start X Position)

Show coordinates X in which the relevant page is located when executed. For the base of coordinates X, designate the number of pixels starting from the area displayed in the screen (except the left frame area in the window) out of the main window of the graphic runtime program.

(7) Top (Start Y Position)

Show coordinates Y in which the page is located when executing. For the base of coordinates Y, designate the number of pixels starting from the area displayed in the screen (any location other than caption bar, menu bar and tool bar) out of the main window of the graphic runtime program.

(8) Right

Designate pixel/ratio and width/right of the page when executing.

(9) Bottom

Designate pixel/ratio and height/bottom of the page when executing

(10) Title Bar

Set up whether to show/hide the caption bar (title bar) of the page when executing.

(11) Maximum size

Show the page in a full screen of the runtime frame when executing.

(12) Sizable

Make the size of the page adjustable. When it is adjustable, the size of the window can be changed by clicking and dragging the edge of the window, and a thick window frame is used. If the size adjustment is not available, the size of the page cannot be changed, and a thin window frame is shown.

(13) Scroll bar

Set up whether to show/hide the scroll bar of the page when executing.

(14)Always On Top

If 'Always on Top' option is selected, the window is always located on top among various graphic pages and it exists above other windows (among graphic monitoring pages).

(15) Menu page

Designate the relevant page as a menu page when executing. The file designated as menu page is not closed and continuously maintained even close (menu, toolbar) command is executed.

(If the user wants to close the page forcibly, he/she may execute 'closing' while pressing shift key.)

(16) Zoom Auto Adjust

The zoom ratio (zoom in/out) is automatically adjusted (fit to the screen) by width and length ratio of the screen when opening the page.

(17) Same Width and Height Rate

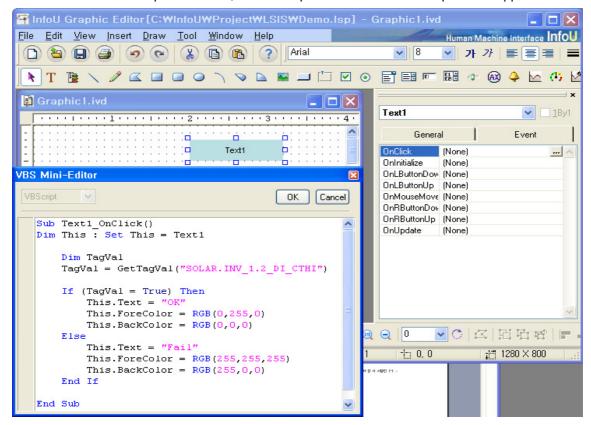
When opening the page, the zoom ratio (zoom in/out) is automatically adjusted (fit to the screen) by maintaining the ratio of width and height.

9.18 **Graphic Script**

This Graphic Script is a function to define the activities for the object event (user's mouse action) or periodic execution (OnUpdate Function: Executed at every set-up time in the screen UpdatePeriod) as VB Script.

The available commands at this time are the ones provided by the standard Microsoft VB Script Engine and the graphic functions exclusively available in the InfoU system.

For the Microsoft VB Script commands, see the help of Microsoft or "VB Script" of the Appendix A.



9.18.1 InfoU Graphic Function List

The functions available at the script of this graphic editor are composed of the standard VB script functions and the InfoU graphic functions. The InfoU graphic functions are as seen in the following table:

(1) List by Item

(a) Function related to screen

Function Name	Mode				
ChangePage	long ChangePage(BSTR OldPageName, BSTR NewPageName)				
ClosePage	long ClosePage(BSTR PageName)				
HardCopy	long	HardCopy()			
MsgTrace	void	MsgTrace(VARIANT MessageValue)			
OpenPage	long	OpenPage(BSTR PageName, BSTR TagParam)			
OpenDagoEy	long	OpenPageEx(VARIANT X,VARIANT Y,			
OpenPageEx	VARIA	VARIANT cX, VARIANT cY, BSTR PageName, BSTR TagParam)			
Onon Dago Object	long	OpenPageObject(BSTR PageName, BSTR TagParam, BSTR			
OpenPageObject	ObjectName)				
Reload	long	ReLoad(BSTR PageName, BSTR TagParam)			
ShowAlarmPage	long	ShowAlarmPage(BSTR AlarmTag)			
ShowHelp	long	ShowHelp(BSTR HelpFile, long idx)			
ShowTagView	void	ShowTagView(BSTR TagName)			
TimeOutMsgBox	void	TimeOutMsgBox(BSTR Msg, BSTR Title, long nSec)			
FindPage	VARIANT FindPage(LPCTSTR PageName)				
IsPageOpen	BOOL IsPageOpen(LPCTSTR PageName)				
SendData	BOOL SendData(LPCTSTR PageName, VARIANT FAR& VariantData)				

(b) Function related to tag

Function Name	Mode
GetTagOFFSTR	BSTR GetTagOFFSTR(BSTR TagName)
GetTagONSTR	BSTR GetTagONSTR(BSTR TagName)
GetTagVal	VARIANT GetTagVal(BSTR TagVarName)
GetTagValMAX	VARIANT GetTagValMAX(BSTR TagName)
GetTagValMIN	VARIANT GetTagValMIN(BSTR TagName)
PulseOff	void PulseOn(BSTR TagName, LONG MilliSec)
PulseOn	void PulseOff(BSTR TagName, LONG MilliSec)
SetTagVal	long SetTagVal(BSTR TagVarName, VARIANT Value,
Serragyai	VARIANT_BOOL bRet)
ShowCrossReference	void ShowCrossReference (BSTR TagName)

(c) Run Function

Function Name		Mode
AckAllAlarm	void	AckAllAlarm()
SetLogging	void	SetLogging(BOOL TrueFalse)
StopAlarmSound	long	StopAlarmSound()

(d) Function related to I/O driver

Function Name		Mode
ShowNetworkStatus	long	ShowNetworkStatus()
ShowRawDataInfo	long	ShowRawDataInfo()

(e) Function related to user

Function Name			Мо	ode	
GetUserID	BSTR	GetUserl	D()		
GetUserLevel	long	GetUserL	_evel()		
UserCommand1	long		mand1(long n0 3, BSTR Str4)	Cmd, BSTR Str1, BS	TR Str2,
UserLogOff	void	UserLog(Off(void)		
UserLogOn	BOOL	UserLog(On(BSTR UserII	D, BSTR PassWord)	
GetUserInfoList	LONG	GetUs	•	G IOutCount, VARIA UserName, VARIAN arrPasswo	
	arrUser	Level,			
			VARIANT	arrHPhoneNo,	VARIANT
	arrPhor	neNo,			
			VARIANT FA	R* arrMailAdd)	
HolidaySet	LONG	HolidayS	Set()		

(f) Function related to report

i dilotion related to report		
Function Name		Mode
ReportDatabase	long	ReportDatabase(BSTR ReportName)
ReportDatabaseEx	long	ReportDatabaseEx(BSTR ReportName, BSTR RefTime)
ReportDialog	long	ReportDialog()
ReportPrint	long	ReportPrint(BSTR ReportName)
ReportPrintEx	long	ReportPrintEx(BSTR ReportName, BSTR RefTime)

(g) Function related to system

Function Name	Mode		
ExecuteCommand	void	ExecuteCommand(BSTR PathName)	
Quit	long	Quit()	
QuitProject	long	QuitProject()	
RunScript	void	RunScript(BSTR ScriptName)	
ShowVirtualKeyboard	void	ShowVirtualKeyboard(void)	
Sleep	void	Sleep(long Milliseconds)	

(h) Function related to external device

Function Name	Mode		
PlaySound	void	PlaySound(BSTR WaveFile)	

(i) Function related to history

Function Name		Mode
GetLoggingModelList	LONG	GetLoggingModelList(LONG IOutCount,VARIANT
	arrModelName,	
		VARIANT arrModelDesc, VARIANT
	arrModelType,	
		VARIANT arrArchivingCycle,
		VARIANT arrSegmentPeriod)
GetLoggingTag	LONG	GetLoggingTag(LPCTSTR ModelName,LONG
	IOutCount,	
		VARIANT arrTagName, VARIANT
	arrTagType)	

(2) The list in the order of dictionary

Function Name	Mode			
AckAllAlarm	void AckAllAlarm()			
ChangePage	Long ChangePage(BSTR OldPageName, BSTR			
	NewPageName)			
ClosePage	long ClosePage(BSTR PageName)			
ExecuteCommand	void ExecuteCommand(BSTR PathName)			
GetLoggingModelList	LONG GetLoggingModelList(LONG IOutCount,VARIANT			
	arrModelName,			
	VARIANT arrModelDesc, VARIANT arrModelType,			
	VARIANT arrArchivingCycle,			
	VARIANT arrSegmentPeriod)			
GetLoggingTag	LONG GetLoggingTag(LPCTSTR ModelName,LONG			
	IOutCount,			
	VARIANT arrTagName, VARIANT			
O-4T OFFOTD	arrTagType)			
GetTagOFFSTR	BSTR GetTagOFFSTR(BSTR TagName)			
GetTagONSTR	BSTR GetTagONSTR(BSTR TagName)			
GetTagVal	VARIANT GetTagVal(BSTR TagVarName)			
GetTagValMAX	VARIANT GetTagValMAX(BSTR TagName)			
GetTagValMIN	VARIANT GetTagValMIN(BSTR TagName)			
GetUserID	BSTR GetUserID()			
GetUserInfoList	LONG GetUserInfoList(LONG IOutCount, VARIANT rrUserID,			
	VARIANT arrUserName, VARIANT arrDesc,			
	VARIANT arrPassword,VARIANT arrUserLevel,			
	VARIANT arrHPhoneNo, VARIANT			
	arrPhoneNo,			
	VARIANT FAR* arrMailAdd)			
GetUserLevel	long GetUserLevel()			
HardCopy	long HardCopy()			
HolidaySet	LONG HolidaySet()			
FindPage	VARIANT FindPage(LPCTSTR PageName)			
IsPageOpen	BOOL IsPageOpen(LPCTSTR PageName)			
MsgTrace	void MsgTrace(VARIANT MessageValue)			
OpenPage	long OpenPage(BSTR PageName, BSTR TagParam)			
OpenPageEx	long OpenPageEx(VARIANT X, VARIANT Y,			
	VARIANT cX, VARIANT cY, BSTR PageName, BSTR TagParam)			
OpenPageObject	long OpenPageObject(BSTR PageName, BSTR TagParam, BSTR ObjectName)			

Function Name		Mode
PlaySound	void	PlaySound(BSTR WaveFile)
PulseOff	void	PulseOn(BSTR TagName, LONG MilliSec)
PulseOn	void	PulseOff(BSTR TagName, LONG MilliSec)
Quit	long	Quit()
QuitProject	long	QuitProject()
Reload	long	ReLoad(BSTR PageName, BSTR TagParam)
ReportDatabase	long	ReportDatabase(BSTR ReportName)
ReportDatabaseEx	long	ReportDatabaseEx(BSTR ReportName, BSTR RefTime)
ReportDialog	long	ReportDialog()
ReportPrint	long	ReportPrint(BSTR ReportName)
ReportPrintEx	long	ReportPrintEx(BSTR ReportName, BSTR RefTime)
ResetStatistic	long	ResetStatistic(BSTR TagName)
RunScript	void	RunScript(BSTR ScriptName)
SendData	BOOL	SendData(LPCTSTR PageName, VARIANT FAR&
	Variant	•
SetLogging	void	SetLogging(BOOL TrueFalse)
SetTagVal	long	SetTagVal(BSTR TagVarName, VARIANT Value,
		VARIANT_BOOL bRet)
ShowAlarmPage	long	ShowAlarmPage(BSTR AlarmTag)
ShowCrossReference	void	ShowCrossReference (BSTR TagName)
ShowHelp	long	ShowHelp(BSTR HelpFile, long idx)
ShowNetworkStatus	long	ShowNetworkStatus()
ShowRawDataInfo	long	ShowRawDataInfo()
ShowTagView	void	ShowTagView(BSTR TagName)
ShowVirtualKeyboard	void	ShowVirtualKeyboard(void)
Sleep	void	Sleep(long Milliseconds)
StopAlarmSound	long	StopAlarmSound()
TimeOutMsgBox	void	TimeOutMsgBox(BSTR Msg, BSTR Title, long nSec)
UserCommand1	long	UserCommand1(long nCmd, BSTR Str1, BSTR Str2,
		BSTR Str3, BSTR Str4)
UserLogOff	void	UserLogOff(void)
UserLogOn	BOOL	UserLogOn(BSTR UserID, BSTR PassWord)

9.18.2 **Explanation of InfoU Graphic Function**

AckAllAlarm Function

(1) Description

ACK all alarms.

(2) Syntax

AckAllAlarm()

ChangePage Function

(1) Description

Change the graphic page. .

(2) Syntax

ChangePage(BSTR OldPageName, BSTR NewPageName)

OldPageName: Save the previous page. The previous page can be designated "abc.ivd" or "*." The page designated as the previous page will be closed, and the page designated in NewPageName will be opened. NewPageName: Designate a name of the page that will be newly opened.

If the user does not designate this dPageName (empty string), the current page will be closed.

(3) Reference

If the user designates "*" to this dPageName, all pages will be closed.

However, if the page designated to this dPageName is a menu page, it will not be closed.

The menu page can be designated on the 'Page settings' screen in this Graphic Editor.

ClosePage Function

(1) Description

Close the graphic page.

(2) Syntax

ClosePage (BSTR PageName)

PageName: Designate the page to be closed. It is available to designate with either "abc.ivd" or

(3) Reference

If the user designates with "*" in the PageName, all pages will be closed.

However, if the page designated to this PageName is a menu page, it will not be closed.

The menu page can be designated on 'Page settings' screen in this Graphic Editor.

FindPage Function

(1) Description

Find the designated page name from the opened pages.

(2) Syntax

FindPage(LPCTSTR PageName)

(3) Reference

The page pointer value is returned in the form of variant type.

GetLoggingModelList Function

(1) Description

It notifies the user on the list of models registered in the history.

(2) Syntax

LONG

 $GetLoggingModelList(LONG\ IOutCount, VARIANT\ arr ModelName,$

VARIANT arrModelDesc, VARIANT arrModelType,

VARIANT arrArchivingCycle, VARIANT arrSegmentPeriod)

(3) Reference(Example)

Sub Text2_OnClick()

Dim This: Set This = Text2

'Declare variable

Dim nCount 'Number of registered models

DimModelName()'Model NameDimModelDesc()'Model DescriptionDimModelType()'Model Type

Dim ArchivCycle() 'Collection cycle (return string: 20 seconds = 20ss, 30 minutes=30mm, 5 hours=05hh,

36 months = 36MM)

Dim SavePeriod() 'Save period

Dim ret

Call 'Graphic Script function (ret value is undefined)

ret = GetLoggingModelList (nCount, ModelName, ModelDesc, ModelType, ArchivCycle, SavePeriod)

Dim strMessage

'The call result into string ...

```
strMessage = "Model Name/Model Description/Model Type/collection cycle/save period"
strMessage = strMessage + vbCrlf
For idx = 0 To nCount -1
        strMessage = strMessage+ ModelName(idx)
        strMessage = strMessage+ "/"
        EditBox1.Text = ModelName(idx) ' Use it to search tag by model name.
        strMessage = strMessage+ ModelDesc(idx)
        strMessage = strMessage+ "/"
        strMessage = strMessage+ ModelType(idx)
        strMessage = strMessage+ "/"
        strMessage = strMessage+ ArchivCycle(idx)
        strMessage = strMessage+ "/"
        strMessage = strMessage+ SavePeriod(idx)
        strMessage = strMessage + vbCrlf
Next
'Check the result
MsgBox
                strMessage
```

GetLoggingTag Function

(1) Description

It notifies the user on the list of tags registered in the given history model.

(2) Syntax

End Sub

LONG GetLoggingTag(LPCTSTR ModelName,LONG lOutCount, VARIANT arrTagName, VARIANT arrTagType)

(3) Reference(Example)

Sub Text3_OnClick() Dim This: Set This = Text3

Dim nCount 'Number of registered models

Dim TagName() 'Tag name array Dim TagType() 'Tag type array

```
Dim ret
Dim modelName
modelName = EditBox1.Text
Call 'Graphic Script function (ret value is undefined)
ret = GetLoggingTag (modelName, nCount, TagName, TagType )
Dim strMessage
'The call result into string ...
strMessage = "Tag name/Tag type"
strMessage = strMessage + vbCrlf
For idx = 0 To nCount -1
        strMessage = strMessage+ TagName(idx)
        strMessage = strMessage+ "/"
        strMessage = strMessage+ TagType(idx)
        strMessage = strMessage + vbCrlf
Next
'Check the result
MsgBox
                 strMessage
```

End Sub

GetUserInfoList Function

(1) Description

Get the list of all users and individual information on each user.

The user information will be used for SMS call or mail sending.

(2) Syntax

LONG GetUserInfoList(LONG IOutCount, VARIANT rrUserID, VARIANT arrUserName, VARIANT arrDesc,

> VARIANT arrPassword, VARIANT arrUserLevel, VARIANT arrHPhoneNo, VARIANT arrPhoneNo, VARIANT FAR* arrMailAdd)

(3) Reference(Example)

Sub Text1_OnClick()

```
'Declare variable
Dim nCount
                         'Number of registered users
Dim UserIDs()
                         'User ID
Dim UserNames()
                          'User name
Dim Descs()
                         'User description
Dim Passwords()
                         'Password
Dim UserLevels()
                         'User level
Dim HPhoneNos()
                         'Mobile phone number
Dim PhoneNos()
                                 'Telephone number
Dim MailAdds()
                         'Mail address
Dim ret
Call 'Graphic Script function (ret value is undefined)
ret = GetUserInfoList (nCount, UserIDs, UserNames, Descs, Passwords,
                                                                              UserLevels, HPhoneNos,
PhoneNos, MailAdds)
Dim strMessage
'The call result into string ...
strMessage = "UserID/UserName/UserDesc/Password/UserLevel/HP No/TeINO/Mail"
strMessage = strMessage + vbCrlf
For idx = 0 To nCount -1
        strMessage = strMessage+ UserIDs(idx)
        strMessage = strMessage+ "/"
        strMessage = strMessage+ UserNames(idx)
        strMessage = strMessage+ "/"
        strMessage = strMessage+ Descs(idx)
        strMessage = strMessage+ "/"
        strMessage = strMessage+ Passwords(idx)
        strMessage = strMessage+ "/"
        strMessage = strMessage+ cstr(UserLevels(idx))
        strMessage = strMessage+ "/"
        strMessage = strMessage+ HPhoneNos(idx)
        strMessage = strMessage+ "/"
        strMessage = strMessage+ PhoneNos(idx)
        strMessage = strMessage+ "/"
```

Dim This: Set This = Text1

strMessage = strMessage+ MailAdds(idx)

strMessage = strMessage + vbCrlf

Next

'Check the result

MsgBox strMessage

End Sub

GetTagOFFSTR Function

(1) Description

Return 'off string' of the digital tag.

(2) Syntax

GetTagOFFSTR(BSTR TagName)

TagName: Designate a name of the tag.

(3) Reference

Return "off string "of the tag whose name is designated in TagName.

Off string of the tag is the string defined in the tag engineering, and it is only applied to the digital tag.

GetTagONSTR Function

(1) Description

Return On String of digital tag.

(2) Syntax

BSTR GetTagONSTR(BSTR TagName)

TagName: Designate a name of the tag.

(3) Reference

Return "on string" of the tag whose name is designated in TagName.

On string of the tag is the string defined in the tag engineering, and it is only applied to the digital tag.

GetTagVal Function

(1) Description

Return the tag value.

(2) Syntax

Variant GetTagVal(BSTR TagVarName)

TagVarName: Designate a name of the tag. The tag name includes the field name of the tag.

Ex: "TagVarName:FieldName"

The items listed in the tag selection dialogue can be used as the field name. For the details, see 'Tag variable' described in the next clause.

(3) Reference

Return the designated tag value in a variant type.

Since the value of the returned tag is in a variant type, the type of the returned value should be used for the inspected number in order to use this appropriately.

To inspect the type of the returned value, use IsNull(variant return value), IsEmpty(variant return value) or VarType(variant return value) functions. For more details on how to use these functions, see the MS VB Script manual or relevant appendix. If the type of returned variant value is null, it means it is in a bad status, that is, the tag value is not normal. If it is empty, it means there is no tag, tag property does not exist or it cannot get any tag value because of the system error.

GetTagValMAX Function

(1) Description

Return the maximum value of the tag.

(2) Syntax

Variant GetTagValMAX(BSTR TagName)

TagName: Designate a name of the tag.

(3) Reference

The maximum value of the designated tag is returned in a variant type.

GetTagValMIN Function

(1) Description

Return the maximum value of tag.

(2) Syntax

Variant GetTagValMAX(BSTR TagName)

TagName: Designate a name of the tag.

(3) Reference

The minimum value of the designated tag is returned in a variant type.

GetUserID Function

(1) Description

If it is called in the graphic runtime, return the ID of the user who is currently logged on.

(2) Syntax

BSTR GetUserID()

(3) Reference

Return the ID of the user logged on, and if there is no user logged on, the string with the length 0 is returned.

GetUserLevel Function

(1) Description

If it is called in the graphic runtime, return the priority of the user currently logged on.

(2) Syntax

LONG GetUserLevel ()

(3) Reference

The level of the user (As the priority of the user authority, there are level 1 through level 255) currently logged on is returned, and if there is no user logged on, it returns the default level 255 (the lowest one)

HardCopy Function

(1) Description

Print the entire screen in the main window area of the Graphic runtime program

(2) Syntax

HardCopy()

(3) Reference

Use the printer set up with default values.

HolidaySet Function

(1) Description

Show the setting screen window to set up holidays in the graphic runtime program.

(2) Syntax

HolidaySet()

IsPageOpen Function

(1) Description

Inspect whether the designated page name is opened in the graphic runtime program.

(2) Syntax

PageOpen(LPCTSTR PageName)

(3) Reference

Return whether the designated file name is opened.

MsgTrace Function

(1) Description

It is used when writing a command formula (script) in the graphic script, and checking the message for the purpose of checking error.

(2) Syntax

MsgTrace (VARIANT MessageValue)

(3) Reference(Example)

```
VBS Mini-Editor
                                 OK
                                      Cancel
   Sub Rect1_OnClick()
   Dim This : Set This = Rect1
       Dim ValTag
       ValTag = GetTagVal("TagName")
       MsgTrace ValTag
       MsgTrace "Debug Message"
   End Sub
```

```
InfoUX_d Graphic Message
 [ 2009/06/05 18:27:36 334 ]Open File
 File "C:\InfoU\Project\LSIS\GraLS\Menu.ivd" Does Not Exist.
[ 2009/06/05 18:27:36 334 ]Open File
File "C:\InfoU\Project\LSIS\GraLS\Main.ivd" Does Not Exist.
          All Clear
                                     Select Clear
                                                                          Scroll Stop
                                                                                                             Close
```

OpenPage Function

(1) Description

Open the graphic page.

(2) Syntax

OpenPage(BSTR PageName, BSTR TagParam)

PageName: Designate a name of the page to be opened.

TagParam: Designate tags to be replaced on the template screen. If there is no tag to be replaced, designate empty string (" ").

(3) Reference

OpenPage "Panel_GIPAM.ivd', "\$MACHINE1=WB_EHV04, \$MACHINE2=WB_EHV05'

While opening "Panel GIPAM.ivd" file, it converts the \$MACHINE1 tag or string into WB EHV04, and \$MACHINE2 tag or string into WB_EHV05.' The template screen can be operated by using this function.

OpenPageEx Function

(1) Description

Open the graphic page.

OpenPageEx(VARIANT X, VARIANT Y, VARIANT cX, VARIANT cY, BSTR PageName, BSTR TagParam)

(3) Reference

The usage method is same as OpenPage Function, and the location to be opened can be designated.

This function is not operated at local, and this supports to designate the location and size of a new page.

The function which handles events about the object at HTML of web page should be realized.

OpenPageObject Function

(1) Description

Open the graphic page and mark the designated object.

(2) Syntax

OpenPageObject(BSTR PageName, BSTR TagParam, BSTR ObjectName)

PageName: Designate a name of the page to be opened.

TagParam: Designate tags to be replaced on the template screen. If there is no tag to be replaced, designate empty string

(" ").

ObjectName: The name of the object to be marked.

PlaySound Function

(1) Description

Create the sound.

(2) Syntax

PlaySound(BSTR WaveFile)

WaveFile: This is the name of a sound file. It should be designated with the path.

Quit Function

(1) Description

Quit the graphic runtime program

(2) Syntax

Quit()

(3) Reference

The graphic runtime program is terminated.

QuitProject Function

(1) Description

Close the current project, and quit the InfoU monitoring system.

(2) Syntax

QuitProject()

(3) Reference

The current InfoU project is terminated and all of the processes are ended.

ReLoad Function

(1) Description

Load the current page again.

(2) Syntax

ReLoad(BSTR PageName, BSTR TagParam)

(3) Reference

The usage method of the function is same as OpenPage()Function. The difference is that the designated page is reopened in the same location with the same screen size instead of the currently loaded page.

If the PageName is not designated, the currently opened page is reopened.

ReportDatabase Function

(1) Description

Create a report.

(2) Syntax

ReportDatabase(BSTR ReportName)

ReportDatabaseEx Function

(1) Description

Create a report for the designated period.

(2) Syntax

ReportDatabaseEx(BSTR ReportName, BSTR RefTime)

ReportDialog Function

(1) Description

The report is created at the designated time.

(2) Syntax

ReportDialog()

(3) Reference

A screen is popped up to show the type of the report set up for the current project and the list of the created reports for searching and printing.

ReportPrint Function

(1) Description

Print a report.

(2) Syntax

ReportPrint(BSTR ReportName)

ReportPrintEx Function

(1) Description

Print a report.

(2) Syntax

ReportPrintEx(BSTR ReportName, BSTR RefTime)

ResetStatistic Function

(1) Description

Reset the tag's statistical value

(2) Syntax

ResetStatistic(BSTR TagName)

TagName: Reset a tag whose statistical data is reset.

(3) Reference

The statistical data is reset only for the tag designated as statistical tag.

RunScript Function

(1) Description

Execute a script that has been written in InfoUD.

(2) Syntax

RunScript(BSTR ScriptName)

ScriptName: The script name written in InfoUD.

SendData Function

(1) Description

Sent the factor value to the page name designated by the graphic runtime program

(2) Syntax

SendData(LPCTSTR PageName, VARIANT FAR& VariantData) OnReceiveData

(3) Reference

The factor value is delivered in a variant type to the Event Function OnReceiveData of the designated page name.

SetLogging Function

(1) Description

It should be logged on to the 'Event' for the change (control) in tag values and tag variable values in InfoU graphic runtime. However, when any specific job is implemented in the specific system, this function is available for controlling or modifying tag variable values without any need to log onto the 'Event.'

After implementing SetLogging(False), execute SetLogging(True) for the safe operation record of the system. If SetLogging(False) is implemented, the event logging for the control is not performed until SetLogging(True) is implemented.

(2) Syntax

SetLogging(BOOL TrueFalse)

SetTagVal Function

(1) Description

Set up (controls) a tag value or a tag variable value.

(2) Syntax

SetTagVal(BSTR TagVarName, VARIANT Value, VARIANT_BOOL bRet)

Parameter	Description
TagVarName	Tag name or tag variant value
Value	Control value(tag name intended to change)
bRet	Return control(True) or No-return control
	(False)

(3) Reference

Since this is a control function to change tag values, the description on the control is recorded upon the occurrence of any event.

For more details on the tag variable, see the 'tag variable' provision of the manual.

ShowAlarmPage Function

(1) Description

Open an Alarm page.

The Bin\GraSys\SysAlarmPage.ivd file of the director where InfoU is installed

(ex: :\InfoU\Bin\GraSys\SysAlarmPage.ivd) is opened as a default alarm page.

If the user wants to change the default alarm pgae, he/she may draw the "SysAlarmPage.ivd" page with the same name (saved in GraLS of the project) in the relevant project, then "SysAlarmPage.ivd" file of the project will be opened.

(2) Syntax

ShowAlarmPage(BSTR AlarmTag)

AlarmTag: The tag name is designated, then it finds the most recent alarm for the tag, and displays it on the alarm screen.

ShowCrossReference Function

(1) Description

Show the cross reference for the designated tag.

(2) Syntax

ShowCrossReference(BSTR TagName)

TagName: Designate a name of the tag.

ShowHelp Function

(1) Description

Open the help file.

This function displays the help file. The help file is made by "Chm" file type and the help file saved in the project directory or GraLS directory of the project will be shown. (The name of the help file is not inspected.)

If there is no help file in the project, then it shows the help file of InfoU.

(2) Syntax

ShowHelp(BSTR HelpFile, long idx)

HelpFile and idx parameter are not currently used. (Reserved parameters for expanding function)

ShowNetworkStatus Function

(1) Description

Open the screen to show the list and communication status of the device (station).

(2) Syntax

ShowNetworkStatus()

ShowRawDataInfo Function

(1) Description

Open the screen to display the information of raw data in communication.

(2) Syntax

ShowRawDataInfo

Sleep Function

(1) Description

Delay the implementation of the function for the designated time.

(2) Syntax

Sleep(LONG dwMilliseconds)

StopAlarmSound Function

(1) Description

If the sound of alarm occurs, this command stops the sound.

(2) Syntax

StopAlarmSound()

TimeOutMsgBox Function

(1) Description

Pop up the message box to be displayed for the designated time

(2) Syntax

TimeOutMsgBox(BSTR Msg, BSTR Title, long nSec)

UserCommand1 Function

(1) Description

This function is not operated at local, and it is reserved to be used for specific purpose by realizing the function that sends commands to the graphic page and handles events for the object at the HTML of the web page.

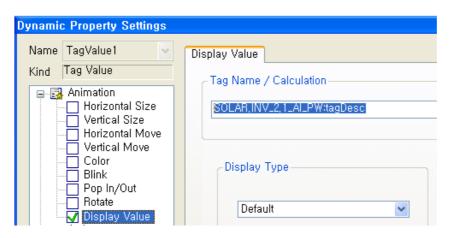
(2) Syntax

UserCommand1 (long nCmd, BSTR Str1, BSTR Str2, BSTR Str3, BSTR Str4)

9.19 **Tag Variable**

Tag variable is referred to variable or invariable related to tag defined by database. The tag variable can be referred by the same way of "Tag name: Tag variable name."

(1) Example used by the dynamic property settings for graphic.



(2) Example of using in Graphic Script

```
VBS Mini-Editor
 VBScript
                                                       Cancel
   Sub Rect1_OnClick()
   Dim This : Set This = Rect1
       Dim ValTag
       ValTag = GetTagVal("SOLAR.INV_2.1_AI_PW:tagDesc")
       MsgTrace ValTag
   End Sub
```

No	Name of Tag Variable	Description	Support	Remark
1	(tagValue)	It means the current value of tag. It is the same as	Read/Write	Analog/Digital/
		the tag's name is used only		String
		(If only the tag's name is inputted, the default value		
		appears)		
2	tagld	Tag unique index(long type)	Read	
3	tagDesc	Tag description (String)	Read	
4		Classification of system tags	Read	
	tagKind	0 : Normal tag		
		1 : System tag		
5		Tag type	Read	
	tagType	1:Analog, 2:Digital, 3:String, 4:Structure, 5:Block,		
		6:Grouop		
6		As an index of structure tag type, it indicates device	Read	
		type and it is used when showing information on the		
		device tag only for LSISdevice.		
	structTagTypeId	0: Used for normal tags. This information is not		
		necessary.		
		More than 1: This information shall be filled in in		
		case of structure tags.		
7	tagPriority	Tag's priority value(Control ranking:1~255)	ii	
8		It shows a type of industrial data value.	Read	
		1 : bool		
		2 : char		
		3:short		
		4: long		
		5 : longlong		
		6 : unsigned char		
	engValueType	7 : unsigned short		
		8 : unsigned long		
		9 : ulonglong		
		10:float		
		11:double		
		101:string type		
		201:raw type		
		More than 1000:Used for structure tag type.		

No	Name of Tag Variable	Description	Support	Remark
9		It shows device value type.	Read	I/O
		1 : bool		Analog/Digital/
		2:int8		String
		3:int16		
		4 : int32		
		5 : int64		
		6 : uint8		
		7 : uint16		
		8 : uint32		
		9 : uint64		
		10: float		
		11: double		
		12: bcd_8		
	devValueType	13: bcd_16		
		14: bcd_32		
		15: bcd_64		
		16: ubcd_8		
		17: ubcd_16		
		18: ubcd_32		
		19: ubcd_64		
		20: bit 1		
		21: bit_2		
		22: bit_3		
		23: bit_4		
		101:string		
		201: raw		
10		String index showing the unit of Analog Tag	Read	Analog
10	engUnitId	Curing index crieving the drift or / thalleg rag	rtodd	, a lalog
11	onStrld	String index displayed when digital tag is 1	Read	Digital
12	offStrld	String index displayed when digital tag is 0	Read	_
13		Delimiter for tag members	Read	Analog/Digital/
	containerId	Parent tag's index when it is a member of structure		String
		tag		
14		Memory tag calculation flag	Read	Memory Tag
	memTagExprFlag	0 : Memory Tag is not calculated	rtodd	Analog/Digital
	тюттад Ехрт ад	1 : Memory Tag is rist calculated		7 ti lalog/Digital
15		Expression of memory tag	Read	Memory Tag
13	memTagExpr	Expression of memory tag	rtcau	Analog/Digital
16		Expression cycle of memory tag	Read	Memory Tag
10	memTagExprCycle	Expression cycle of memory tag	1 Cau	
17		Tagle initial value	Pood	Analog/Digital
17	initVal	Tag's initial value	Read	Analog/Digital/ String
10		Mothod to conver my data to industrial data	Dood M/ rito	-
18	convertRule	Method to conver raw data to industrial data	Read/Write	I/ Analog
10	man a Maine	0:None, 1:Ratio/Bios, 2:Scale	Deed MA	Tog
19	rawMin	Minimum value of tag's raw data	ReadWrite	Tag

No	Name of Tag Variable	Description	Support	Remark
20	rawMax			I/O Analog Tag
21	dataMin	Minimum value of tag's industrial data	Read/Write	Analog Tag
22	dataMax	Maximum value of tag's industrial data	Read/Write	=
23	dataEngRatio	Ratio value to change tag values	Read/Write	I/O Analog Tag
24	dataEngBias	Bias value to change tag values	Read/Write	
25	valueDeadBand	DeadBand value (%)	Read/Write	=
26	alarmDelayTime	Alarm delay time(millisecond)	Read	Analog/Digital
27	alarmDeadBand	Alarm Deadband (%)	Read/Write	Analog Tag
28	alarmKind	Alarm process kinds. It is shown in each bit. [Analog] 0000 0000 0000 0001 : Boundary value(HH) 0000 0000 0000 0100 : Boundary value(HI) 0000 0000 0000 0100 : Boundary value(LO) 0000 0000 0000 1000 : Boundary value(LL) 0000 0000 0010 0000 : Changed amount 0000 0000 0100 0000 : Sub distance [Digital] 0000 0010 0000 0000 : OFF 0000 0100 0000 0000 : ON->OFF 0000 1000 0000 0000 : OFF->ON 0001 0000 0000 0000 : When changed	Read/Write	Analog/Digital
29	alarmLevel	Value to show alarm level 0 : None(Default Value). 1~10: Actual alarm level	Read	
30	alarmGenStrld	String index displayed when alarm is generated	Read	
31	alarmRcvStrld	String index displayed when alarm is recovered	Read	
32	alarmHH	Analog Tag HIHI Alarm boundary value	Read/Write	Analog
33	alarmHI	Analog Tag HI Alarm boundary value	Read/Write	
34	alarmLO	Analog Tag LO Alarm boundary value	Read/Write	
35	alarmLL	Analog Tag LOLO Alarm boundary value	Read/Write	_
36	alarmChangeValue	Absolute value of the alarm's changed value	Read/Write	
37	alarmGapRule	Rule to process distance alarm 0: Not used, 1: Absolute value used, 2:Relative value used(designating tag)	Read/Write	
38	alarmGapBaseValue	Base value of alarm distance(Absolute value)	Read/Write	
39	alarmGapBaseTagld	Tag index as base value of alarm distance	Read/Write]
40	alarmGapMain	Main fluctuation value of alarm distance	Read/Write	
41	alarmGapSub	Sub fluctuation value of alarm distance	Read/Write	
42	usrDefAlarmStr1Id	String 1 index of alarm defined by the user	Read	Analog/Digital
43	usrDefAlarmStr2ld	String 2 index of alarm defined by the user	Read	
44	usrDefAlarmStr3ld	String 3 index of alarm defined by the user	Read	
45	usrDefAlarmStr4ld	String 4 index of alarm defined by the user	Read	

No	Name of Tag Variable	Description	Support	Remark
46	stationId	Station index connected to tag 1:Memory Tag, More than 2:Station index	Read	I/O Tag
47	comAddr	Tag I/O address	Read	-
48	stationName	Station name connected to tag	Read	
49	devld	Device index	Read	
50	devName	Device name	Read	
51	driverName	Driver name	Read	-
52		String showing the unit of tag		Analog
	engUnitStr		Read	Analog
53	onStr	String index displayed when digital tag is On(1)	Read	Digital
54	offStr	String index displayed when digital tag is Off(0)	Read	A 1 /D: 1/ 1
55	alarmGenStr	String index displayed when alarm is generated	Read	Analog/Digital
56	alarmRcvStr	String index displayed when alarm is recovered	Read	
57	alarmGapBaseTagName	Base tag name of distance alarm	Read	Analog
58	pOnOffStr	String index showing On when digital tag is On while showing Off when digital tag is OFF (String)	Read	Digital
59	pAlarmEn	Whether to process tag alarm	Read/Writ e	Analog/Digital
60	pStatEn	Whether to process tag statistics value	Read/Writ e	
61	pAlarmEnHH	Whether to process HH alarm	Read/Writ e	Analog
62	pAlarmEnHI	Whether to process HI alarm	Read/Writ e	
63	pAlarmEnLO	Whether to process LO alarm	Read/Writ e	
64	pAlarmEnLL	Whether to process LL alarm	Read/Writ e	
65	pAlarmEnCHG	Whether to process changed alarm	Read/Writ e	
66	pAlarmEnMAIN	Whether to process main distance alarm	Read/Writ e	
67	pAlarmEnSUB	Whether to process sub distance alarm	Read/Writ e	
68	pAlarmEnDIG	Process kind of digital alarm 1:Off, 2:On,3:On->Off,4:Off->On,5:When changed	Read/Writ e	Digital
69	tagStatus	Information on tag status (Bit)	Read	Analog/Digital/Str
70	pTagStatus	Information on tag status (0:Normal, 1:Alarm, 2:bad)	Read	ing

Chapter 10 Alarm Function

This chapter presents description on each type of alarms and explains how to set up alarms (Runtime and gap alarms).

10.1 Analog Alarm

Alarm settings are available only for analog and digital tags in the InfoU and alarms are processed only when the relevant tag value is normal.

Alarms for analog tags can be divided into the border alarm, change alarm and gap alarm and then, the border alarm can be classified into HH, HI, LO and LL alarm and the gap alarm can be classified into main gap alarm and sub gap alarm.

Those three types of alarms for analog tags can be set up simultaneously.

Alarms for digital tags occur depending on the tag values and they have five types such as On->On, Off, On->On->Off, Off-On->On and alarm when changed.

Each alarm is prompted or cancelled according to the following rules:

Tag Type	Alarm Category	Alarm Name	Occur	Cancel
Analog	Border Alarm	HH Alarm	The alarm occurs when the tag value is more than HH	The alarm is cancelled if HH alarm has previously occurred and the tag value is less than HH.
		HI Alarm	It occurs when the tag value is more than HI but less than HH.	It is cancelled if HI alarm has previously occurred and the tag value is less than HI or more than HH.
		LO Alarm	It occurs when the tag value is more than LL but less than LO.	It is cancelled if LO alarm has previously occurred and the tag value is less than LL or more than LO
		LL Alarm	It occurs when the tag value is less than LL	It is cancelled if LL alarm has previously occurred and the tag value exceeds LL.
	Change Alarm	Change Alarm	It occurs when the tag value is significantly changed from the past value by more than the baseline value.	It is cancelled if the Change Alarm has previously occurred and the tag value is changed from the past value by less than the baseline value.
	Gap Alarm	Main Gap Alarm	It occurs when the tag value is different from the base value by more than the Main Gap Value.	It is cancelled if the Main gap Alarm has previously occurred and the tag value is different from the base value by less than Main Gap Value
		Sub Gap Alarm	It occurs when the tag value is different from the base value by more than the Sub Gap Value but less the than Main Gap Value.	It is cancelled if the Sub gap Alarm has previously occurred and the tag value is different from the base value by more than the Main Gap Value or less than the Sub Gap Value.

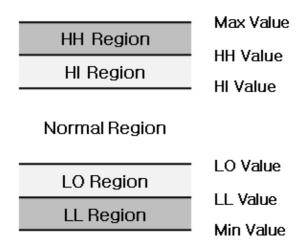
Chapter 10 Alarm Function

Digital	Digital	On Alarm	It occurs when the tag value	It is cancelled if the On Alarm has
	Alarm		is "On."	previously occurred and the tag value
				is Off.
		Off Alarm	It occurs when the tag value	It is cancelled if the Off Alarm has
			is "Off."	previously occurred and the tag value
				is On->On.
		On->Off Alarm	It occurs when the tag value	Occurrence and cancellation takes
			is changed from On to Off.	place at the same time.
		Off->On Alarm	It occurs when the tag value	Occurrence and cancellation takes
			is changed from Off to On	place at the same time
		Alarm when	It occurs when the tag value	Occurrence and cancellation takes
		changed	is changed	place at the same time

(1) Border Alarm (Analog)

The border alarm is used most among other alarms for analog tags.

The user may set up four border values such as HH, HI, LO and LL for the border alarm when tag engineering. If the user divides the region corresponding to the tag values horizontally and locates the baseline values to relevant regions, the relevant alarm occurs when the tag value is in the corresponding region and it is cancelled when the tag value is out of the region.

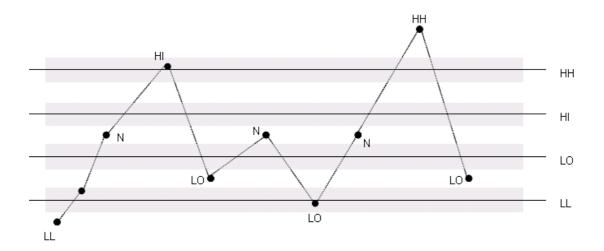


Each border alarm occurs in each corresponding region under the following condition:

- * HH Alarm: The tag value is in HH region.
- * HI Alarm: The tag value is in HI region
- * LO Alarm: The tag value is n LO region
- * LL Alarm: The tag value is in LL region

In addition, if any alarm Deadband value is set up, the Deadband rule applies according to the change trend of the tag value.

The following figure shows a simple example of this:



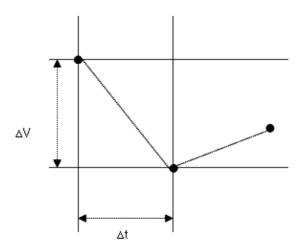
(2) Change Alarm (Analog)

The change alarm occurs when the current tag value is significantly changed from the past tag value by more than the allowed change value designated to each tag property item.

Similarly, if the change alarm occurs and then, the tag value is changed afterward but it does not reach the designated value, the alarm is cancelled. This alarm is set to ensure the tag value is not change significantly.

Ex) For detecting whether the power consumption increases significantly over the past power consumption.

The alarm Deadband rule does not apply to the change alarm.



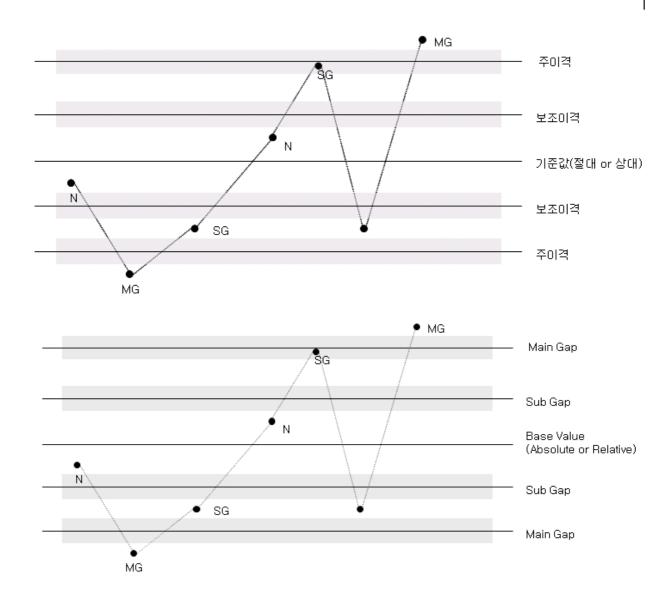
(3) Gap Alarm (Analog)

The gap alarm occurs when the delta rule of the Main and Sub Gap applies based on the baseline value. Similarly to the border alarm, the alarm Deadband rule applies to the gap alarm.

The gap alarm is similar to the border alarm but several differences are as follows:

- There is no upper or lower border. It is only divided into the Main and Sub Gap.
- The base value can be set with an absolute value or a relative value (tag).

Chapter 10 Alarm Function

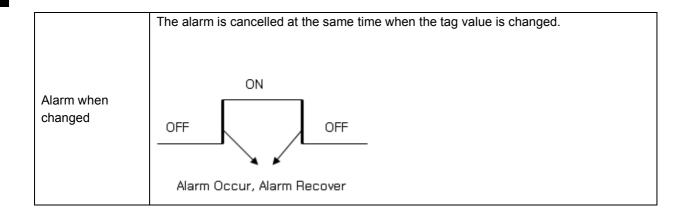


10.2 Digital Alarm

Digital alarms occur only for digital tags and they occur according to the previous value and the current value, consisting of five types such as On->On Alarm, Off Alarm, On->On->Off Alarm, Off->On->On Alarm, Alarm when changed.

Alamy Type	Description
On->On Alarm	The alarm occurs when the tag value is On->On. The alarm is cancelled when the tag value is Off. Alarm Occur OFF OFF
Off Alarm	The alarm occurs when tag value is "Off". The alarm is cancelled when the tag value is On. ON OFF OFF OFF OFF
On->Off Alarm	The alarm is cancelled at the same time when the tag value is changed from On to Off. ON Alarm Occur, Alarm Recover
Off->On Alarm	The alarm is cancelled at the same time when the tag value is changed from Off to On. OFF OFF Alarm Occur Alarm Recover

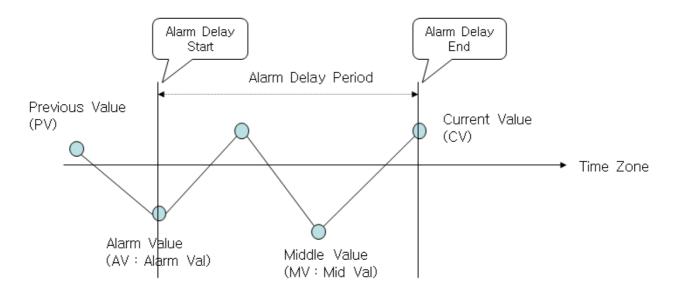
Chapter 10 Alarm Function



10.3 Alarm Delay

Alarm delay is a function to delay the occurrence of an alarm when some abnormal value is produced in the site by assigning a certain delay time to each tag and comparing the value after the specified time and the one before the initial occurrence of the alarm to prompt an alarm in stead of prompting an alarm immediately when some abnormal value is produced. The alarm delay diagram is as follows:

Alarm Delay Diagram



Previous Value: The last value before Alarm Delay logic occurred,

Alarm Value: The value when alarm occurred and enter alarm delay period

Middle Value: The value inside alarm delay period

Current Value: The current value when alarm delay time ended

This delay function applied according to the occurrence of alarm values is described as follows:

- When any alarm occurs, only the current value is renewed and that value is separately retained for delay process. This tag value is displayed as a normal one.
- The alarm delay function becomes activated when a new alarm occurs while the past value is normal, when sequential alarms occur or when the current alarm type is different from the past alarm type.
- After the alarm delay time, the current value (CV) and the previous value (PV) are compared each other in order to prompt or recover an alarm. At this time, the occurrence time is the time of the
- The same routine applies even when InfoU starts initially.
- When an alarm occurs, the occurrence time becomes the time of CV and the previous value becomes the value of PV. Mid values are all negligible (But, they are displayed as normal when the current value is inquired).

Chapter 11 Alarm Management

This chapter explans how the alarm acts from the time of its occurrence to the time of its cancellation according to the alarm group for which alarm actions have been set up.

11.1 Alarm Action during the Runtime

11.1.1 Alarm Occurnece

This explains actions upon the occurrence of an alarm. If any alarm occurs in the tag for which alarm settings are made, the alarm acts according to the action settings designated to its alarm group after a series of inquiry processes to find which alarm group the relevant tag belongs to. If the relevant tag belongs to multiple alarm groups, several actions can be overallaped.

Alarm Sound (1)

If alarm sound has been set up for a certain alarm group to which a tag belongs, a relevant wave file is outputted upon the occurrence of any alarm in the tag. This wave file repeats one through five times or infinitely according to the designated count.

If a higher level alarm occurs while a lower-level wave file is outputted, the existing output is suspended and the new alarm sound is generated. In case that several alarms take place simutaniously, sound is generated for the latest alarm among the highest level alarms. Alarm sound can be stopped by using 'stop sound 'function in the alarm view or scripter. At this time, this suspension is not permanent. It is applied only to the current sound. That is, if other alarm occurs, sound will be generated again. To stop sound permanently, the user may reset sound process in the 'Alarm Group Settings."

Auto Print (2)

If 'Auto Print' has been set up for a certain alarm group to which a tag belongs, information on the occurrence or cancellation of an alarm is automatically printed upon its occurrence or cancellation respectively..

(3) Send E-mail

If mail settings have been set up to make it possible to connect to the SMTP mail server, e-mail is sent to the users registerd in 'Receiving User' upon the occurrence of any alarm. At this time, the mail title and sending address can be prescribed but, contents need to be filled in by the sender according to the alarm's properties or status.

(4) Send SMS

If SMS setting have been selected and the SMS sending terminal is equipped with, SMS is sent to the users registerd in 'Receiving User' upon the occurrence of any alarm.SMS contents need to be filled in by sender according to the alarm's properties or status. Currently SMS is supported by LG Telecom.

Chapter 11 Alarm Management

11.1.2 Acknowledgement

Ack is to leave a record that the user has been acknowledged about the occurrence of alarm. The user can be acknowleged about it by the internal function on the alarm view or the scripter. If 'not acknowleged' (no check mark when acknowledged the time to delete the alarm) has been set up in the 'Alarm Settings,' any alarm can be deleted from the real-time alarm list whenever it is cancelled. However, if there are alarms that must be acknowleged, deleting is possible from the real-time alarm list only when the alarm is acknowledged regardless of the 'Alarm Settings' if 'ACK Process' is checked in the 'Alarm Group Settings.'

11.1.3 **Cancel Alarm**

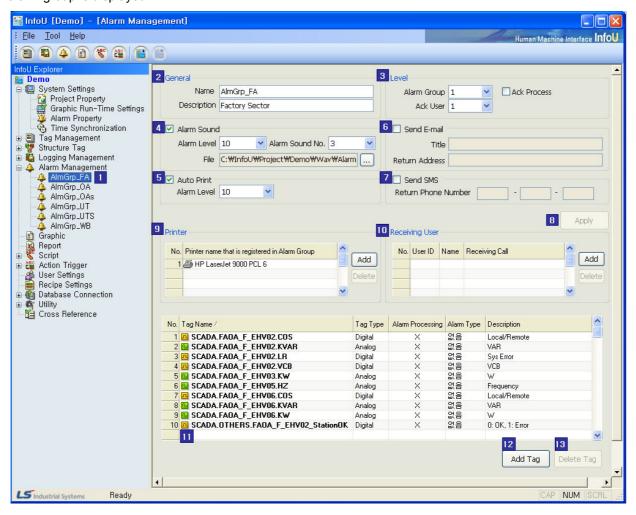
'Cancel Alarm' recovers the tag value to its normal status and actions upon the cancellation of alarm and it is applied only to 'Auto Print.'

11.2 **Alarm Group Settings**

If any alarm occurs, alarm sounding, printing, sending SMS and sending e-mail are carried out as actions against the relevant alarm. Those actions are set up for each alarm group. Therefore, several tags are registered in one alarm group and if any alarm occurs in one of the registerd tags, the designated actions to the group are performed.

11.2.1 Demo

Select 'Alarm Management' from InfoU Exploerer. If there are alarm groups that have been already registered, the alarm group list appears under the 'Alarm Management.' If one alarm group is selected, information on the alarm group is displayed.



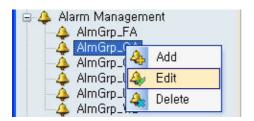
Chapter 11 Alarm Management

(1) Alarm Group List

The currently registered alarm group list is displayed.

Click the right key of the mouse, Add, Edit and Delete menus are displayed.

Edit and Delete are displayed only when any alarm group is selected.



(a) Add

- 1) Click the right key of the mouse and check 'Add' menu.
- 2) An empty box in the Alarm Group Settings' is displayed.
- 3) If a new name is designated to the existing alarm group without any modification, a new alarm group is registered.

(b) Edit: Edit alarm group

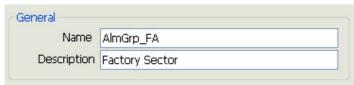
- 1) Click the right key of the mouse after selecting an alarm group and check 'Edit.'
- 2) Selecting an alarm group itself plays the same role as selecting 'Edit' does.
- 3) Existing settings are displayed in the 'Alarm Group Settings"
- 4) After modifying settings, finish editing with saving.

(c) Delete: Delete alarm group

- 1) Select an alarm group to delete.
- 2) Click the right key of the mouse and check 'Delete' menu.
- 3) The selected alarm group is deleted.

(2) General

Input a name and description on the alarm group.



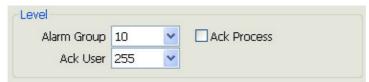
Input a name of the alarm group. If changing the existing name, the group is automatically acknowledged additionally.

(b) Description

Input description on the alarm group.

(3) Level

Set up a level and whether acknowledged or not.



(a) Alarm Group

If 'Alarm Group' is not specified, the entire alarm groups are selected.

(b) Ack User

Select a level of the acknowledged users.

(c) Ack Process

If 'Ack Process' is selected, any alarm can not be deleted from the alarm list without the user's acknowledgement. Even though it has been set up in the 'Alarm Settings' that any alarm can be deleted without acknowledgement, 'Ack Process' check should have priority over those settings.

(4) Alarm Sound

In case of the occurrence of any alarm, output of a wave file is set up for a particular group.



(a) Alarm Level

Alarm sound is generated only when the alarm level is higher than the designated alarm level.

(b) Alarm Sound No.

Select the number of alarm sound. One through five, infinitely or no sound can be seletive.

(c) File

Select an alarm sound to generate. Select a wave file.

(5) Auto Print

Select whether to use Auto Print in case of the occurrence of any alarm in the current group and set up an alarm level. Alarm information is printed by the printer that has been registered in the printer list below.



(a) Alarm Level

Auto Print is processed only when the alarm level is higher than the designated alarm level.

Chapter 11 Alarm Management

(6) Send E-mail

Set up whether to send alarm information to the selected recipients.



(a) Title

Input a title of the mail when sending e-mail on alarm information.

(b) Return Address

Input e-mail address of the sender when sending e-mail on alarm information.

For the blank intended to write the name of the recipient, input his/her e-mail address.

(7) Send SMS

Send alarm information to the recipient through SMS.



(a) Return Phone Number

Input the phone number of the sender who will be displayed when sending SMS. For the blank intended to write the name of the recipient, input his/her mobile phone number

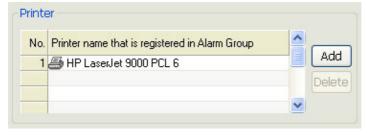
(8) Apply

If basic settings are saved, it is available to add tags, printers or Ack users if necessary.

(9) Printer

Indicate a printer to be used for alarm output.

Be cautious that only line printers are supported in this system.

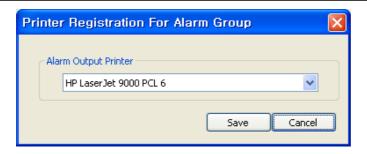


(a) Printer List

This list shows printers designated for Auto Print.

(b) Add

If 'Add' button is selected, the following dialog appears to allow the user to select a printer, and if 'Save' button is clicked after a printer is selected, the selected printer is added to the above list.



(c) Cancel

Select a print in the list of the printers registered for the alarm group and click 'Cancel' button.

The selected print is deleted from the list.

(d) Printer Registration for Alarm Group

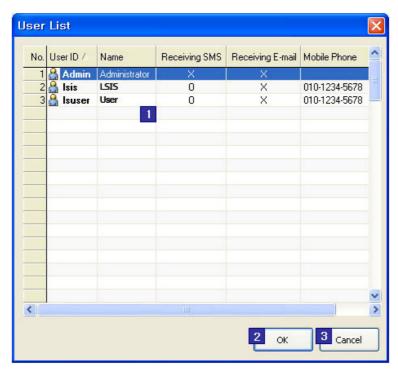
The list of the 'Auto Print' printers registered for the alarm group is displayed

(10) Receiving User

Send e-mail or SMS to the registered users.



- (a) Add: Click 'Add' button to display the user list on a separate window.
- (b) User List

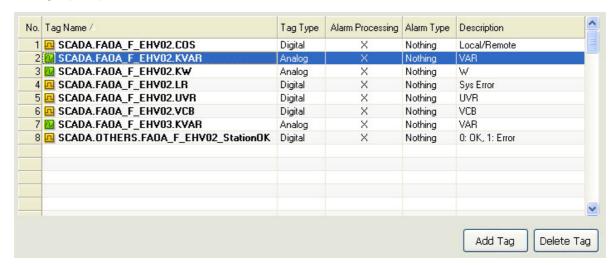


Chapter 11 Alarm Management

- 1) User List: The registered user list is displayed.
- 2) OK: Select a person who will receive the mail and click 'Ok' button.
- 3) Cancel: Close this user list window.
- (c) Delete: Delete a user from the user list.

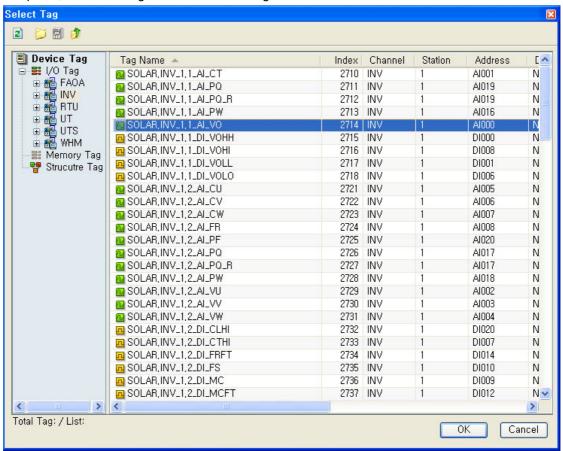
(11) Link Tag Settings

Link Tag Settings are to register a tag to an alarm group. For this, select a tag from the additional tag browser and register it. If any alarm occurs in the selected tag here, the actions that have been designated to the corresponding alarm group are processed.



(12)Add Tag

If the tag browser is displayed, select a tag to add to the list. It is also available to select multiple tags. Once selecting is completed, the selected tag is added to the Link Tag list.



(13) Delete Tag

Delete a registered tag from the group. It is also available to select and delete multiple tags together.

Chapter 12 Logging Management

Important data needed for logging management can be managed under the data logging group created in accordance with the user's purpose.

It is possible to assign one tag data to multiple logging groups for diverse data analyses.

- (1) Inquiry of Logging Data
 - Data can be inquired from Trend, Report or List Trend, and API, which can be inquired from script, is provided
- (2) Automatically calculated statistics/integrated data Statistics/integrated data are created without the use of additional scripts or applications.
- (3) Backup

Backup is available for all of the logging groups and backup files can be inquired from Trend and List Trend.

12.1 **Logging Type**

The logging group editor sets up a logging group that the history server will collect during the Runtime and various logging data are acquired and saved based on the logging group.

Logging groups can be divided into Cycle, Statistics, Accumulation, String, Event and Selective Log groups depending on their type and they are acquired and saved in either periodic or event way.

- (1) Cycle: Tag values are acquired in a certain cycle.
- (2) Statics: Statistics data on the designated tag are created periodically.
- (3) Accumulation: Statistics data on the accumulation values of the tag in which values have been accumulated are created periodically
- (4) String: String tag values are acquired in a certain cycle.
- (5) Event: Tag values are acquired in an event way according to event conditions.
- (6) Selective Log: Tag values are acquired periodically according to event conditions.

The following data can be inquired from Report or Trend according to the type of the logging group.

- (1) Cycle: Analog (Current Value), Digital (Current Status)
- (2) Statistics: Analog (Current Value, Min Value, Min Time, Max Value, Max Time, Average Value), Digital(On Number, Operation Time, Change Number)
- (3) Accumulation: Analog (Change Amount, Current Value)
- (4) Event: Analog (Current Value), Digital(Current Status)
- (5) Selective Log: Analog (Current Value), Digital (Current Status)

[HDD capacity calculation method for data collection]

Up to 40 tags can be registered in one logging group and 40 tag values can be saved at the same time with one saving.

Each logging group can save the following amount of data for one time:

·Cycle Logging: 1,048 byte

·Statistics and accumulation: 3,768 byte

·String Logging: 5,768 byte

Chapter 12 Logging Management

Needed disc capacity: Data to be saved per time x logging count

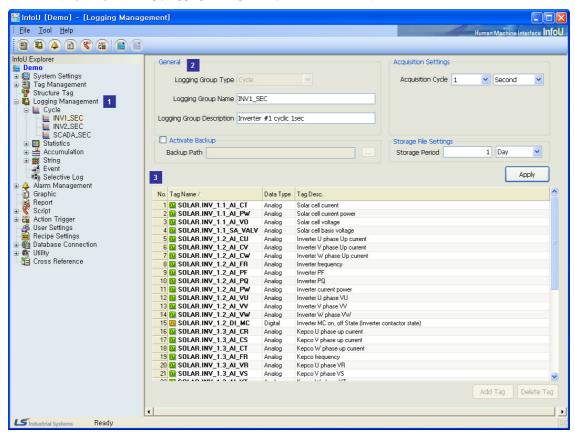
Ex) If it is planned to save data every second for one day under cycle logging, 1,048 X (60 X 60 X 24) = 90,547,200 byte (about 86.35 MB)

Ex) If it is planned to save statistics data every hour for one year, 3,768 X (24 X 365) = 33,007,680 byte (about 31.47 MB)

Ex) If it is planned to save string data every minute for one month, 5,768 X (60 X 24 X 31) = 257,483,520 byte (about 245.55 MB)

12.2 **Logging Settings**

If [Tool] - [Logging Management] is selected from the InfoUD menu, a screen as seen in [Figure] appears. Or, this screen can be displayed by selecting [Logging Management] from the InfoU explorer.



This screen consists of the following components:

(1) Logging Group Tree View

Logging group items are displayed in the InfoU explorer. The kinds of logging groups and the logging groups set up by the user are displayed in the tree. Logging groups can be newly added, edited or deleted by clicking on the right button of the mouse.

(2) Logging Group Property Settings

Properties of the set up logging group are displayed. Adding and editing logging groups are also available. Acquisition method items are displayed differently depending on the kinds of the logging groups and they can be saved by clicking on "Apply" button.

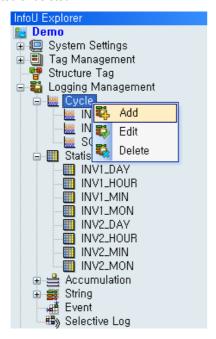
(3) Logging Tag List

Tags registered in the logging group are displayed in a list form.

The acquired tags can be registered or deleted with "Add Tag" or "Delete Tag,"

12.2.1 Add, Edit, Delete

By clicking the right button of the mouse on the "Logging Management" of the InfoU explorer, logging groups can be added, edited or deleted.



(1) Add

A new logging group is added. If "Add" is selected and the initialized right screen is filled in with new information, properties of the newly added logging group can be set up.

The logging group can be saved if "Apply" button is clicked after setting up is completed.

Detailed settings for each logging group type are explained below.

(2) Edit

Properties of the existing logging group are edited. The changed contents are saved if "Apply" button is clicked after property editing is completed.

Also, properties can be edited on the right view screen if the relevant item is selected with the mouse.

* A new logging group can be added by changing only the name of the existing logging group on the editing screen. At this time, the same property information as that of the previously displayed data will be saved.

(3) Delete

The selected logging group is deleted.

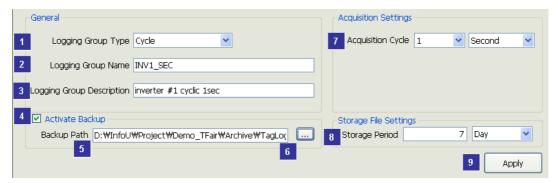
12.2.2 General

Logging group settings are carried out through the following two steps:

(1) Logging Group Settings

An acquisition method is set up along with logging group type, logging group name and logging group description, and storage and backup of logging groups are also set up. Up to 256 logging groups can be set up in a project. Especially, a sufficient space shall be secured in the hard disc when setting backup.

Acquisition methods are displayed on different setting screens depending on the type of logging groups.

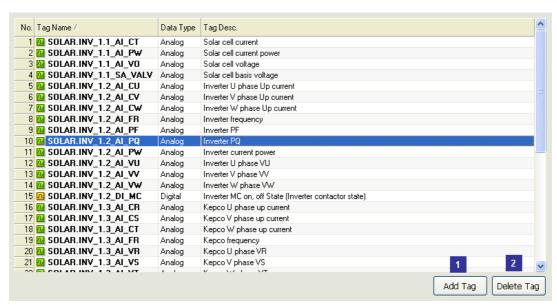


- (a) Logging Group Type: Select one among Cycle, Accumulation and String.
- (b) Logging Group Name: Input a name to identify the relevant logging group. Up to 31 letters are available to input.
- (c) Logging Group Description: Input description on the logging group. Up to 63 letters are available to input.
- (d) Activate Backup: Select this to backup the logging file. Activate backup and then, input a backup path.
- (e) Backup Path: Input a path in which the backup file will be saved. Select a path with the right button of the mouse.
- (f) Path Selection Button: Click the button to display the path selection dial log.
- (g) Acquisition Cycle: Select a cycle to acquire data. Acquisition cycle may set up differently depending on the type of the logging group.
- (h) Storage Period: Set up a storage period of the history file. For storage period, select one between number 1 and 100 and for unit, select one among time, day and month.
- (i) Apply: The inputted information on the logging group is saved.

(2) Tag Settings

Once adding logging groups is completed, select a tag that will acquire logging data. One logging group can save up to 40 tags.

Chapter 12 Logging Management

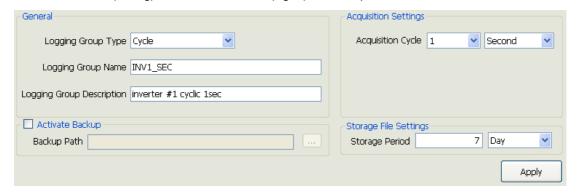


- (a) Add Tag: Select a tag to save from the tag selector.
- (b) Delete Tag: Delete the selected tag.

12.2.3 Cycle

The minimum acquisition cycle is one second so that this cycle is suitable for fast real-time data logging and it is mainly used for inquiring sequential values from Trend or Report.

The current value (analog) and the current status (digital) can be inquired.



During the runtime, cycle logging groups are acquired and saved on time.

Ex) If the acquisition cycle is set 2 seconds, logging data are saved every 0, 2, 4, , ..., seconds.

Acquisition Settings: Select a cycle and unit. The following cycles can be selected for each unit:

- Second: 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30
- Minute: 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30
- Hour: 1, 2, 3, 4, 6, 8, 12
- Day: 1

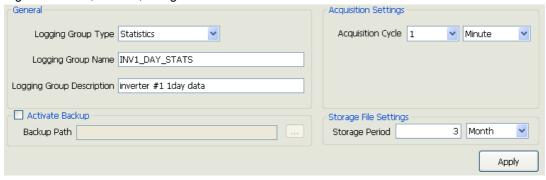
12.2.4 **Statistics**

The minimum acquisition cycle is one second so that this logging group type can automatically calculate and save various

It is used to identify data status for a certain period and mainly used for inquiring statistics status from Report or Trend. The following data can be inquired.

Analog: current value, min value, min time, max value, max time, average value

Digital: ON count, ON time, change count



During the runtime, statistics logging groups are acquired and saved on time.

Ex) If the acquisition cycle is set 2 seconds, logging data are saved every 0, 2, 4, ..., seconds.

Acquisition Settings: Select a cycle and unit. The following cycles can be selected for each unit:

- Minute:1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30
- Hour: 1, 2, 3, 4, 6, 8, 12
- Day:1 ~ 31
- Month: 1 ~ 12

12.2.5 **Accumulation**

Tags that have accumulation property such as power gauge and fuel indicator are registered and the change amount for a certain period is calculated and saved.

Change amount for every hour, day or month, that is, consumption amount can be inquired without using any additional scrip or memory tag.

The following data can be inquired.

Analog: change amount, current value

Digital: Not supported.



Chapter 12 Logging Management

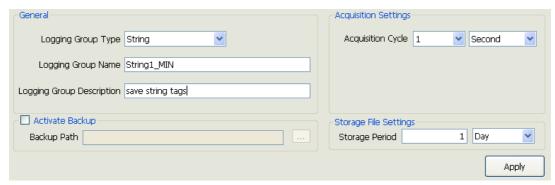
Accumulation statistics data for the accumulation logging group are automatically created every hour, day and month. The created accumulation logging groups are classified into HOUR, DAY and MONTH.

During the runtime, accumulation logging groups are acquired and saved on time.

Ex) Data are acquired and saved on the hour for hourly logging groups, at 00:00:00 every day for daily logging groups and at 00:00:00 on the first day of every month for monthly logging groups.

Mean of the company of the compan

12.2.6 String



During the runtime, string logging groups are acquired and saved on time.

Ex) If the acquisition cycle is set 2 seconds, logging data are saved every 0, 2, 4, , ..., seconds.

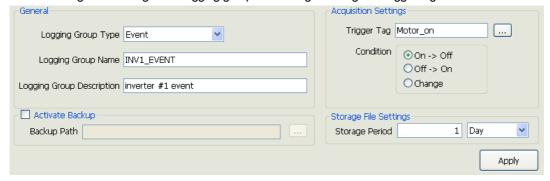
* Only string tags can be set up as string logging groups.

Acquisition Settings: Select a cycle and unit. The following cycles can be selected for each unit:

- Second: 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30
- Minute: 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30
- Hour: 1, 2, 3, 4, 6, 8, 12
- Day:1

12.2.7 **Event**

This logging group type is used for saving tag values according to event conditions. Here, changes in digital tag values play a role in constructing event conditions and these digital tags are called trigger tags. Event logging groups save the values of 40 tags that belong to the logging groups according to changes in trigger tags.



Event logging groups are set up as follows:

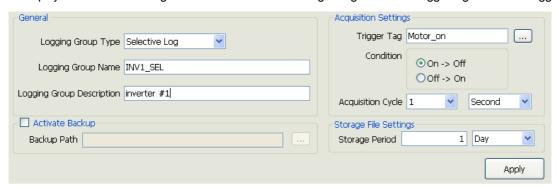
- Trigger Tag: Digital Tag
- Logging: Analog, Digital Tag
- Storage Method: Saved only one time when any event takes place

Event conditions available to be set up in trigger tags are as follows:

- When it is triggered off -> on
- When it is triggered on -> off
- When it is changed on <--> off

12.2.8 **Selective Log**

This logging group type is used for saving tag values periodically according to event conditions. Here, changes in digital tag values play a role in constructing event conditions and these digital tags are called trigger tags. Selective logging groups



Event logging groups are set up as follows:

- Trigger Tag: Digital Tag
- Logging: Analog, Digital Tag
- Storage Method: Saved periodically while the conditions of trigger tags remain when any event takes place.
- Storage Cycle: One second ~ one hour

Event conditions available to be set up in trigger tags are as follows:

- Saved periodically when it is triggered off -> on
- Saved periodically when it is triggered on -> off

Chapter 13 Script

The user can implement various functions he/she needs by writing programs freely.

Since this program is of the Visual Basic Script Type, the user can implement various functions by directly using the internal functions provided in a type of OLE in addition to general functions provided by the Visual Basic Script:

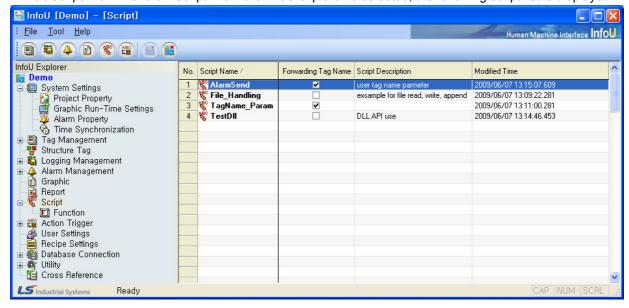
- (1) VB Script support
- (2) COM, OLE Object Import
- (3) VB Style Text Color and Dot Operation support
- (4) Script grammar error check

The registered script is implemented during the runtime according to the Trigger rule defined in the 'Action Trigger.'

13.1 Start

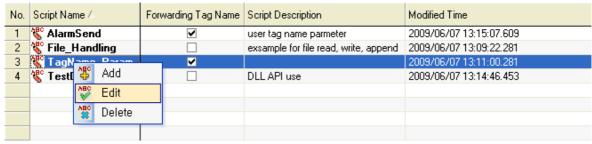


If "Edit Script" from menu or "Script" from the InfoU explorer is selected, the following script list is displayed.



Chapter 13 Script

Click the right button of the mouse to show Add, Edit and Delete menu.



- (1) Add: A new script is added. Double click on the empty item, the new input screen is displayed.
- (2) Edit: The selected script is edited. If the item is double clicked, the edit screen is displayed
- (3) Delete: The selected script is edited.

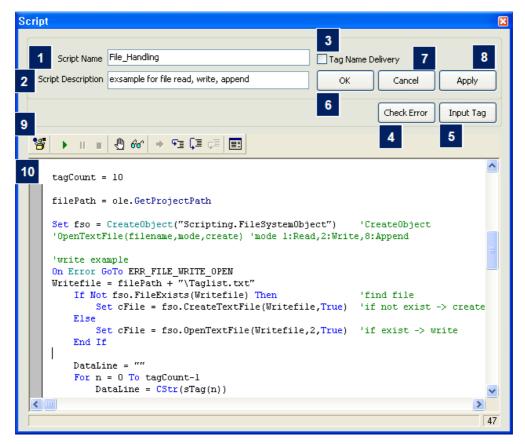
13.2 Editor Screen

The script editor screen is of Modaless dialog type and it is constructed as follow:

If other item is selected from the above script list while the screen is displayed, information on the selected script is displayed.

Except the following cases, script modifications along with other changes during the runtime are all reflected at the same time.

- (1) When a script is added as a new one
- (2) When the option 'Tag Name Deliver' is changed
- (3) When the 'Action Trigger' is redefined



(1) Script Name

- (a) The script name is displayed.
- (b) If the script is renamed and saved, a new script is created and added automatically.
- (c) Any two scripts shall not have the same name.
- (d) Any script name can not be changed when the function option 'Tag Name Deliver' is selected.

(2) Script Description

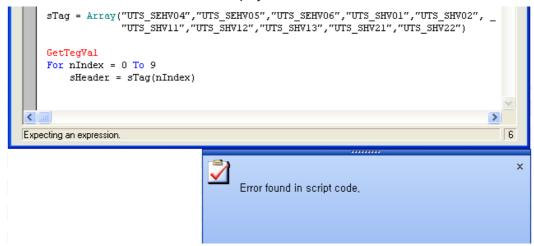
(a) Input description on the script.

(3) Tag Name Delivery

- (a) Tag names can be delivered in script functions during the runtime.
- (b) Be cautious that the contents that are still written is initialized and the mode is changed if this Tag Name Delivery is selected.
- (c) The tag name shall be the same with the one used when any "action upon the alarm status" or "action upon the change in tag values" takes place.
- (d) For details, refer to 'Hot to use script.'

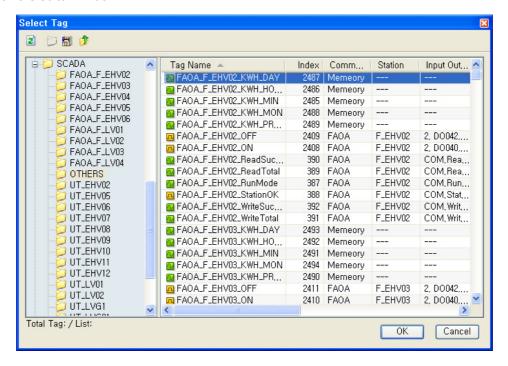
(4) Check Error

- (a) The script that is still written is checked for grammar errors.
- (b) Only grammar errors are detected and neither logical errors nor error data during the runtime can detected.
- (c) If any error is detected, it is displayed on the 'Script Error Message window' and the error type and the line on which the error is detected are displayed in the status bar on the bottom of the editor.



(5) Input Tag

The tag list of the current project is displayed and if a tag is selected, the relevant tag name is inputted on the editor window.



```
Sub Main
Dim TagVal001
ole.Sleep(1000)
TagVal001 = ole.GetTagVal("SCADA.OTHERS.FA0A F EHV02 KWH DAY")
End Sub
```

(6) Ok

The script is checked for grammar errors and if any error is not found, the script is saved and the screen is ended.

(7) Cancer

Any modification is cancelled and the screen is ended.

(8) Apply

· The script is checked for grammar errors and if any error is not found, what has been edited is saved.

(9) Editor Toolbar

The script is implemented on the editor screen and debugging work is also available. For details, refer to 'How to use.'

13.3 How to Use

This explains how to register and use the script.

- (1) Use of Internal Functions
 - (a) Declare iuInterFuncOleSvr objects as follows and assign them by using Set command.

```
Sub Main
   Dim iuOle As iuInterFuncOleSvr
   Set iu0le = New iuInterFuncOleSvr
```

(b) Use one of the public variants that have been already declared when a script is newly registered.

```
Public ole As New iuInterFuncOleSvr 'InfoU Ole Object
```

(c) If "." (Dot) is inputted in a declared object variant, available functions are displayed upon the Dot Operation.

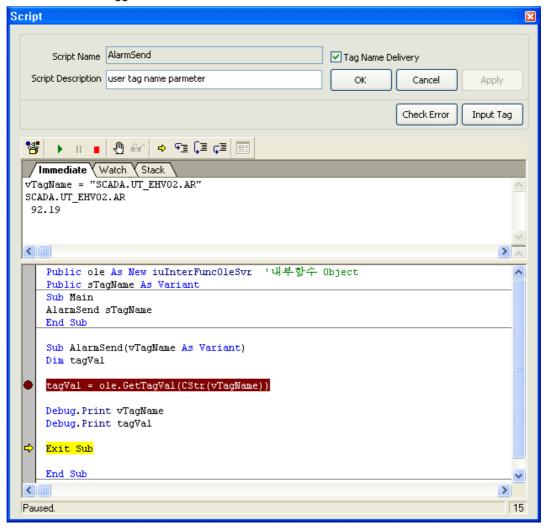
```
iu0le.SetTagVal "UTS_LVG31.WHRST",varVal
iu0le.SetTagVal "UTS_LVG32.WHRST",varVal
iuOle.Set
   🦽 OpenPage
iu( 🚕 OpenPageObject
                                 HRST", varVal
                                HRST",varVal
HRST",varVal
iu( 🚕 OpenProject
iu( 🌉 PlaySoundFile
                                HRST", varVal
iu( 🚕 Quit
                                HRST", varVal
iu( 🚕 ResetStaticValue
                                HRST", varVal
iu( 🚕 RunApp
iu( 🚕 RunScript
                                HRST", varVal
    <u>⊶</u> SendEmail
iu( 🚕 SendSMS
   <u>"</u>
SetTagVal
```

(d) Since even the function parameter as seen in the figure below is displayed, it ensures convenient inputting.

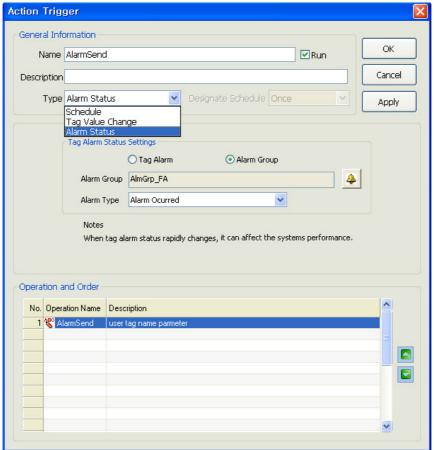
```
iuOle.SettagVal
       SetTagVal( ByVal szTagName As String , ByVal vtTagValAs Variant )
iuOle.SetTagVal "UTS SHVOl.WHRST", varVal
```

(2) Tag Name Delivery

A script function is registered as a 'Tag Name Delivery' script function and processes to trigger actions are explained on the 'Action Trigger' screen.

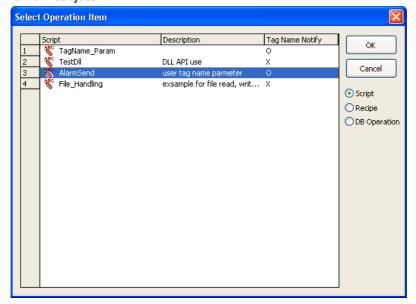


- (a) Select 'Tag Name Delivery' option on the 'Script' screen.
- (b) A sub-routine function (AlarmSend) is created with the same name with that of the script inputted among "Main" function scripts. At this time, Korean letters can not be used for script names.
- (c) sTagName is declared as a Public variant and the Main function is structured to deliver variants upon the call of subroutine function.
- (d) vTagName converted to the parameter of AlarmSend(vTagName As Variant) function in the subroutine function view can be used as a tag name.
- (e) Since the process to deliver the actual tag name can not be carried out on the editor screen while debugging is performed, debugging can be done in a method of inputting variant values manually with a formula by directly substituting each tag name on the execution screen.



(f) Then, 'Action Trigger' is set up to execute the selected script on the 'Action Trigger' screen.

- (g) Select a trigger type on the Action Trigger' screen. Available action trigger types for "Tag Name Deliver" are "Alarm Status" and "Tag Value Change."
- (h) Select and assign a script available for "Tag Name Deliver" as following when selecting a work subject.

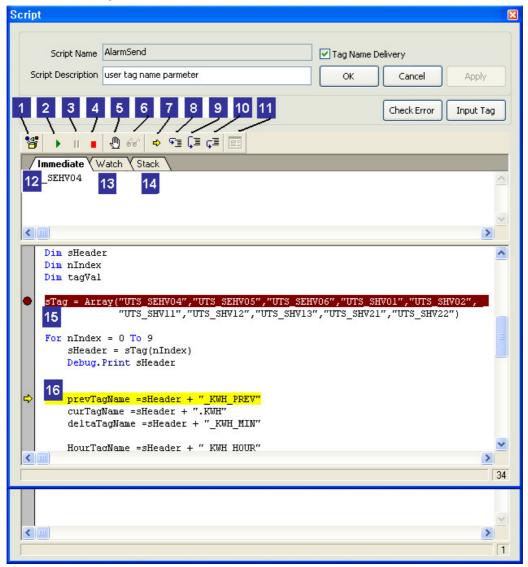


(3) Editor Debugging

Debugging methods are explained on the 'Script' screen.

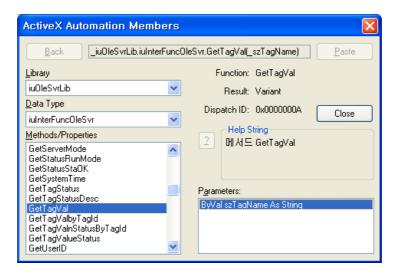
The inquiry of correct results is ensured only when debugging is worked while the Runtime is executed to have internal functions act correctly.

The following screen represents one of the script debug screen. Click on [2], [8], [9] and [10] on the 'Script' to show the followings:



[1] Browse Object

- 1) Methods and Parameters of the referred ActiveX object are displayed.
- 2) As default settings, information on the InfoU Function is displayed.



[2] Start/Resume:

The user may execute a script on the editor (Since the execution of a correct function is ensured only when it is executed during the runtime, a code shall start for debugging).

[3] Pause

The script that is being played is paused.

[4] End

The script that is being played is ended and the debug mode is terminated.

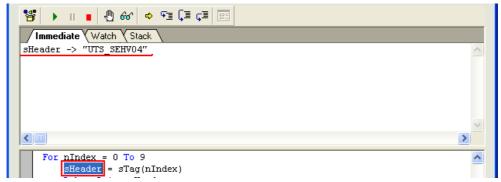
[5] Toggle Break

A break point shall be either selected or cancelled.

When setting a break point, it should be ensured that it stops at the break point to perform debugging.

[6] Evaluate Expression

If a variant selected and clicked on the editor window, values are displayed on [12] immediate window.



[7] Show Current Statement

The current statement is shown when executing.

[8] Step Into

Each line of the scrip is executed one by one. If any 'Sub Routine' exists in the relevant line, it

Chapter 13 Script

steps into the sub routine for debugging.

[9] Step Over

The script statement is executed step by step. If any 'Sub Routine' exists, it moves to the next step after the sub routine is executed and its result is returned.

If any break point exists in the sub routine, it stops at the break point.

[10] Step Over

It gets out of the current step and goes to upper procedure after completely executing all of the current functions and loop statements.

If any break point exists in the sub routine, it stops at the break point.

[11] Edit User Dialog

This dialog box editor can not be selective during the debugging and it is activated only in the 'Edit' mode.

InfoU does not support this dialog box in the runtime environment.

[12] Immediate

This 'Immediate' window is used as follows:

- Result value when clicking on [6] Evaluate Expression
- Show the return valued of debug.Print
- Change variant values when any 'Immediate' formula is inputted.

```
Immediate Watch Stack

SHeader -> "UTS_SEHV04"

UTS_SEHV04

prevTagName = 100

Debug.Print sHeader

'공통부분

prevTagName = sHeader + "_KWH_PREV"
```

[13] Watch

Variants are assigned and function or expression values can be inquired.

For assigning, select a variant and click the right side of the mouse to select Debug-->Add Watch.

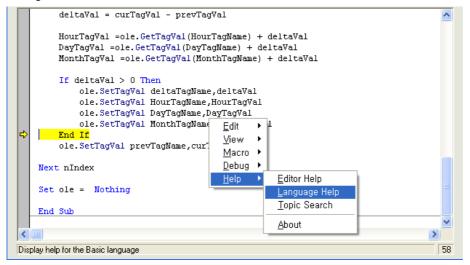
```
| Immediate | Watch | Stack |
| 1: prevTagName -> "UTS_SEHV04_KWH_PREV" |
| 1: curTagName -> "UTS_SEHV04_KWH_"
| 1: deltaTagName -> "UTS_SEHV04_KWH_MIN" |
| 1: HourTagName -> "UTS_SEHV04_KWH_HOUR" |
| 1: DayTagName -> "UTS_SEHV04_KWH_DAY" |
| 1: MonthTagName -> "UTS_SEHV04_KWH_MON" |
```

[14] Stack

The call list of the current execution line is shown.

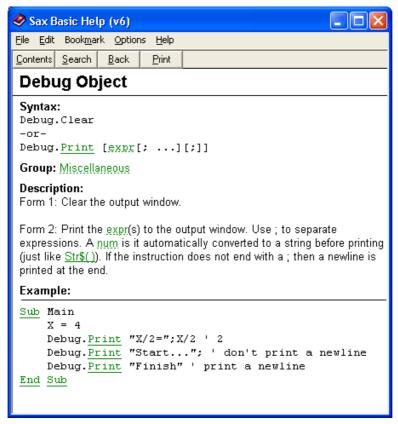
[15] Menu Function

If the right side of the mouse is clicked, a menu is shown and it has functions available in the debug mode as well as functions available in the Edit mode.



[16] Help

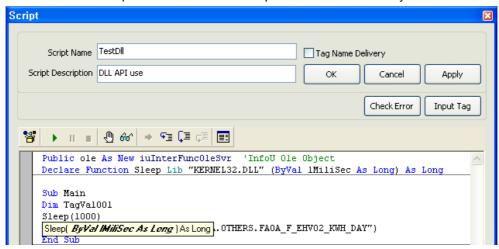
'Help' is available for overall information on various functions and script languages provided by the editor.



(4) How to use DLL

How to use external DLL functions in the script is explained.

DLL API use: It is an example of declaration of 'Sleep' function included in System KERNEL32.DLL.



(5) How to use OLE object

It is an example of file 'Open and Write'. (Refer to examples in Demo projects.)

```
Dim fso As Object
                        'Object declare
Dim cFile As Object
                        'Object declare
Set fso = CreateObject("Scripting.FileSystemObject")
                                                       'CreateObject
'OpenTextFile(filename, mode, create) 'mode 1:Read, 2:Write, 8:Append
'write example
On Error GoTo ERR_FILE_WRITE_OPEN
Writefile = filePath + "\Taglist.txt"
    If Not fso.FileExists(Writefile) Then
                                                         'find file
        Set cFile = fso.CreateTextFile(Writefile,True) 'if not exist -> create
        Set cFile = fso.OpenTextFile(Writefile,2,True) 'if exist -> write
    End If
    DataLine = ""
    For n = 0 To tagCount-1
        DataLine = CStr(sTag(n))
        DataLine = DataLine + vbrf
        cFile.WriteLine(DataLine)
    Next
    cFile.Close
Debug. Print "File Write OK!" + Writefile
```

(6) Error Handle

It is an example of how to handle errors (Refer to examples in Demo projects.)

Write On Error statement on the top of the error to be handled. If any error occurs below the On Error statement, it moves to the Error Handle Statement written next to Goto Statement.

Since On Error statement is effective until it meets with the next On Error statement, 'Error Handle' is possible according to work steps by writing the On Error statement at several locations.

By using the 'On Error Resume Next Statement', the next codes can be performed seamlessly even any minor error occurs.

```
'append example
On Error GoTo ERR_FILE_APPEND_OPEN 'error handler
Appendfile = filePath + "\" + Format(Now(), "YYMMDD") + "_timelist.txt"
    If Not fso.FileExists(Writefile) Then
                                                         'find file?
        Set cFile = fso.CreateTextFile(Appendfile,True) 'No:create
   Else
        Set cFile = fso.OpenTextFile(Appendfile,8,True) 'yes:write option = 8
   End If
   DataLine = Format(Now(), "YYYY/MM/DD hh:mm:ss") + vbTab
    For n = 0 To tagCount-1
        tagVal = ole.GetTagVal(CStr(tagName(n)))
        AppendLine = AppendLine + CStr(tagVal) + ","
   Next
   DataLine = DataLine + AppendLine + vbrf
    cFile.WriteLine(DataLine)
    cFile.Close
   Debug. Print "File Append OK!"
Exit Sub
ERR FILE READ OPEN:
    errFlag = -1
   Debug.Print "ERR_FILE_READ_OPEN"
   Exit Sub
ERR FILE WRITE OPEN:
    errFlag = -1
   Debug.Print "ERR FILE WRITE OPEN"
   Exit Sub
```

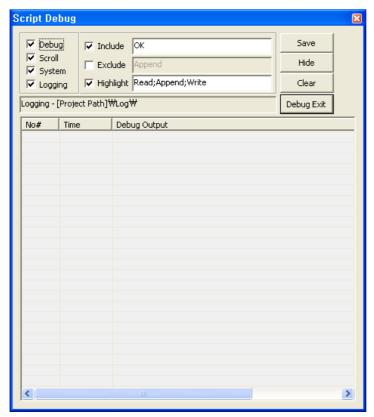
13.4 Runtime Debug

By providing the debug screen during the Runtime, this ensures the user inspects the current execution status of the script and check for any operational and logical error.

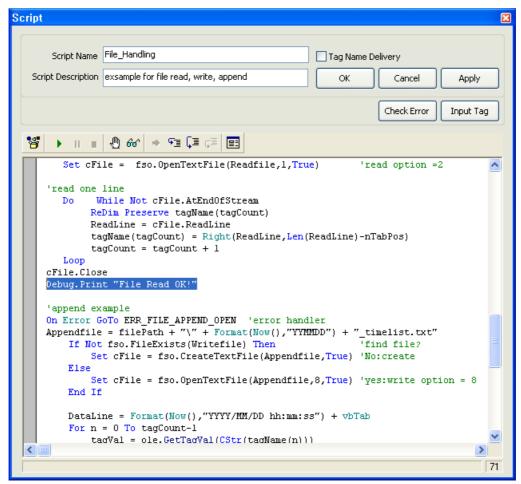
(1) Execution



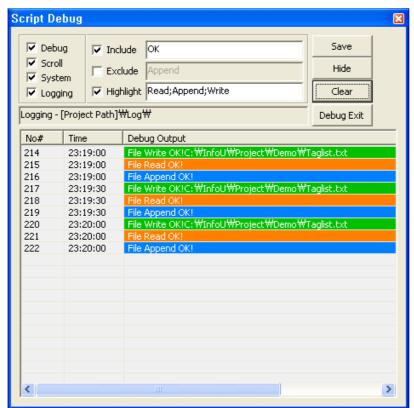
Click the right side of the mouse on the InfoU tray icon during the Runtime to select 'Scrip Debug' from the pop-up menu and show the screen below.



The output written in Debug.Print [expr] on the 'Script' screen is displayed as a result during the runtime.



(2) Runtime Debugging

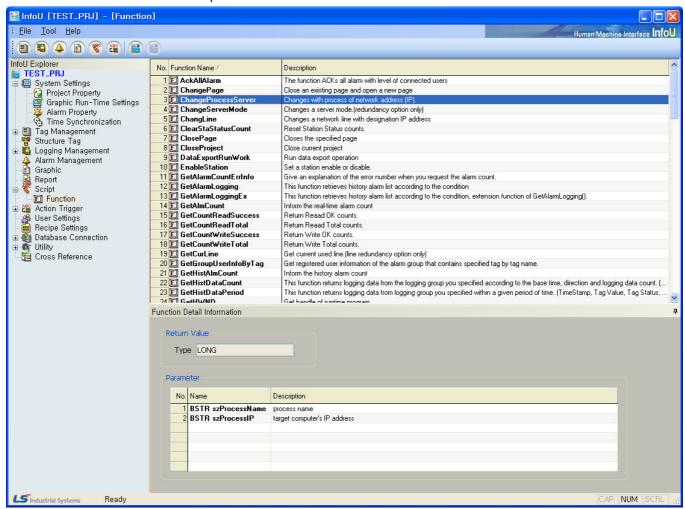


This screen is displayed as the highest window.

- Debug: Select whether to conduct debug works,
- Scroll: Scroll the screen to show the latest list.
- System: Select whether to execute script except Debug.Print construction.
- Logging: Save debugging contents in the \Log folder on the current project path.
- Include: The letters to be displayed are filtered and displayed.
- Exclude: The letters to be excluded are filtered and displayed.
- Highlight: The contents included as inputted letters are highlighted with colors.
- Include, Exclude and Highlight can be overlapped by using [;].
- Save: The current settings and the debug screen size are saved and the current conditions will be displayed on the screen in the next execution.
- Hide: The debug screen is hid.
- Clear: All of the lists currently displayed are deleted.
- Debug Exit: The runtime debug is ended and the screen is closed.

Chapter 14 Function

Various default functions are presented to perform various actions in the InfoU system. The following list shows functions that can be used for Script and OLE Automation.



14.1 Function Inquire

After logging in withInfoUD.exe, select 'Function' under 'Script' on the InfoU Explorer' To see detailed information on a certain function, select and click on the relevant function with mouse. Then, input 'Return Value' of the relevant function on the 'Function Detail Information' screen to display a list that shows the properties and description of each parameter requested in the relevant function in regular sequence.

14.2 Function Classification

14.2.1 List by Item

LONG CloseProject () BSTR GetProjectName () BSTR GetProjectPath () Function BSTR GetProjectType () LONG OpenProject(BSTR szProjectName, BSTR szProjectPath) LONG QuitProject () LONG ChangePage (BSTR szOldPage, BSTR szNewPage) LONG ClosePage (BSTR szPageName)
Project BSTR GetProjectPath () Function BSTR GetProjectType () LONG OpenProject(BSTR szProjectName, BSTR szProjectPath) LONG QuitProject () LONG ChangePage (BSTR szOldPage, BSTR szNewPage)
Function BSTR GetProjectType () LONG OpenProject(BSTR szProjectName, BSTR szProjectPath) LONG QuitProject () LONG ChangePage (BSTR szOldPage, BSTR szNewPage)
LONG OpenProject(BSTR szProjectName, BSTR szProjectPath) LONG QuitProject () LONG ChangePage (BSTR szOldPage, BSTR szNewPage)
LONG QuitProject () LONG ChangePage (BSTR szOldPage, BSTR szNewPage)
LONG ChangePage (BSTR szOldPage, BSTR szNewPage)
LONG ClosePage (BSTR szPageName)
VOID GetHWND(ULONG* hwnd)
Screen VOID HardCopy()
Function BOOL IsRunning ()
VOID OpenPage(BSTR szPageName, BSTR szPageParam)
LONG OpenPageObject (BSTR szPageName, BSTR szObjectName)
VOID Quit()
ULONG GetTagStatus (BSTR szTagName)
BSTR GetTagStatusDesc(ULONG ITagStatus)
VARIANT GetTagVal (BSTR szTagName)
VARIANT GetTagValbyTagId (LONG ITagId)
BSTR GetTagValnStatusByTagId (LONG ITagId, VARIANT* vtTagVal, LON
*IScanMode, LONG *IMainQ, LONG *ISubQCnt, LONG *ISubQ1, LONG *ISubQ2, LON
*ISubQ3, LONG *ISubQ4, LONG *ISubQ5)
Tag Function LONG GetTagValueStatus(BSTR szTagName, VARIANT* vtTagVal)
LONG ResetStaticValue (BSTR szTagName)
VOID ShowCrossReference (BSTR szTagName)
VOID ShowRawDataInfo()
VOID ShowSIMCfgInfo()
VOID SetTagVal (BSTR szTagName, VARIANT vtTagVal)
LONG SetTagValEx (SHORT nMode, BSTR szTagName, VARIANT vtTagVal, BST
szClientName, ULONG hClientHandle)
Scriopt (OCID BunSarint (PSTP arSarintNama)
Function VOID RunScript (BSTR szScriptName)
LONG NotifyAlarmEvent (DOUBLE IIOccur, USHORT nAlmLevel, BSTR szMessage
BSTR szArea, BSTR szOUser, BSTR zInformation, BSTR szUserDefine1, BST
szUserDefine2, BSTR szUserDefine3, BSTR szUserDefine4, BSTR szPrev, BSTR szCu
BSTR szObject)
Event Function VOID SendEmail (BSTR szMailAddTo, BSTR szMailAddFrom, BSTR szMailTitle , BST
szMailMsg)
VOID SendSMS (BSTR szRecvPhoneNo, BSTR szCallBackPhoneNo, BST
szSMSMsg)

Item	Relevant Function	
	VOID NotifyError (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	
	VOID NotifyHostError (BSTR szHostName, USHORT nLogCategory, BSTR szMsg,	
	BSTR szDesc)	
	VOID NotifyHostInfo (BSTR szHostName, USHORT nLogCategory, BSTR szMsg,	
	BSTR szDesc)	
	VOID NotifyHostLog (USHORT nLogType, USHORT nLogCategory, BSTR	
Log Function	szHostName, BSTR szMsg, BSTR szDesc)	
	VOID NotifyHostWarn (BSTR szHostName, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	
	VOID NotifyInfo USHORT nLogCategory (USHORT nLogCategory, BSTR szMsg,	
	BSTR szDesc)	
	VOID NotifyLog (USHORT nLogType, USHORT nLogCategory, BSTR szMsg, BSTR	
	szDesc)	
	VOID NotifyWarn (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	
	BSTR ChangLine (BSTR szLineMode, BSTR szTargetLineIP)	
	LONG ChangeProcessServer (BSTR szProcessName, BSTR szProcessIP)	
	LONG ChangeServerMode (BSTR szServerMode)	
Network	LONG GetCurLine (int* nCurLine)	
Function	INT GetLineStatus (int nLineMode)	
	BSTR GetNodeCurIP (BSTR szNodeName)	
	BSTR GetServerMode ()	
	BOOL IsDualLine ()	
	VARIANT GetSystemTime ()	
	VOID ProcessSleep(LONG dwMiliseconds)	
System	VOID RunApp (BSTR szAppPath)	
Function	VOID ShowAlarmPage(BSTR szAlarmTag)	
	VOID Sleep(LONG dwMiliseconds)	
	BSTR TimeConvert(DOUBLE dTimeValue, INT nTimeType)	
	LONG RecipeChangeData (BSTR ModelName, BSTR RecipeName)	
	LONG RecipeChangeUnit (BSTR ModelName, BSTR UnitName)	
	LONG RecipeControlSet (BSTR szModelName, BSTR szUnitName, BSTR	
	szDataName) LONG RecipeControlSetEx (SHORT nMode, BSTR szClientProgName, ULONG	
	hClientHandle, BSTR szModelName, BSTR szUnitName, BSTR szDataName, LONG*	
	plTransNo, SHORT* pnResult)	
	LONG RecipeControlWorkingSet ();	
Recipe Function	LONG RecipeControlWorkingSetEx (SHORT nMode, BSTR szClientProgName,	
	ULONG hClientHandle, LONG* plTransNo, SHORT* pnResult);	
	LONG RecipeDeleteData (BSTR ModelName, BSTR RecipeName)	
	LONG RecipeDeleteModel (BSTR ModelName)	
	LONG RecipeDeleteUnit (BSTR ModelName, BSTR UnitName)	
	LONG RecipeLoadDynamic (BSTR ModelName, BSTR UnitName)	
	LONG RecipeLoadSet (BSTR ModelName, BSTR UnitName, BSTR RecipeName)	
	LONG RecipeSaveAsArchive (BSTR ModelName, BSTR RecipeName)	
	LONG RecipeSetData (BSTR ModelName, BSTR ItemName, VARIANT setValue)	

Chapter 14 Function

Item	Relevant Function	
Database	VOID DataEvportPupWork (RSTP sz\WorkName)	
Function	VOID DataExportRunWork (BSTR szWorkName)	
	BOOL AckAllAlarm ()	
	BSTR GetAlarmCountErrInfo (LONG ICount)	
	LONG GetAlarmLogging (DOUBLE dtFrom, DOUBLE dtTo, BSTR szTag, BSTR	
	szAlarmGr, LONG IAlarmType, LONG *IOutCount, SAFEARRAY **pszRstrTag,	
	SAFEARRAY **pszRstrGr, SAFEARRAY **pdbRdtOccAlarm, SAFEARRAY	
	**pdbRdtRevAlarm, SAFEARRAY **pdbRdtAckAlarm, SAFEARRAY **plRnAlarmType,	
	SAFEARRAY **pszRstrMsg, SAFEARRAY **pszRstrAlarmKind)	
	LONG GetAlarmLoggingEx (DOUBLE dtFrom,DOUBLE dtTo,BSTR szTag, LONG nAlarmGroupCount, SAFEARRAY **pszAlarmGr,LONG lAlarmType, LONG	
	nAlarmGroupCount, SAFEARRAY **pszAlarmGr,LONG IAlarmType, LONG IAlarmGrade, BSTR szAlarmKind, LONG *IOutCount,	
	SAFEARRAY**pRstrTag,SAFEARRAY**pRstrGr, SAFEARRAY **pdbRdtOccAlarm,	
	SAFEARRAY **pdbRdtRevAlarm, SAFEARRAY **pdbRdtAckAlarm, SAFEARRAY	
	**plRnAlarmType, SAFEARRAY **pszRstrMsg, SAFEARRAY **pszRstrAlarmKind,	
	SAFEARRAY **pdbRvOccDValue, SAFEARRAY**pszRvOccSValue, SAFEARRAY	
	**plAlarmGrade)	
Alarm Function	LONG GetAlmCount (LONG Ikind, LONG IStatus)	
	LONG GetGroupUserInfoByTag (BSTR szTagName, LONG* , SAFEARRAY(BSTR)	
	*pszUserID, SAFEARRAY(BSTR) *pszUserName, SAFEARRAY(LONG) *plMailCheck,	
	SAFEARRAY(LONG) *pISMSMCheck, SAFEARRAY(BSTR) *pszMailAdd,	
	SAFEARRAY(BSTR) *pszHPhoneNo)	
	LONG GetHistAlmCount (LONG lkind, LONG lStatus)	
	LONG GetUserInfoList(LONG* IOutCount, SAFEARRAY(BSTR) *pszUserID,	
	SAFEARRAY(BSTR) *pszUserName, SAFEARRAY(BSTR) *pszDesc,	
	SAFEARRAY(BSTR) *pszPassword, SAFEARRAY(LONG) *plUserLevel,	
	SAFEARRAY(BSTR) *pszPhoneNo, SAFEARRAY(BSTR) *pszPhoneNo,	
	SAFEARRAY(BSTR) *pszMailAdd) LONG SetUserAlarm (BSTR szAlarmMessage, BSTR szAlarmGroup)	
	LONG SetUserAlarmEx (DOUBLE dOccur, LONG IAlarmLevel, BSTR	
	szAlarmMessage ,BSTR szAlarmGroup, DOUBLE dCur, DOUBLE dPrev)	
	LONG SetUserEvent (BSTR szEventMessage)	
	VOID StopAlarmSound()	
External Device	VOID PlaySound (BSTR szWaveFile)	
Function	VOID PlaySoundFile (BSTR szSoundFile)	
	BSTR GetUserID()	
User Function	LONG GetUserLevel()	
Oser i unction	LONG UserLogOff ()	
	LONG UserLogOn (BSTR szUserID, BSTR szPassWord)	
	VOID ReportDialog()	
	VOID ReportListRefresh()	
Report	VOID ReportPrint(BSTR szReportName)	
Function	VOID ReportPrintEx(BSTR szReportName, BSTR szReqTime)	
	VOID ReportSchedulerHide()	
	VOID ReportSchedulerShow()	

Item	Relevant Function	
	VOID GetHistDataCount(BSTR szModelName, BSTR , BSTR szDataKind, DOUBLE	
	dRefTime, LONG ICount, LONG IDir, LONG* IResultCount, SAFEARRAY(double)	
	*psadblTimeStamp, SAFEARRAY(double) *psadblNumValue, SAFEARRAY(LONG)	
	*psalStatus, SAFEARRAY(LONG) *psalResult)	
	VOID GetHistDataPer iod (BSTR szModelName, BSTR szTagName, BSTR szDataKind,	
	DOUBLE dFrom, DOUBLE dTo, LONG* IResultCount, SAFEARRAY(double)	
	*psadblTimeStamp, SAFEARRAY(double) *psadblNumValue, SAFEARRAY(LONG)	
	*psalStatus, SAFEARRAY(LONG) *psalResul)	
History Data	()	
Function	*psaszModelName, SAFEARRAY(BSTR) *psaszModelDesc, SAFEARRAY(BSTR)	
	*psaszModelType, SAFEARRAY(BSTR) *psaszArchvingCycle, SAFEARRAY(BSTR)	
	*psaszSegmentPeriod)	
	LONG GetLoggingTag(BSTR szModelName, LONG *lOutCount ,SAFEARRAY(BSTR)	
	*psaszTagName, SAFEARRAY(BSTR) *psaszTagType)	
	LONG SetHistData (BSTR szModelName, BSTR szTagName, BSTR szDataKind,	
	LONG IDataCount, SAFEARRAY(double) *psadblSaveTime, SAFEARRAY(double)	
	*psadblNumValue, SAFEARRAY(LONG) *psalTagStatus, SAFEARRAY(LONG)	
	*psalResult)	
	VOID ClearStaStatusCount (BSTR szChannelName, BSTR szStaName)	
	VOID EnableStation(BSTR szChannelName, BSTR szStaName, LONG IMode)	
	LONG GetCountReadSuccess(BSTR szChannelName, BSTR szStaName)	
	LONG GetCountReadTotal (BSTR szChannelName, BSTR szStaName)	
I/O Drive	, ,	
Function	LONG GetCountWriteTotal(BSTR szChannelName, BSTR szStaName)	
	LONG GetStatusRunMode(BSTR szChannelName, BSTR szStaName)	
	LONG GetStatusStaOK(BSTR szChannelName, BSTR szStaName)	
	VOID ShowNetworkStatus ()	
	VOID ShowStationInfo()	

14.2.2 List by alphabetical order

Alph abet	Function Name	Item
Α	BOOL AckAllAlarm ()	Alarm Function
С	BSTR ChangLine (BSTR szLineMode, BSTR szTargetLineIP)	Network Function
	LONG ChangeProcessServer (BSTR szProcessName, BSTR szProcessIP)	Network Function
	LONG ChangeServerMode (BSTR szServerMode)	Network Function
	LONG ChangePage (BSTR szOldPage, BSTR szNewPage)	Screen Function
	VOID ClearStaStatusCount (BSTR szChannelName, BSTR szStaName)	I/O Drive Function
	LONG ClosePage (BSTR szPageName)	Screen Function
	LONG CloseProject ()	Project Function
E	VOID EnableStat ion(BSTR szChannelName, BSTR szStaName, LONG	I/O Drive Function
	IMode)	I/O DIIVE FUIICIIOII
D	VOID DataExportRunWork (BSTR szWorkName)	Database Function

Chapter 14 Function

Alph abet	Function Name	Item
G	LONG GetAlmCount (LONG lkind, LONG IStatus)	Alarm Function
	BSTR GetAlarmCountErrInfo (LONG ICount)	Alarm Function
	LONG GetAlarmLogging (DOUBLE dtFrom, DOUBLE dtTo, BSTR szTag, BSTR szAlarmGr, LONG IAlarmType, LONG *IOutCount, SAFEARRAY **pszRstrTag, SAFEARRAY **pszRstrGr, SAFEARRAY **pdbRdtOccAlarm, SAFEARRAY **pdbRdtRevAlarm, SAFEARRAY **pdbRdtAckAlarm, SAFEARRAY **plRnAlarmType, SAFEARRAY **pszRstrMsg, SAFEARRAY **pszRstrAlarmKind)	Alarm Function
	LONG GetAlarmLoggingEx (DOUBLE dtFrom,DOUBLE dtTo,BSTR szTag, LONG nAlarmGroupCount, SAFEARRAY **pszAlarmGr,LONG lAlarmType, LONG lAlarmGrade, BSTR szAlarmKind, LONG *lOutCount, SAFEARRAY**pRstrTag,SAFEARRAY**pRstrGr, SAFEARRAY **pdbRdtOccAlarm, SAFEARRAY **pdbRdtRevAlarm, SAFEARRAY **pdbRdtAckAlarm, SAFEARRAY **plRnAlarmType, SAFEARRAY **pszRstrMsg, SAFEARRAY **pszRstrAlarmKind, SAFEARRAY **pdbRvOccDValue, SAFEARRAY**pszRvOccSValue, SAFEARRAY **plAlarmGrade)	Alarm Function
	LONG GetCountReadSuccess(BSTR szChannelName, BSTR szStaName)	I/O Drive Function
	LONG GetCountReadTotal (BSTR szChannelName, BSTR szStaName)	I/O Drive Function
	LONG GetCountWriteSuccess(BSTR szChannelName, BSTR szStaName)	I/O Drive Function
	LONG GetCountWriteTotal(BSTR szChannelName, BSTR szStaName)	I/O Drive Function
	LONG GetCurLine (int* nCurLine)	Network Function
	LONG GetGroupUserInfoByTag (BSTR szTagName, LONG*, SAFEARRAY(BSTR) *pszUserID, SAFEARRAY(BSTR) *pszUserName, SAFEARRAY(LONG) *plMailCheck, SAFEARRAY(LONG) *plSMSMCheck, SAFEARRAY(BSTR) *pszMailAdd, SAFEARRAY(BSTR) *pszHPhoneNo)	Alarm Function
	LONG GetHistAlmCount (LONG Ikind, LONG IStatus)	Alarm Function
	VOID GetHistDataCount (BSTR szModelName, BSTR , BSTR szDataKind, DOUBLE dRefTime, LONG lCount, LONG lDir, LONG* lResultCount, SAFEARRAY(double) *psadblTimeStamp, SAFEARRAY(double) *psadblNumValue, SAFEARRAY(LONG) *psalStatus, SAFEARRAY(LONG) *psalResult)	History Data Function
	VOID GetHistDataPer iod (BSTR szModelName, BSTR szTagName, BSTR szDataKind, DOUBLE dFrom, DOUBLE dTo, LONG* IResultCount, SAFEARRAY(double) *psadblTimeStamp, SAFEARRAY(double) *psadblNumValue, SAFEARRAY(LONG) *psalStatus, SAFEARRAY(LONG) *psalResul)	History Data Function
	VOID GetHWND(ULONG* hwnd)	Screen Function
	INT GetLineStatus (int nLineMode)	Network Function

Alph	Function Name	Item
abet		1.0111
G	LONG GetLoggingModelList (LONG *IOutCount ,SAFEARRAY(BSTR) *psaszModelName, SAFEARRAY(BSTR) *psaszModelDesc, SAFEARRAY(BSTR) *psaszModelType, SAFEARRAY(BSTR) *psaszArchvingCycle, SAFEARRAY(BSTR) *psaszSegmentPeriod)	History Data Function
	LONG GetLoggingTag (BSTR szModelName, LONG *IOutCount ,SAFEARRAY(BSTR) *psaszTagName, SAFEARRAY(BSTR) *psaszTagType)	History Data Function
	BSTR GetNodeCuriP (BSTR szNodeName)	Network Function
	BSTR GetProjectName ()	Project Function
	BSTR GetProjectPath ()	Project Function
	BSTR GetProjectType ()	Project Function
	BSTR GetServerMode ()	Network Function
	LONG GetStatusRunMode (BSTR szChannelName, BSTR szStaName)	I/O Drive Function
	LONG GetStatusStaOK (BSTR szChannelName, BSTR szStaName)	I/O Drive Function
	VARIANT GetSystemTime ()	System Function
	ULONG GetTagStatus (BSTR szTagName)	Tag Function
	BSTR GetTagStatusDesc(ULONG ITagStatus)	Tag Function
	VARIANT GetTagVal (BSTR szTagName)	Tag Function
	VARIANT GetTagValbyTagId (LONG ITagId)	Tag Function
	BSTR GetTagVaInStatusByTagId (LONG ITagId, VARIANT* vtTagVaI, LONG *IScanMode, LONG *IMainQ, LONG *ISubQCnt, LONG *ISubQ1, LONG *Tag Function *ISubQ2, LONG *ISubQ3, LONG *ISubQ4, LONG *ISubQ5)	
	LONG GetTagValueStatus (BSTR szTagName, VARIANT* vtTagVal)	Tag Function
	BSTR GetUserID()	User Function
	LONG GettUserInfoList (LONG* IOutCount, SAFEARRAY(BSTR) *pszUserID, SAFEARRAY(BSTR) *pszUserName, SAFEARRAY(BSTR) *pszDesc, SAFEARRAY(BSTR) *pszPassword, SAFEARRAY(LONG) *plUserLevel, SAFEARRAY(BSTR) *pszHPhoneNo, SAFEARRAY(BSTR) *pszMailAdd)	Alarm Function
	LONG GetUserLevel()	User Function
Н	VOID HardCopy()	Screen Function
I	BOOL IsDualLine ()	Network Function
	BOOL IsRunning ()	Screen Function

Chapter 14 Function

Alph	Function Name	Item
abet	LONG NotifyAlarmEvent (DOUBLE IIOccur, USHORT nAlmLevel, BSTR	
N	szMessage, BSTR szArea, BSTR szOUser, BSTR zInformation, BSTR szUserDefine1, BSTR szUserDefine2, BSTR szUserDefine3, BSTR szUserDefine4, BSTR szPrev, BSTR szCur, BSTR szObject)	Event Function
	VOID NotifyError (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	Log Function
	VOID NotifyHostError (BSTR szHostName, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	Log Function
	VOID NotifyHostInfo (BSTR szHostName, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	Log Function
	VOID NotifyHostLog (USHORT nLogType, USHORT nLogCategory, BSTR szHostName, BSTR szMsg, BSTR szDesc)	Log Function
	VOID NotifyHostWarn (BSTR szHostName, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	Log Function
	VOID NotifyInfo USHORT nLogCategory (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	Log Function
	VOID NotifyLog (USHORT nLogType, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	Log Function
	VOID NotifyWarn (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	Log Function
0	VOID OpenPage (BSTR szPageName, BSTR szPageParam)	Screen Function
	LONG OpenPageObject (BSTR szPageName, BSTR szObjectName)	Screen Function
	LONG OpenProject (BSTR szProjectName, BSTR szProjectPath)	Project Function
Р	VOID PlaySound (BSTR szWaveFile))	External
	VOID 1 Idy Ood IId (DO 111 32 VI d VOI IIO))	Device Function
	VOID PlaySoundFile (BSTR szSoundFile)	External
	· · ·	Device Function
	VOID ProcessSleep(LONG dwMiliseconds)	System Function
Q	VOID Quit()	Screen Function
	LONB QuitProject ()	Project Function
R	LONG RecipeChangeData (BSTR ModelName, BSTR RecipeName)	Recipe Function
	LONG RecipeChangeUnit (BSTR ModelName, BSTR UnitName)	Recipe Function
	LONG RecipeControlSet (BSTR szModelName, BSTR szUnitName, BSTR szDataName)	Recipe Function
	LONG RecipeControlSetEx (SHORT nMode, BSTR szClientProgName, ULONG hClientHandle, BSTR szModelName, BSTR szUnitName, BSTR szDataName, LONG* plTransNo, SHORT* pnResult)	Recipe Function
	LONG RecipeControlWorkingSet ();	Recipe Function
	LONG RecipeControlWorkingSetEx (SHORT nMode, BSTR szClientProgName, ULONG hClientHandle, LONG* plTransNo, SHORT* pnResult);	Recipe Function
	LONG RecipeDeleteData (BSTR ModelName, BSTR RecipeName)	Recipe Function
	LONG RecipeDeleteModel (BSTR ModelName)	Recipe Function
	LONG RecipeDeleteUnit (BSTR ModelName, BSTR UnitName)	Recipe Function
	LONG RecipeLoadDynamic (BSTR ModelName, BSTR UnitName)	Recipe Function
	LONG RecipeLoadSet (BSTR ModelName, BSTR UnitName, BSTR RecipeName)	Recipe Function
	LONG RecipeSaveAsArchive (BSTR ModelName, BSTR RecipeName)	Recipe Function

Alph		
abet	Function Name	ltem
R	LONG RecipeSetData (BSTR ModelName, BSTR ItemName, VARIANT	Recipe Function
	setValue)	
	VOID ReportDialog()	Report Function
	VOID ReportListRefresh()	Report Function
	VOID ReportPrint(BSTR szReportName)	Report Function
	VOID ReportPrintEx(BSTR szReportName, BSTR szReqTime)	Report Function
	VOID ReportSchedulerHide()	Report Function
	VOID ReportSchedulerShow()	Report Function
	LONG ResetStaticValue (BSTR szTagName)	Tag Function
	VOID RunApp (BSTR szAppPath)	System Function
	VOID RunScript (BSTR szScriptName)	Script Function
S	VOID SendEmail (BSTR szMailAddTo, BSTR szMailAddFrom, BSTR szMailTitle , BSTR szMailMsg)	Event Function
	VOID SendSMS (BSTR szRecvPhoneNo, BSTR szCallBackPhoneNo, BSTR szSMSMsg)	Event Function
	LONG SetHistData (BSTR szModelName, BSTR szTagName, BSTR szDataKind, LONG IDataCount, SAFEARRAY(double) *psadblSaveTime, SAFEARRAY(double) *psadblNumValue, SAFEARRAY(LONG) *psalTagStatus, SAFEARRAY(LONG) *psalResult)	History Data Function
	VOID SetTagVal (BSTR szTagName, VARIANT vtTagVal)	Tag Function
	LONG SetTagValEx (SHORT nMode, BSTR szTagName, VARIANT vtTagVal, BSTR szClientName, ULONG hClientHandle)	Tag Function
	LONG SetUserAlarm (BSTR szAlarmMessage, BSTR szAlarmGroup)	Alarm Function
	LONG SetUserAlarmEx (DOUBLE dOccur, LONG IAlarmLevel, BSTR szAlarmMessage ,BSTR szAlarmGroup, DOUBLE dCur, DOUBLE dPrev)	Alarm Function
	LONG SetUserEvent (BSTR szEventMessage)	Alarm Function
	VOID ShowAlarmPage(BSTR szAlarmTag)	System Function
	VOID ShowCrossReference (BSTR szTagName)	Tag Function
	VOID ShowNetworkStatus ()	I/O Drive Function
	VOID ShowRawDataInfo()	Tag Function
	VOID ShowSIMCfgInfo()	Tag Function
	VOID ShowStationInfo()	I/O Drive Function
	VOID Sleep(LONG dwMiliseconds)	System Function
	VOID StopAlarmSound()	Alarm Function
Т	BSTR TimeConvert(DOUBLE dTimeValue, INT nTimeType)	System Function
U	LONG UserLogOff ()	User Function
	LONG UserLogOn (BSTR szUserID, BSTR szPassWord)	User Function

14.3 Function Description and Example

14.3.1 Project Function

Item	Description
Туре	LONG CloseProject ()
Kind	Project Function
Description	Close current project
Example	Sub Main
	Dim IRet As Long
	<pre>IRet = ole.CloseProject()</pre>
	If IRet <> 0 Then
	MsgBox "CloseProject() False"
	Else
	MsgBox "CloseProject() OK"
	End If
	End Sub

Item	Description
Туре	BSTR GetProjectName ()
Kind	Project Function
Description	Get current project name.
Example	Sub Main
	Dim PrjName As String
	PrjName = ole.GetProjectName()
	If IRet <> 0 Then
	MsgBox "GetProjectName()False"
	Else
	MsgBox "GetProjectName() OK " + PrjName
	End If
	End Sub

Item	Description
Туре	BSTR GetProjectPath ()
Kind	Project Function
Description	Get current project path.
Example	Sub Main
	Dim PrjName As String
	PrjPath = ole.GetProjectPath()
	If IRet <> 0 Then
	MsgBox "GetProjectPath() False"
	Else
	MsgBox "GetProjectPath() OK " + PrjPath
	End If
	End Sub

Item	Description
Туре	BSTR GetProjectType ()
Kind	Project Function
Description	Get current project type
Example	Sub Main
	Dim PrjType As String
	PrjType = ole.GetProjectType
	MsgBox "GetProjectType : " + PrjType
	End Sub

Item	Description
Туре	LONG OpenProject(BSTR szProjectName, BSTR szProjectPath)
Kind	Project Function
Description	Open Project.
	szProjectName : Project Name
	szProjectPath : Project Path
Example	Sub Main
	Dim IRet As Long
	<pre>IRet = ole.OpenProject("ProjectName", "ProjectPath")</pre>
	If IRet <> 0 Then
	MsgBox "OpenProject() False"
	Else
	MsgBox "OpenProject() OK"
	End If
	End Sub

Item	Description
Туре	LONG QuitProject ()
Kind	Project Function
Description	Close current project
Example	Sub Main
	Dim IRet As Long
	IRet = ole.QuitProject()
	If IRet <> 0 Then
	MsgBox " QuitProject () False"
	Else
	MsgBox " QuitProject () OK"
	End If
	End Sub

14.3.2 Screen Function

Item	Description
Туре	LONG ChangePage (BSTR szOldPage, BSTR szNewPage)
Kind	Screen Function
Description	Close an existing page and open a new page
	szOldPage: Old page
	szNewPage: New page
Example	Sub Main
	Dim IRet As Long
	IRet = ole.ChangePage("Page1.ivd", "Page2.ivd")
	If IRet <> 0 Then
	MsgBox "ChangePage() False"
	Else
	MsgBox "ChangePage() OK"
	End If
	End Sub

Item	Description
Туре	LONG ClosePage (BSTR szPageName)
Kind	Screen Function
Description	Closes the specified page
	szPageName : Page name
Example	Sub Main
	Dim IRet As Long
	IRet = ole.ClosePage("page.ivd")
	If IRet <> 0 Then
	MsgBox "ClosePage() False"
	Else
	MsgBox "ClosePage() OK"
	End If
	End Sub

Item	Description
Type	VOID GetHWND(ULONG* hwnd)
Kind	Screen Function
Description	Get handle of runtime program.
	hwnd : Get handle of runtime program
Example	Sub Main
	Dim IRet As Long
	IRet = ole.GetHWND
	If IRet <> 0 Then
	MsgBox "GetHWND()OK" + CStr(IRet)
	Else
	MsgBox "GetHWND() False => IRet : " + CStr(IRet)
	End If
	End Sub

Item	Description
Туре	VOID HardCopy()
Kind	Screen Function
Description	Prints the full screen associated with the graphic runtime window.
Example	Sub Main
	ole.HardCopy
	End Sub

Item	Description
Туре	BOOL IsRunning ()
Kind	Screen Function
Description	Get current run-time status.
Example	Sub Main
	Dim bRet As Boolean
	bRet = ole.IsRunning
	If bRet = True Then
	MsgBox "IsRunning OK"
	Else
	MsgBox "IsRunning FALSE"
	End If
	End Sub

Item	Description
Туре	VOID OpenPage (BSTR szPageName, BSTR szPageParam)
Kind	Screen Function
Description	Opens the file whose page name is specified by szPageName
	szPageName: Graphic page name
	szPageParam: Replace name for template screen.
Example	Sub Main
	Dim IRet As Long
	IRet = ole. OpenPage ("PAGE1.ivd", "\$GR=PLC1")
	If IRet <> 0 Then
	MsgBox "OpenPage() False"
	Else
	MsgBox "OpenPage () OK"
	End If
	End Sub

Item	Description
Туре	LONG OpenPageObject (BSTR szPageName, BSTR szObjectName)
Kind	Screen Function
Description	Opens the file whose pagename is specified by szPageName.
	szPageName: Graphic page name
	szObjectName: Replace name for template screen
Example	Sub Main
	Dim IRet As Long
	<pre>IRet = ole.OpenPageObject("Page.ivd", "ObjectName")</pre>
	If IRet <> 0 Then
	MsgBox "OpenPageObject() False"
	Else
	MsgBox "OpenPageObject() OK"
	End If
	End Sub

Item	Description
Туре	VOID Quit()
Kind	Screen Function
Description	Terminates the execution of InfoU rumtime.
Example	Sub Main
	ole.Quit
	End Sub

14.3.3 Tag Function

Item	Description
Туре	ULONG GetTagStatus (BSTR szTagName)
Kind	Tag Function
Description	Get the current tag status.
	szTagName : Tag Name
Example	Sub Main
	Dim IRet As Long
	IRet = ole.GetTagStatus("SCAnaTag")
	If IRet <> 0 Then
	MsgBox "GetTagStatus() OK"
	Else
	MsgBox "GetTagStatus() False"
	End If
	End Sub

Item	Description
Туре	BSTR GetTagStatusDesc (ULONG ITagStatus)
Kind	Tag Function
Description	Let you know the explanation of the tag status by string.
	ITagStatus: Tag Status Value
Example	Sub Main
	Dim IRet As Long
	Dim szDesc As String
	IRet = ole.GetTagValueStatus("tagName")
	szDesc = ole.GetTagStatusDesc(IRet)
	MsgBox szDesc
	End Sub

Item	Description
Туре	VARIANT GetTagVal (BSTR szTagName)
Kind	Tag Function
Description	Get tag value.
	szTagName: Tag Name
Example	Sub Main
	Dim Value As Variant
	Value = 12.0
	Value = ole.GetTagVal("FAOA_F_EHV02.AR")
	MsgBox "FAOA_F_EHV02.AR Value = " + CStr(Value)
	End Sub

Item	Description
Туре	VARIANT GetTagValbyTagId (LONG ITagId)
Kind	Tag Function
Description	Get tag values by tag id
	ITagId: Tag ID (Index)
Example	Sub Main
	Dim Value As Variant
	Dim Tagid As Long
	Value = ole.GetTagValbyTagId(Tagid)
	MsgBox "SCDigTag Value = " + Value
	End Sub

Item	Description
Туре	BSTR GetTagValnStatusByTagId (LONG ITagId, VARIANT* vtTagVal, LONG
	*IScanMode, LONG *IMainQ, LONG *ISubQCnt, LONG *ISubQ1, LONG
	*ISubQ2, LONG *ISubQ3, LONG *ISubQ4, LONG *ISubQ5)
Kind	Tag Function
Description	Get tag status by tag id. The return values is the description of tag status
	ITagId: Tag ID (Index)
	vtTagVal: Tag Value
	IScanMode: The main mode of tag status. (1:SCANSTOP, 2:SCANSTART)
	IMainQ: Main Quality Code
	ISubQCnt: Detail Quality Code Count
	ISubQ1: Detail Quality Code1
	ISubQ2: Detail Quality Code2
	ISubQ3: Detail Quality Code3
	ISubQ4: Detail Quality Code4
	ISubQ5: Detail Quality Code5

Description
G GetTagValueStatus (BSTR szTagName, VARIANT *vtTagVal)
Function
In the tag value and status. In this time, the return value let you know the us value TagName: Tag Name gVal: Tag Value
Main Dim IRet As Long Dim TagValue As Variant IRet = ole.GetTagValueStatus("Tag Name", TagValue) If IRet <> 0 Then
Su

Item	Description
Туре	LONG ResetStaticValue (BSTR szTagName)
Kind	Tag Function
Description	Reset statistic value of specified tag
	szTagName: Tag Name
Example	Sub Main
	Dim IRet As Long
	IRet = ole.ResetStaticValue("SCAnaTag")
	If (IRet = 0) Then
	MsgBox "ResetStaticValue() OK "
	Else
	MsgBox "ResetStaticValue() FALSE "
	End If
	End Sub

Item	Description
Туре	LONG ShowCrossReference (BSTR szTagName)
Kind	Tag Function
Description	Inform you where specific tag is being used currently.
	szTagName : Tag Name
Example	Sub Main
	ole.ShowCrossReference("SCADA.UT_EHV07KWHRESET_")
	End Sub

Item	Description
Туре	VOID ShowRawDataInfo()
Kind	Tag Function
Description	Display Raw Data Information dialog.
Example	Sub Main
	ole. ShowRawDataInfo
	End Sub

Item	Description
Туре	VOID ShowSIMCfgInfo()
Kind	Tag Function
Description	Display Simulation Setting dialog.
Example	Sub Main
	ole. ShowSIMCfgInfo
	End Sub

Item	Description
Туре	VOID SetTagVal (BSTR szTagName, VARIANT vtTagVal)
Kind	Tag Function
Description	Control tag
	szTagName: Tag Name
	vtTagVal: Tag value which was controled
Example	Sub Main
	Dim Value As Variant
	MsgBox " Before : SetTagVal(SCDigTag) Value = "
	+ ole.GetTagVal("tagName")
	Value = 1
	ole.SetTagVal("tagName", Value)
	MsgBox " After : SetTagVal(Tag Name) Value = "
	+ ole.GetTagVal("tagName")
	End Sub

Item	Description
Туре	LONG SetTagValEx (SHORT nMode, BSTR szTagName, VARIANT
	vtTagVal, BSTR szClientName, ULONG hClientHandle)
Kind	Tag Function
Description	Control specified tag value (nMode: 1:No Return, 2:Return). The program
	which want to receive control result must pass it's handle value.
	nMode: Control Mode (nMode: 1:No Return, 2:Return)
	szTagName: Tag Name
	vtTagVal: Tag Value to Control
	szClientName: Client Program Name that perform control
	hClientHandle: Client Window Handle that perform control
Example	Sub Main
	Dim Value As Variant
	Dim TransNo As Long
	MsgBox " Before : SetTagValEx(SCDigTag) Value = "
	+ ole.GetTagVal("tagName")
	Value = 1
	TransNo = ole.SetTagVaLEx (1, "tagName", Value ,
	"Script", NULL)
	MsgBox " After : SetTagValEx(tagName) Value = "
	+ ole.GetTagVal("tagName")
	End Sub

14.3.4 Script Function

Item	Description
Туре	VOID RunScript (BSTR szScriptName)
Kind	Script Function
Description	Run script.
	szScriptName: Script name
Example	Sub Main
	Dim scTime As Variant
	ole.RunScript("ScriptName")
	End Sub

14.3.5 Event Function

Item	Description
Туре	LONG NotifyAlarmEvent (DOUBLE IIOccur, USHORT nAlmLevel, BSTR
	szMessage, BSTR szArea, BSTR szOUser, BSTR szInformation, BSTR
	szUserDefine1, BSTR szUserDefine2, BSTR szUserDefine3, BSTR
	szUserDefine4, BSTR szPrev, BSTR szCur, BSTR szObject)
Kind	Event Function
Description	Occur Alarm & Event.
	IlOccur: Occurrence Time. NULL = current time
	nAlmLevel: The level of alarm
	szMessage: Message
	szArea: Occurrence Region
	szOUser: Occurrence User
	szInformation: Information
	szUserDefine1: User Definition1
	szUserDefine2: User Definition2
	szUserDefine3: User Definition3
	szUserDefine4: User Definition4
	szPrev: Pervious Value of Message
	szCur. New Value of Message
	szObject: Message Object
Example	Sub Main
	Dim IRet As Long
	Dim dDate As Double
	Dim iLevel As Integer
	ole.ChangeProcessServer("ALM_PROC","150.150.32.60")
	If IRet <> 0 Then
	MsgBox "ChangeProcessServer() False : ret = " + CStr(IRet)
	Else
	MsgBox "ChangeProcessServer() OK : ret = " + CStr(IRet)
	End If
	dDate = NUL
	iLevel = 1
	IRet = ole.NotifyAlarmEvent(dDate, iLevel,"NotifyAlarmEvent Test", _ "Script", "Script", "Interfun API Success", "", "", "",
	"", "before bun", "after Run","")
	If IRet <> 0 Then
	MsgBox "NotifyAlarmEvent() False : ret = "
	+ CStr(IRet)
	Else
	MsgBox "NotifyAlarmEvent() OK : ret = " + CStr(IRet)
	End If
	End Sub

Item	Description
Туре	VOID SendEmail (BSTR szMailAddTo, BSTR szMailAddFrom, BSTR
	szMailTitle , BSTR szMailMsg)
Kind	Event Function
Description	Send E-Mail
	szMailAddTo: E-Mail Sender
	szMailAddFrom : E-Mail Receiver
	szMailTitle: E-Mail Title
	szMailMsg : E-Mail Body
Example	Sub Main
	ole.SendEmail("receiver@lsis.biz","sender@lsis.biz","title","message:
	Script Test Good~~!!")
	End Sub

Item	Description
Туре	VOID SendSMS (BSTR szRecvPhoneNo, BSTR szCallBackPhoneNo, BSTR
	szSMSMsg)
Kind	Event Function
Description	Send SMS.
	szRecvPhoneNo : The phone number of SMS receiver
	szCallBackPhoneNo : The phone number of SMS sender
	szSMSMsg: SMS Message (English:80, Korean:40)
Example	Sub Main
	ole.SendSMS("0194851234","0194850000","Script Test")
	End Sub

14.3.6 Log Function

Item	Description
Туре	VOID NotifyError (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)
Kind	Log Function
Description	Error Logs are recorded.
	nLogCategory: Log Category (Default : LOG_CATE_SCRIPT= 1, 0)
	szMsg: Log Message
	szDesc: Log Description
Example	Sub Main
	ole.NotifyError(1, "ERROR", "Notify_ERROR")
	End Sub

Item	Description
Туре	VOID NotifyHostError (BSTR szHostName, USHORT nLogCategory, BSTR
	szMsg, BSTR szDesc)
Kind	Log Function
Description	Error Logs are recorded to specified computer.
	szHostName: Target computer Name
	nLogCategory: Log Category (Default : LOG_CATE_SCRIPT= 1, 0)
	szMsg: Log Message
	szDesc: Log Description
Example	Sub Main
	ole.NotifyHostError("127.0.0.1",1, "HOST_ERROR",
	"Notify_HOST_ERROR")
	End Sub

Item	Description
Туре	VOID NotifyHostInfo (BSTR szHostName, USHORT nLogCategory, BSTR
	szMsg, BSTR szDesc)
Kind	Log Function
Description	Information Logs are recorded to specified computer.
	szHostName: Target computer Name
	nLogCategory: Log Category (Default : LOG_CATE_SCRIPT= 1, 0)
	szMsg: Log Message
	szDesc: Log Description
Example	Sub Main
	ole.NotifyHostInfo ("127.0.0.1",1, "HOST_INFO",
	"Notify_HOST_INFO")
	End Sub

Item	Description
Туре	VOID NotifyHostLog (USHORT nLogType, USHORT nLogCategory, BSTR
	szHostName, BSTR szMsg, BSTR szDesc)
Kind	Log Function
Description	Logs are recorded to specified computer.
	nLogType: Log type(1 = INFO_LOG_TYPE(information), 2 =
	WARN_LOG_TYPE (warning), 3 = ERRO_LOG_TYPE(error))
	nLogCategory: Log Category (Default : LOG_CATE_SCRIPT= 1, 0)
	szHostName: Target computer Name
	szMsg: Log Message
	szDesc: Log Description
Example	Sub Main
	ole.NotifyHostLog(1, 1, "127.0.0.1", "HOST_INFO",
	"NotifyHostLog INFO")
	ole.NotifyHostLog(2, 1, "127.0.0.1", "HOST_WARN",
	"NotifyHostLog _ WARN")
	ole.NotifyHostLog(3, 1, "127.0.0.1","HOST_ERROR",
	"NotifyHostLog _ ERROR")
	End Sub

Item	Description
Туре	VOID NotifyHostWarn (BSTR szHostName, USHORT nLogCategory, BSTF
	szMsg, BSTR szDesc)
Kind	Log Function
Description	Warning Logs are recorded.
	szHostName: Target computer Name
	nLogCategory: Log Category (Default : LOG_CATE_SCRIPT= 1, 0)
	szMsg: Log Message
	szDesc: Log Description
Example	Sub Main
	ole.NotifyHostWarn ("127.0.0.1",1, "HOST_WARN'
	"Notify_HOST_WARN")
	End Sub

Item	Description
Туре	VOID NotifyInfo (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)
Kind	Log Function
Description	Information Logs are recorded. nLogCategory: Log Category (Default : LOG_CATE_SCRIPT= 1, 0) szMsg: Log Message szDesc: Log Description
Example	Sub Main ole.NotifyInfo(1, "INFO", "Notify_INFO") End Sub

Item	Description
Туре	VOID NotifyLog (USHORT nLogType, USHORT nLogCategory, BSTR szMsg,
	BSTR szDesc)
Kind	Log Function
Description	Logs are recorded
	nLogType: Log type (1 = INFO_LOG_TYPE(Information), 2 =
	WARN_LOG_TYPE (Warning), 3 = ERRO_LOG_TYPE(Error))
	nLogCategory: Log Category (Default : LOG_CATE_SCRIPT= 1, 0)
	szMsg: Log Message
	szDesc: Log Description
Example	Sub Main
	ole.NotifyLog(1, "INFO", 1,"NotifyLog _ INFO")
	ole.NotifyLog(2, "WARN", 1, "NotifyLog _ WARN")
	ole.NotifyLog(3, "ERROR",1,"NotifyLog _ ERROR")
	End Sub

Item	Description
Туре	VOID NotifyWarn (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)
Kind	Log Function
Description	Warning Logs are recorded nLogCategory: Log Category (Default : LOG_CATE_SCRIPT= 1, 0) szMsg: Log Message szDesc: Log Description
Example	Sub Main
	ole.NotifyWarn(1, "WARN", "Notify_WARN")
	End Sub

14.3.7 Network Function

Item	Description
Type	BSTR ChangLine (BSTR szLineMode, BSTR szTargetLineIP)
Kind	Network Function
Description	Changes a network line with designation IP address
	szLineMode: line mode (DEFAULT, SUB)
	szTargetLineIP: line IP address
Example	Sub Main
	Dim bRet As Boolean
	bRet = ole.ChangLine("DEFAULT", "xxx.xxx.xxx.xxx")
	If bRet <> 0 Then
	MsgBox "ChangLine FALSE"
	Else
	MsgBox "ChangLine OK"
	End If
	End Sub

Item	Description
Туре	LONG ChangeProcessServer (BSTR szProcessName, BSTR szProcessIP)
Kind	Network Function
Description	Changes with process of network address (IP).
	szProcessName: process name
	szProcessIP: target computer's IP address
Example	Sub Main
	Dim IRet As Long
	IRet = ole.ChangeProcessServer(
	"ALM_PROC","xxx.xxx.xxx.xxx")
	If IRet <> 0 Then
	MsgBox "ChangeProcessServer() False"
	Else
	MsgBox "ChangeProcessServer() OK "
	End If
	End Sub

Item	Description
Туре	LONG ChangeServerMode (BSTR szServerMode)
Kind	Network Function
Description	Changes a server mode.(redundancy option only)
	szServerMode: target mode (NORMAL, RUN, STANDBY)
Example	Sub Main
	Dim IRet As Long
	IRet = ole.ChangeServerMode("NORMAL")
	If IRet <> 0 Then
	MsgBox "ChangeServerMode()False"
	Else
	MsgBox "ChangeServerMode() OK"
	End If
	End Sub

Item	Description
Туре	LONG GetCurLine (int* nCurLine)
Kind	Network Function
Description	Get current used line.(line redundancy option only)
	nCurLine: network line(0: Default IP, 1: Sub IP , 2: stop)
Example	Sub Main
	Dim IRet As Long
	Dim ILineSatus As Long
	IRet = ole.GetCurLine(ILineSatus)
	If IRet <> 0 Then
	MsgBox "GetCurLine()False => Status "
	+ CStr(ILineSatus)
	Else
	MsgBox "GetCurLine() OK => Status "
	+ CStr(ILineSatus)
	End If
	End Sub

Item	Description
Туре	INT GetLineStatus (INT nLineMode)
Kind	Network Function
Description	Get current line status
	nLineMode: line status
Example	Sub Main
	Dim IRet As Long
	Dim ILineStatus As Long
	ILineStatus = 0 IRet = ole.GetLineStatus(ILineStatus)
	If (IRet = 1) Then
	MsgBox "GetLineStatus() OK: " + CStr(ILineStatus) Else
	MsgBox "GetLineStatus() FALSE: "
	+ CStr(ILineStatus)
	End If
	End Sub

Item	Description
Туре	BSTR GetNodeCuriP (BSTR szNodeName)
Kind	Network Function
Description	Get current used node IP address
	szNodeName: node name(computer name)
Example	Sub Main
	Dim IP As String
	IP = ole.GetNodeCurlP("SERVER")
	MsgBox "GetNodeCurIP() => SERVER " + IP
	End Sub

Item	Description
Туре	BSTR GetServerMode ()
Kind	Network Function
Description	Get server mode(mode: run, standby)
Example	Sub Main
	Dim SvrMode As String
	SvrMode = ole.GetServerMode
	MsgBox "GetServerMode : " + SvrMode
	End Sub

Item	Description
Туре	BOOL IsDualLine ()
Kind	Network Function
Description	Get network line redundancy settings
Example	Sub Main
	Dim bRet As Boolean
	bRet = ole.IsDualLine
	If bRet = True Then
	MsgBox "IsDualLine OK"
	Else
	MsgBox "IsDualLine FALSE"
	End If
	End Sub

14.3.8 System Function

Item	Description
Туре	VARIANT GetSystemTime ()
Kind	System Function
Description	Get current system time
Example	Sub Main
	Dim scTime As Variant
	scTime = ole.GetSystemTime
	If IRet <> 0 Then
	MsgBox "GetSystemTime() False"
	Else
	MsgBox "GetSystemTime() OK" + scTime
	End If
	End Sub

Item	Description
Туре	VOID ProcessSleep(LONG dwMiliseconds)
Kind	System Function
Description	Suspends the current process for a specified time
	dwMiliseconds: millisecond
Example	Sub Main
	Dim scTime As Variant
	ole.ProcessSleep(1000)
	End Sub

Item	Description
Туре	VOID RunApp (BSTR szAppPath)
Kind	System Function
Description	Excute application. (include path)
	szAppPath: application path
Example	Sub Main
	Dim scTime As Variant
	ole.RunApp("C:\WINDOWS\System32\mspaint.exe")
	End Sub

Item	Description
Туре	VOID ShowAlarmPage(BSTR szAlarmTag)
Kind	System Function
Description	This function displays the alarm page with tag which is occured alram
	szAlarmTag: Tag name
Example	Sub Main
	Dim scTime As Variant
	ole. ShowAlarmPage ("Tag name")
	End Sub

Item	Description
Туре	VOID Sleep(LONG dwMiliseconds)
Kind	System Function
Description	Suspends the current script for a specified time
	dwMiliseconds: millisecond
Example	Sub Main
	Dim scTime As Variant
	ole. Sleep(1000)
	End Sub

Item	Description
Туре	BSTR TimeConvert(DOUBLE dTimeValue, INT nTimeType)
Kind	System Function
Description	According to format outputs a time in string
	dTimeValue current time. (ex <=0 : curent time)
	nTimeType :
	0 : HH:MM
	1 : HH:MM AM/PM
	2 : HH:MM:SS
	3 : HH:MM:SS AM/PM
	7 : mm/dd
	8 : mm-dd
	10 : yy/mm/dd
	11 : yy-mm-dd
	13 : yyyy/mm/dd
	14 : yyyy-mm-dd
	15 : mm/dd HH:MM
	16 : mm/dd HH:MM AM/PM
	17 : mm/dd HH:MM:SS
	18 : mm/dd HH:MM:SS AM/PM
	19 : mm-dd HH:MM
	20 : mm-dd HH:MM AM/PM
	21 : mm-dd HH:MM:SS
	22 : mm-dd HH:MM:SS AM/PM
	23 : yy/mm/dd HH:MM
	24 : yy/mm/dd HH:MM AM/PM
	25 : yy/mm/dd HH:MM:SS
	26 : yy/mm/dd HH:MM:SS AM/PM
	27 : yy-mm-dd HH:MM
	28 : yy-mm-dd HH:MM AM/PM
	29 : yy-mm-dd HH:MM:SS
	30 : yy-mm-dd HH:MM:SS AM/PM
	31 : yyyy/mm/dd HH:MM
	32 : yyyy/mm/dd HH:MM AM/PM
	33 : yyyy/mm/dd HH:MM:SS
	34 : yyyy/mm/dd HH:MM:SS AM/PM
	35 : yyyy-mm-dd HH:MM
	36 : yyyy-mm-dd HH:MM AM/PM 37 : yyyy-mm-dd HH:MM:SS
	38 : yyyy-mm-dd HH:MM:SS AM/PM
	30 . yyyy-Hilli-uu HH.IVIIVI.33 AIVI/FIVI
	I .

Item	Description
Туре	BSTR TimeConvert(DOUBLE dTimeValue, INT nTimeType)
Kind	System Function
	45 : yy/mm
	47 : yyyy/mm
	49 : yy
	50 : yyyy
	51 : mm
	52 : dd
	53 : HH
	54 : MM
	55 : SS
	65 : "SUN" ~ "SAT"
Example	Sub Main
	Dim szTime As String
	szTime = ole.TimeConvert (1, 33)
	MsgBox szTime
	End Sub

14.3.9 Recipe Function

Item	Description
Туре	LONG RecipeChangeData (BSTR szModelName, BSTR szRecipeName)
Kind	Recipe Function
Description	Change Data Set of the workingset (Model must be the workingset's model)
	szModelName: Recipe Model Name
	szRecipeName: The data name belonging to the recipe model
Example	Sub Main
	Dim Result As Long
	Result = ole.RecipeLoadSet("CoffeDispenser2",NUL,"CreamCoffee")
	If (Result <> 0) Then
	MsgBox("RecipeLoadSet : Error~~!!")
	Else
	Result =
	ole.RecipeChangeData("CoffeDispenser2","CoffeeA")
	If (Result <> 0) Then
	MsgBox("RecipeChangeData : Error~~!!")
	Else
	MsgBox("RecipeChangeData : OK~~!!")
	End If
	End If
	End Sub

Item	Description
Туре	LONG RecipeChangeUnit (BSTR szModelName, BSTR szUnitName)
Kind	Recipe Function
Description	Change Unit Set of the workingset (Model must be the workingset's model)
	szModelName: Recipe Model Name
	szUnitName: The unit name belonging to the recipe model
Example	Sub Main
	Dim Result As Long
	Result = ole.RecipeLoadSet("CoffeDispenser2",NUL,NUL)
	If (Result <> 0) Then
	MsgBox("RecipeLoadSet : Error~~!!")
	Else
	Result = ole.RecipeChangeUnit("CoffeDispenser2", _
	"OfficeDispenser")
	If (Result <> 0) Then
	MsgBox("RecipeChangeUnit : Error~~!!")
	Else
	MsgBox("RecipeChangeUnit : OK~~!!")
	End If
	End If
	End Sub

Item	Description
Туре	LONG RecipeControlSet (BSTR szModelName, BSTR szUnitName, BSTR szDataNamet)
	LONG RecipeControlSetEx (SHORT nMode, BSTR szClientProgName, ULONG hClientHandle, BSTR szModelName, BSTR szUnitName, BSTR szDataName, LONG*
	plTransNo, SHORT* pnResult)
Kind	Recipe Function
Description	Perform recipe control with the model, unit and data information. (This function has no connection with workingset) nMode: Control Mode (nMode: 1: no return, 2: Return)
	szClientProgName: The program name that want to recipe control.
	hClientHandle: The client handle that want to recipe control.
	szModelName : Recipe Model Name
	szUnitName : The unit name belonging to the recipe model
	szDataName : The data name belonging to the recipe model
	plTransNo: In the case of recipe control return mode, The transaction number when
	control was succefully completed
	pnResult: Result of function
Example	Sub Main
	Dim TransNo As Long
	Dim Result As Long
	Dim Result2 As Integer
	Result = ole.RecipeLoadSet("CoffeDispenser2",NUL,"CoffeeA")
	If (Result <> 0) Then
	MsgBox("RecipeLoadSet : Error~~!!")
	Else
	Result2 = ole.RecipeControlSetEx(2,"Script",this, "CoffeDispenser2","OfficeDispenser","CoffeeA",TransN
	o) Result2 = ole.RecipeControlSet("CoffeDispenser2", _ "OfficeDispenser","CoffeeA")
	If (Result2 <> 0) Then MsgBox("RecipeControlSet : Error~~!!")
	Else
	MsgBox("RecipeControlSet : OK~~!! , TransNO = " _ + CStr(TransNo))
	End If
	End If
	End Sub

Item	Description
Туре	LONG RecipeControlWorkingSet ();
	LONG RecipeControlWorkingSetEx (SHORT nMode, BSTR szClientProgName,
	ULONG hClientHandle, LONG* plTransNo, SHORT* pnResult);
Kind	Recipe Function
Description	Control workingset data with unit name. (There is no control return)
	nMode: Control Mode (nMode: 1:No Return, 2:Return)
	szClientProgName: The program name that want to control
	hClientHandle: The client handle that want to control
	plTransNo: In the case of recipe control return mode, The transaction number when
	control was succefully performed
	pnResult: Result of function
Example	Sub Main
	Dim TransNo As Long
	Dim Result As Long
	Dim Result2 As Integer
	Result = ole.RecipeLoadSet("CoffeDispenser2",NUL,"CoffeeA")
	If (Result <> 0) Then
	MsgBox("RecipeLoadSet : Error~~!!")
	Else
	Result2 = ole.RecipeControlWorkingSetEx(2,"Script",this, TransNo)
	' Result2 = ole.RecipeControlWorkingSet()
	If (Result2 <> 0) Then
	MsgBox("RecipeControlWorkingSet : Error~~!!")
	Else
	MsgBox("RecipeControlWorkingSet: OK~~!!, TransNO = "
	+ CStr(TransNo))
	End If
	End If
	End Sub

Item	Description
Type	LONG RecipeDeleteData (BSTR szModelName, BSTR szRecipeName)
Kind	Recipe Function
Description	Delete Recipe Data
	szModelName : Recipe Model Name
	szRecipeName : The data name belonging to the recipe model
Example	Sub Main
	Dim Result As Long
	Result = ole.RecipeDeleteData("CoffeDispenser2","Amerian")
	If (Result <> 0) Then
	MsgBox("RecipeDeleteData : Error~~!!")
	Else
	MsgBox("RecipeDeleteData : OK~~!!")
	End If
	End Sub

Item	Description
Туре	LONG RecipeDeleteModel (BSTR szModelName)
Kind	Recipe Function
Description	Delete Recipe Model (The recipe unit and data will be deleted belonging to recipe model)
	szModelName: Recipe Model Name
Example	Sub Main
	Dim Result As Long
	Result = ole.RecipeDeleteModel("CoffeDispenser3")
	If (Result <> 0) Then
	MsgBox("RecipeDeleteModel : Error~~!!")
	Else
	MsgBox("RecipeDeleteModel : OK~~!!")
	End If
	End Sub

Item	Description
Туре	LONG RecipeDeleteUnit (BSTR szModelName, BSTR szUnitName)
Kind	Recipe Function
Description	Delete Recipe Unit
	szModelName: Recipe Model Name
	szUnitName: The unit name belonging to the recipe model
Example	Sub Main
	Dim Result As Long
	Result = ole.RecipeDeleteUnit("CoffeDispenser2","SubOfficeDispenser")
	If (Result <> 0) Then
	MsgBox("RecipeDeleteUnit : Error~~!!")
	Else
	MsgBox("RecipeDeleteUnit : OK~~!!")
	End If
	End Sub

Item	Description
Туре	LONG RecipeLoadDynamic (BSTR szModelName, BSTR szUnitName)
Kind	Recipe Function
Description	Get real-time data of the unit and put recipe model and data on workingset
	szModelName: Recipe Model Name
	szUnitName: The unit name belonging to the recipe model
Example	Sub Main
	Dim Result As Long
	Result = ole.RecipeLoadDynamic("CoffeDispenser2","MainOfficeDispenser
	")
	If (Result <> 0) Then
	MsgBox("RecipeLoadDynamic : Error~~!!")
	Else
	MsgBox("RecipeLoadDynamic : OK~~!!")
	End If
	End Sub

Item	Description
Туре	LONG RecipeLoadSet (BSTR szModelName, BSTR szUnitName, BSTR
	szRecipeName)
Kind	Recipe Function
Description	Put recipe model, unit and data on workingset
	szModelName: Recipe Model Name
	szUnitName: The unit name belonging to the recipe model
	szRecipeName : The data name belonging to the recipe model
Example	Sub Main
	Dim Result As Long
	Result = ole.RecipeLoadSet("CoffeDispenser2",NUL,"CreamCoffee")
	If (Result <> 0) Then
	MsgBox("RecipeLoadSet : Error~~!!")
	Else
	MsgBox("RecipeLoadSet : OK~~!!")
	End If
	End Sub

Item	Description
Туре	LONG RecipeSaveAsArchive (BSTR szModelName, BSTR szRecipeName)
Kind	Recipe Function
Description	Save workingset data as recipe name of the model name. (Model must be the
	workingset's model)
	szModelName: Recipe Model Name
	szRecipeName: The data name belonging to the recipe model
Example	Sub Main
	Dim Result As Long
	Result = ole.RecipeLoadSet("CoffeDispenser2",NUL,"BlackCoffee")
	If (Result <> 0) Then
	MsgBox("RecipeLoadSet : Error~~!!")
	Else
	Result = ole.RecipeSaveAsArchive("CoffeDispenser2", _
	"BlackCoffee")
	If (Result <> 0) Then
	MsgBox("RecipeSaveAsArchive : Error~~!!")
	Else
	MsgBox("RecipeSaveAsArchive : OK~~!!")
	End If
	End If
	End Sub

Item	Description
Туре	LONG RecipeSetData (BSTR szModelName, BSTR szItemName, VARIANT szSetValue)
Kind	Recipe Function
Description	Change position value of the item in the workingset (Model must be the workingset's model) szModelName: Recipe Model Name szItemName: Item in the workingset
	szSetValue: The change value of the item position in the workingset
Example	Sub Main Dim vtSetValue As Variant Dim Result As Long Result = ole.RecipeLoadSet("CoffeDispenser2",NUL,NUL) vtSetValue = 55 If (Result <> 0) Then MsgBox("RecipeLoadSet : Error~~!!") Else Result = ole.RecipeSetData("CoffeDispenser2", "Coffee",vtSetValue)
	If (Result <> 0) Then MsgBox("RecipeSetData : Error~~!!") Else MsgBox("RecipeSetData : OK~~!!") End If End If End Sub

14.3.10 Database Function

Item	Description
Type	VOID DataExportRunWork (BSTR szWorkName)
Kind	Database Function
Description	Run data export operation
	szWorkName : Operation Name
Example	Sub Main
	ole. DataExporRuntWork ("OperationName")
	End Sub

14.3.11 Alarm Function

Item	Description
Туре	BOOL AckAllAlarm()
Kind	Alarm Function
Description	The function ACKs all alarm with level of connected users.
Example	Sub Main
	ole. AckAllAlarm
	End Sub

Item	Description
Туре	BSTR GetAlarmCountErrInfo (LONG ICount)
Kind	Alarm Function
Description	Give an explanation of the error number when you request the alarm count.
	ICount: Function GetAlmCount(),The Return Value of GetHistAlmCount() < 0
Example	Sub Main
	Dim szErrDesc As String
	Dim AlmCount As Long
	AlmCount = ole. GetHistAlmCount (1, 0)
	If AlmCount < 0 Then
	szErrDesc = ole.GetAlarmCountErrInfo (AlmCount)
	MsgBox szErrDesc
	End If
	End Sub

Item	Description
Type	LONG GetAlarmLogging (DOUBLE dtFrom, DOUBLE dtTo, BSTR szTag, BSTR szAlarmGr, LONG IAlarmType, LONG *IOutCount, SAFEARRAY **pszRstrTag, SAFEARRAY **pszRstrGr, SAFEARRAY **pdbRdtOccAlarm, SAFEARRAY **pdbRdtRevAlarm, SAFEARRAY **pdbRdtAckAlarm, SAFEARRAY **plRnAlarmType, SAFEARRAY **pszRstrMsg, SAFEARRAY **pszRstrAlarmKind) Alarm Function
Description	This function retrieves history alarm list according to the condition dtFrom: Beginning Time (from the first data, if 0) dtTo: End Time (to the last data, if 0) szTag: Tag Name (NULL: ALL) szAlarmGr. Alarm Group (NULL: ALL) [AlarmType: Alarm Type (0: ALL, 1: Alarm(Occurrence), 2: Event, 3: Alarm(Recovery), 4: Alarm (Occurrence+Recovery)) [OutCount: Alarm List Count pszRstrTag: Tag Name List pszRstrGr. Alarm Group List pdbRdtOccAlarm: List of Alarm Occurrence Times pdbRdtRevAlarm: List of Alarm Occurrence Times pdbRdtAckAlarm: List of Alarm Ack Times plRnAlarmType: List of Alarm Types pszRstrMsg: List of Alarm Messages pszRstrAlarmKind: Alarm Type (HH,HI,LO,LL,Achange,MGap,SGap,On,Off,OnOff,OffOn,DChagne)

Item	Description (Continued)
Example	Sub Main
	Dim dtFrom As Date
	Dim dtTo As Date
	Dim szTag As String
	Dim szAlarmGr As String
	Dim IAlarmType As Long
	Dim lOutCount As Long
	Dim aRstrTag() As String
	Dim aRstrGr() As String
	Dim aRdtOccAlarm() As Double
	Dim aRdtRevAlarm() As Double
	Dim aRdtAckAlarm() As Double
	Dim aRnAlarmType() As Long
	Dim aRstrMsg() As String
	Dim aRstrAlarmKind() As String
	dtFrom = Now() - 27
	dtTo =Now()
	szTag =""
	szAlarmGr = "AlmGrp_UT"
	IAlarmType = 0
	<pre>IRet = ole.GetAlarmLogging(dtFrom ,dtTo , szTag ,</pre>
	szAlarmGr,lAlarmType,lOutCount, aRstrTag, aRstrGr,
	aRdtOccAlarm, aRdtRevAlarm,aRdtAckAlarm,
	aRnAlarmType,
	aRstrMsg,aRstrAlarmKind)
	For i = 0 To IOutCount - 1
	MsgBox (aRstrTag(i) + " " + aRstrGr(i) + " "+
	CStr(CDate(aRdtOccAlarm(i))) + " "+
	CStr(CDate(aRdtRevAlarm(i))) + " "+
	CStr(CDate(aRdtAckAlarm(i))) + " " +
	CStr(aRnAlarmType(i)) + " " +
	aRstrMsg(i) + " " + aRstrAlarmKind(i))
	Next i
	End Sub

lée se	Decovintion
Item	Description
Туре	LONG GetAlarmLoggingEx (DOUBLE dtFrom,DOUBLE dtTo,BSTR szTag,
	LONG nAlarmGroupCount, SAFEARRAY **pszAlarmGr,LONG lAlarmType,
	LONG IAlarmGrade, BSTR szAlarmKind, LONG *IOutCount,
	SAFEARRAY**pRstrTag,SAFEARRAY**pRstrGr, SAFEARRAY
	**pdbRdtOccAlarm, SAFEARRAY **pdbRdtRevAlarm, SAFEARRAY
	**pdbRdtAckAlarm, SAFEARRAY **plRnAlarmType, SAFEARRAY
	**pszRstrMsg, SAFEARRAY **pszRstrAlarmKind, SAFEARRAY
	pdbRvOccDValue, SAFEARRAYpszRvOccSValue, SAFEARRAY
16: 1	**plAlarmGrade)
Kind	Alarm Function
Description	This function retrieves history alarm list according to the condition, extension
	function of GetAlarmLogging()
	dtFrom: Beginning Time (from the first data, if 0)
	dtTo: End Time (to the last data, if 0)
	szTag: Tag Name (NULL: ALL)
	nAlarmGroupCount: Alarm Group Count
	pszAlarmGr. Alarm Group List (NULL: ALL)
	IAlarmType: Alarm Type (Event has only Occurrence.)
	(0: ALL,
	1: Alarm(Occurrence),
	2: Event,
	3: Alarm(Recovery),
	4: Alarm (Occurrence+Recovery))
	IAlarmGrade: Alarm Level.
	You can get the alarm lists that have higher level than this level.
	Level1 is higher than Level10.
	ex) If input is level5, returns level1 ~ level5.
	szAlarmKind:: Alarm Type
	(NULL: ALL, HH,HI,LO,LL,Achange,MGap,SGap,
	On,Off,OnOff,OffOn,DChagne)
	IOutCount: Alarm List Count
	pszRstrTag: List of Tag Names
	pszRstrGr. List of Alarm Groups
	pdbRdtOccAlarm: List of Alarm Occurrence Times
	pdbRdtRevAlarm: List of Alarm Recovery Times
	pdbRdtAckAlarm: List of Alarm Ack Times

Item	Description
Туре	LONG GetAlmCount (LONG IKind, LONG IStatus)
Kind	Alarm Function
Description	Inform the real-time alarm count.
	IKind: Alarm Kind
	0 : All
	1 : Process Alarm
	2 : System Alarm
	3 : Process + System
	4 : Event
	5 : Process + Event
	6 : System + Event
	IStatus: Alarm Status
	0 : All
	1 : Unrecovered
	2 : Unrecognized
	3 : Unrecovered + Unrecognized
Example	Sub Main
	Dim AlmCount As Long
	AlmCount = ole. GetAlmCount(1, 0)
	If AlmCount > 0 Then
	MsgBox " Real-time Alarm Count = " + CStr(AlmCount)
	End If
	End Sub

Item	Description
Туре	LONG GetGroupUserInfoByTag (BSTR szTagName, LONG* IOutCount,
	SAFEARRAY(BSTR) *pszUserID, SAFEARRAY(BSTR) *pszUserName,
	SAFEARRAY(LONG) *plMailCheck, SAFEARRAY(LONG) *plSMSMCheck,
12: 1	SAFEARRAY(BSTR) *pszMailAdd, SAFEARRAY(BSTR) *pszHPhoneNo)
Kind	Alarm Function
Description	Get registered user information of the alarm group that contains specified tag by tag name.
	It is used to call the relevant user with tag information (alarm information)
	szTagName: Tag Name
	IOutCount: Count of the returned user information
	pszUserID: User ID pszUserName: User Name
	plMailCheck:: Check whether to receive a mail notice in case of the occurrence
	of alarm
	plSMSMCheck: Check whether to receive a SMS notice in case of the
	occurrence of alarm
	pszMailAdd: Mail address registered by the user
	pszHPhoneNo: Mobile phone number registered by the user
Example	Sub Main
	Dim lOutCount As Long
	Dim aUserID() As String
	Dim aUserName() As String
	Dim aMailCheck() As Long
	Dim aSMSCheck() As Long
	Dim aMailAdd() As String
	Dim aHPhoneNo() As String
	ole.GetGroupUserInfoByTag("AlmGrpTestTag", lOutCount, aUserID,
	aUserName, aMailCheck, aSMSCheck, aMailAdd, aHPhoneNo)
	For i = 0 To IOutCount
	MsgBox (aUserID(i) + " " + aUserName(i) + " "
	+ CStr(aMailCheck(i)) + " " + CStr(aSMSCheck(i))
	+ " " + aMailAdd(i) + " " + aHPhoneNo(i))
	Next i
	End Sub

Item	Description (Continued)
Description	plRnAlarmType: List of Alarm Types
	pszRstrMsg: List of Alarm Messages
	pszRstrAlarmKind: Alarm Type
	(HH,HI,LO,LL,Achange,MGap,SGap,On,Off,OnOff,OffOn,DChag
	ne)
	pdbRvOccDValue List of Current Values (Only if Alarm)
	pszRvOccSValue: List of Current Values (only if Event)
	plAlarmGrade: List of Alarm Levels
Example	Sub Main
	Dim dtFrom As Double
	Dim dtTo As Double
	Dim szTag As String
	Dim nAlarmCount As Long
	Dim aAlarmGr(0) As String
	Dim IAlarmType As Long
	Dim IAlarmGrade As Long
	Dim szAlarmKind As String
1	Dim lOutCount As Long
	Dim aRstrTag() As String
	Dim aRstrGr() As String
	Dim aRdtOccAlarm() As Double
	Dim aRdtRevAlarm() As Double
	Dim aRdtAckAlarm() As Double
	Dim aRnAlarmType() As Long
	Dim aRstrMsg() As String
	Dim aRstrAlarmKind() As String
	Dim aOccDvalue() As Double
	Dim aOccSValue() As String
	Dim aAlarmLevel() As Long
	dtFrom = Now() - 27
	dtTo = Now
	szTag =""
	nAlarmCount = 1
	aAlarmGr(0) = "AlmGrp_UT"
	IAlarmType = 4
	IAlarmGrade = 10
	szAlarmKind = "LO"

Item	Description (Continued)
Example	IRet = ole.GetAlarmLoggingEx (dtFrom ,dtTo , szTag , nAlarmCount, aAlarmGr,lAlarmType,lAlarmGrade, szAlarmKind, lOutCount, aRstrTag, aRstrGr, aRdtOccAlarm, aRdtRevAlarm,aRdtAckAlarm, aRnAlarmType, aRstrMsg,aRstrAlarmKind, aOccDvalue, aOccSValue, aAlarmLevel)
	For i = 0 To IOutCount MsgBox (aRstrTag(i) + " " + aRstrGr(i) + " " + CStr(CDate(aRdtOccAlarm(i)))+ " " + CStr(CDate(aRdtRevAlarm(i)))+ " " + CStr(CDate(aRdtAckAlarm(i))) + " " + CStr(aRnAlarmType(i)) + " " + aRstrMsg(i) + " " + aRstrAlarmKind(i)) Next i End Sub

Item	Description
Туре	LONG GetHistAlmCount (LONG IKind, LONG IStatus)
Kind	Alarm Function
Description	Inform the history alarm count
	IKind: Alarm Kind
	0 : All,
	1 : Process Alarm,
	2 : System Alarm,
	3 : Process + System,
	4 : Event,
	5 : Process + Event,
	6 : System + Event
	IStatus: Alarm Status
	0 : All,
	1 : UnRecover,
	2 : UnAck
Example	Sub Main
	Dim AlmCount As Long
	AlmCount = ole. GetHistAlmCount (1, 0)
	If AlmCount > 0 Then
	MsgBox "History Alarm Count = " + CStr(AlmCount)
	End If
	End Sub

Chapter 14 Function

Item	Description
Туре	LONG GetUserInfoList(LONG* IOutCount, SAFEARRAY(BSTR) *pszUserID,
	SAFEARRAY(BSTR) *pszUserName, SAFEARRAY(BSTR) *pszDesc,
	SAFEARRAY(BSTR) *pszPassword, SAFEARRAY(LONG) *plUserLevel,
	SAFEARRAY(BSTR) *pszHPhoneNo, SAFEARRAY(BSTR) *pszPhoneNo,
	SAFEARRAY(BSTR) *pszMailAdd)
Kind	Alarm Function
Description	The function returns all user's list and informations.
	IOutCount: Returned list count
	pszUserID : User ID
	pszUserName: User name
	pszDesc : User's description
	pszPassword : User's password
	plUserLevel : User's level
	pszHPhoneNo: User's cell phone number
	pszPhoneNo: User's phone number
	pszMailAdd: User's e-mail address
Example	Sub Main
	Di ID (A. I
	Dim IRet As Long
	Dim IOutCount As Long
	Dim aUserID() As String
	Dim aUserName() As String
	Dim aDesc() As String
	Dim aPassword() As String Dim aUserLevel() As Long
	Dim adsertever() As Eding Dim aMailAdd() As String
	Dim all all all all all all all all all al
	Dim aPhoneNo() As String Dim aPhoneNo() As String
	Dilli di Horierio() As otting
	IRet = ole.GetUserInfoList(IOutCount, aUserID, aUserName, aDesc,
	aPassword, aUserLevel, aHPhoneNo, aPhoneNo, aMailAdd)
	ar accircia, accorector, arm richerto, ar noncreo, armain laay
	For i = 0 To IOutCount - 1
	MsgBox (aUserID(i) + " " + aUserName(i) + " "+
	CStr(aUserLevel(i)) + " " + aHPhoneNo(i) + " "
	+ aPhoneNo(i) + " " + aMailAdd(i))
	Next i
	End Sub

Item	Description
Туре	LONG SetUserAlarm (BSTR szAlarmMessage, BSTR szAlarmGroup)
Kind	Alarm Function
Description	This is the alarm function that user occurs. The alarm occurrence and recovery
	are generated at the same time
	szAlarmMessage: Alarm Message
	szAlarmGroup : Alarm Group
	Automatic entry user: Current logon user Occured Time, Recovery Time: current system time Alarm type: Process Alarm AlarmLevel: Input 10 level. Input "\$Script" in tag name
Example	Sub Main ole.SetUserAlarm "UserAlarm ", "AlarmGroup"
	End Sub

Item	Description
Туре	LONG SetUserAlarmEx (DOUBLE dOccur, LONG IAlarmLevel, BSTR
	szAlarmMessage ,BSTR szAlarmGroup, DOUBLE dCur, DOUBLE dPrev)
Kind	Alarm Function
Description	This is the extended alarm function that user occurs. The alarm occurrence
	and recovery are generated at the same time
	dOccur. Time Input - If you set this value by 0, current time value will be applied
	IAlarmLevel: Alarm Level - If you set this value from 0 to 10, then setting value
	will be applied. Else 10 will be applied automatically
	szAlarmMessage : Alarm Message
	szAlarmGroup : Alarm Group
	dCur : Current Value
	dPrev : Previous
	Automatic entry
	user: Current logon user
	Alarm type: Process Alarm
	Input "\$Script" in tag name
Example	Sub Main
	ole.SetUserAlarmEx Now(), 2, " UserAlarm ", "AlarmGroup", _
	22.34, 44.56
	End Sub

Chapter 14 Function

Item	Description
Туре	LONG SetUserEvent (BSTR szEventMessage)
Kind	Alarm Function
Description	This is the event function that user occurs.
	szEventMessage : Event Message
	자동입력 Item Event time: Current system tme Event occur: Current logon user Occurred Computer: This Computer Name Alarm type: Event Alarm
Example	Sub Main
	ole.SetUserEvent "UserEvent"
	End Sub

Item	Description
Туре	VOID StopAlarmSound ()
Kind	Alarm Function
Description	The function stops playing sound of the alarm.
Example	Sub Main
	ole. StopAlarmSound
	End Sub

14.3.12 External Device Function

Item	Description
Туре	VOID PlaySound (BSTR szWaveFile)
Kind	External Device Function
Description	Play sound (Async)
	szSoundFile: wave file path
Example	Sub Main
	ole.PlaySound ("wave file .wav")
	End Sub

Item	Description
Туре	VOID PlaySoundFile (BSTR szSoundFile)
Kind	External Device Function
Description	Play sound (Sync)
	szSoundFile : wave file path
Example	Sub Main
	ole.PlaySoundFile("SoundFile.wav")
	End Sub

14.3.13 User Function

Item	Description
Туре	BSTR GetUserID ()
Kind	User Function
Description	The function returns ID of the user
Example	Sub Main
	Dim szUserID As String
	szUserID = ole. GetUserID
	MsgBox " User ID = "szUserID
	End Sub

Item	Description
Туре	LONG GetUserLevel ()
Kind	User Function
Description	The function returns Level of the user
Example	Sub Main
	Dim IRet As Long
	IRet = ole. GetUserLevel
	MsgBox " User Priority = "+ CScr(IRet)
	End Sub

Item	Description
Туре	LONG UserLogOff ()
Kind	User Function
Description	Notifies InfoU Runtime that a user has logged off
Example	Sub Main
	Dim IRet As Long
	IRet = ole.UserLogOff
	If IRet <> 0 Then
	MsgBox "UserLogOff() False"
	Else
	MsgBox "UserLogOff() OK"
	End If
	End Sub

Item	Description
Туре	LONG UserLogOn (BSTR szUserID, BSTR szPassWord)
Kind	User Function
Description	Notifies InfoU Runtime that a user has logged on
	szUserID: User ID
	szPassWord: User Password
Example	Sub Main
	Dim IRet As Long
	<pre>IRet = ole.UserLogOn("UserID", " UserPassword ")</pre>
	If IRet <> 0 Then
	MsgBox "UserLogOn() False"
	Else
	MsgBox "UserLogOn() OK"
	End If
	End Sub

14.3.14 Report Function

Item	Description
Туре	VOID ReportDialog ()
Kind	Report Function
Description	Shows the report windows
Example	Sub Main
	ole. ReportDialog
	End Sub

Item	Description
Туре	VOID ReportListRefresh()
Kind	Report Function
Description	Refresh a report list
Example	Sub Main
	ole. ReportListRefresh
	End Sub

Item	Description
Туре	VOID ReportPrint (BSTR szReportName)
Kind	Report Function
Description	Outputs the report
	szReportName : report name
Example	Sub Main
	ole. ReportPrint ("ReportName")
	End Sub

Item	Description
Туре	VOID ReportPrintEx (BSTR szReportName, BSTR sz ReqTime)
Kind	Report Function
Description	Outputs the report
	szReportName : report name
	szReqTime : request report start time
Example	Sub Main
	ole. ReportPrintEx ("ReportName", "2007/02/01")
	End Sub

Item	Description
Туре	VOID ReportSchedulerHide ()
Kind	Report Function
Description	Hide the report schedule windows
Example	Sub Main
	ole.ReportSchedulerHide
	End Sub

Item	Description
Туре	VOID ReportSchedulerShow ()
Kind	Report Function
Description	Shows report schedule list
Example	Sub Main
	ole. ReportSchedulerShow
	End Sub

14.3.15 History Data Function

Item	Description
Туре	VOID GetHistDataCount (BSTR szModelName, BSTR , BSTR szDataKind,
	DOUBLE dRefTime, LONG ICount, LONG IDir, LONG* IResultCount,
	SAFEARRAY(double) *psadblTimeStamp, SAFEARRAY(double)
	*psadblNumValue, SAFEARRAY(LONG) *psalStatus, SAFEARRAY(LONG)
	*psalResult)
Kind	History Data Function
Description	This function returns logging data from the logging group you specified
	according to the base time, direction and logging data count. (TimeStamp, Tag
	Value, Tag Status, Result)
	szModelName : Logging Group Name
	szTagName : Logging Tag Name
	szDataKind: Value Type (Default : Current Value)
	dRefTime: Base Time
	ICount, : Logging Data Count
	IDir: Direction (forward or backward from base time)
	IResultCount: Logging Data Count
	psadblTimeStamp: Array of TimeStamps
	psadblNumValue: Array of Tag Values
	psalStatus: Array of Tag Status
	psalResult: Result

Chapter 14 Function

Item	Description (Continued)
Example	Sub Main
	Dim ModelName As String
	Dim TagName As String
	Dim DataKind As String
	Dim RefTime As Date
	Dim ICount As Long
	Dim IDir As Long
	Dim ResultCount As Long
	Dim TimeStamp() As Double
	Dim NumValue() As Double
	Dim Status() As Long
	Dim Result() As Long
	ModelName = "INV1_SEC"
	TagName = "INV_1.1_AI_CT"
	DataKind = "Current"
	RefTime = Now()
	ICount = 5
	IDir = 1
	ole.GetHistDataCount(ModelName, TagName, DataKind, RefTime, ICount, IDir, ResultCount,
	TimeStamp, NumValue, Status, Result)
	For count = 0 To ResultCount -1
	MsgBox("NumValue = " + CStr(NumValue(count))
	+ " Status = " + CStr(Status (count)))
	Next
	End Sub

Item	Description
Туре	VOID GetHistDataPeriod (BSTR szModelName, BSTR szTagName, BSTR
	szDataKind, DOUBLE dFrom, DOUBLE dTo, LONG* IResultCount,
	SAFEARRAY(double) *psadblTimeStamp, SAFEARRAY(double)
	*psadblNumValue, SAFEARRAY(LONG) *psalStatus,
	SAFEARRAY(LONG) *psalResul)
Kind	History Data Function
Description	This function returns logging data from logging group you specified within a
	given period of time. (TimeStamp, Tag Value, Tag Status, Result)
	szModelName : Logging Group Name
	szTagName : Logging Tag Name
	szDataKind: Value Type (Default : Current Value)
	dFrom: Beginning Time
	dTo: End Time
	IResultCount: Logging Data Count
	psadblTimeStamp: Array of TimeStamps
	psadblNumValue: Array of Tag Values
	psalStatus: Array of Tag Status
	psalResult: Result
Example	Sub Main
	Dim ModelName As String
	Dim TagName As String
	Dim DataKind As String
	Dim dFrom As Date
	Dim dTo As Date
	Dim ResultCount As Long
	Dim TimeStamp() As Double
	Dim NumValue() As Double
	Dim Status() As Long
	Dim Result() As Long
	MadalNarra IIINIVA OFOII
	ModelName = "INV1_SEC"
	TagName = "INV_1.1_AI_CT"
	DataKind = "Current Value"
	dFrom = Now() -1
	'To = Now()
	ole.GetHistDataPeriod(ModelName, TagName, DataKind,
	dFrom, dTo, ResultCount, TimeStamp, NumValue, Status, Result)
	For count = 0 To ResultCount -1
	MsgBox("NumValue = " + CStr(NumValue(count))
	+ " Status = " + CStr(Numvalue(count))
	Next
	End Sub

Chapter 14 Function

Item	Description
Туре	LONG GetLoggingModelList (LONG *IOutCount ,SAFEARRAY(BSTR)
	*psaszModelName, SAFEARRAY(BSTR) *psaszModelDesc,
	SAFEARRAY(BSTR) *psaszModelType, SAFEARRAY(BSTR)
	*psaszArchvingCycle, SAFEARRAY(BSTR) *psaszSegmentPeriod)
Kind	History Data Function
Description	This function returns Logging Group Lists
	IOutCount: Logging Group count
	psaszModelName: List of Logging Group Names
	psaszModelDesc: List of Logging Group Descriptions
	psaszModelType: List of Logging Group Types
	psaszArchvingCycle: List of Acquisition Cycles
	psaszSegmentPeriod: List of Storage Periods
Example	Sub Main
	Dim IRet As Long
	Dim lOutCount As Long
	Dim aModelName() As String
	Dim aModelDesc() As String
	Dim aModelType() As String
	Dim aArchivingCycle() As String
	Dim aSegmentPeriod() As String
	IRet = ole.GetLoggingModelList(lOutCount, aModelName, _
	aModelDesc, aModelType,aArchivingCycle,
	aSegmentPeriod)
	For i = 0 To IOutCount -1
	MsgBox (aModelName(i) + " " + aModelDesc(i) + " "+ _
	aModelType(i) + " " + aArchivingCycle(i) + _
	" " + aSegmentPeriod(i))
	Next i
	End Sub

Item	Description
Туре	LONG GetLoggingTag(BSTR szModelName, LONG
	*IOutCount ,SAFEARRAY(BSTR) *psaszTagName, SAFEARRAY(BSTR)
	*psaszTagType)
Kind	History Data Function
Description	This function returns Logging Tags according to the Logging Group.
	szModelName: Logging Group Name
	IOutCount: Logging Tag Count in the Logging Group
	psaszTagName: List of Tag Names
	psaszTagType: List of Tag Types
Example	Sub Main
	Dim IRet As Long
	Dim lOutCount As Long
	Dim aTagName() As String
	Dim aTagType() As String
	<pre>IRet = ole.GetLoggingTag ("INV1_MON", IOutCount, aTagName,</pre>
	For i = 0 To IOutCount -1 MsgBox ("INV1_MON " + aTagName(i) + " " + aTagType(i))
	Next i
	End Sub

Item	Description
Туре	VOID SetHistData (BSTR szModelName, BSTR szTagName, BSTR
	szDataKind, LONG IDataCount, SAFEARRAY(double) *psadblSaveTime,
	SAFEARRAY(double) *psadblNumValue, SAFEARRAY(LONG)
	*psalTagStatus, SAFEARRAY(LONG) *psalResult)
Kind	History Data Function
Description	This function saves tag values as logging data in the logging group user
	specified.
	szModelName : Logging Group Name
	szTagName : Logging Tag Name
	szDataKind: Value Type (Default : Current Value
	IDataCount : Data Count
	psadblSaveTime: TimeStamp
	psadblNumValue : Tag Value
	psalTagStatus : Tag Status
	psalResult: Result
Example	Sub Main
	Dim szModelName As String
	Dim szTagName As String
	Dim szDataKind As String
	Dim ICount As Long
	Dim dtSaveTime(3) As Double
	Dim dblTagNumValue(3) As Double
	Dim ulTagStatus(3) As Long
	Dim IResult() As Long Dim IRet As Long
	szModelName = "INV1 SEC"
	szTagName = "INV 1.1 AI CT"
	szDataKind = "Current Value"
	ICount = 3
	For count = 0 To ICount -1
	dtSaveTime(count) = Now() + count*0.1
	dblTagNumValue(count) = 100 + count + 1
	ulTagStatus(count) = 0
	Next
	IRet = ole.SetHistData(szModelName ,szTagName , szDataKind,
	ICount , dtSaveTime, dblTagNumValue, ulTagStatus,
	IResult)
	If IRet <> 0 Then
	MsgBox("SetHistData Error!!!1")
	Else
	For count = 0 To ICount -1
	MsgBox("Result = " + CStr(IResult (count)))
	Next
	End If
	End Sub

14.3.16 I/O DRIVE FUNCTION

Item	Description
Туре	VOID ClearStaStatusCount (BSTR szChannelName, BSTR szStaName)
Kind	I/O DRIVE FUNCTION
Description	Reset Station Status counts.
	szChannelName : Channel Name
	szStaName : Station Name
Example	Sub Main
	ole. ClearStaStatusCount ("ChannelName", "StationName")
	End Sub

Item	Description
Туре	VOID EnableStation (BSTR szChannelName, BSTR szStaName, LONG IMode)
Kind	I/O DRIVE FUNCTION
Description	Set a station enable or disable.
	szChannelName : Channel Name
	szStaName : Station Name
	IMode: 0: Enable, 1: Disable
Example	Sub Main
	Dim SuccessCount As Long
	SuccessCount = ole. EnableStation ("ChannelName", StationName",0)
	End Sub

Item	Description
Туре	LONG GetCountReadSuccess (BSTR szChannelName, BSTR szStaName)
Kind	I/O DRIVE FUNCTION
Description	Return Read OK counts
	szChannelName : Channel Name
	szStaName : Station Name
Example	Sub Main
	Dim SuccessCount As Long
	SuccessCount = ole. GetCountReadSuccess
	("ChannelName", "StationName")
	MsgBox "Read Success Count = " + CStr(SuccessCount)
	End Sub

Chapter 14 Function

Item	Description
Туре	LONG GetCountReadTotal (BSTR szChannelName, BSTR szStaName)
Kind	I/O DRIVE FUNCTION
Description	Return Read Total counts
	szChannelName: Channel Name
	szStaName: Station Name
Example	Sub Main
	Dim ReadCount As Long
	ReadCount = ole. GetCountReadTotal ("ChannelName", "StationName")
	MsgBox "Read Count = " + CStr(ReadCount)
	End Sub

Item	Description
Туре	LONG GetCountWriteSuccess (BSTR szChannelName, BSTR szStaName)
Kind	I/O DRIVE FUNCTION
Description	Return Write OK counts.
	szChannelName: Channel Name
	szStaName: Station Name
Example	Sub Main
	Dim SuccessCount As Long
	SuccessCount = ole. GetCountWriteSuccess
	("ChannelName", "StationName")
	MsgBox "Write Success Count = " + CStr(SuccessCount)
	End Sub

Item	Description
Туре	LONG GetCountWriteTotal (BSTR szChannelName, BSTR szStaName)
Kind	I/O DRIVE FUNCTION
Description	Return Write Total counts.
	szChannelName: Channel Name
	szStaName: Station Name
Example	Sub Main
	Dim WriteCount As Long
	WriteCount = ole. GetCountWriteTotal ("ChannelName", "StationName")
	MsgBox "Write Count = " + CStr(WriteCount)
	End Sub

Item	Description
Туре	LONG GetStatusRunMode (BSTR szChannelName, BSTR szStaName)
Kind	I/O DRIVE FUNCTION
Description	Return Station Runmode.
	szChannelName: Channel Name
	szStaName: Station Name
Example	Sub Main
	Dim RunModeCount As Long
	RunModeCount = ole. GetStatusRunMode ("ChannelName", "StationName")
	MsgBox "Station Run Mode Count = " + CStr(RunModeCount)
	End Sub

Item	Description
Туре	LONG GetStatusStaOK (BSTR szChannelName, BSTR szStaName)
Kind	I/O DRIVE FUNCTION
Description	Return Station OK status.
	szChannelName: Channel Name
	szStaName: Station Name
Example	Sub Main
	Dim OKCount As Long
	OKCount = ole. GetStatusStaOK ("ChannelName", "StationName")
	MsgBox "OK Count = " + CStr(OKCount)
	End Sub

Item	Description
Туре	VOID ShowNetworkStatus ()
Kind	I/O DRIVE FUNCTION
Description	Display Station Status dialog.
Example	Sub Main
	ole.ShowNetworkStatus
	End Sub

Item	Description
Туре	VOID ShowStationInfo ()
Kind	I/O DRIVE FUNCTION
Description	Display Station Status dialog.
Example	Sub Main
	ole.ShowStationInfo
	End Sub

Notice

Program to use Function

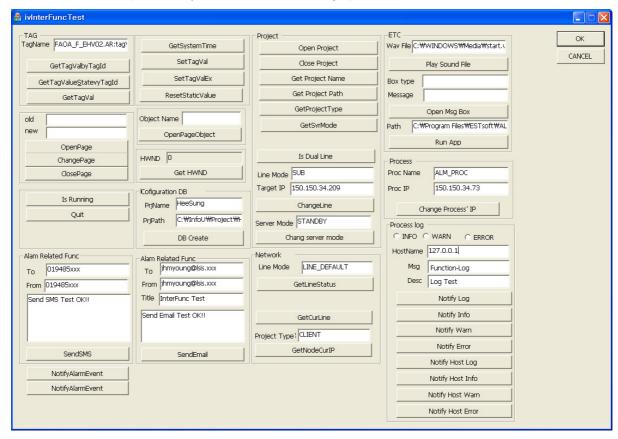
- (1) Office MACRO such as Script or Excel
- (2) Other Visual C++, Visual Basic Development Programs in which OLE Automation can be used

14.4 Application Example of OLE Automation

As an example to show how to use OLE Automation supported in the InfoU, an 'Object' can be created with VB(Visual Basic) or VC++(Visual C++) as seen in the following table to make a 'Form' to be used.

Program	How to Create an Object
VB	
	Public iuOle as Object
	Set iuOle = CreateObject("iuOleSvr.iuInterFuncOleSvr")
VC++	
	#import "iuOleSvr\Debug\iuOleSvr.tlb" no_namespace
	liuInterFuncOleSvrPtr m_pivOleSvr; m_pivOleSvr = liuInterFuncOleSvrPtr(uuidof(iuInterFuncOleSvr));

The created 'Object' can import or control tag values registered in the InfoU by using OLE Automation and it also use all of the functions provided by Function to ensure the graphic runtime is controlled.



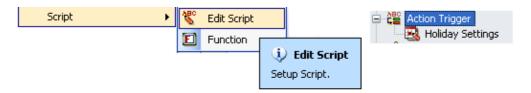
Chapter 15 **Action Trigger**

Action Trigger is set up during the runtime to execute input and output works of the script, recipe and database.

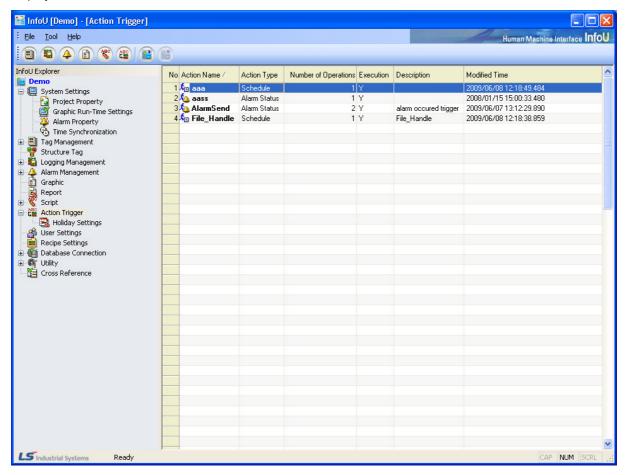
The relevant application can be executed by setting up Action Trigger according to 'Schedule', 'Tag Value Chang' or 'Alarm Status'.

One-stop settings for various application actions at one place ensure an effective and systematic control.

15.1 **Start**

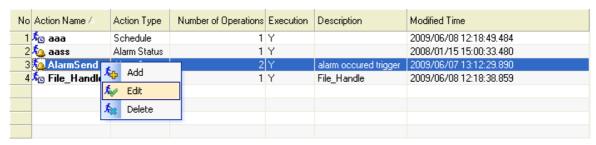


If 'Action Trigger' in the menu or 'Edit Script' in the InfoU explorer is selected, the list of action trigger is displayed as follows:



Chapter 15 Action Trigger

Click the right button of the mouse on the list of Action Trigger to show Add, Edit and Delete menu.



- (1) Add: A new action trigger is added. Double click on the empty item to display the new input screen.
- (2) Edit: The selected action trigger is edited. Double click an item to display the edition screen.
- (3) Delete: Delete the selected action trigger.

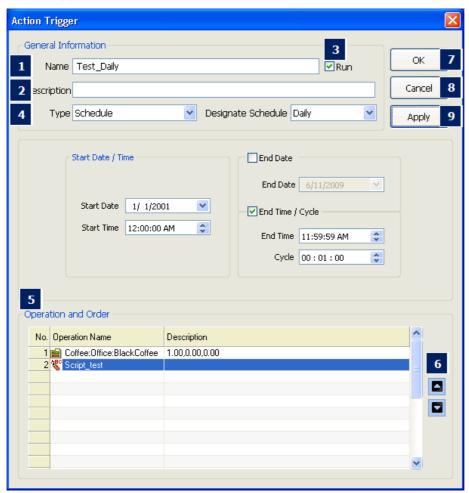
15.2 **Edition Screen**

15.2.1 **General Information**

The edition screen of action trigger is of the modaless dialog type and it is organized as follows:

If other item is selected from the above action trigger list while the screen is displayed, information on the selected action trigger is displayed.

Any modification, which may be created during the edition of action trigger, is not reflected during the runtime and it is applied after the runtime is restarted.



(1) Name

- The action trigger name is displayed.
- If the action trigger is renamed and saved, a new action trigger is created and added automatically.
- Any two action trigger s shall not have the same name.

(2) Description

Input description on the action trigger.

(3) Run

It displays whether to execute the script actions during the runtime.

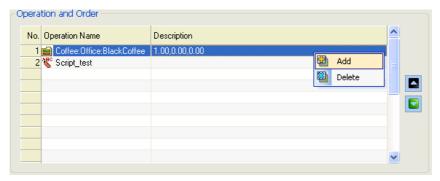
Chapter 15 Action Trigger

(4) Type

- The script actions are triggered by one of the following condition according to the designated settings:
 - (a) Schedule
 - (b) Tag Vale Change
 - (c) Alarm Status

(5) Operation and Order

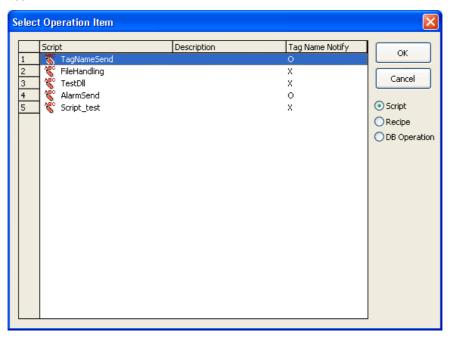
- Select an operation to execute among the defined actions and designate the order.
 - (a) Click the right side of the mouse on the list to show the menu below.



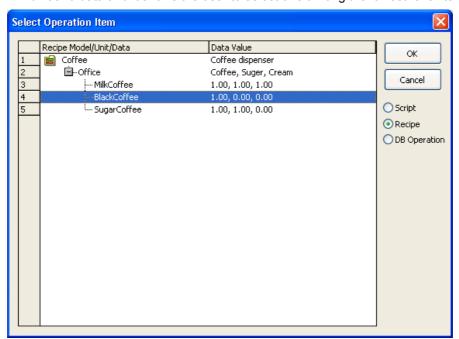
- (b) Add: Click an additional menu or double click to show a screen for selecting a new operation.
- (c) Delete: Delete the selected operation from the list.

(d) Select Operation Item

Select an operation item on the screen below. The screen below shows an example of the script



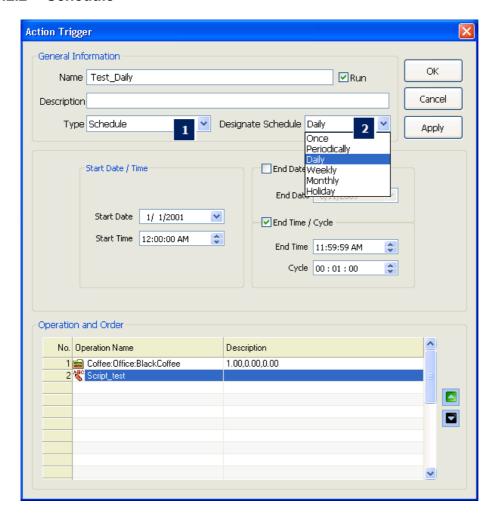
- If one of items on the right side is clicked, operation types are changed and a list is displayed.
- After selecting an item, click on Ok button or double click on the item to add it to the list.
- For scripts, Tag Name Delivery is displayed.
- The next screen is about selecting a recipe. A recipe is constructed in a tree type of model, unit and data and it allows the user to select one among the lowest-level items.



Chapter 15 Action Trigger

- (6) Up/Down
 - Select an operation and change the order by clicking arrows.
- (7) OK
 - The work is saved and the screen is closed.
- (8) Cancel
 - The work is stopped and the screen is closed.
- (9) Apply
 - The work is saved and the work is continued.

15.2.2 Schedule

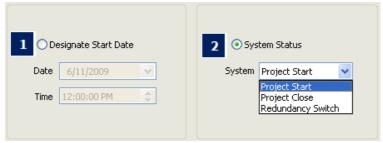


(1) General Information

- (a) If 'Schedule' is selected for 'Type',
- (b) 'Designate Schedule' becomes activated and here, one of [Once], [Periodically], [Daily], [Weekly], [Monthly] and [Holiday] can be selected.

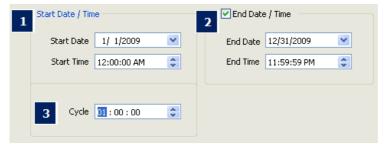
Chapter 15 Action Trigger

(2) Once



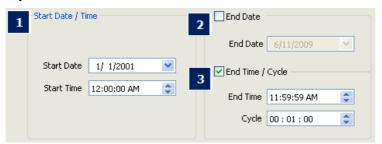
- (a) Designate Start Date: If 'Designate Start Date' is selected, it is allowed to input date and time to start the selected operation and it is executed only once at the designated time.
- (b) System Status: The operation is executed according to the system's management status over the project.
 - Project Start
 - Project Close
 - Redundancy Switch

(3) Periodically



- (a) Start Date/Time: Time point at which the periodic cycle actions start
- (b) Cycle: Input a cycle of actions.
- (c) End Date/Time: Designate date and time to stop actions.

(4) Daily



- (a) Start Date/Time: Time to start daily actions. If any time interval is not set up, actions are executed only once at the relevant time.
- (b) End Date: Designate date to stop actions
- (c) End Time/Cycle: Input 'End time' and a cycle of actions executed from the time to execute to the time to end.

(5) Weekly



- (a) Set Day: Select a day to execute actions. Multiple choices are allowed.
- (b) Start Time: Time to start actions on the relevant day. If any time interval is not set up, actions are executed only once at the relevant time.
- (c) Endtime/Cycle: Input 'End time' and a cycle of actions executed from the time to execute to the time to end.

(6) Monthly



- (a) Select Day: Select a day to execute actions in the relevant month. Click the mouse on a day to select and the selected day is highlighted in yellow.
- (b) Start Time: Time to start actions on the relevant day and If any time interval is not set up, actions are executed only once at the relevant time.
- (c) End Time/Cycle Setting: Input 'End time' and a cycle of actions executed from the time to execute to the time to end.

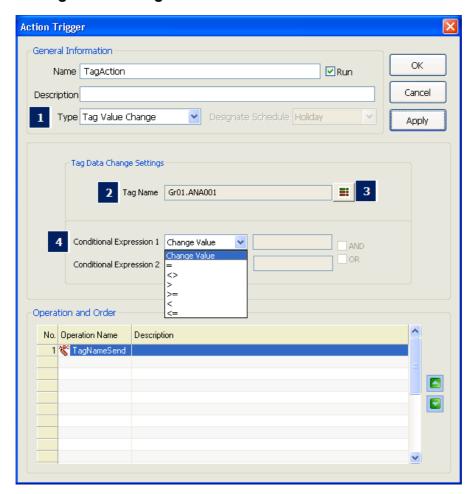
Chapter 15 Action Trigger

(7) Holiday

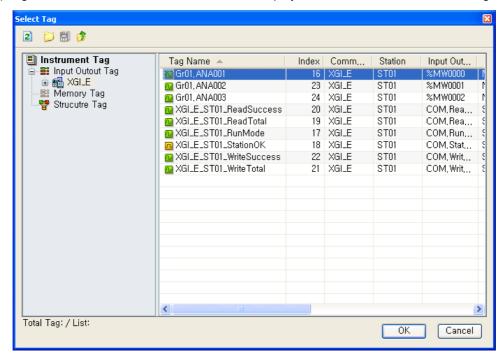


- (a) Start Time: Time to start actions on the relevant date and If any time interval is not set up, actions are executed only once at the relevant time. It is possible to include Saturday.
- (b) End Time/Cycle: Input 'End time' and a cycle of actions executed from the time to execute to the time to end.

15.2.3 Tag Value Change



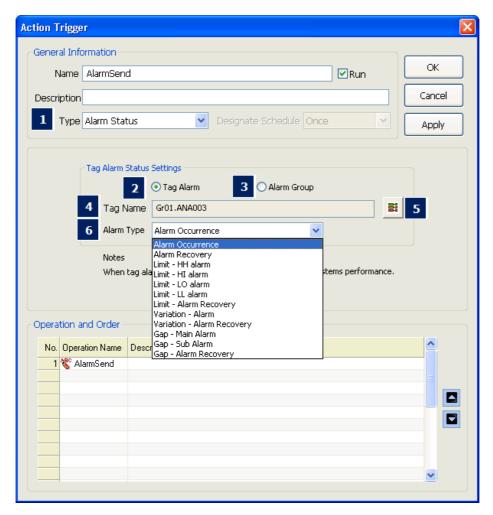
- (a) Type: Select [Tag Value Change].
- (b) Tag Name: If [3] is clicked, the tag selection screen is displayed and the tag name is inserted in box for tag name if one of tags is selected.



(c) Tag Selection Button: Click on this button to display the screen below and select a tag on the screen.

- (a) Conditional Expression: Conditional Expression of the inputted tag value.
 - Change Value: The operation is conducted if the tag value is changed. Not available to set up Conditional Expression. Not available to input values
 - The action is performed if the tag value is equal to the set-up value.
 - <>: The condition is implemented when the tag value is different form the set-up value
 - >: The condition is implemented when the tag value is higher than the set-up value
 - >=: The condition is implemented when the tag value is higher or equal to the set-up value
 - <: The condition is implemented when the tag value is lower than the set-up value
 - <=: The condition is implemented when the tag value is lower or equal to the set-up value
- (b) AND / OR: Implemented when Conditional Expression 1 and Conditional Expression 2 are AND(OR).

15.2.4 Alarm Status



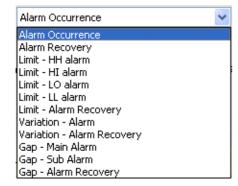
- (a) Type: Select [Alarm Status].
- (b) Tag Alarm: Set up actions for each tag upon the occurrence of alarm.
- (c) Alarm Group: Set up actions for each tag that belongs to an alarm group upon the occurrence of alarm.
- (d) The following items are displayed according to the mode of Tag Alarm and Alarm Group:
 - Tag Name: Click on [5] to display the tag selection screen and input the selected tag name on the screen.
 - Alarm Group: Click on [5] to display the alarm group selection screen and input the selected alarm group name on the screen.
- (e) Tag/Alarm Group Selection: Click to display the 'Tag/Alarm Group Selection' screen and select a tag (alarm group) on the screen.

(f) Alarm Type: Available alarm types are displayed differently between digital and analog tags.

- Digital



Analog



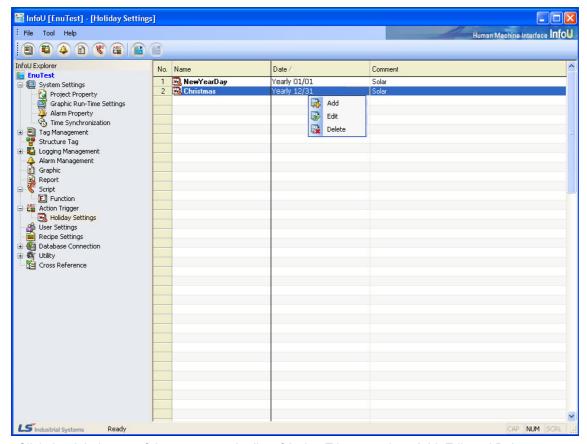
- Alarm Group



15.3 Holiday Settings

(1) Start

If 'Holiday Settings' in the menu or the InfoU explorer is selected, the list of action trigger is displayed as follows:



Click the right button of the mouse on the list of Action Trigger to show Add, Edit and Delete menu.

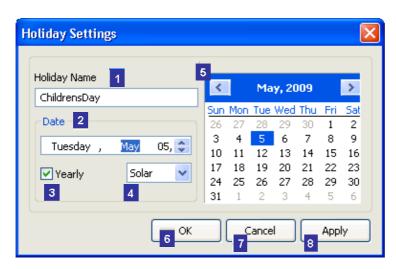
- (a) Add: A new action trigger is added. Double click on the empty item to display the new input screen.
- (b) Edit: The selected action trigger is edited. Double click on the item is double clicked to display the edition screen.
- (c) Delete: Delete the selected action trigger.

(2) Edition Screen

The holiday setting edition screen is of modaless dialog type and it is constructed as follows:

If other item is selected from the above action trigger list while the screen is displayed, information on the selected action trigger is displayed.

Any modification, which may be created during the edition of action trigger, is not reflected during the runtime and it is applied after the runtime is restarted.



- (a) Holiday Name: The selected holiday name is displayed or other holiday name can be newly inputted.
- (b) Date: Input the holiday date.
- (c) Yearly: Set up whether it is repeatedly executed every year or only one time.
- (d) Solar/Lunar: Select either Solar or Lunar.
- (e) Calendar: If a date is selected, the date is inputted in the box for holiday date.
- (f) OK: The contents newly designated are saved and the screen is closed.
- (g) Cancel: The contents newly designated are not saved and the screen is closed.
- (h) Apply: The contents newly designated are saved and the work is continued.

Chapter 16 User Settings

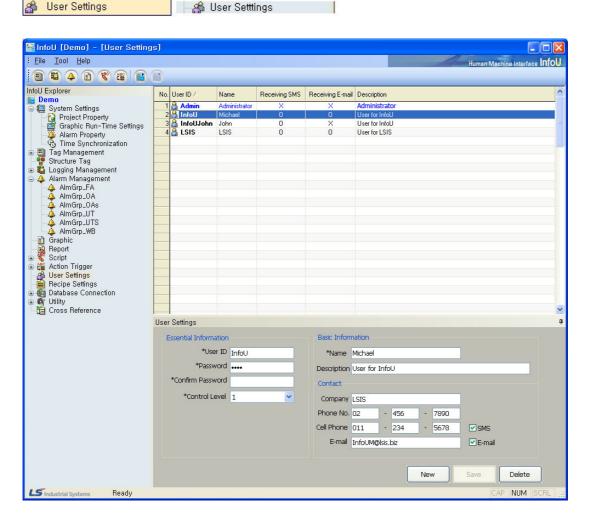
Users are registered for each project and only the registered users are allowed to use InfoU. That is, the registered users are given authority to access to various InfoU functions but their access to some InfoU functions is limited according to their level of authority.

- (1) Alarm Notification Function: The users registered in Alarm Group are notified through e-mail or SMS in case of the occurrence of any alarm.
- (2) Limited Functions by Level: The same as tag levels (1~255), the users are assigned level 1 through 255 and according to this level, their authority to execute and use various InfoU functions such as tag control, inquiry and alarm acknowledgement is limited.

16.1 **Setting Screen**

(1) Start

Select User Settings in the InfoU explorer or the menu.



Chapter 16 User Settings

(2)**User List**

A list of the registered users is displayed to allow the user to delete or modify.

Notice

Admin: System administrator's account as the user registered basically. This account can not be changed except password.

User Settings (3)

When newly registering a user or modifying, input information on the user

- (a) Essential Information
 - 1) User ID: Input User ID
 - 2) Password: Input password.
 - 3) Confirm Password: Re-input password for confirmation.

(b) Control Level (1~255)

Select a control level. Level 1 through level 255 is available. This level is linked with the tag level and used for control and access authority

(c) Basic Information

- 1) Name: Input the user name.
- 2) Description: Give additional information except the user name.

- 1) Company: The usesr's company
- 2) Phone No.: Contactable wire telephone number
- 3) Cell Phone: User's mobile phone number. This information is essential if Option SMS receiving has been selected.
- 4) E-mail: User's e-mail information. This information is essential if Option e-mail receiving has been selected.
- 5) SMS: Check √ in the box if the user wants to receive alarm information through SMS in case of the occurrence of alarm in the alarm group the user is registered in.
- 6) E-mail: Check $\sqrt{\ }$ in the box if the user wants to receive alarm information through e-mail in case of the occurrence of alarm in the alarm group the user is registered in

(e) New

The user information is initialized to allow the user to input new information.

(f) Save

- The registered new information or modified information is saved.
- In case of new registration, the user is added to the list.

(g) Delete

The selected user's information in the list is deleted from DB.

Chapter 17 Recipe

Recipe is a function used when producing various products on the same production/process line according to the prescribed values and it defines a model and controls the process with the combination of component tags (units) and defined values (data). As a method used for large varieties but in small volumes, this function is applicable to various fields.

17.1 Component

To perform the recipe functions, the user needs to understand model, unit and data concept.

17.1.1 Model

It defines a recipe template on the same production/process line and it also defines a model and its components (items).

After a model is defined, settings on the model items are not changeable if units and data have been already inputted.

17.1.2 Unit

It defines a tag collection subordinated to the defined model and it also defines the tags that have the same number with that of the defined items for the model. That is, it has information on the set for the actually controlled points.

17.1.3 Data

It defines a data set subordinated to the defined model and it also defines the actually controlled values.

[Example of Recipe Definition – Coffee Vending Machine]

Recipe can be explained with an application example of coffee vending machine.

When a person is about to drink a cup of coffee from the vending machine, he/she needs to define its model.

The coffee vending machine (model) consists of three components- coffee, cream and sugar (item).

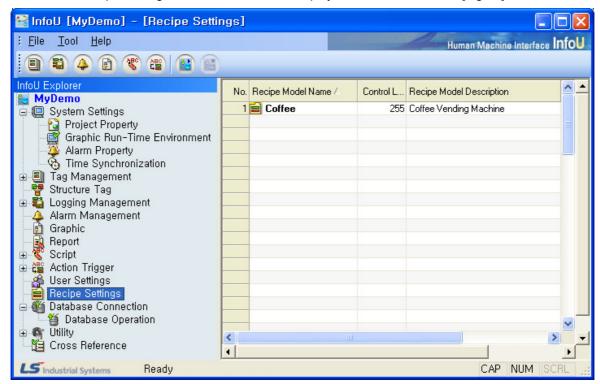
If the vending machine is in the building, its physical location (unit) has to be defined as a vending machine on the first floor or a vending machine on the second floor.

Then, the person needs to decide whether to drink black coffee, cream coffee or cream and sugar coffee and define data on the amount of coffee, milk or sugar that will be actually inserted.

If the person wants to drink a cup of milk coffee from the second floor vending machine, he/she can drink the coffee he/she wants at the place he/she wants with three combinations of model (coffee vending machine), unit (second floor) and data (milk coffee).

17.2 Settings

Select 'Tool'-'Recipe Settings' in the InfoU menu to display the screen as seen in [Figure].



A list of the registered recipe models appears on the screen and detailed information on the relevant model can be viewed or edited through double clicking on each item or pop-up menu.

The following pop-up menu appears upon the click on the list screen.



(1) Add

This is a menu used when a new recipe model is registered. A blank screen appears if this menu is selected.

Up to 64 models can be registered.

(2) Edit

This is a menu used when the information on the selected model is viewed or modified.

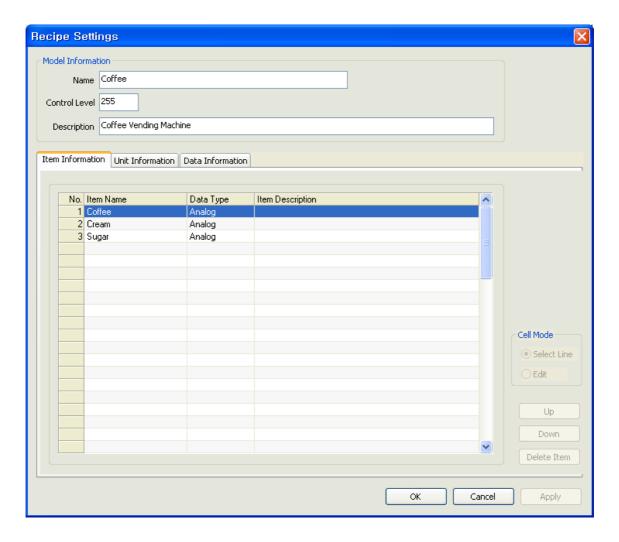
(3) Delete

If selected, a window appears to ask the user to confirm deleting. If "Yes" is selected, the relevant model is deleted and the units and data that belong to the model are also deleted.

17.2.1 **Model Information**

A screen to set up a recipe model

Item on the model and three items such as item, unit and data are divided with taps.



(1) Item

- (a) Name: Register a name of the model. Duplication is not allowed.
- (b) Control Level: Input a control level 1 through 255. The level of the InfoU user shall be higher than the control level of the recipe model to control the recipe (Level 1 is the highest one and Level 255 is the lowest one).
- (c) Description: Register description on the recipe model.
- (d) Item Information: Register items (components) that consist of the model. Up to 64 items are available to register for one model.
 - Item Name: A name of the item
 - Data Kind: Nature of the item data. Select one among Analog, Digital and String.
 - Item Description: Description on the item

Chapter 17 Recipe

(2) Function Button

- (a) Cell Mode
 - Select Line: An option used when selecting an item as a line mode.
 - Edit: An option used when editing grids. The user may directly input information on the grid by double clicking on the grid or selecting 'Edit' button.
- (b) Up: It is activated upon the selection of an item and used for moving the selected item to higher order.
- (c) Down: It is activated upon the selection of an item and used for moving the selected item to lower order.
- (d) Delete Item: It is activated upon the selection of an item and used for deleting the selected item.

Caution

If the linked unit or data is added after a model is defined, the model can not be modified.

17.2.2 **Unit Information**

A screen to register a recipe unit

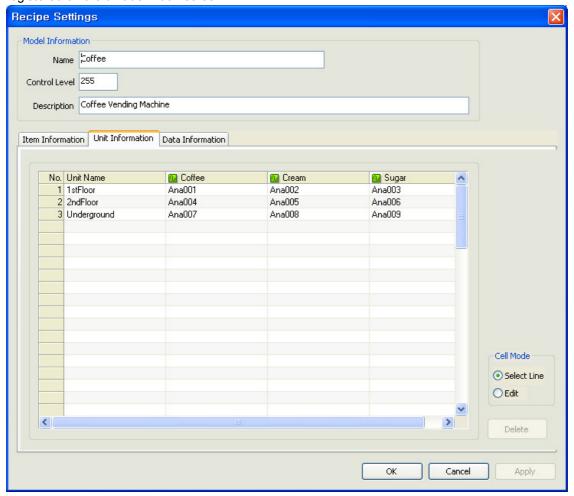
Here, a unit means a tag collection and 10 tags and memory tags can be used.

First, items that have been defined for a certain model appear on the unit screen and unit tags suitable for each item type are registered for the relevant unit.

An editing method to directly input in each cell is used.

Up to 64 unit sets are available for registering (64 unit sets per model).

As seen above, since the item number available to register for one model is 64, total 64 X 64 tags can be registered on the unit definition screen.



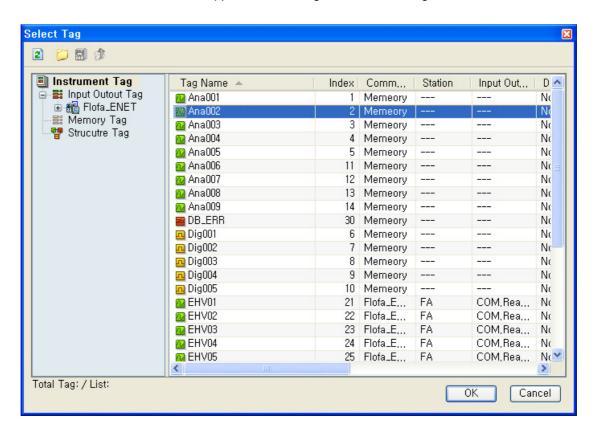
Chapter 17 Recipe

(1) Item

- (a) Unit Name: Register a name of the unit.
- (b) Unit: A screen to register units (tags) suitable for the item type defined for the model.

Directly input the tag name in the cell or click the cell ____ button to select a tag to register.

If ____ button is clicked, the tag selection screen appears and if a tag is double clicked on the tag selection screen, the screen disappears and the tag is marked on the grid.



(2) Function Button

(a) Delete: The currently selected unit name is deleted.

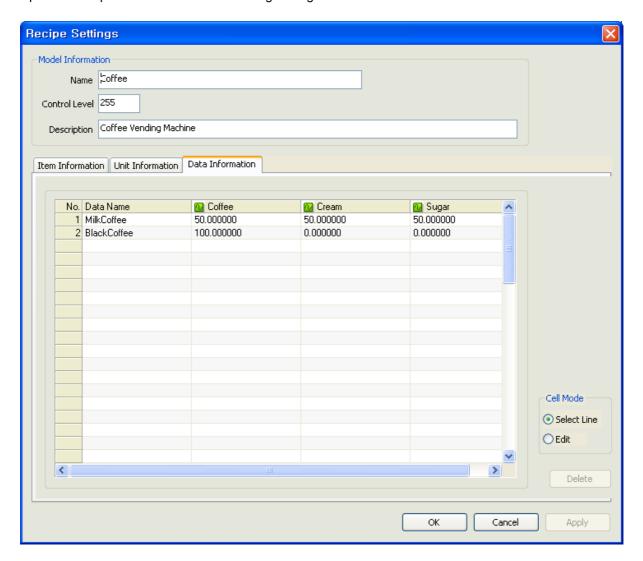
17.2.3 **Data Information**

A screen to register a recipe data

Here, data mean the values that are actually controlled. Register a value suitable for the item type defined for the

An editing method to directly input in each cell is used as done for unit information.

Up to 64 data per model are available for registering



(1) Item

- (a) Data Name: Register a name of the data.
- (b) Data: Define the control value suitable for the item type defined for the model.

(2) Function Button

(a) Delete: The currently selected data name is deleted.

17.3 Recipe Runtime

To use recipe functions online, InfoU provides two methods- one is to use OCX components and the other is to use the internal functions. In addition, they can be used in the Script and Action Trigger.

Recipe functions that can be performed in the Runtime are broadly divided into three functions; Working Set, Real-time Data Inquiry and Saving, and Recipe Control.

17.3.1 **Function**

(1) Working Set Composition (Working Set)

The recipe uses a virtual collection concept called working set.

The recipe's essential components are model, unit and data. Those three components are combined to perform recipe control.

For OCX, relevant components are clicked in the list to form a working set and the internal functions corresponding to each function have been already defined.

(2) Real-time Data Inquiry and Saving

This is a function to import the defined tags for the unit from the real-time server during the runtime to form a working set.

The working set is formed with models and units and this function does not only import the predefined data from DB but also inquire the current tag values.

Surely, the real-time data can be saved in DB. But, only up to 64 data are available per model.

OCX does not provide the 'Delete' function.

(3) Recipe Control

The purpose of forming a working set is to perform recipe control.

Recipe control is ready for performing once the combination of the prescribed model, unit and data (real-time data) is equipped with.

Recipe viewer OCX can perform simple control in InfoU as well as direct control is also available by using data saved the control DB using the working set.

17.3.2 **Recipe Viewer OCX**

Recipe Viewer OCX provides specialized functions at control. For detailed functions, refer to 'Recipe Viewer of Graphic Object"

Internal Function 17.3.3

Recipe functions can be implemented by internal functions.

For internal functions that can perform recipe functions during the runtime, refer to 'Help,'

Chapter 18 Database

18.1 Database Connection

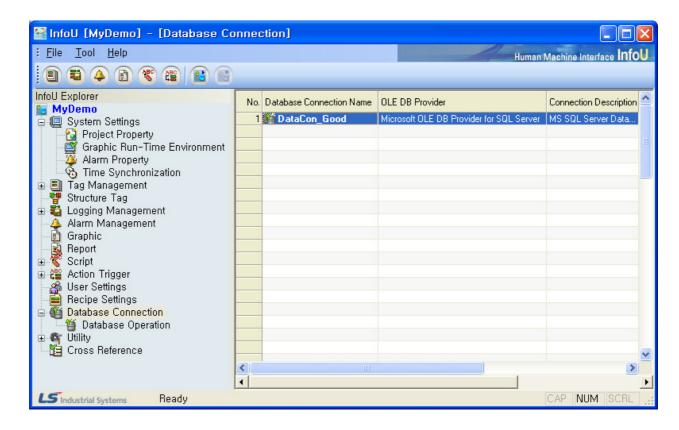
This Database Connection is a function to connect this InfoU system to universal database systems such as SQLServer and Oracle by using Microsoft OLE DB interface.

InfoU can define up to 64 connections.

For Database Operation, it is necessary to define connection with external databases in advance and the connection is used as follows:

18.1.1 Demo

Select 'Tool'- 'Database Connection' in the InfoUD menu, the screen as seen in the [Figure] below is displayed. This 'Database Connection' menu is displayed on the lower-level of the Database Operation.



A list of the registered database for 'Database Connection' appears on the screen and detailed information on the relevant database can be viewed or edited through double clicking on each item or pop-up menu.

Chapter 18 Database

The following pop-up menus appear upon the click on the list screen.



(1) Add

This is a menu used when a new database is registered for 'Database Connection'. If this menu is selected, an empty screen is displayed. Up to 64 databases can be registered for 'Database Connection.'

(2) Edit

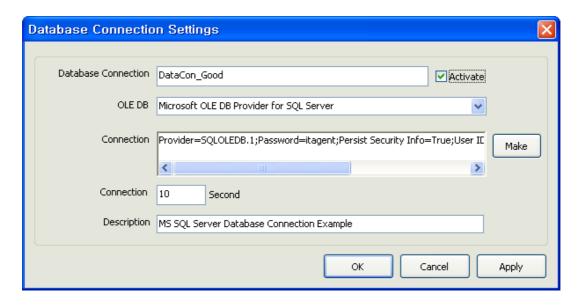
This is a menu used when the information on the selected database is viewed or modified. If one of existing databases is renamed and saved, the database is automatically added and saved as a new database.

(3) Delete

If selected, a window appears to ask the user to confirm deleting. If "Yes" is selected, the relevant database is deleted from the 'Database Connection' list and the 'Database Operation' that belongs to the database is also deleted.

18.1.2 Database Connection Settings

This screen is the one to define 'Data Connection.' Double click on the list or select the pop-up menu to display the following screen. Each item and button is explained as follows:



(1) Database Connection

Register its own name for identification and any duplicated name is not allowed.

(2) Activate

This 'Database Connection' is attempted only for activated data when the database engine is running. The user may register several Data Connections and designate this 'Activate' option only to the necessary connections to prevent the external database support from being wasted

(3) OLE DB

Register OLEDB providers. The combo box shows the list of the registered OLEDB providers in the local computer and the user may either select one from the list or input the name of the OLEDB provider directly on the combo box.

(4) Connection

Register a Connection String used for the relevant OLEDB provider. (If the user does not know a Connection String, he/she may create a file with extension '. udl' and double click it to display a screen to define 'OLEDB Data Connection.' If the user opens the .udl file on the notepad after setting up each item as shown on the screen, he/she may check whether the Connection String has been saved. The user may copy and register this syntax. But, he/she needs to select [Allow saving password] on the checkbox of [Connect] tap to ensure the password is also saved when the .udl file is saved).

(5) Connection (second)

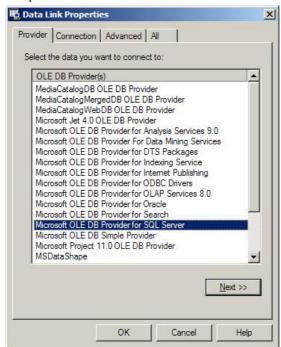
Register a limit time when attempting connection with the designated database. The user needs to input a value higher than 0. If 0 is inputted, connecting is attempted endlessly until it is finally succeeded.

(6) Description

Input description on the 'Data Connection.'

(7) Make

Click 'Make' button to display a screen to define 'OLEDB Data Connection.' If [OK] button is clicked after setting Connection Strings in the order, those are automatically registered on the 4) Connection String text box. (If Check box [Allow saving password] is selected on the [Connection] tap, the password can be included in those Connection Strings.)





[Figure : Screen of Data Link Properties]

18.2 Database Operation

Database Operation is a function to ensure the InfoU system links to universal database systems such as SQLServer and Oracle. That is, the user can record and save InfoU tag values in an external database or import records from the external database to map to InfoU tag values.

For Database Operation, the user may use the predefined Database Connection and register up to 256 operations.

18.2.1 Operation Type

DataExport Operation can be divided into 'SELECT' Operation and 'ETC' Operation according to operation properties.

(1) SELECT

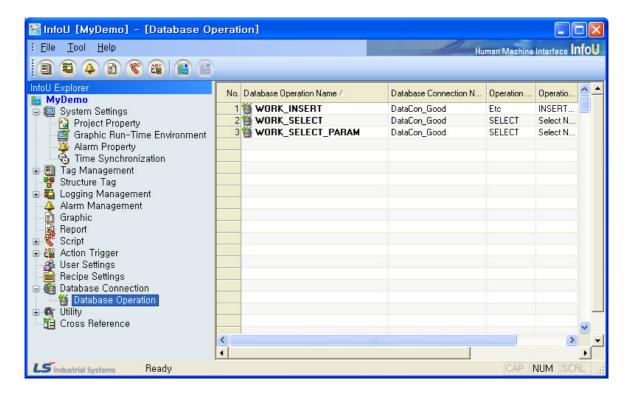
This operation is to import records from an external database and save them in InfoU tag values.

(2) ETC

This operation is to record and save InfoU tag values in an external database.

18.2.2 Demo

Select 'Tool'- 'Database Operation' in the InfoUD menu to display the following screen.



A list of the registered database operations is displayed on the screen. If an item is double clicked or selected from the pop-up menu, the detailed information on the relevant operation can be viewed or modified.

The following pop-up menu appears upon the click on the list screen.



(1) Add

A menu to be used when registering a new database operation. An empty screen appears if this menu is selected.

Up to 256 database operations can be registered.

(2) Edit

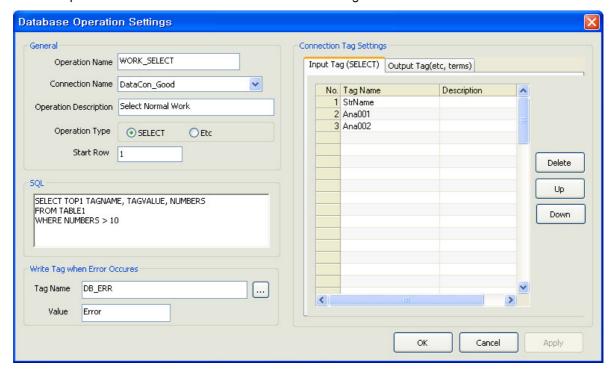
This is a menu used when the information on the selected database operation is viewed or modified.

(3) Delete

If selected, a window appears to ask the user to confirm deleting. If "Yes" is selected, the relevant database operation is deleted.

18.2.3 Database Operation Settings

A screen to define 'Database Operation' Database Operation can be divided into 'SELECT' operation used to import data from an external database and set up them as tag values and 'ETC' operation used to conduct operation for an external database with the InfoU tag data.



[Figure : Example of SELECT Operation (Input Tag)]

Chapter 18 Database

(1) Operation Name

This is the name of a database operation and it has a unique value.

(2) Connection Name

Select a name of the Database Connection to be used for the relevant operation.

The Connections set up in the Database Connection Settings are displayed on the combo box. If no Connection is set up, the database operation can not be saved. (Set up 'Database Connection' first to define 'Database Operation').

(3) Operation Description

Describe the operation.

(4) Operation Type

Specify a type of the Operation.

For 'SELECT', input and output tag taps are activated on the "Connection Tag Setting' window.

For 'ETC', only output tag tap is activated on the window (The input tag is not activated).

(5) Start Row

When a record set is imported from a database, the user can define a record start number that will be a baseline when extracting data of the record set (The value shall not less than 1).

For example, if there are five record sets and the start row is 1, tag mapping starts from the first record while if the start row is designated 2, it starts from the second record, skipping the first record.

(6) SQL

Input a proper SQL syntax (Refer to the following detailed information on SELECT and ETC)

(7) Tag Name

The tag name that will record errors if the operation fails

(8) Value

The tag value that will record errors if the operation fails

(9) Input Tag

- (a) This menu is activated only when 'SELECT' is selected.
- (b) Up to 64 tags per operation can be registered.
- (c) It is used when importing external data and controlling them with tag values.
- (d) Date of the record set is applied in the order of input.

(10) Output Tag

- (a) This menu is activated only when 'SELECT' and other is selected.
- (b) Up to 64 tags per operation can be registered.
- (c) It is used when carrying out operations external data with InfoU tag data.
- (d) Input order is not important. (Tag reservation numbers of SQL syntax and registered output tag numbers are mapped.)
- (e) If the same tag reservation numbers are repeated in the SQL syntax, the corresponding tags to those numbers are also repeated.

(11) Delete

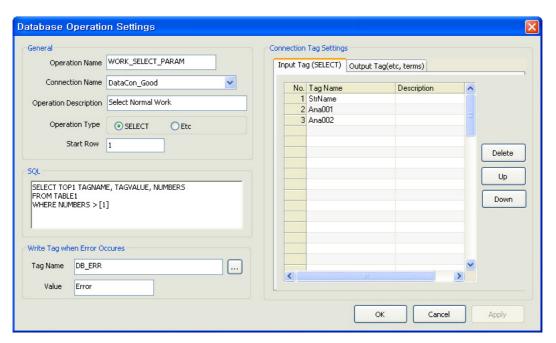
(a) Delete the selected tag.

(12) Up

(a) Move the selected tag one step up.

(13) Down

(a) Move the selected tag one step down.



[Figure: EXAMPLE OF SELECT OPERATION(Output Tag)]

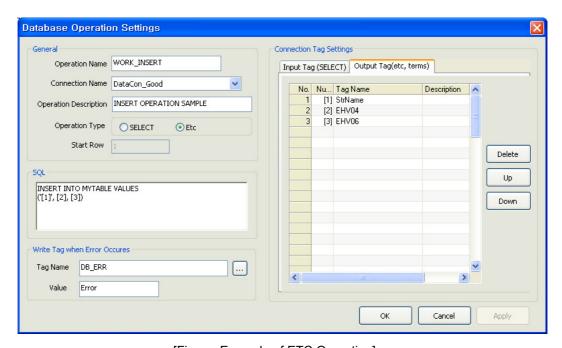
(14) SELECT

- (a) It is used only when the operation type is 'SELECT.'
- (b) The column name next to 'SELECT' is mapped with the tag defined on the phase of Input Tag one
- (c) Tag reservation numbers ([digit]) derived from FROM clause or WHERE clause are mapped with the output tags.
- (d) Tag reservation numbers (numbers higher than 1) are recorded in positive numbers
- (e) Tag reservation numbers or the registered tag values (digit, character..) are expressed properly such as using quotation marks in the SQL syntax according to properties of each DB.
- (Ex) In case that tag values to be used are digits: SELECT COL1 FROM TABLE1 WHERE ID = [1] In case that tag values to be used are characters: SELECT COL1 FROM TABLE1 WHERE ID = '[1]'
- (f) The record set to be inquired starts from the location where the Start Row is registered and controlling tag values starts from that location.
- (g) Controlling tag values is applied in the order of being registered for input tags (It is not necessarily matched between the data number and the input tag number).

Chapter 18 Database

(h) Example of application

Case	SQL	Input Tag	Query Result	Tag Value Control Result
If the input tag number is less than the inquiry data number (Start Row 1)	SELECT ID, NAME FROM TABLE1	Tag01 Tag02 Tag03	ID NAME 1 Hong Gil Dong 2 Gu Sang Yeop 3 SCOTT	Tag01 = 1 Tag02 = Hong Gil Dong Tag03 = 2
If the input tag number is less than the inquiry data number (Start Row 2)	SELECT ID, NAME FROM TABLE1	Tag01 Tag02 Tag03	ID NAME 1 Hong Gil Dong 2 Gu Sang Yeop 3 SCOTT	Tag01 = 2 Tag02 = Gu Sang Yeop Tag03 = 3
If the input tag number is more than the inquiry data number	SELECT ID, NAME FROM TABLE2	Tag01 Tag02 Tag03 Tag04 Tag05 Tag06	ID NAME 1 Hong Gil Dong 2 Gu Sang Yeop	Tag01 = 1 Tag02 = Hong Gil Dong Tag03 = 2 Tag04 = Gu Sang Yeop



[Figure: Example of ETC Operation]

(15) ETC (INSERT, UPDATE, DELETE)

- (a) It is used only when the operation type is 'ETC.'
- (b) It is not used for input tags.
- (c) Tag reservation numbers ([Digit]) used in this syntax are mapped with output tags
- (d) Tag reservation numbers are recorded in positive numbers equal to or higher than 1.

(e) Tag reservation numbers or the registered tag values (digit, character.) are expressed properly such as using quotation marks in the SQL syntax according to properties of each DB. (Ex) In case that tag values to be used are digits: INSERT INTO TABLE1 VALUES ([1], [2]) In case that tag values to be used are characters: INSERT INTO TABLE1 VALUES('[1]', '[2]')

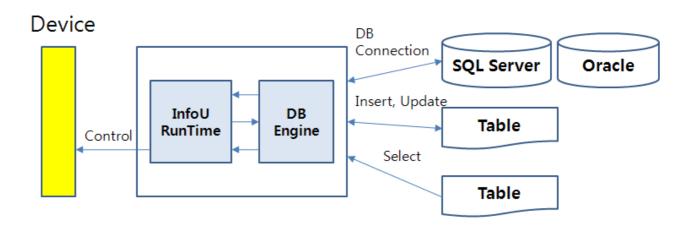
(f) Example of application

Case	SQL	Input Tag and Value	Completed SQL Sentence
If tag reservation numbers are matched with output tags	INSERT INTO TABLE1 VALUES ([1], [2], '[3]')	[2] Tag02 30 VA [3] Tag03 40	SERT INTO TABLE1 LUES (20, 30, '40')
If tag reservation numbers are not matched with output tags	UPDATE HYTAG SET VALUE = [1] WHERE NAME = '[2]' AND TIME = [3]	[1] Tag02 Dong-A SE WH	ET VALUE = 50 HERE NAME = ong-A' ID TIME = [3]
If tag reservation numbers are repeated	INSERT INTO TABLE1 VALUES ([1], [2], [1])	[2] Tan02 30 1	SERT INTO TABLE1 LUES (20, 30, 20)

18.3 Database Runtime

Database Runtime plays a role in performing operations linked to external databases during the Runtime by using the Connect Data and Operation Data set up during the engineering.

The user can perform Database Runtime Function in the stages of 'Script' or 'Action Trigger,'



Log of Database Operation is created everyday in ,csv file on Log Folder of Project Path.

This log contains information on whether connecting to the external database is successful or not, whether each database is successfully perform its operation or other error information.

If the database operation is not normally done, the user can open the file and inspect it to find its cause.

Chapter 18 Database

18.3.1 Function

(1) SELECT

This function is to inquire data from the external database and control them as InfoU's internal tag data. Even though the inquired data number is not equal to that of the registered input tags, it can control them by extracting common data.

(2) ETC

This function is to record and save InfoU tag values in an external database. SQL syntaxes supported by each database can be used in various ways.

18.3.2 Internal Function

The internal functions that perform database operations are supported. For details and how to use, see 'Help' of 'Internal Function.'

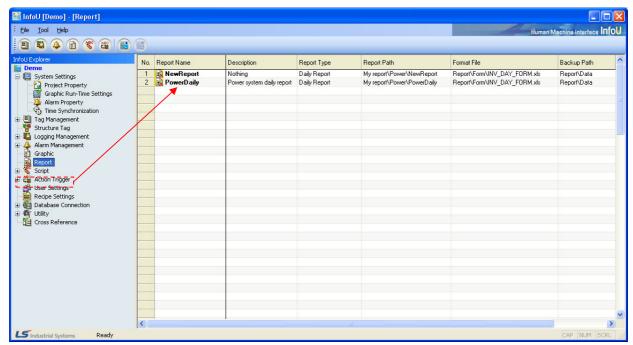
Chapter 19 Report

InfoU Report is to inquire the logging data defined in the logging group and create or output Excel files at a designated time. It can be outputted at any desired interval such as hourly, daily, weekly, monthly, yearly or

- (1) Conveniently create similar reports with the report copy function.
- (2) Report layout formation is easy to use with dragging and dropping function.
- (3) Editing is also available with copying/pasting to Excel.
- (4) Since Excel is used, graph, diagram, formula, statistics and other functions supported by Excel program can be utilized.
- (5) Since the tags used for report can be cross-referred, it is possible to check which tags are used when deleting or modifying tags.

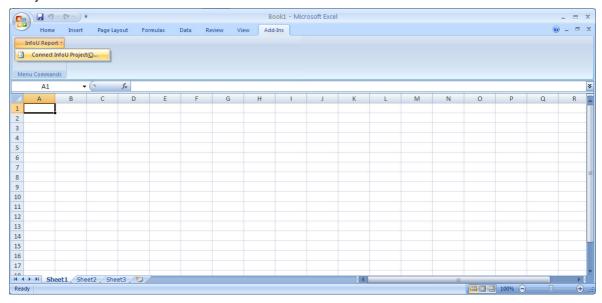
19.1 **Start**

- (1) Select 'Report' from the menu or the InfoU explorer to display the report list.
- (2) Double click the list to execute Excel to make a report.

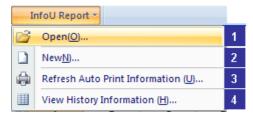


Report Menu 19.2

InfoU Report is produced with ActiveX COM and executed on the MS Excel menu and if it is normally installed, the following [Report] menu items are displayed to allow the user to select "Connect InfoU Project"menu.

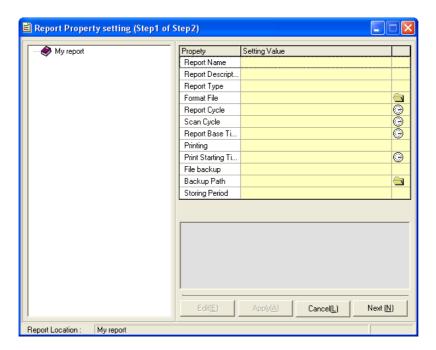


The following menu appears upon the selection of "Connect InfoU Project"if InfoU has been installed and it is available to download basic projects.



19.3 New

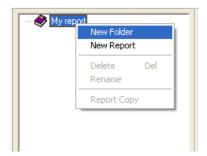
If [New] is selected from the menu, a screen for creating a new report for basic projects or the current project appears.



19.3.1 Menu

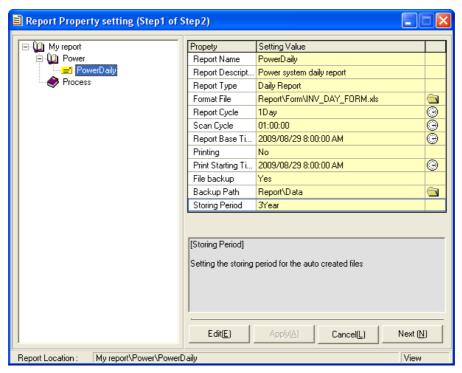
If [New] is selected, the following dialog box appears.

If the highest-level folder is selected and the right side of the mouse is clicked, the following pop-up menu appears.

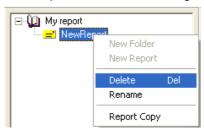


- (1) New Folder: A new folder is created below the selected item.
- (2) New Report: The following screen appears to create a new report.

Chapter 19 Report

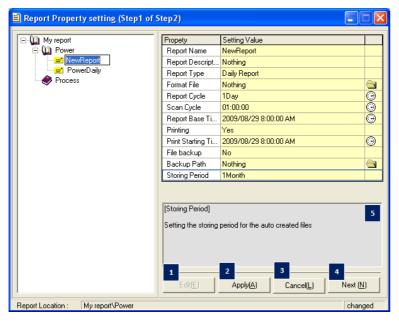


Select the report icon and click the right side of the mouse to display the following pop-up menu.

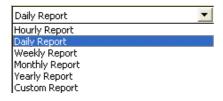


- (3) Delete: Delete the selected report.
- (4) Rename: Change the report name.
- (5) Report Copy: A report can be copied or moved to other folder.

19.3.2 Edit



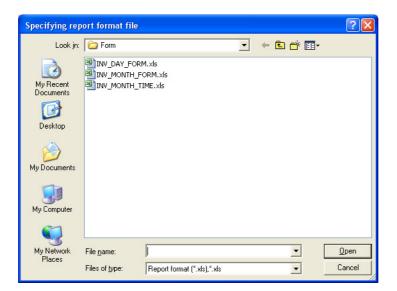
- (1) Edit: Click Edit button to change the properties of the report.
- (2) Apply: The changed properties are saved.
- (3) Cancel: The edition is cancelled.
- (4) Next: The properties are saved and the next step [Report Layout] is displayed.
- (5) Help Window: If the relevant item is selected, instruction and information on its properties are displayed.
- (6) Information on each item consists of the following:
 - (a) Report Name: (16 letters) A name to identify the report. Any two reports do no have the same name in the same folder. It is used to distinguish one from the other.
 - (b) Description: (49 letters) In case that it is obscure to distinguish one from the other only by their names, the user may record details on the report.
 - (c) Type: For regular reports (hourly, daily, weekly, monthly, annually reports), the [Report Interval] property on the bottom is automatically set up while for free reports, the user needs to select an interval. If a value area is selected, a list box showing report types is displayed to allow the user to select one among them.



(d) Format File: Reports are made in Excel file format with functions edited by the user. Therefore, Excel file (.xls) or Excel format file (.xlt) shall be designated.

Click one of the right folder icons to display a selection screen. Then, select a file and click Open button to make the relevant file name automatically inserted. If these files are selected in the project folder, previous project paths are omitted and saved and if other paths except them are selected, the computer name is also automatically added and displayed.

Chapter 19 Report



- (e) Report Cycle: Only Free Report can be set up a cycle to create a report. Data are collected according to the following cycle and once data are collected, a report is created.
 - Ex1) Cycle of daily report = 1 day, Scan cycle = If it is set up one hour, one daily report is created once 24 set of one hour data are collected.
 - Ex2) Cycle of daily report = 1 day, Scan cycle = If it is set up 15 minutes, one daily report is created once 96 set of 15-minute data are collected.



(f) Scan Cycle: An interval displayed in the report as a time distance to collect data. If the clock mouse on the right side of the relevant item is clicked, the following screen appears to allow the user to select a cycle.



(g) Report Base time: A report is written from the set up time on the set up date. Scan Cycle is the same with Report Cycle. That is, a report is created on 'the set up time every report cycle' based on the time to write a report.



- (h) Printing: It displays whether to execute 'Auto Print.' If 'No' is selected, 'Auto Print' is not executed and the following output time is ignored.
- Printing Start Time: A report is outputted from the set up time on the set up date. For example, if it is set up as seen in the following figure, output or backup of the relevant report is executed at noon.



- (j) File Backup: It displays whether to execute 'Report Backup.' Backup is executed at the same time at the same interval with the 'Printer Output' above. If 'No' is selected, backup is not executed while if 'Yes' is selected, the user needs to designated a backup path.
- (k) Backup Path: Select a backup path on the screen and press 'Ok' button. If these files are selected in the project folder, previous project paths are omitted and saved and if other paths except them are selected, the computer name is also automatically added and displayed.



Chapter 19 Report

- (I) Storing Period: The user may set up a period to store backup data. If this period passes, the backup data is automatically deleted.
- (m) Writer: Record a person who writes the report.

19.3.3 Copy/Move

(1) Select 'Report Copy' from the menu below to display the following screen.

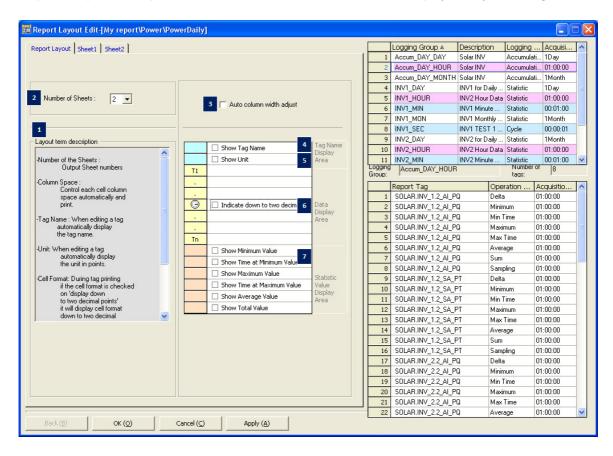


- (2) Destination Path: Designate a path to which a report is copied or moved.
- (3) Report Name: Designate a report name to be created after being copied. If the same name already exists, a new report is created in a form of [report name] (1).
- (4) After modifying the report properties, the user may select [Auto Print Information Renew] from the main menu to apply the modified information online.

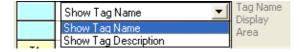
19.4 Layout

19.4.1 **Report Layout**

Once 'which tag is outputted on which date/time is defined, the next step is to define which type the outputted report has. Open the Excel file defined as a format file and display the layout setting screen.



- (1) Layout term description: Each item on the Report Layout screen is described.
- (2) Number of sheet: Set up the Excel sheet number of the relevant report. (Up to 5 sheets are available)
- (3) Auto column width adjust: The row distance of each cell is automatically adjusted and displayed.
- (4) Show Tag Name/Show Tag Description: During the tag edition, the selected tag name or description is automatically displayed.
- (5) To select a tag name or description, press the cell next to the check box.

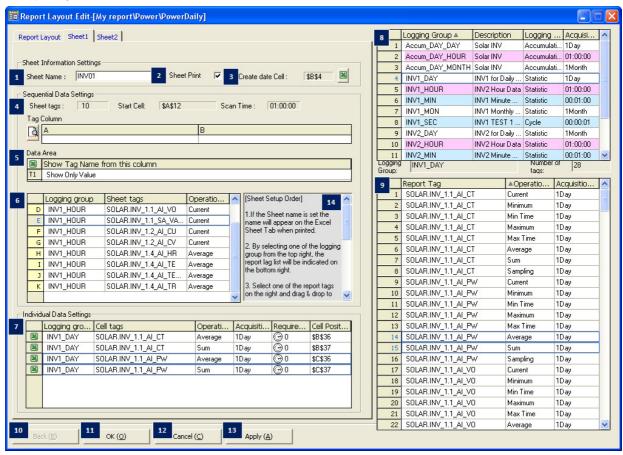


Chapter 19 Report

- (6) Unit: During the tag edition, properties of the selected tag unit are automatically displayed.
- (7) Statistics value: A desired statistics value is automatically computed and displayed.
- (8) Statistics values are displayed on the report in the same order with the one currently displayed and if any specific statistics value is not selected, they are moved up one row and displayed.
- (9) The statistics values computed here will be those of cells and they are the same with the ones for which expressions are used.
- (10) To obtain more precise data statistics values than the actual ones, it is desirable to designate data statistics values to cell tags and display them.

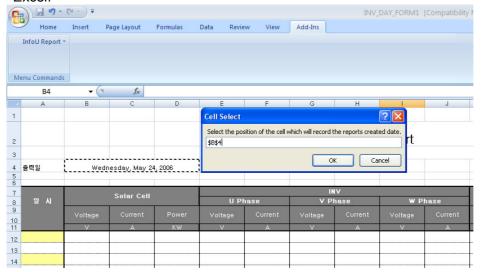
19.4.2 Sheet Property

Set up a layout for each sheet.

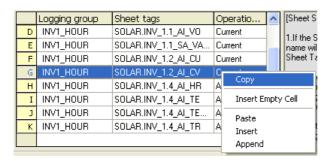


Chapter 19 Report

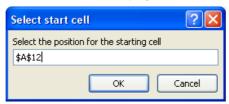
- (1) Sheet Name: Designate and save a name of the Excel sheet when saving a report backup.
- (2) Sheet Print: Set up whether to output the sheet.
- (3) Created Date Cell: Designate a cell to display the date on which the report is created.
- (4) Click on to display the following screen and input cell address by clicking the relevant cell of Excel.



- (5) Sequential Data Settings: Because reports such as daily reports, monthly reports or annual reports have multiple tags that repeat at regular intervals with the same characteristics, those tags shall be registered for 'Sequential Data Settings.'
 - (a) Sheet Tag: The count of the currently selected 'Continual Data' is displayed.
 - (b) Scan Time: The collection cycle of data is displayed.
 - (c) Tag Column: Click on , the currently set-up layout can be viewed in advance and through this, it is available to check whether the settings are properly done.
 - (d)Copy/Paste: Click the right side of the mouse to display the following menu and copy a list to paste, insert or append to Excel files.



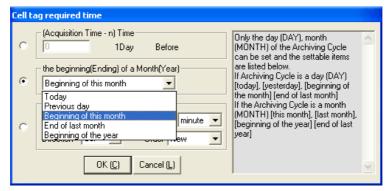
- (e) Empty: If the user wants to add a column on which no data is displayed, he/she may select 'Empty' and use Excel expression because the relevant column can not be filled in with data.
- (f) Data Area: If is clicked, the following screen appears and if a cell is clicked, the cell's address is entered on the box and the value display medthod is displayed.



- (6) A list of the tags to be displayed as continual data appears.
- (7) Select a tag to add to the relevant sheet among the report tags on the right side and drag and drop it to add to the continual setting list (Use the method of dragging and dropping also in deleting).
- (8) Individual Data Settings: Contrary to continual settings, this menu is used to display a report on a particular cell.
 - (a) Select a tag to add to the relevant sheet among the report tags on the right side and drag and drop it to add to the individual setting list (Use the method of dragging and dropping also in deleting).
 - (b) If a clock icon displayed on the demand time row of the tag for individual settings, the following window appears. If multiple-selection is setup, it can apply to the all of the selected tags.



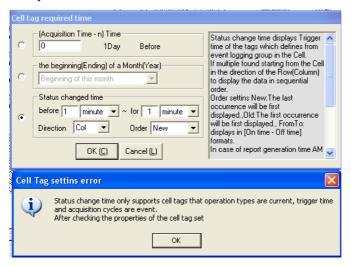
(c) The above screen shows that on which date past individual setting data is imported based on the written time of the report. That is, the entire past data within a range of time of N*collection cycle are imported.



(d) On the above screen, the written time continuously fluctuates over time but, a certain absolute time zone can be set up. Settings are available only for a daily or monthly collection cycle. A collection

cycle can be set up as follows:

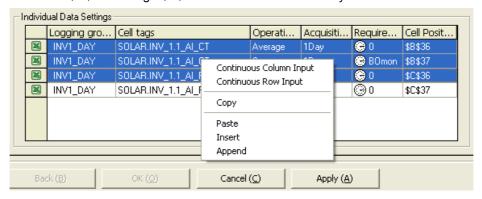
- For a daily collection cycle: "Today", "Previoue day", "Beginning of this month", "End of last month"
- For a monthly collection cycle: "This month", "Last month", "Beginning of this year", "End of last year"



The above screen can be used only for the tags with I/O status values and the time when the relevant tag starts or stops is displayed on the relevant cell.

If [OK] is selected after a certain setting method is set up among the above three setting methods, the selected setting method is entered on the column for demand time.

- 1) Copy/Paste: Click the right side of the mouse to display the following menu. Copy a list and then, Paste, Insert or Append it to Excel.
- 2) Column (vertical) continual input/Row (horizontal) continual input: Because it is inconvenient to click on everytime when inputting cell location to each cell, a certain rule can be used to input cell tags automatically. For example, if "\$B\$30"is selected as the tag location value, select five cells and then select [Continuous Column Input], values from "\$B\$31" through "\$B\$34" are entered. If [Continuous Row Input] is selected, values from "\$C\$31" through "\$F\$31" are entered automatically.



- (9) History Data Collection Model: A list of the data collection models registered in the history data collection models.
 - (a) Red Color: Collection models corresponding to the report data creation cycle. That is, they are data

collection models that have one -hour collection cycle in case of daily reports.

- (b) Blue Color: Collection models that can create reports even though they are not corresponding. That is, they are data collection models that have collection cycle less than one hour and multiples of integers (one minute, 10 minutes, 30 minutes, etc).
- (c) White Color: Collection models that can not be applied to continual tag data areas.

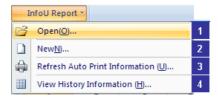
If a certain collection model is selected, a list of tags that belong to the model is displayed at the bottom and the data kinds available to select vary according to the properties of the collection model.

Collection	Analog	Digital	Remark
Model			
Cycle	Current value	Current status	Minimum collection 1second
			Mainly used for the history trend at fast
			time intervals(less than 1minute)
Statistics	Current value,	On count,	Minimum collection 1 minute
	Min value,	Running time,	Min value,
	Max value,	Change count	Max value, Min time,
	Min time,		Max time, Operation cycle 1 minute
	Max time,		Mainly used for report of trend more than
	Average value,		time unit
	Total value,		
	Sampling count		
Accumulation	Change value,	Not supported	Collection cycle: 1hour, 1day, 1month
	Sampling count		When tag data has incremental properties
			such as accumulation electric power meter
			or flow meter, data fluctuation is saved for
			each time unit.
Event	Trigger Time,	Trigger Time,	Display the trigger time and the current
	Current Value	Current Value	value of that time.

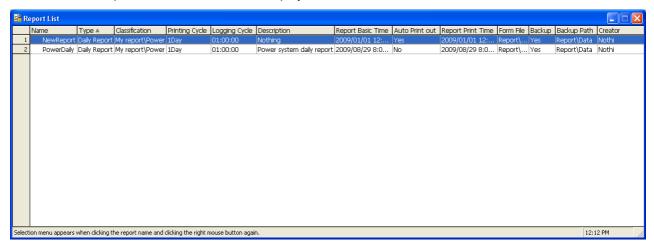
When an item is selected, the tag list to which the relevant collection model belongs and its data kind is displayed in the bottom list. The continual tags are limited to the available tags to be set up according to the report preparation rules but, individual data cells can be entered with any tag regardless of cycle or model kind.

- 1) Tag List: As a tag list of the collection models selected above, it can be set up by dragging and dropped to the continual data or individual data table.
- 2) Back: The screen moves back to the property edition screen after edition is stopped.
- 3) Ok: The settings are saved and the screen is closed.
- 4) Cancel: The edition work is stopped and the screen is closed.
- 5) Apply: Settings are saved.
- 6) Help: Some brief information on the report layout is presented.

19.5 Report Open



If ① [Open] is selected from the report main menu. The report list screen is displayed as seen in the figure below. All of the report information defined is displayed on this screen.



The following actions can be applied to the list.

- (1) Click on the header of each column to rearrange the column contents in the ascending or descending order.
 - (The arrangement order is displayed while the direction of the arrow displayed on the column header is changed)
- (2) The column is rearranged if a certain column is clicked and held while moving the clicked column before other column.
- (3) Double click on the border of the column on the right side to make the width of the column automatically adjusted.
- (4) If some contents are not visible on a cell, bring the mouse on the cell to view all of the hidden part.
- (5) Select an item and click the right side of the mouse to pop up available menus.

19.5.1 Edit

Two steps of [New] are editable individually. That is, operations such as property edition and layout are available individually.



- (1) Property Edit: The report property edition screen is displayed to allow the user to set up properties.
- (2) Layout: The report layout edition screen is displayed to allow the user to set up properties.

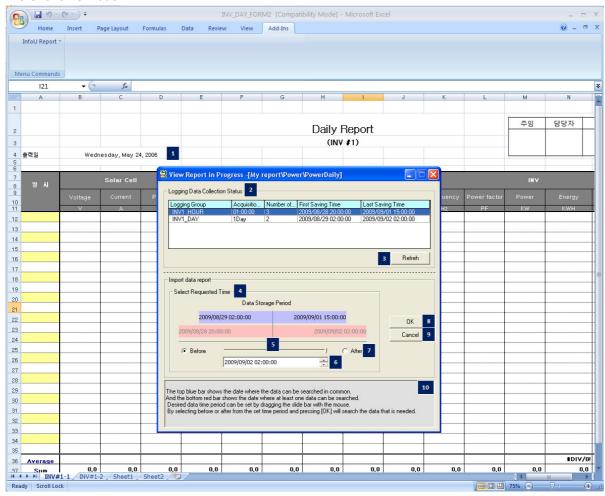
19.5.2 Report View

This function ensures not only the newly saved or created reports are properly searched but also the data retained by the history data collection model are searched during a certain period the user wants.



(1) Database

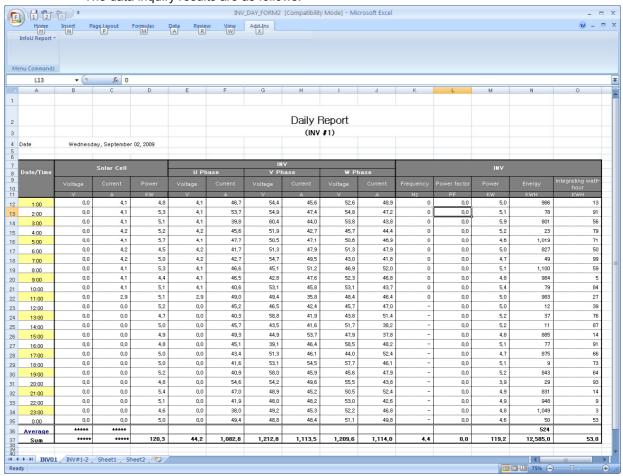
Reports are automatically printed or saved as backup files periodically. This section explains a function to view reports in the time zone the user wants on the screen unlike printer outputs. This function can run only in the runtime mode.



- 1) Format File: A setup Excel format file is opened and the [Progressing Report View] screen is displayed.
- 2) Report Data Collection status: The data saving status of the history data collection models set up in the current report is displayed in a form of list and the list can be referred by the user when he/she searches data and each column is described as follows:
 - (a) Logging Group: The name of the data Logging Group.
 - (b) Acquisition Cycle: Data collection cycle of the relevant collection model.
 - (c) Number of records: The number of the stored data until now.
 - (d) First Saving time: The oldest data time
 - (e) Last Saving Time: The latest data time
- 3) Refresh: The list is renewed with the latest information.

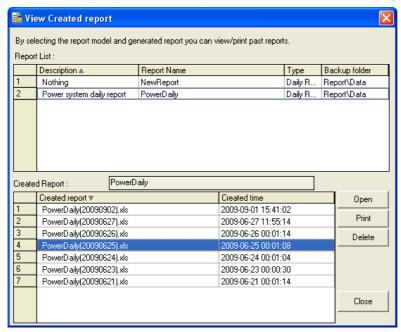
Chapter 19 Report

- 4) Select Requested Time: Select a time period to inquire.
- 5) Data Storage Period: The blue upper bar shows the period during which all of the data can be displayed while the red bottom bar shows the period during which at least one data can be displayed.
 - Reference Date Designate: Set up data in the desired time zone by pressing and holding the slider knob of [Demand Time Select] and moving the mouse.
- 6) Reference Time: Enter and set up a reference time when the user wants to inquire.
- 7) Before/After: Previous or next data to the reference time can be inquired if either Previous or Next is selected.
- 8) OK: Data in the selected time zone are outputted (report cycle/collection cycle).
- 9) Cancel: The inquiry is stopped and the screen is closed.
- 10) Help: The user can be received on the information on how to inquire database reports.
 - The data inquiry results are as follows:



(2) Created Report

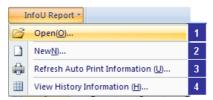
Periodic backup report files are inquired. Click on the menu to display the following screen.



At first, the created report list of the selected report models from the previous report list is displayed at the bottom [2],

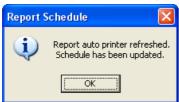
- (a) Report List: The report list defined in the current project is displayed.
 - 1) Click on the list header to arrange the created reports in the order and each column is described as follows:
 - 2) Kind: Kind of the report (Daily report, monthly report, free report, etc....)
 - 3) Report Name: The report name including its folder is displayed.
 - 4) Backup Folder: As the path in which report backup files are saved, the project path displays the corresponding path and any path other than the project path displays the entire paths including computer name.
- (b) Created Report: If the above report model item is selected, the report file list in which the relevant backup path is created appears.
- (c) Open: The report file selected from the bottom list is opened. The same actions are triggered by double click.

19.5.3 Refresh Auto Output Information

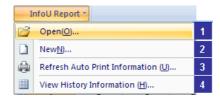


If [Refresh Auto Output Information] is clicked from the Report Main Menu after the report properties are modified, the modified contents are applied online.

This function is executed only in the runtime mode and it renews the report operation list by sending the Renewal Command to the report scheduler responsible for auto output and backup of the report after the report completes its modification process.

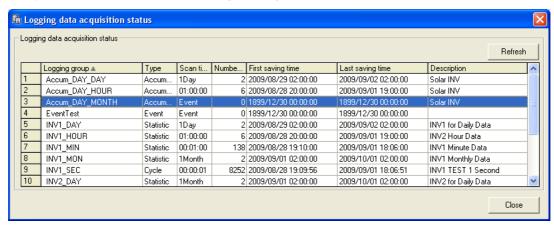


19.5.4 View History Information



If ④ [History Information View] is selected from the report main menu after Report Properties are edited, the following screen appears.

If it is executed only in the runtime mode and the data save status of the history data collection models can be inquired and data can be renewed if [Refresh] button is clicked.



19.5.5 How to Use during the Runtime

During the runtime, a report can be defined on the plant screen and [Created File] report view is called.

- Refer to 'Help' of 'Graphic Script.'

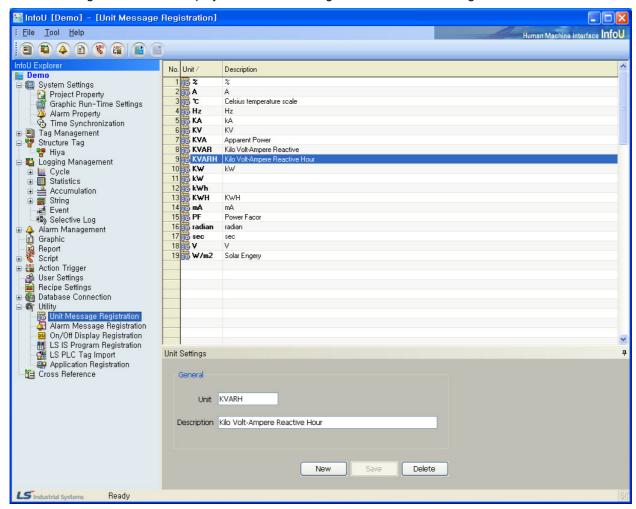
20.1 Unit

A function to set up unit for tag analog values such as length, width, current or voltage

(1) Start Select 'Unit Message Registration' in the menu or the InfoU explorer.



A list of the registered units is displayed as seen in the figure below and the setting screen is shown on the bottom.



(2) Settings

- (a) Unit
 - 1) Input a unit. If any existing unit is selected and changed, the changed unit is automatically added as a new unit.
 - 2) Up to 63 Korean letters and 126 Alphabet letters can be inputted and this item is one of essential information to be inputted.
- (b) Description
 - 1) Up to 127 Korean letters and 254 Alphabet letters can be inputted for this item and this item is allowed to be omitted.
 - 2) Briefly explain on the inputted unit.
- (c) New
 - 1) If this button is selected, [Unit]and [Descirption] are initialized to allow the user to input a new unit.
- (d) Save
 - 1) If [Save] button is selected, the inputted unit is saved and it is shown on the notice window.
- (e) Delete
 - 1) If [Delete] button is selected, the inputted unit is deleted and it is shown on the notice window.
 - 2) It is also available to select multiple items in the list and delete them at once.

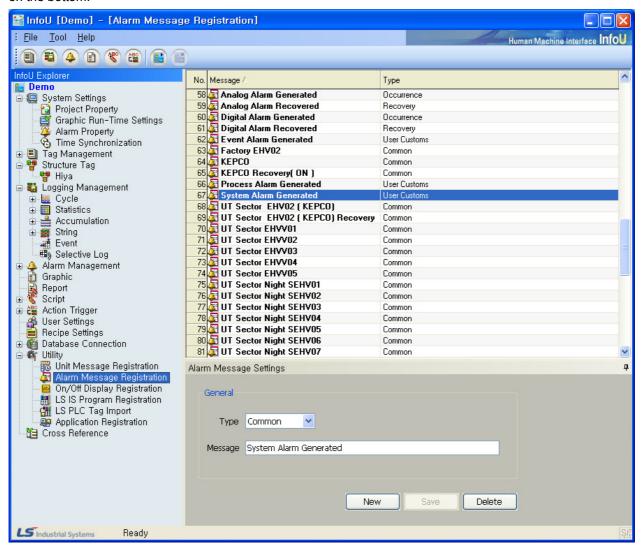
20.2 Alarm Message Registration

A function to register a string to be displayed in case of the occurrence or recovery of alarm

(1) Start Select 'Alarm Message Registration' in the menu or the InfoU explorer.



A list of the registered alarm messages is displayed as seen in the figure below and the setting screen is shown on the bottom.



(2) Settings

- (a) Type
 - 1) Select a type of the occurred alarm.
 - 2) Four types such as [Common], [Occurrence], [Cancellation] and [User Customs] are available and this information is mandatory.
- (b) Message
 - 1) Input a string to be displayed in case of the occurrence of alarm.
 - 2) Up to 63 Korean letters and 126 Alphabet letters can be inputted and it is one of essential information to be inputted.
- (c) New
- (d) The existing [Message] is initialized to allow the user to input a new message.
- (e) Save
 - 1) If [Save] button is selected, the inputted message is saved and it is shown on the notice window.
- (f) Delete
 - 1) If [Delete] button is selected, the selected item is deleted and it is shown on the notice window.
 - 2) It is also available to select multiple items in the list and delete them at once.

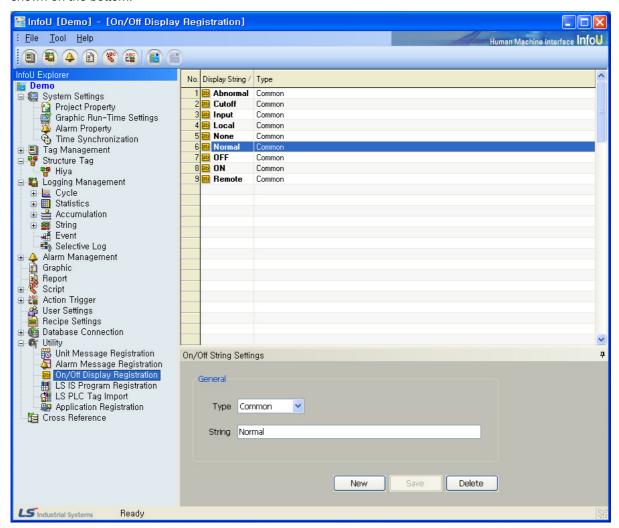
20.3 On/Off Display Message Registration

A function to set up a string to be displayed regarding to On/Off of the tag digital value.

Select 'On/Off Display Message Registration' in the menu or the InfoU explorer.



A list of the registered On/Off messages is displayed as seen in the figure below and the setting screen is shown on the bottom.



(2) Settings

- (a) Type
 - 1) Three types such as [Common], [On (1)] and [Off (0)] are available and this information is mandatory
 - 2) Up to 63 Korean letters and 126 Alphabet letters can be inputted and it is one of essential information to be inputted.
- (b) String
 - 1) Input a string to be displayed.
 - 2) Up to 63 Korean letters and 126 Alphabet letters can be inputted and it is one of essential information to be inputted.
- (c) New
 - 1) The existing [String] is initialized to allow the user to input a new one.
- (d) Save
 - 1) Select [Save] button to save the inputted On/Off string and display it on the notice window.
- (e) Delete
 - 1) Select [Delete] button to delete the selected item display it on the notice window
 - 2) It is also available to select multiple items in the list and delete them at once.

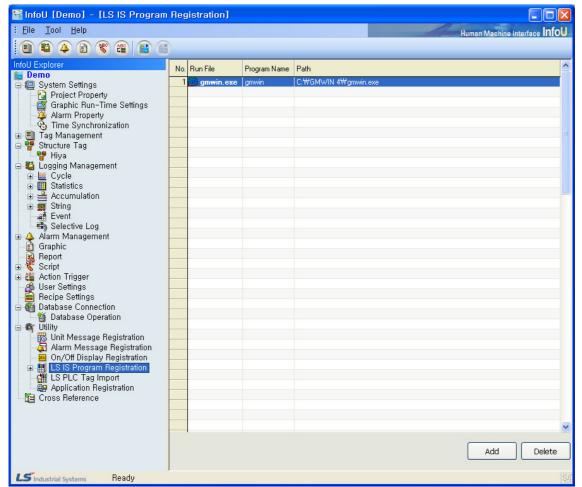
20.4 LSIS Program Registration

A function to register GLOFA-GM programs such as GMWIN.exe, PLC products of LS Industrial System, in the InfoUD to execute GMWIN programs and also to import information on the point engineered by GMWIN.exe in form of tag information

(1) Start Select 'LS IS Program Registration' in the menu or the InfoU explorer.

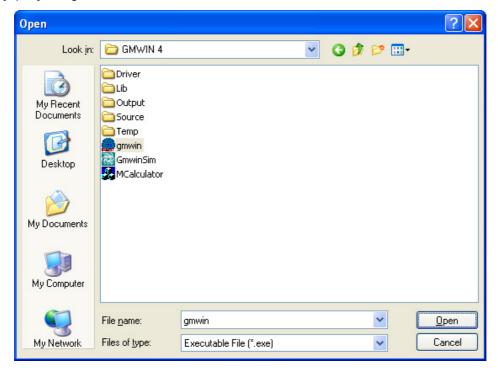


A screen as seen in the figure below is displayed and [Add] and [Delete] button appear on the bottom.

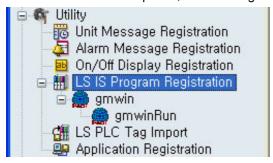


(2)Add

(a) Select [Add] button to display the [Open] dialog box. Select GMWIN.exe-LS PLS program-on the [Open] dialog box.



(b) Once the selection is completed, the following menu is created on the InfoU Explorer.



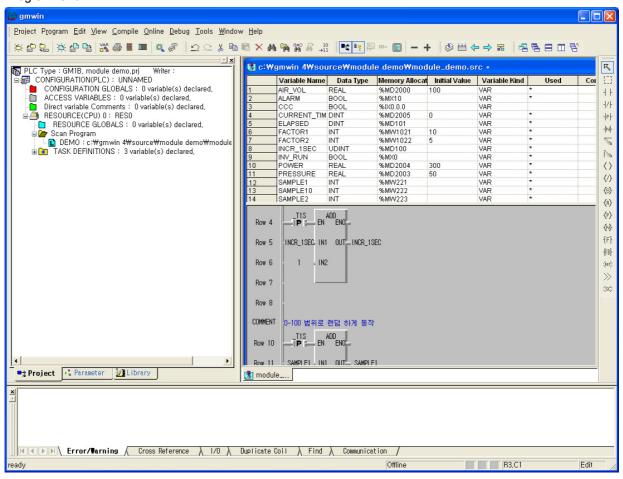
(c) The sub items consist of the following components:

Sub Item 1	Sub Item 2	Description
[Program Name]	[Program Name] + Run	Run the registered program
	[Program Name] + DB Import	Import DB registered in the program

(3)Run

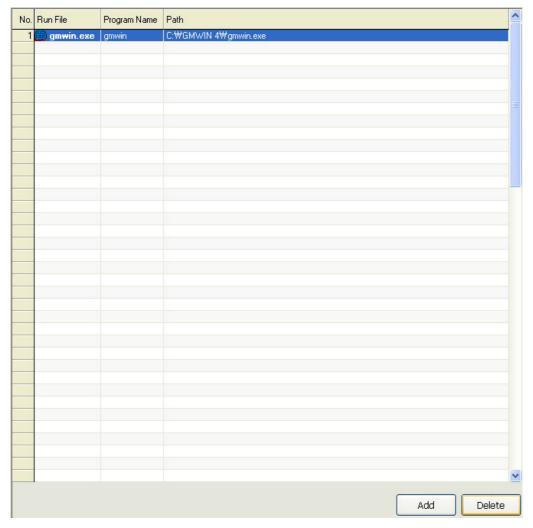


If [GMWIN Run], one of registered programs, is selected on the [InfoU Explorer], GMWIN.exe Program is run.



(4)Delete

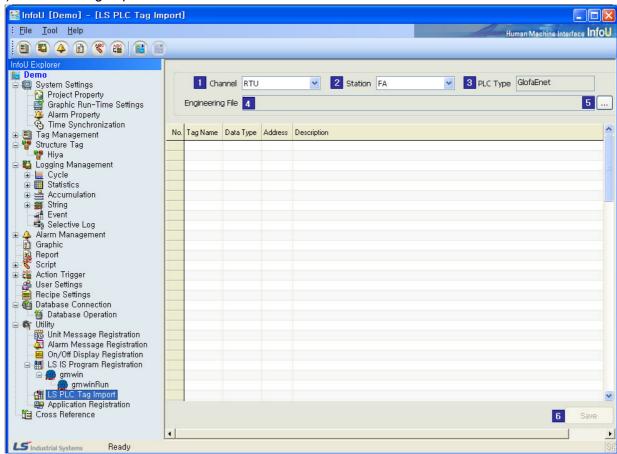
(a) Select a program to delete and select [Delete] button to display the dialog box.



- (b) Once deleting is completed, the 'Delete Notice' window appears and the sub item that used to be displayed on the [InfoU Explorer] is also deleted.
- (c) If 'DB Import' is performed and a tag is saved on the deleted program, the saved tag exists regardless of the deletion of the program.

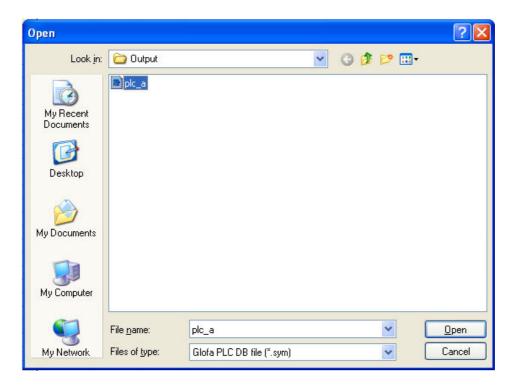
20.5 LS PLC Tag Import

(1) LS PLC Tag Import

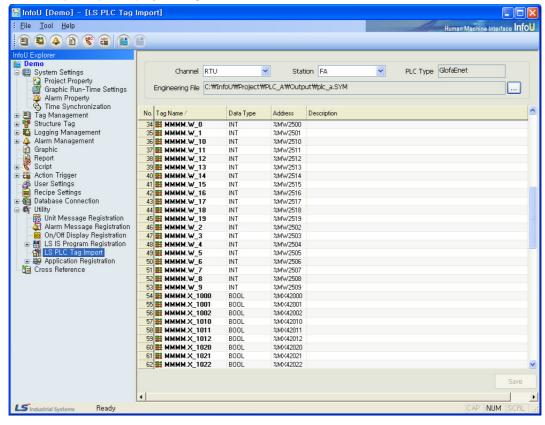


If [LS PLC Tag Import] is selected on the InfoU Explorer, menu bar or tool bar, the screen above is displayed.

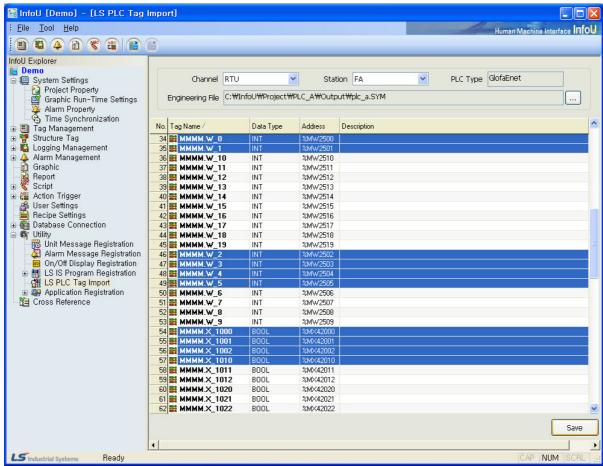
- ① GLOFA, XGK or XGI communication channels registered in the current project are displayed (GLOFA-Enet/Cnet/Loader, XGK-Enet/Cnet, XGI-ENet/CNet).
- ② If a communication channel is selected, a list of the stations registered in the relevant channel is displayed..
- ③ The PLC type of the currently selected communication channel is displayed.
- ④ The path of the PLC engineering file to be imported is is displayed.
- ⑤ A PLC engineering file can be slected. For GMWIN files, a *.sym file, for XG5000 engineering files, a *.csv file can be slected according to their types.



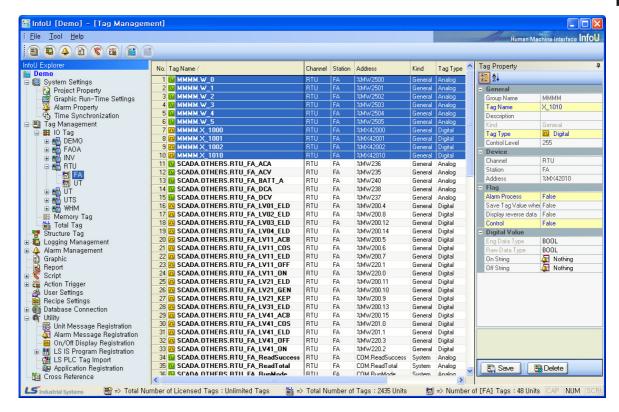
- Select an engineering file of the relevant PLC project.
- Click on [Open] to display the tag list below on the screen.



6 PLC Tag Save



- Select [IO Device] and [Device Name] to save.
- Select the LS PLC Application through mouse drag or Ctrl + LClick, Shift + LClick, Grid RowColumn Header(Select All). Click on [Save] button.



- The screen moves to the list of the tags that belong to the selected IO Device/Instrument.
- The program name created during the PLC engineering operation is inputted as a tag group and variables are inputted as individual tags.

Data types are saved according to PLC data kinds as follows:

PLC Type	Tag Type	Device Type
tBOOL	BOOL	BOOL
tBYTE	UINT8	UINT8
tWORD	UINT16	UINT16
tDWORD	UINT32	UINT32
tLWORD	DOUBLE	UINT64
SINT	INT8	INT8
tINT	INT16	INT16
DINT	INT32	INT32
LINT	DOUBLE	INT64
USINT	UINT8	UINT8
tUINT	UINT16	UINT16
UDINT	UINT32	UINT32
ULINT	DOUBLE	UINT64
REAL	FLOAT	FLOAT
LREAL	DOUBLE	DOUBLE
TIME	UINT32	UINT32
tDATE	UINT16	UINT16
TIME_OF_DAY	UINT32	UINT32
DATE_AND_TIME	DOUBLE	UINT64
PLC_STRING	STRING	STRING
BIT	BOOL	BOOL
NIBBLE	UINT8	BIT4
DIRECT_TYPE	Not supported	
ENUM_TYPE		
ENUM_MEMBER		
ARRAY_TYPE		
ARRAY_MEMBER		
STRUCTURE_TYPE		
STRUCTURE_MEMBER		
FB_NAME		
LABEL_TYPE		
VAR_ARRAY_MEM		
FB_OUTPUT		
NOTYPE		
VAR_TYPE_NUM		

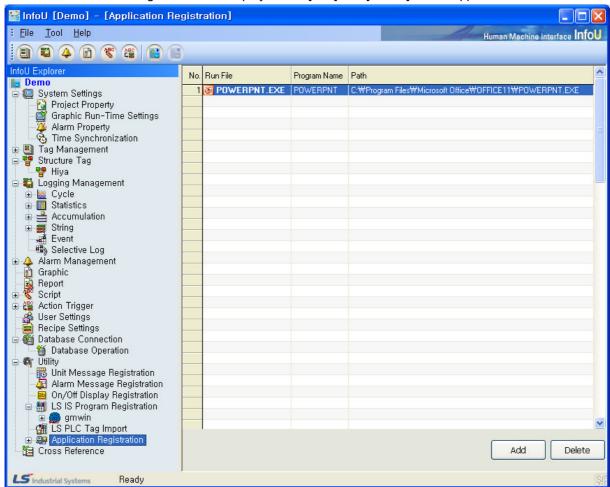
20.6 Application Registration

A function to register other programs of the computer in which InfoUD is installed in InfoUD to run external programs.

(1) Start
Select and run 'LS IS Program Registration' in the menu or the InfoU explorer.

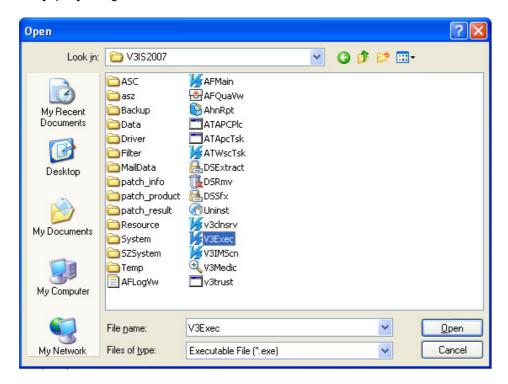


A screen as seen in the figure below is displayed and [Add] and [Delete] button appear on the bottom.



(2) Add

Select [Add] button to display the [Open] dialog box and select *.exe of the program to be registered on the [Open] dialog box.

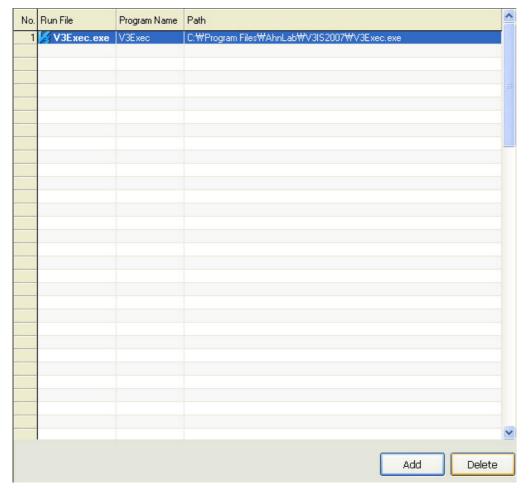


(b) The list above is displayed and the relevant program is registered as an icon in the Application Registration menu of the InfoU Explorer.



(c) Run: Click on the icon to run the relevant program.

(3) Delete



- (a) Select a program to delete and select [Delete] button to display the dialog box
- (b) Once deleting is completed, the 'Delete Notice' window appears and the sub item that used to be displayed on the [InfoU Explorer] is also deleted.

Chapter 21 Cross Reference

The correlation of the registered tags such as graphic file, history settings (collection model), alarm group and report can be inquired on one screen. Especially, it has a function to detect any wrong use of deleted tags so that it is possible to conduct more effective engineering operations and in addition, moving to each engineering screen is available for editing. The status of tag use or module use can be monitored even during the runtime and it is also possible to move to the inquiry screen conveniently.

21.1 Function and Property

(1) Tag Use Trace

It is possible to inquire information on the modules used for each tag at one place and the use details of each tag can be traced conveniently.

(2) Filtering

Through filtering by tag, the correlation can be inquired only among the desired tags

(3) File Save

The inquired contents are saved in CSV files to make it possible to edit and output in various types of documents.

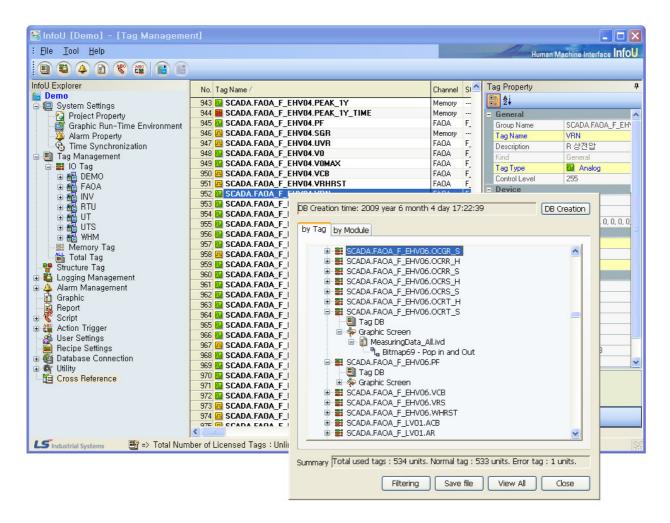
(4) Inquiry Function by tag or by module

Two types of correlation inquiry screens are presented, one is to compile the used modules by tag and the other is to compile the used tags by module.

(5) Jump

This jump function, which enables to skip to the tag builder, history setting, graphic editor or alarm group setting screen, helps better effectiveness inspection and immediate inquiry or edition.

21.2 **Engineering Function**

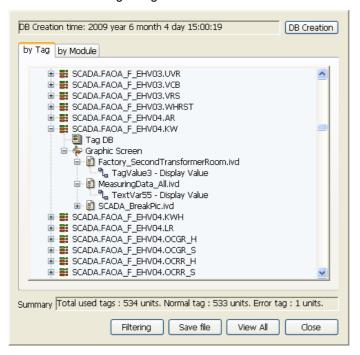


If 'Cross Reference Inquiry' button is pressed during the engineering, the correlation among the entire tags is displayed. However, if it is executed on the screen where any tag builder or other tags are selected, the screen shows the relevant tags only.

The correlation can be expressed by two taps; by tag and by module.

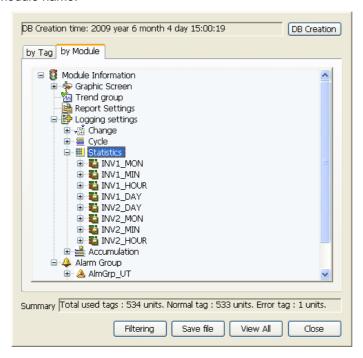
(1) Inquiry Screen (By Tag)

This inquiry screen by tag shows module details used for each tag. If the connected module name is double clicked, it jumps to the setting screen of the relevant module to set up Tag Connection of the selected module Ex) If an alarm group is double clicked, the screen is converted to the setting screen of the alarm group in which the relevant tag is registered.



Inquiry Screen (By Module)

This inquiry screen by module shows information on the tags used in each module under the relevant module name.



Chapter 21 Cross Reference

- (3) Details
 - (a) DB Creation Time

It displays the time when Cross Reference DB (Hereinafter refer to CRDB) is created. If there is no CRDB, a message saying 'There is no DB' is displayed. At this time, it needs to create a new DB.

- (b) DB Creation
 - This button is used when there is no CRDB or when CRDB is attempted to be recreated after engineering operation, and if this button is pressed, an existing CRDB is recreated.
- (c) Summary

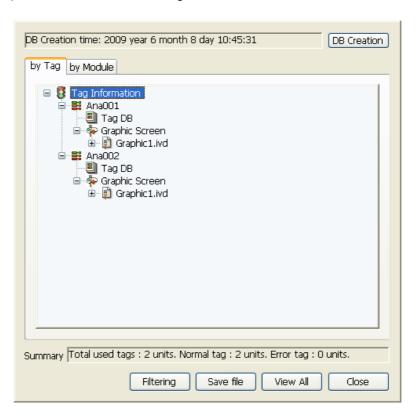
The number of the entire use tags and the number of normal tags and error tags are displayed.

(d) Filtering

Through filtering by tag, only the desired tags can be inquired.



- Tag Name: Input a tag name to inquire the use information on the relevant tag. At this time, be cautious to distinguish capital and small letters for the tag name and it is possible to inquire multiple tag names at first when only small amount of tag names are inputted.
 - Ex) Results when filtered with tag names



- 2) Tag Browser Button: Tag Browser is outputted to select a tag.
- 3) View Error Tag: A function to identify only the tags that have errors.



- Save File (e)
 - The inquired contents are saved in CSV Files and it is possible to output as other documents.
- (f)

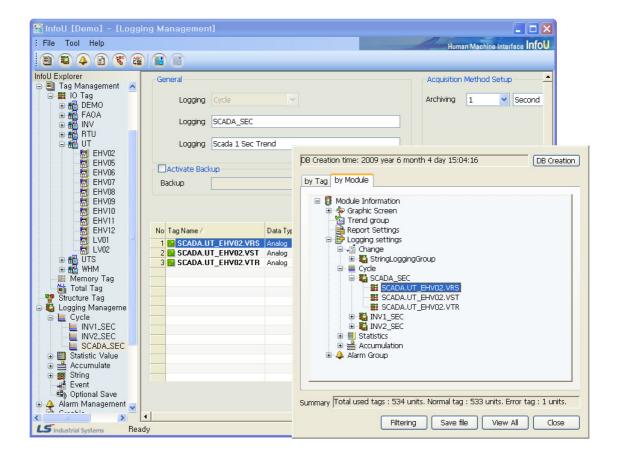
The 'Filtering Result' screen is converted to the 'Inquire All' screen

Chapter 21 Cross Reference

(4) Screen Jump

If one of graphic settings, history settings and alarm group settings is double clicked, the screen is converted to an engineering screen of the relevant settings and collection models of the relevant file and the tag settings of the alarm group are displayed.

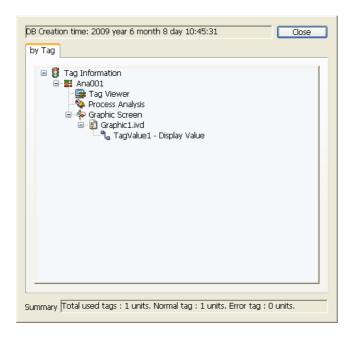
Also, if a tag is selected on the engineering screen, the inquiry contents on the cross reference screen is changed into the selected tag contents in real-time.



21.3 Runtime Function

The use details on the relevant tag can be inquired on the graphic screen or on the screen that uses the tag during the online running.

A full screen is not provided during the engineering and only the use details corresponding to the selected tag are expressed.



(1) Tag Viewrer

It moves to the tag viewer on which tag information can be inquired.

(2) Accident Analysis

When 'Accident Analysis' is double clicked, it moves to the history alarm inquiry screen and shows the status of alarm occurrence in the relevant tag. (Refer to Alarm Viewer Manual)

(3) Screen Move

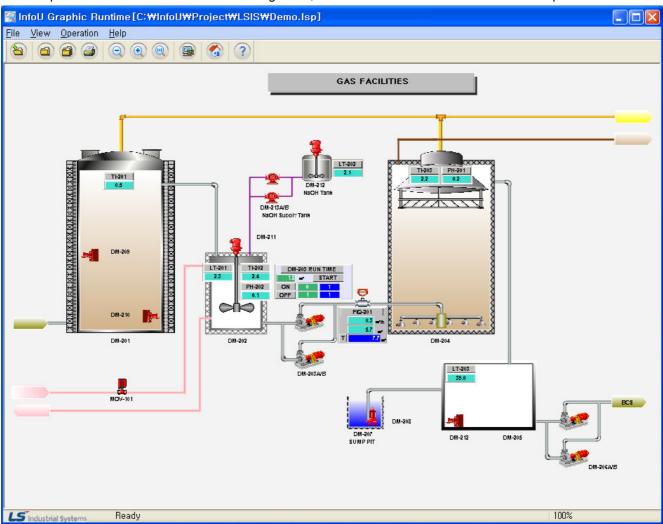
If the graphic screen is double clicked, it moves to the relevant screen.

Chapter 22 Graphic Runtime

This Graphic Runtime helps to monitor or control real-time or history information through the graphic screen by using HMI real-time or history information.

22.1 Run

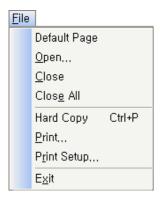
This Graphic Runtime runs if the runtime running starts, and the initial screen is created in the Graphic Runtime Settings.



22.2 Menu Composition

Upper-level Menu	Menu	Description
File	Default Page	Open the default page set up in the runtime execution environment.
	Open	Open existing document.
	Close	Close the current document.
	Close All	Close all of documents.
	Hard Copy	Print the graphic runtime execution screen in a type of hard copy.
	Print	Print document.
	Print Setup	Set up print options
	Exit	End the graphic runtime.
View	Standard Bar	Show or hide color tools.
	Status Bar	Show or hide status tools.
	Menu	Show or hide main menus.
	Zoom	Zoom in/out the page in set up ratios.
	Auto Adjust	Automatically adjust the screen size to a certain ratio fit to the current view size.
	Navigation	Show or hide navigation tools for screen moving.
	Full Screen	Convert the page into a full screen size.
Operatin	User Login	Log in the registered user.
	User Logout	Log out the logged in user.
	Change Password	Change the log-in password of the registered user.
	Alarm Popup	Show or hide the alarm pop-up screen.
	Tag View	Show all of the tags registered in DB
	Cross Reference View	Show Cross Reference information on the current project.
	I/O Station Status	Show the communication status of the currently connected I/O driver.
Help	Help Finder	Show the Help items of the graphic runtime.
	Graphic Runtime Information	Show the version information of the graphic runtime.

22.3 File

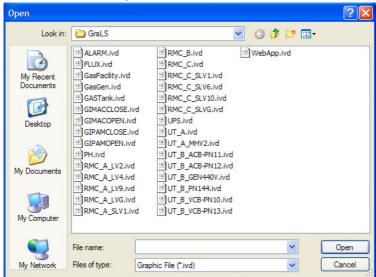


(1) Default Page

Open the default page set up in the runtime execution environment.

(2) Open

This command is used to open one of the existing documents. If this command is made, a dialog box appears to allow the user to designate a file to open from it.



(3) Close

The current document is closed.

(4) Close All

The currently opened documents are all closed.

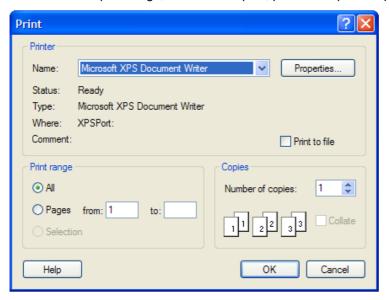
(5) Hard Copy

Execute the graphic runtime execution screen as hard copy.

Chapter 22 Graphic Runtime

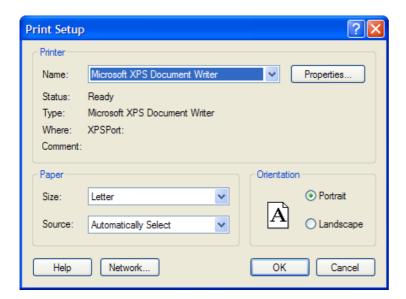
(6) Print

It is used when printing document. If this command is executed, "print" conversation box appears to allow the user to select a print range, number of copies, printer and printer option.



(7) Print Setup

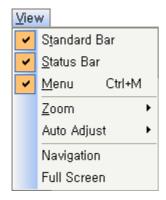
Paper size and orientation for printing documents can be selective.



(8) Exit

The graphic editor is ended.

22.4 View



(1) Standard Bar

It is used when showing or hiding standard tools. Standard tools are command icons the most frequently used in the graphic editor. When they are shown, checkmark appears in front of the relevant menu.



(2) Status Bar

It is used when showing or hiding status tools. Status bar provides Help Message. When it is shown, checkmark appears in front of the relevant menu.

(3) Menu

It is used when showing or hiding main menus. When it is shown, checkmark appears in front of the relevant menu.

(4) Zoom

It is used when enlarging or reducing the screen in a predefined ratio during the page runtime.

(5) Auto Adjust

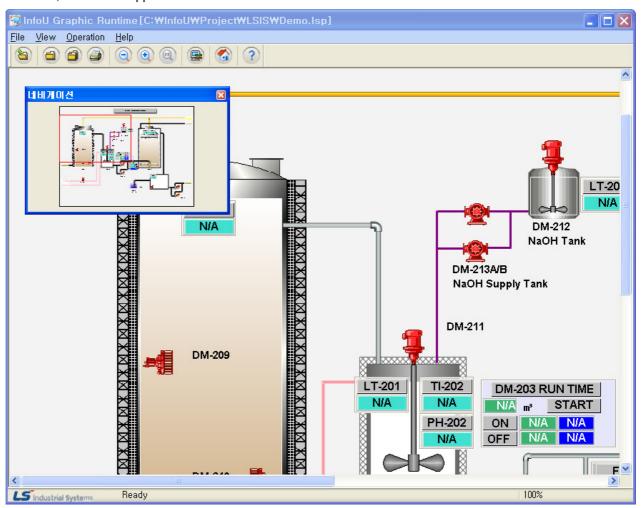
It is used when adjusting the view size of the current runtime screen to a certain ratio or setting up to automatically adjust the horizontal and vertical proportions. When it is shown, checkmark appears in front of the relevant menu.

Chapter 22 Graphic Runtime

(6) Navigation

It is used when showing or hiding navigation tools.

If the current view screen is narrow, this navigation is used to move it to a desired screen area. When it is shown, checkmark appears in front of the relevant menu.



(7) Full Screen

It is used when displaying the current page in a full screen size.

22.5 Operation



(1) User Login

It is used when the registered user logs in. If login is done, the user may use the functions assigned to him/her. If this command is performed, the following InfoU Administrator Login window is displayed.



(2) User Logout

It is used when the logged in user logs out.

(3) Change Password

It is used when changing the registered user's login password.

(4) Alarm Popup

It is used when showing or hiding the alarm pop-up screen.

This menu is activated only when the user logs in.

If this item is not checked, the following alarm pop-up window appears to call the operator's attention in case of the occurrence of a new alarm in the system.

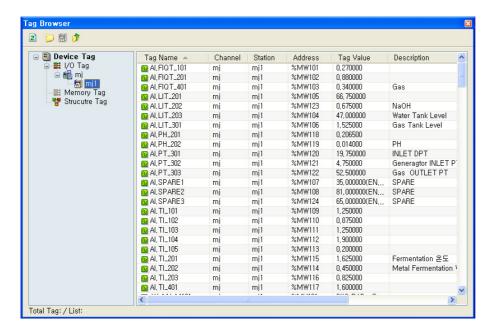
If the Cross Reference button on the right is clicked, the Cross Reference Search screen is displayed to move to the Accident Analysis or Plant screen (See Cross Reference Function)



(5) Tag View

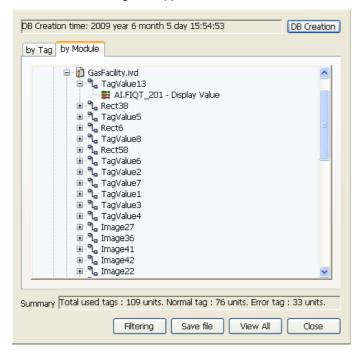
Information on the all of the tags registered in DB and status of tag values are displayed. Once this command is executed, the tag browser dialog box appears.

Chapter 22 Graphic Runtime



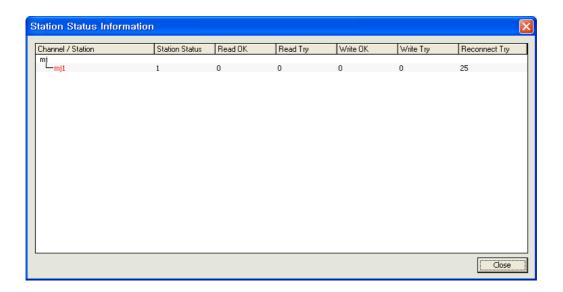
(6) Cross Reference View

The Cross Reference information on the current project is displayed. Once this command is executed, the Cross Reference View dialog box appears. For details, see 'Cross Reference.'



(7) I/O Station Status

The communication status of the currently connected I/O driver is displayed. Once this command is executed, the I/O Station View dialog box appears



22.6 Help



22.6.1 **Help Finder**

It is used to show Help items. Step-by-step commands and instructions on the displayed items are presented. Click the contents button on the Help screen, the corresponding item screen appears.

22.6.2 **Graphic Runtime Information**

It is used to show the version information of the graphic runtime. If this command is executed, version information and copyright message appear.



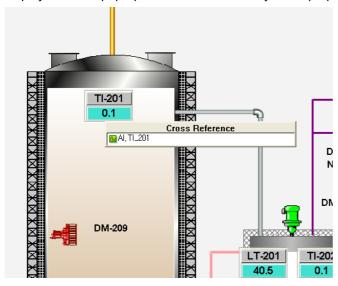
22.7 Cross Reference Function

The Cross Reference is available to use during the runtime.

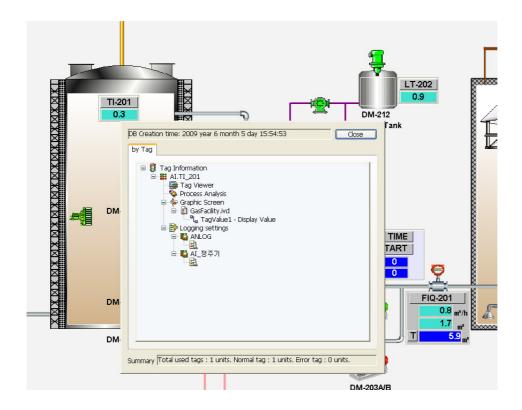
This Cross Reference function is available not only on the plant screen or tag view but also the trend or alarm viewer, and if any alarm occurs during the running, it is possible to move to a relevant screen or trend for analysis by using this Cross Reference function.

(1) Execution on the plant screen

If a graphic object with dynamic properties is clicked with the right button of the mouse (MouseUP), the tag list is displayed on the pop-up screen used for the dynamic properties of the graphic object.



If the relevant tag is double clicked with the left button of the mouse on the Cross Reference pop-up screen,

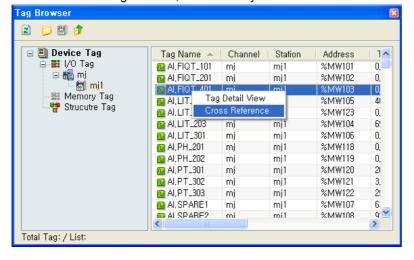


the Cross Reference screen is shown as seen in the figure below and if an item on the screen is double clicked, it moves to a point on the relevant screen and it shows the location of the designated graphic object.

(2) Execution on the Tag View

The Cross Reference of a tag can be inquired on the tag browser.

To execute the tag browser, the user may use the menu or tool bar.



If a tag list is clicked with the right button of the mouse on the tag browser, a pop-up menu appears and at this time, if Cross Reference is selected, the Cross Reference screen is displayed.

Also, if the tag detailed view is selected, the detailed screen of the selected tag is displayed as seen in the figure below. At this time, if Cross Reference button is clicked on the tag detailed screen, the Cross Reference screen is inquired.

Subsequent actions are the same.

Chapter 22 Graphic Runtime

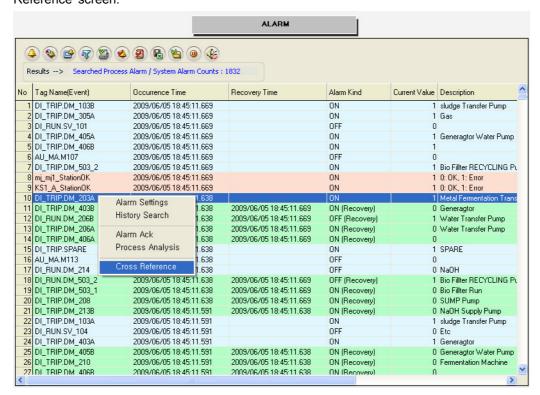
(3) Execution on the Alarm Pop-up Screen

In case that a new alarm occurs, the pop-up window as seen in the figure below appears as the top-level screen and if the 'Cross Reference' button is clicked on the right, the Cross Reference search screen is displayed, and 'Cross Reference' button is clicked with the mouse, the Cross Reference screen can be inquired.



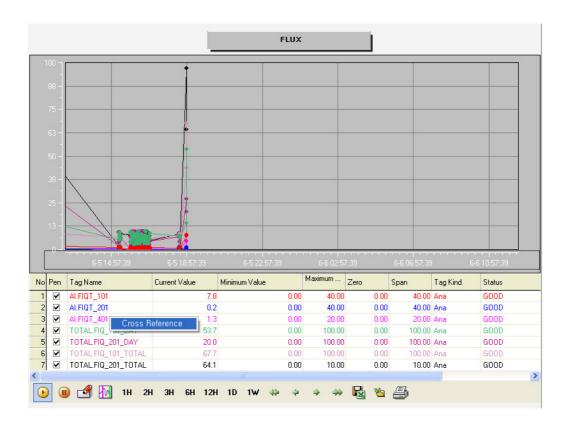
(4) Execution on the Graphic Object

1. Alarm Viewer: If an alarm item is selected on the alarm viewer graphic object and the right mouse is clicked, the pop-up menu below is displayed and 'Cross Reference' can be selected to display the 'Cross Reference' screen.



2. Trend: If a data grid item is selected on the trend graphic object and the right mouse is clicked, the popup menu below is displayed and 'Cross Reference' can be selected to display the 'Cross Reference' screen.

Chapter 22 Graphic Runtime



Graphic Objects can provide various functions that can be implemented just with simple settings such as Alarm Monitoring, Logging Search, Recipe Control and Analog Gauge Display during the real-time running on the graphic screen since they are provided in a form of finished Active X.

In addition, the Graphic Page can be used by inserting external Active X control as a role of ActiveX's container. InfoU Graphic Objects include Alarm Viewer, Trend, List Trend, Recipe Viewer, Gauge Control and Picture Lamp Control and it is planned to add new objects continuously.

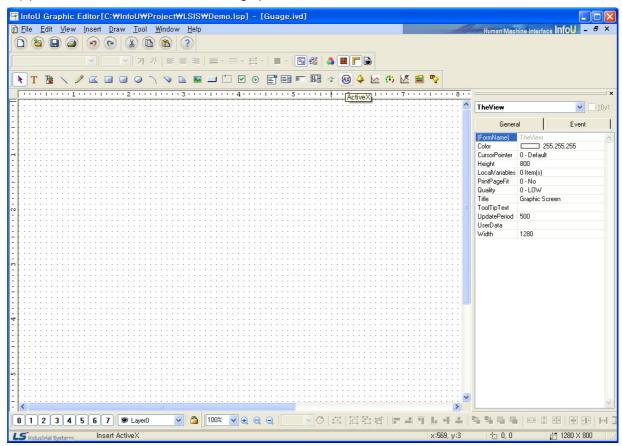
23.1 **How to Use Graphic Object**

23.1.1 **Object Insert**

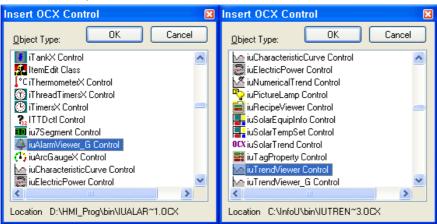
(a) To insert an object immediately, select a relevant shortcut icon on the tool bar.



(b) Select (on the tool menu of the graphic editor.



(c) Select an object.

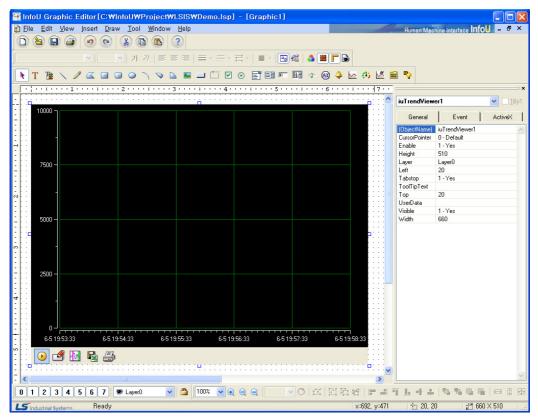


1) iuAlarmViewer: Alarm Viewer iuTrendViewer: Trend Viewer 2)

3) iuNumericalTrend: List Trend Viewer 4) iuRecipeViewer: Recipe Viewer iuArcGaugeX: Gauge Control

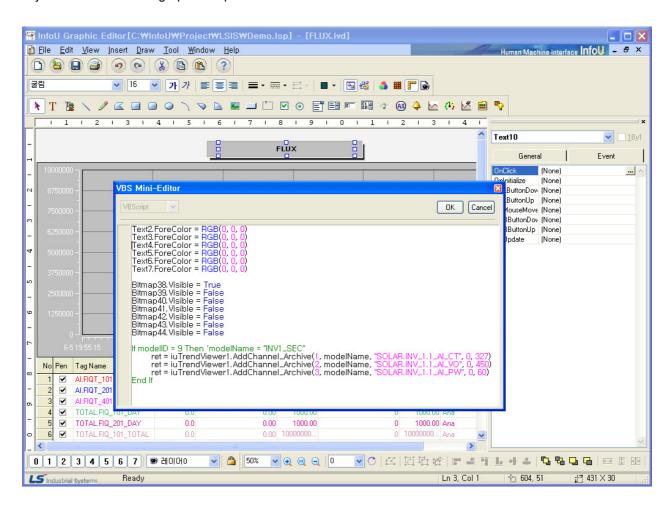
iuPictureLamp: Picture Lamp Control

(d) The selected object is inserted. The default screen is as follows:



23.1.2 Script Use

It is possible to organize a more diverse and simple screen by controlling the method and properties of the object inserted from the graphic script.



23.2 Alarm Viewer

The Alarm Viewer is an Active X module that searches and displays the occurrence details of alarms and events from the alarm server and performs various functions such as Filtering Search, Backup Alarm File Search and Print.

23.2.1 Prerequisite and Environment

Alarm properties can be set up only after the ActiveX module is inserted to the InfoU graphic editor. Alarm properties such as update cycle, alarm color, alarm/event item can be selected. Since those alarm properties have been already set up with default values, they act with those default values if the user does not set up additionally.

The property values are saved and then displayed on the alarm/event monitoring screen when the screen opens in the runtime environment.

The engineering values are inserted in the following order on the alarm viewer.

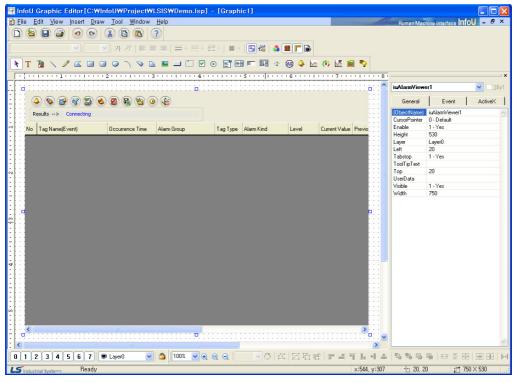


is selected without selecting 600 on the InfoU explorer, those values are immediately inserted.

23.2.2 Screen Configuration

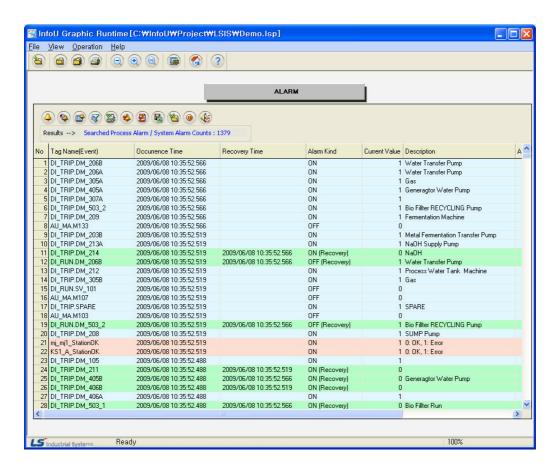
(1) Engineering Screen

The following screen appears after the alarm viewer is inserted.



(2) Runtime Screen

If default trend settings are completed on the engineering screen above and then, the runtime is executed, the screen shows that data values are changed over time.



The splash screen shows real-time alarms and it is for searching alarms, system alarms and events

(a) Tool Bar

Results of tool bar, which consists of necessary functions for operating the viewer, and search are presented in text.

(b) Status Display

The current search status is displayed in text.

(c) Alarm/Event List

Alarm and event details searched from the Alarm Server are displayed in the Grid.

Various colors are displayed to distinguish progress alarm, system alarm, event, recovery alarm and Ack alarm to let the user easily identify them. Colors are as follows and text and background colors can be changed.



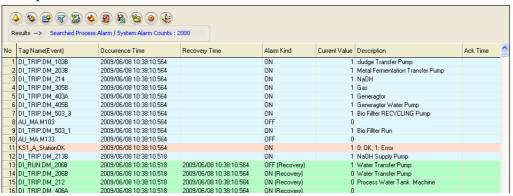
23.2.3 Real-time Function

(1) Real-time Alarm Search (



All of the real-time alarms and events that currently take place are searched.

The number that marks real-time alarms is the value decided from alarm server settings and the number of process alarms and system alarms is twice of the number of tags and 500 is the default value of events. Real-time alarms become invisible in the list through Alarm Recovery and Alarm Ack. That is, the alarm list is updated every second.



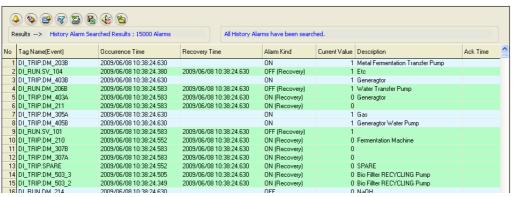
(2) History Alarm Search (



All of the real-time alarms and events that have taken place are searched

All of the alarms that have taken place are displayed in the chronicle order such as occurred alarms, recovered alarms, acknowledged alarms and events including invisible alarms in the real-time alarm list.

The storing reference for history alarms is decided from alarm server settings and the default values are set up to save up to 10,000 process alarms for 3 years, up to 5000 system alarms for 3 years and up to 5000 events for 3 years.



(a) Viewer Settings (



Setting values of the current alarm viewer can be changed.

The setting window will be further explained later.

(b) History Inquiry Setting Screen (

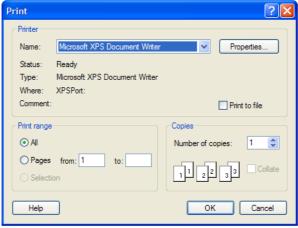


It is used when the user wants to search history alarms after filtering by conditions. The history searching setting window will be further explained later.

(c) Print ()

The screen is printed.

Select a printer on the Print window and click on Properties button to set up whether to print vertically or horizontally. Information is outputted from the selected printer.



(d) Individual Alarm Ack (



The currently selected alarm is acknowledged. For individual alarm ack, double click after selecting an alarm to acknowledge it.

For event and history alarms, Ack is not available.

(e) Page Ack (

All of the alarms in the current page are acknowledged.

For event and history alarms, Ack is not available.

(f) Save as Excel File (

Occurred alarms are saved in Excel files. Press Save button and insert a file name for each alarm to be saved.

(g) Backup File Import (

The alarm backup files saved on the server or other storage devices can be imported and viewed in a

Detailed functions on Load Backup File are described later.

(h) Sound Stop (4)

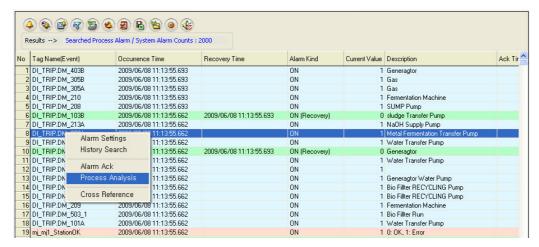
Basically, sound is generated upon the occurrence of alarm, and this function is to stop the sound. If another new alarm occurs, sound is regenerated. To stop sound for the entire alarms, alarm settings shall be modified on the engineering program.

) and Stop (

Buttons to start and stop updating the alarm list in real-time.

(j)Process Analysis

This function is to move the alarm that takes place on the real-time alarm list to the history alarm list for analyzing. Select an alarm from the real-time alarm list and press the right side of the mouse to display the following menu. If Process Analysis is selected here, it moves the relevant alarm, which will be highlighted in red, to the history alarm list for analyzing.

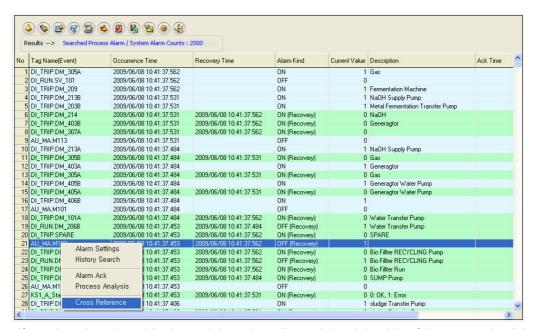


The following screen shows the result of moving to the history alarm list.

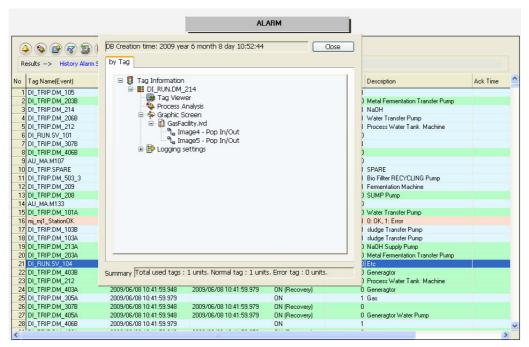


(k) Cross Reference

Cross Reference makes the screen move to the graphic screen to which the selected alarm belongs.



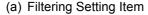
If an alarm is selected in the real-time alarm list and the right side of the mouse is clicked to select Cross Reference, the following Cross Reference screen is displayed. It is possible to move to the place the user wants to go.

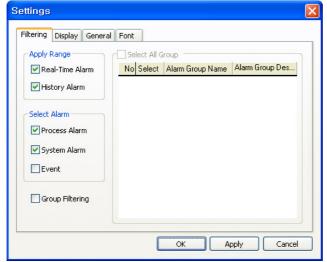


23.2.4 Settings

If the right side of the mouse is clicked on Alarm Viewer ocx and the 'Properties' menu is selected, the setting window appears.

The same setting screen is used for the engineering screen and the run-time screen. In addition, once property values are saved on the engineering screen, those values will be continuously valid however, if those property values are changed on the runtime screen, the changed values are applied only to the runtime screen. The changed values are cancelled if the runtime stops and the property values set up on the engineering screen are applied.





For filtering setting items, Select Alarm, Group Filtering and other menus to select an application range of the viewer are available.

- 1) Apply Range: The user may decide whether to apply the selected alarm item or the group filtering to Real-Time Alarm or History Alarm.
- 2) Select Alarm: The user may decide whether to apply the filtering settings to Process Alarm, System Alarm or Event.
- 3) Group Filtering: The user may decide whether to apply Group Filtering or not and decide which group to see.

Settings Filtering Display General Font Select Items Alarm Group 1900 Tag Name(Event) 2300 Tag Type 900 Occurrence Time 2300 Recovery Time Alarm Kind 450 2300 1500 Previous Value 1100 TOP Message User Ack 2000 Current Value 1100 >> 900 Description 3000 Unit 500 [->] Ack Time 2000 Up User Define 1 1100 On/Off Text 2000 User Define 2 <-Down User Define 3 1100 Alarm Maintain Time << ВТ ОК Apply Cancel

(b) Alarm Display Item

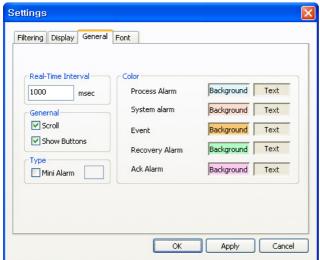
This function is to select an alarm item to display on the alarm viewer.

For Alarm Display Items, total 18 items are available such as Tag Name(Event), Occurrence Time, Alarm Group(Occurrence Place), Tag Type, Alarm (Event) Kind, Level, Current Value (Status), Previous Value(Status), Message, Description, Recovery Time, User Ack, Ack Time, Unit, User Define 1, User Define 2, User Define 3 and User Define 4.

Numbers next to items indicate the column size of each item and they are changeable. Double click on the item to change and insert a new size.

- 1) >>: Move the marked item from the entire list to the selected list.
- <<: Move the unmarked item from the entire list to the selected list 2)
- 3) ->: Move from the entire list to the selected list.
- 4) <-: Move from the selected list to the entire list.
- 5) TOP: Move the marked location on the selected column from the selected list to the top left.
- 6) Up: Move the marked location on the selected column from the selected list to the next cell on the left side.
- 7) Down: Move the marked location on the selected column from the selected list to the next cell on the right side.
- BT: Move the marked location on the selected column from the selected list to the end on the right side.

(c) General



1) Real-Time Interval

Enter a time to update the viewer. Alarms are searched on the relevant severs at the entered time and a list is displayed.

2) Scroll

Select whether to scroll or not on the viewer.

3) Show Buttons

Select whether to show/hide tool bars on the viewer.

4) Mini Alarm

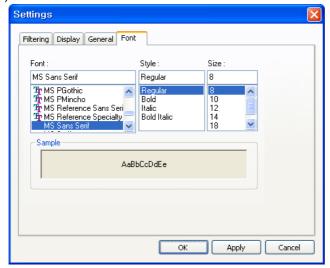
Select whether to use mini alarms.

If mini alarms are used, tool bars or scroll do not exist.

5) Color

Select each background color and text color of Process Alarm, System Alarm, Event, Recovery Alarm and Ack Alarm.

(d) Font



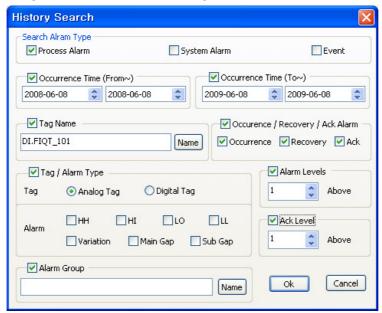
Decide Font, Style and Size to be displayed on the alarm viewer list.

23.2.5 Filtering

As a function used when the user wants filter history alarms with several conditions and searches only the history alarms that meet those conditions, it proceeds filtering as follows:

- (a) Select items to filter and search with the input condition 'And' condition.
- (b) The alarms that meet the search conditions are displayed on the alarm list.
- (c) The timer stops automatically.
- (d) To return to the real-time search, click on Real-Time Alarm Search button.

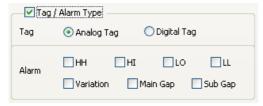
The Filtering Search Screen is as following:



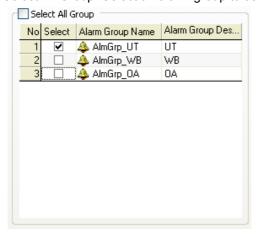
- Search Alarm Type/Event: Select one among Process Alarm, System Alarm and Event to search.
- Occurrence Time (From): Only the alarms that have occurred since the occurrence time can be 2) searched.
- Occurrence Time (To): Only the alarms that had occurred before the occurrence time can be searched.
- 4) Tag Name: Designate the tag name to search. Click on Name button to display the tag browser screen to select a tag from it.



5) Tag/Alarm Type: Select either Analog Tag or Digital Tag and then, select an alarm kind that fits to it to search. If Digital Tag is selected, the following screen appears.



6) Select All Group: Select an alarm group to search.



- 7) Alarm Occurrence/Recovery: Select either Alarm Occurrence or Recovery and then, proceed searching.
- 8) Alarm Level: Only the alarms higher than the entered alarm level are searched. Levels 1 through 99 are available.
- 9) Ack Level: Only the alarms higher than the entered tag priority are searched. Levels 1 through 99 are available.

23.2.6 Other Functions

(1) Alarm Ack

As a function to acknowledge (Ack) the occurred alarm, it is available to acknowledge not only by alarm but also by page unit.

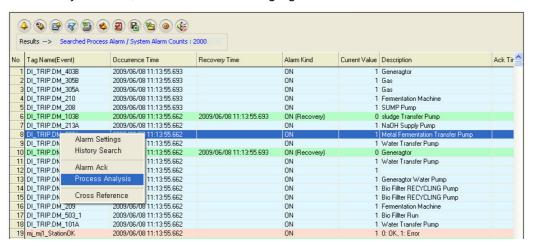
For individual alarm Ack, double click after selecting an alarm or press [Ack] button on the tool bar. It is also available to press the right button of the mouse. For Event and History Alarm, Ack is not available.

For Page Ack, press [Page Ack] button on the tool bar to acknowledge all of the alarms seen in the current page.

Process Analysis (2)

This function is to move the alarm that takes place on the real-time alarm list to the history alarm list for analyzing. Process Analysis is useful when viewing the alarms before and after the occurred alarm in the realtime.

Select an alarm on the real-time alarm list and press the right side of the mouse to display the following menu. Select Process Analysis from the menu to move the relevant alarm to the history alarm list for analyzing. Once moved to the history alarm list, the selected alarm is highlighted in red for convenient identification.

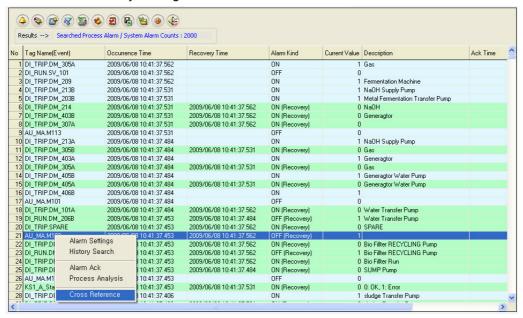


The following screen shows the result of moving to the history alarm list.

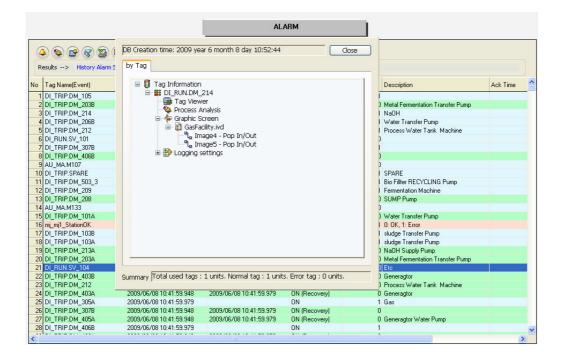


(a) Cross Reference

This function is to make the screen move to the graphic screen to which the selected alarm belongs. This function helps to immediately identify at which object the current alarm takes place and analyze the current situation by moving to the screen.

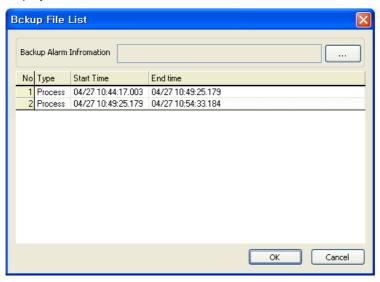


Select an alarm on the real-time alarm list and press the right side of the mouse to display the following menu. Select Cross Reference from the menu to display the following Cross Reference screen. If the user selects one among the screens to which the occurred alarm belongs, it moves to the relevant screen. Select a screen and double click the mouse to move to the screen.

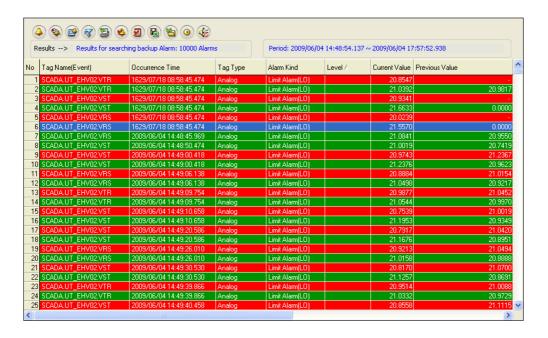


(b) Backup Alarm Import

If Backup Alarm Import button is selected, the following screen appears. The list shows backup files on the saving path. If a list is selected and Ok button is pressed, a list of the alarms in the backup files is displayed.



The other way is to select a file directly by pressing File Selection button and then selecting a file on the location the user wants.



23.2.7 How to Use Script

(1) Function List

Function Name	Туре
AckPageAlarm	LONG AckPageAlarm(void)
ExportToExcel	LONG ExportToExcel(void)
SetAlarmList	LONG SetAlarmList(LONG alarmKind, LPCTSTR str)
SetHistoryList	LONG SetHistoryList(LONG alarmKind, LPCTSTR str)
SetAlarmGroupList	void SetAlarmGroupList(LPCTSTR strGroupList)
SetFilteringOption	LONG SetFilteringOption(USHORT almType, LPCTSTR startTime, LPCTSTR endTime, LPCTSTR tagName, SHORT tagType, SHORT almKind, SHORT almOccurType, SHORT almGrade, SHORT ackGrade, LPCTSTR almGroup)
ShowConfigDlg	void ShowConfigDlg(void)
ShowFilteringDlg	LONG ShowFilteringDlg(void)
StopAlarm	LONG StopAlarm(LONG bFlag)
StopSound	LONG StopSound(void)
PrintList	LONG PrintList(void)

(2) Function Description

AckAllAlarm Function

Description

It is used when the user wants to acknowledge the entire alarms seen on the current page of the alarm viewer.

Syntax

LONG AckPageAlarm(void)

Reference

To acknowledge the entire alarms on the page, AckPageAlarm ()

ExportToExcel Function

Description

It is used when the user wants to export the alarms taken place on the alarm viewer to Excel.

Syntax

LONG ExportToExcel(void)

Reference

To export the occurred alarms to Excel, ExportToExcel ()

SetAlarmList Function

Description

This function is to allow the user to select an alarm kind to make it appear on the relevant list. With this function, the user can see the real-time alarm list by kinds in the early stages.

LONG SetAlarmList(LONG alarmKind, LPCTSTR str)

alarmKind – Enter an alarm kind. Insert the following values according to alarm kinds.

- 0: All
- 1: Process Alarm
- 2: System Alarm
- 3: Process Alarm + System Alarm
- 4: Event
- 5: Process Alarm + Event
- 6: System Alarm + Event

str - enter empty value ("").

Reference

To see a process alarm list, SetAlarmList(1, "")

SetHistoryList Function

Description

This function is to allow the user to select an alarm kind to make it appear on the relevant list. With this function, the user can see the history alarm list by kinds in the early stages.

Syntax

LONG SetHistoryList(LONG kind, LPCTSTR str)

kind - Enter an alarm kind. Insert the following values according to alarm kinds.

- 0: All
- 1: Process Alarm
- 2: System Alarm
- 3: Process Alarm + System Alarm
- 4: Event
- 5: Process Alarm + Event
- 6: System Alarm + Event

str - enter empty value ("").

Reference

To see a process alarm list out of history alarm lists, SetHistoryList(1, "")

SetAlarmGroupList Function

Description

This function is to allow the user to search only the alarm groups he/she wants and make them appear on the relevant list.

With this function, the user can see the list by alarm groups in the early stages.

Syntax

void SetAlarmGroupList(LPCTSTR strGroupList)

strGroupList – Insert the alarm groups in a form of string values. Be careful about distinguishing one alarm group from the other with ";" and not forgetting to insert ";" at the end.

Reference

To see AlarmGroup_1 and AlarmGroup_2, SetAlarmGroupList ("AlarmGroup_1;AlarmGroup_2;")

SetFilteringOption Function

Description

This function is to filter and see alarms by kind or group with several filtering options.

This function applies only to history alarms. It does not apply to real-time alarms.

Syntax

LONG SetFilteringOption (USHORT almType, LPCTSTR startTime, LPCTSTR endTime, LPCTSTR tagName, SHORT tagType, SHORT almKind, SHORT almOccurType, SHORT almGrade, SHORT ackGrade, LPCTSTR almGroup)

almType – Enter an alarm kind. Insert the following values according to alarm kinds.

- 0 : All
- 1: Process Alarm
- 2: System Alarm
- 3 : Process Alarm + System Alarm
- 4 : Event
- 5: Process Alarm + Event
- 6: System Alarm + Event

startTime – Write a start time to search. Write a string value in a form of "2007/08/17."

Enter empty value ("") if the user wants to search regardless of the start time.

endTime – Write an end time to search. Write a string value in a form of "2007/08/17." Enter empty value ("") if the user wants to search until now.

tagName – Write a tag name to search. Enter empty value ("") to search the entire tags.

tagType - Enter a tag type. Enter 1 for analog tags and 2 for digital tags.

almKind - Enter 0.

almOccurType – A parameter to select one from occurrence, recovery or Ack of alarm and search Enter the following value.

- 1: Occurrence / Recovery / Ack
- 2: Occurrence / Recovery
- 3: Occurrence / Ack

- 4: Occurrence
- 5: Recovery / Ack
- 6: Recovery
- 7 : Ack

almGrade - Enter an alarm grade to search.

ackGrade - Enter an ack grade to search.

almGroup - Enter an alarm group to search.

If the user wants to search multiple groups, he/she needs to distinguish one group from the other with ";".

If empty value or 0 is entered for each parameter, searching is conducted regardless of the conditions of the relevant parameter.

Reference

To search the recovered alarms and ack alarms among the alarms that took place in Alarm Group1 from August 15 through 17, 2007,

SetFilteringOption (0, "2007/08/15", "2007/08/17", "", 0, 0, 5, 0, 0, "Alarm_Group1")

ShowConfigDlg Function

Description

The configuration setting window of the alarm viewer is shown.

void ShowConfigDlg(void)

Reference

To show the configuration setting window, ShowConfigDlg()

ShowFilteringDlg Function

Description

The filtering search window of the alarm viewer is shown.

Syntax

LONG ShowFilteringDlg(void)

Reference

To show the filtering search window, ShowFilteringDlg ()

StopAlarm Function

Description

It is used when stopping for a while or restarting the alarm viewer.

Syntax

LONG StopAlarm(LONG bFlag)

bFlag – To stop the alarm: True

– To start the alarm: False

Reference

To stop the alarm, StopAlarm(True)

To restart the alarm, StopAlarm(False)

StopSound Function

Description

It is used when stopping alarm sound.

Syntax

LONG StopSound(void)

Reference

To stop alarm sound, StopSound ()

PrintList Function

Description

It is used when printing the alarms that take place on the alarm viewer.

Syntax

LONG PrintList(void)

Reference

To print the occurred alarms, PrintList ()

23.3 **Trend Viewer**

This Trend Viewer is an Active X module that displays charts monitoring online and logging data on the tags registered in the Trend.

As a screen to monitor changes in tag values over time, this Trend Viewer is an Active X module that displays online tag data graphs as well as logging data graphs stored in the logging server.

In addition, this Trend can be used in analyzing changes in logging data.

23.3.1 **Prerequisite and Environment**

Trend properties can be set up only after the ActiveX module is inserted to the InfoU graphic editor.

Channel assignment, tag assignment, X axis/Y axis value input and other tag properties to be monitored are designated in advance as Trend properties

Channels or tags are not basically assigned to the trend status inserted in the graphic editor. It is essential to display the Property Window before setting up the property values needed to run the Trend.

After the property values are saved, the Trend Monitoring Screen appears upon the opening of the screen under the Run time environment.

The engineering values are entered in the following order on the Trend viewer.

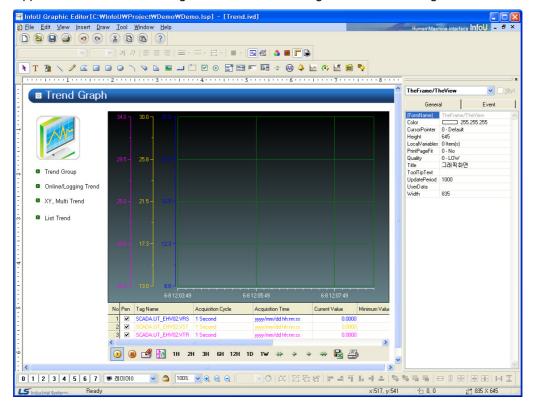


is selected from the tools, immediate inserting can be performed.

23.3.2 Screen Configuration

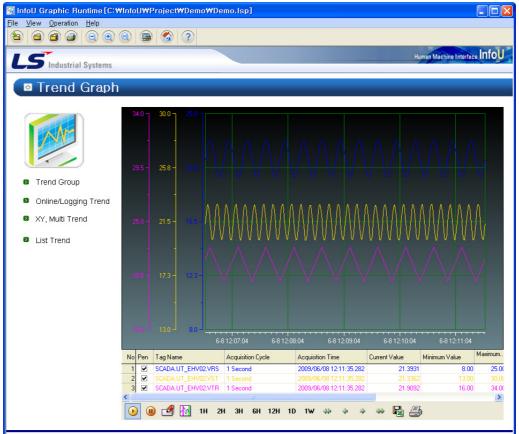
(1) **Engineering Screen**

If Trend is inserted and a tag to be monitored is selected and then settings are all completed, the following screen appears. The screen in this figure shows that three tags are selected and gradation effect is applied.



(2) Runtime Screen

If default trend settings are completed on the engineering screen above and then, the runtime is executed, the screen shows that data values are changed over time.

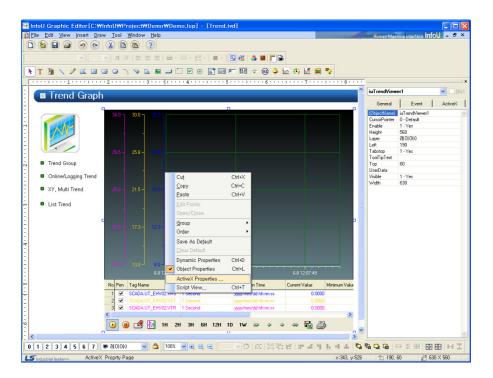


23.3.3 Settings

If the right side of the mouse is clicked on the Trend and the 'Active X Properties' menu is selected, the setting window appears.

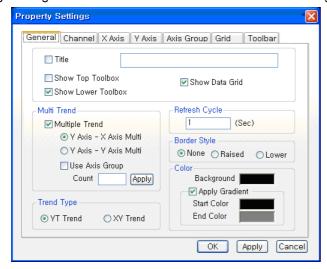
The same setting screen is used for the engineering screen and the runtime screen. But, the runtime screen does not have Grid Settings and Toolbar Selection functions.

In addition, once property values are saved on the engineering screen, those values will be continuously valid however, if those property values are changed on the runtime screen, the changed values are applied only to the runtime screen. The changed values are cancelled if the runtime stops and the property values set up on the engineering screen are applied.



General (1)

The Engineering Screen and the Runtime Screen have the same general functions.



(a) Title

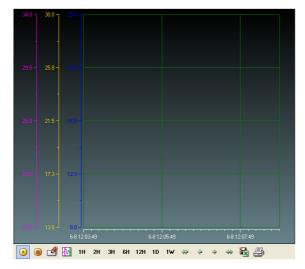
Designate a title of the trend.

(b) Show Top Toolbox/Show Lower Toolbox

Set up whether to display Trend Toolbox on the top or the bottom during the running. Both can not be selected at the same time and it is allowed neither of them is selected. 'Show Lower Toolbox' has been selected as the default value.

(c) Show Data Grid

Select whether to show or hide data grid during the running. The following figure shows the screen when 'Show Data Grid' is not selected. 'Show Data Grid' has not been selected as the default value.

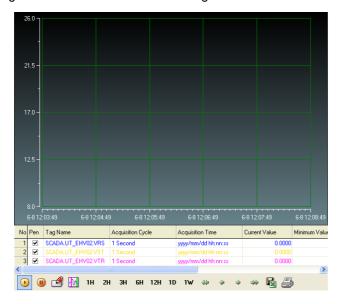


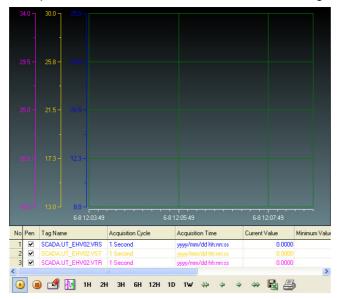
1) Multiple Trend

Select 'Multiple Trend.'

In general trends, the axis that takes the lowest value among the minimum values as the min value and the highest value among the maximum values as the max value is commonly used. However, if 'Multiple Trend' is selected, any axis can be used individually.

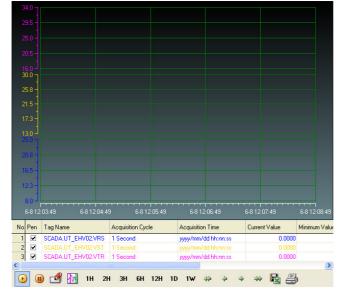
A general trend screen is as following:





A Multiple Trend screen that selects x-axis is as following:

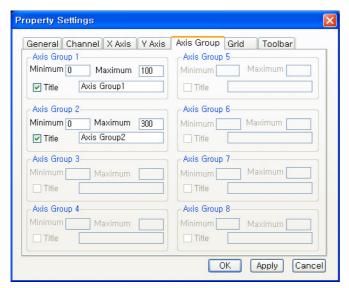
A Multiple Trend screen that selects y-axis is as following:



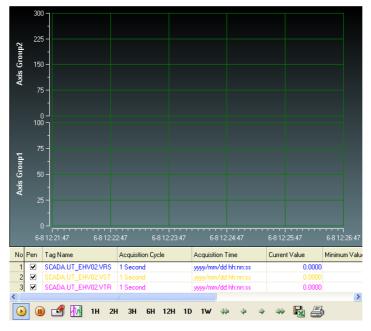
Multiple trends can use Axis Group.

Axis Group is a function to re-divide several channels into desired groups to view them. It is used as following:

- . Check 'Use Axis Group.'
- . Input the number and press 'Apply' button.
- . Once it moves to 'Axis Group' tap upon the 'Apply' button, Axis Groups as much as the inputted number are activated. Input the min value and max value here.



If it is set up that Channel 1 and 2 belong to Axis Group 1 and Channel 3 belongs to Axis 2, the following screen appears.



- 2) Refresh Cycle
 - Designate a refresh cycle. One second has been set up as the default value.
- 3) Color
 - Select a background color and set up 'Apply Gradient.' If 'Apply Gradient' is selected, the background color is not applied. In addition, it needs to set up 'Start Color' and 'End Color.'
- 4) Border Style
 - Select a Border Style for the relevant trend.
- 5) Trend Type
 - Select either YT Trend or XY Trend.
 - If YT Trend is selected, x axis indicates time and Y axis shows tag values while if XY Trend is selected, both X and Y axis consist of tag values.

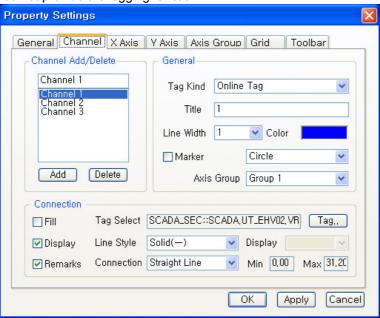
(2) Channel

The Engineering Screen and the Runtime Screen have the same Channel functions.

Up to 20 channels are provided in the Trend. There is at least one channel and it is available to designate Tag Kind, Title, Color, Marker, Tag, Line Style and Connection for each channel.

If one channel is an online tag, the rest channels have to be online tags. Likely, if one channel is a logging tag, the rest channels have to be logging tags.

Also, if any logging data needs to be saved or searched, the logging tag should be selected and used. Online tags do not provide the logging function.

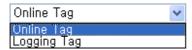


(a) Channel Add/Delete

A channel is added or deleted. At least one or more than one but not more than 20 channels can be selected

(b) Tag Kind

Select either Logging Tag or Online Tag.



(c) Title

Designate a title of the channel.

(d) Line Width

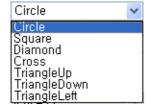
Select a Line Width of the channel. Numbers O through 4 are available.

(e) Color

Decide color of the channel.

(f) Marker

As a point to indicate the tag value of the relevant channel, the following shapes are available.



(g) Tag Select

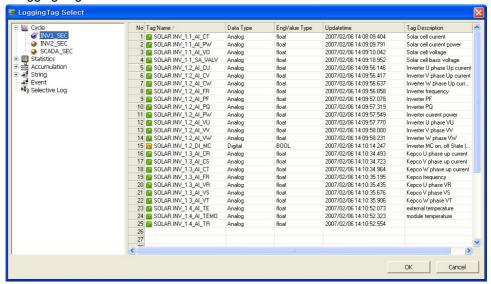
Assign a tag to the channel. One tag can be assigned to one channel.

If a logging tag is selected, the logging tag browser where the associated logging group is set up is called while the online tag browser is called for an online tag.

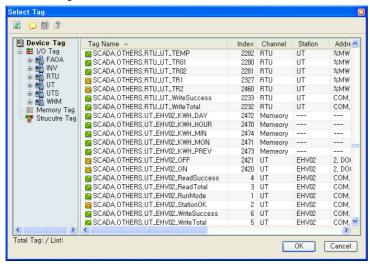
Be cautious that logging data can be searched only when a logging tag is selected.

If a tag is selected, the tag's minimum value and maximum values are displayed. These min/max values apply to Y-axis values for Y axis settings.

* 'Logging Tag Select' screen



* 'Online Tag Select' screen



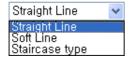
1) Line Style

Select one among Solid, Dash and Dot.



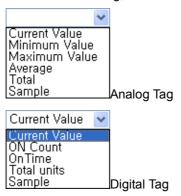
2) Connection

Select one among Straight Line, Soft Line and Staircase Type.



3) Display

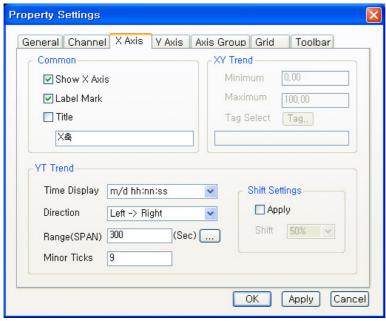
This function becomes active only when a statistics tag is selected. For analog tags, select one value among current value, min value, max value, average value, sum and sample and for digital tags, select one value among current value, On count, On Time, Total count and sample. If the tag is not a statistics tag, it becomes inactive.



(3)X Axis

The Engineering Screen and the Runtime Screen have the same X axis functions.

If XY Trend is selected on the General tap, setting items for YT Trend become inactive while those for XY Trend become inactive if YT Trend is selected.



(a) Common

- 1) SHOW X AXIS: Decide whether to show or hide X Axis.
- Label Mark: Select whether to mark labels on X axis or decide a title for identification.
- Title: Decide whether to mark a title on X axis.

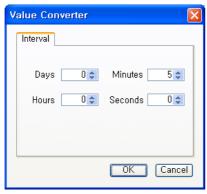
(b) YT Trend

1) Time Display: Decide one among various Time Display types.



- Direction: Select either 'Left->Right' or 'Right->Left.'
- 3) Range (Span): Save a range of X-axis values. Input a value directly or select one from the range window.

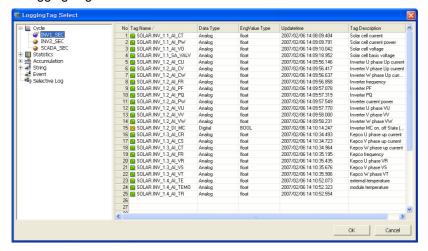
Input time and the unit is second.



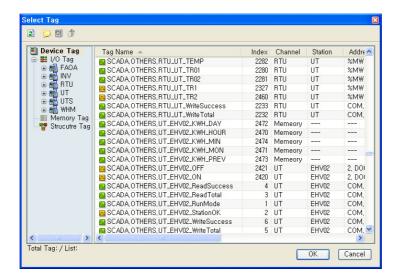
4) Shift Settings: The Shift function refers to moving the current value to the left as much as the shift setting value of x axis if the graph goes to the end. If the user wants this function, he/she needs to set up Shift and press 'Apply' button. Available values are from 0 through 100 %. Shift has not been set up as the default value and it starts after drawing a range of X axis.

(c) XY Trend

- 1) Minimum/Maximum: Specify the min/max value of the relevant tag to be marked on X-axis. Only one tag can be on X axis.
- 2) Tag Select: Assign a tag to the channel. If a logging tag is selected, the logging tag browser where the associated logging group is set up is called while the online tag browser is called for an online tag. Logging data can be searched only when a logging tag is selected.
 - * 'Logging Tag Select' screen

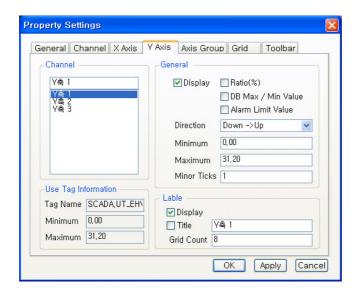


* 'Online Tag Select' screen



(4)Y Axis

The Engineering Screen and the Runtime Screen have the same Y axis functions.



(a) Channel

Show the channels that are assigned to Y axis.

(b) Display

Select either show/hide Y axis.

(c) Ratio (%)

Display Y axis value in Ratio not in tag value. If Display in Ratio is selected for even one Y axis, Y axis is displayed in percentage unless Multi Trend has selected. Also, if this item is selected, moving with the mouse becomes impossible along Y axis.

(d) DB Max/Min Value

DB Max/Min Value is used instead of user's max/min value.

(e) Alarm Limit Value

If this item is selected, HH value and LL value of tag alarm are drawn in lines and Alarm Limit Value is displayed.

HH value and LL value are not visible.

(f) Direction

Select either 'Down->Up' or 'Up->Down.'

(g) Minimum/Maximum

Set up max/min value needed to show Y axis. Tag's max/min value has been set up as the default value.

(h) Use Tag Information

Show tag information of the channel, which has been set up on Channel tap. Tag name and tag's max/min value are displayed.

(i)Display

Select whether to display label and how to mark the title.

(j)Grid Count

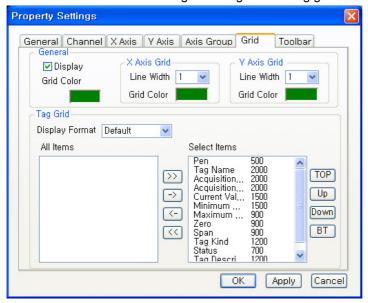
Grid Count is the number of lines for Y axis values visible on the screen. This value is adjusted to display max/min value. But, this value may not be adjusted in case that Y axis is too narrow to display channels.

(5) Axis Group

This tap becomes active when 'Multi Trend' is selected on the General tap and Axis Group is applied. Input the min/max value and title if necessary for each Axis Group.

(6) Grid

This Grid setting function is only available on the Engineering Screen. It is not displayed on the Runtime Screen. It consists of General for color settings and Tag Grid for tag grid item and order settings.



(a) General

- 1) Grid Color: Select a color of grid border.
- 2) X Axis Grid: Select Line Width and Grid Color for X Axis Grid.
- 3) Y Axis Grid: Select Line Width and Grid Color for Y Axis Grid.

(b) Tag Grid

Select alarm items to display on the viewer.

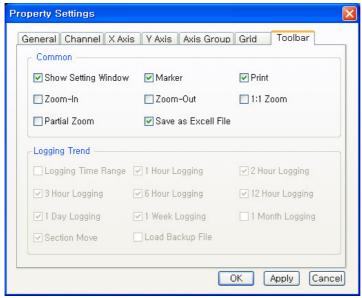
'Select Items' include Pen, Tag Name, Acquisition Cycle, Acquisition Time, Current Value, Minimum, Maximum, Zero, Span, Tag Kind, Status and Tag Description.

Numbers next to items indicate the column size of each item on the Trend Viewer and they are changeable. Double click on the item to change and insert a new size.

- 1) >>: Move the marked item from the entire list to the selected list.
- 2) <<: Move the unmarked item from the entire list to the selected list.
- 3) ->: Move from the entire list to the selected list.
- 4) <-: Move from the selected list to the entire list.
- 5) TOP: Move the marked location on the selected column from the selected list to the top left.
- 6) Up: Move the marked location on the selected column from the selected list to the next cell on the left side.
- 7) Down: Move the marked location on the selected column from the selected list to the next cell on the right side.
- BT: Move the marked location on the selected column from the selected list to the end on the right side.

(7)Toolbar

Select Toolbar (menus).



(a) Start ()

This is a common function that applies to both online trends and logging trends,

Once trend monitoring starts, data is renewed every second and 5 minutes is set as the time for X axis. To view the current screen again during the logging search, press this button and the current time screen is displayed.

This Start button can not be selected as the default value.

(b) Stop ()

Trend Monitoring stops. This function is available only for logging trends. That is, this function exists only when logging tags are selected and online trends do not have this function.

This Stop button can not be selected as the default value like the Start button

(c) Show Setting Window ()



Setting values of the current trend can be changed. This is a common function that applies to both online trends and logging trends.

This Setting Window has been described before.

(d) Zoom-in ()

The current trend screen is enlarged twice to search data. Zoom-in makes both of X axis and Y axis enlarged twice. This is a common function that applies to both online trends and logging trends.

(e) Zoom-out (W)

The current trend screen is reduced twice to search data. Zoom-in makes both of X axis and Y axis This is a common function that applies to both online trends and logging trends.

(f) 1:1 Zoom (

This function is used to return the enlarged or reduced trend screen into its original size -100% screen. This is a common function that applies to both online trends and logging trends.

(g) Partial Zoom (

The trend screen is partly zoomed in/out to search data. A part of the trend screen is dragged with the mouse and it becomes enlarged as soon as the mouse is released. At this time, both X axis and Y axis are enlarged.

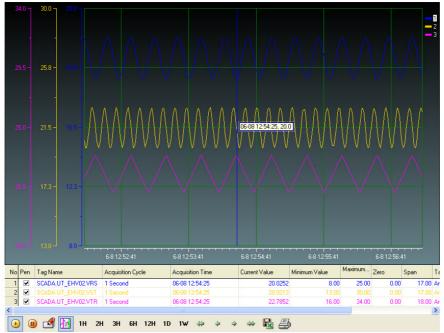
This is a common function that applies to both online trends and logging trends.

(h) Marker (

If Marker button is pressed, a yellow marker line is displayed and values of channels are also listed on the right screen. If the right button of the mouse is pressed on the marker, the channel value can be seen in a tool tip type. To see another channel value, select it on the right screen. At this time, the marker color is the same with the channel color.

To prevent the bar appeared on the screen from re-appearing, click on Marker button one more time. This is a common function that applies to both online trends and logging trends.





(i)Save as Excel File (

Convert and save the current trend being monitored into a csv file.

The csv file can be opened in Excel and it can be viewed as normal text.

(j)Print (4)

Print the current trend being monitored. In printing, the background screen is changed in white color.

(k) Logging Trend (Logging Time Range) ()
Input logging conditions (period and sampling conditions) to search past data. The logging search will

be further explained later. This function is available only for logging trends.

(I)1 Hour Logging (1H)

View 1 hour logging data including the current time. For example, if it is 10:20, the logging data between 10 and 11o'clock are viewed. This function is available only for logging trends.

(m) 2 Hour Logging (2H)

View logging data in the time range where the current time belongs to by dividing 24 hours into 0~2 o'clock, 2~4 o'clock, 4~6 o'clock, ..., 20~22 o'clock and 22~midnight. For example, if it is 10:20, the logging data between 10 o'clock and noon are viewed. This function is available only for logging trends.

(n) 3 Hour Logging (3H)

View logging data in the time range where the current time belongs to by dividing 24 hours into 0~3 o'clock, 3~6o'clock, 6~9o'clock, ..., 18~21 o'clock and 21~midnight. For example, if it is 10:20, the logging data between 9 o'clock and noon are viewed. This function is available only for logging trends.

(o) 6 Hour Logging (6H)

View logging data in the time range where the current time belongs to by dividing 24 hours into 0~6 o'clock, 6~12o'clock, 12~18o'clock and 18~midnight For example, if it is 10:20, the logging data between 6 o'clock and noon are viewed. This function is available only for logging trends.

(p) 12 Hour Logging (12H)

View logging data in the time range where the current time belongs to by dividing 24 hours into 0~noon and noon~ midnight. For example, if it is 10:20, the logging data between 0 o'clock and noon are viewed. This function is available only for logging trends.

(q) 1 Day Logging (1D)

View today's logging data. This function is available only for logging trends.

(r) 1 Week Logging (1W)

View this week's logging data. The period is from Monday through the midnight of Sunday. This function is available only for logging trends.

(s) 1 Month Logging (^{1M})

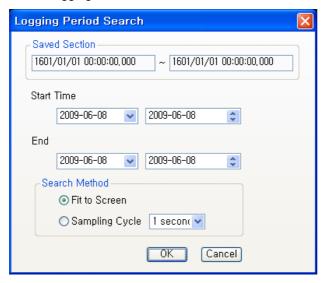
View this month's logging data This function is available only for logging trends.

(t) Load Backup File ()

The trend backup files saved on the server or other storage devices can be imported and viewed in a list. Detailed functions on Load Backup File are described later.

23.3.4 Logging Search

This is a function to select logging search conditions. The following Logging Period Search window is displayed upon the click on Logging Search button.



'Saved Section' shows the starting time and the end time to save the entire data and this is referred when deciding the start time and the end time.

'Search Method' consists of 'Fit to Screen' and 'Sampling Cycle.'

'Fit to Screen' is to search and show data in the logging search range, which fit to the current screen.

'Sampling Cycle' is to search and show data in the logging search range by sampling to fit to the selected time.

That is, if '1 second' is selected, data in the relevant search range are saved and loaded every second.

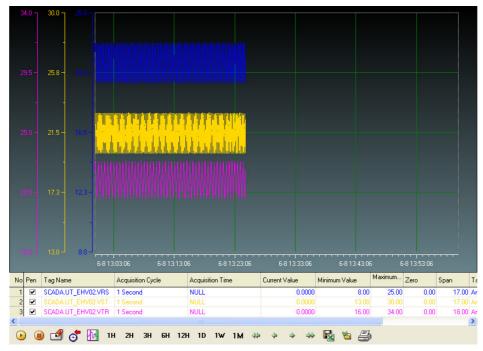
If '5 second' is selected, data are saved and loaded every 5 seconds and if '1 minute' is selected, data are saved and loaded every minute.

The first figure below shows the result of using 'Fit to Screen' method to search and the second figure shows the result when saving and loading data every second. 'Fit to Screen' has been set up as the default value.

* Logging screen to load data to 'Fit to Screen'



* Logging screen to load the saved value every second.



* Difference between logging search and online search

For online search, the mouse can be moved along the axis. Y axis can be selected with the mouse and moving is available with mouse wheel, mouse drag or keyboard arrow. For logging search, any axis can not be moved and only data in the search range are displayed.

23.3.5 Others

Save as Excel File (1)

'Save as Excel File' is a function to save data on the current screen in a csv files.

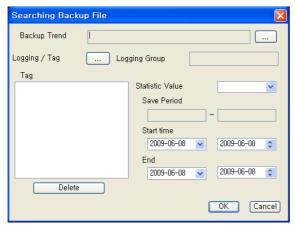
(2) Print

The current trend screen is printed. Basically, the screen is changed from black to white when printing.

Searching Backup File (3)

This function is to load stored trend backup files.

Press the relevant button on Toolbar tap to display the following screen.



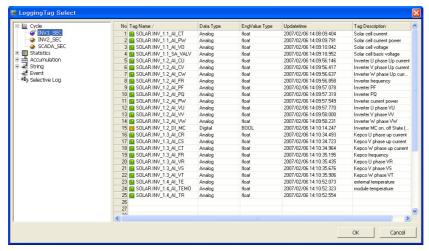
(a) Backup Trend

Select a trend backup file. The file has extension '.bak.'

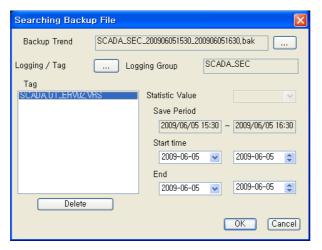
(b) Logging/Tag

Select a logging group and tags. Only one logging group can be selected but, selecting multiple tags is allowed. The selected tags are displayed in the list box. The saving period of the relevant tag appears in 'Save Period.'

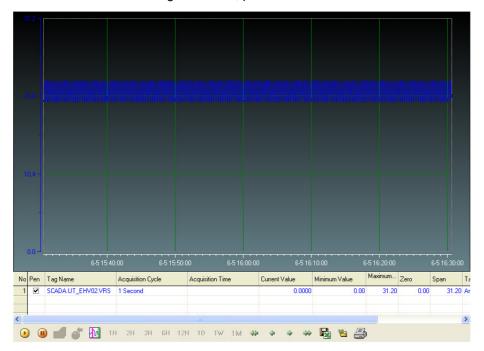
'Start Time' and 'End' are used to select the time to search.



This screen shows the selected logging group and tags.



After data are searched on the search screen, those data can be moved by using Section Search function. To return to the original screen, press Start button.



(c) Axis Move

For online monitoring trends, axis can be moved with the mouse. X axis or Y axis can be selected with the mouse and moving is available with mouse wheel, mouse drag or keyboard arrow. For multiple trends, Y axis for each trend can be moved individually. For logging search, any axis can not be moved.

23.3.6 How to Use Script

Function List (1)

Function Name	Туре		
AddChannel_Archive	LONG AddChannel_Archive(LONG channelNo, LPCTSTR modelName,		
	LPCTSTR tagName, DOUBLE minValY, DOUBLE maxValY)		
AddChannel_RealTime	LONG AddChannel_RealTime(LONG channelNo, LPCTSTR tagName,		
	DOUBLE minValY, DOUBLE maxValY)		
AddTrendChannel	LONG AddTrendChannel(LPCTSTR dataSource, LONG channelNo,		
	LONG modelID, LPCTSTR modelName, LONG tagID, LPCTSTR		
	tagName, LONG minValueY, LONG maxValueY)		
ApplyTrend	void ApplyTrend(void)		
ApplyProperty	LONG ApplyProperty(VARIANT_BOOL bFlag)		
DeleteTrendChannel	LONG DeleteTrendChannel(LPCTSTR channelStr, LONG channelNo)		
LoadHistoryTrend	void LoadHistoryTrend(LONG year, LONG month, LONG week, LONG		
	day, LONG hour, BOOL bFlag)		
LoadPeriodTrend	LONG LoadPeriodTrend(LONG days, BOOL bFlag)		
Load_ArchivePeriod_Tre	LONG Load_ArchivePeriod_Trend(LONG type, LONG days,		
nd	VARIANT_BOOL bFlag)		
LoadPeriodDataTrend	LONG LoadPeriodDataTrend(LPCTSTR strFrom, LPCTSTR strTo, LONG		
	bSample, LPCTSTR bFlag)		
ShowTrendProperty	void ShowTrendProperty(void)		
SetTrendGridVisible	LONG SetTrendGridVisible(LONG isVisible)		
ShowMarkerValue	LONG ShowMarkerValue(void)		

(2) **Function Description**

AddChannel Archive Function

Description

This function is to add a channel in a logging group name or a tag name to the trend if the tag is a logging trend.

Syntax

LONG AddChannel Archive(LONG channelNo, LPCTSTR modelName, LPCTSTR tagName, DOUBLE minValY, DOUBLE maxValY)

channelNo - Enter LONG-type numbers as the number of the channel to be added

modelName – Enter the logging group name of the logging tag to be added. This is a string value.

tagName - Enter the name of the logging tag to be added. This is a string value.

minValY – Enter DOUBLE-type numbers as the tag's minimum value.

maxValY – Enter DOUBLE-type numbers as the tag's maximum value.

Reference

To add three channels to the Trend,

AddChannel_Archive(1, "SCADA_SEC", "SCADA.UT_EHV02.VRS", 0, 400)

AddChannel_Archive(2, "SCADA_SEC", "SCADA.UT_EHV02.VST", 0, 400)

AddChannel_Archive(3, "SCADA_SEC", "SCADA.UT_EHV02.VTR", 0, 400)

AddChannel_RealTime Function

Description

This function is to add a channel in a tag name to the trend if the trend is an online tag. But, in most cases, online monitoring is possible with logging tags, therefore, AddChannel_Archive is more widely used than this function.

Syntax

LONG AddChannel RealTime(LONG channelNo, BSTR tagName, DOUBLE minValY, DOUBLE maxValY)

channelNo - Enter LONG-type numbers as the number of the channel to be added.

tagName - Enter the name of the online tag to be added. This is a string value.

minValY – Enter DOUBLE-type numbers as the tag's minimum value.

maxValY – Enter DOUBLE-type numbers as the tag's maximum value.

Reference

To add three channels to the Trend

AddChannel_RealTime (1, "SCADA.UT_EHV02.VRS", 0, 400)

AddChannel RealTime (2, "SCADA.UT EHV02.VST", 0, 400)

AddChannel_RealTime (3, "SCADA.UT_EHV02.VTR", 0, 400)

AddTrendChannel Function

Description

This function is to add a channel to the trend. But, this function is rarely used because the user needs to input ID information on the logging group and tag as parameters. It is recommended to use AddChannel_Archive if the logging group ID or tag ID is unknown.

Syntax

LONG AddTrendChannel(LPCTSTR dataSource, LONG channelNo, LONG modelID, LPCTSTR modelName, LONG tagID, LPCTSTR tagName, LONG minValueY, LONG maxValueY)

dataSource – Enter the following string values to identify whether the tag to be added is a logging tag or an online tag.

Logging Tag: "Archive"

Online Tag: "OnLine"

channelNo – Enter LONG-type numbers as the number of the channel to be added.

modelID – Enter LONG- type numbers as the logging group ID of the logging tag to be added.

modelName – Enter the logging group name of the logging tag to be added. This is a string value.

tagID – Enter LONG- type numbers as the tag ID of the logging tag to be added.

tagName - Enter the tag name of the online tag to be added.

minValY - Enter DOUBLE-type numbers as the tag's minimum value.

maxValY - Enter DOUBLE-type numbers as the tag's maximu value.

Reference

To add three channels to the Trend,

AddTrendChannel("Archive", 1, 9, "INV1_SEC", 2710, "SOLAR.INV_1.1_AI_CT", 0, 327)

AddTrendChannel("Archive", 2, 9, "INV1_SEC", 2714, "SOLAR.INV_1.1_AI_VO", 0, 450)

AddTrendChannel("Archive", 3, 9, "INV1_SEC", 2713, "SOLAR.INV_1.1_AI_PW", 0, 60)

ApplyTrend Function

Description

This function is to make the monitoring start after adding a channel to the trend.

Syntax

void ApplyTrend(void)

Reference

ApplyTrend ()

ApplyProperty Function

Description

This function is to apply the trend properties to the trend after changing them.

Syntax

LONG ApplyProperty(VARIANT_BOOL bFlag) bFlag - Input TRUE.

Reference

ApplyProperty(TRUE)

DeleteTrendChannel Function

Description

This function is to delete a certain channel or the entire channels from the trend.

Syntax

LONG DeleteTrendChannel(LPCTSTR channelStr, LONG channelNo)

channelStr - Enter the channel name to be deleted.

channelNo - Enter the channel number to be deleted.

But, when deleting the entire channels, enter empty character ("") in channelStr and '0' in channelNo.

Reference

To delete 'Channel 3' from the trend (in case that the channel name is known) DeleteTrendChannel ("Channel3", 3)

To delete 'Channel 3' from the trend (in case that the channel name is unknown) DeleteTrendChannel ("", 3)

The above two methods are used because there are same tags in channels.

To delete the entire channels from the trend, DeleteTrendChannel ("", 0)

LoadHistoryTrend Function

Description

This function is to search logging data in a designated time zone. Enter year, month, week, day, hour to set the start and end time of a certain time zone and search logging data in the time zone. Enter a number in year, moth, week and day and enter 0 in the rest. Make bFlag TRUE to search data on a specific day/month/year or make it FALSE to search them as of the current time.

Syntax

void LoadHistoryTrend(LONG year, LONG month, LONG week, LONG day, LONG hour, BOOL bFlag)

year – Enter the year to search in LONG-type numbers.

month – Enter the month to search in LONG-type numbers.

week - To search data in a certain week, enter 1. If not, enter 0.

day - Enter the day to search in LONG-type numbers.

hour - Enter the hour to search in LONG-type numbers.

bFlag - To search data on a specific day/month/year, enter TRUE. If not, enter FALSE.

Reference

To search logging data in the time zone in which a certain date belongs to.

Year Search - LoadHistoryTrend(2007, 0, 0, 0, 0, TRUE)

Month Search - LoadHistoryTrend(2007, 5, 0, 0, 0, TRUE)

Week Search - LoadHistoryTrend(2007, 5, 1, 0, 0, TRUE)

Day Search - LoadHistoryTrend(2007, 5, 0, 13, 0, TRUE)

Hour Search-LoadHistoryTrend(2007, 5, 0, 13, 10, TRUE)

For One Hour Search as of the current time,

LoadHistoryTrend(0, 0, 0, 0, 1, FALSE)

For 2-Hour Search as of the current time,

LoadHistoryTrend(0, 0, 0, 0, 2, FALSE)

For 6-Hour Search as of the current time, LoadHistoryTrend(0, 0, 0, 0, 6, FALSE)

LoadPeriodTrend Function

Description

As a function to search data in a certain period, it presents One Hour Search, One Day search, One Week Search and One Month Search as of the current time.

Svntax

LONG LoadPeriodTrend(LONG days, BOOL bFlag)

days – Enter the following as parameters for searching.

One Hour Search: 1 One Day search: 2

One Week Search: 3 One Month Search: 4

bFlag – Input TRUE.

Reference

For One Hour Search, LoadPeriodTrend (1, TRUE)

For One Day Search, LoadPeriodTrend (2, TRUE)

For One Week Search, LoadPeriodTrend (3, TRUE)

For One Month Search, LoadPeriodTrend (4, TRUE)

Load_ArchivePeriod_Trend Function

Description

As a function to search data in a certain period, it presents Hour Search, One Day search, One Week Search and One Month Search as of the current time. For Hour Search, 1, 2, 3, 6 and 12 hour search are available.

Syntax

LONG Load ArchivePeriod Trend (LONG type, LONG days, VARIANT BOOL bFlag)

type - Enter the following as period parameters for searching.

One Hour Search: 1 One Day Search: 2 One Week Search: 3 One Month Search: 4

days – This parameter is applied only when One Hour Search is entered for type.

Enter 1, 2, 3, 6 or 12 to search 1, 2, 3, 6 or 12 hour search.

bFlag - Input TRUE.

Reference

For One Hour Search after selecting Hour Search, Load ArchivePeriod Trend(1, 1, TRUE)

For One Day Search, Load_ArchivePeriod_Trend(2, 0, TRUE)

For One Week Search. Load ArchivePeriod Trend(3, 0, TRUE)

For One Month Search, Load_ArchivePeriod_Trend(4, 0, TRUE)

LoadPeriodDataTrend Function

Description

As a function to search data in a certain period, if the user inputs the Start time and End time, this function searches logging data during the period

Syntax

LONG LoadPeriodDataTrend(LPCTSTR strFrom, LPCTSTR strTo, LONG bSample, LPCTSTR bFlag)

strFrom – Enter the following type as the Start time. Number 0 through 24 are used for Time Display. "2008/02/14 11:00:00"

strTo – Enter the following type as the End time. Number 0 through 24 are used for Time Display. "2008/02/14 11:10:00"

bSample - Decide whether to display data after applying sampling. Enter 0 as the default value.

bFlag - This is a reserve parameter variable. Enter"".

Reference

To search data from 13 o'clock on February 11, 2008 to 14 o'clock on February 11, LoadPeriodDataTrend("2008/02/11 13:00:00", "2008/02/11 14:00:00", 0, "")

Remember that there is empty space between day and time.

ShowTrendProperty Function

Description

This function is to show the property window of the trend.

Syntax

void ShowTrendProperty()

Reference

ShowTrendProperty(TRUE)

SetTrendGridVisible Function

Description

This function is to show grid visible properties of the trend.

Syntax

void SetTrendGridVisible()

Reference

SetTrendGridVisible ()

ShowMarkerValue Function

Description

This function is to show marker visible properties of the trend.

void ShowMarkerValue()

Reference

ShowMarkerValue ()

23.4 List Trend Viewer

As a screen to monitor changes in tag values over time, this List Trend Viewer is an Active X module that displays online data as well as logging data stored in the logging server.

23.4.1 Prerequisite and Environment

List trend properties can be set up only after the ActiveX module is inserted to the InfoU graphic editor. Channel assignment, tag assignment and other tag properties to be monitored are designated in advance as List Trend properties.

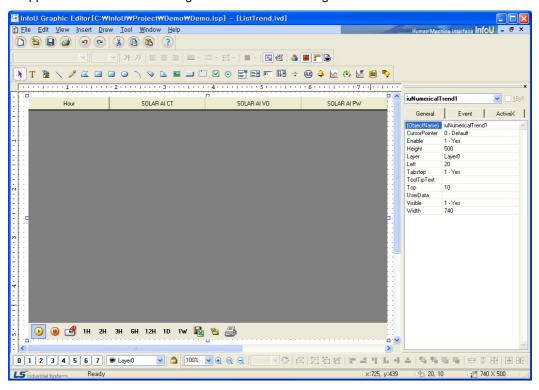
Channels or tags are not basically assigned to the list trend status inserted in the graphic editor. It is essential to display the Property Window before setting up the property values needed to run the List Trend. After the property values are saved, the List Trend Monitoring Screen appears upon the opening of the screen under the Run time environment.

The engineering values are entered in the following order on the List Trend viewer. If is selected tools, immediate inserting can be performed.

23.4.2 Screen Configuration

(1) Engineering Screen

If List Trend is inserted and a tag to be monitored is selected and then settings are all completed, the following screen appears. The screen in this figure shows that three tags are selected.



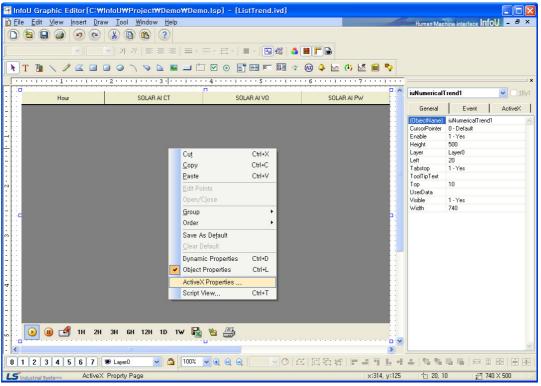
(2) Runtime Screen

If default list trend settings are completed on the engineering screen above and then, the runtime is executed, the screen shows that data values are changed over time.

06-08 14:08:18 06-08 14:08:17 06-08 14:08:16	14 47	0	3
06-08 14:08:16	47		•
		0	2
00.0044.0045	44	0	3
06-08 14:08:15	49	0	4
06-08 14:08:14	68	0	6
06-08 14:08:13	89	0	2
06-08 14:08:12	76	0	2
06-08 14:08:11	28	0	7
06-08 14:08:10	26	0	2
06-08 14:08:09	79	0	6
06-08 14:08:08	15	0	2
06-08 14:08:07	63	0	6
06-08 14:08:06	3	0	3
06-08 14:08:05	11	0	1
06-08 14:08:04	1	0	4
06-08 14:08:03	10	0	7
06-08 14:08:02	10	0	6
06-08 14:08:01	26	0	1
06-08 14:08:00	94	0	5
06-08 14:07:59	42	0	6
06-08 14:07:58	30	0	10
06-08 14:07:57	33	0	6
06-08 14:07:56	18	0	4
06-08 14:07:55	32	0	7
06-08 14:07:54	33	0	4
06-08 14:07:53	78	0	0
06-08 14:07:52	26	0	8
00.00.14.07.51	EC	٥	,

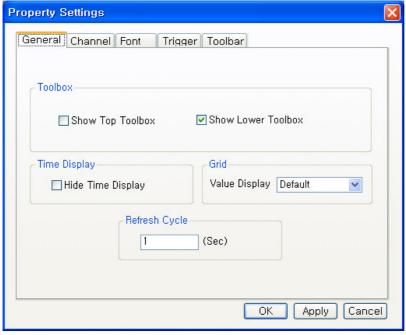
23.4.3 Settings

If the right side of the mouse is clicked on the List Trend and the 'Active X Properties' menu is selected, the setting window appears. The same setting screen is used for the engineering screen and the run-time screen. But, the runtime screen has Channel Setting function only. In addition, once property values are saved on the engineering screen, those values will be continuously valid however, if those property values are changed on the runtime screen, the changed values are applied only to the runtime screen. The changed values are cancelled if the runtime stops and the property values set up on the engineering screen are applied.



(1) General

The Engineering Screen and the Runtime Screen have the same general function.



(a) Show Top Toolbox/Show Lower Toolbox

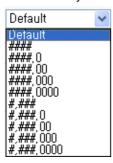
Set up whether to display Trend Toolbox on the top or the bottom during the running. Both can not be selected at the same time and it is allowed neither of them is selected. 'Show Lower Toolbox' has been selected as the default value.

(b) Time Display

Select whether to show/hide Time Display column on the screen. 'Hide Time Display" has not been selected as the default value.

(c) Grid (Value Display)

The user may select a method to display grid value.



Default: Display the default vale.

: Do not display decimal points regardless of tag DB

####.0: Indicate down to one decimal point regardless of tag DB.

####.00 : Indicate down to two decimal points regardless of tag DB.

####.000 : Indicate down to three decimal points regardless of tag DB.

####.0000: Indicate down to four decimal points regardless of tag DB.

#,###: Do not display decimal points regardless of tag DB and indicate digit with comma.

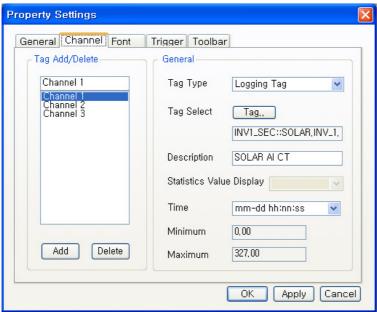
#,###.0: Indicate down to one decimal point regardless of tag DB and indicate digit with comma

#,###.00 : Indicate down to two decimal points regardless of tag DB and indicate digit with comma. #,###.000 : Indicate down to three decimal points regardless of tag DB and indicate digit with comma. #,###.0000: Indicate down to four decimal points regardless of tag DB and indicate digit with comma.

(d) Refresh Cycle

Designated a refresh cycle. One second has been set up as the default value.

(2) Channel



The Engineering Screen and the Runtime Screen have the same Channel functions. Up to 20 channels are provided in the List Trend. There is at least one channel and it is available to designate tags for each channel. If one channel is an online tag, the rest channels have to be online tags. Likely, if one channel is a logging tag, the rest channels have to be logging tags. Also, if any logging data needs to be saved or searched, the logging tag should be selected and used. Online tags do not provide the logging function.

(a) Channel Add/Delete

A channel is added/deleted. At least one or more than one but not more than 20 channels can be selected.

(b) Tag Kind

Select either Logging Tag or Online Tag.



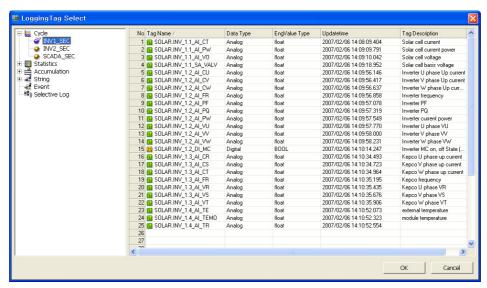
(c) Tag Select

Assign a tag to the channel. One tag can be assigned to one channel.

If a logging tag is selected, the logging tag browser where the associated logging group is set up is called while the online tag browser is called for an online tag.

Be cautious that logging data can be searched only when a logging tag is selected.

If a tag is selected, the tag's minimum value and maximum values are displayed. These min/max values apply to Y-axis values for Y axis settings.



'Logging Tag Select' screen

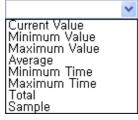
(d) Description

Describe the selected tag. If any description has already been written, this description is displayed instead of the tag name on the grid. If not, the tag name is displayed. If any tag description has been recorded in the tag DB, that description is displayed even the user skips this item.

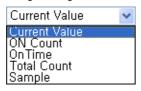
(e) Statistics Value Display

This function becomes active only when a statistics tag is selected. For analog tags, select one value among current value, min value, max value, average value, sum and sample and for digital tags, select one value among current value, On count, On Time, Total count and sample. If the tag is not a statistics tag, it becomes inactive.

* Analog Tag



* Digital Tag

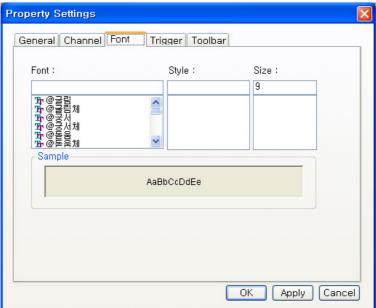


(f) Time Display

Select a time type to display on the grid.

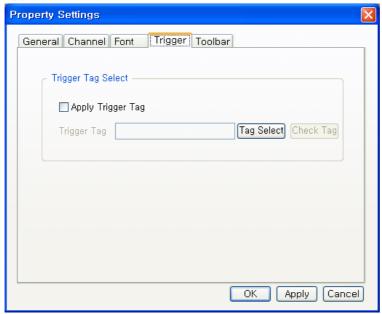


(3) Font



Decide Font, Style and Size to be displayed on the list trend viewer list.

(4) Trigger

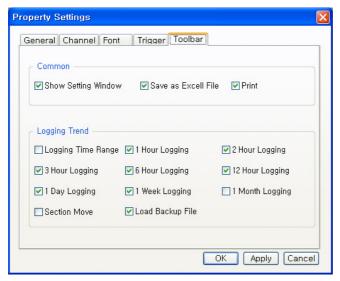


This Trigger is a function used to run the list trend only when the base tag for trigger is True.

If 'Apply Trigger Tag' is clicked, the trigger tag item becomes active. Press 'Tag Select' to select a tag. At this time, be cautious that the tag should be corresponding to the tag kind designated during the channel settings. Usually, they are corresponding to each other and the Tag Selection Window is displayed but, if Tag Kind is changed latter, trigger setting items shall be checked.

If the value of a trigger tag is O, that is, if it is False, the List Trend does not show any tag value. It shows tag values only the value is not O.

(5)Toolbar Select toolbar (menus).



(a) Start ()

This is a common function that applies to both online trends and logging trends,

Once trend monitoring starts, data is renewed every second and 5 minutes is set as the time for X axis. To view the current screen again during the logging search, press this button and the current time screen is displayed.

This Start button can not be selected as the default value.

(b) Stop ()

Trend Monitoring stops. This function is available only for logging trends. That is, this function exists only when logging tags are selected and online trends do not have this function.

This Stop button can not be selected as the default value like the Start button

(c) Show Setting Window ()



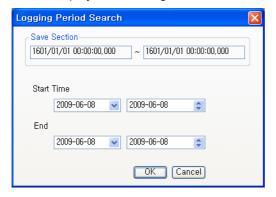
Setting values of the current trend can be changed. This is a common function that applies to both online trends and logging trends,

This Setting Window has been described before.

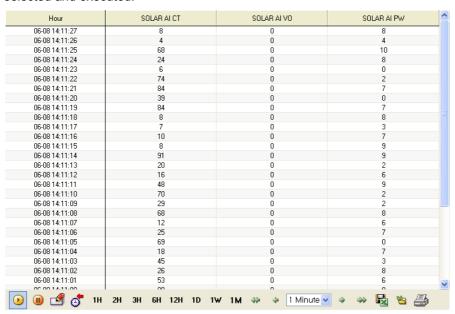
(d) Logging Trend (Logging Time Range) ()



Input logging conditions (period and sampling conditions) to search past data. Click on the relevant icon to display the following window.



'Saved Section' shows the starting time and the end time to save the entire data and this is referred when deciding the start time and the end time. The following figure shows that 'One Hour Search' is selected and executed.



This function is available only for logging trends.

(e) 1 Hour Logging (1H)

View 1 hour logging data including the current time. For example, if it is 10:20, the logging data between 10 and 11o'clock are viewed. This function is available only for logging trends.

(f) 2 Hour Logging (^{2H})

View logging data in the time range where the current time belongs to by dividing 24 hours into 0~2 o'clock, 2~4 o'clock, 4~6 o'clock, ..., 20~22 o'clock and 22~midnight. For example, if it is 10:20, the logging data between 10 o'clock and noon are viewed. This function is available only for logging trends.

(g) 3 Hour Logging (^{3H})

View logging data in the time range where the current time belongs to by dividing 24 hours into 0~3 o'clock, 3~6o'clock, 6~9o'clock, ..., 18~21 o'clock and 21~midnight. For example, if it is 10:20, the logging data between 9 o'clock and noon are viewed. This function is available only for logging trends.

(h) 6 Hour Logging (6H)

View logging data in the time range where the current time belongs to by dividing 24 hours into 0~6 o'clock, 6~12o'clock, 12~18o'clock and 18~midnight For example, if it is 10:20, the logging data between 6 o'clock and noon are viewed. This function is available only for logging trends.

(i)12 Hour Logging (12H)

View logging data in the time range where the current time belongs to by dividing 24 hours into 0~noon and noon~ midnight. For example, if it is 10:20, the logging data between 0 o'clock and noon are viewed. This function is available only for logging trends.

(j)1 Day Logging (1D)

View today's logging data. This function is available only for logging trends.

(k) 1 Week Logging (1W)

View this week's logging data. The period is from Monday through the midnight of Sunday. This function is available only for logging trends.

(I)1 Month Logging (1M)

View this month's logging data This function is available only for logging trends.

1 Minute 🕶 (m) Section Search (

For logging search, the user may use 'Section Search' to select a section and search data in the section.

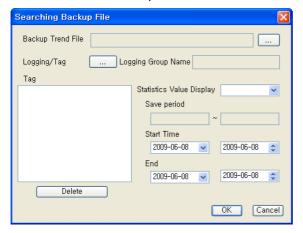
- Press the very left button to move the first time from which data began to be stored in the current 1) DB server.
- 2) Press the second button to move to the section one minute before the current search. It is possible to keep moving until it reaches the very first.
- 3) The third button is to select a time section to search time. It is available to select a time section among 1minute, 5 minutes, 10 minutes, 30 minutes and 1 hour.



If 'Section Search' is conducted after selecting 'Logging Data View', the section can be conveniently moved up and down. But, in case that 'Logging Data View' is selected instead of direct setting of a logging section, the times in the middle can not be selected.

- 4) The fourth button is to move the section one minute after the current search section. It is possible to keep moving until it reaches the very end.
- 5) Press the very right button to move to the current time. This Section Search function is available only for logging search. This function is not applied to online monitoring. This function is available only for logging trends.
- (n) Load Backup File (

The trend backup files saved on the server or other storage devices can be imported and viewed in a list. If a relevant button is pressed on the toolbar, the following window appears.

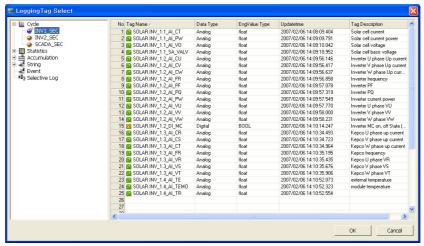


(o) Backup Trend File

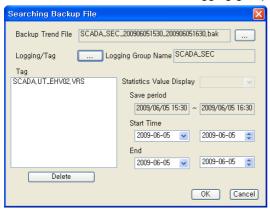
Select a trend backup file. The file has extension '.bak.'

(p) Logging/Tag

Only one logging group can be selected but, selecting multiple tags is allowed. The selected tag appears on the list box and the saving period is displayed on 'Save Period.'

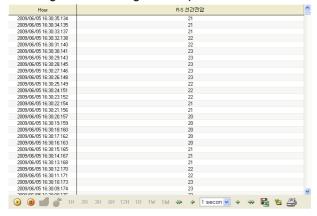


This screen shows the selected logging group and tags.



(q) Start Time/Emd

'Start Time' and 'End' are used to select the time to search. The following screen shows the results of selecting and searching a backup file.



After data are searched on the search screen, those data can be moved by using Section Search function. To return to the original screen, press Start button.

23.4.4 How to Use Script

(1) **Function List**

Function Name	Туре		
AddChannel_Archive	LONG AddChannel_Archive(LONG channelNo, LPCTSTR modelName,		
	LPCTSTR tagName, LPCTSTR tagDesc)		
AddChannel_RealTime	LONG AddChannel_RealTime(LONG channelNo, LPCTSTR tagName, ,		
	LPCTSTR tagDesc)		
AddTrendChannel	LONG AddTrendChannel(LPCTSTR dataSource, LONG channelNo,		
	LONG modelID, LPCTSTR modelName, LONG tagID, LPCTSTR		
	tagName, LPCTSTR tagDesc)		
ApplyTrigger	void ApplyTrigger(bApply, modelName, tagName)		
LoadHistoryTrend	void LoadHistoryTrend(LONG year, LONG month, LONG week, LONG		
	day, LONG hour, BOOL bFlag)		
LoadPeriodTrend	LONG LoadPeriodTrend(LONG type, BOOL bFlag)		
Load_ArchivePeriod_Tre	LONG Load_ArchivePeriod_Trend(LONG type, LONG hours,		
nd	VARIANT_BOOL bFlag)		
LoadPeriodDataTrend	LONG LoadPeriodDataTrend(LPCTSTR strFrom, LPCTSTR strTo, LONG		
	bSample, LPCTSTR bFlag)		
ShowTrendProperty	void ShowTrendProperty(void)		
SetVisibleChanGrid	LONG SetVisibleChanGrid(LPCTSTR strTagName, LONG chanNo,		
	LONG bFlag)		

(2) **Function Description**

AddChannel_Archive Function

Description

This function is to add a channel in a logging group name or a tag name to the list trend if the tag is a logging list trend.

Syntax

LONG AddChannel_Archive(LONG channelNo, LPCTSTR modelName, LPCTSTR tagName, LPCTSTR tagDesc)

channelNo – Enter LONG-type numbers as the number of the channel to be added.

modelName - Enter the logging group name of the logging tag to be added. This is a string value.

tagName – Enter the name of the logging tag to be added. This is a string value.

tagDesc - Enter the description of the logging tag to be added. If there is no value, use the description value of the tag DB and if there is a value, use it. This is a string value.

Reference

To add three channels to the List Trend,

AddChannel_Archive(1, "SCADA_SEC", "SCADA.UT_EHV02.VRS", "VRS")

AddChannel_Archive(2, "SCADA_SEC", "SCADA.UT_EHV02.VST", "VST")

AddChannel_Archive(3, "SCADA_SEC", "SCADA.UT_EHV02.VTR", "VTR")

AddChannel_RealTime Function

Description

This function is to add a channel in a tag name to the trend if the trend is an online tag. But, in most cases, online monitoring is possible with logging tags, therefore, AddChannel_Archive is more widely used than this function.

Syntax

LONG AddChannel_RealTime(LONG channelNo, BSTR tagName, LPCTSTR tagDesc)

channelNo – Enter LONG-type numbers as the number of the channel to be added.

tagName - Enter the name of the online tag to be added. This is a string value.

tagDesc – Enter the description of the logging tag to be added. If there is no value, use the description value of the tag DB and if there is a value, use it. This is a string value.

Reference

To add three channels to the List Trend,

AddChannel RealTime (1, "SCADA.UT EHV02.VRS", "VRS")

AddChannel_RealTime (2, "SCADA.UT_EHV02.VST", "VST")

AddChannel RealTime (3, "SCADA.UT EHV02.VTR", "VTR")

AddTrendChannel Function

Description

This function is to add a channel to the list trend. But, this function is rarely used because the user needs to input ID information on the logging group and tag as parameters. It is recommended to use AddChannel_Archive if the logging group ID or tag ID is unknown.

Syntax

LONG AddTrendChannel(LPCTSTR dataSource, LONG channelNo, LONG modelID, LPCTSTR modelName, LONG tagID, LPCTSTR tagName, LPCTSTR tagDesc)

dataSource – Enter the following string values to identify whether the tag to be added is a logging tag or an online tag.

Logging Tag: "Archive"

Online Tag: "OnLine"

channelNo – Enter LONG-type numbers as the number of the channel to be added.

modelID - Enter LONG- type numbers as the logging group ID of the logging tag to be added.

modelName – Enter the logging group name of the logging tag to be added. This is a string value.

tagID – Enter LONG- type numbers as the tag ID of the logging tag to be added.

tagName – Enter the name of the online tag to be added. This is a string value.

tagDesc – Enter the description of the logging tag to be added. If there is no value, use the description value of the tag DB and if there is a value, use it. This is a string value.

Reference

To add three channels to the List Trend.

```
AddTrendChannel("Archive", 1, 9, "INV1_SEC", 2710, "SOLAR.INV_1.1_AI_CT", "CT")
AddTrendChannel("Archive", 2, 9, "INV1_SEC", 2714, "SOLAR.INV_1.1_AI_VO", "VO")
AddTrendChannel("Archive", 3, 9, "INV1_SEC", 2713, "SOLAR.INV_1.1_AI_PW", "PW")
```

ApplyTrigger Function

Description

This function is to apply the trigger tag properties to the list trend. A value appears unless the trigger value is O (True). If the value is O, it is not displayed.

void ApplyTrigger(bApply, modelName, tagName)

bApply - Trigger is applied: True

Trigger is not applied: False

modelName - Make sure to enter the logging group name if the list trend is logging.

If the user fails to enter the logging group name, noting is recorded because the trigger tag can not be searched.

tagName - Enter the tag name.

Reference

ApplyTrigger(TRUE, "SCADA SEC", "Update"): In case that a tag, whose logging group is "SCADA_SEC" and name is "Update," is designated as a trigger tag

LoadHistoryTrend Function

Description

This function is to search logging data in a designated time zone. Enter year, month, week, day, hour to set the start and end time of a certain time zone and search logging data in the time zone. Enter a number in year, moth, week and day and enter 0 in the rest. Make bFlag TRUE to search data on a specific day/month/year or make it FALSE to search them as of the current time.

Syntax

void LoadHistoryTrend(LONG year, LONG month, LONG week, LONG day, LONG hour, BOOL bFlag)

year - Enter the year to search in LONG-type numbers.

month – Enter the month to search in LONG-type numbers.

week – To search data in a certain week, enter 1. If not, enter 0.

day - Enter the day to search in LONG-type numbers.

hour - Enter the hour to search in LONG-type numbers.

bFlag - To search data on a specific day/month/year, enter TRUE. If not, enter FALSE.

Reference

To search logging data in the time zone in which a certain date belongs to,

Year Search - LoadHistoryTrend(2007, 0, 0, 0, 0, TRUE)

Month Search - LoadHistoryTrend(2007, 5, 0, 0, 0, TRUE) Week Search - LoadHistoryTrend(2007, 5, 1, 0, 0, TRUE) Day Search - LoadHistoryTrend(2007, 5, 0, 13, 0, TRUE) Hour Search-LoadHistoryTrend(2007, 5, 0, 13, 10, TRUE)

For One Hour Search as of the current time, LoadHistoryTrend(0, 0, 0, 0, 1, FALSE)

For 2-Hour Search as of the current time, LoadHistoryTrend(0, 0, 0, 0, 2, FALSE)

For 6-Hour Search as of the current time, LoadHistoryTrend(0, 0, 0, 0, 6, FALSE)

LoadPeriodTrend Function

Description

As a function to search data in a certain period, this presents One Hour Search, One Day search, One Week Search and One Month Search as of the current time.

Syntax

LONG LoadPeriodTrend(LONG days, BOOL bFlag)

type - Enter the following as parameters for searching.

One Hour Search: 1 One Day search: 2 One Week Search: 3 One Month Search: 4

bFlag – Input TRUE.

Reference

For One Hour Search, LoadPeriodTrend (1, TRUE)

For One Day Search, LoadPeriodTrend (2, TRUE)

For One Week Search, LoadPeriodTrend (3, TRUE)

For One Month Search, LoadPeriodTrend (4, TRUE)

Load ArchivePeriod Trend Function

Description

As a function to search data in a certain period, Hour Search, this presents One Day search, One Week Search and One Month Search as of the current time. For Hour Search, 1, 2, 3, 6 and 12 hour search are available.

Syntax

LONG Load_ArchivePeriod_Trend (LONG type, LONG hours, VARIANT_BOOL bFlag)

type - Enter the following as period parameters for searching.

One Hour Search: 1 One Day search: 2 One Week Search: 3 One Month Search: 4

hours – This parameter is applied only when One Hour Search is entered for type.

Enter 1, 2, 3, 6 or 12 to search 1, 2, 3, 6 or 12 hour search.

bFlag - Input TRUE.

Reference

For One Hour Search after selecting Hour Search, Load ArchivePeriod Trend(1, 1, TRUE)

For One Day Search,

Load ArchivePeriod Trend(2, 0, TRUE)

For One Week Search,

Load ArchivePeriod Trend(3, 0, TRUE)

For One Month Search,

Load_ArchivePeriod_Trend(4, 0, TRUE)

LoadPeriodDataTrend Function

Description

As a function to search data in a certain period, if the user inputs the Start time and End time, this function searches logging data during the period

Syntax

LONG LoadPeriodDataTrend(LPCTSTR strFrom, LPCTSTR strTo, LONG bSample, LPCTSTR bFlag)

strFrom - Enter the following type as the Start time. Number 0 through 24 are used for Time Display. "2008/02/14 11:00:00"

strTo – Enter the following type as the End time. Number 0 through 24 are used for Time Display. "2008/02/14 11:10:00"

bSample - Decide whether to display data after applying sampling. Enter 0 as the default value.

bFlag - This is a reserve parameter variable. Enter"".

Reference

To search data from 13 o'clock on February 11, 2008 to 14 o'clock on February 11, LoadPeriodDataTrend("2008/02/11 13:00:00", "2008/02/11 14:00:00", 0, "")

Remember that there is empty space between day to time.

SetVisibleChanGrid Function

Description

This function is to show or hide channels

Syntax

LONG SetVisibleChanGrid(LPCTSTR strTagName, LONG chanNo, LONG bFlag)

strTagName – Enter the tag name of the channel to be shown or hidden. chanNo - Enter the number of the channel to be shown or hidden.

bFlag – Input True if the channel is to be shown or input False if it is to be hidden.

Reference

ShowTrendProperty(TRUE)

ShowTrendProperty Function

Description

This function is to show the property window of the trend.

Syntax

void ShowTrendProperty()

Reference

ShowTrendProperty(TRUE)

23.5 Recipe Viewer

This Recipe Viewer is an Active X module that displays the recipe contents set up during the engineering and has Recipe Control, Dynamic Data Loading and Dynamic Data Saving function.

23.5.1 **Prerequisite and Environment**

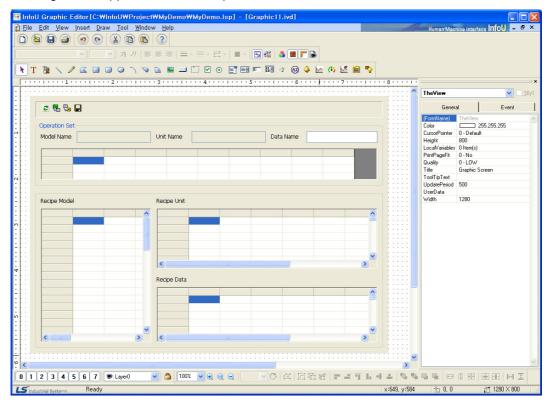
The ActiveX module is inserted to the InfoU graphic editor. The recipe viewer displays the data that have been set up during the engineering and does not have its own property.

is selected from the tools, immediate inserting can The recipe viewer is executed in the following order. If be performed.

23.5.2 Screen Configuration

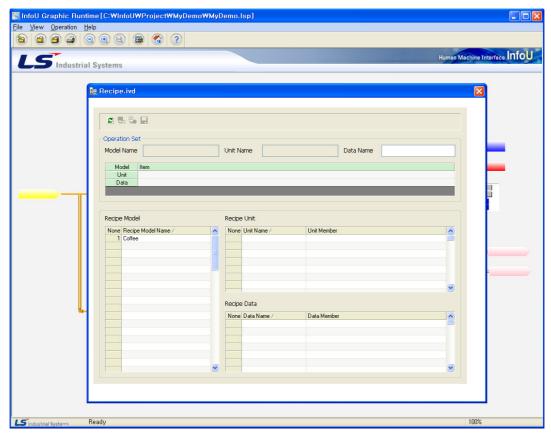
Engineering Screen

The following screen appears after the recipe viewer is inserted.



(2) Runtime Screen

If the above Engineering Screen is executed on the runtime, the screen is as follows:

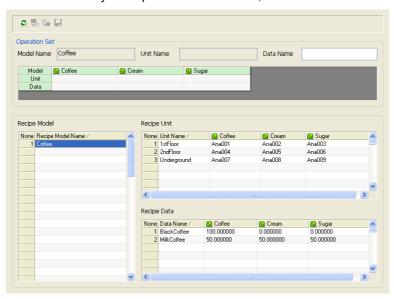


23.5.3 **Real-Time Function**

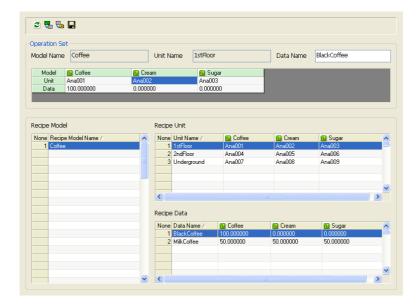
Setting Data Diplay (1)

The recipe viewer is largely divided into the upper Operation Set part and the lower setting data display and select part. The user may conduct various operations by selecting a recipe model and each unit and data item and combining them properly and the selected items are displayed on the upper Operation Set part. During the runtime, a recipe model list is displayed on the left and if a certain model is selected, the associated recipe units and data to the selected model appear.

* In case that only a recipe model is selected;

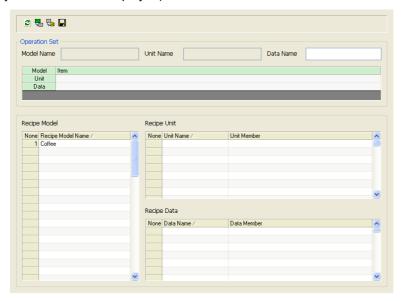


If recipe units and data are selected, the selected items are displayed on 'Operation Set' as seen in the figure.



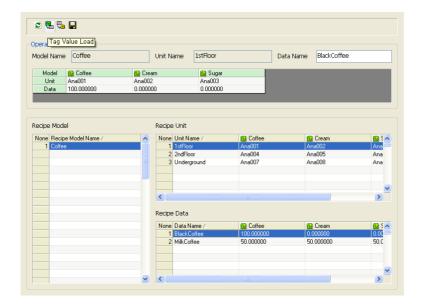
(2) Refresh (2)

If Refresh button is selected, the existing operations are cleared and it is converted to the initial screen. (Only the model list is displayed)



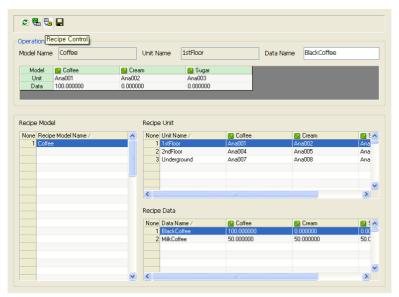
(3) Tag Value Load(🔀)

If models and units are selected, 'Tag Value Load' becomes active. If this button is selected, the tag values displayed on the Operation Set are loaded from the real-time data service module and they are displayed on the Operation Set. In this status, the user may control the recipe since its requirements (model + unit + data) are met and also save this data in the DB. Of course, it is also available to edit it directly on the data cell of 'Operation Set' to control and save it in the DB.



Recipe Control (🧓) (4)

For Recipe Control, each Model, Unit and Data should be selected. Of course, if some data are secured through Tag Value Load instead of selecting data, Recipe Control is also possible even in such a case. Recipe Control is possible only when the current user's authority is higher than the required authority according to the recipe model.



If Recipe Control is performed, the following dialog box appears to confirm the function.



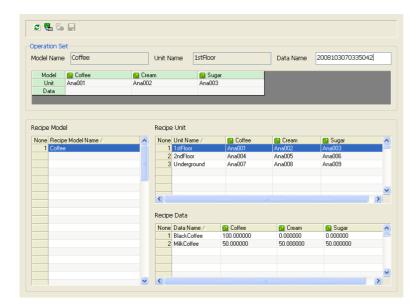
For 'Control Results,' two options-Returns or No Return-are presented. Select one and press OK to perform the Recipe Control. If the user has an insufficiently high authority to control the recipe model, he/she may not control it. At this time, the following error dialog box appears.



Chapter 23 Graphic Object

(5) Recipe Data Save(🔲)

The recipe viewer presents a function to save recipe data during the runtime. But, only up to 64 recipe data can be saved in a model. The user may load the corresponding values to the unit displayed on the Operation Set from the real-time data service and save them through the recipe viewer or input data directly to cells to save them. When saving data, use the same data name as set up on the Operation Set and if other recipe data has the same name, it is updated. If the data name item is empty, the default string value of "year + month+ day + hour+ minute+ second + msecond" is given as a data name.



(6) Recipe Control Result

The control result can be received if "Return" has been set during the Recipe Control (But, this is applied only when it is not under the web environment). The following screen is displayed automatically if the Recipe Control Result is received from I/O Manager.



23.5.4 How to Use Script

Function List (1)

Function Name	Туре
SelectRecord	SHORT SelectRecord(LPCTSTR ModelName, LPCTSTR UnitName,
	LPCTSTR DataName)
ControlWorkingSet	SHORT ControlWorkingSet(SHORT nMode)
ControlSet	SHORT ControlSet(SHORT nMode, LPCTSTR ModelName, LPCTSTR
	UnitName, LPCTSTR DataName)
LoadDynamic	SHORT LoadDynamic(void)
ChangeWSDataName	SHORT ChangeWSDataName(LPCTSTR DataName)
SaveData	SHORT SaveData(void)

Function Description (2)

SelectRecord Function

Description

This function is to make WorkingSet organized by selecting model name, unit name and data name.

Model name is an essential item while unit name and data name are optional.

If this function is executed, the relevant items are selected on the Recipe Viewer Control to form WorkingSet.

Syntax

SHORT SelectRecord(LPCTSTR ModelName, LPCTSTR UnitName, LPCTSTR DataName)

ModelName – As the name of the model to be selected and it is a string value.

UnitName – As the name of the unit to be selected and it is a string value.

DataName – As the name of the data to be selected and it is a string value.

Reference

Recipe Model Name: Coffee vending machine, Unit Name: Vending machine on the1 floor, Data Name: Milk coffee is selected,

SelectRecord("Coffee vending machine", "Vending machine on the1 floor", "milk coffee")

To select only model and unit except data,

SelectRecord("Coffee vending machine", "Vending machine on the1 floor", "")

ControlWorkingSet Function

Description

A function used when controlling recipe with the WorkingSet data that have been already selected.

Controlling is performed only model, unit and data are all equipped with.

For local system (not web), it is available to select whether to receive the control result or not.

Chapter 23 Graphic Object

Syntax

SHORT ControlWorkingSet(SHORT nMode)

nMode - A flag whether to receive the control result or not.

(1: Won't receive, 2: Will receive)

Reference

To perform controlling WorkingSet (Will receive), ControlWorkingSet(2)

To perform controlling WorkingSet (Won't receive),

ControlWorkingSet(1)

ControlSet Function

Description

A function used to perform controlling through a certain method without selecting directly on the recipe viewer control.

Internally, this function derives the same result as SelectRecord and ControlWorkingSet are executed simultaneously.

Syntax

SHORT ControlSet(SHORT nMode, LPCTSTR ModelName, LPCTSTR UnitName, LPCTSTR DataName)

nMode – A flag whether to receive the control result or not.

(1: Won't receive, 2: Will receive)

ModelName – As the name of the model to be selected and it is a string value.

UnitName – As the name of the unit to be selected and it is a string value.

DataName – As the name of the data to be selected and it is a string value.

Reference

Recipe Model Name: Coffee vending machine, Unit Name: Vending machine on the1 floor, Data Name: Milk coffee is selected.

(Not received the result),

Controlset(1, "Coffee vending machine", "Vending machine on the1 floor", "milk coffee")

LoadDynamic Function

Description

This function is to load tag values that belong to the unit member from the InfoU real-time server controlling without using the values that have been set up during the engineering. This function can be executed only when model and unit are selected.

Syntax

SHORT LoadDynamic(void)

Reference

With this function, it is possible to load real-time values of a tag and save them as recipe data or re-control them.

ChangeWSDataName Function

Description

This function is to rename the recipe data of WorkingSet.

Syntax

SHORT ChangeWSDataName(LPCTSTR DataName)

DataName – As the name of the data to be selected and it is a string value.

Reference

With this function, the name of the data shown on the current WorkingSet can be modified.

This function only helps to modify the name seen on the screen and does not mean to save it in the DB. After this function is used, the data can be saved in the DB with the following SaveData function.

SaveData Function

Description

This function is to save the recipe data displayed on the WorkingSet.

Syntax

SHORT SaveData(void)

Reference

This function is to save the data seen on the current WorkingSet in the DB. At this time, use the data name on the WorkingSet of the recipe viewer control for the name of the saved data. This name is changeable by using ChangeWSDataName function.

If the same data name already exists in the DB, a new data value is renewed in the DB, If the same data name does not exists in the DB, a new recipe data is added to the DB.

Also, if any recipe data name is not given, a new name is created and saved by combining time such as year, month, day, hour, minute, second and msecond.

23.6 Gauge Control

This Gauge Control is an ActiveX module to display and monitor the registered tag information in a type of gauge that fits well to the user's purpose by selecting or setting up a proper gauge. A gauge can be formed in various types by designating basic range or degree.

23.6.1 Prerequisite and Environment

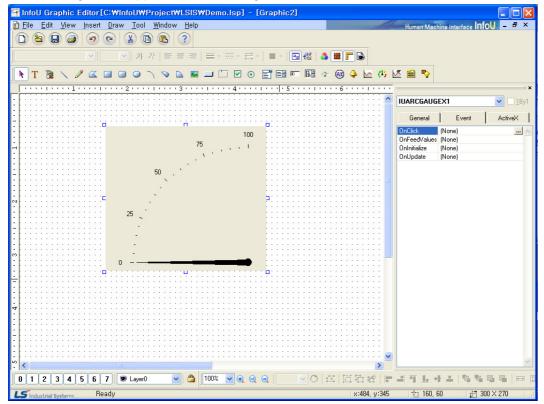
Gauge properties can be set up only after the ActiveX module is inserted to the InfoU graphic editor. Gauge properties broadly consist of General and Ticks. Since those gauge properties have been already set up with default values, they act with those default values if the user does not set up additionally.

The Gauge Control is executed in the following order. If is selected from the tools, immediate inserting can be performed.

23.6.2 Screen Configuration

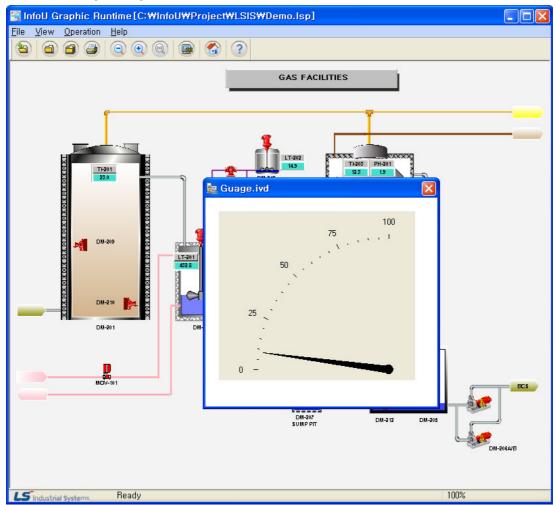
(1) Engineering Screen

The following screen appears after the Gauge Control is inserted.



(2) Runtime Screen

If the above Engineering Screen is executed on the runtime, the screen is as follows:



23.6.3 Property Settings

(1) General



(a) Tag Name

Enter a tag name to be gauged. A tag name can be designated with 'Search' (button on the right. Its update cycle is the same with the screen cycle of the InfoU graphic editor.

(b) Sample List

Various gauge samples are presented to fit to the user's purpose.

(c) Range

Designate a gauge range, that is, enter the min/max value.

(d) Degrees

Designate a gauge degree, that is, enter the start degree and the range degree.

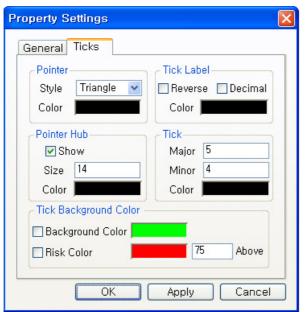
(e) Title

Designate a gauge title and color.

(f) Border

Designate background color and border style.

(2) Ticks



(a) Pointer

Designate a gauge pointer style and color.

(b) Tick Label

Designate a tick label color and style (Reverse, Decimal).

(c) Pointer Hub

Designate whether to show the pointer hub or not and decide its size and color.

(d) Tick

Designate the size of the Major Tick and Minor Tick and color.

(e) Tick Background Color

Designate the tick background color and the user may set up an additional risk reference value to designate a risk color.

If not checked, the background color is set up to display.

23.7 **PictureLamp Control**

This Picture Lamp Control is an ActiveX module to display and monitor the registered tag value in a proper picture form.

23.7.1 **Prerequisite and Environment**

Picture Lamp properties can be set up only after the ActiveX module is inserted to the InfoU graphic editor. Since those picture lamp properties have been already set up with default values, they act with those default values if the user does not set up additionally.

The Picture Lamp Control is executed in the following order. If inserting can be performed.

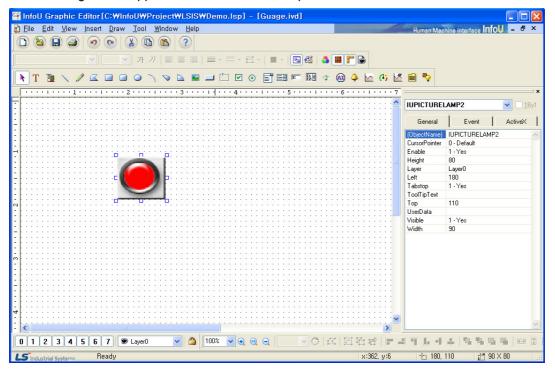


is selected from the tools, immediate

23.7.2 Screen Configuration

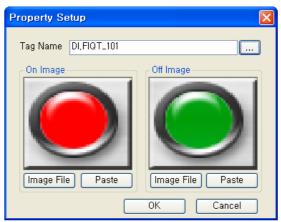
Engineering Screen (1)

The following screen appears after the Picture Lamp Control is inserted.



23.7.3 Function Description

(1) Property



(a) Tag Name

Enter a tag name to be a base for Picture Lamp. A tag name can be designated with 'Search' (-) button on the right. Its update cycle is the same with the screen cycle of the InfoU graphic editor.

- (b) On Image
 - If the tag value is TRUE, this image is shown. There are imagefile type and clipboard type as methods to select a picture.
- (c) Off Image

If the tag value is FALSE, this image is shown. There are imagefile type and clipboard type as methods to select a picture.

23.8 How to Use External ActiveX

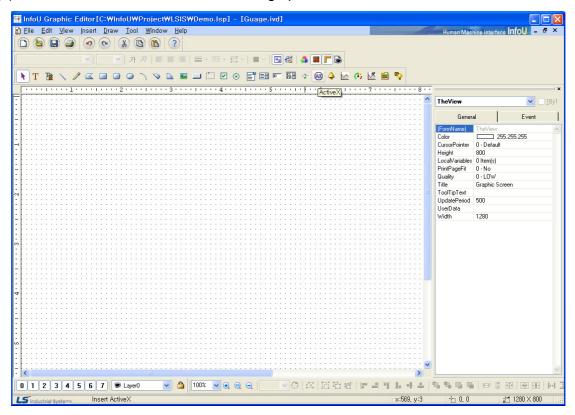
External ActiveX can be inserted and used on the graphic editor.

Here, Calendar OCX is used as an example of external
The same way can be applied to other ActiveX.

23.8.1 How to Use

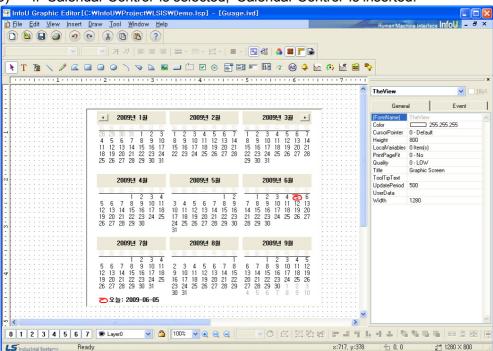
ActiveX module is inserted to the InfoU graphic editor. Calendar control is executed in the following order.

(1) Select from the tool menu of the graphic editor.



(2) Select 'Calendar Control'.





If 'Calendar Control' is selected, 'Calendar Control' is inserted.

Chapter 24 Web

24.1 InfoU Web Navigator Automatic Setup

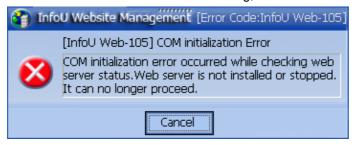
This Site Automatic Setup ensures even an inexperience user who does not have any expertise in web servers or website formation can configure a website easily and assess to the InfoU HMI system through the web browser from anywhere. The user can form a site he/she wants and create a new website. In addition, he/she may add a virtual folder to the existing website.

24.1.1 **Prerequisite**

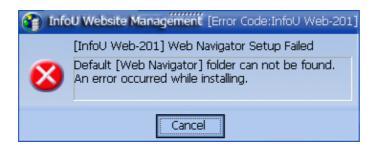
- (1) Internet Information Server (IIS) has been installed and executed.
- (2) InfoU HMI Server has been installed.
- (3) If the prerequisite is not met, the following error message appears and the prograom does not proceed anymore.
 - If Internet Information Services has not been installed, (a)



Internet Information Services is not running,

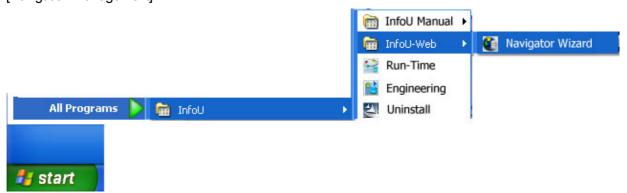


If InfoU Web can not be installed because Web Package has not been selected during the installation of (c) InfoU,



24.1.2 InfoU Web Navigator Run

(1) Run the InfoU Web Navigator by selecting 'Start ⇒ All Programs ⇒ InfoU Folder ⇒ InfoU-Web ⇒ [Navigation Management].'



(2) Brief information on the program is displayed on the splash screen.



(3) Create New Website

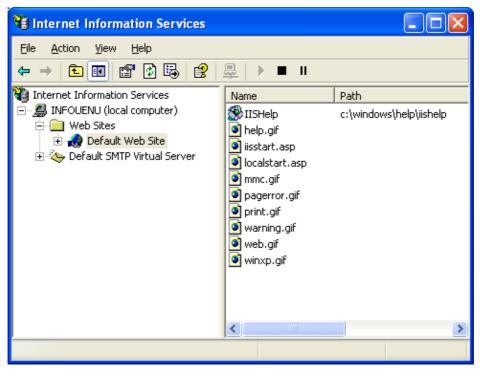
1. Create New Website

Two menus such as 'Create New Website' and 'Get Project Information' are presented. A new user needs to select 'Create New Website' after installing InfoU. The menu 'Create New Website' forms a site needed for monitoring the existing plant screen from the web.



2. Internet Information Service

A web site to be directly controlled by the InfoU user is created on the Internet Information server (IIS). To make only the default web site remain, check information by selecting 'Start ⇒ Contol Board ⇒ Management Tools ⇒ Internet Information Services.' Ths process is to check the current sites and it can be omitted.



3. InfoU Website Existence Error

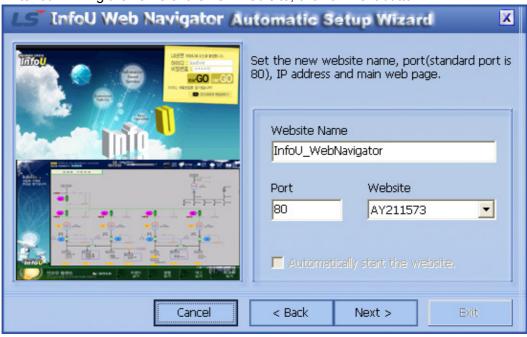
If the default web site [InfoU_WebNavigator] has been already created, the following message appears. Click on [Ok] button to install another website in addition to the default web site or click on [Cancel] button to exit.



4. Create New Website

Now, there are two menus 'Create New Website' and 'Adding Virtual Folder on Existing Website.' If any new web site has not been created before, a new web site shall be created to make the InfoU web run normally. If it is selected, click on Next button.





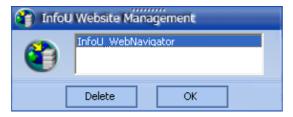
5. The InfoU user needs to input the name of the desired site and the port number. After confirming the name of the new web site, click on Next button.

6. Existing Website Name

If the user enters a name that is already used, the following message appears. Since the following message implicates that the default web site '[InfoU WebNavigator]' already exists, click on [Cancel] button to exit or click on [Ok] button to move on to the next screen.

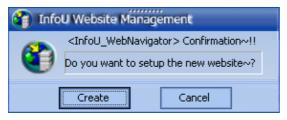


7. If the website name is already being used, the user may delete the existing website and recreate it by using the name. To delete the existing website, press [Delete] button and then, click on [OK] button to move on to the next step.



8. Confirmation

To create the new website, click on [Create] button.

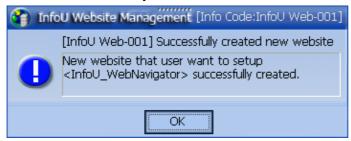


9. Setup is under progress

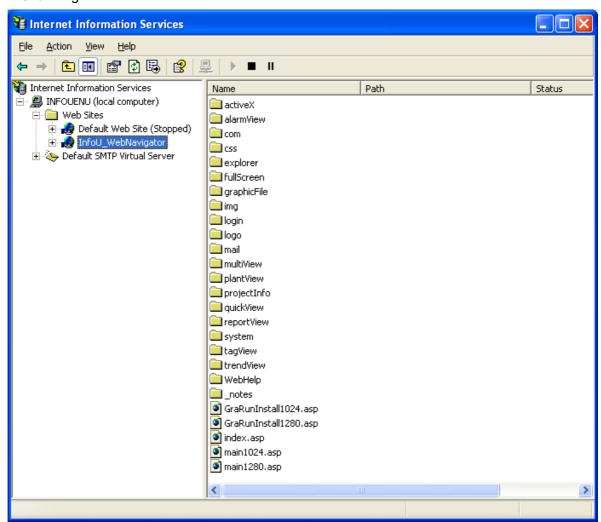
The progress of creating the new website is shown on the program bar.



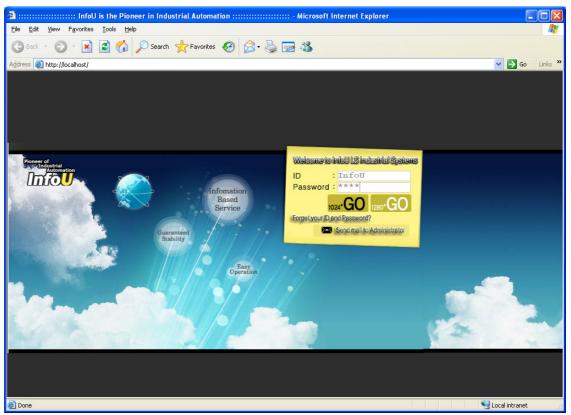
10. If the message saying that 'the new website is successfully created' appears, it moves automatically to the website.



11. If the user checks the information by selecting 'Start \Rightarrow Control Board \Rightarrow Management Tool ⇒ Internet Information Service', the user may find that the new website is successfully created, Currently, the default website is stopped and the newly created website is running.

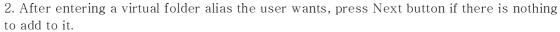


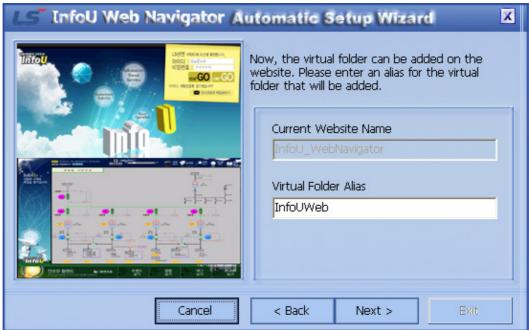
12. To correct the Explorer that has been automatically opened after the creating the new website, assess to it at 'http://localhost/index.asp or http://User ID/index.asp.'



- (4) Adding Virtual Folder on Existing Website
 - 1. A virtual folder is added to the existing website.







3. Select a directory to which the virtual folder to add.

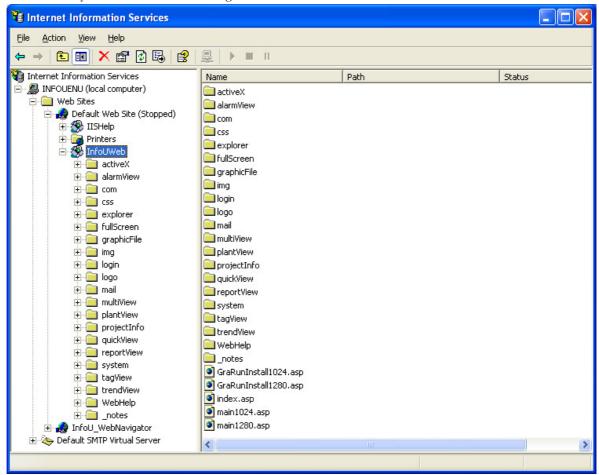


Chapter 24 Web

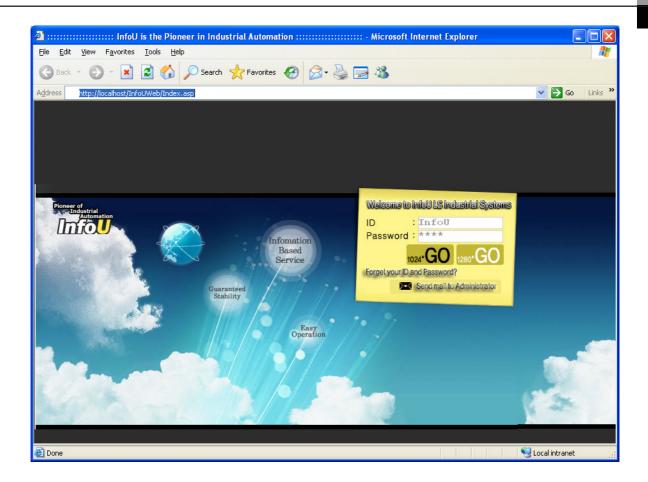
4. The messasy saying that 'the virtual folder is successfully added' appears.



5. If the user checks the information by selecting 'Start ⇒ Control Board ⇒ Management Tool ⇒ Internet Information Service', the user may find that the new InfoUWeb, a virtual folder, is successfully created on the existing website.



To access to the website, visit http://User IP/Virtual Folder Name.



(5) Get Project Information

Select a project in the InfoU HMI server to get project information such as graphic and report files. To get project information, the InfoU HMI server has been installed in advance. Through this menu, relevant cross reference information such as Graphic.ivd and Report.xsl files are obtained.

1. Select Get Project Information Select [Get Project Information] menu to get project information of the current InfoU server.



2. Get Project Information

The projet name of the current InfoU server appears and the screen turns to the next step screen upon the click on [OK].

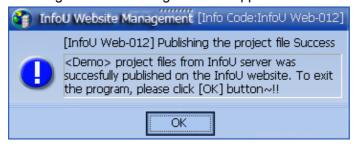


Window for the Distribution Status of Project Information
 The distribution status of the project information to the InfoU Web appears on the progress bar.



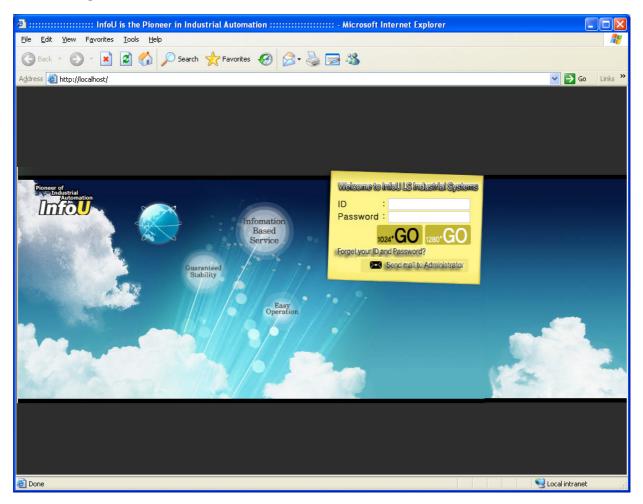
4. Publishing Project File Success

Once the current InfoU server project is successfully distributed to the InfoU Website, the following success message window appears.



InfoU Web Navigator 24.2

24.2.1 **Login Screen**



<InfoU Web Login Screen>

24.2.2 **Prerequisite**

- (1) A web server has been already created by [Navigation Management].
- (2) The InfoU WebHMI server is running.

24.2.3 Description of Login Screen Menu

Enter the user ID and password after logging in the InfoU Website on the web browser. If the user is not authorized or the password is not correct, the error message is displayed.

- (1) Screen Mode Select
 The user may select either 1024*768 or 1280*1024 mode as the optimized mode.
- (2) Send Mail to Administrator
 If the user ID or password is forgotten, send e-mail to the administrator
- (3) Error Message when login fails
 - 1. If the user ID or password is empty



2. If the user ID is not correct



3. If the password is not correct

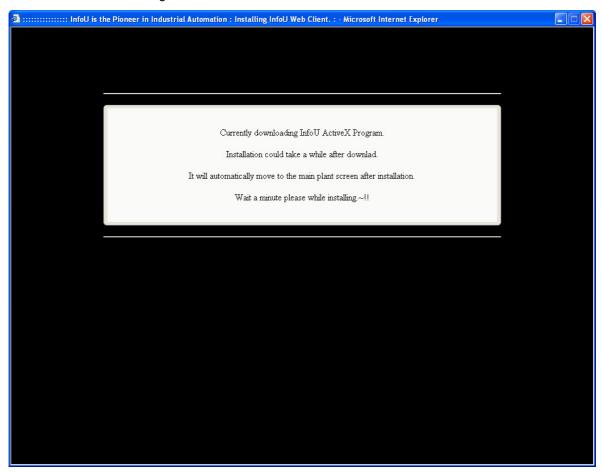


4. If the password is not correct



24.2.4 **Downloading InfoU ActiveX**

1) When the user logged in the InfoU Web Site for the first time, the user will see the following screen to download InfoU ActiveX Program.

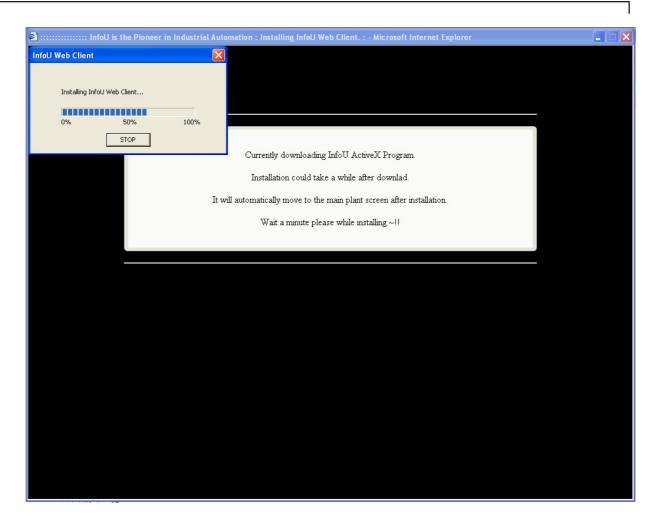


2) To install InfoU ActiveX, click [Install] to proceed with the next step of operation.

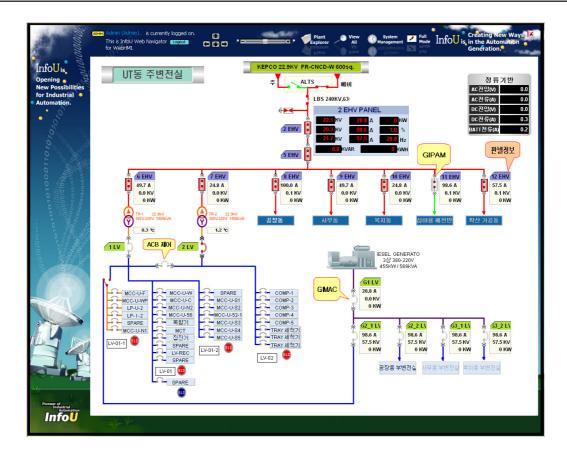


3) The progress bar will show up while processing the task and installing InfoU Web Client.

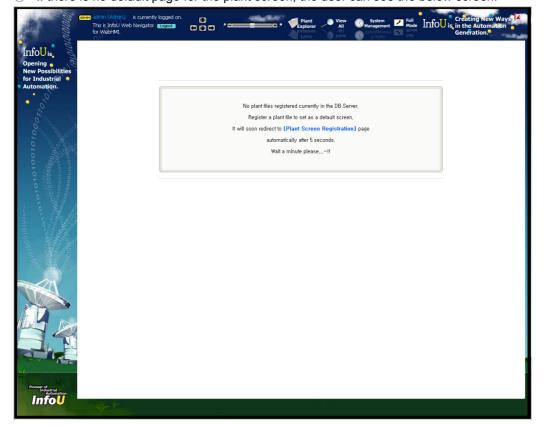
Chapter 24 Web



- 4) After installing InfoU ActiveX Program, the web site will automatically move to the main plant screen. If the user set up the default plant page, otherwise it will soon redirect to [Plant Screen Registration] page.
 - ① If there is a default page for the plant screen, the user can see the below screen.



② If there is no default page for the plant screen, the user can see the below screen.



24.2.5 Main Screen

This InfoU Plant main screen shows the relevant user's ID and the current system time after the user is successfully logged in. Click on a menu to show the functions the menu can perform on the InfoU Plant main screen. At first right after log in, the initial Plant screen which has been set up when the user registered the Plant Screen is loaded.



< InfoU Plant Screen>

24.2.6 Menu

(1) Arrow

Upper and bottom, left and right screens are previewed in snap cuts according to the hierarchy of the registered plant screen.

(2) Zoom
The currently displayed plant screen can be enlarged or shortened.

Preview

The full screen of the registered plant is seen at a glance in Thumnail method.

Plant Explorer

The plant screen registered by the current screen is shown in a form of tree.

① If there is no registered file, the user can see the below window.



② If there are registered files, the user can see the below window.

Chapter 24 Web



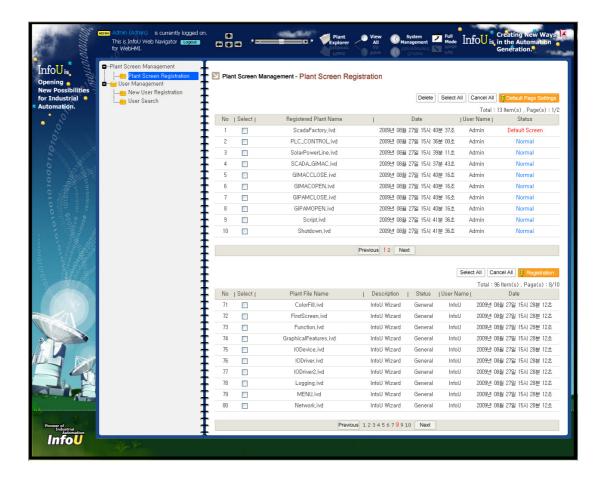
(5) View All

As a menu that makes the plant, trend, alarm, report screen better manageable, this menu show the entire menus on a single screen.

(6) System Management

Through this menu, a new user is created and the management and plant screen is registered.

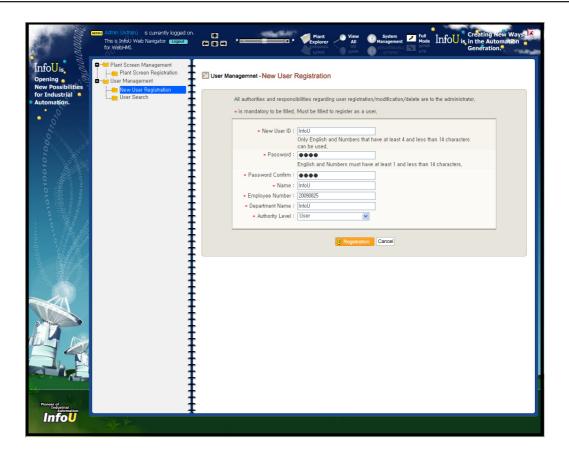
(a) Screen to register and manage the Plant Screen
The user may register or manage the initial screen with a plant screen he/she wants.

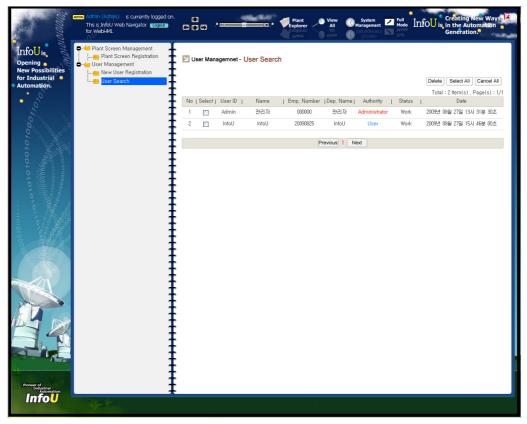


(b) User Management Screen

A new user is created and assigned to manage and register plant files.

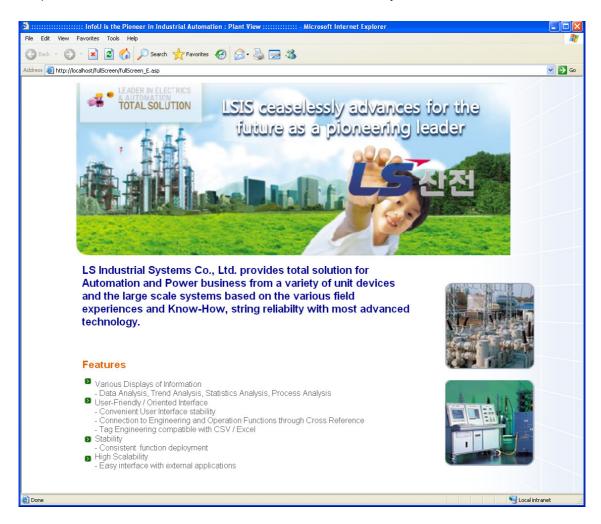
Chapter 24 Web





(7) III Mode

The plant screen is shown in a full screen without administrator's system menu.



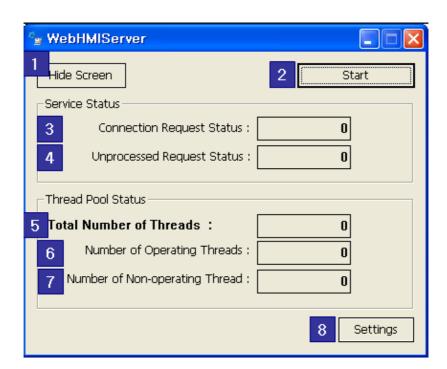
24.3 Web HMI Server

24.3.1 Introduction

The Web HMI Server is to play a role in extending to access to the InfoU HMI through web. To present HMI service through the web, the Web HMI Server has to run on the server side. To make the Web HMI Server run normally, the InfoU project should run. Also, to access to the Web HMI Server, the Internet Information Server has to be installed and relevant files needed for the HMI service should be copied and installed.

24.3.2 Operation Screen

The Web HMI Server (** iuWebHMIServer, exe*) runs automatically if the web is set up on the project option. Select 'View Screen' menu from the Tray Icons to display the following screen.



- (1) Hide Screen
 Hide this operation screen.
- (2) Stop
 The Web HMI Server stops.
- (3) Connection Request Status

 The total number of request for connection from the client is displayed.
- (4) Unprocessed Request Status

 The number of remaining requests from the client, which has not been processed, is displayed.

(5) Total Number of Threads

The total number of the created threads in the thread pool is displayed to process connection.

(6) Number of Operating Threads

The number of operating threads after being assigned the current work is displayed.

(7) Number of Non-operating Thread

The number of standby threads in the thread pool is displayed.

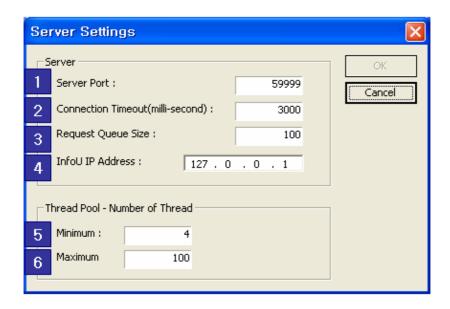
(8) Settings

This button is used when opening the server setting screen to change setting values.



If the system runs, the tray icons are created and Show Screen, Hide Screen, Settings and Exit function are performed through the pop-up menu.

24.3.3 **Server Settings**



(1) Server Port

Set up a server port

(2) Connection Timeout

Set up a timeout time (milliseconds) used for connecting to TCP/IP Socket. When setting up, make sure to take into account of some delay due to the network status.

Chapter 24 Web

- (3) Request Queue Size Set up a size of the request queue
- (4) InfoU IP Address
 As the address of the PC where the InfoU HMI is located, it has been set up with "127.0.0.1" as the default address.
- (5) Thread Pool Number of Thread: Minimum

 Set up the number of the created threads during the first running.
- (6) Thread Pool Number of Thread: Maximum

 Set up the maximum number of the available threads to be created in the thread pool

Chapter 25 Project DB Migration

Various changes in the data structure may take place during the update process such as adding new functions to the InfoU software. At this time, the data migration program is presented to synchronize the projects written under the past version with those under the current version.

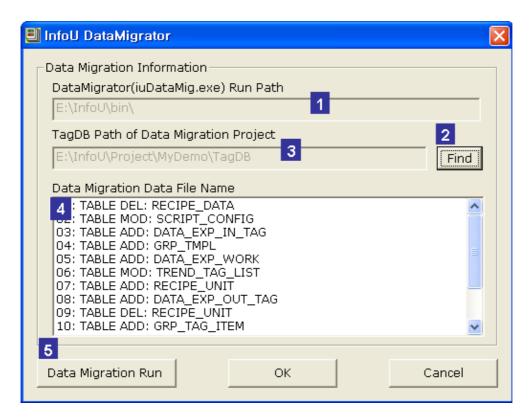
This Data Migrator(Data Migrator) application program is for Data Migration in case any physical change happens in the InfoU engineering DB(Hereinafter, "Tag DB"). The Data Migrator has the following functions:

- (1) Add new data files Create new data files in the tag DB.
- (2) Change the physical structure of the data files Create new data files that have new structures in the tag DB. Migrate the data of existing data files to the new data files.
- (3) Remove data files Delete the existing data files from the tag DB.

Notice

Because the Data Migrator physically changes the data files in the InfoU tag DB, the user should run it only after closing the currently running InfoU application program.

25.1 InfoU Data Migrator



(1) Data Migrator Run Path

Show the path on which the Data Migration running files exist.

(2) "Find" button

This button is used when finding project files (".lsp"files) subject to Data Migration.

(3) "TagDB Path of Data Migration Project"

Once the project files (".lsp"files) subject to Data Migration are found, the Data Migrator automatically calculate the tag DB path and shows it on the screen.

(4) "Data Migration Data File Name"

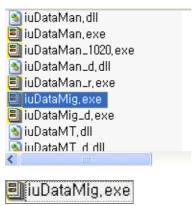
A list of the data files subject to Data Migration is shown along with migration.

(5) "Data Migration Run" button

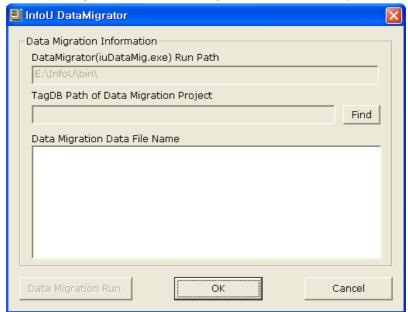
Run Data Migration.

25.2 **Data Migration Information**

- (1) Running Environemnt
 - Because the Data Migrator physically changes the data files in the InfoU tag DB, the user should run it only after closing the currently running InfoU application program.
- (2) Start
 - Double click on a Data Migrator Run file (iuDataMig.exe) from "InfoU\bin" to run the Data Migrator. (a)

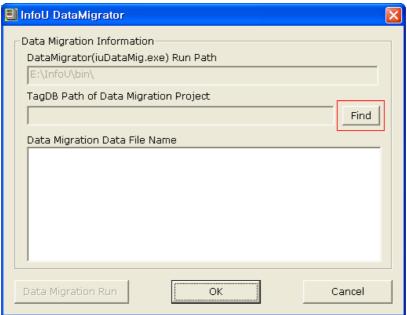


(b) Once the Data Migrator runs, the following initial screen is displayed.



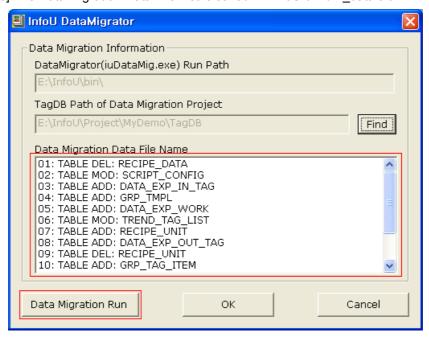
Chapter 25 Project DB Migration

- (3) Select Project
 - Press "Find" button on the main screen.



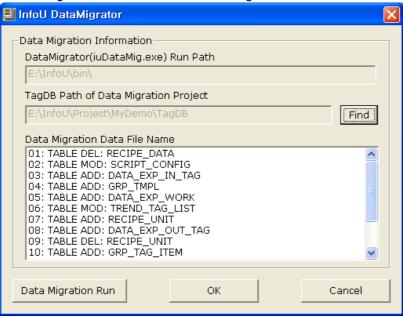
- (b) Select a project file (".lsp"file) subject to Data Migration, press "Open" button.
- The Data Migrator shows a list of the data files subject to the Data Migration on the screen and at the (c) same time, activates "Data Migration Run" button.

[Note] The Data Migration Data File List is saved in "InfoU\bin\dm_data.txt."



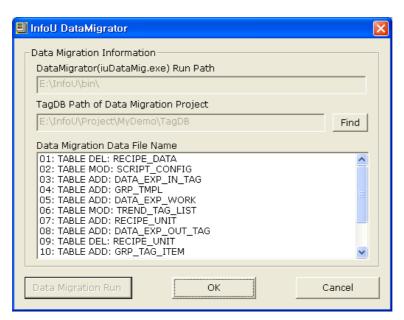
(4) Run

(a) Press"Data Migration Run"button to run Data Migration.



While Data Migration is running, a sandglass is displayed on the Data Migrator screen. (b) Once Data Migration is completed, the following message appears. Press"OK" button to close the message window.

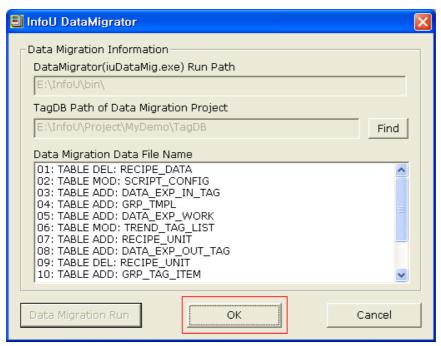




Chapter 25 Project DB Migration

(5) Exit

(a) Press"OK" (or "Cancel") button to exit the Data Migrator.



(b) Examine "Data Migration Log File" to check the performance of the Data Migrator. "Data Migration Log File"saves detailed processing information on the Data Migration. The log file information is as follows:

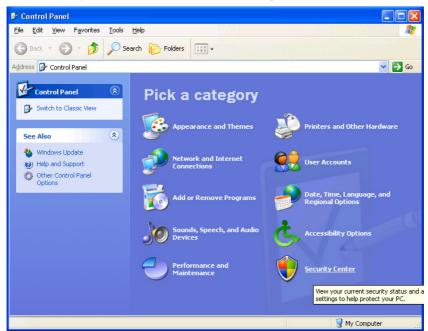
Classification		Description				
Log	File	•	"InfoU\bin"			
Location						
			Naming Rule: dm_log_YYYY.MM.DD_H24MISS			
	File		Token	Description	Length	
			dm	Abbreviated word of Data		
				Migration		
			log	Refer to Log File		
			YYYY	Year	4 digits	
Log			MM	Month	01~12	
Log Name			DD	Date	01~28/29, 30/31	
Name			H24	Hour	2 digits(00~23)	
			MI	Minute	2 digits(00~59)	
			SS	Second	2 digits(00~59)	
		Ex) dm_log_2008.01.16_151709.txt				
		[Mea	aning]This file is the log file on the Data Migration that ran at 3 o'clock 17			
minutes 9 seconds on January 16, 2008.						

Chapter 26 Redundancy and Client Server

26.1 **System Environment Settings**

26.1.1 **Cancel Network Security**

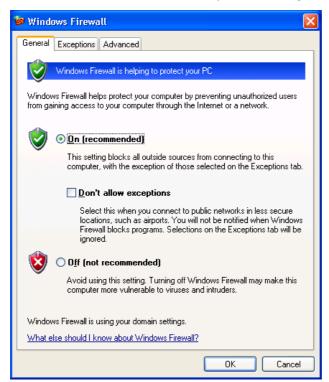
Since data shall be exchanged on the network to configure Redundancy and Client Server systems, all of the related computers have to cancel their security as follows:



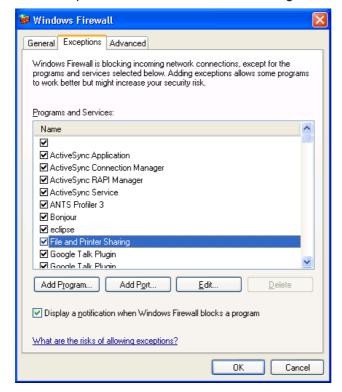
Run the security center on the control board.



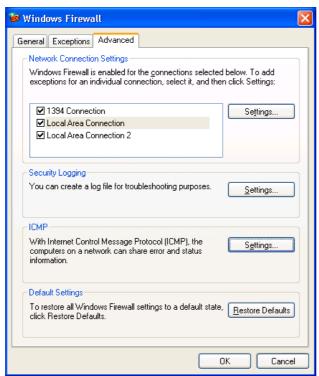
2 Run Windows Firewall on the security center settings.



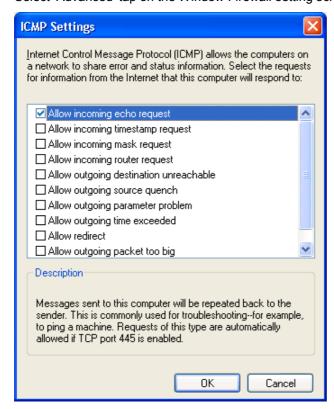
3 Allow exceptions on the Windows Firewall setting screen as follows:



4 Select "Fire and Printer Sharing" on the Exceptions tap.

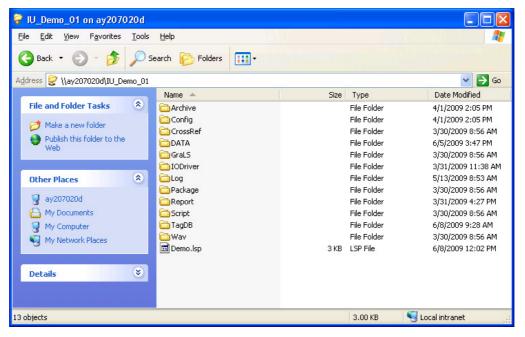


Select 'Advanced' tap on the Window Firewall setting screen and click on "Settings" button of ICMP.

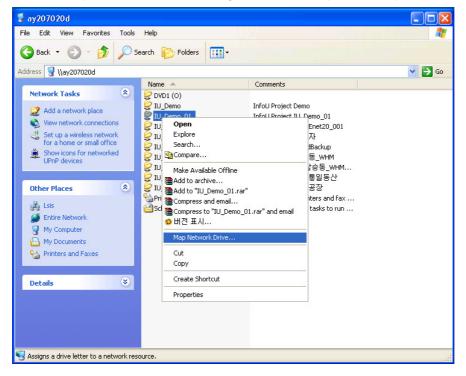


Chapter 26 Redundancy and Client Server

Select "Allow incoming echo request."



It should be possible to connect to the joint folder of the partner computer.



To make it connectable even when rebooting, set up 'Reconnect at logon' and make it reconnect to the relevant folder automatically when booting.



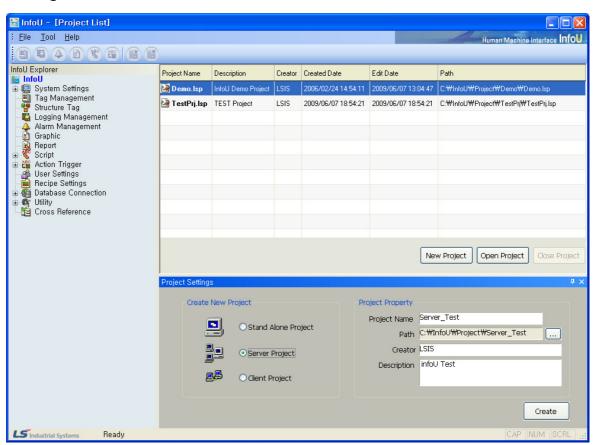
Select 'Reconnect at logon' on the Map Network Driver.

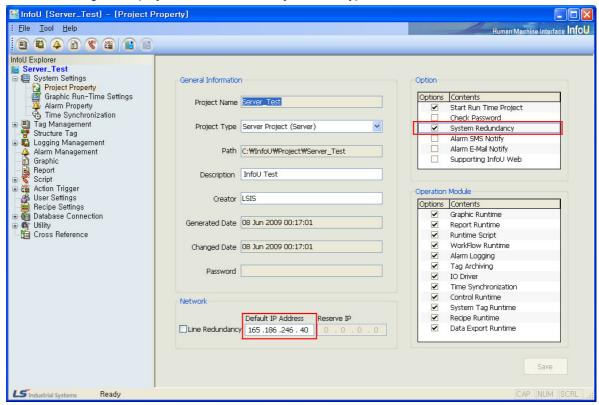
26.2 **Redundancy Settings**

26.2.1 **Procedure**

- ① Set up the system network environment allowing data communication.
- ② Create Project: Create a server project.
- Select Redundancy Option: Click on 'Redundancy' option on the project Property settings.
- ④ Set up Redundancy: Save redundancy settings.
- 5 Duplicate Redundancy Project: Duplicate the redundancy project on the current server in the partner server. At this time, the settings of the partner server are automatically changed.
- 6 Check the project on the partner server: Check whether the redundancy settings are correctly duplicated and if any item (ex> serial port, etc) is differently set up from the master server, re-set up and save it.
- Texecute real-time running.

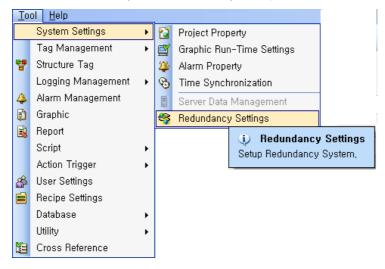
26.2.2 **Setting Screen**





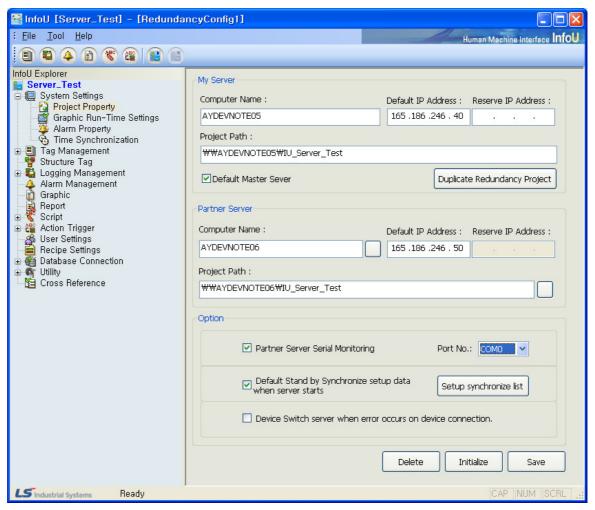
① When creating a new project, select 'Server Project' for its type.

- Also, it is possible to change and use any existing Stand Alone project to a server project on the Project Property setting screen.
- ③ Enter the current computer's IP Address on Network.
- ④ When using Line Redundancy, that is, when using two LAN cards, select 'Line Redundancy' and enter the reserve IP Address.
- Select and save 'System Redundancy' on Option.



Chapter 26 Redundancy and Client Server

6 Select 'Tool-System Settings-Redundancy Settings.'

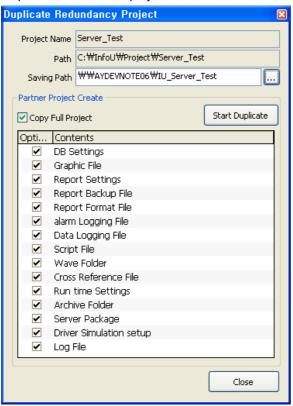


- Set up 'My Server' and 'Partner Server' on the Redundancy Configuration screen.
 - My Server
 - [1] Computer Name: Enter a computer name for identification on the computer network
 - [2] Default IP Address: Enter the default IP Address.
 - [3] Reserve IP Address: When line Redundancy option is used, enter the reserve IP Address.
 - [4] Project Path: Enter an accessible path of the current project on the network.
 - [5] Default Master Server: Assign a role of the master server to 'My Server.'
 When two servers run simultaneously, 'My Server' always runs as the master server.
 If this option is not set up, 'My Server' always runs as the Standby server when two servers run simultaneously but it runs as the Master server if 'Partner Server' does not run.

Caution

This option shall be set up for only one of two servers.

If this is set up for both of them or it is set up neither of them, errors may take place.



[6] Duplicate Redundancy Project: A screen on which the current project can be duplicated in the partner server is displayed.

- . Project Name: The current project name is displayed.
- . Path: The local Project Path of the current project is displayed.
- . Saving Path: Input a project path on which the redundancy project is saved in the partner server
- . Click on the button to display the 'Select File Backup Path' screen and select a saving location.



. Copy Full Project: All of the settings various engineering files and logging data files on the current project path are copied.

Chapter 26 Redundancy and Client Server

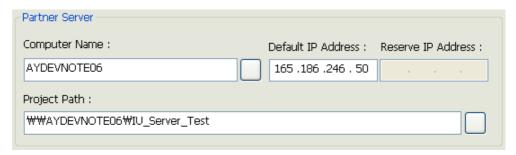
Caution

At this time, any data other than on the project path can not be duplicated.

After engineering works, it is essential to duplicate and synchronize settings between projects before starting another running.

- . After this option (Copy Full Project) is cancelled, individual items can be selected and copied.
- . Start Duplicate: Duplication is executed. It may take a lot of time depending on the file capacity. In case that duplication has been executed before, only the current files are updated.

Partner Server

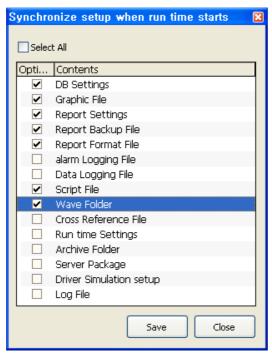


- [7] Computer Name: Enter a computer name for identification on the computer network environment.
- [8] Default IP Address: Enter the default IP Address.
- [9] Reserve IP Address: When line Redundancy option is used, enter the reserve IP Address.
- [10] Click on the button to select a computer on the network.
- [11] Project Path: Enter an accessible path of the current project on the network.
- [12] Click on the button to display the Select File Backup Path screen and select a saving location.

Option



- [13] Partner Server Serial Monitoring: Connect a serial cable to monitor the Partner Server.
- [14] Port No: Show the serial port number subject to connection.
- [15] Default Stand by Sychronize setup data when server starts: The Standby server starts after the settings of the Master server is duplicated.
- [16] Setup synchronize list: When starting, a screen to select synchronization items is displayed.



- . Select All: All of the sub items are selected.
- . After this option (Select All) is cancelled, individual items can be selected and copied.
- . Save: Save the settings.
- . Close: Close the screen.

Caution

It may take a lot of time to start the Standby server depending on the file capacity. Since most of the logging files have a large capacity, it is better to select basic settings only.

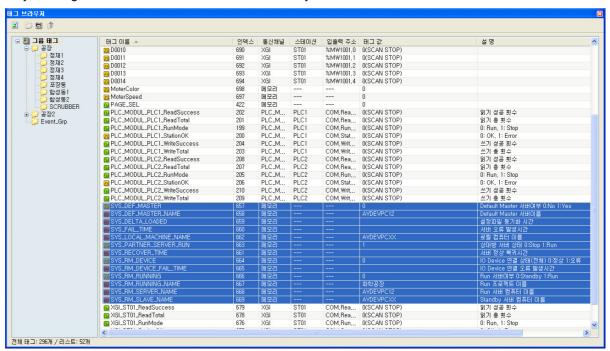
- [17] Device Switch server when error occurs on device connection: When any communication error occurs, switching action is performed.
- [18] Delete: Delete redundancy settings.
- [19] Initialize: Return the settings to the previous ones.
- [20] Save: Save the settings.

26.2.3 Realtime Run

(1) System Tag

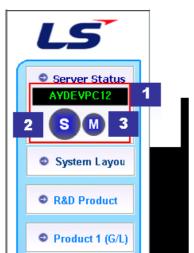
When running in the redundancy mode, a tag that shows the system status during the real-time running is created to allow the user to monitor the current redundancy running status.

① System tags and status codes related to redundancy



The above tag browser screen during the real-time running shows the status of each tag value as follows:

Tag Name	Description	Туре
SYS_DEF_MASTER	Default Master server, 0:No 1:Yes	Digital
SYS_DEF_MASTER_NAME	Default Master server name	String
SYS_DELTA_LOADED	Time to synchronize setup files	String
SYS_FAIL_TIME	Time when server errors take place	String
SYS_LOCAL_MACHINE_NA ME	Local computer name	String
SYS_PARTNER_SERVER_R UN	Partner server status, 0:Stop 1:Run	Digital
SYS_RECOVER_TIME	Time to recover the server	String
SYS_RM_DEVICE	IO Device connection status (All) 0: normal 1: fail	Digital
SYS_RM_DEVICE_FAIL_TIM E	Time when IO Device connection errors take place	String
SYS_RM_RUNNING	Running server, 0:Standby 1:Run	Digital
SYS_RM_RUNNING_NAME	Running project name	String
SYS_RM_SERVER_NAME	Running server comput name	String
SYS_RM_SLAVE_NAME	Standby server comput name	String



② By using the above system tags, each tag's status can be displayed on the graphic screen.

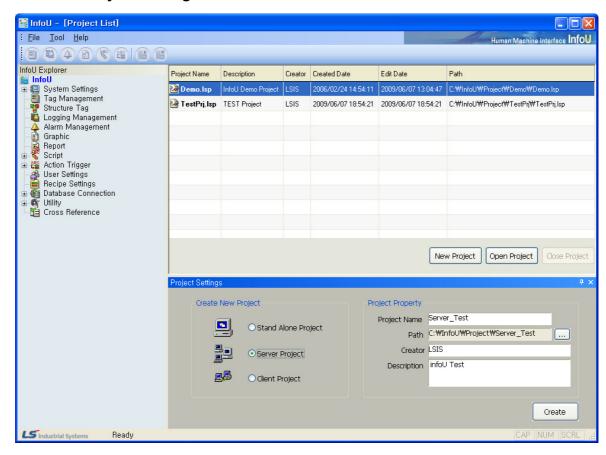
- [1]: Running Server computer name
- [2]: Current server's status and run mode
- [3]: Partner server's status and run mode

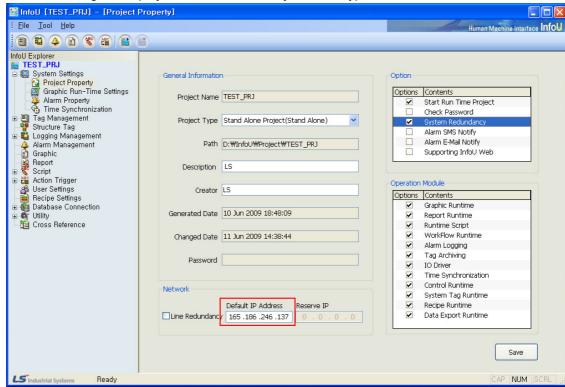
26.3 Client-Server Settings

26.3.1 Procedure

- ① Set up the system network environment allowing data communication.
- 2 Create Server Project: Create a server project.
- ③ Create Server Data File: Create a server data file that saves information on IP Address, Redundancy Information, Joint folder name to connect to the client server project.
- 4 Create Client Project: Create a client project.
- ⑤ Load Server Data File: Load the server data file to connect from the client project.
- 6 Execute server real-time running.
- ② Execute client real-time running.

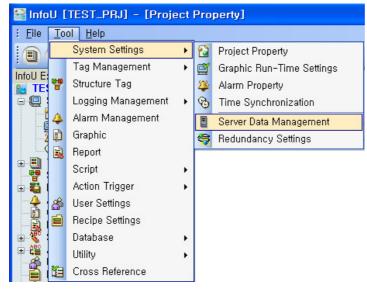
26.3.2 Server Project Settings





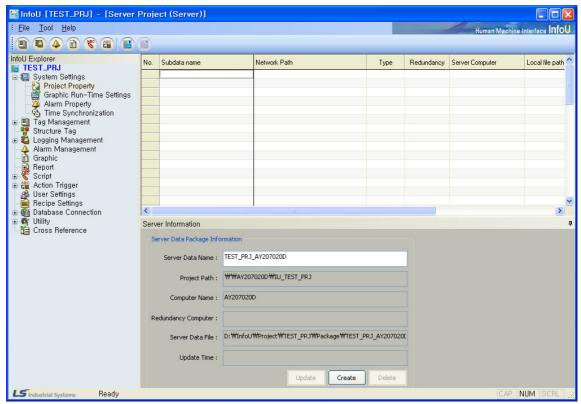
① When creating a new project, select 'Server Project' for its type.

- ② Also, it is possible to change and use any existing Stand Alone project to a server project on the Project Property setting screen.
- ③ Enter the current computer's IP Address on Network.
- When using Line Redundancy, that is, when using two LAN cards, select 'Line Redundancy' and enter the reserve IP Address.

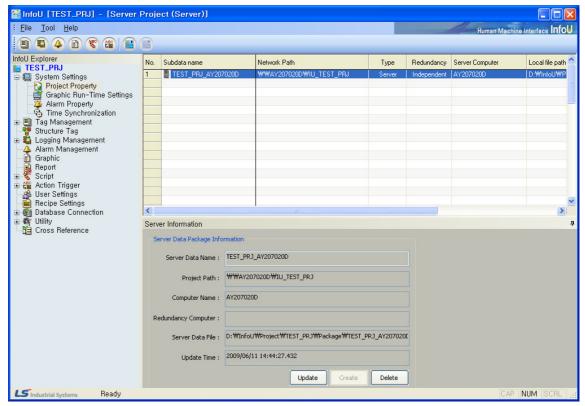


Chapter 26 Redundancy and Client Server

⑤ Select 'Tool-System Settings-Server Data Management.'



⑥ In case that any server data has not been saved, a server data name can be entered and the rest items are displayed automatically and the 'Create' button becomes active.

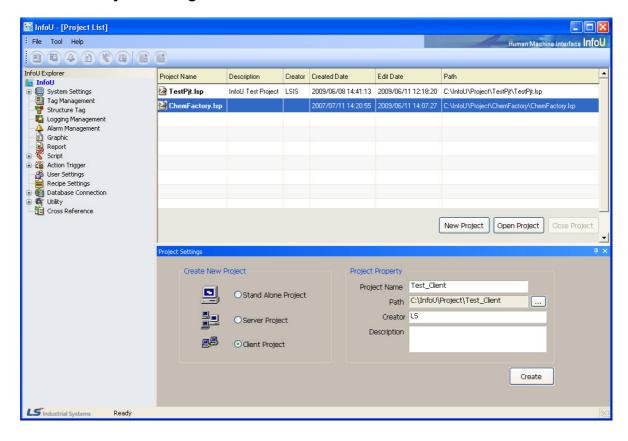


- (7) If "Create' button is clicked, the server data file is created and the associated information is displayed on the screen.
- If 'Update' button is clicked, the changed engineering settings are updated and saved.
- 9 If 'Delete' button is clicked, the server data file is deleted

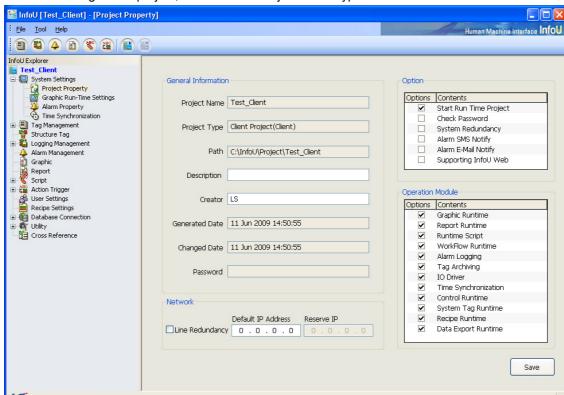


10 The file is created in "Package" among the project folders. At this time, extension "sdp" is used.

26.3.3 Client Project Settings

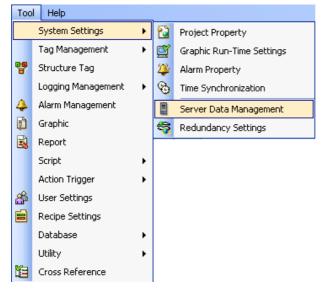


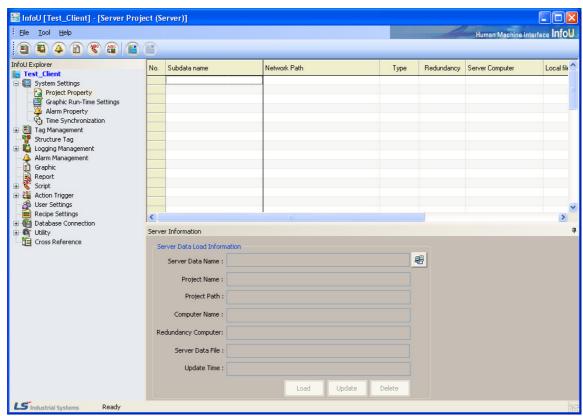
Chapter 26 Redundancy and Client Server



① When creating a new project, select 'Client Project' for its type.

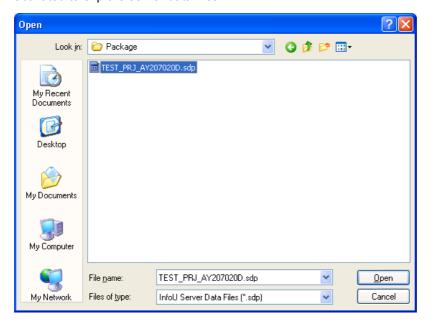
- 2 The client project type can not be changed.
- ③ Enter the current computer's IP Address on Network.
- ④ When using Line Redundancy, that is, when using two LAN cards, select 'Line Redundancy' and enter the reserve IP Address.





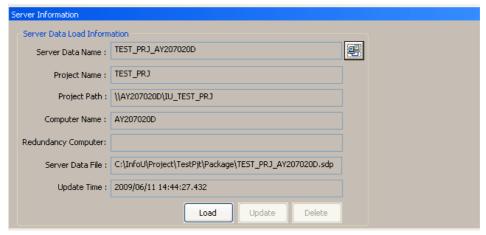
⑤ Select 'Tool-System Settings-Server Data Management.'

⑤ In case that any server data has not been saved, any information is not displayed and a button is activated to explore server data files.

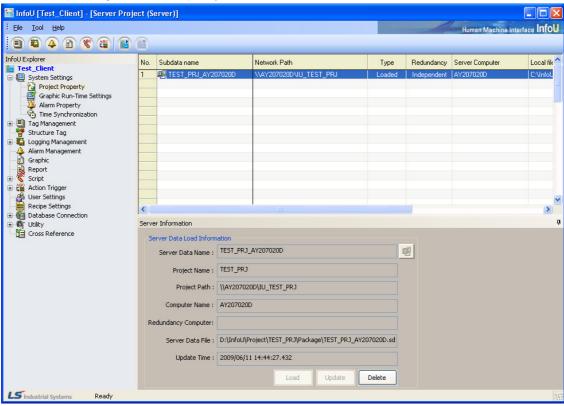


Chapter 26 Redundancy and Client Server

Select a server data file to connect to.



- The information loaded from the selected server data file is displayed and 'Load' button become active
- Click on 'Load' to copy the needed server data file to the local computer. At this time, it may take a lot
 of time depending on the file capacity.



Chapter 26 Redundancy and Client Server

① Once loading and duplicating the server data file are completed, information is displayed on the list and 'Delete' button becomes active.



- ① The file is created in "Package" among the project folders.
- ① If 'Delete' button is clicked, the server data file is deleted. At this time, the copied filed in the "Package" folder are not deleted.

Appendix1. Visual Basic Script Help

Appendix 1.1 Script Function of Microsoft Script Engine

Visual Basic Script (VB Script)

Function Name	Form
Abs	Function Abs(Number)
Array	Function Array(ArgList)
Asc	Function Asc(String As String)
Atn	Function Atn(Number As Double) As Double
CBool	Function CBool(Expression) As Double
CByte	Function CByte(Expression) As Byte
CCur	Function CCur(Expression) As Currency
CDate	Function CDate(Expression) As Date
CDbl	Function CDbl(Expression) As Double
Chr	Function Chr(CharCode As Long
CInt	Function CInt(Expression) As Integer
CLng	Function CLng(Expression) As Long
Cos	Function Cos(Number As Double) As Double
CreateObject	Function CreateObject(Class As String)
CSng	Function CSng(Expression) As Single
CStr	Function CStr(Expression) As String
Date	Function Date() As Date
DateAdd	Function DateAdd(Interval As String, Number As Double, Date)
DateDiff	Function DateDiff(Interval As String, Date1, Date2, [FirstDayOfWeek As VbDayOfWeek = vbSunday], [FirstWeekOfYear As VbFirstWeekOfYear = vbFirstJan1])
DatePart	Function DatePart(Interval As String, Date, [FirstDayOfWeek As VbDayOfWeek = vbSunday], [FirstWeekOfYear As VbFirstWeekOfYear = vbFirstJan1])
DateSerial	Function DateSerial(Year As Integer, Month As Integer, Day As Integer)
DateValue	Function DateValue(Date As String)
Day	Function Day(Date)
Eval	Function Eval(Expression As String)
Ехр	Function Exp(Number As Double) As Double
Filter	Function Filter(InputStrings As Array, Value As String [, Include As Boolean [,Compare As VbCompareMethod = vbBinaryCompare]]])
Fix	Function Int(Number)
FormatCurrency	Function FormatCurrency(Expression[,NumDigitsAfterDecimal [, [,UseParensForNegativeNumbers [,GroupDigits]]]])
FormatDateTime	Function FormatDateTime(Date[, NamedFormat])
FormatNumber	Function FormatNumber(Expression [,NumDigitsAfterDecimal [,IncludeLeadingDigit [,UseParensForNegativeNumbers [,GroupDigits]]]])

Appendix1. Visual Basic Script Help

Function Name	Form	
FarmetDareaut	Function FormatPercent(Expression[,NumDigitsAfterDecimal	
FormatPercent	[,IncludeLeadingDigit [,UseParensForNegativeNumbers [,GroupDigits]]]])	
GetObject	Function GetObject([PathName], [Class])	
Hex	Function Hex(Number)	
Hour	Function Hour(Time)	
land Davi	Function InputBox(Prompt, [Title], [Default], [XPos], [YPos], [HelpFile],	
InputBox	[Context]) As String	
1.01	Function InStr([Start], [String1], [String2], [Compare As VbCompareMethod	
InStr	= vbBinaryCompare])	
La OtaD acc	Function InStrRev([Start], [String1], [String2], [Compare As	
InStrRev	VbCompareMethod = vbBinaryCompare])	
Int	Function Int(Number)	
IsArray	Function IsArray(VarName) As Boolean	
IsDate	Function IsDate(Expression) As Boolean	
IsEmpty	Function IsEmpty(Expression) As Boolean	
IsNull	Function IsNull(Expression) As Boolean	
IsNumeric	Function IsNumeric(Expression) As Boolean	
IsObject	Function IsObject(Expression) As Boolean	
Join	Function Join(List As Array[, Delimiter As String])	
LBound	Function LBound(Arrayname[, Dimension As Number])	
LCase	Function LCase(String)	
Left	Function Left(String, Length As Long)	
Len	Function Len(Expression)	
	Function LoadPicture([filename], [widthDesired As Long], [heightDesired	
LoadPicture	As Long], [flags As LoadPictureConstants = Default]) As IpictureDisp	
Log	Function Log(Number As Double) As Double	
LTrim	Function LTrim(String)	
Mid	Function Mid(String, Start As Long, [Length])	
Minute	Function Minute(Time)	
Month	Function Month(Date)	
MonthName	Function MonthName(Month As Number[, Abbreviate As Boolean])	
	Function MsgBox(Prompt, [Buttons As VbMsgBoxStyle = vbOKOnly],	
MsgBox	[Title], [HelpFile], [Context]) As VbMsgBoxResult	
Now	Function Now()	
Oct	Function Oct(Number)	
	Function Replace(Expression As String, Find As String, ReplaceWith As	
Replace	String [, Start As Number[, Count As Number[, Compare As	
	VbCompareMethod = vbBinaryCompare]]])	
Right	Function Right(String, Length As Long)	
Rnd	Function Rnd([Number]) As Single	
RGB	Function RGB(red, green, blue)	
Round	Function Round(Expression[, NumDecimalPlaces])	
RTrim	Function RTrim(String)	
ScriptEngine	Function ScriptEngine() As String	
ScriptEngineBuildVersion	Function ScriptEngineBuildVersion As Number	
Completing in Country Croision	1 anotion completing moduling vorsion 7 to Number	

Function Name	Form	
ScriptEngineMajorVersio	Function ScriptEngineMajorVersion Function As Number	
n		
ScriptEngineMinorVersio	Function ScriptEngineMinorVersion Function As Number	
n		
Second	Function Second(Time)	
Sgn	Function Sgn(Number)	
Sin	Function Sin(Number As Double) As Double	
Space	Function Space(Number As Long)	
Split	Split(Expression[, Delimiter As String [, Count As Number[, Compare As Number]]])	
Sqr	Function Sqr(Number As Double) As Double	
	Function StrComp(String1, String2, [Compare As VbCompareMethod =	
StrComp	vbBinaryCompare])	
String	Function String(Number As Long, Character)	
StrReverse	Function StrReverse(String)	
Tan	Function Tan(Number As Double) As Double	
Time	Function Time() As Date	
Time Conicl	Function TimeSerial(Hour As Integer, Minute As Integer, Second As	
TimeSerial	Integer)	
TimeValue	Function TimeValue(Time As String)	
Trim	Function Trim(String)	
TypeName	Function TypeName(VarName) As String	
UBound	Function UBound(Arrayname[, Dimension As Number])	
UCase	Function UCase(String)	
VarType	Function VarType(VarName) As VbVarType	
Mookdoy	Function WeekDay(Date, [FirstDayOfWeek As VbDayOfWeek =	
Weekday	vbSunday])	
WeekdayName	Function WeekdayName(Weekday As Number, Abbreviate As Boolean,	
vveekuayivaille	FirstDayOfWeek As Number)	
Year	Function Year(Date)	

Abs Function

Description

The absolute value of a number is returned.

Syntax

Abs(number)

The factor of a number should be a valid expression. If the number has the value zero, null is returned. And if the number is a non-initialized variable, 0 is returned.

Reference

The absolute value of a number shows only its size without code. For example, in both Abs(-1) and Abs(1), 1 is returned.

In the following example, Abs Function calculates the absolute value of a number.

Dim MyNumber

MyNumber = Abs(50.3) '50.3 is returned.

Appendix1. Visual Basic Script Help

MyNumber = Abs(-50.3) '50.3 is returned.

Array Function

Description

A Variant including Array is returned.

Syntax

Array(arglist)

The arglist factor consists of values distinguished with pauses and it designates the Array factor included in the Variant to this value. If any factor is not designated, Array whose length is 0 is created.

Reference

This notation is used when referring to variable elements consisting of variable names next to a bracket that has index numbers indicating essential elements. For example, the first statement creates variable A equal to Variant. The second statement designates Array to variable A. The last statement designates a value that includes the variable's second Array element.

Dim A As Variant A = Array(10,20,30)B = A(2)

The minimum limit of the Array created with this Array Function is decided with the minimum limit that Option Base statement designates.

The lower limit of the Array created with this Array Function is decided with the lower limit that Option Base statement designates or the name of the Array Function is decided with the name of the Form Library (ex: VBA.Array). Once the name is designated with the Form Library, this Array Function is not influenced by Option Base statement anymore.

Even Variants not declared by Memo Array can include Array. Variant variables can include all forms of Array except fixed-length string forms and customized forms. Even though Variant Form including Array is different from Array, whose element is of variant form in terms of concept, the elements of the relevant Array can be equally accessed.

Asc Function

Description

ANSI character code corresponding to the first character of a string is returned.

Syntax

Asc(string)

The string factor should be a valid character expression. If there is no character in the string, Runtime errors take place.

Reference

Atn returns the ANSI character code of the first character of each string as seen in the following example.

Dim MyNumber

MyNumber = Asc("A") '65 is returned.

MyNumber = Asc("a") '97 is returned.

MyNumber = Asc("Apple") '65 is returned.

Memo

AscB Function uses Byte data that the string has. AscB does not return the character code of the first character but the first Byte of the character. AscW Function is provided for 32bit platforms using Unicode characters. Since this Function returns the character codes of Unicode, it is not necessary to convert Unicode to ANSI code.

Atn Function

Description

The arctangent of a number is returned.

Syntax

Atn(number)

The number factor should be a valid expression.

Reference

This Atn Function uses the ratio between lengths of two sides of a right triangle and returns alternate angles in radian unit. This ratio is obtained by dividing the length of the side facing the relevant angle by the length of the side adjacent to this angle. The results are ranged between pi/2 and pi/2 radian.

To convert angle to radian, multiply angle by pi/180 and to convert radian to angle, multiply radian by 180/pi.

In the following example, Atn calculates pi value.

Dim pi

pi = 4 * Atn(1) 'Calculate pi value.

Memo

Atn Function is a reverse triangle function of Tan that returns the ratio between two sides of a right triangle by using angle as a factor. Do not confuse Atn with cotangent, a reverse tangent (1/tangent).

CBool Function

Description

The expression converted to a Boolean subtype of Variant is returned.

Syntax

CBool(expression)

The expression factor should be a valid expression.

Reference

If the expression is 0, False is returned or if not, True is returned. This expression can not be interpreted into number values and Runtime errors take place.

In the following example, CBool Function converts the expression to Boolean. If the expression is not 0, CBool returns True or if it is 0, It returns False

Appendix1. Visual Basic Script Help

Dim A, B, Check

A = 5: B = 5 'Initialize variables.
Check = CBool(A = B) 'Check gets True.

A = 0 ' Define a variable. Check = CBool(A) ' Check gets False.

CByte Function

Description

The expression converted to a Byte subtype of Variant is returned.

Syntax

CByte(expression)

The expression factor should be a valid expression.

Reference

Typically, if a code is created with a subtype of conversion function, the operation result can be seen in a particular data type rather than in a basic data type. For example, If Cbyte is used for call, single precision, double precision or integer arithmetic operation, Byte arithmetic operation is carried out.

If CByte Function is used, other types of data can be converted to a Byte subtype widely used throughout the world. For example, symbols identified with different digit numbers such as symbols identified with a number in the thousands are properly recognized according to the locale settings of the computer.

If an expression exceeds the tolerance range of Byte subtype, errors take place. In the following example, Cbyte Function converts the expression to Byte.

Dim MyDouble, MyByte

MyDouble = 125.5678 'MyDouble is equal to Double.

MyByte = CByte(MyDouble) 'MyByte gets value 126.

CCur Function

Description

The expression converted to a Currency subtype of Variant is returned.

Syntax

CCur(expression)

The expression factor should be a valid expression.

Reference

Typically, if a code is created with a subtype of conversion function, the operation result can be seen in a particular data type rather than in a basic data type. For example, if Ccur is used for an integer arithmetic operation, a call arithmetic operation is executed.

If CCur Function is used, other types of data can be converted to a widely-used Currency subtype throughout the world. For example, symbols identified with different digit numbers such as symbols identified with a number

in the thousands are properly recognized according to the locale settings of the computer. In the following example, CCur Function converts the expression to Currency.

Dim MyDouble, MyCurr

MyDouble = 543.214588

' MyDouble is equal to Double.

MyCurr = CCur(MyDouble * 2)

'The value multiplied MyDouble by 2 (1086.429176) is converted to Currency

(1086.4292).

CDate Function

Description

The expression converted to a Date subtype of Variant is returned.

Syntax

CDate(date)

The date factor should be a valid expression.

Reference

If IsDate Function is used, it is decided whether to convert a date factor into date or time. CDate Function recognizes numbers within the tolerance range of date as well as date literal and time literal. If number is converted to date, the integer part is converted to date. The fractional part of number is converted to time of the day that starts from the midnight.

CDate Function recognizes date type according to the computer's locale settings. If any unrecognized date type is provided, it is impossible to decide a correct order of year, month and day. It is also impossible to recognize the marked date type in detail even though there is a corresponding string to the day of the week.

In the following example, CDate Function converts a string to date. Typically, it is not good to use the date and hard coded time with strings as seen in this example. Instead, it is better to use date and time literal such as #10/19/1962# or #4:45:23 PM#,

MyDate = "October 19,1962"

' Decide date.

MyShortDate = CDate(MyDate)

' It is converted to Date data type.

MyTime = "4:35:47 PM"

' Decide time

MyShortTime = CDate(MyTime)

' It is converted to Time data type.

CDbl Function

Description

The expression converted to a Double subtype of Variant is returned.

Syntax

CDbl(expression)

The expression factor should be a valid expression.

Reference

Typically, if a code is created with a subtype of conversion function, the operation result can be seen in a particular data type rather than in a basic data type. For example, if CDbl or CSng is used for a call or integer arithmetic operation, single or double precision arithmetic operation is carried out.

If CDbl Function is used, other types of data can be converted to a widely-used Double subtype throughout the world. For example, symbols identified with different digit numbers such as symbols identified with a number in the throusands are properly recognized according to the locale settings of the computer.

In the following example, CDbl Function converts the expression to Double.

```
Dim MyCurr, MyDouble

MyCurr = CCur(234.456784) 'MyCurr is equal to Currency.

MyDouble = CDbl(MyCurr * 8.2 * 0.01) 'The result is converted to Double.
```

Chr Function

Description

A character related to the designated ANSI character code is returned.

Syntax

Chr(charcode)

The charcode factor is a number to identify characters.

Reference

Numbers 0 through 31 are equal to the ASCII standard codes that can not be printed. For example, Chr(10) returns line feed characters. In the following example, Chr Function returns the character related to the designated ANSI character.

```
Dim MyChar

MyChar = Chr(65) 'A is returned.

MyChar = Chr(97) 'a is returned.
```

MyChar = Chr(62) '> is returned.

MyChar = Chr(37)

' %i s returned.

Memo

ChrB Function uses Byte data that the string has. 1Byte is always returned rather than 1Byte or 2Byte characters. ChrW Function is provided for 32bit platforms using Unicode characters. Since the factor of this Function is a Unicode character code, it is not necessary to convert ANSI to Unicode.

CInt Function

Description

The expression converted to an Integer subtype of Variant is returned.

Syntax

CInt(expression)

The expression factor should be a valid expression.

Reference

Typically, if a code is created with a subtype of conversion function, the operation result can be seen in a particular data type rather than in a basic data type. For example, if Clnt or CLng Function is used for call, single precision or double precision arithmetic operation, integer arithmetic operation is carried out.

If Clnt Function is used, other types of data can be converted to a widely-used Integer subtype throughout the world. For example, symbols identified with different digit numbers such as symbols identified with a number in the throusands are properly recognized according to the locale settings of the computer.

If an expression exceeds the tolerance range of Integer subtype, error takes place.

In the following example, Clnt Function converts the value to integer.

Dim MyDouble, MyInt

MyDouble = 2345.5678 'MyDouble is equal to Double.

MyInt = CInt(MyDouble) 'MyInt gets value 2346.

Memo

Cint is different from Fix or Int Function that does not round off but cut away decimal points. If the decimal point is exactly 0.5, Clnt Function always rounds off the nearest even number. For example, 0.5 is rounded off to 0 and 1.5 is rounded off to 2.

CLng Function

Description

The expression converted to a Long subtype of Variant is returned.

Syntax

CLng(expression)

The expression factor should be a valid expression.

Reference

Typically, if a code is created with a subtype of conversion function, the operation result can be seen in a particular data type rather than in a basic data type. For example, if Clnt or CLng Function is used for call, single precision or double precision arithmetic operation, integer arithmetic operation is carried out.

If CLng Function is used, other types of data can be converted to a widely-used Long subtype throughout the world. For example, symbols identified with different digit numbers such as symbols identified with a number in the throusands are properly recognized according to the locale settings of the computer.

If an expression exceeds the tolerance range of Integer subtype, errors take place.

In the following example, CLng Function converts the value to Long.

Dim MyVal1, MyVal2, MyLong1, MyLong2

MyVal1 = 25427.45: MyVal2 = 25427.55 'MyVal1 and MyVal2 are Doubles.

MyLong1 = CLng(MyVal1) 'MyLong1 gets value 25427.
MyLong2 = CLng(MyVal2) 'MyLong2 gets value 25428.

Memo

CLng Function is different from Fix or Int Function that does not round off but cut away decimal points. If the decimal point is exactly 0.5, CLng Function always rounds off the nearest even number. For example, 0.5 is rounded off to 0 and 1.5 is rounded off to 2.

Cos Function

Description

The cosine of an angle is returned.

Syntax

Cos(number)

The number factor should be a valid expression showing the angle in a radian.

Reference

Cos Function uses angles and returns the ratio between two sides of a right triangle that is, the ratio between the length of the adjoining side and the length of the oblique side of each angle. The results are between -1 through 1.

To convert angle to radian, multiply angle by pi/180 and to convert radian to angle, multiply radian by 180/pi. In the following example, Cos Function returns the cosine value of the angle.

Dim MyAngle, MySecant

MyAngle = 1.3 'Define an angle in a radian value.

MySecant = 1 / Cos(MyAngle) 'Caculate the Secant value. .

CreateObject Function

Description

A reference for an automatic object is created and returned.

Syntax

CreateObject(class)

The class factor uses Syntax servername.typename and gets the following factors.

Description on factor

servername: the name of an application program providing objects.

typename: The type or class of an object to be produced.

Reference

An automatic server provides at least more than one object type. For example, the word processing program provides application program objects, document object and toolbar objects.

To create automatic objects, designate the returned object from CreateObject Function as an object variable.

Dim ExcelSheet

Set ExcelSheet = CreateObject("Excel.Sheet")

This code starts an application program while creating the relevant object, that is, Microsoft Excel spread sheet. Once the object is created, the object can be referred by using customized variables. In the following example, properties and methods of the new object are accessed by using ExcelSheet object variables and Excel objects including Application objects and ActiveSheet.Cells collections. The application examples are as follows:

' Make Excel program appear through Application objects.

ExcelSheet.Application.Visible = True

' Put the relevant text on the first cell of the sheet.

ExcelSheet.ActiveSheet.Cells(1,1). Value = "Column A, Row 1."

'Save the sheet.

ExcelSheet.SaveAs "C:\DOCS\TEST.XLS"

' Close Excel program by using 'Quit' method of the Application object.

ExcelSheet.Application.Quit

'Release the object variable.

Set ExcelSheet = Nothing

CSng Function

Description

The expression converted to a Single subtype of Variant is returned.

Syntax

CSng(expression)

The expression factor should be a valid expression.

Reference

Typically, if a code is created with a subtype of conversion function, the operation result can be seen in a particular data type rather than in a basic data type. For example, if CDbl or CSng is used for a call or integer arithmetic operation, single or double precision arithmetic operation is carried out.

If CSng Function is used, other types of data can be converted to a widely-used Single subtype throughout the world. For example, symbols identified with different digit numbers such as symbols identified with a number in the throusands are properly recognized according to the locale settings of the computer.

If an expression exceeds the tolerance range of Byte subtype, errors take place.

In the following example, CSng Function converts the value to Single.

Dim MyDouble1, MyDouble2, MySingle1, MySingle2 'MyDouble1 and MyDouble2 are Doubles.

MyDouble1 = 75.3421115: MyDouble2 = 75.3421555

MySingle1 = CSng(MyDouble1) 'MySingle1 gets value 75.34211.

MySingle2 = CSng(MyDouble2) 'MySingle2 gets value 75.34216.

CStr Function

Description

The expression converted to a String subtype of Variant is returned.

Syntax

CStr(expression)

The expression factor should be a valid expression.

Reference

Typically, if a code is created with a subtype of conversion function, the operation result can be seen in a particular data type rather than in a basic data type. For example, if CStr is used, the result is expressed in a String.

If CStr Function is used instead of Str Function, other types of data can be converted to a widely-used String subtype throughout the world. For example, symbols identified with different digit numbers are properly recognized according to the computer's locale settings.

The data used in an expression decide the returned results as seen in the table.

Expression Data	CStr Returned Result
Boolean	String containing True or False
Date	String containing simplified date by computer settings.
Null	Runtime error
Empty	String whose length is 0 ("")
Error	String showing error number next to word 'Error'
Other	String containing numbers

In the following example, CStr Function converts the value to String.

Dim MyDouble, MyString

MyDouble = 437.324 'MyDouble is equal to Double.

MyString = CStr(MyDouble) 'MyString gets value "437.324."

Date Function

Description

The Date of the current system is returned.

Syntax

Date

Reference

In the following example, Date Function returns the Date of the current system.

Dim MyDate

MyDate = Date 'MyDate gets the date of the current system.

DateAdd Function

Description

The Date that adds the designated time interval is returned.

Syntax

DateAdd(interval, number, date)

DateAdd Function Syntax has following factors.

Description on Factor

interval	A mandatory factor. A literal expression indicating the date that adds interval. For the value,
interval	refer to the following settings.
	A mandatory factor. A numerical expression of the interval number to be added. In the
number	numerical expression, a positive number refers to a future date while a negative number
	refers to a past date.
date	A mandatory factor. Variant or literal indicating the date that adds interval.

Settings

The value of the interval factor is as follows:

Setting	Description
уууу	Year
q	Quarter
m	Month
у	Year(As of one year)
d	Day
W	Weekday
ww	Week(As of one year)
h	Hour
n	Minute
S	Second

Reference

If DateAdd Function is used, it is possible to add the designated time interval to the date or subtract it from the date. For example, if DateAdd Function is used, it is possible to calculate the date that will be in 30 days from today or the time that will be in 45 minutes from the current time. To add a certain day to the date, use year("y"), day("d") or weekday("w").

In DateAdd Function, any wrong date is not returned. In the following example, one month is added to January 31.

NewDate = DateAdd("m", 1, "31-Jan-95")

In this case, DateAdd Function returns 28-Feb-95 rather than 31-Feb-95. If the date is 31-Jan-96, 29-Feb-96 is returned because year 1996 is a leap year.

If the date to be calculated is before AD 100, errors take place.

If it is not a number of Long value, the nearest integer shall be rounded off before calculation.

DateDiff Function

Description

The difference between two dates is returned.

Syntax

DateDiff(interval, date1, date2 [,firstdayofweek[, firstweekofyear]])

DateDiff Function Syntax has following factors.

Factor	Description	
interval	A mandatory factor. A <u>literal expression</u> of <i>interval</i> used to calculate the difference between Date 1 and Date 2. For the value, refer to the following settings.	
date1, date2	A mandatory factor. Two dates used to calculate in the date expression.	
firstdayofweek	An optional factor. Sunday is deemed as the day of the week if any Constant is not designated to indicate which day of the week. For the value, refer to the following settings.	

	An optional factor. A constant indicating the first week of the year. If not in	idicated,
firstweekofyear	the week that contains January 1 is deemed as the first week in the year.	For the
	value, refer to the following settings.	

Description on Factor

interval: A mandatory factor. A literal expression of *interval* used to calculate the difference between Date 1 and Date 2. For the value, refer to the following settings.

date1, date2: A mandatory factor. Two dates used to calculate in the date expression.

firstdayofweek: An optional factor. Sunday is deemed as the day of the week if any Constant is not designated to indicate which day of the week. For the value, refer to the following settings.

firstweekofyear: An optional factor. A constant indicating the first week of the year. If not indicated, the week that contains January 1 is deemed as the first week in the year. For the value, refer to the following settings.

Settings

The value of the interval factor is as follows:

Setting	Description
уууу	Year
q	Quarter
m	Month
У	Year(As of one year)
d	Day
W	Weekday
ww	Week(As of one year)
h	Hour
n	Minute
s	Second

The value of the firstweekofvearl factor is as follows:

Constant	Value	Description	
vbUseSystem	0	NLS(National Language Support). Use API settings	
vbSunday	1	Sunday(Default Value)	
vbMonday	2	Monday	
vbTuesday	3	Tuesday	
vbWednesday	4	Wednesday	
vbThursday	5	Thursday	
vbFriday	6	Friday	
vbSaturday	7	Saturday	

Description on Constant Value

Constant	Value	Description
vbUseSystem	0	NLS(National Language Support). Use API settings
vbFirstJan1	1	Start from the week that contains January 1 (Default Value)
vbFirstFourDays	2	Start from the week that contains the first four days in the
		new year.
vbFirstFullWeek	3	Start from the week that contains the first full week (7 days)
		in the new year.

Reference

If DateDiff Function is used, it is possible to know the time difference between two dates. For example, if DateDiff Function is used, it is possible to calculate the day number between two dates or the week number between today and the end of the year.

To calculate the day number between Date 1 and Date 2, use year ("y") or day ("d") as of the year. If interval is expressed with weekday ("w"), DateDiff Function returns the week number between two dates. If date1 is on Monday, DateDiff Function calculates how many Mondays are there by date2. That is, date2 is calculated instead of date 1. But, if interval is expressed with week ("ww"), DateDiff Function returns the calendar week numbers between two dates. That is, the Sunday number is calculated between date1 and date2. In DateDiff Function, if date2 is on Sunday, it calculates date2 but, if date1 is also on Sunday, it does not calculate date1.

If date1 is latter than date2, DateDiff Function returns a negative number.

The firstdayofweek factor has influence on the calculation using interval symbols such as "w" and "ww."

If date1 or date2 is a date literal, the designated year becomes the permanent element of the relevant date. But, if a pair of quotation marks (" ") is place around date1 or date2 and the relevant year is omitted, the current year shall be inserted to the user code whenever date1 or date2 expression is calculated. The code that can be used for the next year can be written in this way.

If December 31 is compared to the next January 1, DateDiff returns 1 to the year ("yyyy") factor even only one day is elapsed.

In the following example, DateDiff Function indicates the elapsed days until now from the given date.

Function DiffADate(theDate)
DiffADate = "Elapsed Day: " & DateDiff("d", Now, theDate)
End Function

DatePart Function

Description

The designated part of a given date is returned.

Syntax

DatePart(interval, date[, firstdayofweek[, firstweekofyear]])

DatePart Function Syntax has following factors.

factor	Description		
intonial	A mandatory factor. A literal expression of the time interval to be returned. For the		
interval	value, refer to the following settings.		
date	A mandatory factor. A date expression to be calculated.		
firstdayof week	An optional factor. Sunday is deemed as the day of the week if any Constant is not		
	designated to indicate which day of the week. For the value, refer to the following		
	settings.		
	An optional factor. A constant indicating the first week of the year. If not indicated, the		
_	week that contains January 1 is deemed as the first week in the year. For the value,		
	refer to the following settings.		

Settings

The value of the interval factor is as follows:

Settings	Description	
уууу	Year	
q	Quarter	
m	Month	
У	Year (As of one year)	
d	Day	
W	Weekday	
ww	Week(As of one year)	
h	Hour	
n	Minute	
S	Second	

The value of the firstweekofyear factor is as follows:

Constant	Value	Description	
vbUseSystem	0	NLS(National Language Support). Use API settings	
vbSunday	1	Sunday(Default Value)	
vbMonday	2	Monday	
vbTuesday	3	Tuesday	
vbWednesday	4	Wednesday	
vbThursday	5	Thursday	
vbFriday	6	Friday	
vbSaturday	7	Saturday	

Description on Constant Value

Constant	Value	Description
vbUseSystem	0	NLS(National Language Support). Use API settings
vbFirstJan1	1	Start from the week that contains January 1 (Default Value)
vbFirstFourDays	2	Start from the week that contains the first four days in the
		new year.
vbFirstFullWeek	3	Start from the week that contains the first full week (7 days)
		in the new year.

Reference

If DatePart Function is used, the date is calculated and a certain time interval is returned. For example, it is possible to calculate a weekday or the current time by using DatePart.

The firstdayofweek factor has influence on the calculation using interval symbols such as "w" and "ww."

If the date is a date literal, the designated year becomes the permanent element of the relevant date. But, if a pair of quotation marks (" ") is place around the date and the relevant year is omitted, the current year shall be inserted to the user code whenever the date expression is calculated. The code that can be used for the next year can be written in this way.

The following example shows Quarter of the relevant year by using DatePart Function and date.

Function GetQuarter(TheDate)

GetQuarter = DatePart("q", TheDate)

End Funtion

DateSerial Function

Description

The Date subtype of Variant of the designated year, month and day is returned.

Syntax

DateSerial(year, month, day)

DateSerial Function Syntax has following factors

Factor	Description
year	Numbers 100 through 9999 or a numerical expression
month	All expressions
day	All expressions

Reference

For example, to designate a certain date such as December 31, 1991, the number of each DateSerial factor should be within the tolerance range of the relevant. That is, for day, it should be one from the first day through the thirty first day and it should be one from January through December for month. But, the counter date of each factor can be designated by using a numerical expression showing the year, month or day before and after a particular date.

The following example uses a numerical expression instead of an absolute date number. Here, DateSerial Function returns the date subtracted one day from the first day(1-1) on the month two months before August (8-2) in the year 10 years before 1990 (1990-10).

DateSerial(1990 - 10, 8 - 2, 1 - 1)

If the value of the year factor is one from 0 through 99, it is interpreted from 1900 through 1999. In all other year factors except that, a four-digit number such as 1800 is used.

If the tolerance range of the factor is exceeded, it goes to the next bigger unit For example, if the thirty fifth day is designated, it is calculated to which day of one month according to the applicated year. But, if a certain factor exceeds range -32,768~32 and 767 or a date designated directly or through an expression by using three factors. Errors take place.

DateValue Function

Description

The Date subtype of Variant is returned.

Syntax

DateValue(date)

Typically, the date factor is a literal expression indicating from January 1, 100 to December 31, 9999 and the expression shows the date, time or date/time in this range.

Reference

If the date factor contains time information, DateValue Function does not return the information. However, some wrong information such as "89:98" is in the date factor, errors take place.

If the date factor is a string that has only the numbers divided by valid date separating symbols, DateValue Function recognizes year, month and day according to the simplified date marking type of the computer settings.

Also, it is possible to recognize an accurate date that contains the formal or informal date name. For example, DateValue Function recognizes December 30, 1991 or Dec 30, 1991 as well as 2/30/1991 or 12/30/91.

If the year part is omitted from the date factor, DateValue Function uses the current year as set up on the computer.

In the following example, DateValue Function converts a string to date. It is possible to directly assign the date to Variant variable by using the date literal (ex. MyDate = #9/11/63).

Dim MyDate

Day Function

Description

The integer indicating the date from 1 through 31 is returned.

Syntax

Day(date)

The date factor is an expression showing the date. If the date factor has the value zero, Null is returned.

In the following example, Day Function brings the date of the month from the designated date

Dim MyDay

MyDay = Day("Oct 19, 1962") 'MyDay gets value 19.

Eval Function

Description

A sentence as a factor is calculated and Result Value or the stack value of variable is returned.

Syntax

Eval(Expr[, Depth])

Factor:

Expr : As an expression to be operated. VB script syntax should be inputted as a string value.

Depth: Numbers should be inputted. Input the value location to the local variable's stack. If Depth = 0, the current value at the operation stage is returned and if omitted, Depth is 0.

Reference

The following example shows how Eval Function is used.

Ex1)

Dim Guess, RndNum

RndNum = Int((100) * Rnd(1) + 1)

```
Guess = CInt(InputBox("Enter your guess:",,0))
      If Eval("Guess = RndNum") Then
         MsgBox "Congratulations! You guessed it!"
         Exit Sub
      Else
          Guess = CInt(InputBox("Sorry! Try again.",,0))
      End If
   Loop Until Guess = 0
Ex2)
Sub Main
    Dim X As String
    X = "Hello"
    Debug.Print Eval("X")
                                            'Return Hello
End Sub
Sub A
    Dim X As String
    X = "Bye"
    Debug.Print Eval("X")
                                            'Return Bye
    Debug.Print Eval("X",1)
                                            'Return the previous value of Hello
End Sub
```

Exp Function

```
Description
```

'e', a root of natural logarithms raised by the multiplier is returned.

Syntax

Exp(number)

The number factor should be a valid expression.

Reference

If the number value exceeds 709.782712893, errors take place. Constant is about 2.718282.

Memo

Exp Function makes up for Log Function and it is sometimes called an antilogarithm. In the following example, Exp Function returns e for angle.

```
\label{eq:MyAngle} \begin{tabular}{ll} MyAngle = 1.3 & ' Define an angle in a radian value. \\ MyAngle = 1.3 & ' Calculate the sine value of a hyperbola. \\ MyHSin = (Exp(MyAngle) - Exp(-1 * MyAngle)) / 2 \\ \end{tabular}
```

Filter Function

Description

The Array based on 0, which has a subset of String Array according to the designated filter base, is returned.

Syntax

Filter(InputStrings, value[, Include[, compare]])

Filter Function Syntax has following factors.

Factor	Description
InputStrings	A mandatory factor. One dimension Array of the String to be searched.
value	A mandatory factor. The String to be searched.
Include	An optional factor. A boolean value showing whether to return the sub string including or excluding <i>value</i> . If it is True, Filter Function returns a subset of the Array corresponding to <i>value</i> as a sub string. If it is False, Filter Function returns a subset of the Array not corresponding to <i>value</i> as a sub string.
compare	An optional factor. A number value showing the <u>comparison</u> type to be used. For the value, refer to the following settings.

Settings

Comparison factors have the following values.

Constant	Value	Description
vbBinaryCompare	0	Run Binary Comparison.
vbTextCompare	1	Run Text Comparison.

Reference

If there is no value agreed with InputStrings, Filter Function returns empty Array. If InputStrings has the value zero or it is not one dimension Array, errors take place.

The returned Array in Filter Function has only the elements enough to contain the agreed item number.

In the following example, Filter Function returns the Array that has "Mon" search reference.

Dim MyIndex

Dim MyArray (3)

MyArray(0) = "Sunday"

MyArray(1) = "Monday"

MyArray(2) = "Tuesday"

MyIndex = Filter(MyArray, "Mon") 'MyIndex(0) contains "Monday"

Fix Function

Description

The integer part of a number is returned.

Syntax

Int(number)

Fix(number)

The number factor should be a valid expression. The number has the value zero, Null is returned.

Reference

Both of Int Function and Fix Function return the integer value that shows the results of removing the part after the decimal point.

If the difference between Int Function and Fix Function is a negative number, Int Function returns the first negative integer that is lower or equal to the number but, Fix Function returns the negative integer that is higher or equal to the number. For example, Int Function converts -8.4 to -9 and Fix Function converts -8.4 to -8.

Fix (number) Function is equal to the following expression.

Sgn(number) * Int(Abs(number))

The following example shows how Int and Fix Function convert the integer part of a number.

MyNumber = Int(99.8) '99 is returned.

MyNumber = Fix(99.2) '99 is returned.

MyNumber = Int(-99.8) '-100 is returned.

MyNumber = Fix(-99.2) '-99 is returned.

MyNumber = Fix(-99.2) '-99 is returned.

FormatCurrency Function

Description

A formalized expression with the call values that use the defined call symbols on the computer's [Control Board] is returned.

Syntax

FormatCurrency(expression[,NumDigitsAfterDecimal [,IncludeLeadingDigit [,UseParensForNegativeNumbers [,GroupDigits]]]])

FormatCurrency Function Syntax has following factors.

Factor	Description
expression	A mandatory factor. An expression whose format is specified.
NumDigitsAfterDecimal	An optional factor. A number value indicating the number of digits after the decimal point.
	The Default Value is -1, showing the use of the computer's national settings.
IncludeLeadingDigit	An optional factor. A Tristate Constant showing whether to mark 0 before a valid number
	after the decimal point. For the value, refer to the following settings.
UseParensForNegative	An optional factor. A Tristate Constant showing whether to place a negative value in a
Numbers	paranthesis. For the value, refer to the following settings.
GroupDigits	An optional factor. A Tristate Constant showing whether to identify a number with the
	designated group distinguishing symbols according to the computer's national settings.

Settings

IncludeLeadingDigit, UseParensForNegativeNumbers and GroupDigits factor have the following values,

Constant	Value	Description
TristateTrue	-1	True
TristateFalse	0	False
TristateUseDefault	-2	Use of the computer's national
		settings.

Reference

If more than one factor is omitted, the factor value that omits the computer's national setting value is used. The location of the call symbols related to call values is decided by the computer's national setting value.

Memo

The entire setting information except previously acted 0 (zero) on [Number] tap is on [Call] tap of [National settings].

FormatDateTime Function

Description

A formalized expression with date or time is returned.

Syntax

FormatDateTime(date[,NamedFormat])

FormatDateTime Function Syntax has following factors.

Factor	Description		
date	A mandatory factor. Specify a date expression.		
INamed-ormat	An optional factor. A number value showing the used date/time format.		
	If this factor is omitted, vbGeneralDate is used.		

Settings

NamedFormat factor has the following settings.

Constant	Value	Description
		Mark date or time. If there is date part, mark the date in a simple date type, If there
vbGeneralDate	0	is time part, mark the time in a detailed time type. If date and time exist at the same
		time, mark both of them.
vbLongDate	1	Show date in the detailed date type designated by the computer's national settings.
vbShortDate	2	Show date in the simple date type designated by the computer's national settings.
vbLongTime	3	Show time in the time type designated by the computer's national settings.
vbShortTime	4	Show time in 24-hr type (hh:mm).

Reference

In the following example, FormatDateTime Function designates the expression in a long date type and exports it to MyDateTime.

Function GetCurrentDate

'FormatDateTime specifies Date in a long date type.

GetCurrentDate = FormatDateTime(Date, 1)

End Function

FormatNumber Function

Description

A formalized expression with number is returned.

Syntax

FormatNumber(expression[,NumDigitsAfterDecimal [,IncludeLeadingDigit [,UseParensForNegativeNumbers [,GroupDigits]]]])

FormatNumber Function Syntax has following factors.

Factor	Description		
expression	A mandatory factor. An expression whose format is specified.		
	An optional factor. A number value indicating the number of digits after the decimal		
NumDigitsAfterDecimal	point. The Default Value is -1, showing the use of the computer's national		
	settings.		
Included coding Digit	An optional factor. A Tristate Constant showing whether to mark 0 before a valid		
IncludeLeadingDigit	number after the decimal point. For the value, refer to the following settings.		
UseParensForNegativeN	An optional factor. A Tristate Constant showing whether to place a negative value in		
umbers	a parenthesis. For the value, refer to the following settings.		
	An optional factor. A Tristate Constant showing whether to identify a number with		
GroupDigits	the designated group distinguishing symbols set up on [Control Board]. For the		
	value, refer to the following settings.		

Settings

IncludeLeadingDigit, UseParensForNegativeNumbers and GroupDigits factor have the following settings.

Constant	Value	Description
TristateTrue	-1	True
TristateFalse	0	False
TristateUseDefault	-2	The computer's national settings are used.

Reference

If more than one optional factor is omitted, the factor value that omits the computer's national setting value is used.

Memo The entire setting information is on [Number] tap of [National Settings].

In the following example, FormatNumber Function specifies a number in a four-digit number of a decimal system.

Function FormatNumberDemo

Dim MyAngle, MySecant, MyNumber

MyAngle = 1.3 'Define an angle in a radian value.

MySecant = 1 / Cos(MyAngle) 'Caculate the Secant value. .

FormatNumberDemo = FormatNumber(MySecant,4) ' Specify MySecant in a four-digit number of a decimal system.

End Function

FormatPercent Function

Description

An expression that has character % at the end and whose format is designated in percentage multiplied by 100 is returned.

Syntax

FormatPercent(expression[,NumDigitsAfterDecimal [,IncludeLeadingDigit [,UseParensForNegativeNumbers [,GroupDigits]]]])

FormatPercent Function Syntax has following factors.

factor	Description
expression	A mandatory factor. A formalized expression is returned.
NumDigitsAfterDecimal	An optional factor. A number value indicating the number of digits after the decimal
	point. The Default Value is -1, showing the use of the computer's national settings.
Included coding Digit	An optional factor. A Tristate Constant showing whether to mark 0 before a valid
IncludeLeadingDigit	number after the decimal point. For the value, refer to the following settings.
UseParensForNegativeNu	An optional factor. A Tristate Constant showing whether to place a negative value in
mbers	a parenthesis. For the value, refer to the following settings.
	An optional factor. A Tristate Constant showing whether to identify a number with the
GroupDigits	designated group distinguishing symbols set up on [Control Board]. For the value,
	refer to the following settings.

Settings

IncludeLeadingDigit, UseParensForNegativeNumbers, GroupDigits factor

 have the following settings.

Constant	Value	Description
TristateTrue	-1	True
TristateFalse	0	False
TristateUseDefault	-2	The computer's national settings are used.

Reference

If more than one optional factor is omitted, the factor value that omits the computer's national setting value is used.

Memo The entire setting information is on [Number] tap of [National Settings].

In the following example, FormatPercent Function specifies an expression in a type of percentage.

Dim MyPercent

MyPercent = FormatPercent(2/32) 'MyPercent gets 6.25%.

GetObject Function

Description

A reference on automatic objects in the file is returned.

Syntax

GetObject([pathname] [, class])

GetObject Function Syntax has following factors.

Factor	Description
	An optional factor. A string showing the name and entire paths of a file that has the object to be searched. If <i>pathname</i> is omitted, <i>class</i> should exist.
class	An optional factor. A string showing the <u>Class</u> of an object.

The class factor uses appname.objectype syntax, and its configuration element is described as follows:

Configuration Element	Description
appname	A mandatory factor. A string showing the name of the application program supplying
	the relevant object.
objectype	A mandatory factor. A string showing the class of format of the object to be created.

Hex Function

Description

A string showing the hexadecimal value of a number is returned.

Syntax

Hex(number)

The number factor should be a valid expression.

Reference

Numbers that are not integer are rounded off to the nearest integer before calculating.

Number Value				Result of Hex Function
Null				Null
Empty				0
Numbers	other	than	two	Hexadecimal characters less than 8
value above				

Hexadecimal can be expressed by attaching &H before the numbers in a valid range. For example, &H10 expresses hexadecimal mark of decimal number 16.

In the following example, Hex Function returns the hexadecimal value of a number.

Dim MyHex

MyHex = Hex(5) '5 is returned. MyHex = Hex(10) 'A Is returned. MyHex = Hex(459) '1CB is returned.

Hour Function

Description

An integer indicating the hour of a day from 0 through 23 is returned.

Syntax

Hour(time)

The time factor is an expression showing time. If time has the value zero, Null is returned.

In the following example, Hour Function exports hour from the current time.

Dim MyTime, MyHour

MyTime = Now

MyHour = Hour(MyTime) ' MyHour gets a number showing the current hour.

InputBox Function

Description

It displays a prompt in the dialog box or it waits until any text is inputted or a button is pressed or it returns the inputted information.

Syntax

InputBox(prompt[, title][, default][, xpos][, ypos][, helpfile, context]) InputBox Function Syntax has following factors.

factor	Description
	A literal expression showing message in the dialog box. prompt is about 1,024 characters
	long at the most even though it varies depending on the width of the characters. If
prompt	prompt is long enough to take more than two lines, the lines can be divided by carriage
	return character (Chr(13)), line feed character(Chr(10)) or a combination of carriage
	return and line feed characters (Chr(13) & Chr(10)).
title	A literal expression showing on the title line in the dialog box. If title< is omitted, the
uuc	application program title appears on the tile line.
default	A literal expression basically displayed on the text box if noting is inputted. If default is
acraun	omitted, an empty text box appears.
	A numerical expression specifying the horizontal length from the screen left end to the
xpos	dialog left end in a twip unit. If xpos is omitted, the dialog box is located in the middle on
	the screen.
	A numerical expression specifying the vertical length from the screen upper end to the
ypos	dialog upper end in a twip unit. If ypos is omitted, the dialog box is located on the point
	1/3 above from the bottom of the screen.
helpfile	A literal expression identifying the 'Help' file to provide assistance according to the
	situation If helpfile is provided, it is necessary also to provide context.
context	A numerical expression identifying the Help context number as the relevant Help items
Context	designated by the author. If <i>context</i> is provided, it is necessary also to provide <i>helpfile</i> .

Reference

If both of helpfile and context are provided, [Help] button is automatically added in the dialog box.

In InputBox Function, the contents in the text box are returned if [Ok] or Key is pressed. If [Cancel] is pressed, string ("") whose length is 0 is returned.

In the following example, InputBox Function displays a text box and assigns string to variable.

Dim Input

Input = InputBox("Input a name")

MsgBox ("The following name is inputted: " & Input)

InStr Function

Description

The position on which a string firstly appears in other string is returned.

Syntax

InStr([start,]string1, string2[, compare])

InStr Function Syntax has following factors.

factor	Description
	An optional factor. A numerical expression specifying a position at which each search starts. If
start	omitted, search starts from the first character. The start factor is Null, errors takes place. To
	specify Compare, the start factor is needed.
string1	A mandatory factor. A literal expression to be searched.
string2	A mandatory factor. A literal expression subject to searching.
compare	An optional factor. A number value displayed in a comparison type to be used for assessing sub strings. For the value, refer to the following settings. If omitted, Run Binary Comparison.

Settings

Comparison factors have the following values.

Constant	Val	Description
	ue	
vbBinaryCompare	0	Run Binary Comparison.
vbTextCompare	1	Run Text Comparison.

Result Value

InStr Function returns the following values.

Condition	Result of InStr Function
If the length of string1 is 0	0
If string1 is Null	Null
If the length of string2 is 0	start
If string2 is Null	Null
If there is no string2	0
If string1 has string2	The relevant location in string1
start > Len(string2)	0

Reference

In the following example, InStr searches a string.

Dim SearchString, SearchChar, MyPos

SearchString ="XXpXXpXXPXXP" 'String to be searched

SearchChar = "P" 'Search "P"

'A text comparison starts at position 4. 6 is returned.

MyPos = Instr(4, SearchString, SearchChar, 1)

'A binary comparison starts at position 1. 9 is returned.

MyPos = Instr(1, SearchString, SearchChar, 0)

' Comparison is binary by default (last argument is omitted).

MyPos = Instr(SearchString, SearchChar) 'Returns 9

' A binary comparison starting at position 1. Returns 0 ("W" is not found). MyPos = Instr(1, SearchString, "W")

Memo

InStrB Function uses Byte data that the string has. InStrB Function returns Byte position rather than the position

where a string appears in the other string for the first time.

InStrRev Function

Description

The position on which a string firstly appears in other string is returned at the end of the string.

Syntax

InStrRev(string1, string2[, start[, compare]])
InStrRev Function Syntax has following factors.

factor	Description
string1	A mandatory factor. A literal expression to be searched.
string2	A mandatory factor. A literal expression subject to searching.
	An optional factor. A numerical expression specifying a position at which each search starts. If
	omitted, -1 is used and it means search starts at the last character If start is Null, errors takes
	place.
compare	An optional factor. A number value displayed in a comparison type to be used for assessing sub
	strings. If omitted, Run Binary Comparison. For the value, refer to the following settings.

Settings

Comparison factors have the following values.

Constant	Valu	Description
	е	
vbBinaryCompare	0	Run Binary Comparison.
vbTextCompare	1	Run Text Comparison.

Result Value

InStrRev Function returns the following values.

Condition	InStrRev FunctionResult of
If the length of string1 is 0	0
If string1 is Null	Null
If the length of string2 is 0	start
If string2 is Null	Null
If there is no string2	0
If string1 has string2	The relevant location in string1
start > Len(string2)	0

Reference

In the following example, InStrRev Function searches a string.

Dim SearchString, SearchChar, MyPos

SearchString ="XXpXXpXXPXXP" 'String to be searched

SearchChar = "P" 'Search "P"

'A binary comparison starts at position 10. 9 is returned. MyPos = InstrRev(SearchString, SearchChar, 10, 0)

'A text comparison starts at the last position. 12 is returned.

MyPos = InstrRev(SearchString, SearchChar, -1, 1)

'For comparison, the Default Value is binary (The last factor is omitted). 0 is returned. MyPos = InstrRev(SearchString, SearchChar, 8)

Memo InStrRev Function Syntax is different from InStr Function Syntax.

Int Function

Description

The integer part of a number is returned.

Syntax

Int(number)

Fix(number)

The number factor should be a valid expression. number ℍ Has the value zeroNull is returned.

Reference

Both of Int Function and Fix Function return the integer value that shows the results of removing the part after the decimal point.

If the difference between Int Function and Fix Function is a negative number, Int Function returns the first negative integer that is lower or equal to the number but, Fix Function returns the negative integer that is higher or equal to the number. For example, Int Function converts -8.4 to -9 and Fix Function converts -8.4 to -8.

Fix(number) Function is equal to the following expression.

Sgn(number) * Int(Abs(number))

The following example shows how Int and Fix Function convert the integer part of a number.

MyNumber = Int(99.8) '99 is returned.

MyNumber = Fix(99.2) '99 is returned.

MyNumber = Int(-99.8) '-100 is returned.

MyNumber = Fix(-99.2) '-99 is returned.

MyNumber = Fix(-99.2) '-99 is returned.

IsArray Function

Description

A boolean value showing whether variable is Array or not is returned.

Syntax

IsArray(varname)

The varname factor is variable.

Reference

IsArray Function returns True if variable is Array and it returns False if variable is not Array. IsArray Function is especially useful if Variant that contains Array is used.

In the following example, IsArray Function tests whether MyVariable is Array or not.

Dim MyVariable

Dim MyArray(3)

MyArray(0) = "Sunday"

MyArray(1) = "Monday"

MyArray(2) = "Tuesday"

MyVariable = IsArray(MyArray) ' MyVariable gets "True."

IsDate Function

Description

A boolean value showing whether an expression is converted to date is returned

Syntax

IsDate(expression)

The expression factor is a literal expression that can be recognized as a date expression or date or time.

Reference

IsDate Function returns True if the relevant expression is date or it can be converted to a valid date while if not, False is returned. Even though a date range from January 1, AD 100 to January 1, AD 9999 is valid on the Microsoft Windows, its range varies depending on the operation system.

In the following example, IsDate Function decides whether an expression can be converted to date.

Dim MyDate, YourDate, NoDate, MyCheck

MyDate = "October 19, 1962": YourDate = #10/19/62#: NoDate = "Hello"

MyCheck = IsDate(MyDate) 'True is returned.

MyCheck = IsDate(YourDate) 'True is returned.

MyCheck = IsDate(NoDate) 'False is returned.

IsEmpty Function

Description

A boolean value showing whether Variable is initialized or not is returned.

Syntax

IsEmpty(expression)

The expression factor can be any expression. However, a typical expression factor is the name of a single variable because IsEmpty Function is used to ensure whether an individual variable is initialized or not.

Reference

IsEmpty Function returns True if the variable is not initialized or the variable is expressly designated as Empty or False is returned. If the expression has more than two variables, False is always returned.

In the following example, IsEmpty Function judges whether the variable is initialized.

Dim MyVar, MyCheck

MyCheck = IsEmpty(MyVar) 'True is returned.

MyVar = Null 'Assign Null.

MyCheck = IsEmpty(MyVar) 'False is returned.

MyVar = Empty 'Assign Empty.

MyCheck = IsEmpty(MyVar) 'True is returned.

IsNull Function

Description

A boolean value showing whether a Null expression has any valid date that is, it is Null or not is returned.

Syntax

IsNull(expression)

The expression factor can be any expression.

Reference

IsNull Function returns True if the expression is Null that is, it have no valid date, otherwise, False is returned. If the expression consists of more than two variables, Null that belongs to the entire configuration variables returns True for the full expressions.

Null value means the relevant variable has no valid data. Null is different from Empty, which shows the relevant variable is not yet initialized and it is also different from String (""), whose length is 0 and it is sometimes called Null String.

Memo

It is decided whether an expression has Null value by using IsNull Function. In some cases, an expression is always False even though True is expected as seen in expressions such as If Var = Null and If Var <> Null. It becomes False because the expression itself contains Null.

In the following example, IsNull Function decides whether the variable has Null value.

Dim MyVar, MyCheck

MyCheck = IsNull(MyVar) 'False is returned.

MyVar = Null 'Designate Null.

MyCheck = IsNull(MyVar) 'True is returned.

MyVar = Empty ' Designate Empty.

MyCheck = IsNull(MyVar) ' Designate False.

IsNumeric Function

Description

A boolean value showing whether an expression can be evaluated with numbers.

Syntax

IsNumeric(expression)

The expression factor can be any expression.

Reference

If the expression is a date expression, IsNumeric Function returns False.

In the following example, IsNumeric Function decides whether the variable can be evaluated with numbers.

Dim MyVar, MyCheck

MyVar = 53 'Designate a value.

MyCheck = IsNumeric(MyVar) 'True is returned.

MyVar = "459.95" ' Designate a value. MyCheck = IsNumeric(MyVar) ' True is returned.

MyVar = "45 Help" ' Designate a value.

MyCheck = IsNumeric(MyVar) ' False is returned.

IsObject Function

Description

A boolean value showing whether an expression refers valid automatic objects is returned.

Syntax

IsObject(expression)

The expression factor can be any expression.

Reference

IsObject Function returns True if the relevant expression is a subtype variable of Object or a customized object, otherwise, False is returned.

In the following example, IsObject Function decides whether the identifier shows individual variables.

Dim MyInt, MyCheck, MyObject

Set MyObject = Me

MyCheck = IsObject(MyObject) 'True is returned.

MyCheck = IsObject(MyInt) 'False is returned.

Join Function

Description

A string made by combining various sub strings included in Array is returned.

Syntax

Join(list[, delimiter])

Join Function Syntax has following factors.

factor	Description
list	A mandatory factor. A one-dimension Array including sub strings to be combined.
	An optional factor. A character used to separate sub strings of the returned string. If omitted,
delimiter	empty character (" ") is used. If delimiter is a string whose length is 0, the entire items in the
	list are combined without any symbol to divide them.

LBound Function

Description

The smallest subscript available in the designated Array level is returned.

Syntax

LBound(arrayname[, dimension])

LBound Function Syntax has following factors.

Factor	Description
arrayname	Comply with the notation rule of general variable with Array variable name
dimension	An integer showing the lower limit of the returned level is returned. Use 1 for one dimension, 2 for two dimension, etc. If <i>dimension</i> is omitted, 1 is deemed.

Reference

LBound Function is used along with UBound Function to decide the size of Array. If UBound Function is used, the upper limit can be known in the Array level.

The lower limit of the entire dimensions is always 0.

LCase Function

Description

A string converted to decimal point is returned.

Syntax

LCase(string)

The string factor should be a valid character expression. If there is Null in string, Null is returned.

Reference

This function converts only capital letters to small letters and makes other characters except them remain as they are without converting.

In the following example, CDate Function converts capital letters to small letters.

Dim MyString

Dim LCaseString

MyString = "VBSCript"

LCaseString = LCase(MyString) ' LcaseString gets "vbscript."

Left Function

Description

Characters as many as designated from the left are returned.

Syntax

Left(string, length)

Left Function Syntax has following factors.

Factor	Description
Istrina	A literal expression to be returned from the character at the left end. If there is Null in string, Null
	is returned.
	A numerical expression showing the character count to be returned. If it is 0, a string whose
	length is 0("") is returned. Their entire strings are returned if it is equal or more than the
	character count that is in string.

Reference

To decide the character count of string, use Len Function.

In the following example, Left Function returns the first three characters of MyString.

Dim MyString, LeftString

MyString = "VBSCript"

LeftString = Left(MyString, 3) 'LeftString gets "VBS."

Memo

LeftB Function uses Byte data that the string has. It designates Byte count rather than character count to be returned to length factor.

Len Function

Description

The Byte count needed for saving character count or variable that is in the string is returned.

Syntax

Len(string | varname)

Len Function Syntax has following factors.

Factor	Description
string	A literal expression. If there is Null in string, Null is returned.
varname	It should be a valid <u>variable</u> name. <i>Varname</i> has the value zero, Null is returned.

Reference

In the following example, Len Function returns the character count of a string.

Dim MyString

MyString = Len("VBSCRIPT") 'MyString gets value 8.

Memo

LenB Function uses Byte data that the string has. Instead of returning character count to the string, LenB returns Byte count used for expressing this string.

LoadPicture Function

Description

If a picture object is returned, it can be used only on the 32-bit platform.

Syntax

LoadPicture(picturename)

The picturename factor is a literal expression showing the name of a picture file to be loaded.

Reference

LoadPicture Function recognizes the following graphic types; bitemap files(.bmp), icon files(.ico), execution length symbolization files(.rle), meta files(.wmf), improved meta files(.emf), GIF files(.gif) and JPEG files(.jpg).

Log Function

Description

A natural logarithm of a number is returned.

Syntax

Log(number)

The number factor should be a valid expression more than 0.

Reference

A natural logarithm is a log whose base is e. Constant e is about 2.718282.

Natural logarithm logn(x) whose base is n and whose value is x can be calculated by dividing natural logarithm logn(x) by natural logarithm logn(n).

```
Logn(x) = Log(x) / Log(n)
```

The following example shows a customized Function that calculates Log Function whose base is 10.

Function Log10(X)

Log10 = Log(X) / Log(10)

End Function

LTrim Function

Description

LTrim Function returns a string copy which has no left empty space or no right empty space or neither left nor right empty space.

Syntax

LTrim(string)

RTrim(string)

Trim(string)

The string factor should be a valid character expression. If there is Null in string, Null is returned.

Reference

In the following example, LTrim Function trims the left empty space, RtrimFunction trims the right empty space

and Trim Function trims both of them.

Dim MyVar

MyVar = LTrim(" vbscript ") 'MyVar gets "vbscript ."

MyVar = RTrim(" vbscript ") 'MyVar gets " vbscript."

MyVar = Trim(" vbscript ") 'MyVar gets "vbscript"."

Mid Function

Description

Charaters as many as designated by string are returned.

Syntax

Mid(string, start[, length])

Mid Function Syntax has following factors.

factor	Description
otring	A literal expression containing the characters to be returned. If there is Null in
string	string, Null is returned.
	A character position at which returning starts in the string. If start factor is more
start	than character count of string, Mid Function returns a string ("") whose length is 0
	is returned.
	Character count to be returned. If omitted or the actual character count is less
length	than the character count of length factor including start factor, the entire
lengin	characters from the start position to the end character of the relevant string shall
	be returned.

Reference

The character count of a string can be decided by Len Function.

In the following example, Mid Function returns six characters starting from the fourth character of a string.

Dim MyVar

MyVar = Mid("VB Script is fun!", 4, 6) 'MyVar returns "Script."

Memo

MidB Function uses Byte data that the string has. In this Function, Byte count is designated rather than character count.

Minute Function

Description

An integer indicating the minute from 0 through 59 is returned.

Syntax

Minute(time)

The time factor is an expression showing time. The time has the value zero, Null is returned.

Reference

In the following example, Minute Function returns minute.

Dim MyVar

MyVar = Minute(Now)

Month Function

Description

The integer indicating month from 1 through 12 is returned.

Syntax

Month(date)

The date factor is an expression showing the date. If date has the value zero, Null is returned.

Reference

In the following example, Month Function returns the current month.

Dim MyVar

MyVar = Month (Now) ' MyVar gets a number corresponding to the current month.

MonthName Function

Description

A string showing the designated month is returned.

Syntax

MonthName(month[, abbreviate])

MonthName Function Syntax has following factors.

Factor	Description
month	A mandatory factor. Number showing month such as 1 for January and 2 for February, etc.
	An optional factor. A boolean value showing whether to display a month name in an abbreviated word type. If omitted, it become False, the Default Value, and shows its full name.

Reference

In the following example, MonthName Function returns the month name to be inputted to the date expression in an abbreviated word type.

Dim MyVar

MyVar = MonthName(10, True) ' MyVar gets "Oct."

MsgBox Function

Description

A value that shows whether any message is displayed in the dialog box or whether it waits until any text is inputted, or indicates the button pressed by the user is returned.

Syntax

MsgBox(prompt[, button][, title][, helpfile, context])

MsgBox Function Syntax has following factors.

Factor	Description
prompt	A literal expression showing message in the dialog box. prompt is about 1,024 characters
	long at the most even though it varies depending on the width of the characters. If
	prompt is long enough to take more than two lines, the lines can be divided by carriage
	return character (Chr(13)), line feed character(Chr(10)) or a combination of carriage return
	and line feed characters (Chr(13) & Chr(10)).
	A numerical expression showing the kind and number of the button to be displayed, icon
button	types to be used, the sum of values designating targets of basic button confirmation and
ballon	message box. For the value, refer to the following settings. If omitted, 0 appears, the
	Default Value of the button.
title	A literal expression showing on the line for the title in the dialog box. If title is omitted, the
litie	application program title appears on the tile line.
	A literal expression identifying the 'Help' file to provide assistance according to the situation
helpfile	If helpfile is provided, it is necessary also to provide context. Not available in the 16-bit
	platform.
context	A numerical expression identifying the Help context number as the relevant Help items
	designated by the author. If helpfile is provided, it is necessary also to provide context. Not
	available in the 16-bit platform.

Settings

The button factor has the following settings.

Constant	Value	Description
vbOKOnly	0	Display OK button only.
vbOKCancel	1	Display OK and Cancel button.
vbAbortRetryIgnore	2	Display Abort, Retry and Ingnore button.
vbYesNoCancel	3	Display Yes, No, and Cancel button.
vbYesNo	4	Display Yes and No button.
vbRetryCancel	5	Display Retry and Cancel button.
vbCritical	16	Display Critical Error Message Icon.
vbQuestion	32	Display Warning Query Icon.
vbExclamation	48	Display Warning Message Icon.
vbInformation	64	Display Information Message Icon.
vbDefaultButton1	0	The first button is the default one.
vbDefaultButton2	256	The second button is the default one.
vbDefaultButton3	512	The third button is the default one.
vbDefaultButton4	768	The fourth button is the default one.
		An application modal. The user has to answer to the questions in the
vbApplicationModal	0	message box before continuing to work on the current application
		program.
vbSystemModal	4096	A system modal. All application programs aborts until the user answer to
Vocysternivioual	7000	the questions in the message box.

The first group values (0-5) show the button types and numbers displayed in the dialog box. The second group (16, 32, 48, 64) shows icon types. The third group (0, 256, 512, 768) decides basic buttons and the fourth group (0, 4096) decides the modal of the message box. When the final value of the button factor is obtained by adding values, only one number is used from each group.

Result Value

MsgBox Function returns the following values.

Constant	Value	Button
vbOK	1	OK
vbCancel	2	Cancel
vbAbort	3	Abort
vbRetry	4	Retry
vblgnore	5	Ignore
vbYes	6	Yes
vbNo	7	No

Reference

If both helpfile and context are provide, press F1 key to see Help items for each situation.

If the dialog box has [Cancel] button, pressing Esc key leads to the same results as pressing [Cancel] button. If the dialog box has [Help] button, Help items are provided for each situation. However, one of other buttons is pressed, any value is not returned.

In the following example, MsgBox Function returns the value that displays the message box and explains the selected button.

Dim MyVar

MyVar = MsgBox ("Hello?", 65, "MsgBox Example") ' MyVar gets value 1 or 2 according to the selected button.

Now Function

Description

The current date and time are returned according to the computer's system date and time.

Syntax

Now

Reference

In the following example, Now Function returns the current date and time.

Dim MyVar

MyVar = Now ' MyVar gets the current date and time.

Oct Function

Description

An octal value of a number is returned.

Syntax

Oct(number)

The number factor should be a valid expression.

Reference

Numbers that are not integer are rounded off to the nearest integer before calculating.

Value Kind of a number	Result of Oct Function
Null	Null
Empty	0
Numbers other than two value	Octal characters less than 11
above	

Octal can be expressed by attaching &O before the numbers in a valid range. For example, &O10 shows the octal notation of decimal number 8.

In the following example, Oct Function returns the octal value.

Dim MyOct

MyOct = Oct(4) '4 is returned.

MyOct = Oct(8) '10 is returned.

MyOct = Oct(459) '713 is returned.

Replace Function

Description

A string that replaces the designated sub string with other sub strings is returned.

Syntax

Replace(expression, find, replacewith[, start[, count[, compare]]])

Replace Function Syntax has following factors.

factor	Description
expression	A mandatory factor. A literal expression containing the sub strings to be replaced.
find	A mandatory factor. A sub string to be found.
replacewith	A mandatory factor. A sub string to be replaced.
start	An optional factor. A position at which searching a sub string starts in the expression. If
	omitted, 1 is deemed. It shall be used along with count.
number	An optional factor. Number to replace the sub string. If omitted, Default Value becomes -1
number	and replaces all of the relevant strings. It shall be used along with start.
compare	An optional factor. A number value showing the comparison type of the sub string to be
	used. For the value, refer to the following settings. If omitted, Default Value becomes 0 and
	performs binary comparison.

Settings

Comparison factors have the following values.

Constant	Value	Description
vbBinaryCompare	0	Run Binary Comparison.
vbTextCompare	1	Run Text Comparison.

Result Value

Replace Function returns the following values.

Condition	Result of Replace Function
If it is expression whose length is 0	A string whose length is 0("")
If expression is Null	Error
If it is find whose length is 0	Expression copy
If it is replacewith whose length is 0	Omission of the entire sesarches for find and expression copy
start > Len(expression)	A string whose length is 0
If number is 0	expression copy

Reference

Replace Function starts at the start position and replaces at the end of the expression string, and then, returns the string. That is, the returned string as a result is not the copy of the original strings.

In the following example, Replace Function returns the string.

Dim MyString

'If a Binary Comparison starts at the beginning of a string, "XXYXXPXXY" is returned. MyString = Replace("XXpXXPXXp", "p", "Y")

'If a Text comparison starts at position 3, "YXXYXXY" is returned. MyString = Replace("XXpXXPXXp", "p", "Y", 3, -1, 1)

Sgn Function

Description

Characters as many as designated from the right are returned.

Syntax

Right (string, length)

Right Function Syntax has following factors.

Factor	Description
string	A literal expression to be returned from the right end character. If there is Null in string, Null is returned.
length	A numerical expression showing the character count to be returned. If it is 0, a string whose length is 0("") is returned. Their entire strings are returned if it is equal or more than the character count that is in <i>string</i> .

Reference

To decide the character count of string, use Len Function.

In the following example, Right Function returns characters as many as designated from the right side of the string.

Dim AnyString, MyStr

AnyString = "Hello?" 'Designate a string.

MyStr = Right(AnyString, 1) ' "?" is returned.

MyStr = Right(AnyString, 6) ' "llo?" is returned.

MyStr = Right(AnyString, 20) ' "Hello?" is returned.

Memo

RightB Function uses Byte data that the string has. It designates Byte count rather than character count to be returned to length factor.

Rnd Function

Description

A random number is returned.

Syntax

Rnd[(number)]

The number factor should be a valid expression.

Reference

Rnd Function returns a value that is bigger than 0 and smaller than 1. The number value decides how Rnd Function generates random numbers.

Number Value	Result of Rnd Function
< 0	Every time the same number by <u>seed number</u>
> 0	Next random number created sequentially.
= 0	The latest random number
Omitted	Next random number created sequentially.

In case of any given initial seed, numbers in the same sequence are generated because previous numbers are used as seeds to have next sequential numbers in continual calls for Rnd Function at each time.

Before calling Rnd, use Randomize statement and initialize the random number generator with seed based on the system timer.

To make random integer in a given range, use the following formula:

Int((upperbound - lowerbound + 1) * Rnd + lowerbound)

upperbound refers to the biggest number and lowerbound refers to the smallest number in the range.

Memo

To repeat the sequence of generating random numbers, call Rnd Function with negative factors right before using Randomize with number factors. To use Randomize with the same value, the previous sequence is not repeated.

RGB Function

Description

The integers showing RGB color values are returned.

Syntax

RGB(red, green, blue)

RGB Function has following factors.

factor	Description
red	A mandatory factor. Numbers 0 through 255 indicating red color elements.
green	A mandatory factor. Numbers 0 through 255 indicating green color elements.
blue	A mandatory factor. Numbers 0 through 255 indicating blue color elements.

Reference

Numbers showing RGB color values are necessary for methods and properties of the application program receiving color designation. RGB color values designate relative concentration of red, green and blue and shows particular colors.

Lower Byte contains red value, middle byte contains green value and upper byte contains blue value.

Even though the following Function shows the reverse byte order, it provides the same information to the application programs that need that sequence.

Function RevRGB(red, green, blue)

RevRGB= CLng(blue + (green * 256) + (red * 65536))

End Function

For RGB Function's factor values that exceed 255, they are deemed as 255.

Round Function

Description

The number rounded off at the designated decimal place is returned.

Syntax

Round(expression[, numdecimalplaces])

Round Function Syntax has following factors.

Factor	Description
expression	A mandatory factor. A numerical expression to be rounded off.
Inumgecimalplaces	An optional factor. A number showing decimal place. If omitted, Round Function returns integers.

Reference

In the following example, Round Function rounds off a number two-digit decimal.

Dim MyVar, pi

pi = 3.14159

MyVar = Round(pi, 2) 'MyVar gets 3.14.

RTrim Function

Description

LTrim Function returns a string copy which has no left empty space or no right empty space or neither left nor right empty space.

Syntax

LTrim(string)

RTrim(string)

Trim(string)

The string factor should be a valid character expression. If there is Null in string, Null is returned.

Reference

In the following example, LTrim Function trims the left empty space, Rtrim Function trims the right empty space and Trim Function trims both of them.

```
Dim MyVar

MyVar = LTrim(" vbscript ") 'MyVar gets "vbscript ".

MyVar = RTrim(" vbscript ") 'MyVar gets " vbscript".

MyVar = Trim(" vbscript ") 'MyVar gets "vbscript".
```

ScriptEngine Function

Description

A string showing the script language in use is returned.

Syntax

ScriptEngine

Result Value

ScriptEngine Function returns the following strings.

String	Description
VBScript	It shows that Microsoft(R) Visual Basic(R) Scripting Edition is the current script engine.
JScript	It shows that Microsoft JScript(R) is the current script engine
VBA	It shows that Microsoft Visual Basic for Applications is the current script engine

Reference

In the following example, ScriptEngine Function returns the script language in use.

Function GetScriptEngineInfo

Dim s

s = "" 'Script a string with necessary information.

s = ScriptEngine & "Version"

s = s & ScriptEngineMajorVersion & "."

s = s & ScriptEngineMinorVersion & "."

s = s & ScriptEngineBuildVersion

GetScriptEngineInfo = s 'Result is returned.

End Function

ScriptEngineBuildVersion Function

Description

The configuration version number of the script language in use is returned.

Svntax

ScriptEngineBuildVersion

Reference

The returned Result Value is equal to the version information contained in the DLL of the script language in use. In the following example, ScriptEngineBuildVersion Function returns the configuration version number of the script engine.

Function GetScriptEngineInfo

```
Dim s
```

s = "" 'Script a string with essential information.

s = ScriptEngine & "Version "

s = s & ScriptEngineMajorVersion & "."

s = s & ScriptEngineMinorVersion & "."

s = s & ScriptEngineBuildVersion

GetScriptEngineInfo = s 'Result is returned.

End Function

ScriptEngineMajorVersion Function

Description

The major version number of the script language in use is returned.

Syntax

ScriptEngineMajorVersion

Reference

The returned Result Value is equal to the version information contained in the DLL of the script language in use. In the following example, ScriptEngineMajorVersion Function returns the version number of the script engine.

Function GetScriptEngineInfo

Dim s

s = "" 'Script a string with essential information.

s = ScriptEngine & "Version "

s = s & ScriptEngineMajorVersion & "."

s = s & ScriptEngineMinorVersion & "."

s = s & ScriptEngineBuildVersion

GetScriptEngineInfo = s 'Result is returned.

End Function

ScriptEngineMinorVersion Function

Description

The minor version number of the script language in use is returned.

Syntax

ScriptEngineMinorVersion

Reference

The returned Result Value is equal to the version information contained in the DLL of the script language in use. In the following example, SScriptEngineMinorVersion Function returns the minor version number of the script engine.

Function GetScriptEngineInfo

```
Dim s
```

s = "" 'Script a string with necessary information.

s = ScriptEngine & "Version "

s = s & ScriptEngineMajorVersion & "."

s = s & ScriptEngineMinorVersion & "."

s = s & ScriptEngineBuildVersion

GetScriptEngineInfo = s 'Result is returned.

End Function

Second Function

Description

An integer indicating the second from 0 through 59 is returned.

Syntax

Second(time)

The time factor is an expression showing time. If the time factor has the value zero, Null is returned.

Reference

In the following example, Second Function returns the current second.

Dim MySec

MySec = Second(Now) 'MySec has information on the current second.

Sgn Function

Description

An integer showing number sign is returned.

Syntax

Sgn(number)

The number factor should be a valid expression.

Result Value

Sgn Function returns the following values.

Number Value Result of Sgn Function

> 0, 1

= 0, 0

< 0 -1

Reference

The sign of a number factor decides the result value of Sgn Function.

In the following example, Sgn Function decides number signs.

```
Dim MyVar1, MyVar2, MyVar3, MySign

MyVar1 = 12: MyVar2 = -2.4: MyVar3 = 0

MySign = Sgn(MyVar1) '1 is returned.

MySign = Sgn(MyVar2) '-1 is returned.

MySign = Sgn(MyVar3) '0 is returned.
```

Sin Function

Description

The sine value of an angle is returned.

Syntax

Sin(number)

The number factor should be a valid expression showing the angle in a radian.

Reference

Sin Function returns the ratio between lengths of two sides by using alternate angles of a right triangle. This ratio is the one obtained by dividing the length of the count side to the relevant angle by the length of the hypotenuse of a right triangle. The returned results are ranged between -1 and 1.

To convert angle to radian, multiply angle by pi/180 and to convert radian to angle, multiply radian by 180/pi.

.

In the following example, Sin Function returns the tangent value of the angle.

```
Dim MyAngle, MyCosecant

MyAngle = 1.3 ' Define an angle in a radian value.

MyCosecant = 1 / Sin(MyAngle) ' Calculate cosine value.
```

Space Function

Description

The string that contains empty space as many as designated is returned.

Syntax

Space(number)

The number factor is a number of spaces to be filled in the string with.

Appendix1. Visual Basic Script Help

Reference

In the following example, Space Function returns a string that contains spaces as many as designated.

Dim MyString

MyString = Space(10)

'A string that has 10 spaces is returned.

MyString = "He" & Space(10) & "low?"

'10 spaces are inserted between strings.

Split Function

Description

One dimension Array based on 0, which contains sub strings as many as designated is returned.

Syntax

Split (expression[, delimiter[, number[, compare]]]) Split Function Syntax has following factors.

Factor	Description
	A mandatory factor. A literal expression containing sub strings and distinguishing
expression	symbols. If the length of expression is 0, Split returns empty Array, that is, Array
	that has no configuration element and data.
	An optional factor. A string character used to identify the limit of a sub string. If
delimiter	omitted, empty character (" ") is deemed as a distinguishing symbol. Delimiter is 0
	long, a single element containing expression strings is returned.
number	An optional factor. A number of a sub string to be returned1 shows that all of
number	sub strings are to be returned.
oomnoro	An optional factor. A number value showing the comparison type of the sub string
compare	to be used. For the value, refer to the following settings.

Settings

Comparison factors have the following values

Constant	Value	Description
vbBinaryCompare	0	Run Binary Comparison.
vbTextCompare	1	Run Text Comparison.

Reference

In the following example, Split Function returns the string. This Function executes text comparison of distinguishing characters and returns all of sub strings.

Dim MyString, MyArray

MyString = Split("VBScriptXisXfun!", "x", -1, 1)

- ' MyString(0) gets "VBScript".
- ' MyString(1) gets "is"
- ' MyString(2) gets value"fun!"

Sqr Function

Description

The square root of a number is returned.

Syntax

Sqr(number)

The number factor is a valid expression that is equal or higher than 0.

Reference

In the following example, Sqr Function calculates the square root of a number.

Dim MySqr

MySqr = Sqr(4) '2 is returned.

MySqr = Sqr(23) '4.79583152331272 is returned.

MySqr = Sqr(0) ' 0 is returned.

MySqr = Sqr(-4) 'It causes runtime errors.

StrComp Function

Description

The result value of comparing strings is returned.

Syntax

StrComp(string1, string2[, compare])

StrComp Function Syntax has following factors.

factor	Description
string1	A mandatory factor. A valid character expression.
string2	A mandatory factor. A valid character expression.
icompare	An optional factor. A number value displayed in a comparison type to be used for assessing
	strings. If omitted, Run Binary Comparison. For the value, refer to the following settings.

Settings

Comparison factors have the following values.

Constant	Value	Description
vbBinaryCompare	0	Run Binary Comparison.
vbTextCompare	1	Run Text Comparison.

Result Value

StrComp Function returns the following values.

Condition	Result of StrComp Function
If string1 is less than string2	-1
If string1 is equal to string2	0
If string1 is more than string2	1
If string1 or string2 is Null	Null

Reference

In the following example, StrComp Function returns the result of comparing strings. If the third factor is 1, a text comparison is executed. If the third factor is 0 or omitted, a binary Comparison is performed.

Dim MyStr1, MyStr2, MyComp

String Function

Description

The expression converted to a String subtype of Variant is returned.

Syntax

CStr(expression)

The expression factor should be a valid expression.

Reference

Typically, if a code is created with a subtype of conversion function, the operation result can be seen in a particular data type rather than in a basic data type. For example, if CStr is used, the result is expressed in a String.

If CStr Function is used instead of Str Function, other types of data can be converted to a widely-used String subtype throughout the world. For example, symbols identified with different digit numbers are properly recognized according to the computer's locale settings.

The data used in an expression decide the returned results as seen in the table.

Expression Data	CStr Returned Result
Boolean	String containing True or False
Date	String containing simplified date by computer settings
Null	Runtime error
Empty	String whose length is 0 ("")
Error	String showing error number next to word 'Error'
Other Numbers	String containing numbers

In the following example, CStr Function converts the value to String.

Dim MyDouble, MyString

MyDouble = 437.324 ' MyDouble is equal to Double.

MyString = CStr(MyDouble) ' MyString gets value"437.324"

StrReverse Function

Description

The reverse string of a designated string is returned.

Syntax

StrReverse(string1)

The string1 factor is a string of characters in the reverse order. If string1 is a string whose length is 0(""), it

returns a string whose length is 0 while if it is Null, errors takes place.

Reference

In the following example, Replace Function returns a string in the reverse order.

Dim MyStr

MyStr = StrReverse("VBScript") 'MyStr gets "tpircSBV."

Tan Function

Description

The tangent value of an angle is returned.

Syntax

Tan(number)

The number factor should be a valid expression showing the angle in a radian.

Reference

Tan Function returns the ratio between lengths of two sides by using angles of a right triangle. This ratio is obtained by dividing the length of the side facing the relevant angle by the length of the side adjacent to this angle.

To convert angle to radian, multiply angle by pi/180 and to convert radian to angle, multiply radian by 180/pi.

In the following example, Tan Function returns the tangent value of the angle.

Dim MyAngle, MyCotangent

MyAngle = 1.3 'Define an angle in a radian value.

MyCotangent = 1 / Tan(MyAngle) 'Calculate cotangent Value.

Time Function

Description

A Date subtype Variant showing the current system time is returned.

Syntax

Time

Reference

In the following example, Time Function returns the current system time.

Dim MyTime

MyTime = Time 'The current system time is returned.

TimeSerial Function

Description

The Date subtype of Variant of the designated year, month, day is returned.

Syntax

TimeSerial(hour, minute, second)

TimeSerial Function Syntax has following factors.

Appendix1. Visual Basic Script Help

Factor	Description		
time	Numbers (Including 0 and 23) between 0(AM 12:00) and 23(PM 11:00) or a numerical		
	<u>expression</u>		
minute	All expressions		
second	All expressions		

Reference

To designate time such as 11:59: 59, each TimeSerial factor number should be within the tolerance range of the relevant factor. That is, it should be between 0 and 23 for time and between 0 and 59 for minute or second. However, it is also possible to designate relative time of each factor by using a numerical expression showing hour, minute and second before and after a particular time.

The following example is using an expression rather than an absolute time. Here, TimeSerial Function returns the time 15 minutes before 6o'clock, that is AM 5:45.

TimeSerial(12 - 6, -15, 0)

If the factor exceeds the tolerance range, it goes to the next bigger unit. For example, if 75 minutes are designated, it is calculated into 1:15. But, if a certain factor exceeds range -32,768~32 and 767 or a date designated directly or through an expression by using three factors, errors take place.

TimeValue Function

Description

The Date subtype of Variant that contains time is returned.

Syntax

TimeValue(time)

Typically, it is the time factor as a literal expression showing time from 0:00:00 (AM 12:00:00) to 23:59:59(PM 11:59:59), therefore, an expression showing time within this range can be a time factor. The time has the value zero, Null is returned.

Reference

A valid time unit can be inputted to the 12-h or 24-h clock. That is, "2:24PM" and "14:24" are all valid time factors. If the time factor contains time information, TimeValue Value Function does not return the information. However, if the time factor has wrong date information, errors takes place.

In the following example, TimeValue Function converts a string to time. If the date literal is used, it is possible to directly designate time to variant (Ex, MyTime = #4:35:17 PM#).

Dim MyTime

MyTime = TimeValue("4:35:17 PM") 'MyTime gets value"4:35:17 PM".

Trim Function

Description

LTrim Function returns a string copy which has no left empty space or no right empty space or neither left nor right empty space.

Syntax

LTrim(string)

RTrim(string)

Trim(string)

The string factor should be a valid character expression. If there is Null in string, Null is returned.

Reference

In the following example, LTrim Function trims the left empty space, RtrimFunction trims the right empty space and Trim Function trims both of them.

Dim MyVar

```
MyVar = LTrim(" vbscript ") 'MyVar gets "vbscript."

MyVar = RTrim(" vbscript ") 'MyVar gets " vbscript."

MyVar = Trim(" vbscript ") 'MyVar gets "vbscript."
```

TypeName Function

Description

A string that provides Variant subtype information on variable is returned.

Syntax

TypeName(varname)

A mandatory factor. The varname factor may be any kind of variable.

Result Value

TypeName Function returns the following values.

Value	Description
Byte	Byte Value
Integer	Integer Value
Long	Long integer Value
Single	Single precision floating point number value
Double	Double precision floating point number value
Currency	Call Value
Decimal	Decimal Value
Date	Date or Time Value
String	String Value
Boolean	Boolean Value True or False
Empty	Not initialized
Null	No valid data
<object type=""></object>	Real type name of object
Object	General object
Unknown	Unknown object type
Nothing	Object variable that does not refer to object instance
Error	Error

Reference

In the following example, TypeName Function returns variable reference information.

Dim ArrayVar(4), MyType

Appendix1. Visual Basic Script Help

NullVar = Null 'Null Designate a value.

MyType = TypeName("VBScript") ' "String" is returned.

MyType = TypeName(4) ' "Integer" is returned.

MyType = TypeName(37.50) ' "Double" is returned.

MyType = TypeName(NullVar) ' "Null" is returned.

MyType = TypeName(ArrayVar) ' "Variant()" is returned.

UBound Function

Description

The biggest subscript available in the designated Array level is returned.

Syntax

UBound(arrayname[, dimension])

UBound Function Syntax has following factors.

Factor	Description
orrovnomo	A mandatory factor. Comply with the notation rule of general variable with Array variable
arrayname	name
dimension	An optional factor. An integer showing the upper limit of the returned level is returned. Use
difficitsion	integers 1 for one dimension, 2 for two dimension, etc. If dimension is omitted, 1 is deemed.

Reference

LBound Function is used along with UBound Function to decide the size of Array. If Lbound Function is used, the lower limit can be known in the Array level.

The lower limit of the entire dimensions is always 0. Therefore, UBound Function returns Array values as following according to each dimension.

Dim A(100,3,4)

Statement	Result Value
UBound(A, 1)	100
UBound(A, 2)	3
UBound(A, 3)	4

UCase Function

Description

A string converted to decimal point is returned.

Syntax

UCase(string)

The string factor should be a valid character expression. If there is Null in string, Null is returned.

Reference

This function converts only capital letters to small letters and remains other characters except those as they are without converting.

In the following example, UCase Function returns the capital version of a string.

Dim MyWord

MyWord = UCase("Hello World") ' "HELLO WORLD" is returned.

VarType Function

Description

The value showing variable subtype is returned.

Syntax

VarType(varname)

The varname factor may be any kind of variable.

Result Value

VarType Function returns the following values.

Constant	Value	Description		
vbEmpty	0	Empty(Not initialized)		
vbNull	1	Null(No valid data)		
vbInteger	2	Integer		
vbLong	3	Long integer		
vbSingle	4	Single precision floating point number		
vbDouble	5	Double precision floating point number		
vbCurrency	6	Call		
vbDate	7	Date		
vbString	8	String		
vbObject	9	Automatic Object		
vbError	10	Error		
vbBoolean	11	Boolean Value		
vbVariant	12	Variant(used only in variant Array)		
vbDataObject	13	Data Access Object		
vbByte	17	Byte		
vbArray	8192	Array		

Memo

It is possible to use those names in any place instead of actual values since they are the Constant designated in the VBScript.

Reference

VarType Function itself never returns the relevant Array value and shows a particular type of Array by being combined with other values at all times. The value corresponding to Variant is returned only when it shows that VarType Function is Array by being combined with a particular value. For example, the integer value returned to Array is calculated to 2+8192 or 8194. If an object that has basic properties, VarType (object) Function returns the basic properties of the relevant object.

In the following example, VarType Function decides variable's subtype.

Dim MyCheck

MyCheck = VarType(300) '2 is returned. MyCheck = VarType(#10/19/62#) '7 is returned. MyCheck = VarType("VBScript") '8 is returned.

Weekday Function

Description

An integer showing weekday is returned.

Syntax

Weekday(date, [firstdayofweek])

Weekday Function Syntax has following factors.

Factor	Description		
date	An expression showing <u>Date</u> . If there is date <u>Null</u> in <i>date</i> , Null is returned.		
firstdayofweek	Constant that designates the first weekday. If omitted, vbSunday is deemed.		

Settings

The firstdayofweek factor has the following settings:

Constant	Value	Description		
vbUseSystem	0	NLS(National Language Support). Use API settings		
vbSunday	1	Sunday		
vbMonday	2	Monday		
vbTuesday	3	Tuesday		
vbWednesday	4	Wednesday		
vbThursday	5	Thursday		
vbFriday	6	Friday		
vbSaturday	7	Saturday		

Result Value

Typically, Weekday Function returns the following values.

Constant	Value	Description
vbSunday	1	Sunday
vbMonday	2	Monday
vbTuesday	3	Tuesday
vbWednesday	4	Wednesday
vbThursday	5	Thursday
vbFriday	6	Friday
vbSaturday	7	Saturday

Reference

In the following example, Weekday Function brings a weekday from the designated date

Dim MyDate, MyWeekDay

MyDate = #October 19, 1962# 'Designate date.

MyWeekDay = Weekday(MyDate) 'MyWeekDay gets value 6 because MyDate is on Friday .

WeekdayName Function

Description

A string showing the designated weekday is returned.

Syntax

WeekDayName(weekday, abbreviate, firstdayofweek) WeekDayName Function Syntax has following factors.

Fastan	a distance of the first state of					
Factor	Description					
weekday	A mandatory factor. A number showing weekday. Each weekday has a different value					
Weekday	according to firstdayofweek settings.					
	An optional factor. A boolean value showing whether weekday names are expressed with					
abbreviate	abbreviated words. If omitted, it becomes False, the Default Value, full names are					
	displayed.					
finatela va fiva ale	An optional factor. A number value showing the first weekday. For the value, refer to the					
firstdayofweek	following settings.					

Settings

The value of the firstdayofweek factor is as follows:

Constant	Value	Description		
vbUseSystem	0	NLS(National Language Support). Use API settings		
vbSunday	1	Sunday(Default Value)		
vbMonday	2	Monday		
vbTuesday	3	Tuesday		
vbWednesday	4	Wednesday		
vbThursday	5	Thursday		
vbFriday	6	Friday		
vbSaturday	7	Saturday		

Reference

In the following example, WeekDayName Function returns the designated day.

Dim MyDate

MyDate = WeekDayName(6, True) 'MyDate gets Fri.

Year Function

Description

An integer showing year is returned.

Syntax

Year(date)

The date factor is an expression showing the date. If the date factor has the value zero, Null is returned.

Reference

In the following example, YearFunction brings the date of the year from the designated date

Appendix1. Visual Basic Script Help

Dim MyDate, MyYear

MyDate = #October 19, 1962# 'Designate date.

MyYear = Year(MyDate) 'MyYear gets 1962

Appendix 2 LSIS InfoU OPC Server Help

2.1 Introduction

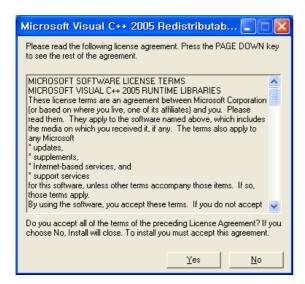
OPC is intended to standardize communication with various industrial devices under the Microsoft window operation system. This operates based on COM/DCOM functions of the window operation system, consisting of two kinds of software- OPC Client software such as InfoU and OPC Server software. This session explains how to use the Company's OPC Server that can exchange data through **OPC (OLE for Process Control)-** method communication. LS InfoU OPC Server presents a Configuration Program to select tags acquired from LSIS OPC DA Server, which will provide data during the runtime, and OPC DA Server. The Configuration Program contains monitoring function for local and remote diagnosis with its own InfoU OPC Server.

2.2 Install Program

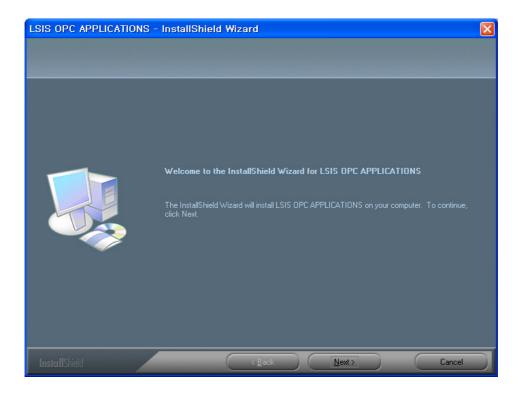
2.2.1 Install

The program is installed according to the following sequence.

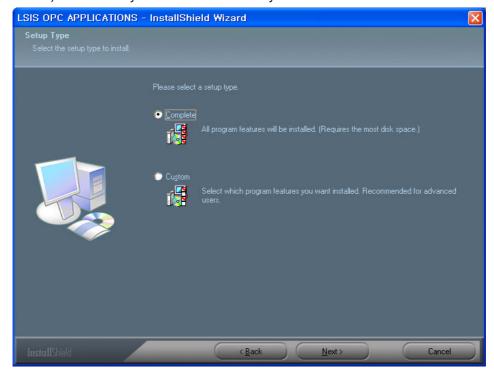
- (1) Install .NET Framework 2.0
 - (a) Run "dotnetfx.exe"
- (2) Install Microsoft Visual C++ 2005 Run Time library"
 - (a) Run "vcredist_x86.exe"
- (3) Install LS InfoU OPC DA Server"
 - (a) Run "LSISOPC Server For InfoU V 1 4.exe"
 - (b) Select "Yes" to "Accept" all of the terms of the License Agreement for using Microsoft Visual C++ 2005 Run Time library."



Appendix2. LSIS InfoU OPC Server



- (c) Select the "Setup" type
- 1) Complete: All program features will be installed.
- 2) Custom: Only the selected features by the user will be installed.



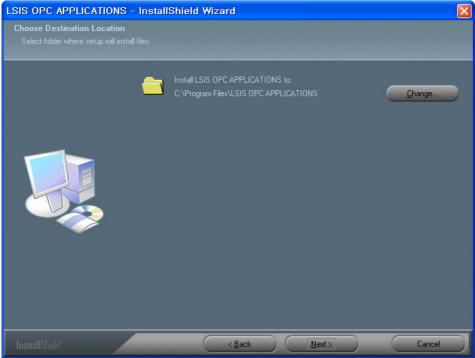
3) Choose a destination location. When changing the default folder, press "Change" button to select a folder where setup files will be installed.

3) Choose a destination location. When changing the default folder, press "Change" button to select a folder where setup files will be installed.

3) Choose a destination location. When changing the default folder, press "Change" button to select a folder where setup files will be installed.

3) Choose a destination location. When changing the default folder, press "Change" button to select a folder where setup files will be installed.

3) Choose a destination location. When changing the default folder, press "Change" button to select a folder where setup files will be installed.

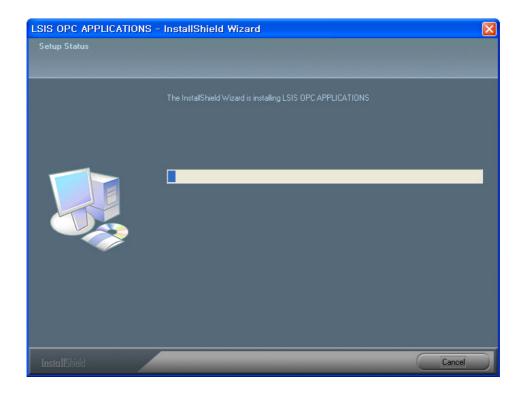


2.2.2 Start Installation

(1) Once all of settings are completed, press "Install" button to start installing.

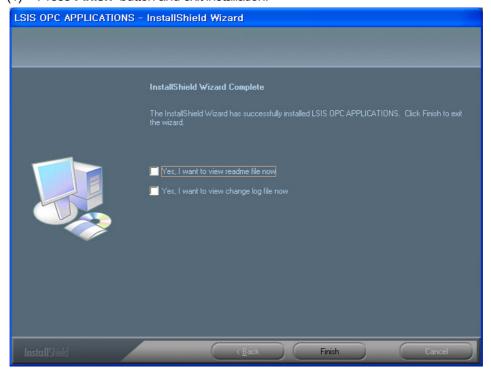


Appendix2. LSIS InfoU OPC Server



2.2.3 Complete Installation

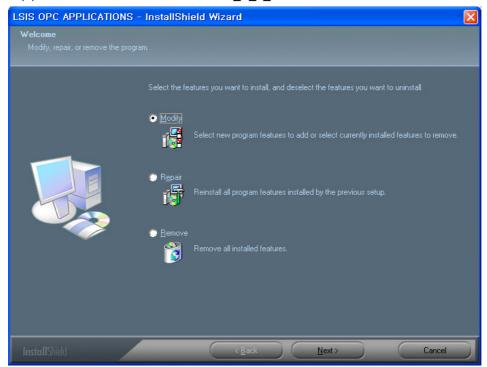
(1) Press"Finish" button and exit installation.



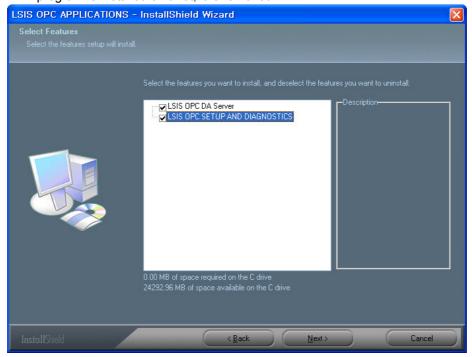
2.2.4 Modify/Repair/Remove Program

(1) Modify

(a) Run "LSISOPC Server For InfoU_V_1_4.exe"

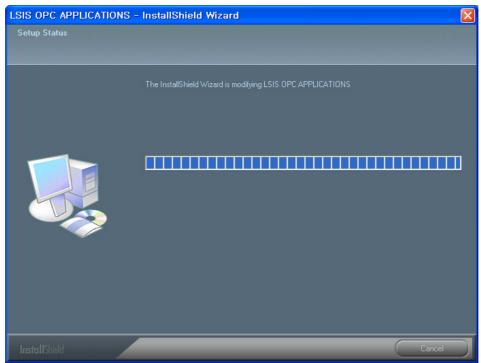


(b) Decide whether to install or remove each program. If the check box is selected, the relevant program is installed or it not, it is removed.

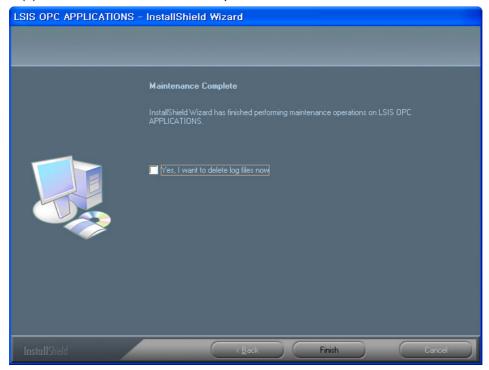


Appendix2. LSIS InfoU OPC Server

(c) Click on"Next"to run and complete modifying.

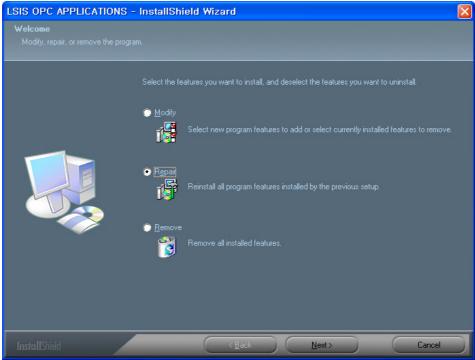


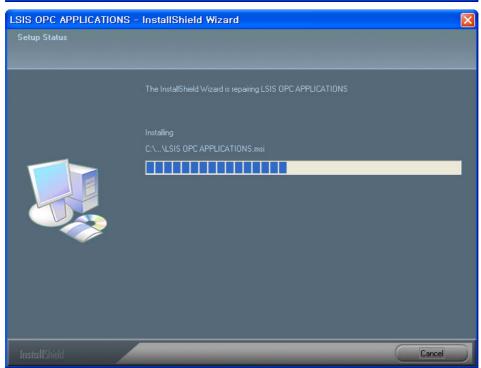
(d) Press "Finish" button and complete the installation.



(2) Repair

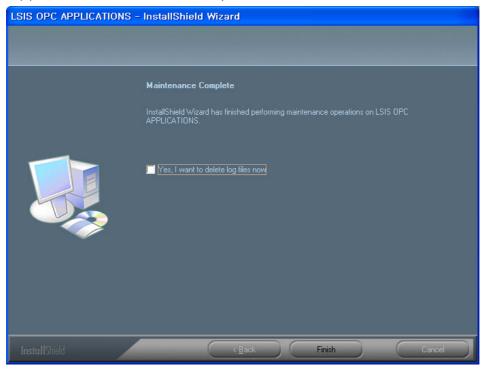
- (a) Run "LSISOPC Server For InfoU_V_1_4.exe" file.
- (b) To reinstall all program features installed by the previous setup, select "Repair."
- (c) Click on "Next" to run and complete repairing.





Appendix2. LSIS InfoU OPC Server

(d) Press "Finish" button and complete the installation.



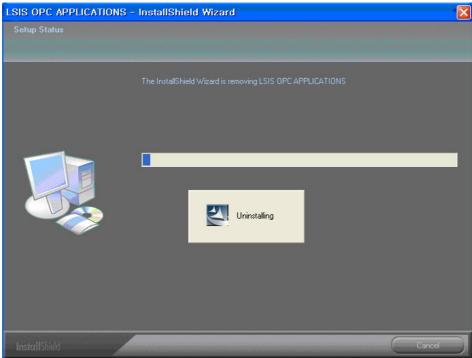
(3) Remove

- (a) Run "LSISOPC Server For InfoU_V_1_4.exe"file.
- (b) Select "Remove" to remove all OPC programs related to installation.

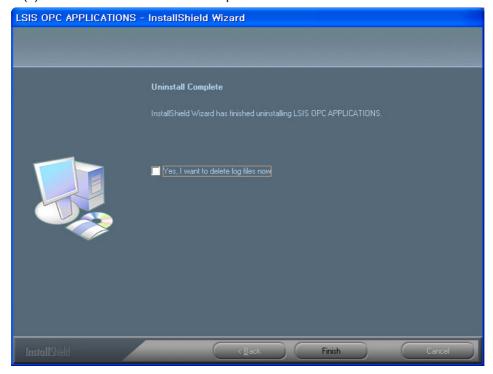


(c) - To remove, click on "Yes" button.



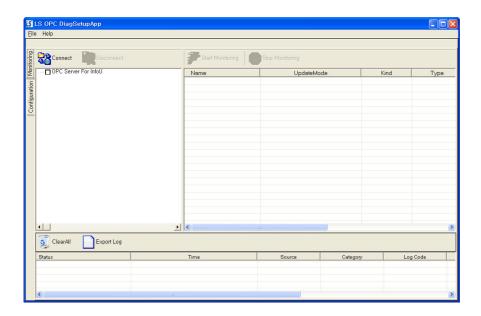


(d) Press "Finish" button and complete the installation.



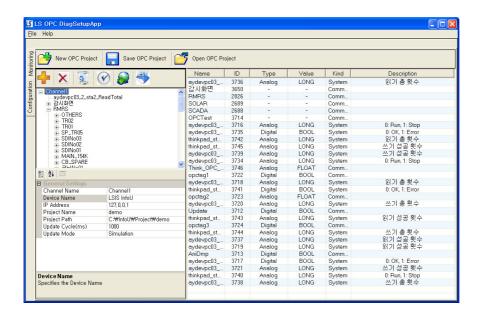
2.3 Configuration of Default Screen

Select "Start -> All Programs -> LSIS OPC Server APPLICATION-> LSIS OPC SETUP AND DIAGNOSTICS" on the operation display line.



The program above has OPC DA Server Configuration and OPC DA Server Monitoring features. Among them, this shows a monitoring screen during the execution. If "Configuration" tap is selected from the program, it becomes available to select a tag list to be provided by the server as a configuration screen for OPC DA Server.

2.3.1 Configuration



(1) Toolbar (OPC Project)



- (a) New OPC Project: Contain information on new tags to be acquired in the runtime and create a new OPC project.
- (b) Save OPC Project: Save a new or modified OPC project.
- (c) Open OPC Project: Load an existing OPC project to be acquired by OPC DA server in the runtime.

(2) Toolbar (Channel)



- (a) : Add a channel, that is, set up one infoU project to be executed on other computer for each channel.
- (b) : Delete the selected channel.
- (c) Delete all of the channels registered with OPC project.
- (d) Check the selected channel for its communication status.
- (e) Zz : Load tag information on the selected channel from InfoU project.
- (f) : Load the created CSV from InfoUD and set up tags.

(3) Channel Window

View the tag DB loaded from the selected channel's InfoU project in a tree view.



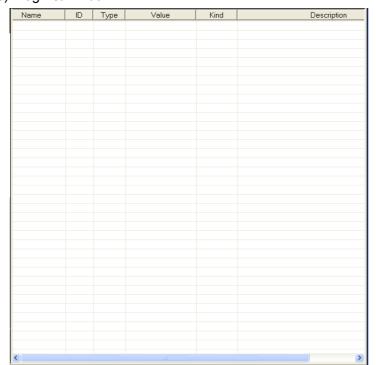
Appendix2. LSIS InfoU OPC Server

(4) Channel Property Window

⊟	General Settings			
	Channel Name	Channel1		
	Device Name	LSIS InfoU		
	IP Address	127, 0, 0, 1		
	Project Name	Demo		
Project Path		C:₩InfoU₩Project₩		
	Update Cycle(ms)	1000		
	Update Mode	Realtime		
Г				
l				
l				
ı				
L				

- (a) Channel Name: The channel name should be a unique one. It helps the user to identify InfoU project.
- (b) Device Name: It lets the user know it is LSIS InfoU.
- (c) IP Address: Set up the computer's IP address where InfoU project runs.
- (d) Project Name: Set up the name of a project which is currently running on InfoU.
- (e) Project Path: Set up a location where InfoU project exists.
- (f) Update Cycle(ms): Set up a cycle to import tag values from InfoU.
- (g) Update Mode: Decide whether to provide real tag values on InfoU or simulation in OPC DA Server.

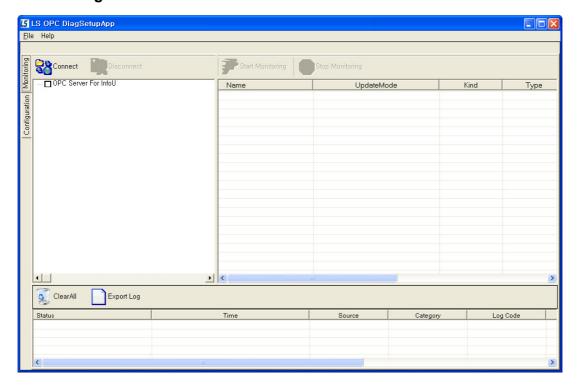
(5) Tag List Window



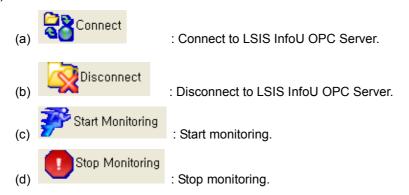
(6) Description Window: Description on each field of the Property Window is displayed.



2.3.2 Monitoring



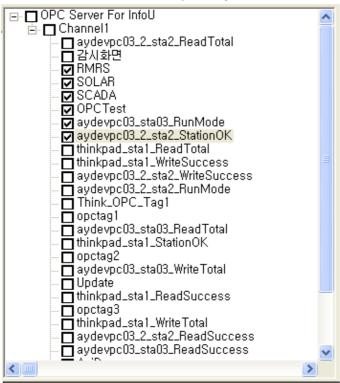
(1) Toolbar



Appendix2. LSIS InfoU OPC Server

(2) Tag Window

(a) This window shows all of the tag lists registered with the connected LSIS InfoU OPC Server and allows the user to select only the tags to be monitored.



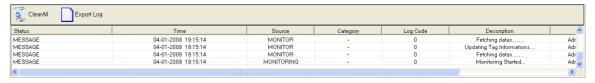
(3) Monitoring Window

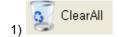
(a) This window shows the selected tags' values.

Name	UpdateMode	Kind	Туре	Value	Time
Channel1/RMRS	Realtime	Common		0	01-01-1601 09:00:0
Channel1/SOLAR	Realtime	Common		0	01-01-1601 09:00:0
Channel1/SCADA	Realtime	Common		0	01-01-1601 09:00:0
Channel1/OPCTest	Realtime	Common		0	01-01-1601 09:00:0
Channel1/aydevpc03_sta0	Realtime	System		0	01-01-1601 09:00:0
Channel1/aydevpc03_2_st	Realtime	System		False	01-01-1601 09:00:0
<					>

(4) Log Window

(a) This window shows the log information generated during the monitoring.



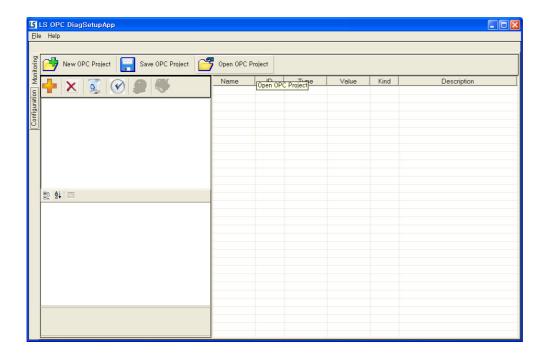


: Delete the contents from the log list.



: Export log information to a file

2.4 OPC Project Configuration



2.4.1 New OPC Project



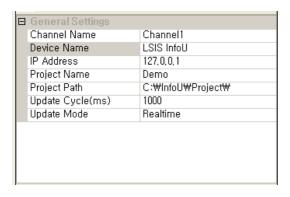


(2) To add a channel to the OPC project, click on

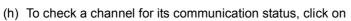


(3) On the created channel property window, select basic information on InfoU project to be acquired from LS InfoU OPC DA Server and a cycle to acquire data.

Appendix2. LSIS InfoU OPC Server



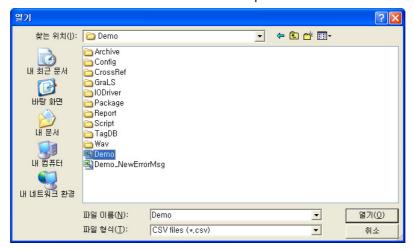
- (a) Channel Name: The channel name should be a unique one. It helps the user to identify InfoU project.
- (b) Device Name: It lets the user know it is LSIS InfoU.
- (c) IP Address: Set up the computer's IP address where InfoU project runs.
- (d) Project Name: Set up the name of a project which is currently running on InfoU.
- (e) Project Path: Set up a location where InfoU project exists.
- (f) Update Cycle(ms): Set up a cycle to import tag values from InfoU.
- (g) Update Mode: Decide whether to provide real tag values on InfoU or simulation in OPC DA Server.





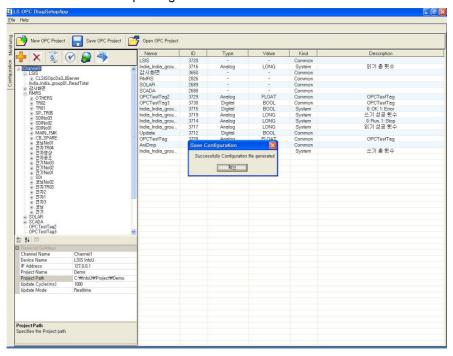


(i) To load a tag list from the channel to register, click on . Or, click on to select relevant CSV file created on InofUD and import it.



(j) Click on Save OPC Project to create Configuration File(LSISOpcDaServer.config.xml),

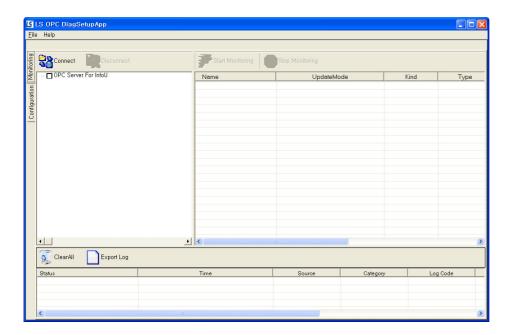
a OPC project. Configuration File is created at the position where the program is installed and this file is loaded to import tag data when LSIS InfoU OPC DA Server starts running.



2.4.2 Open OPC Project / Modify

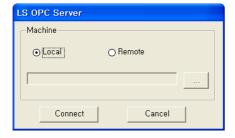
- (1) Select to import Configuration File (LSISOpcDaServer.config.xml), one of existing OPC projects.
- (2) Update the Project Name, Project path or IP Address on the channel property window.
- (3) Click on Save OPC Project to save the modified information on OPC project in Configuration File(LSISOpcDaServer.config.xml).

2.5 Run Monitoring

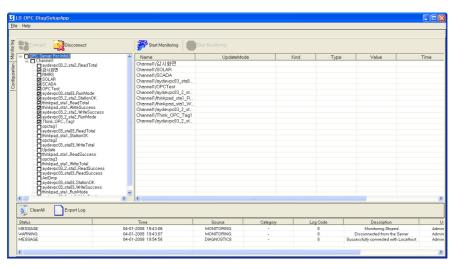


2.5.1 Connect LS InfoU OPC Server / Start Monitoring

(1) Click on tool bar. The following window appears to allow the user to select an OPC server to be connected to.

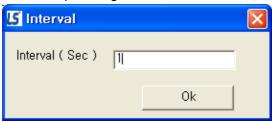


(2) To select Local, click on "Connect" button. To select Remote, press"Remote" radio button and directly input the name of a computer on which LS InfoU OPC Server runs or input the name through the browser and then, click on "Connect" button.

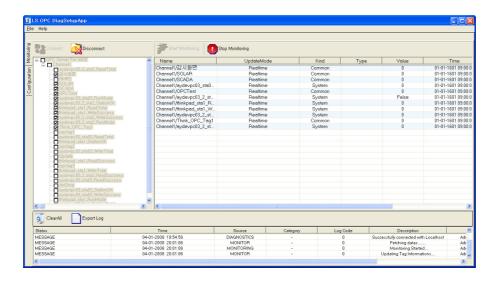


- (3) Select a tag to be monitored through the tag Window. Register the selected tag with the monitoring Window.
- (4) Click on Start Monitoring

 Toolbar. The following window appears to allow the user to select an interval to update tag data from LS InfoU OPC Server. Input seconds for interval and select OK button.



(5) Start monitoring after making the tag window inactive because it is impossible to select additional tags to be monitored any more. The monitoring window shows the results of monitoring on tag data.



2.5.2 Disconnect LS InfoU OPC Server / Stop Monitoring

(1) To stop monitoring, click on



(2) To stop connecting to the LSIS InfoU OPC Server, click on



2.6 DCOM Configuration

DCOM configuration is necessary to connect to OPC server on the remote computer. For DCOM configuration, there are computer configuration and LS InfoU OPC DA Server configuration. In this appendix, configuration is explained with an example of DCOM configuration on the Window XP service Pack 2 and similar procedure applies to other OSs, too.

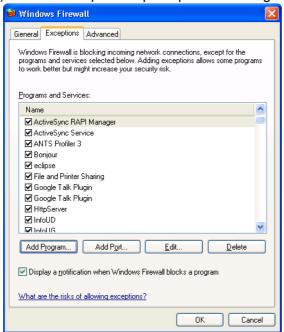
Basic Procedure)

- The local computer and the remote computer should be registered with the same account, sharing the same password.
- The local computer and the remote computer should be logged in with the same account
 - To use OPC DA Server on the remote computer, basic OPC files have been installed on the local computer.

2.6.1 Firewall Configuration

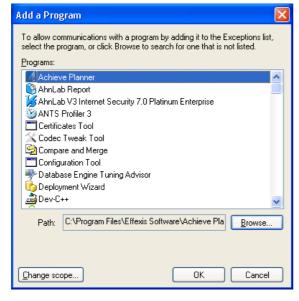
(1) Select"Start->Control Board->Security Center->Window Firewall." "On (recommended)"has been set up as the default configuration. This configuration is recommended to protect the computer from Microsoft and OPC. However, select"Not Use" for OPC communication or follow the instruction below:

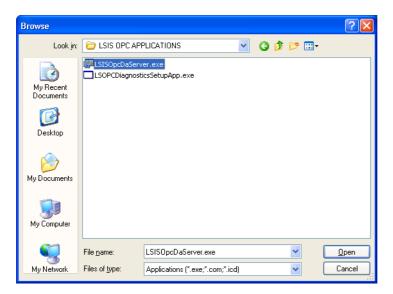


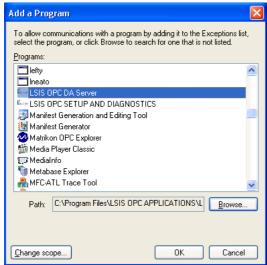


(2) Select "Exceptions" tap and press "Add Program" as below.

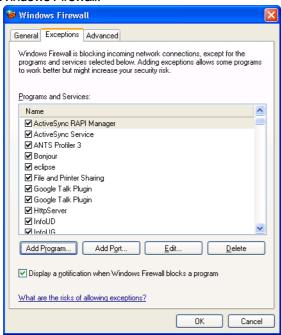
(3) Select "Browse" button to register LSISOpcDaServer.exe and LSOPCDiagnosticsSetuipApp.exe - the runtime and the configuration program of LSIS InfoU OPC DA Server respectively.



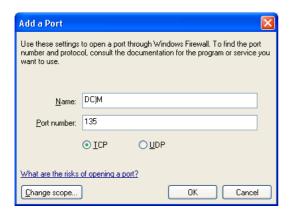




(4) Add TPC Port 135 to initialize DOCM communication. Click on "Add Port" button on "Exceptions" tap of Windows Firewall.

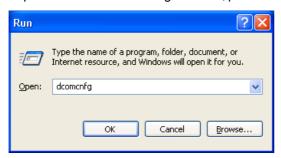


- (5) Input the following items on" Add a Port" window.
 - (a) Name: DCOM(b) Port Number: 135
 - (c) "TCP" Select radio button

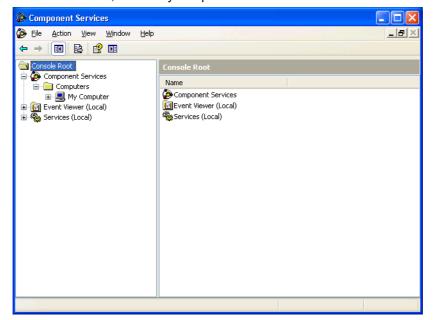


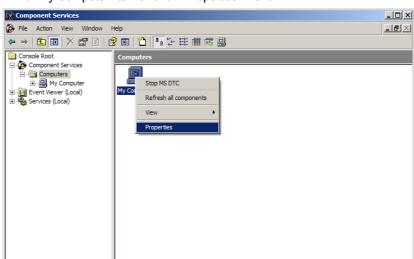
2.6.2 DCOM Configuration

- (1) DCOM Computer Configuration
 - (a) Input"Start->Run->DCOMCnfg" and then, press "OK" button.



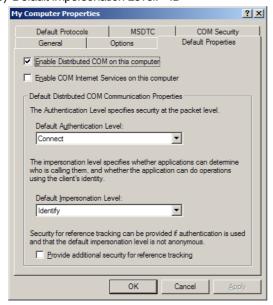
(b) "Component Services" window is displayed as below. Click on Console Root to enlarge the Component Services and then, click on My Computer.





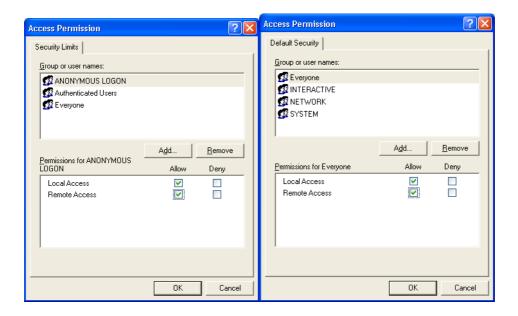
(c) Find "My Computer" item and run "Properties" menu.

- (d) Set up designated values for the following items on "Default Properties" tap as below.
- 1) "Enable Distributed COM on this computer": Check "Use."
- 2) Default Authentication Level: Connect.
- 3) Default Impersonation Level: ID

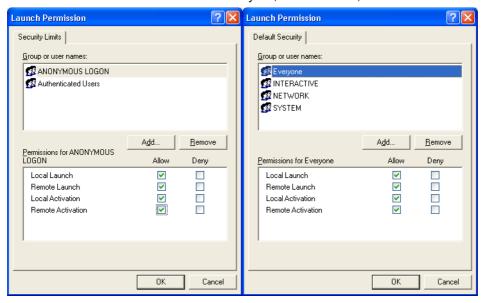


(e) After selecting "COM Security" tap, set up "Access Permission" and "Launch Permission" as below.

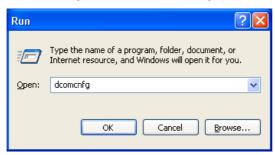
- (f) For "Access Permission," set up "Security Limits" and "Default Security."
- 1) Security Limits: After adding ANONYMOUS LOGIN, Everyone and Authenticated Users to 'Group or User Names' item, Check $\sqrt{}$ in all of the boxes under "Allow" for "Local Assess" and "Remote Access.".
- 2) Default Security: Check $\sqrt{\ }$ in all of the boxes under "Allow" for "Local Access" and "Remote Access" for Everyone, INTERACTIVE, NETWORK and SYSTEM user.



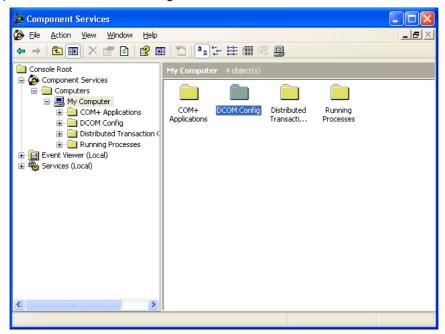
- (g) For "Launch Permission," set up "Security Limits" and "Default Security" as below.
- 1) Security Limits: Check √ in all of the boxes under "Allow" for "Local Launch", "Remote Launch", "Local Launch" and "Remote Launch" for ANONYMOUS LOGON and Authenticated Users
- 2) Default Security: Check √ in all of the boxes under "Allow" for "Local Launch", "Remote Launch", "Local Launch" and "Remote Launch" for Everyone, INTERACTIVE, NETWORK and SYSTEM user.



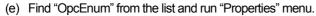
- (2) DCOM configuration of LSIS InfoU OPC Server
 - (a) After entering "Start->Run->DCOMCnfg," press OK button.

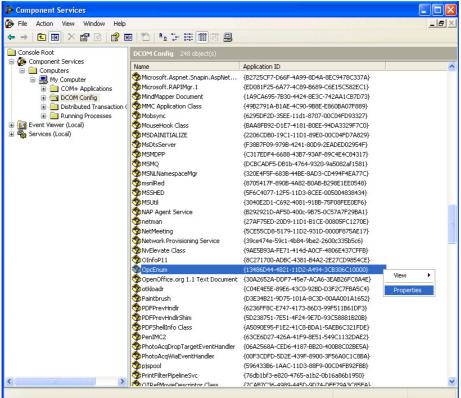


- (b) "Component Services" window is displayed as below. Click on Console Root to enlarge the Component Services and then, click on Computers. Find My Computer and set up properties.
- (c) Double click on "DCOM Config."

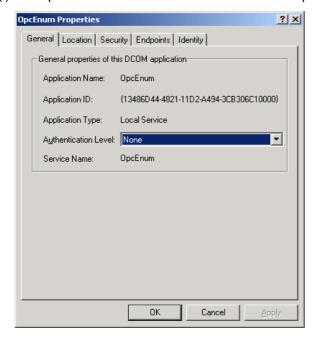


(d) Set up properties on "OpcEnum" to permit ANONYMOUS LOGON. (If "OpcEnum" is not used, Remote Access by any anonymous user is not permitted.)

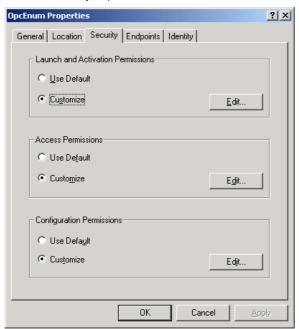




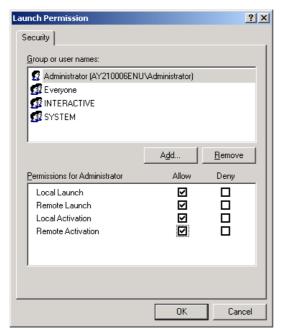
(f) Set up "None" for "Authentication Level" on "General" tap of "OpcEnum Property" Window.



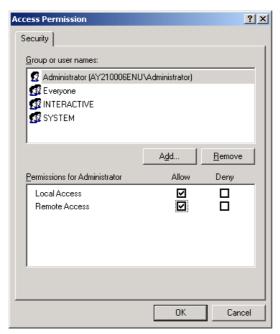
(g) Select "Security" tap.



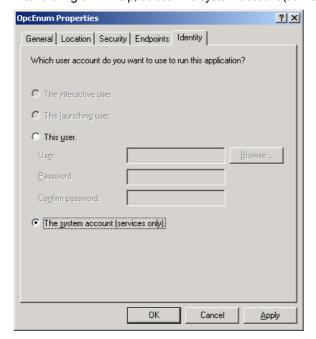
- (h) After selecting "User Specified" for "Launch Permission" and pressing "Edit" button, set up as below.
- 1) "Group or User Names": Check $\sqrt{\ }$ in all of the boxes under "Allow" for "Local Launch", "Remote Launch", "Local Launch" and "Remote Launch" for each user.



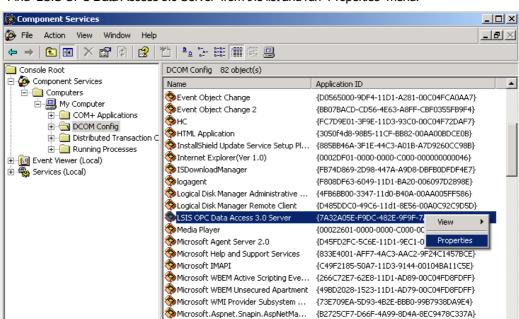
- (i) After setting up "Access Permission" and pressing "Edit" button, set up as below.
- 1) "Group or User Names": Check √ in all of the boxes under "Allow" for "Local Access" and "Remote Access" for each user.



(j) After clicking on "ID" tap, select "The system account (services only)"



{49B2791A-B1AE-4C90-9B8E-E860BA07F889}

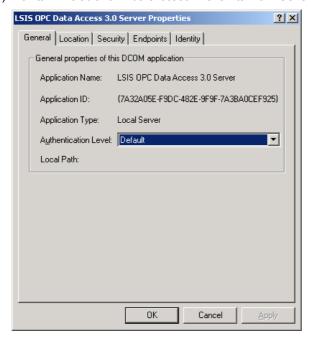


(k) Find" LSIS OPC Data Access 3.0 Server" from the list and run "Properties" menu.

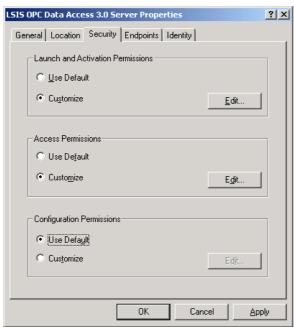
(I) "LSIS OPC Data Access 3.0 Server Property" Window is displayed.

MMC Application Class

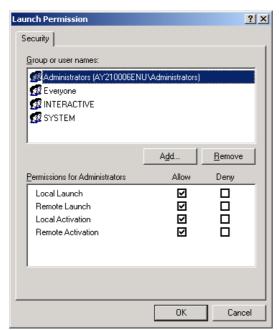
(m) Maintain "Default" for "Authentication Level" item on "General" tap.



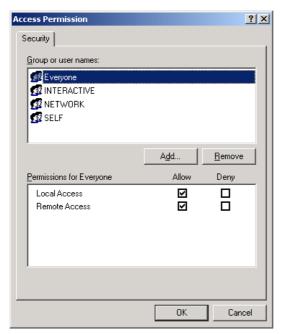
(n) Select "Security" tap.



- (o) After selecting "User Specified" for Launch Permission" and pressing "Edit" button, set up as below.
 - 1) "Group or User Names": Check $\sqrt{\ }$ in all of the boxes under "Allow" for "Local Launch", "Remote Launch", "Local Activation" and "Remote Activation" for each user.



- (p) After setting up "Access Permission" and pressing "Edit" button, set up as below.
 - 1) "Group or User Names": Check √ in all of the boxes under "Allow" for "Local Access" and "Remote Access" for each user.



- (q) Set up "Use Default" for "Configuration Authority."
- (r) After selecting "Identity" tap, select "The interactive user."



(s) After clicking on "Apply" button, press "OK" button.



LS values every single customer.

Quality and service come first at LSIS.

Always at your service, standing for our customers.

www.lsis.biz



10310001053

HEAD OFFICE

LS tower,Hogye- dong, Dongan- gu, Anyang- si, Gyeonggi- do 1026- 6, Korea http://eng.lsis.biz
Tel. (82-2)2034- 4689~ 4888 Fax.(82-2)2034- 4648

■ LS Industrial Systems Tokyo Office >> Japan

Address: 16F, Higashi- Kan, Akasaka Twin Towers 17- 22, 2- chome, Akasaka, Minato- ku, Tokyo 107- 8470, Japan Tel: 81- 3- 3582- 9128 Fax: 81- 3- 3582- 2667

e- mail: jschuna@lsis.biz

■ LS Industrial Systems Dubai Rep. Office >> UAE

Address: P.O.BOX- 114216, API World Tower, 303B, Sheikh
Zayed road, Dubai, UAE. e- mail: hwyim@lsis.biz
Tel: 971- 4- 3328289 Fax: 971- 4- 3329444

■ LS- VINA Industrial Systems Co., Ltd. >> Vietnam

Address: LSIS VINA Congty che tao may dien Viet-Hung

Dong Anh Hanoi, Vietnam e- mail: srjo@hn.vnn.vn

Tel: 84- 4- 882- 0222 Fax: 84- 4- 882- 0220

■ LS Industrial Systems Hanoi Office >> Vietnam

Address: Room C21, 5th Floor, Horison Hotel, 40 Cat Linh, Hanoi, Vietnam

Tel: 84- 4- 736- 6270/1 Fax: 84- 4- 736- 6269

■ Dallan LS Industrial Systems co., Ltd, >> China

Address: No. 15 Liaohexi 3 Road, economic and technical development zone, Dalian, China e- mail: lixk@lgis.com.cn
Tel: 86- 411- 8273- 7777 Fax: 86- 411- 8730- 7560

■ LS Industrial Systems (Shanghai) Co., Ltd. >> China

Address: Room E- G, 12th Floor Huamin Empire Plaza, No. 726, West Yan'an Road, Shanghai, China Tel: 86- 21- 5237- 9977

■ LS Industrial Systems(Wuxi) Co., Ltd. >> China

Address: 102- A National High & New Tech Industrial

Development Area, Wuxi, Jiangsu, China e- mail: Xugh@lgis.com.cn

Tel: 86-510-534-6666 Fax: 86-510-522-4078

■ LS Industrial Systems Beijing Office >> China

Address: B- tower 17th Floor, Beijing Global Trade Center building, No. 36, BeiSanHuanDong- Lu, DongCheng- District, Beijing, China Tel: 86-10-5825-6025

■ LS Industrial Systems Guangzhou Office >> China

Address: Room 1403, 14F, New Poly Tower, 2 Zhongshan Liu
Rad, Guangzhou, China e- mail: zhangch@lgis.com.cn
Tel: 86- 20- 8326- 6754 Fax: 86- 20- 8326- 6287

■ LS Industrial Systems Chengdu Office >> China

Address: Room 2907, Zhong Yin B/D, No. 35, Renminzhong(2)Road, Chengdu, China e- mail: hongkonk@vip.163.com
Tel: 86- 28- 8612- 9151 Fax: 86- 28- 8612- 9236

■ LS Industrial Systems Qingdao Office >> China

Address: 12th Floor, Guodong building, No52 Jindun Road,
Chengdu, China e- mail: bellkuk@hanmail.net
Tel: 86-532-580-2539 Fax: 86-532-583-3793

* LS Industrial Systems constantly endeavors to improve its product so that Information in this manual is subject to change without notice.

InfoU / 2009.9