EasyBuilder Pro

Ver. 6.07.01

Copyright © 2013 Weintek Labs., Inc. All rights reserved.

Table of Contents

1. EasyBuilder Pro Installation and Startup Guide 1-1
   1.1. Installation Requirements 1-2
   1.2. Steps to Install EasyBuilder Pro 1-3
   1.3. Software Update Reminder 1-7
2. Utility Manager 2-1
   2.1. Overview 2-2
   2.2. Design 2-3
   2.3. Analysis & Testing 2-4
   2.4. Publish 2-6
      2.4.1. Download (PC->HMI) 2-7
      2.4.2. Upload (HMI->PC) 2-8
      2.4.3. Build Download Data for SD/USB Disk 2-9
      2.4.4. Steps to Download Project to HMI via USB Disk or SD Card 2-10
   2.5. Maintenance 2-10
   2.6. Data Conversion 2-12
3. Create an EasyBuilder Pro Project 3-1
   3.1. Overview 3-2
   3.2. Create a New Project File 3-2
   3.3. Save and Compile the Project File 3-4
   3.4. Run On-Line or Off-Line Simulation 3-5
   3.5. cMT Viewer 3-5
   3.6. Download the Project File to HMI 3-6
      3.6.1. Configure in EasyBuilder Pro 3-6
      3.6.2. Use HMI Name 3-9
      3.6.3. Use USB cable 3-11
      3.6.4. Use USB Drive or SD Card 3-11
   3.7. Upload the Project File from HMI 3-13
4. Hardware Settings 4-1
   4.1. Overview 4-2
   4.2. I/O Ports 4-2
   4.3. LED Indicators 4-2
   4.4. System Reset 4-3
   4.5. System Toolbar 4-4
      4.5.1. System Setting 4-4
      4.5.2. System Information 4-12
5. System Parameter Settings 5-1
   5.1. Overview 5-2

5.2. Device 5-2
5.2.1. How to Control a Local PLC 5-2
5.2.2. How to Control a Remote Device 5-7
5.2.3. How to Control a Remote HMI 5-9
5.3. Model 5-11
5.4. General 5-13
5.5. System Setting 5-20
5.6. Remote 5-23
5.7. Security 5-25
5.7.1. General Mode 5-26
5.7.2. Enhanced Security Mode 5-26
5.8. Extended Memory 5-28
5.9. Printer / Backup Server 5-31
5.10. Time Sync./DST 5-32
5.11. e-Mail 5-34
5.12. Cellular Data Network 5-36
5.13. FTP 5-38
6. Window Operations 6-1
6.1. Overview 6-2
6.2. Window Types 6-2
6.2.1. Base Window 6-2
6.2.2. Fast Selection Window 6-2
6.2.3. Common Window 6-3
6.2.4. System Message Window 6-3
6.3. Create, Set, and Delete a Window 6-4
6.3.1. Creating and Setting a Window 6-5
6.3.2. Open, Close, and Delete a Window 6-7
6.4. Window Transparency 6-8
6.5. Editing Tips 6-10
6.5.1. Non-display Area, Select Tool, and Hand Tool 6-10
6.5.2. Ruler 6-11
6.5.3. Quick Copy 6-13
7. Event Log 7-1
7.1. Overview 7-2
7.2. Event Log Management 7-2
7.2.1. eMT, iE, XE, mTV, iP Series 7-3
7.2.2. cMT, cMT X Series 7-4
7.2.3. Excel Editing 7-8

7.2.4. Quick View of Errors 7-9
7.3. Creating a New Event Log 7-9
8. Data Sampling 8-1
8.1. Overview 8-2
8.2. Data Sampling Management 8-2
8.3. Creating a New Data Sampling 8-2
8.3.1. Demonstration of Auto. stop 8-9
8.3.2. Customized File Handling 8-10
8.4. Synchronizing cMT Viewer data and Saving to External Device or Database 8-13
8.5. Checking History Data of a Specific Date / File on cMT Viewer. 8-13
9. Object General Properties 9-1
9.1. Overview 9-2
9.2. Selecting PLC and Setting Read/Write Address 9-2
9.3. Using Shape Library and Picture Library 9-3
9.3.1. Picture 9-3
9.3.2. Shape 9-4
9.3.3. Shape manager 9-6
9.4. Setting Label Text 9-7
9.5. Adjusting Profile Size 9-9
9.6. Overlapping Objects (cMT / cMT X Series) 9-10
9.6.1. Enable Touch Area 9-10
9.6.2. Find Overlapping Objects 9-11
9.6.3. Convert to Combo Button 9-11
10. User Password and Object Security 10-1
10.1. Overview 10-2
10.2. User Password and Operable Object Classes 10-2
10.2.1. General Mode 10-2
10.2.2. Enhanced Security Mode 10-3
10.2.3. LDAP Mode 10-4
10.3. Enhanced Security Mode and Control Address 10-6
10.3.1. Control Address Settings 10-6
10.3.2. Commands 10-7
10.3.3. Command Execution Results 10-9
10.4. Enhanced Security Mode Usage 10-10
10.4.1. Importing User Accounts 10-10
10.4.2. Login with USB Security Key 10-11
10.4.3. Login / Logout Automatically with USB Security Key 10-13
10.4.4. Enhanced Security Mode with Option List Object 10-14

10.4.5. LDAP Mode 10-14
10.5. Object Security Settings 10-23
10.5.1. Security Tab 10-23
10.5.2. Enable/Disable 10-24
10.5.3. User Restriction 10-26
10.6. Example of Object Security Settings 10-28
10.7. Protecting Password Settings from Unauthorized Editing 10-31
10.8. Bulk Changing of Security Settings of Multiple Objects 10-32
11. Index Register 11-1
11.1. Overview 11-2
11.2. Examples of Index Register 11-3
11.3. Examples of Using Tag-Based PLC and Index Registers 11-5
12. Keyboard Design and Usage 12-1
12.1. Overview 12-2
12.2. Steps to Design a Popup Keyboard 12-2
12.3. Steps to Design a Keyboard with Direct Window 12-5
12.4. Steps to Design a Fixed Keyboard on Screen 12-6
12.5. Steps to Design a Unicode Keyboard 12-7
13. Objects 13-1
13.1. Bit Lamp 13-3
13.1.1. Overview 13-3
13.1.2. Configuration 13-3
13.2. Word Lamp 13-5
13.2.1. Overview 13-5
13.2.2. Configuration 13-5
13.3. Set Bit 13-10
13.3.1. Overview 13-10
13.3.2. Configuration 13-10
13.4. Set Word 13-13
13.4.1. Overview 13-13
13.4.2. Configuration 13-13
13.5. Function Key 13-22
13.5.1. Overview 13-22
13.5.2. Configuration 13-22
13.6. Toggle Switch 13-29
13.6.1. Overview 13-29
13.6.2. Configuration 13-29
13.7. Multi-State Switch 13-31

13.7.1. Overview 13-31
13.7.2. Configuration 13-31
13.8. Slider 13-35
13.8.1. Overview 13-35
13.8.2. Configuration 13-35
13.8.3. Combo Setting 13-38
13.9. Numeric 13-42
13.9.1. Overview 13-42
13.9.2. Configuration 13-42
13.10. ASCII 13-63
13.10.1. Overview 13-63
13.10.2. Configuration 13-63
13.11. Indirect Window 13-67
13.11.1. Overview 13-67
13.11.2. Configuration 13-67
13.12. Direct Window 13-75
13.12.1. Overview 13-75
13.12.2. Configuration 13-75
13.13. Moving / Rotating Shape 13-81
13.13.1. Overview 13-81
13.13.2. Configuration 13-81
13.14. Animation 13-87
13.14.1. Overview 13-87
13.14.2. Configuration 13-87
13.15. Bar Graph 13-91
13.15.1. Overview 13-91
13.15.2. Configuration 13-91
13.15.3. Combo Setting 13-97
13.16. Meter Display 13-100
13.16.1. Overview 13-100
13.16.2. Configuration 13-100
13.17. Trend Display 13-108
13.17.1. Overview 13-108
13.17.2. Configuration 13-108
13.18. History Data Display 13-130
13.18.1. Overview 13-130
13.18.2. Configuration 13-130
13.19. Data Block Display 13-139

13.19.1. Overview 13-139
13.19.2. Configuration 13-139
13.20. XY Plot 13-147
13.20.1. Overview 13-147
13.20.2. Configuration 13-147
13.21. Alarm Bar and Alarm Display 13-154
13.21.1. Overview 13-154
13.21.2. Configuration 13-154
13.22. Event Display 13-162
13.22.1. Overview 13-162
13.22.2. Configuration 13-162
13.23. Data Transfer 13-175
13.23.1. Data Transfer (Per-page) 13-175
13.23.2. Data Transfer (Global): Time-based 13-178
13.23.3. Data Transfer (Global): Bit Trigger 13-181
13.24. Backup 13-183
13.24.1. Overview 13-183
13.24.2. Configuration 13-183
13.25. Media Player 13-191
13.25.1. Overview 13-191
13.25.2. Configuration 13-191
13.26. BACnet Schedule 13-197
13.26.1. Overview 13-197
13.26.2. Configuration 13-197
13.27. PLC Control 13-201
13.27.1. Overview 13-201
13.27.2. Configuration 13-201
13.28. Scheduler 13-208
13.28.1. Overview 13-208
13.28.2. Configuration 13-208
13.29. Option List 13-218
13.29.1. Overview 13-218
13.29.2. Configuration 13-218
13.30. Timer 13-226
13.30.1. Overview 13-226
13.30.2. Configuration 13-226
13.31. Video In 13-231
13.31.1. Overview 13-231

13.31.2. Configuration 13-231
13.32. System Message 13-238
13.32.1. Overview 13-238
13.32.2. Configuration 13-238
13.33. Recipe View 13-240
13.33.1. Overview 13-240
13.33.2. Configuration 13-240
13.34. Flow Block 13-248
13.34.1. Overview 13-248
13.34.2. Configuration 13-248
13.35. Operation Log 13-253
13.35.1. Operation Log Settings 13-253
13.35.2. Operation Log View 13-257
13.35.3. Operation Log Printing 13-259
13.36. Combo Button 13-266
13.36.1. Overview 13-266
13.36.2. Configuration 13-266
13.37. Circular Trend Display 13-274
13.37.1. Overview 13-274
13.37.2. Configuration 13-274
13.38. Picture View 13-284
13.38.1. Overview 13-284
13.38.2. Configuration 13-284
13.39. File Browser 13-287
13.39.1. Overview 13-287
13.39.2. Configuration 13-287
13.40. Import/Export 13-291
13.40.1. Overview 13-291
13.40.2. Configuration 13-291
13.41. Pie Chart 13-296
13.41.1. Overview 13-296
13.41.2. Configuration 13-296
13.41.3. Combo Setting 13-298
13.42. Barcode 13-301
13.42.1. 2D Barcode Display 13-301
13.42.2. Push Notification Barcode Display 13-302
13.42.3. Barcode Scanner (Android Camera) 13-305
13.43. String Table 13-311

13.43.1. Overview 13-311
13.43.2. Configuration 13-311
13.44. Database 13-313
13.44.1. Database Server 13-313
13.44.2. SQL Query 13-321
13.44.3. SQL Query Result Viewer 13-329
13.45. Dynamic Scale 13-331
13.45.1. Overview 13-331
13.45.2. Configuration 13-331
13.46. Dynamic Drawing 13-334
13.46.1. Overview 13-334
13.46.2. Configuration 13-334
13.47. PDF Reader 13-340
13.47.1. Overview 13-340
13.47.2. Configuration 13-340
13.48. Table 13-342
13.48.1. Overview 13-342
13.48.2. Configuration 13-342
13.49. VNC Viewer 13-344
13.49.1. Overview 13-344
13.49.2. Configuration 13-344
13.50. Contacts Editor 13-349
13.50.1. Overview 13-349
13.50.2. Configuration 13-349
13.51. Event Bar Chart 13-352
13.51.1. Overview 13-352
13.51.2. Configuration 13-352
13.52. Action Trigger 13-357
13.52.1. Overview 13-357
13.52.2. Configuration 13-357
13.53. Calendar 13-364
13.53.1. Overview 13-364
13.53.2. Configuration 13-364
13.54. Touch Gesture 13-366
13.54.1. Overview 13-366
13.54.2. Configuration 13-366
13.55. PLC Web Browser 13-368
13.55.1. Overview 13-368

13.55.2. Configuration 13-368
13.55.3. Notes on PLC Web Browser 13-370
13.56. Condition 13-372
13.56.1. Overview 13-372
13.56.2. Configuration 13-372
14. Shape Library and Picture Library 14-1
14.1. Overview 14-2
14.2. Building Shape Library 14-2
14.2.1. Shape manager 14-2
14.2.2. Steps to Build Shape Library 14-4
14.3. Building Picture Library 14-9
14.3.1. Picture manager 14-9
14.3.2. Steps to Build Picture Library 14-11
14.3.3. Steps to Import Picture by Pasting 14-13
14.4. Immediate Modification and Preview 14-15
14.4.1. Replacing a Picture 14-16
14.4.2. Replacing a Color 14-16
14.4.3. Extended Features 14-18
14.4.4. Docking a Window 14-19
15. Label Tag Library and Multi-Language 15-1
15.1. Overview 15-2
15.2. Label Tag Library Manager 15-2
15.3. Steps to create Label Tag Library 15-3
15.4. Using Label Tag Library 15-4
15.5. Selecting Language & Font 15-5
15.5.1. Language 15-5
15.5.2. Non-ASCII Fonts 15-6
15.5.3. Font Management 15-8
15.5.4. Font Mapping 15-8
15.6. Settings of Multi-Language 15-9
16. Address Tag Library 16-1
16.1. Overview 16-2
16.2. Building Address Tag Library 16-2
16.3. Using Address Tag Library 16-5
17. Transferring Recipe Data 17-1
17.1. Overview 17-2
17.2. Steps to Update Recipe Data with Ethernet or USB Cable 17-2
17.3. Steps to Update Recipe Data with SD Card or USB Disk 17-3

17.4. Transferring Recipe Data 17-4
17.5. Saving Recipe Data Automatically 17-5
18. Macro Reference 18-1
18.1. Overview 18-2
18.2. Instructions to use the Macro Editor 18-2
18.3. Configuration 18-7
18.4. Syntax 18-8
18.4.1. Constants and Variables 18-8
18.4.2. Operators 18-11
18.5. Statement 18-14
18.5.1. Definition Statement 18-14
18.5.2. Assignment Statement 18-14
18.5.3. Logical Statements 18-14
18.5.4. Selective Statements 18-16
18.5.5. Iterative Statements 18-17
18.6. Function Blocks 18-19
18.7. Built-In Function Block 18-22
18.7.1. Table of Functions 18-22
18.7.2. Device 18-26
18.7.3. Free Protocol 18-36
18.7.4. Process Control 18-41
18.7.5. Data Operation 18-42
18.7.6. Data Type Conversion 18-47
18.7.7. String Operation 18-56
18.7.8. Mathematics 18-79
18.7.9. Statistics 18-85
18.7.10. Recipe Database 18-87
18.7.11. Data/Event Log 18-89
18.7.12. Checksum 18-93
18.7.13. Miscellaneous 18-95
18.8. How to Create and Execute a Macro 18-97
18.8.1. How to Create a Macro 18-97
18.8.2. Execute a Macro 18-101
18.9. User Defined Macro Function 18-101
18.9.1. Import Function Library File 18-102
18.9.2. How to Use Macro Function Library 18-103
18.9.3. Function Library Management Interface 18-105
18.10. Some Notes about Using the Macro 18-114

18.11. Use the Free Protocol to Control a Device 18-114
18.12. Compiler Error Message 18-119
18.13. Sample Macro Code 18-125
18.14. Macro TRACE Function 18-130
18.15. Example of String Operation Functions 18-135
18.16. Macro Password Protection 18-142
18.17. Reading / Writing CAN bus Address Using Variable 18-144
19. Configure HMI as a MODBUS Server 19-1
19.1. Overview 19-2
19.2. Steps to Create a MODBUS Server 19-2
19.3. Steps to Access a MODBUS Server 19-4
19.4. Changing MODBUS Server Station Number Online 19-7
19.5. MODBUS Address Type 19-7
20. How to Connect a Barcode Reader 20-1
20.1. Overview 20-2
20.2. Steps to Connect a Barcode Reader 20-2
21. Ethernet Communication and Multi-HMI Connection 21-1
21.1. Overview 21-2
21.2. HMI to HMI Communication 21-2
21.3. PC to HMI Communication 21-3
21.4. Operating the PLC Connected with Other HMI 21-4
21.4.1. Settings of eMT / iE / XE / mTV / iP Series 21-4
21.4.2. Settings of cMT / cMT X Series models 21-5
22. System Registers 22-1
22.1. Overview 22-3
22.2. The Address Ranges of Local HMI 22-4
22.2.1. Bits 22-4
22.2.2. Words 22-4
22.3. System Registers 22-4
22.3.1. HMI Time 22-5
22.3.2. HMI Operation 22-7
22.3.3. Touch Position 22-8
22.3.4. Local HMI Network Information 22-9
22.3.5. Project File Information 22-11
22.3.6. Storage Space Management 22-12
22.3.7. Recipe and Extended Memory 22-13
22.3.8. Data Sampling 22-16
22.3.9. Event Log 22-18

22.3.10. Station Number Variables 22-20
22.3.11. Index Registers 22-22
22.3.12. MODBUS Server Communication 22-24
22.3.13. Communication Parameter Settings 22-25
22.3.14. Communication Status and Control with Device (COM) 22-29
22.3.15. Communication Status and Control with Device (Ethernet) 22-31
22.3.16. Communication Status and Control with Device (USB) 22-41
22.3.17. Communication Status and Control with Device (CAN Bus) 22-42
22.3.18. Communication Status and Control with Remote HMI 22-43
22.3.19. Communication Status and Control with Remote Device 22-58
22.3.20. Local/Remote Operation Restrictions 22-60
22.3.21. Communication Error Codes 22-61
22.3.22. Driver ID 22-63
22.3.23. DLT645 Controller 22-63
22.3.24. [Device No Response] Window Control 22-63
22.3.25. [Fast Selection] Window Control 22-64
22.3.26. EasyAccess 22-64
22.3.27. EasyAccess 2.0 22-64
22.3.28. Remote Print/Backup Server 22-65
22.3.29. Pass-Through Settings 22-66
22.3.30. VNC Control 22-68
22.3.31. Project Key and HMI Key 22-68
22.3.32. USB Security Key 22-69
22.3.33. User Name and Password 22-69
22.3.34. Macro 22-70
22.3.35. Input Object Function 22-73
22.3.36. Time Sync./Daylight Saving Time 22-73
22.3.37. Cellular Data Network 22-75
22.3.38. WiFi Setting 22-76
22.3.39. OPC UA Server 22-78
22.3.40. e-Mail 22-79
22.3.41. cMT Viewer and Diagnoser 22-80
22.3.42. CODESYS Information 22-83
22.3.43. Miscellaneous 22-84
23. HMI Supported Printers 23-1
23.1. The Supported Printer Types 23-2
23.2. Steps to Add a New Printer and Start Printing 23-5
23.3. Installing Network Printer on cMT / cMT X HMI 23-7

23.4. Installing PPD File on cMT / cMT X HMI 23-10
24. Recipe Editor 24-1
24.1. Overview 24-2
24.2. Recipe / Extended Memory Editor Setting 24-2
24.3. Recipe Database Settings 24-4
24.3.1. Definition 24-4
24.3.2. Transfer 24-6
24.3.3. Data 24-7
24.3.4. System Registers 24-9
25. EasyConverter 25-1
25.1. Overview 25-2
25.2. Converting Data Log File to Excel File 25-2
25.3. Converting Event Log File to Excel File 25-4
25.4. Converting Operation Log File to Excel File 25-6
25.5. Converting Multiple Files 25-7
25.6. Scaling Function 25-8
25.7. Batch File 25-9
25.8. Examination of Historical Data Integrity 25-13
26. EasyPrinter 26-1
26.1. Overview 26-2
26.2. Using EasyPrinter as a Printer Server 26-3
26.2.1. Setup Procedure in EasyPrinter 26-3
26.2.2. Setup Procedure in EasyBuilder Pro 26-4
26.3. Using EasyPrinter as a Backup Sever 26-6
26.3.1. Setup Procedure in EasyPrinter 26-6
26.3.2. Setup Procedure in EasyBuilder Pro 26-7
26.4. EasyPrinter Operation Guide 26-10
26.4.1. EasyPrinter Managing Window 26-10
26.4.2. Operation Guide 26-11
26.5. Convert Batch File 26-15
26.5.1. The Default Value of Convert Batch File 26-16
26.5.2. Specialized Criteria 26-17
26.5.3. The Format of a Convert Batch File 26-17
26.5.4. The Order of Examining Criterion 26-18
27. Simulation in EasySimulator / cMT Viewer 27-1
27.1. EasySimulator 27-2
27.1.1. Overview 27-2
27.1.2. Steps to setup EasySimulator 27-2

27.2. cMT Viewer 27-3
27.2.1. Overview 27-3
27.2.2. Steps to setup cMT Viewer 27-3
28. EasySystemSetting 28-1
28.1. EasySystemSetting 28-2
28.2. Configuring EasySystemSetting 28-5
29. Pass-through 29-1
29.1. Overview 29-2
29.2. Serial Pass Through Ethernet Mode 29-2
29.2.1. Steps to Install Virtual Serial Port Driver 29-2
29.2.2. Steps to Change the Virtual Serial Port Number 29-3
29.2.3. Steps to Uninstall the Virtual Serial Port 29-3
29.2.4. Steps to Update Virtual Serial Port Driver 29-4
29.2.5. Settings of Ethernet mode 29-5
29.3. Serial Pass-Through COM Port Mode 29-7
29.3.1. Settings of COM Port Mode 29-7
29.3.2. Using Utility Manager 29-7
29.3.3. Using System Registers 29-9
29.4. Pass-through Control 29-9
29.5. SIEMENS S7-200 PPI and S7-300 MPI Pass-through Settings 29-10
29.5.1. EasyBuilder Pro Settings 29-10
29.5.2. S7-200 PPI Connection 29-10
29.5.3. S7-300 MPI Connection 29-11
29.5.4. Registers of SIEMENS Pass-through 29-14
29.6. Ethernet Pass-Through 29-15
30. Project Protection 30-1
30.1. Overview 30-2
30.2. EXOB/CXOB Password 30-2
30.3. Decompilation is Prohibited 30-3
30.4. Disable HMI Upload Function 30-3
30.5. Project Key 30-3
30.6. MTP File Password 30-4
31. cMT Diagnoser 31-1
31.1. Overview 31-2
31.2. How to Start Diagnoser 31-2
31.3. cMT Diagnoser User Interface 31-4
31.3.1. Object Settings 31-4
31.3.2. Device Settings 31-7

31.3.3. Packet Settings 31-7
31.3.4. Macro Settings 31-9
31.3.5. MQTT Settings 31-16
31.3.6. OPC UA Settings 31-18
31.3.7. JS Settings 31-19
31.4. Additional Notes 31-19
32. FTP Server Application 32-1
32.1. Overview 32-2
32.2. Steps to Log in FTP Server 32-2
32.3. Backup History Data and Update Recipe Data 32-3
33. EasyDiagnoser 33-1
33.1. Overview 33-2
33.2. Configuration 33-2
33.3. EasyDiagnoser Settings 33-3
33.3.1. Main Menu 33-3
33.3.2. Activity Area 33-4
33.3.3. Polling Packages 33-6
33.3.4. Devices 33-8
33.3.5. Output (Macro debug) 33-8
33.4. Error Code 33-9
33.5. Window Adjustment 33-10
34. Control Token (cMT / cMT X) 34-1
34.1. Overview 34-2
34.2. Configuration 34-2
34.3. Demonstration 34-4
35. EasyWatch 35-1
35.1. Overview 35-2
35.2. Configuration 35-2
35.2.1. Basic Functions 35-2
35.2.2. Quick Selection Tools 35-3
35.3. Monitor Settings 35-4
35.3.1. Adding a Monitor 35-4
35.3.2. Monitor Settings 35-4
35.3.3. Steps to Add a New Monitor 35-5
35.4. Macro Settings 35-8
35.4.1. Adding a Macro 35-8
35.4.2. Macro Settings 35-8
35.4.3. Steps to Add a New Macro 35-8

35.5. HMI Manager 35-9
35.5.1. Opening HMI Setting 35-9
35.5.2. HMI Manager 35-10
35.6. Object List 35-11
35.6.1. Object List Columns 35-11
35.6.2. Editing Object List 35-12
36. Administrator Tools 36-1
36.1. Overview 36-2
36.2. User Accounts 36-2
36.2.1. User Accounts Settings 36-2
36.2.2. Steps to Set User Accounts 36-4
36.2.3. Steps to Import Accounts Using EasyBuilder Pro 36-5
36.3. USB Security Key 36-6
36.3.1. USB Security Key Settings 36-6
36.3.2. Steps to Set USB Security Key 36-7
36.3.3. Steps to Set USB Security Key Using EasyBuilder Pro 36-8
36.4. e-Mail SMTP Server Settings 36-9
36.4.1. Steps to set e-Mail SMTP Server Settings 36-9
36.5. e-Mail Contacts 36-10
36.5.1. e-Mail Contacts Settings 36-10
36.5.2. Steps to set e-Mail Contacts 36-11
36.5.3. Steps to Import e-Mail Settings and Contacts Using EasyBuilder Pro 36-13
37. MODBUS Gateway 37-1
37.1. Overview 37-2
37.2. Steps to Create an Address Mapping Table 37-2
37.3. Notes about Configuring Address Mapping 37-5
38. EasyDownload 38-1
38.1. Overview 38-2
38.2. Configuration 38-2
39. Data Security 39-1
39.1. Overview 39-2
39.2. Configuration 39-2
39.2.1. Word address settings 39-3
39.2.2. Bit address settings 39-5
40. Web Streaming 40-1
40.1. Overview 40-2
40.2. Configuration 40-2
40.3. Operation 40-2

40.4. Applicable Models 40-3
41. Energy 41-1
41.1. Energy Demand Setting 41-2
41.1.1. Overview 41-2
41.1.2. Configuration 41-2
41.2. Energy Demand Display 41-6
41.2.1. Overview 41-6
41.2.2. Configuration 41-6
42. IIoT 42-1
42.1. MQTT 42-2
42.1.1. Overview 42-2
42.1.2. Configuration 42-2
42.2. OPC UA Server 42-28
42.2.1. Overview 42-28
42.2.2. Configuration 42-28
42.2.3. Device Statistics 42-36
42.2.4. Limitation 42-37
43. JS Object and JS Library 43-1
43.1. Overview 43-2
43.1.1. Usage Suggestions 43-2
43.1.2. Hardware and Software Requirements 43-2
43.1.3. Warning 43-2
43.1.4. JS Object SDK 43-2
43.2. JS Object 43-3
43.3. JS Resource 43-5
43.3.1. Temporary Folder 43-5
43.4. Demonstrations 43-7
43.4.1. Mimic a [Toggle Switch] object 43-7
43.4.2. Examples of JS Resource Usage 43-16
43.5. Limitations 43-20
43.6. How a JS object operates behind the scenes 43-20
Appendix A. Comparison of HMI Software Features A-1
Appendix B. cMT Viewer Limitations B-1
Appendix C. PC Simulation Limitations C-1
Appendix D. Notes on the Use of 64-bit Data Type D-1

Windows is a trademark or a registered trademark of Microsoft Corporation in the United States and/or other countries.
Android is a trademark or a registered trademark of Google, Inc. in the United States and/or other countries.
IOS is a registered trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.
Other company names, product names, or trademarks in this document are the trademarks or registered trademarks of their respective companies.
This document is subject to change without prior notice.

1. EasyBuilder Pro Installation and Startup Guide

This chapter explains how to install EasyBuilder Pro.

1.1. Installation Requirements 1-2
1.2. Steps to Install EasyBuilder Pro 1-3
1.3. Software Update Reminder 1-7

1.1. Installation Requirements

Software:

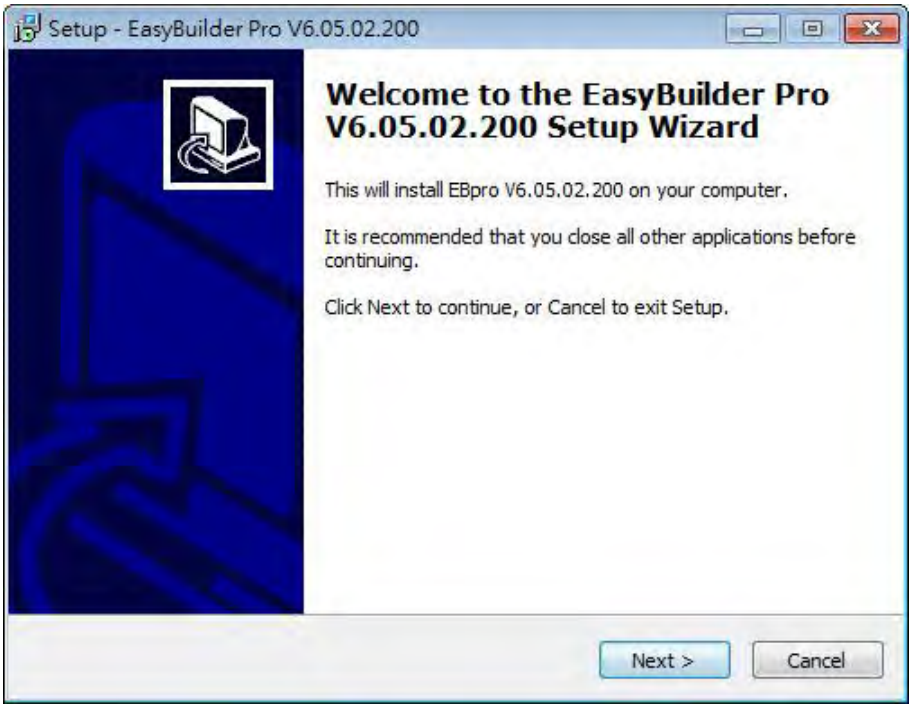
Visit Weintek Labs, Inc.'s website at http://www.weintek.com and download the installation package which include the following languages: Simplified Chinese, Traditional Chinese, English, Japanese, German, Italian, Korean, Spanish, Russian, French, Polish, and Turkish. The latest updates can be downloaded too.

Operating System:

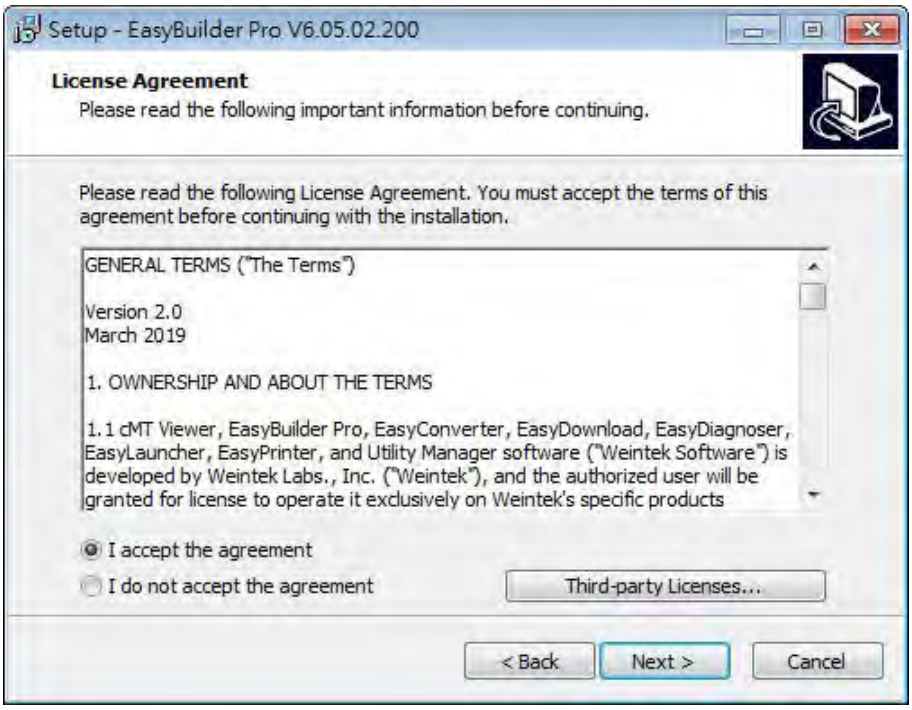
Windows® 7 SP1 (32bit / 64bit)
Windows® 8 (32bit / 64bit)
Windows® 8.1 (32bit / 64bit)
Windows® 10 (32bit / 64bit)

1.2. Steps to Install EasyBuilder Pro

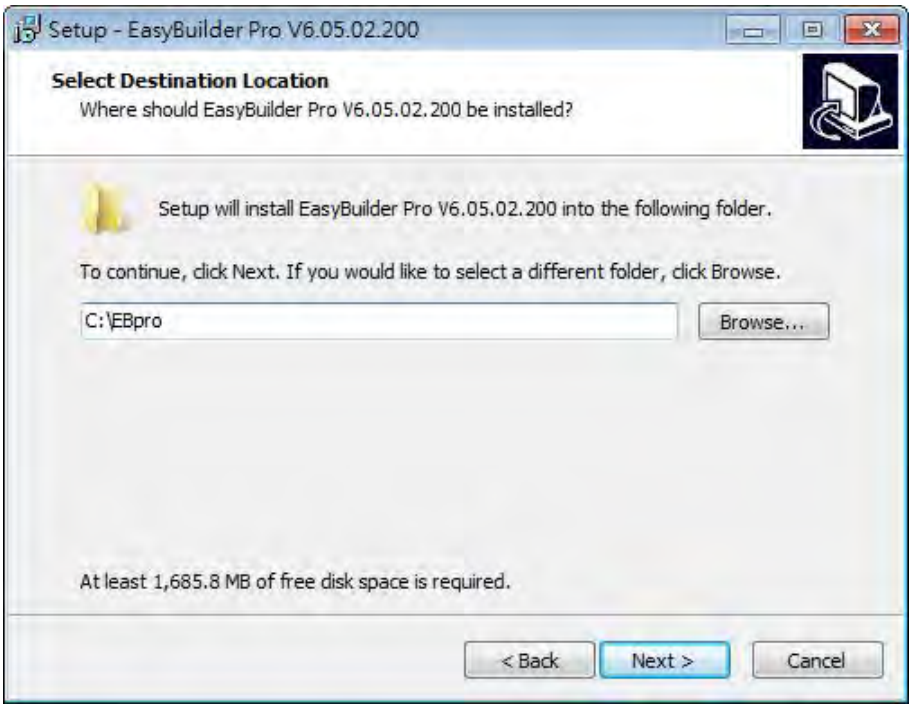
1. Click [Install] and select the language for the installation process, and then click [Next].



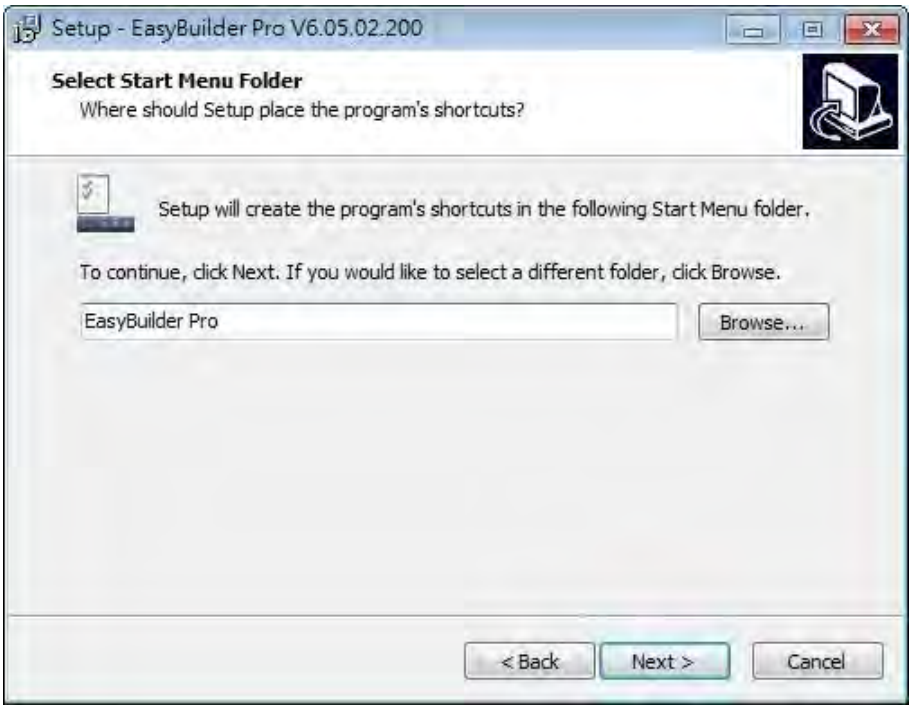
2. If there is a previous version of EasyBuilder Pro on the PC, please remove it before installation.
3. Please read and accept the License Agreement before continuing.



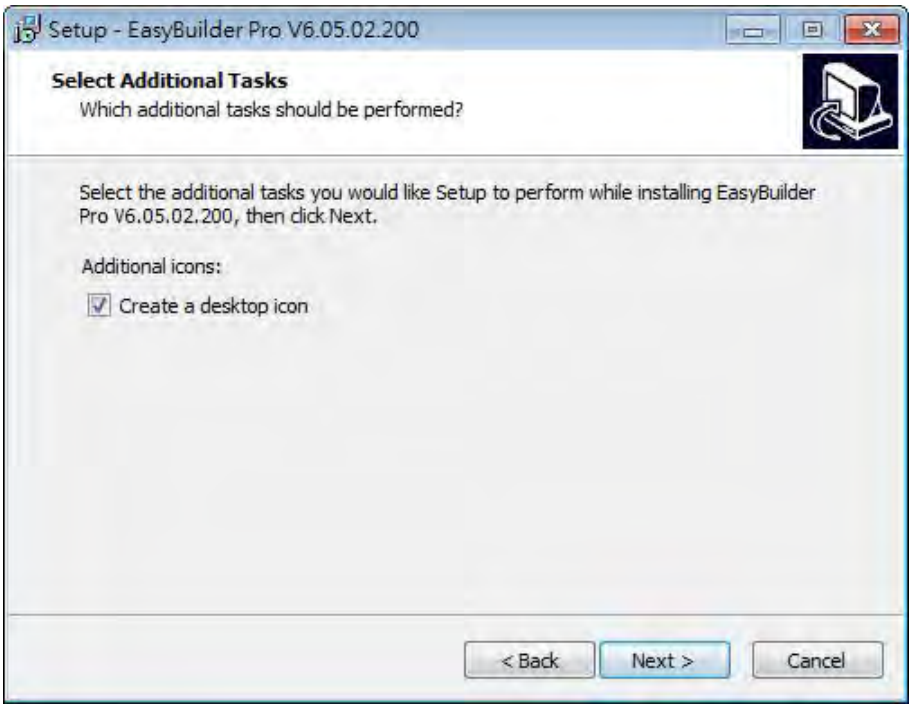
4. Select a folder for EasyBuilder Pro installation, or use the default folder. Click [Next].



5. Select a start menu folder, or use the default folder. Click [Next].

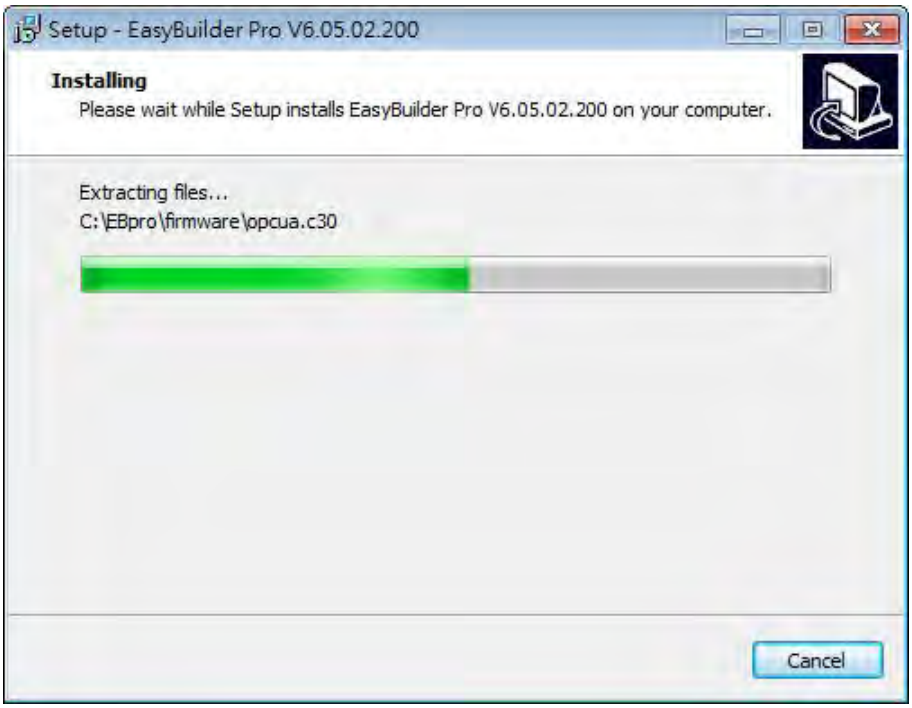


6. Select additional tasks, for example: [Create a desktop icon]. Click [Next].



7. Confirm all setting. To change the setting, click [Back]. To start installation, click [Install].

8. Installation in progress.



9. Click [Finish] to complete the installation.

10. The EasyBuilder shortcut can be found in [Start] » [All Programs] » [EasyBuilder Pro].

The description of each item in EasyBuilder Pro menu:

Installed file	Description
Administrator Tools	Saves User Accounts, USB Security Key, e-Mail SMTP Server Setting, e-Mail Contacts to an external device (USB drive or SD card) so that the data become

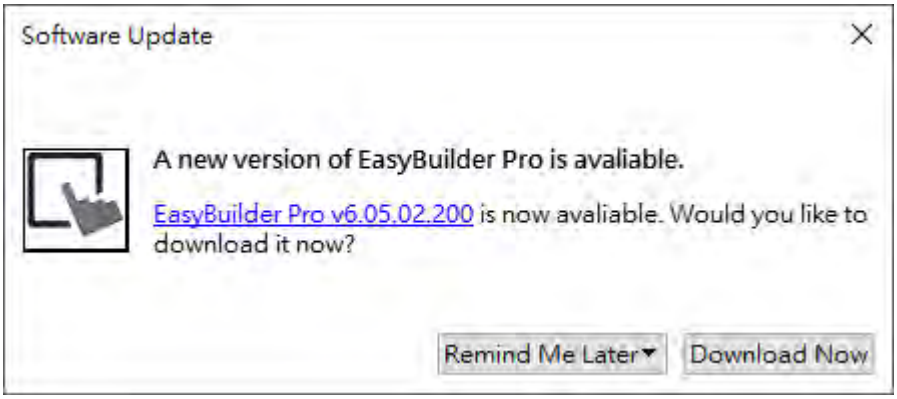
	portable and can be imported to HMI.
cMT Viewer	Supports access to cMT / cMT X Series HMI via network, for operating cMT / cMT X Series HMI on PC.
Easy Access	Supports access to remote HMI by connecting EasyAccess server over Internet.
EasyBuilder Pro	EasyBuilder Pro project editor.
EasyConverter	Conversion tool for Data Sampling and Event Log.
EasyDiagnoser	Monitoring and debugging tool operated on HMI.
EasyPrinter	Allows exporting data to a remote server on PC via Ethernet, realizing remote printing / backup.
EasySimulator	Executes simulation.
EasySystemSetting	Allows updating hardware system settings by using SD card or USB drive.
EasyWatch	An easy to use debugging and remote monitoring tool running on PC that allows monitoring or setting HMI and PLC addresses and executing macro.
Recipe Editor	Tool for setting the format of Recipe data. Users can open Recipe data or the data in the External Memory.
Release Note	Software release notes.
UtilityManagerEx	EasyBuilder Pro management tool.

Note

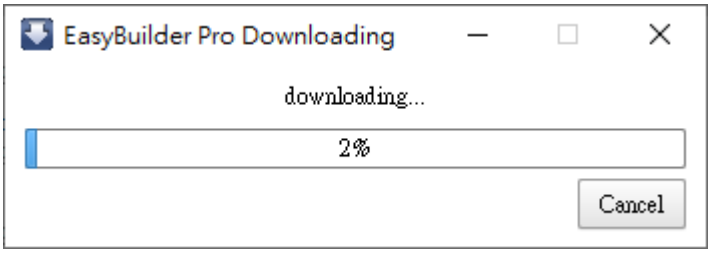
- Certain HMI supports downloading/uploading projects via USB cable. After installing EasyBuilder Pro, the USB driver will be automatically installed. A message saying “Windows can’t verify the publisher of this driver” may show, please continue anyway. After installing the USB driver, open [Computer Management] » [Device Manager] to check if installation succeeded.

1.3. Software Update Reminder

After launching EasyBuilder Pro, the system will check whether the latest version has been installed, and show a reminder dialog if there is a newer version.



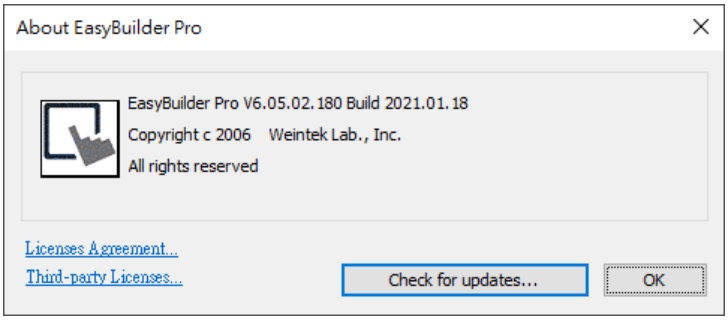
After clicking [Download Now], the latest version of EasyBuilder Pro will be downloaded to a temporary folder, and a progress bar is shown as below. After download, please follow the steps explained in the previous section to install EasyBuilder Pro.



Note

- Please note that the Software Update reminder cannot be turned off permanently. It may be rescheduled to one day or one week later.

Users can open File » About in EasyBuilder Pro to see current version and check for updates.



2. Utility Manager

This chapter explains how to use Utility Manager.

2.1. Overview 2-2
2.2. Design 2-3
2.3. Analysis & Testing 2-4
2.4. Publish 2-6
2.5. Maintenance 2-10
2.6. Data Conversion 2-12

2.1. Overview

After installing EasyBuilder Pro, double click [UtilityManagerEx] shortcut on the desktop to start. Utility Manager is for launching several utilities and it is a stand-alone program. Please select a model first and note that certain features may not work as expected if an incorrect model is selected.

Utilities	Description
	Minimize Utility Manager window.
	Close Utility Manager.
	Add the frequently used utilities to the toolbar at the bottom of the window.
Run	Run the selected utility on the toolbar.
Edit	Delete the selected utility on the toolbar.

2.2. Design

Utilities	Description
EasyBuilder Pro	Launch EasyBuilder Pro to edit projects.
EasyAddressViewer	Review the address ranges and formats of supported PLCs.
Simulation	Simulate an HMI on PC by running On-line Simulation or Off-line Simulation. Please select an .exob / .cxob file as source first. **Off-line simulation:** Simulate project operation on PC without connecting any devices. **On-line simulation:** Simulate project operation on PC and PLCs are directly connected with PC. **Note** ■ When using [On-line simulation] on PC, if the target device is a local PLC (the PLC directly connected to PC), there is a 10 minutes simulation limit.

2.3. Analysis & Testing

Utilities	Description
cMT Diagnoser	A diagnostic tool that can monitor data and status of all objects in the current window when it is connected to a cMT Series HMI, show the status of individual communication packet, show communication status with MQTT server, and debug macros. See “31 cMT Diagnoser”.
EasyDiagnoser	Diagnoses the connection status between eMT/iE/XE/mTV HMI and PLC. See “33 EasyDiagnoser”.
EasyWatch	Allows users to monitor or configure HMI or PLC address values via Ethernet on PC. See “35 EasyWatch”.
Reboot HMI	Reboots the HMI without unplugging. After reboot, the system returns to the initial state.
Serial Pass-Through	Opens a local serial port (virtual/physical) and allow other PC

	applications to connect to PLC via HMI. The Serial Pass-Through feature provides two modes: Ethernet and COM port. Before using [Ethernet] mode, please check if virtual serial port driver has been installed. See “29 Pass-through”.
Ethernet Pass-Through	When PC and PLC are respectively connected to different Ethernet ports of a Dual-port HMI, this feature allows running the application on PC to control PLC through HMI. This feature is only supported on cMT / cMT X Series models. See “29 Pass-through”.

Pass-through feature allows the PC application to connect PLC via HMI. In this case, the HMI works like a converter.

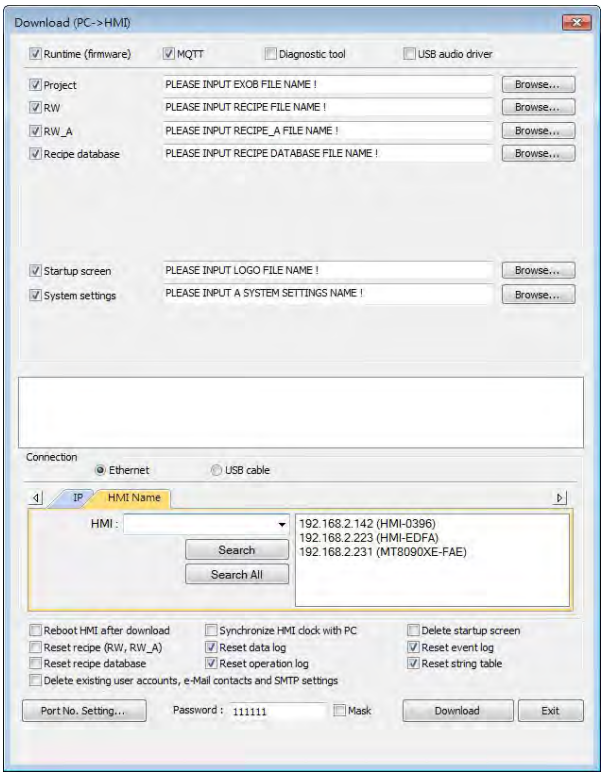
2.4. Publish

Utilities	Description
Download	Downloads project file to HMI via Ethernet or USB cable.
Upload	Uploads files on HMI to PC via Ethernet or USB cable.
Build Download Data for SD/USB Disk	Builds the data to be downloaded to HMI in an SD card / USB drive. For cMT / cMT X Series, the data needs to be built in EasyBuilder Pro.

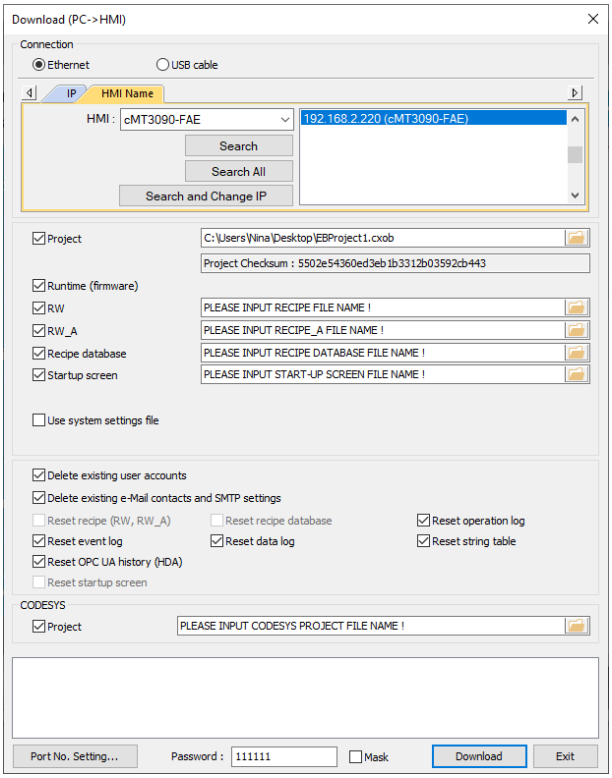
2.4.1. Download (PC->HMI)

Download files to HMI via Ethernet or USB cable.

eMT, iE, XE, mTV Series



cMT, cMT X Series



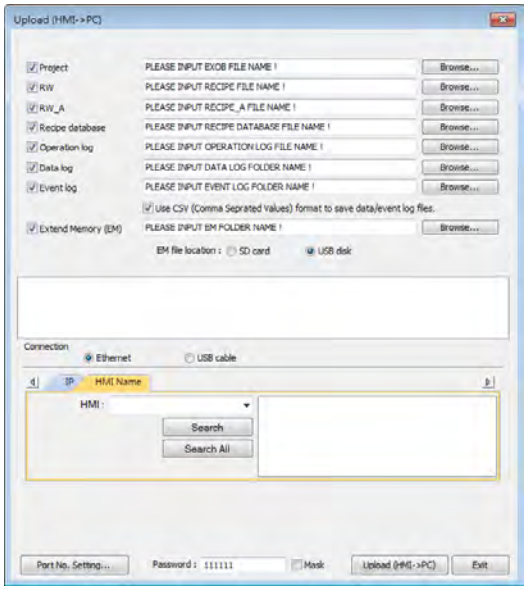
Setting	Description
Project	Select an .exob or .cxob project file. For a cMT / cMT X project, the Project Checksum will be shown under the directory of the project for users to check whether the project file has been tempered with.
Runtime (firmware)	Update HMI kernel programs. The firmware must be downloaded at the first time downloading data to HMI.
MQTT	If MQTT is used in the project and HMI is the broker, this checkbox must be selected before download.
Diagnostic tool	Enable the diagnostic tool to send diagnostic data to the developers when an error occurs for quick troubleshooting.
USB Audio Driver	Enable audio output on HMI so it can immediately emit a sound when a USB sound

	card and a speaker are connected.
RW / RW_A	Select an .rcp recipe file.
Recipe database	Select a .db file of Recipe Database.
Startup screen	Download a .bmp file to HMI. After HMI is rebooted, this .bmp file will be shown before project starts.
Use system settings file	Select a .conf system settings file.
Reboot HMI after download	Automatically reboot after download.
Port No. Setting	Select the port by which to download the project file via Ethernet.
Synchronize HMI clock with PC	Synchronize HMI time with PC time when downloading project file.
Reset recipe / recipe database / event log / data log / operation log / string table / OPC UA history (HDA) / startup screen / Delete existing user accounts, e-Mail contacts and SMTP settings	Erase the selected files on HMI before download.
CODESYS	Use Utility Manager to download CODESYS boot project.

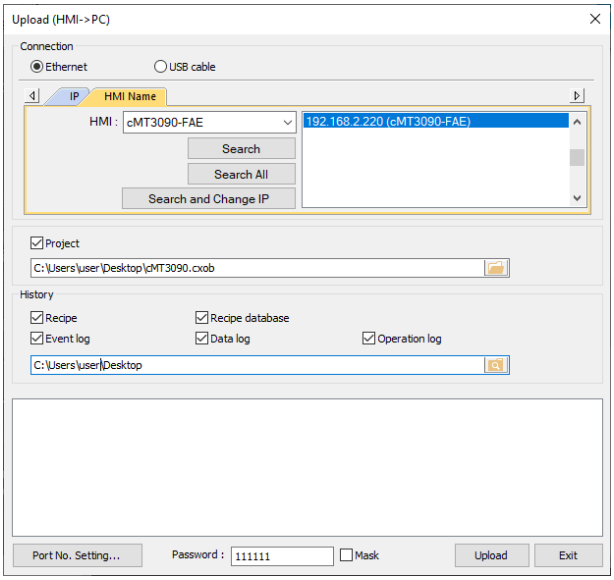
2.4.2. Upload (HMI->PC)

Upload files from HMI to PC via Ethernet or USB cable. Click [Browse] and assign the file path before uploading.

eMT, iE, XE, mTV Series



cMT, cMT X Series



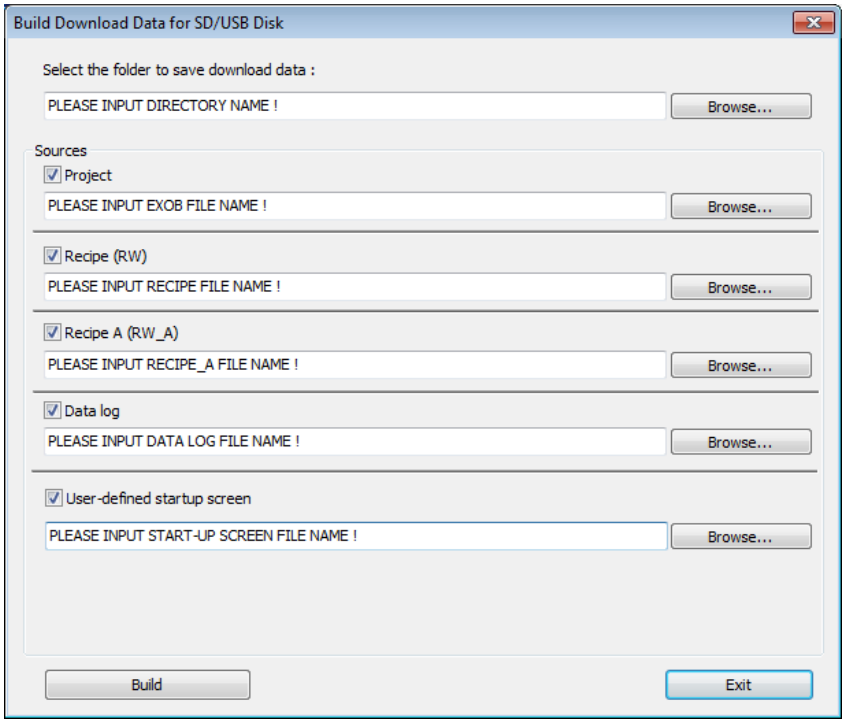
Setting	Description
Event log	Upload the .evt file on HMI to PC.
Extended Memory (EM)	Upload the .emi file saved in SD card or USB disk to PC.

For information about [Project], [RW / RW_A], [Recipe database] or [Data log], see “2.4.1 Download” in this chapter.

Note

- The file will be uploaded to PC in .exob / .cxob format. Please decompile it into editable .emtp / .cmtp file first and open the .emtp / .cmtp file in EasyBuilder Pro.
- To upload the historical files saved in the external device, please use FTP. See “32 FTP Server Application” for more information.

2.4.3. Build Download Data for SD/USB Disk



1. Insert an external device (SD card or USB drive) to PC.
2. Designate the directory to save data.
3. Browse for the directory of the source files.
4. Click [Build] to create download file.

Files will then be saved to the inserted device for users to download to HMI without having to use a USB cable or Ethernet.

2.4.4. Steps to Download Project to HMI via USB Disk or SD Card

To download data in a folder (e.g. a folder named “123” (K:\123)) in a USB disk.

1. Insert USB (in which the project is saved) to HMI.
2. In [Download / Upload] dialog box select [Download].
3. Enter Download Password.
4. In [Download Settings] dialog box, select [Download project files] and [Download history files].
5. Press [OK].
6. In [Pick a Directory] dialog box, select directory: usbdisk\disk_a_1\123.
7. Press [OK].

Project will then be updated.

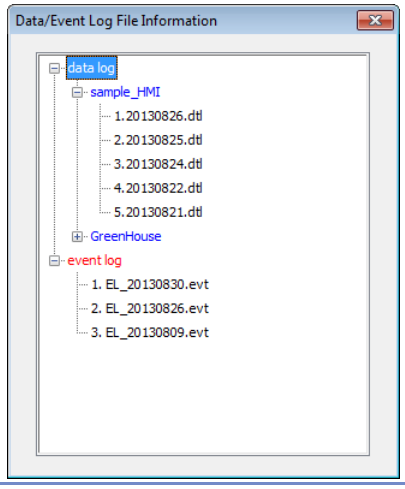
Note

- If history data is downloaded but not the project file, please reboot HMI to update files.

2.5. Maintenance

Options in this tab may vary according to model.

Utilities	Description
Administrator Tools	Allows saving the data of [User Accounts], [USB Security Key],

	[e-Mail SMTP Server Settings], and [e-Mail Contacts] to USB. See “36 Administrator Tools”.
cMT Viewer	cMT Viewer can connect to a cMT / cMT X Series model. PC is the display terminal for cMT / cMT X project.
cMT-iV5/cMT-iV6 Firmware Update	Updates cMT-iV5/iV6 runtime (firmware) or OS.
cMT-Server OS Update	Updates cMT-SVR/SVR-200 OS.
cMT-Gateway/CTRL OS Update	Updates cMT-Gateway/CTRL OS.
CODESYS Firmware Upgrade	Updates CODESYS firmware on cMT HMI.
SSL/TLS Certificate Upgrade	Upgrades the built-in expired SSL/TLS certificate on HMI.
HMI Information Collector	Obtains a file with diagnostic information from HMI.
EasyPrinter / Backup Server	Runs remote backup/printer server on PC.
Data/Event log Information	Connects with HMI via USB cable or Ethernet to check the number of history files on HMI. This feature is not supported on cMT / cMT X Series. 
EasyAccess 1.0	Supports access to remote HMI by connecting EasyAccess server over Internet. You can find more information at: www.ihmi.net

Note

- HMI Information Collector is available only on cMT / cMT X Series models, and the applicable OS version is 20210705 or later.

2.6. Data Conversion

Utilities	Description
Recipe Database Editor	Edits recipe data. Click the icon to download the document about Recipe Database.
Easy Converter	Reads HMI-generated data logs, event logs, and operation logs and converts the files to Excel (.xlsx) format. See “25 EasyConverter”.
Recipe Editor	Creates new recipe data to be used on HMI and modifies the existing recipe data. See “24 Recipe Editor”.

3. Create an EasyBuilder Pro Project

This chapter explains the basic steps to create an EasyBuilder Pro project.

3.1. Overview 3-2
3.2. Create a New Project File 3-2
3.3. Save and Compile the Project File 3-4
3.4. Run On-Line or Off-Line Simulation 3-5
3.5. cMT Viewer 3-5
3.6. Download the Project File to HMI 3-6
3.7. Upload the Project File from HMI 3-13

3.1. Overview

The following is the process of creating a project.

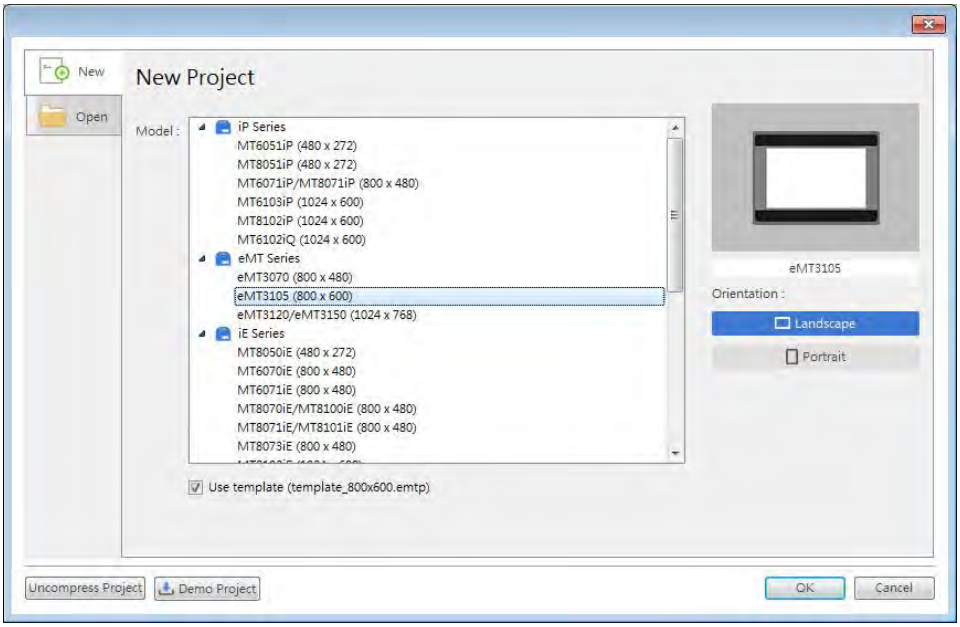
1. Create a new project file.
2. Save and compile the project file.
3. Run On-line or Off-line simulation.
4. Download the project file to HMI.

The following describes each process.

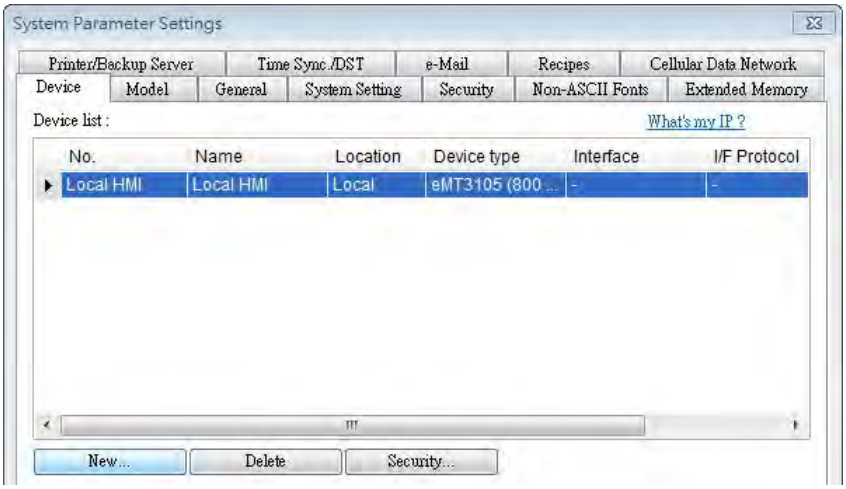
Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

3.2. Create a New Project File

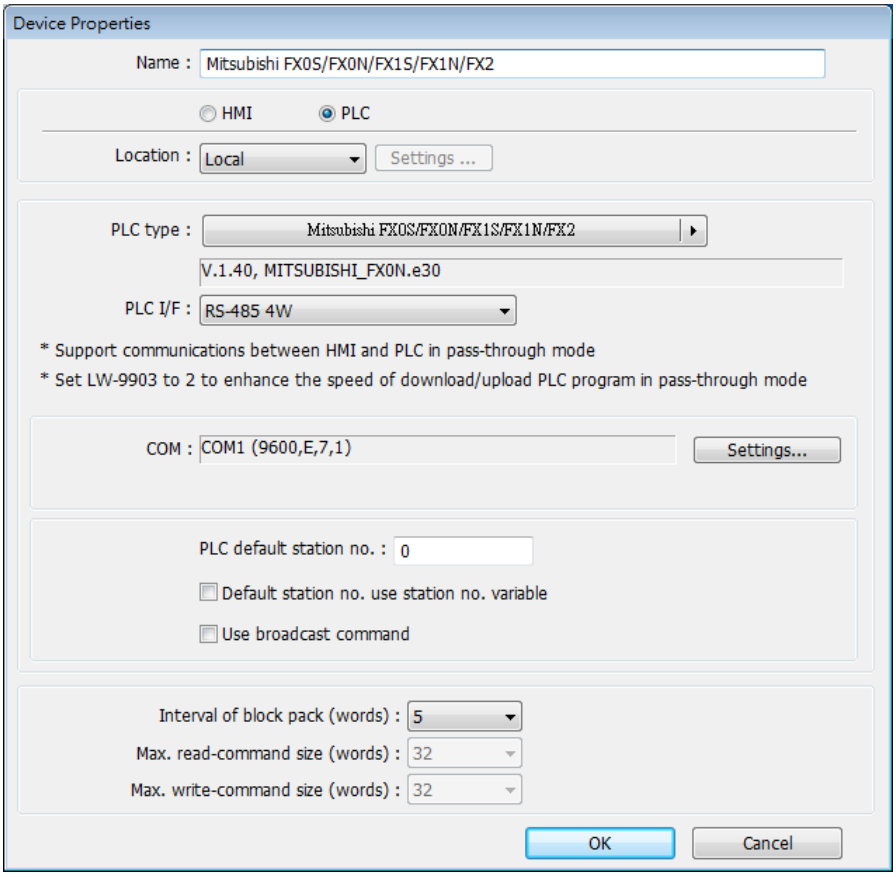
1. Launch EasyBuilder Pro and open a new project file.
2. Select a model and select [Use template] check box.



3. Click [New] button in the Device tab and select a device.

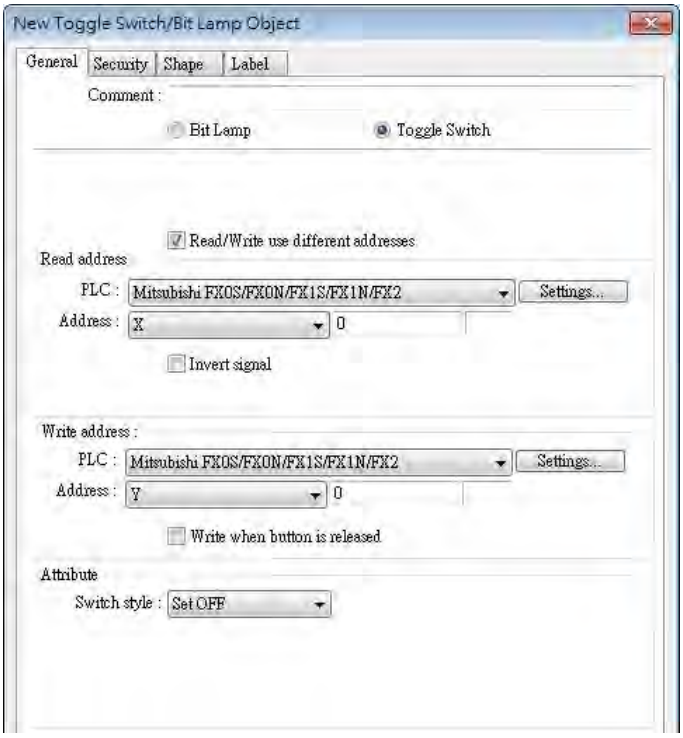


4. Configure parameters.

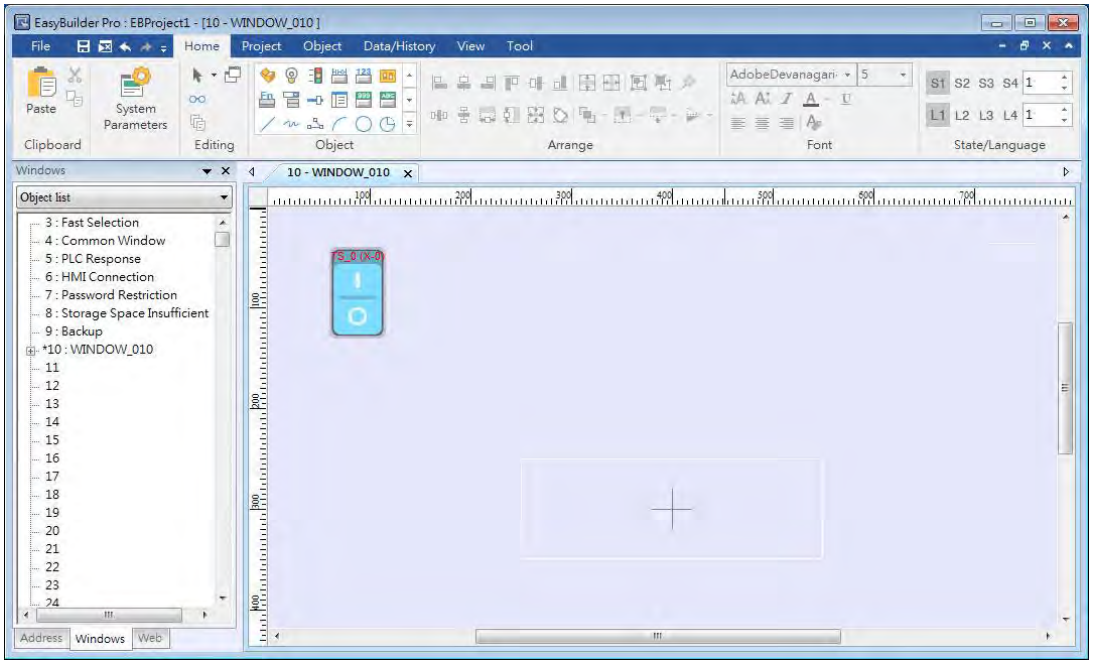


5. A new device is added to the [Device List].

6. Create an object, for example, a Toggle Switch, and then set the address.



7. Place the object in the edit window. A project is now created.



3.3. Save and Compile the Project File

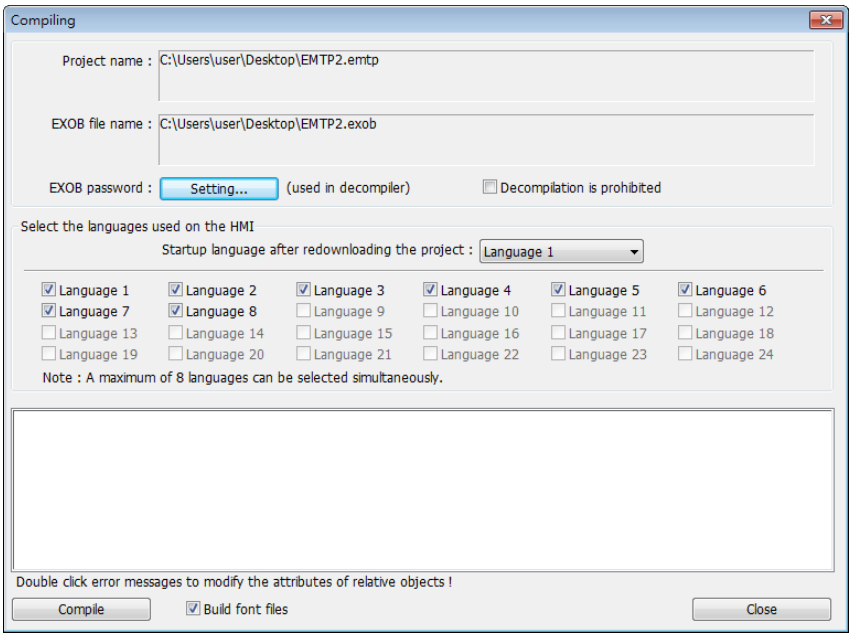
1. In EasyBuilder Pro toolbar click [File] » [Save] to save the .emtp file.

2. In EasyBuilder Pro toolbar click [Project] » [Compile] to compile .emtp file as .exob file, which could be downloaded to HMI. This also checks if the project can run correctly.

Note

- For cMT / cMT X Series models, the project file extension is .cmtp, and the compiled file extension is .cxob.

3. To use multiple languages, all languages must be configured in Label Tag Library first. When downloading the project to HMI, select the needed languages only. A successful compilation is shown in the following figure.



3.4. Run On-Line or Off-Line Simulation

Off-line simulation: Simulate project operation on PC without the need for connecting any device.

On-line simulation: Simulate project operation on PC without downloading the project to HMI. The PLC is connected to PC, please set correct parameters.

Note

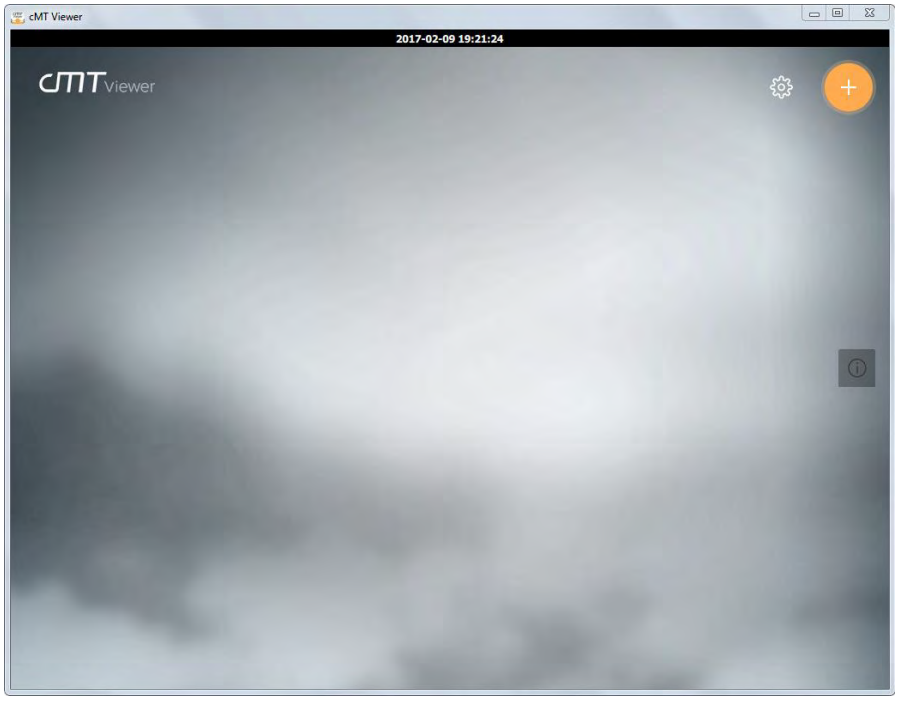
- When using On-line Simulation on PC, if the target device is the PLC directly connected to PC, there is a 10-minute simulation limit.

3.5. cMT Viewer

This program connects to cMT Series models via network. To run this program, execute

cMTViewer.exe under installation directory. Or, in EasyBuilder Pro toolbar, click [Tools] » [cMT Viewer].

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.



3.6. Download the Project File to HMI

The following explains four ways to download the project file to HMI.

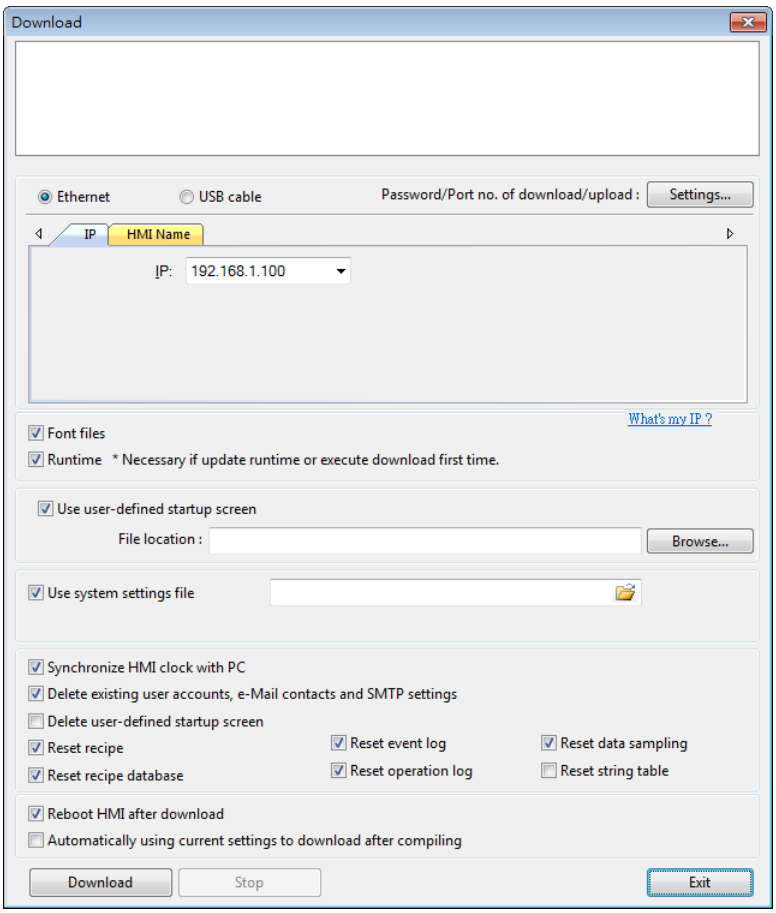
Note

- For cMT-SVR, only the way described in 3.6.1 is available.
- When download project file to HMI using mini USB cable, please do not connect PLC simultaneously, in order to avoid noise from PLC interfering with HMI.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

3.6.1. Configure in EasyBuilder Pro

1. In EasyBuilder Pro toolbar, click [Project] » [Download]. Make sure that all the settings are correct.
2. Select [Ethernet], set password and HMI IP.



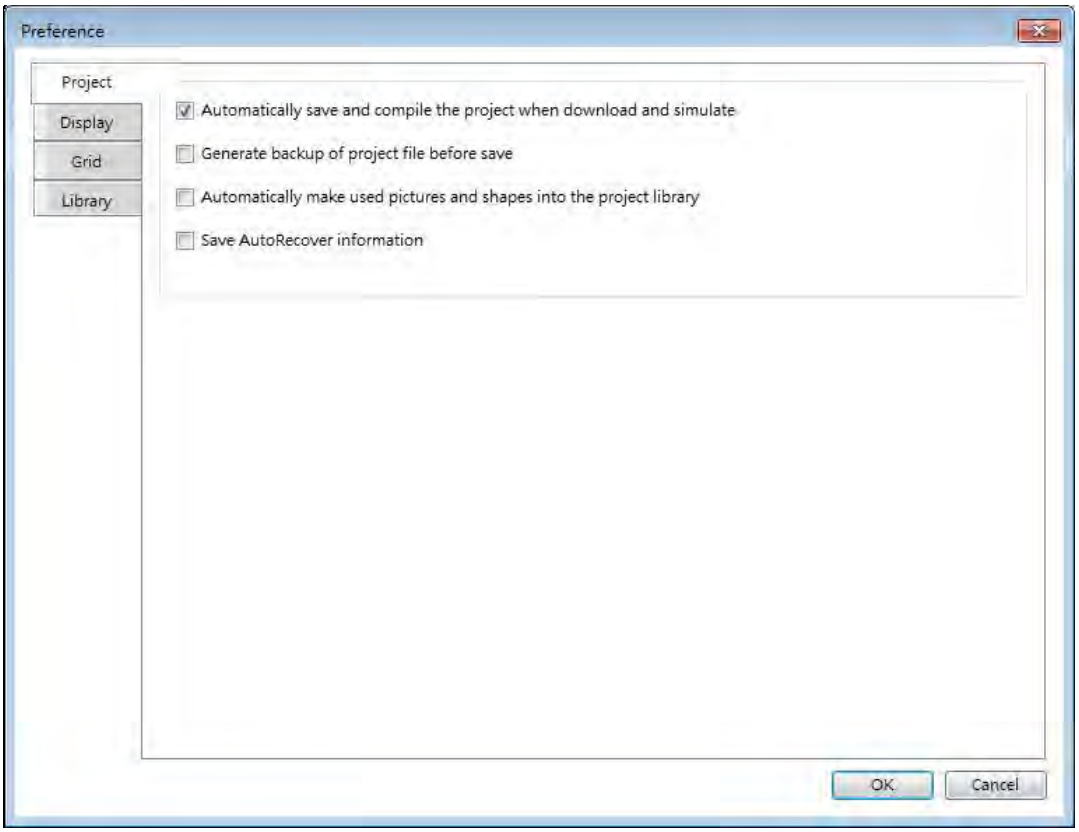
Setting	Description
Font files	Download the font used in the project to HMI.
Runtime	Select the check box to update the HMI kernel programs. If this is the first time downloading file or EasyBuilder Pro version is updated, please download the firmware before downloading files to HMI.
EasyAccess 2.0	Download EasyAccess 2.0 driver to HMI. (This option is available only for MT8000iE models. For other models, EasyAccess 2.0 will be automatically downloaded.)
Use user-defined startup screen	Download the selected .bmp picture to HMI as the startup screen.
Use system settings file	Download the system settings file to update hardware settings.
Synchronize HMI clock with PC	Synchronize HMI time with PC time when downloading project file.
Delete existing user accounts, e-Mail contacts and SMTP settings	Select the check box to delete existing user accounts, e-Mail contacts or SMTP settings before downloading the project. This setting is only

	effective when one of the following settings is enabled: 1. [System Parameters] » [Security] » [Enhanced security mode] » [Use existing user accounts on HMI first (if existed), Othrewise, use settings below]. 2. [System Parameter s] » [e-Mail] » [Use existing user accounts on HMI first (if existed), Othrewise, use settings below].
Reset recipe/ event log/ data sampling/ recipe database/ operation log/ string table/ user-defined startup screen	The selected files will be erased before downloading.
Reboot HMI after download	HMI will reboot after the downloading process is done.
Automatically using current settings to download after compiling	The system will compile the project and download it to the latest target HMI. The way to enable this function is described in the following part.

[Automatically using current settings to download after compiling]

If this check box is selected, EasyBuilder Pro will compile the project and download it to the latest target HMI.

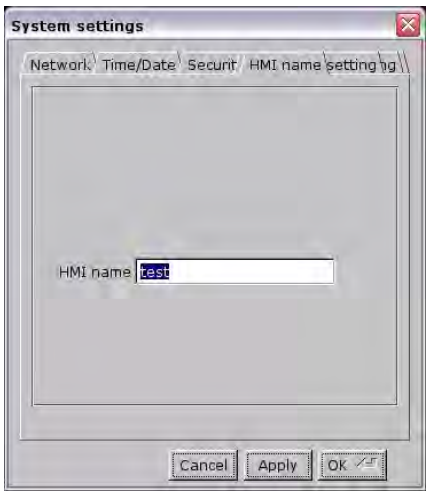
1. In EasyBuilder Pro toolbar, click [File] » [Preferences].
2. Select [Automatic save and compile when download and simulate] check box.



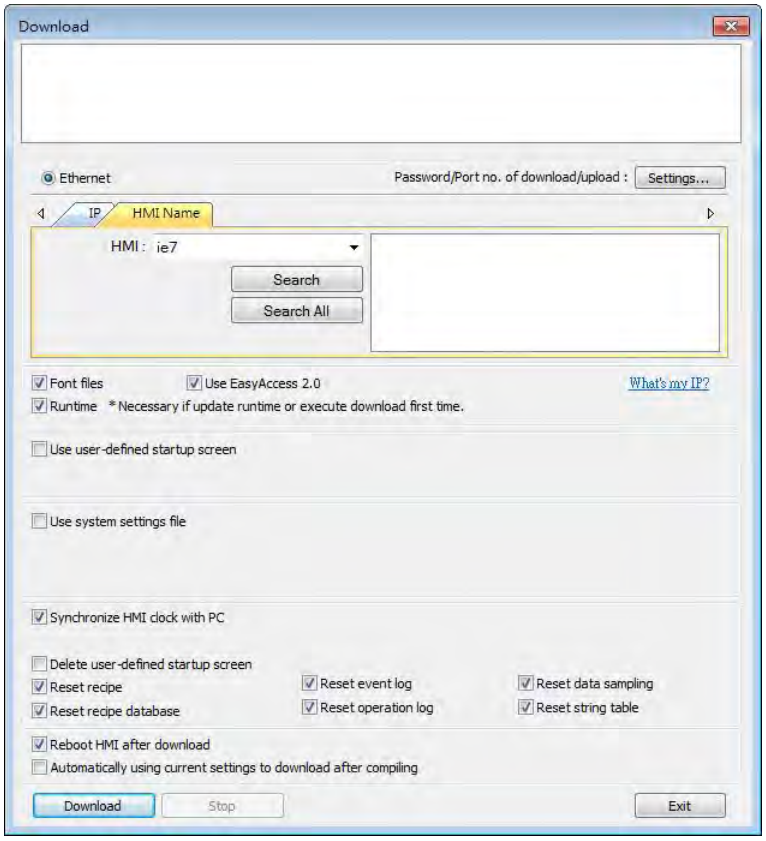
3. In EasyBuilder Pro toolbar, click [Save] and then [Download].
4. In the dialog box, select [Automatically using current settings to download after compiling] check box.
5. Click [Download].
6. When finished, next time when [Download] is clicked, EasyBuilder Pro will automatically compile and download the project to the latest target HMI.

3.6.2. Use HMI Name

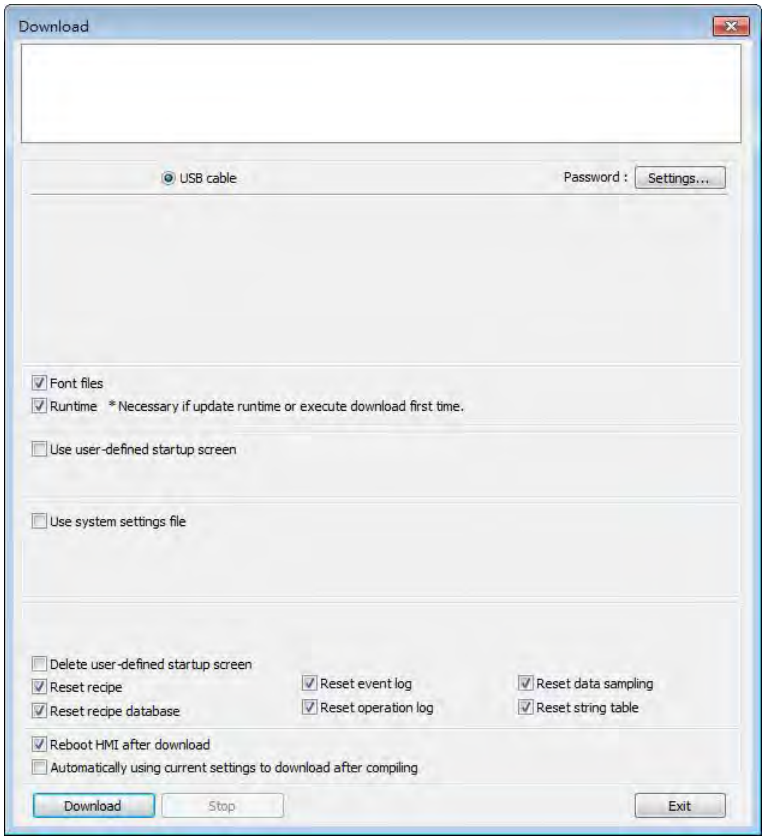
1. Go to [System settings] on HMI and then set HMI name first.



2. On PC, select the HMI name and start downloading. To use [Search], enter the HMI name first to search for the HMI. [Search all] searches for all HMIs in the same subnet network.



3.6.3. Use USB cable

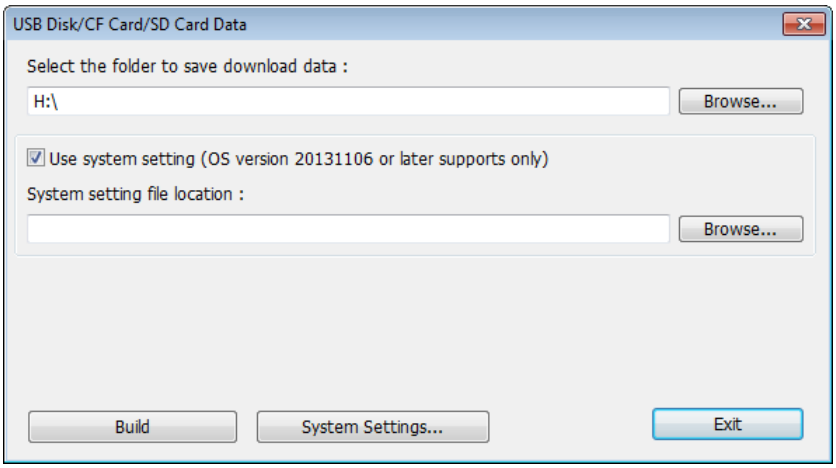


Select USB cable to download project to HMI. The way of setting is same as "3.6.1 Configure in EasyBuilder Pro". Before downloading via USB cable, please make sure the USB driver is installed. Go to [Computer Management] » [Device Manager] to check if USB driver is installed. If it is not installed, please refer to installation steps to install manually.

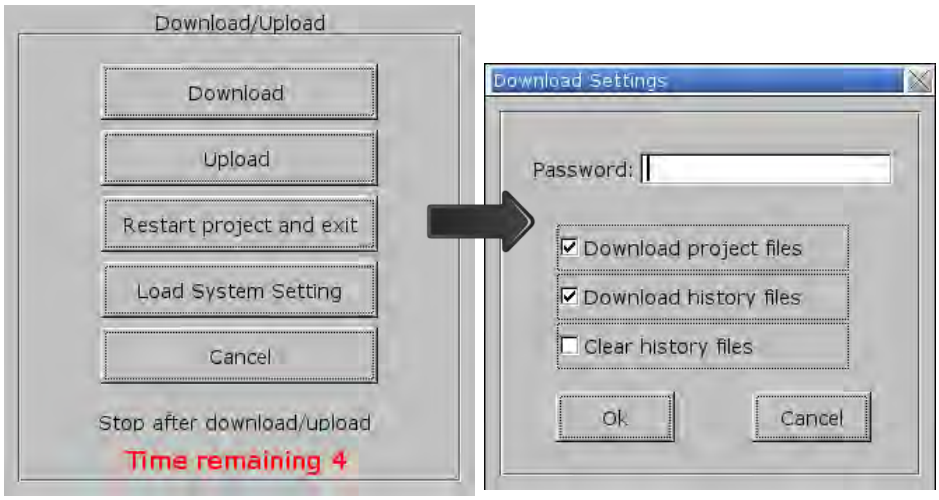
3.6.4. Use USB Drive or SD Card

The following explains how to download project file by using USB drive or SD card.

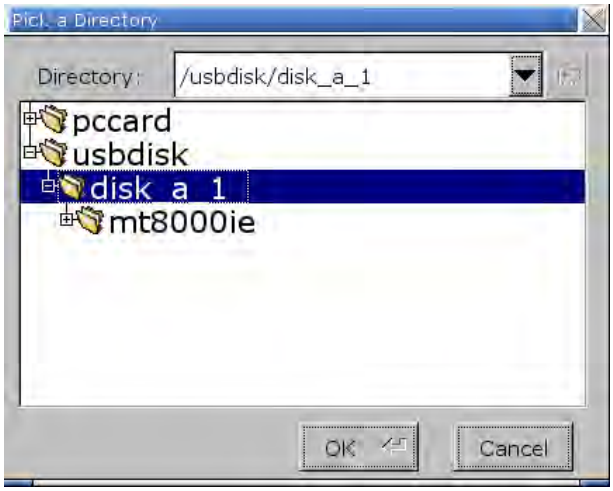
1. On EasyBuilder Pro toolbar click [Project] » [Build for SD card / USB disk download]. Browse for the project file and then click [Build], the file for download will be generated in the connected external device.



2. Insert an external device to HMI.
3. Select [Download] on HMI and enter password.



4. After password is confirmed, it will show the directories in the external device. (pccard: SD Card; usbdisk: USB Drive)



5. Select the directory that contains project, and then click [OK] to start downloading.

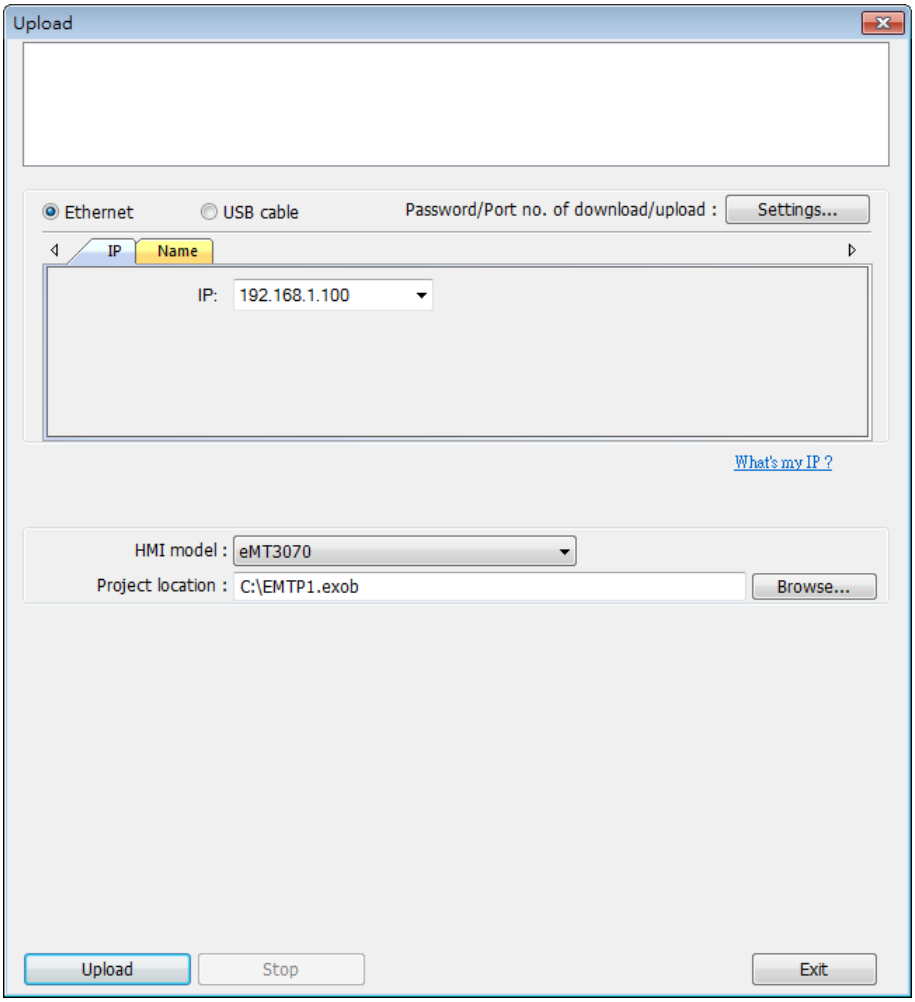
Note

- Please select the parent directory of the generated files when downloading. For the structure above, please select **disk_a_1**, not **mt8000ie**.
- You may click [System Settings] to save the hardware settings configured in EasyBuilder Pro into SD card or USB disk, and then download the settings file to HMI. See “4 Hardware Settings” for more information.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

3.7. Upload the Project File from HMI

1. On EasyBuilder Pro toolbar click [File] » [Upload].
2. Set HMI IP, HMI model, project location and project name, and then click [Upload].



4. Hardware Settings

This chapter explains HMI settings.

4.1. Overview 4-2
4.2. I/O Ports 4-2
4.3. LED Indicators 4-2
4.4. System Reset 4-3
4.5. System Toolbar 4-4

4.1. Overview

This chapter discusses the HMI settings.

4.2. I/O Ports

The I/O ports may vary between models; please see the datasheet for more information.
The I/O Ports include:

- SD card slot: Allows downloading / uploading project file, including recipe data, event log, data log…etc and also allows recording history data and data backup.
- COM Port: Connects PLC or other peripheral devices. The type of serial port include: RS-232, RS-485 2W, RS-485 4W, and CAN Bus.
- Ethernet: Allows downloading / uploading project file, including recipe data, event log, data log…etc. Connects to Ethernet devices, such as PLC, laptop.
- USB Host: Supports USB devices, such as mouse, keyboard, USB disk, printer, or barcode device.
- USB Client: Allows downloading / uploading project file, including recipe data, event log, data log…etc and also allows recording history data and data backup.

For the first time operating HMI, please complete the following system settings. When finished, the project files designed using EasyBuilder Pro can be used on HMI.

4.3. LED Indicators

The LED indicators on the HMI indicate:

Models: MT8121XE, MT8150XE, MT8121iE, MT8150iE

LED	Description
PWR (Orange)	Indicates power status.
CPU (Green)	Blinks when read/write Flash Memory.
COM (Blue)	Indicates COM port communication status, blinks during communication. When communication is good, it may stay on. (Not including network communication)

Other Models:

LED	Description
PWR (Orange)	Indicates power status.
CPU (Green)	Indicates CPU status. If it blinks or goes out, there

	may be a CPU error.
COM (Blue/Red)	Indicates communication status, blinks during communication. When communication is good, it may stay on.

4.4. System Reset

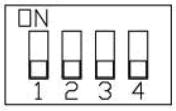
Certain HMI models are equipped with a reset button and a set of DIP switches. When using the DIP switches to change modes, the corresponding functions will be triggered. If system password is lost or forgotten, see the following steps to restore factory default.

1. Flip DIP Switch 1 to ON and the rest to OFF, and then reboot HMI. HMI will switch to touch screen calibration mode.
2. A "+" sign appears on the screen, touch the center of the sign, after all 5 signs are touched, "+" disappears and the touch screen parameters will be stored in HMI system.
3. After calibration, confirm to restore the default password, select [Yes].
4. Confirm to restore the default password again by typing [yes] and clicking [OK]. The project files and history records stored in HMI will all be removed. (The default Local Password is 111111. However, other passwords, such as Download/Upload passwords have to be reset.)

The following lists the DIP switch settings of different models. Please see the relevant installation instruction.

cMT / cMT X / eMT / iE

Dip Switch



SW1	SW2	SW3	SW4	Mode
ON	OFF	OFF	OFF	Touch Screen Calibration Mode
OFF	ON	OFF	OFF	Hide HMI System Setting Bar
OFF	OFF	ON	OFF	Boot Loader Mode
OFF	OFF	OFF	ON	Reserved
OFF	OFF	OFF	OFF	Normal

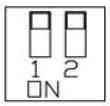
Note

- The state of DIP Switch 4 on each unit may be different. If it should be ON when out from factory, the Dip Switch 4 would be set ON and cut off. If it should be OFF, the Dip Switch 4 would be set OFF but the switch is not cut.

SW1	SW2	Mode
ON	ON	Restore factory default
ON	OFF	Hide system setting bar

mTV

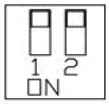
Dip Switch



OFF	ON	Boot loader mode
OFF	OFF	Normal mode

cMT-SVR

Dip Switch



SW1	SW2	Mode
ON	ON	Restore factory default
ON	OFF	Restore Ethernet IP Settings
OFF	ON	Boot loader mode
OFF	OFF	Normal mode

4.5. System Toolbar

After rebooting HMI, you can set the system with [System Toolbar] at the bottom of the screen. Normally, this bar is hidden automatically. Only by touching the arrow icon at the bottom-right corner of the screen will the System Toolbar pop up. From left to right the icons are: System Settings, System Information, Text Keyboard, and Number Keyboard.

How to hide HMI System Setting Toolbar:

- When [DIP Switch 2] is set ON, the system setting toolbar is disabled. When set OFF; the system setting toolbar is enabled. Please restart HMI to enable/disable the toolbar.
- For mTV Series, flip DIP Switch 1 to ON to hide system setting toolbar.
- System register [LB-9020] can also enable/disable system setting toolbar. When [LB-9020] is set ON, the toolbar is displayed, and set OFF to hide the toolbar.

4.5.1. System Setting

cMT/cMT X Series

Set or modify system parameters. Confirm password for security first. The factory default password is 111111.

CODESYS

Enter the activation code to activate CODESYS.

General

General settings including FTP settings.

[Direction]

Set screen orientation mode to landscape or portrait by selecting degrees from 0 / 90 / 180 / 270.

[Brightness]

Adjust screen brightness.

[Popup download window]

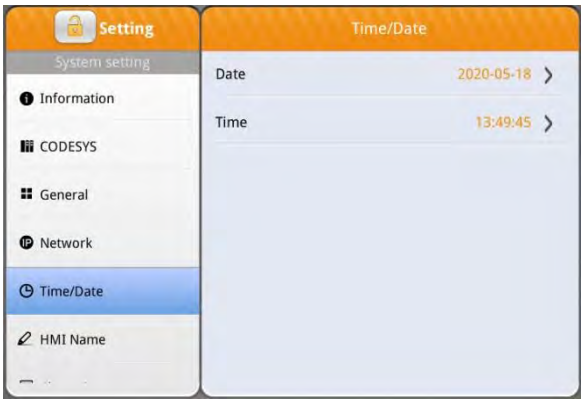
When enabled, the download window can pop up.

[Show Start button]

When enabled, the Start button is displayed.

Network

Download project file to HMI via Ethernet. The IP address of the target device (HMI) must be correctly set. Users can choose from obtaining an IP address automatically or entering an IP address manually. To use Email or EasyAccess 2.0, please set DNS correctly.



Time/Date

Set HMI local date and time.

HMI name

Set the HMI name for identification when downloading / uploading a project.

Clear History

Clear historical data stored in HMI.

Project Management

HMI can be rebooted by clicking the Restart button.

System Password

By default the password is 111111.

User-defined passwords can be used after verification.

Enhanced Security

Add or delete user accounts and manage user privileges.

EasyAccess 2.0

Activate EasyAccess 2.0 service.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Printer

Please see Chapter 23.3 in EasyBuilder Pro user manual for more information on installing printers.

OS Upgrade

Select an OS file saved in an SD card or USB disk to upgrade OS. When OS is upgrading, please do not cut off HMI power. For more information on upgrade steps or notes, see the OS upgrade manual of the model in use.

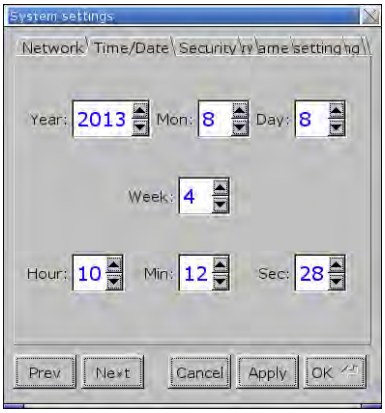
Reset options

Restore HMI to factory settings.

eMT/iE/XE/mTV Series

Network

When downloading project file to HMI via Ethernet, set the correct IP of the target HMI. You can obtain an IP address automatically or enter the IP address manually. To use Email and EasyAccess2.0, please set correct DNS address.



Time / Date

Set HMI local time and date.

Security

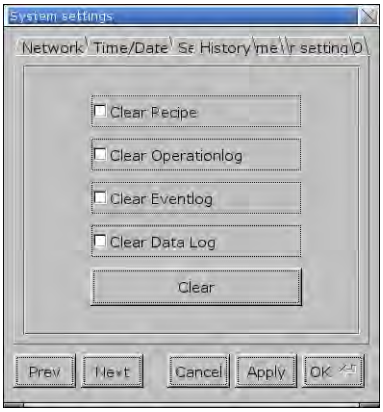
Password protection, the default is 111111. Please click the buttons to set the passwords, and finish password confirmation.

[Password for entering system]

[Password for uploading project]

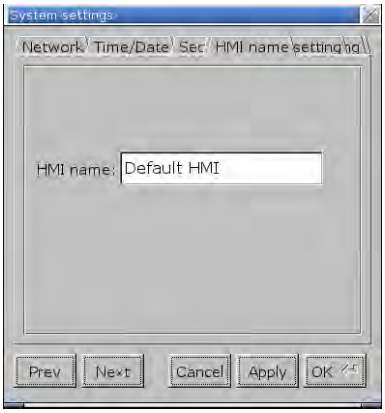
[Password for downloading project]

[Password for uploading history data]



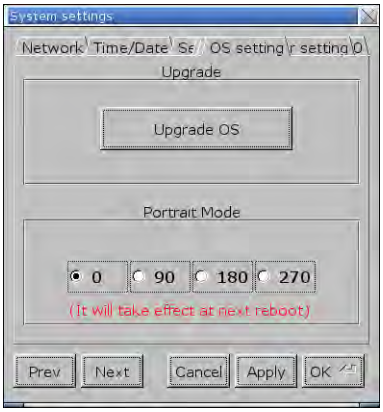
History

Clears history data in HMI.



HMI name

Set HMI name to be used when download/upload project.



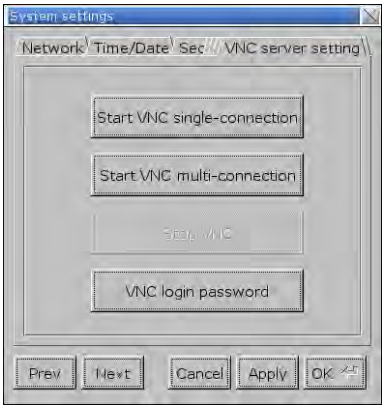
OS setting

[Upgrade OS]

Upgrade firmware. Please do not turn off or unplug HMI during OS upgrade process. For more details, see the instruction about OS upgrade of the model used.

[Portrait Mode]

Set screen orientation mode. After changing the mode, reconnect HMI to power supply, for the setting to take effect. That is, disconnect all power from HMI, and then connect again. If portrait mode is used (90 or 270 degree), the project must be designed for portrait mode, otherwise it cannot be correctly displayed.



VNC server setting

Remote HMI monitoring and controlling via Ethernet.

[Start VNC single-connection]

Allows connection with one VNC client.

[Start VNC multi-connection]

Allows connection with multiple VNC clients. Connecting more VNC clients may slow down the communication speed.

Please see the settings steps in the later part.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

Misc 1

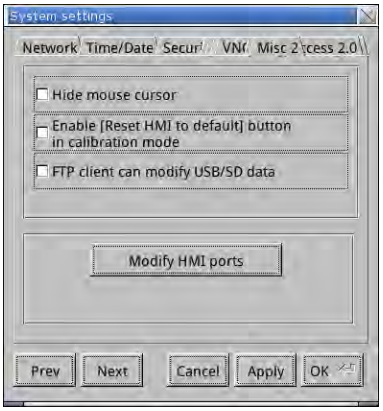
Rotary switch for adjusting LCD brightness.

[Popup download window]

If selected, after inserting USB disk or SD card to HMI, the Upload / Download dialog box shows.

[Restart after download/upload]

If selected, restarts HMI automatically after uploading / downloading project.



Misc 2

[Hide mouse cursor]

If selected, the mouse cursor will be hidden.

[Enable [Reset HMI to default] button in calibration mode]

If selected, when the operator presses and holds anywhere on the screen for more than 2 seconds during HMI startup, the touch screen calibration mode will start. After calibration, [Reset HMI to default] option shows.

[FTP client can modify USB/SD data]

If selected, USB/SD data can be modified using FTP.

[Modify HMI ports]

Change the port number for Upload/Download and FTP.

EasyAccess 2.0

Activate EasyAccess 2.0 service.

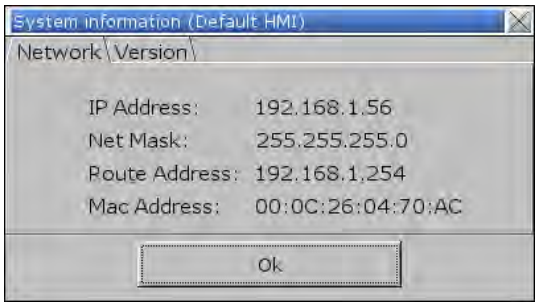
Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

The following steps explain how to set VNC server.

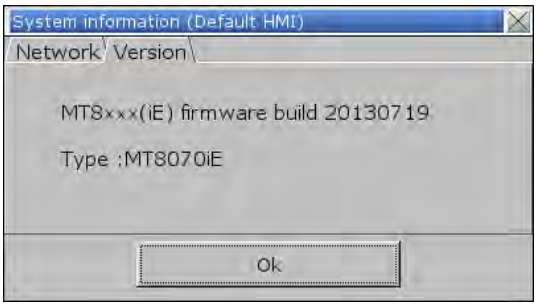
1. Enable HMI VNC server, set password.
2. Install Java IE or VNC Viewer on PC.
3. Enter remote HMI IP in Internet Browser. Or, in VNC Viewer enter remote HMI IP and password.

4.5.2. System Information

Network: Displays network information & HMI IP address.



Version: Displays HMI firmware version and model type.



5. System Parameter Settings

This chapter introduces the system parameter settings.

5.1. Overview 5-2
5.2. Device 5-2
5.3. Model 5-11
5.4. General 5-13
5.5. System Setting 5-20
5.6. Remote 5-23
5.7. Security 5-25
5.8. Extended Memory 5-28
5.9. Printer / Backup Server 5-31
5.10. Time Sync./DST 5-32
5.11. e-Mail 5-34
5.12. Cellular Data Network 5-36
5.13. FTP 5-38

5.1. Overview

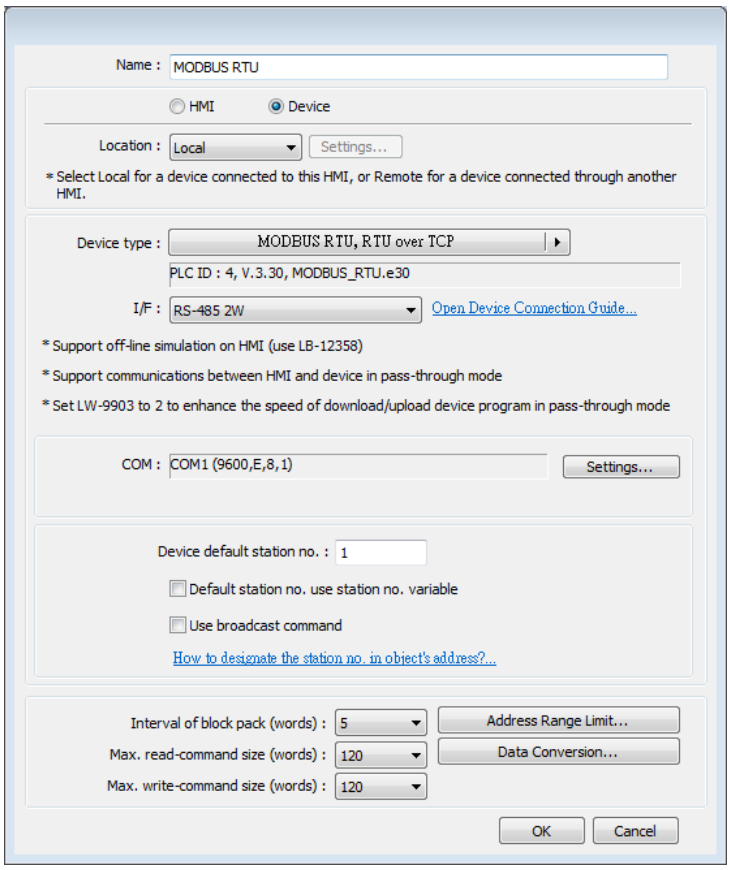
Launch EasyBuilder Pro, in the main menu select [Home] » [System Parameters] to open the [System Parameter Settings] dialog box. System Parameter Settings are divided into several tabs, these tabs will be introduced respectively in this chapter.

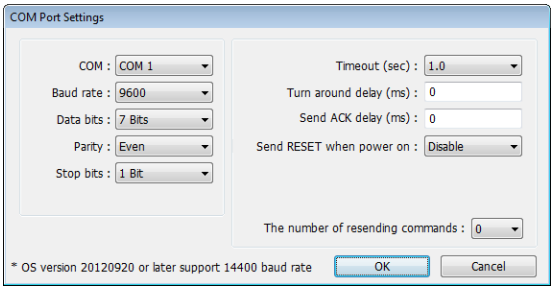
5.2. Device

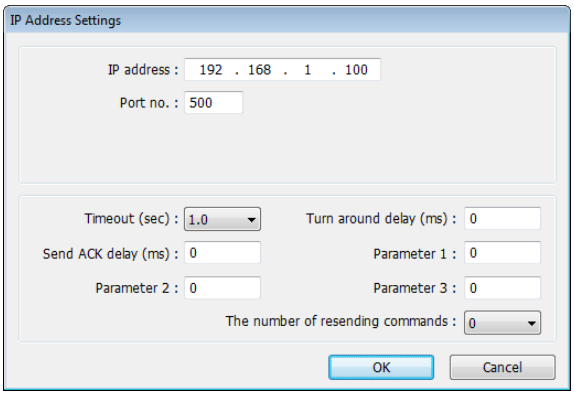
Parameters in this tab determine the attributes of each device connected with HMI. The device can be a Local / Remote HMI / PLC. A default device "Local HMI" exists when creating a new project. This device is the HMI that will be updated and programmed.

5.2.1. How to Control a Local PLC

“Local Device” is a device connected to the local HMI. To control/connect a Local Device, add this type of device first. Click [System Parameters] » [New] to open [Device Properties] dialog box. For example, when connecting MODBUS RTU as a Local Device:



Setting	Description
Name	The name of the device.
HMI / Device	In this example a device is used, so select [Device].
Location	Select [Local] or [Remote]. In this example the device is connected to the Local HMI, so select [Local].
Device type	Select the type of the device.
I/F	The available interface: [RS-232], [RS-485 2W], [RS-485 4W], [Ethernet], [USB], and [CAN Bus]. ● If the interface used is [RS-232], [RS-485 2W], or [RS-485 4W], configure communication parameters by clicking [Device Properties] » [Settings] and then [Com Port Settings] dialog box opens.  **Timeout** If the communication has been disconnected for more than preset time limit configured in [Timeout] (in sec), Window No. 5 will pop up and show “Device No Response” message. **Turn around delay** While sending the next command to the device, HMI will delay the sending according to the time interval set in [Turn around delay]. This may influence the efficiency of the communication between HMI and the device. Default value is “0”. Note: If the device used is SIEMENS S7-200 Series, it is recommended to assign “5” to [Turn around delay] and “30” to [Send ACK delay]. ● If the interface used is [Ethernet], click [Device Properties] » [Settings] and the [IP Address Settings] dialog box opens. Please set correct device IP address and port number.

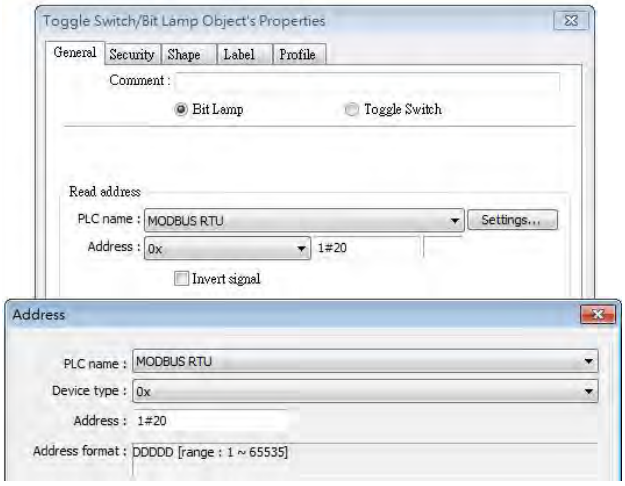


- If the interface is [USB], no further setting is required. Please check the settings in [Device Properties].
- If the interface is [CAN (Controller Area Network) Bus], please see “Device Connection Guide” for “CANopen” and import the .eds device file.

Device default station no.

Set the default station number for device address if the device station number is not included in the address. Device station no. can be set in its address. The address format: ABC#Addr

ABC stands for device station number and ranges from 0 to 255. Addr stands for device address. And the “#” sign separates the station number and the address. As shown in the following figure, the data is read from device station number 1, and address 0x-20.



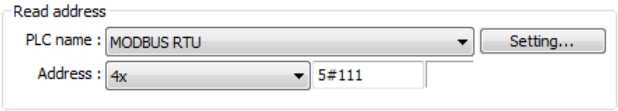
Default station no. use station no. variable

Use the station number variables as the default device station number. Select one from LW-10000 to LW-10015 (var0 to var15) as the station number variables. If the

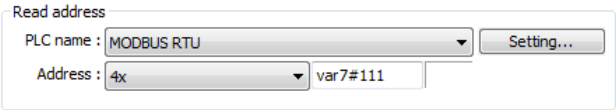
station no. is not specified in device address, the station number will be determined by the station no. variable. For example, if var3 is set for default station no:

The followings demonstrate some examples:

- The device station number is “5”.



- The device station number is determined by var7 (LW-10007)

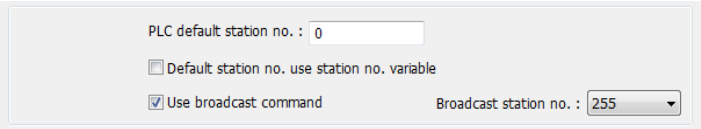


- Device address is set to “111”, since device station no. is not specified, and the default station no. is var3, the device station no. is determined by var3 (LW-10003).

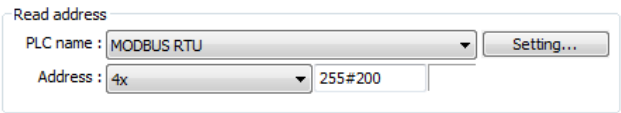
Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Use broadcast command

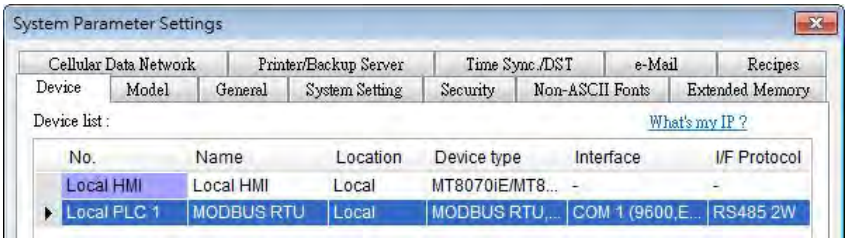
When [Use broadcast command] check box is selected, please fill in [Broadcast station no.] according to the broadcast station number defined by the device. When HMI sends a broadcast command to the station number set here, the device will only receive the command and not reply to HMI.



As shown in the following figure:

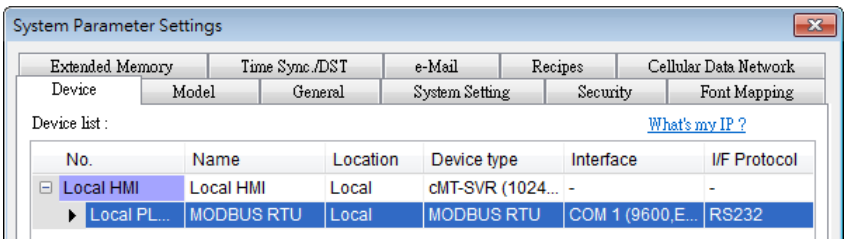
	 When HMI sends a command to address 255#200, all the devices will receive this command and will not reply. Only devices that support broadcast command can use this feature.
Interval of block pack (words)	If the interval between read addresses of different commands is less than this value, the commands can be combined to one. The combining function is disabled if this value is set to “0”. For example, the interval value is set to “5”, to read 1 word from LW-3 and 2 words from LW-6 respectively (read from LW-6 to LW-7), since the interval of addresses between LW-3 and LW-6 is less than 5, these two commands can be combined to one. The result is to read 5 consecutive words from LW-3 to LW-7. Note: The maximum size of command combination data must be less than [Max. read-command size].
Max. read - command size (words)	The maximum data size to read from the device at one time. Unit: word
Max. write - command size (words)	The maximum data size to write to the device at one time. Unit: word.

After all settings are completed, a new device named “Local PLC 1” is added to the [Device list].



Note

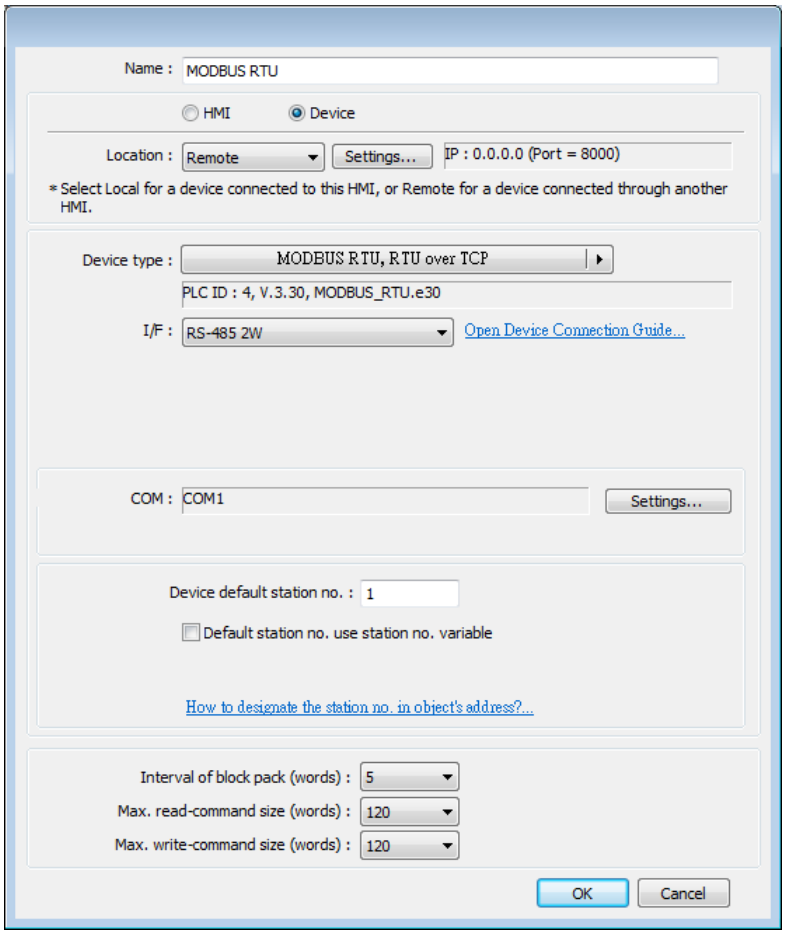
- When using a cMT / cMT X Series model, select “Local HMI” in [System Parameter Settings] dialog box and then click [New Device] to add a “Local PLC 1” under “Local HMI”.

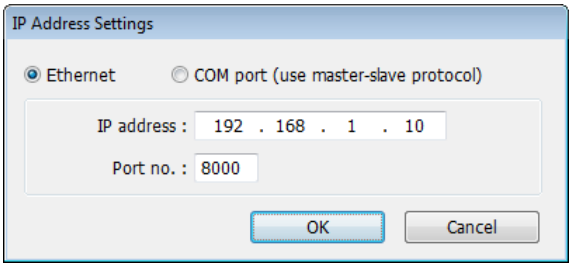


Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

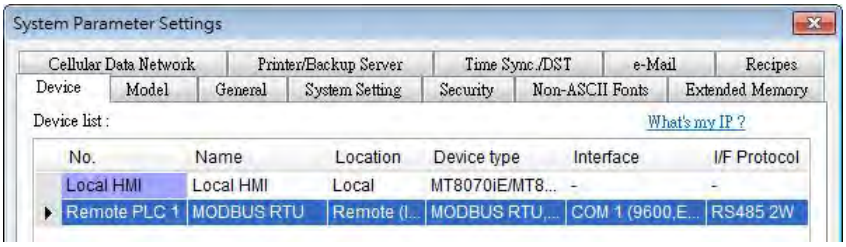
5.2.2. How to Control a Remote Device

"Remote Device" is a device being connected to a remote HMI. To control a remote device, add this type of device first. Please click [System Parameters] » [New] to open [Device Properties] dialog box. For example, use MODBUS RTU as the Remote Device:



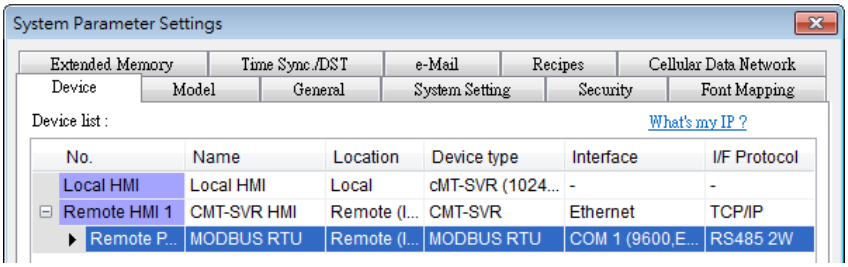
Setting	Description
HMI or Device	In this example, a device is used, so select [Device].
Location	Select [Local] or [Remote]. In this example the device is connected to Remote HMI, so select [Remote]. Set the IP address and port number of the Remote HMI by clicking [Settings] next to [Remote]. 
Device Type	Select the type of the device.
I/F	The interface used for Remote Device. If the remote device uses a COM port, select [RS-232], [RS-485 2W], or [RS485 4W].
COM	Set the correct COM port used by the Remote Device.
Device default station no.	Set the station number of Remote Device.

After all settings are completed, a new device named “Remote PLC 1” is added to the [Device list].



Note

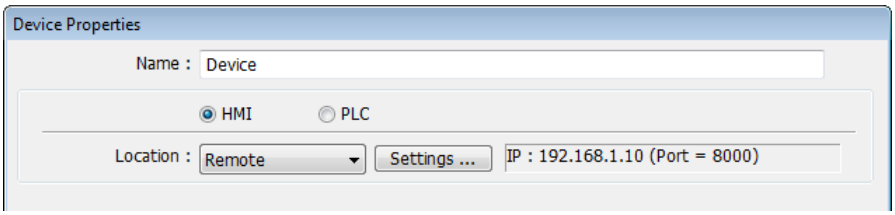
- When using a cMT / cMT X Series model, select the defined “Remote HMI 1” in [System Parameters] dialog box and then click [New Device] to add a “Remote PLC 1” under “Remote HMI 1”.

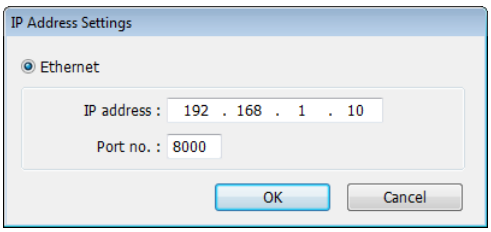


Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

5.2.3. How to Control a Remote HMI

"Remote HMI" is the HMI other than "Local HMI", and PC is also a "Remote HMI". To control a Remote HMI, add this type of device first. Click [System Parameters] » [New] to open [Device Properties] dialog box as shown in the following figure:

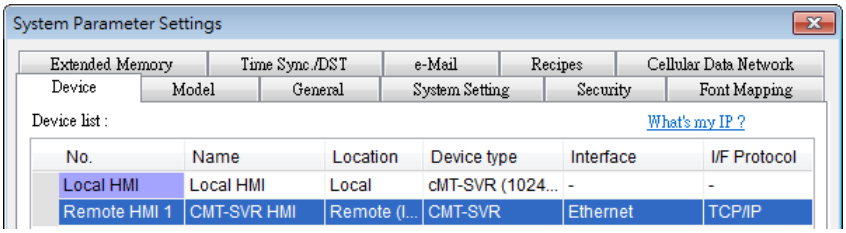


Setting	Description
HMI or Device	In this example, the device used is a HMI, so select [HMI].
Location	Select [Local] or [Remote]. In this example Remote HMI is used, select [Remote]. Set the IP address and port number of the Remote HMI by clicking [Device Properties] » [Location] » [Settings]. 

After all settings are completed, a new device named "Remote HMI 1" is added to the [Device list].

Note

- When using a cMT / cMT X Series model, in [Device list] click [New HMI] to add a "Remote HMI 1".

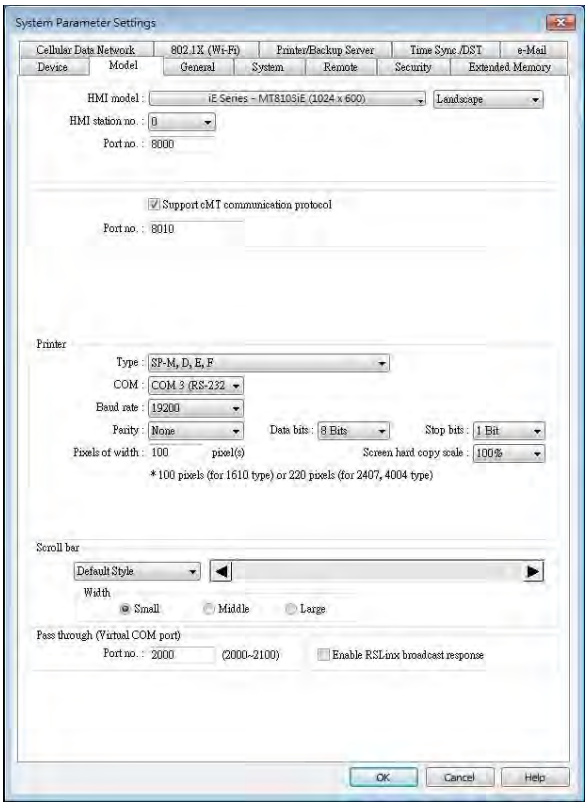


Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

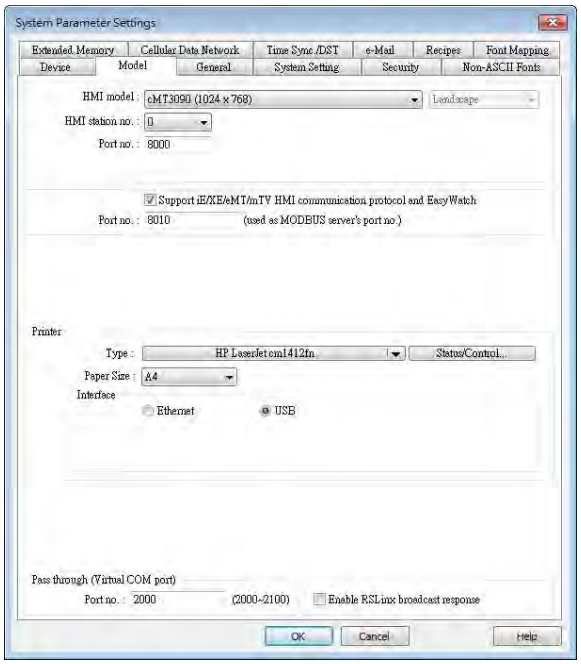
5.3. Model

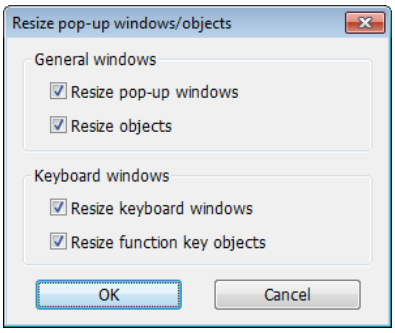
Configure the [HMI model], [Timer], [Printer] and [Scroll bar] settings, and more.

eMT, iE, XE, mTV Series



cMT, cMT X Series



Setting	Description
HMI model	Select an HMI model. The [Resize pop-up windows / objects] dialog box will pop up when changing to a model with a different resolution. Select required adjustment and click [OK]. In most cases, select all options.  **Landscape/Portrait** Change the orientation of the project.
HMI station no.	Set the station number of current HMI. The default value is "0".
Port no.	Set the port number of current HMI. The default value is "8000".
Support cMT communication protocol	Supports communication with cMT / cMT X Series models. cMT / cMT X Series project file should use the same communication port.
Printer (eMT, iE, XE, mTV Series)	**Type** A printer can be connected with the HMI. The HP PCL Series printer is connected through USB interface while other printers through a COM port. For more information, see "23 HMI Supported Printers". If the printer is connected through [COM], configure the parameters correctly. If the printer type is [SP-M, D, E, F], the [pixels of width] has to be set accurately, i.e. the set pixel(s) cannot exceed printer's default setting, or the HMI will fail to print data.
Printer (cMT, cMT X Series)	The printer driver can be installed on HMI. **Type** Select printer type. **Status address**

Shows printer status.

LW-n	Status
0	Printer driver is not installed yet.
1	Installing printer driver.
2	Printer is ready.
3	Printer is printing.
4	Print job is canceled.
LW-n+1	Error
0	None
1	Printer is not found.
2	Unknown error.

Control address

Updates printer connection parameters.

LW-n	Command
0	None
1	Update connection parameters.
2	Restart printer driver.
LW-n+1	I/O
0	Ethernet
1	USB
LW-n+2	IP address (Total: 4 words)
LW-n+6	Port (Default port number: 9100)

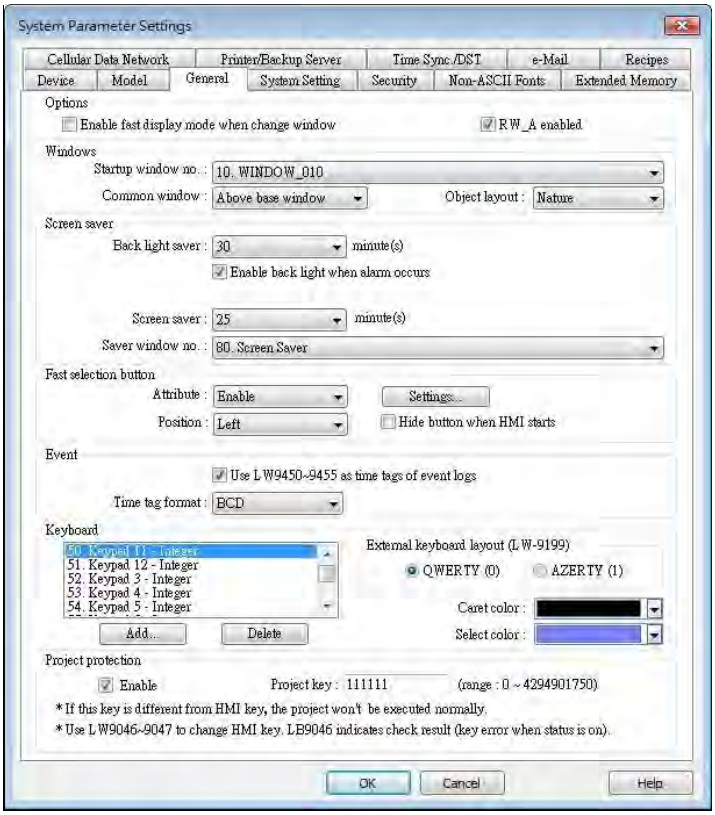
Please note that using only the ppd file for a printer may not be able to run the printer since the corresponding driver is absent. Using tested models is recommended.

Scroll bar	Set the width of Scroll Bar, when the size of the object is too small to display the contents, a scroll bar is displayed in the object. This feature can be applied to objects that allow scrolling, such as Alarm Display, Event Display, History Data Display, and Option List.
Pass through (Virtual COM port)	Set the port number for Pass-through communication.

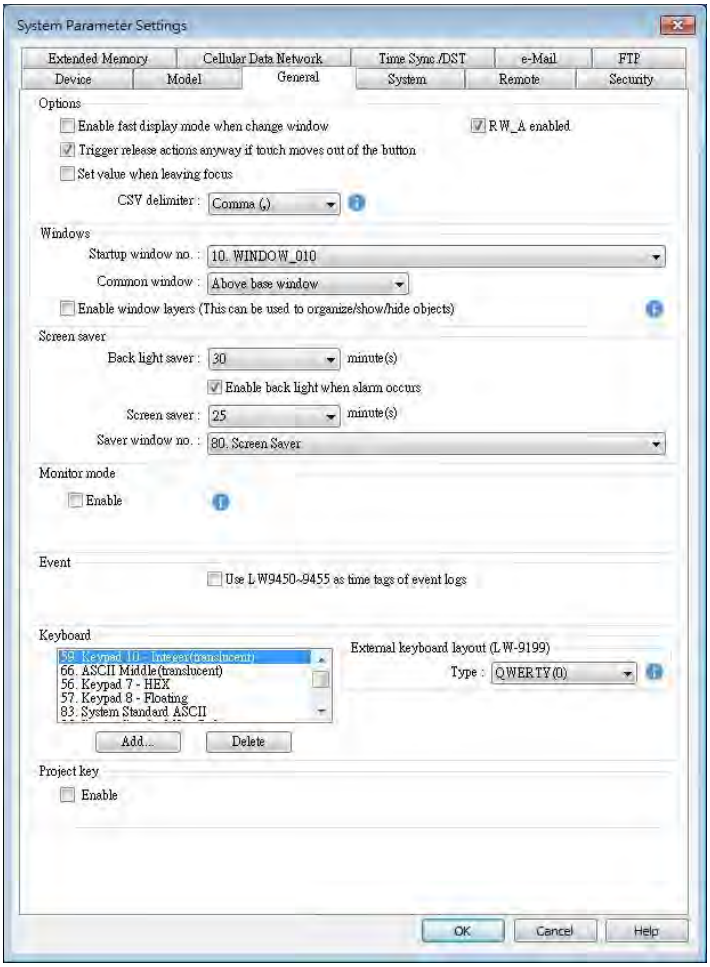
5.4. General

Configure the properties related to screen display.

eMT, iE, XE, mTV Series



cMT, cMT X Series



Setting	Description
Options	**Enable fast display mode when change window** With this option selected, certain objects will cache the most recent value and display it after changing window, to achieve fast display. The objects are: Bit Lamp, Word Lamp, Numeric, ASCII, Bar Graph, Meter Display, Animation, and Moving Shape…etc. **RW_A enabled** Enable or disable recipe data RW_A. Enable this, the objects can then control RW_A .The size of RW_A is 64K. **Trigger release action anyway if touch moves out of the button.** (cMT / cMT X Series) With this option selected, releasing the finger after touching the button and then sliding the finger out of the button area is still considered as button release; its release action(s) will be triggered. **Set value when leaving focus (cMT / cMT X Series)** With this option selected, values will be written upon leaving the input mode of an object (e.g. Numeric object), even if not by pressing the ENTER key. For example, data entry for the current numerical object is considered completed and value is written when one selects another object to enter value. An alert message shows when an invalid value is entered. **Note** 1. Values will not be retained after pressing the ESC key to leave. 2. When the system keyboard is used, values will be directly written after closing the keyboard. 3. With [Input order] enabled, the sequential input is performed after leaving the object by pressing the ENTER key. The sequential input will be stopped if another object is touched during the input process. **CSV delimiter (cMT / cMT X Series)** Save the CSV file generated from Backup or Import/Export with Comma (,) or Semicolon (;) delimiter.
Windows	**Startup window no.** Designate the window shown when start up HMI. **Common window** The objects in the common window (Window No. 4) will be

shown in each base window. This determines that the objects in common window are placed above or below the objects in the base window.

Object layout

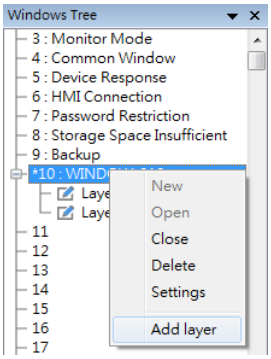
If [Control] mode is selected, when operating HMI, [Animation] and [Moving Shape] objects will be displayed above other kinds of objects neglecting the sequence that the objects are created. If [Nature] mode is selected, the display will follow the sequence that the objects are created, the first created will be displayed first.

Enable window layers (This can be used to organize/show/hide objects)

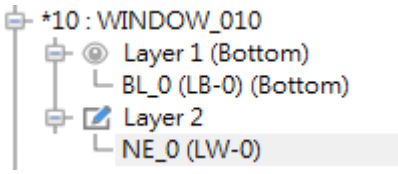
With this option selected, layers within a window can be created in the window tree.

At most 10 layers can be added to a window.

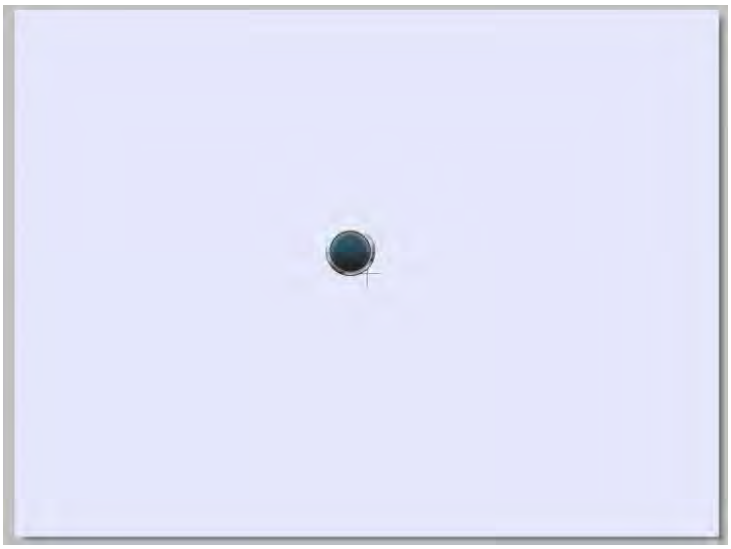
As shown in the following figure, right-click on a window number and then select [Add layer].



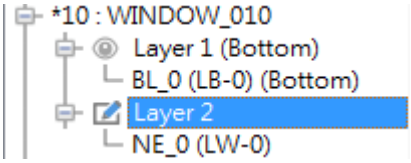
The following is an example of creating a layered window: Place a Bit Lamp object in Layer 1 and a Numeric object in layer 2.

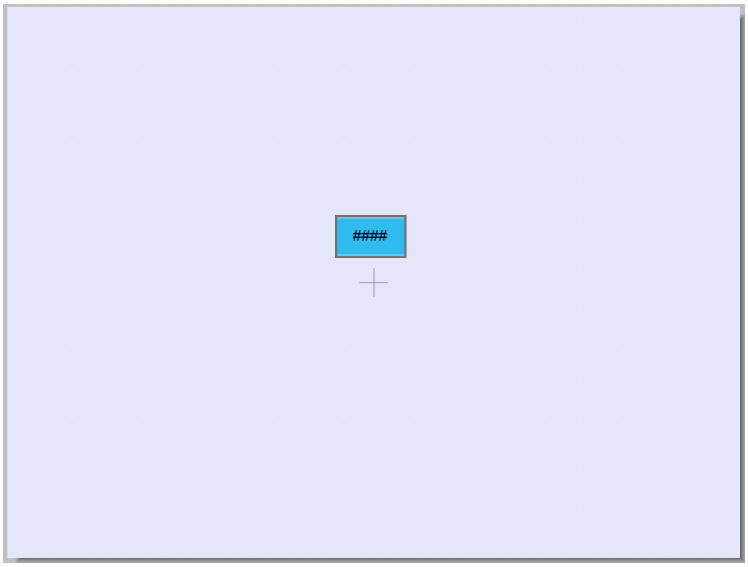


Selecting Layer 1 shows only the Bit Lamp object in the editing window.

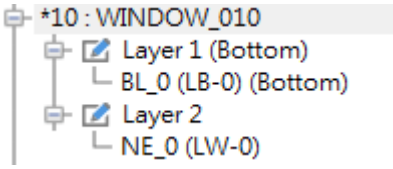


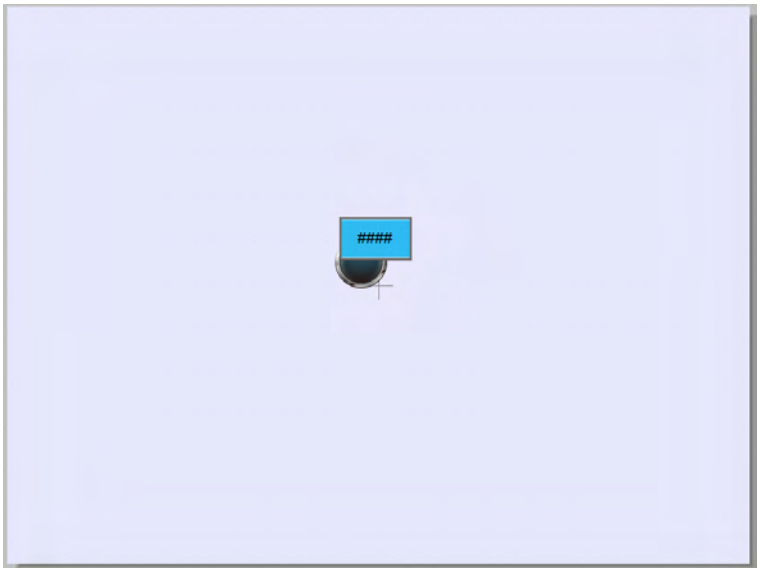
Selecting Layer 2 shows only the Numeric object in the editing window.





Selecting window 10 shows all objects.





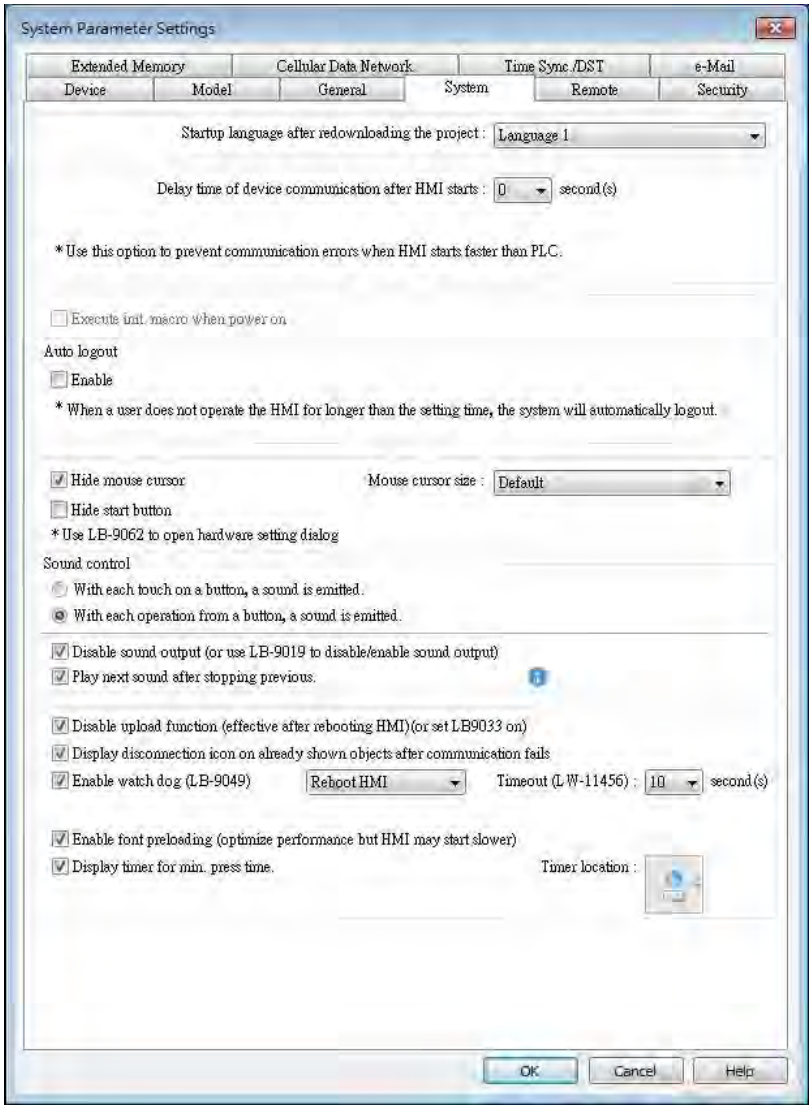
Screen saver	**Back light saver** If the screen is left untouched and reaches the time limit set here, the back light will be turned off. The setting unit is minute. Back light will be on again once the screen is touched. If [none] is set, the back light will always be on. **Screen saver** If the screen is left untouched and reaches the time limit set here, the current screen will automatically switch to a window assigned in [Saver window no.]. The setting unit is minute. If [none] is set, this feature is disabled. **Saver window no.** Specify a window to be displayed when HMI goes to screen saver mode. Note: In a saver window, the [Return to previous window] option of a Function Key can be used to wake up HMI from screen saver mode; therefore, please avoid editing the project in the way shown as the example below, to prevent the HMI from going to screen saver mode repeatedly. Example: A Function Key used to “Return to previous window” is placed in Window no. 11, and the HMI goes into screen saver mode when Window no. 11 is displayed. In this case, the HMI can wake up from screen saver mode when any point on its screen is touched, and Window no. 11 will be displayed again. At this moment, if the Function Key in Window no. 11 is touched, the HMI goes into screen saver mode again. Solution: 1. When a Function Key with “Return to previous window” selected is placed in the saver window, please use “Change full-screen window” for all other windows.

	2. When “Change full-screen window” is used for the saver window to change to a specific window (e.g., home page), please make sure that “Return to previous window” is not used in this destination window. The “Return to previous window” option can still be used for other windows.
Fast selection buttons	This setting is applicable for eMT, iE, XE, mTV, and iP Series. Setting the attributes for fast selection button for Window No. 3. To use the fast selection button, create Window No. 3 first. **Attribute** Enable or disable fast selection window. Select [Enable] and click [Settings] to set the attributes, including color and text of the button. **Position** Select the button position on the screen. If [Left] is chosen, the button will show at bottom left side of the screen; if [Right] is chosen, the button will show at the bottom right side of the screen. **Settings** Set the shape and label font of the Fast Selection Button. **Hide button when HMI starts** The Fast Selection Button will be hidden, calling it out requires system registers LB-9013~LB-9015.
Monitor Mode	This setting is applicable for cMT / cMT X Series. Window no. 3 in the project is monitored in this mode in cMT Viewer. The user can select up to 50 cMT / cMT X Series HMIs to monitor in cMT Viewer, and in 3x3 or 5x4 tile view.
Event	**Use LW9450~9455 as time tags of event logs** When this option is selected, Event Log will use the following time tags to show the time that the event is triggered. LW-9450: time tag of event log – second (range: 0~59) LW-9451: time tag of event log – minute (range: 0~59) LW-9452: time tag of event log – hour (range: 0~23) LW-9453: time tag of event log – day (range: 1~31) LW-9454: time tag of event log – month (range: 1~12) LW-9455: time tag of event log – year (range: 1970~2037) Please note that the system will not be able to use these time tags when one of them exceeds acceptable range. Click the icon to download the demo project. Please confirm your internet connection before downloading the

	demo project.
Keyboard	The window number in which the keyboard is placed. When using Numeric Input or ASCII Input objects, the type of keyboards can be selected. Up to 32 keyboards can be added. To design a keyboard, a window should be designated for creating it. Press [add] after creating and add the window to the list. See “12 Keyboard Design and Usage”. **External keyboard layout** The available USB keyboard layouts are QWERTY, AZERTY, and Cyrillic. System register LW-9199 allows switching keyboard layouts on HMI. **Caret color / Select color** This setting is applicable for eMT, iE, XE, mTV, and iP Series. Set the color of caret that appears when entering data in Numeric Input and ASCII Input objects or change the selection color.
Project protection	Projects can be restricted to be executed by a specific HMI. See “30 Project Protection”.

5.5. System Setting

[System Setting] is used to configure different features of EasyBuilder Pro.



Some features are duplicated from system registers, such as, [Hide system setting bar (LB-9020)], [Hide mouse cursor (LB-9018)], [Disable buzzer (LB-9019)], and [Disable upload function (LB-9033)]. Users can also set these features via system tag.

To select a system tag, select [Address] » [System tag] check box when adding a new object and then select the [Address Type].

To browse all the system tags, Select [Project] » [Tag] » [System] from the main menu of EasyBuilder Pro.

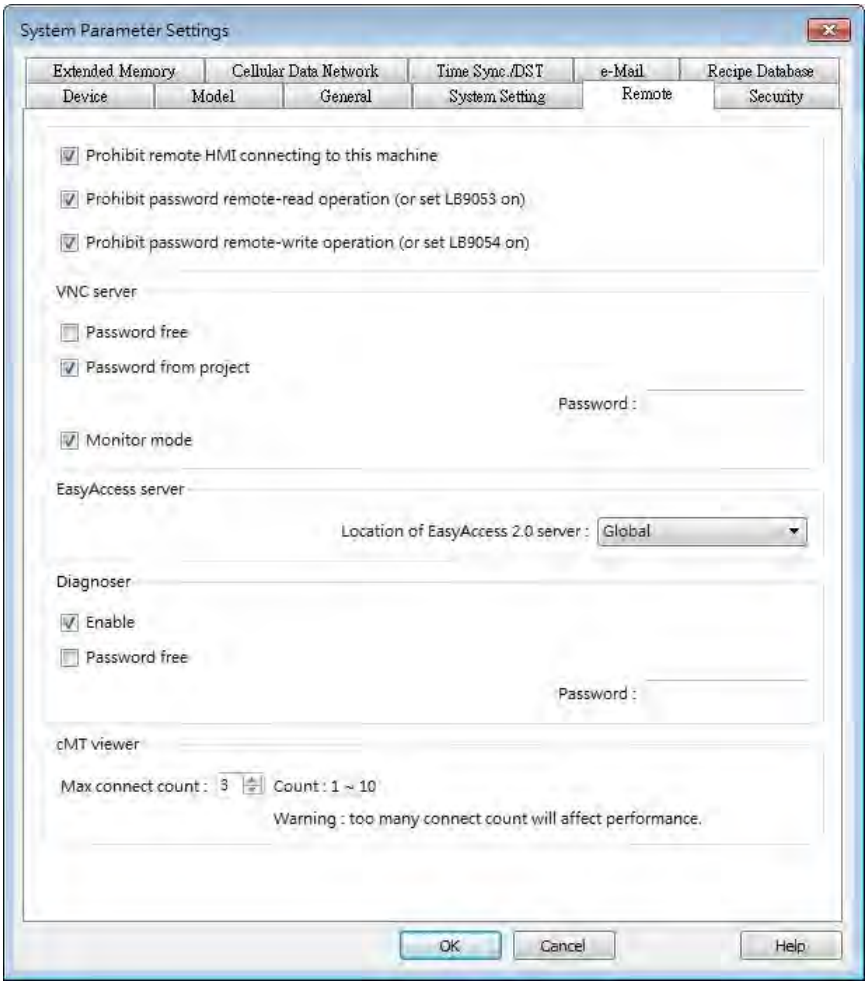
Setting	Description
Startup language after redownloading the project	Set the language to use when HMI starts after the project is re-downloaded.
Delay time of device communication after HMI starts	Certain PLC models need relatively longer startup time. In order to prevent communication error that occurs when HMI

	attempts to communicate with a PLC that has not yet started, a delay time can be set. **Include delaying the execution of all init. Macros when HMI starts.** With this option selected, the macros that have [Execute one time when HMI starts] enabled will also be delayed.
Execute init. MACRO when power on	Designate the macro to be executed when HMI power on.
Auto logout	If leaving HMI untouched for longer than the set time, the objects protected by security classes will not be able to operate. The user ID and password must be entered again to unlock it.
Hide system setting bar / Start Button	Hide the system setting bar from HMI screen.
Hide mouse cursor	Hide the mouse cursor in HMI screen.
Mouse cursor size	Set mouse cursor size.
Sound control	**With each touch on a button, a sound is emitted:** A sound is emitted when touching a button. **With Each operation from a button, a sound is emitted:** When the [Min. press time] is specified, there may be a time gap between touching the object and the action of the object. With this option selected, the sound is emitted when the object actions. **Disable sound output:** Mute HMI. (Not including the sound played when tapping system setting button on HMI.) **Play next sound after stopping previous:** (cMT / cMT X Series only) When the next sound is triggered at the time the current sound is playing, the current sound will stop, and the next sound will start.
Disable upload function (effective after rebooting HMI) (or set LB9033 ON)	Disable HMI to upload project.

Use a disconnection icon or relative objects when device communication fails	If selected, a disconnection icon is displayed on relevant objects when failing to communicate with the device. This icon will be shown in the lower right corner of the object. This icon will only show for disconnection after successful connection.
Enable watch dog (LB-9049)	Watchdog automatically reboots the system after the HMI stops functioning for a specified period of time.
Enable font preloading (optimize performance but HMI may start slower)	(cMT / cMT X Series only) With this option selected, the speed of changing windows that contain more texts is optimized. Please note that using this feature may slow down HMI boot time.
Display timer for min. press time	(cMT / cMT X Series only) With this option selected, a countdown timer is shown when an object with min. press time set is being pressed.

5.6. Remote

Parameters in this tab configure remote connections. Certain functions can be carried out using system registers.



Setting	Description
Prohibit remote HMI connecting to this machine	Prohibit the connection with a remote HMI or cMT Viewer. The remote HMI will not be able to control the local HMI.
Prohibit password remote-read operation (or set LB9053 ON)	Prohibit Remote HMI to read Local HMI's project password and user password.
Prohibit password remote-write operation (or set LB9054 ON)	Prohibit Remote HMI to write to Local HMI's project password and user password.
VNC Server	If **[Password free]** check box is selected, the client can connect with HMI via VNC without entering the password. If **[Monitor mode]** check box is selected, the HMI connected via VNC can only be monitored but not controlled.

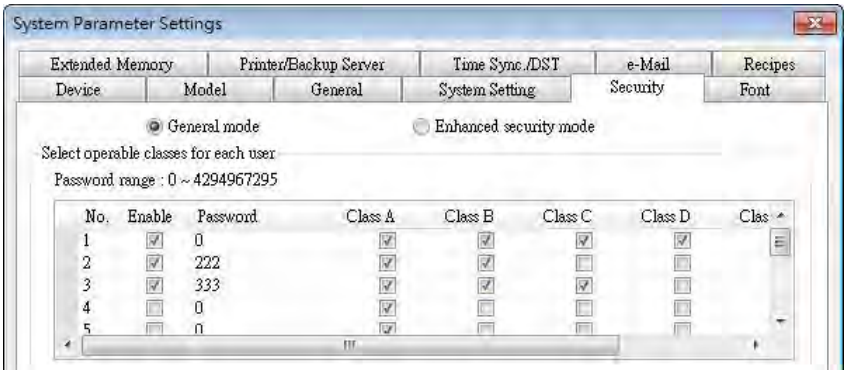
	If **[Password from project]** check box is selected, set the password for VNC login.
Easy Access server	**Login EasyAccess 1.0 server:** Through this technology, users can easily access to any HMI connected to the internet and operate them on PC just like holding touch screen in hand. Easy Access does not transmit updated graphic images directly but only the real time data. This makes transmission really quick and efficient. Please refer to "EasyAccess Manual" for more information. **Location of EasyAccess 2.0 server:** Current location includes Global and China. Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.
Diagnoser	**Enable** When selected, cMT Viewer will allow diagnostic operations when connecting to a cMT / cMT X model. System register LB-12656 can also be used to control Diagnoser. **Password free** Enabling Disgnoser will not require password login. System Register LB-12657 can also be used to set whether a password is required. **Password** Enter the password for logging in Diagnoser. System register LW-11756 (4 words) can also be used to set this password.
cMT Viewer	**Max connect count** Allows specifying the allowable number of cMT Viewer connections for a local cMT / cMT X HMI.

5.7. Security

Parameters in this tab configure the user passwords and security classes. There are two authentication modes: General Mode and Enhanced Security Mode.

For more information, see "10 Security".

5.7.1. General Mode



Up to 12 sets of user and password are available. Password should be one non-negative integer. Once the password is entered, the objects that the user can operate are classified.

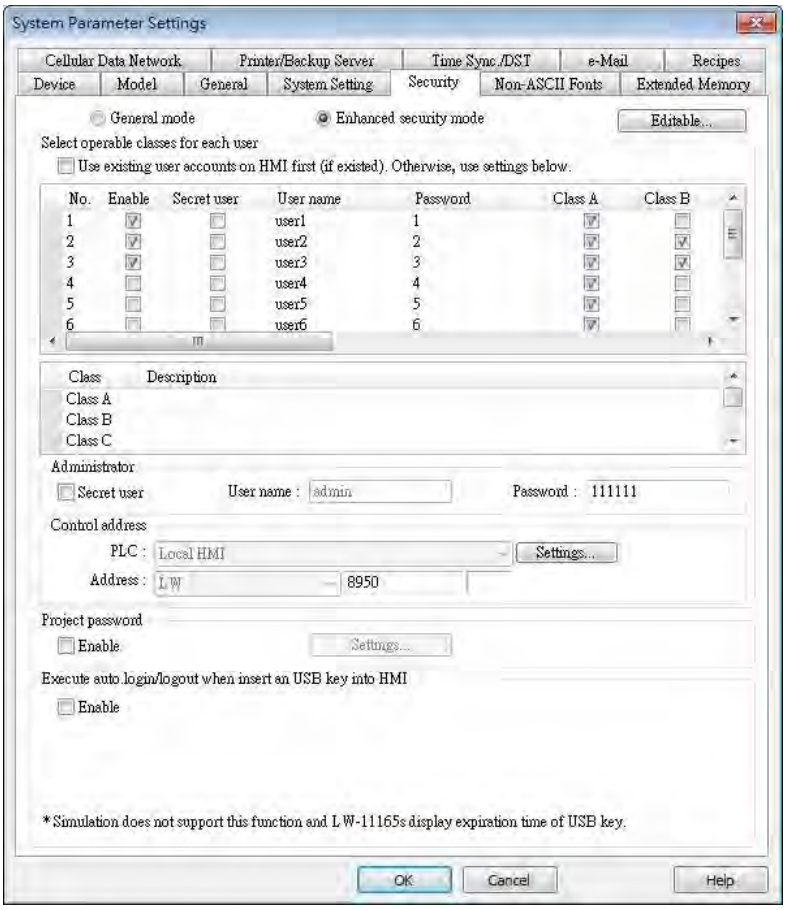
There are six security classes available: A to F.

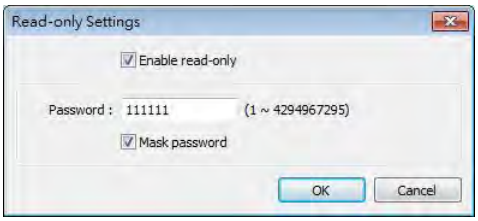
If **[None]** is selected for an object, every user can access this object.

For example, when the security class of User No. 3 is set as the preceding figure, User No. 3 could only access objects of classes A, B, C and “none”.

5.7.2. Enhanced Security Mode

At most 11 users can be set here. An [Administrator] user is provided in this mode. An [Administrator] has all privileges and can operate all object classes. User passwords must be alphanumeric, and each user can have up to 12 classes: A to L.

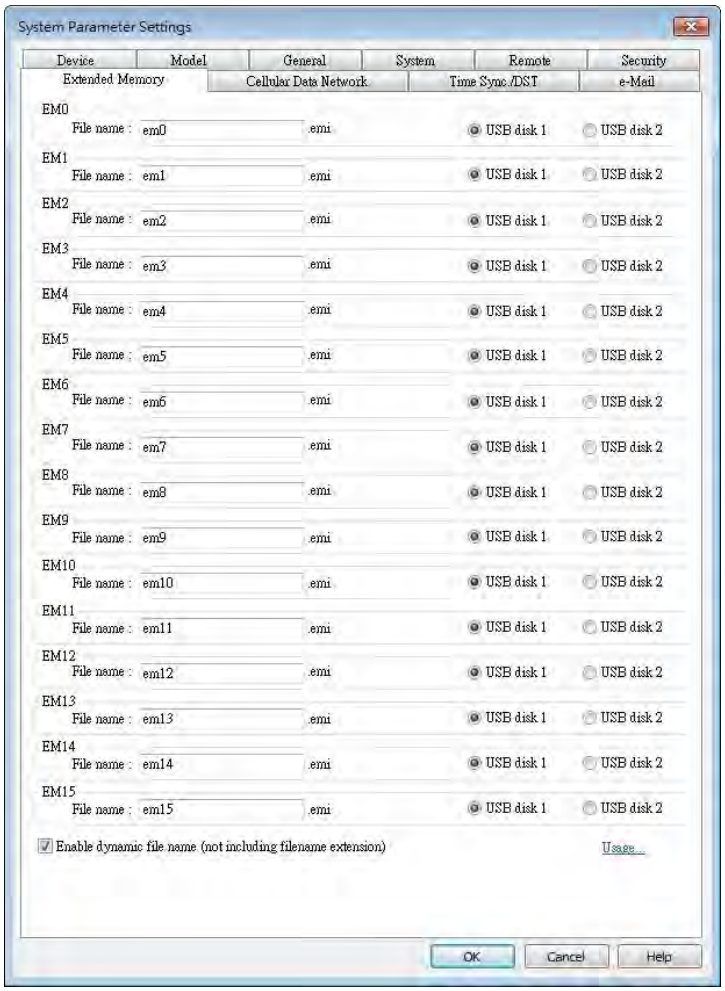


Setting	Description
Editable	Determines whether other users can change the password settings or see passwords.  **Enable read-only** Under this mode, the settings can be viewed but not changed. **Mask password** Passwords are masked by asterisks (*).
Select operable classes for each user	When [Use existing user accounts on HMI] check box is selected, the user accounts existing on HMI will not be erased after downloading the project file to HMI.
Administrator	Default administrator account, cannot be deleted, has all privileges, and cannot be changed. Enhanced Security Mode can be used with Option List object. It displays the account names and privileges. If [Secret

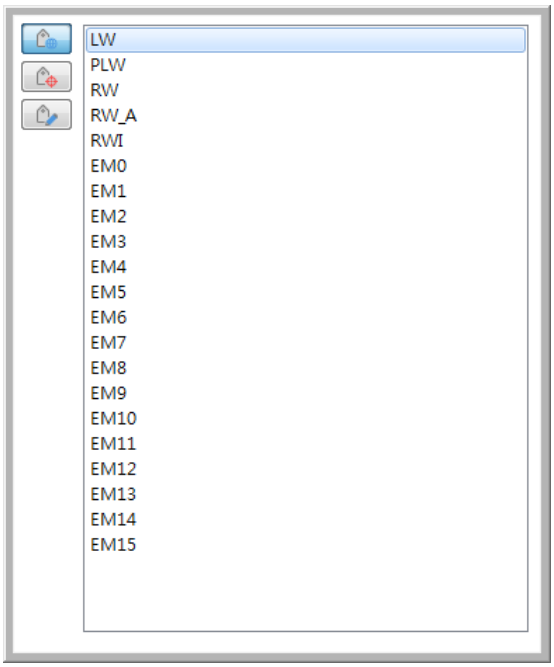
	user] is checked, the account names and privileges will be hidden in Option List.
Control address	An address for users to manage the accounts directly on HMI.
Project password	When this password is enabled, it has to be entered before editing the project file. Select [Enable] and then click [Settings] to set the password.
Execute auto. Login/logout when insert an USB key into HMI	This feature allows automatic login / logout using an USB security key. The login / logout status will be written into a designated address. Insert the USB disk to HMI to log in, and remove the USB disk to log out. The result codes of login / logout: 0x00: No action, 0x01: Login succeeds, 0x04: Login fails, 0x08: Logout succeeds, 0x10: Logout fails.

5.8. Extended Memory

Parameters in this tab determine the location of the extended memory.



Extended Memory is numbered from EM0 to EM15. It works in a way similar to other device types (i.e. LW or RW address). Users can simply select from EM0 to EM15 in the [Address] setting while creating a new object. Maximum allowable data size for each Extended Memory is 2G words.



Data in the extended memory is saved as a file into [SD card] or [USB disk]. By default, the files in [EM0] to [EM15] are with the file names “em0.emi” to “em15.emi” respectively, although users are free to change the file names. To open these files and edit the data in the extended memory, RecipeEditor.exe may be used.

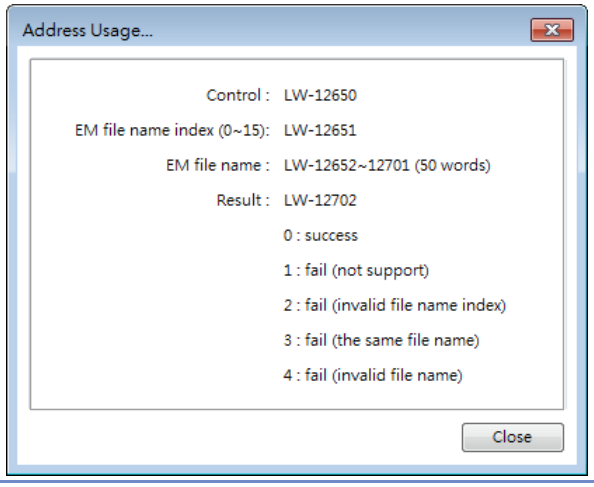
Data in the extended memory will not be erased when HMI power is cut, which means data in the extended memory remains the same as before power off when HMI is started again. This is a behavior similar to recipe data (RW, RW_A).

When the device used as extended memory does not exist, reading the data in the extended memory will get “0”, and writing data to the extended memory will display "Device no response" message on HMI.

Cutting HMI power is not necessary for inserting or removing the external device to or from HMI, making it easy for users to immediately update or obtain data in the extended memory.

Setting	Description
Enable dynamic file name	With this option selected, the file names of EM0~EM15 can be changed dynamically. Select “EM file name index (0~15)”, enter “EM file name”, set “Control” to 1, and then see the “Result”. After successfully changing the file name, the new file name will immediately be the data source of EM address. Please note that after changing the EM file name dynamically, the new file name is retained after

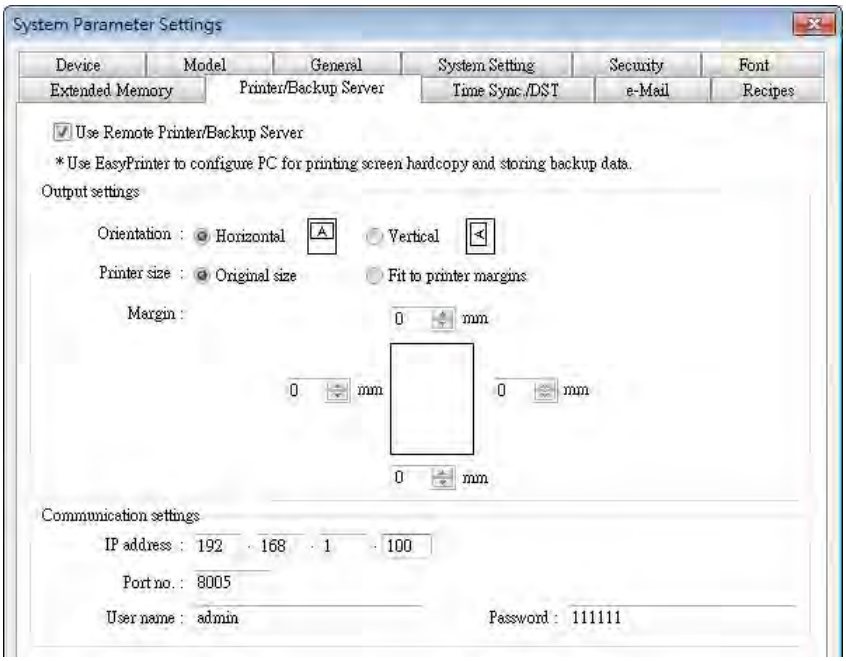
rebooting HMI, which may be different from the file name specified in the project file.



Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

5.9. Printer / Backup Server

Configure remote printer / backup server. This feature is not supported on cMT / cMT X Series HMI.



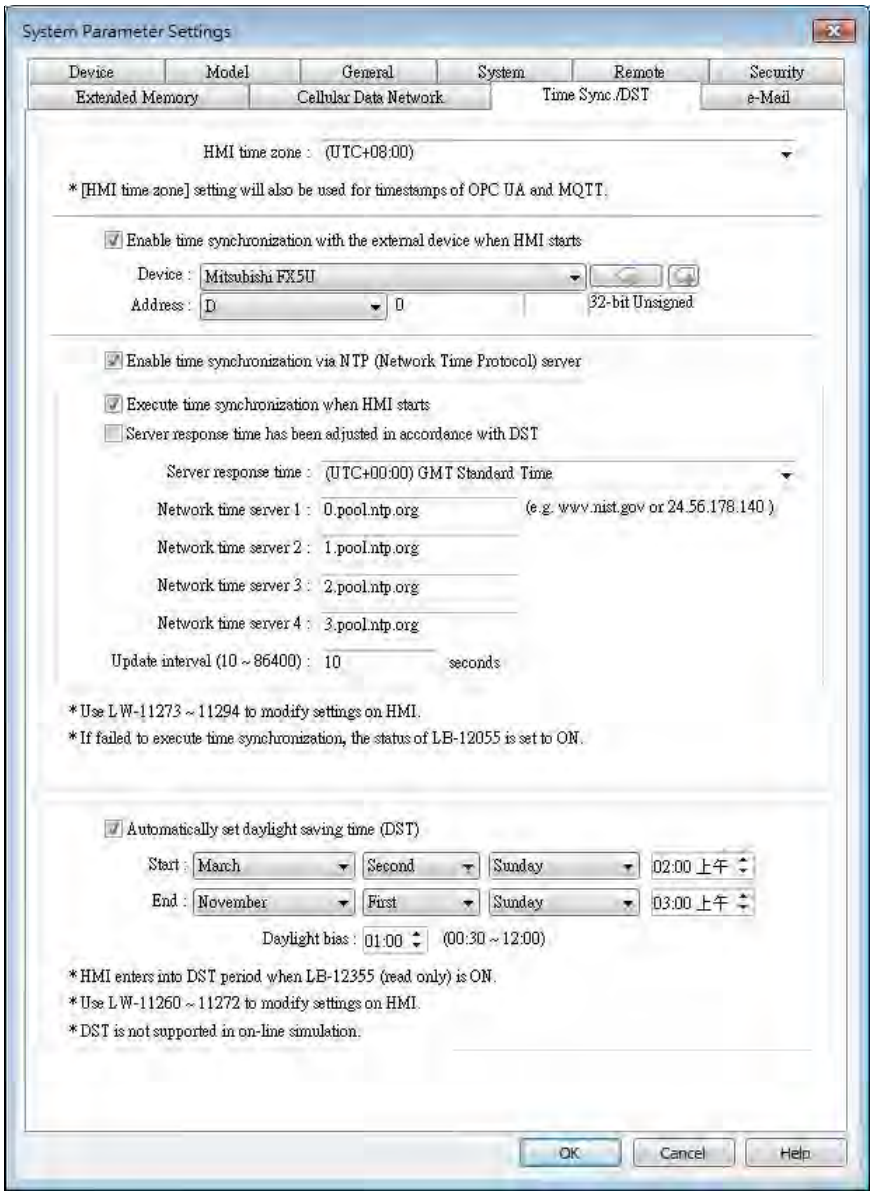
Setting	Description
Output settings	**Orientation** Set how will words or pictures be printed out, [horizontal] or [vertical]. **Printer size** Set to print out in [Original size] or to [Fit to printer margins]. **Margin** Set the top, bottom, right and left margin width.
Communication settings	**IP address** Assign the IP address of the printer via network. **[Port], [User name], [Password]** Specify the data to log in server. Port can be set from 1 to 65535. Maximum length of user name or password is 12 characters.

For more information, see “26 EasyPrinter”.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

5.10. Time Sync./DST

Configure HMI time synchronization and daylight saving time.



Setting	Description
HMI time zone	Select HMI time zone.
Enable time synchronization with the external device when HMI starts	Automatically synchronizes HMI time with the designated source address when HMI starts. As shown in the settings above, the time source is set to D address of the device Mitsubishi FX5U, so 6 consecutive word addresses starting from D-0 will be the source addresses and each of them should contain time information as follows: D-0 → Second (range: 0~59)

	D-1 → Minute (range: 0~59) D-2 → Hour (range: 0~23) D-3 → Day (range: 1~31) D-4 → Month (range: 1~12) D-5 → Year (range: 1970~2037)
Enable time synchronization via NTP (Network Time Protocol) server	**Execute time synchronization when HMI starts** Automatically synchronizes HMI time with the designated NTP server when HMI starts. **Server response time** Select NTP server time zone. **Network time server** Provides four fields to fill in Network Time Servers for user's device. If the synchronization with server 1 fails, the system will try to synchronize with server 2, and so on. If HMI time cannot synchronize with any of the Network Time Servers, the system register LB-12055 will change to ON status. **Update interval** The frequency of synchronization, the range is from 10 to 86400 seconds.
Automatically set daylight saving time (DST)	**Start / End** Set the start/end of Daylight Saving Time. The option [Last] may refer to the 5th or 6th week depending on the selected month. **Daylight bias** Set the time zone's offset during DST.

Note

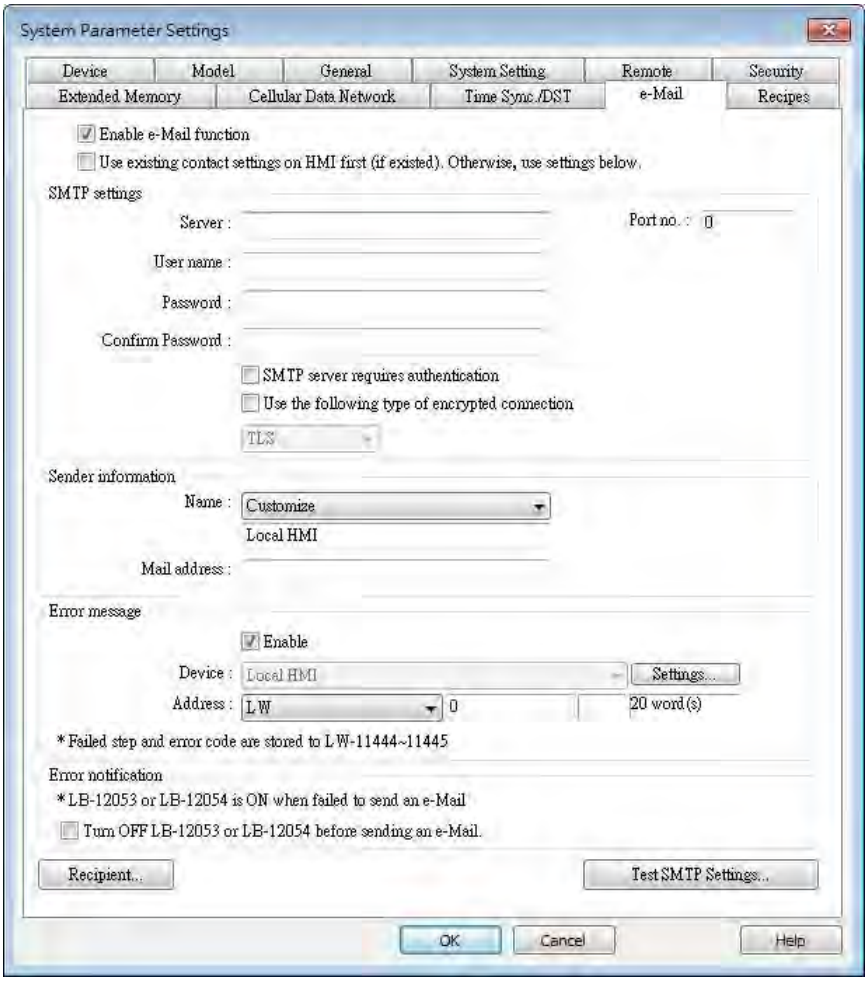
- When DST ends, and the time falls back, the Trend Display object in Real-time mode will stop refreshing for the overlapping hour(s) when it just going back to standard time. However, historical data will not be affected.
- When DST ends, adjusting HMI time (manually or by network) back to the DST period will not be effective to the system. The system register LB-12355 stays OFF.
- When DST ends, and the time falls back, the system will not start DST for the overlapping hour(s). The system register LB-12355 stays OFF.
- Before DST starts, adjusting HMI time (manually or by network) into the DST period will start DST. The system register LB-12355 will be set ON. The HMI time will be the specified time without adding the offset.
- During DST, adjusting HMI time (manually or by network) to a time that is not within DST period will end DST. The system Register LB-12355 will be set OFF. The HMI time will be the specified time without subtracting the offset.

- This feature **does not yet support the DST in Southern Hemisphere**.

For more information on system registers relevant to DST, see “22 System Registers”.

5.11. e-Mail

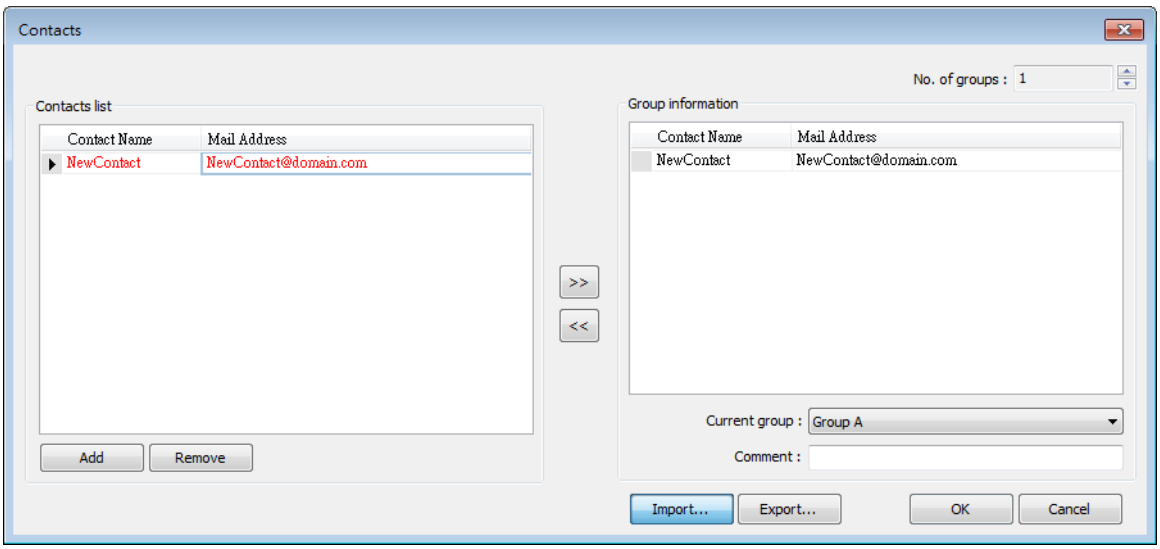
Select [Enable e-Mail function] check box to configure the parameters.
When [Use existing contact settings on HMI] check box is selected, the system will use the contact settings in HMI after downloading the project to HMI, or use the following settings when no contact is found in HMI.



Setting	Description
SMTP Settings	**Server:** Set SMTP Server. **Port:** Set communication port. **User name:** Set e-mail address. **Password:** Set e-mail password. **Confirm Password:** Confirm e-mail password. **SMTP server requires authentication:**

	Decide whether Secure Password Authentication is needed when log in e-mail. **Use the following type of encrypted connection:** Decide whether the encrypted connection (TLS, SSL) is needed when sending e-mail.
Sender Information	**Name:** Enter the sender name or use the local HMI name in the device list, or the name specified using system register LW-10884 (HMI name). **Mail Address:** Setting e-mail address.
Error message	When an error occurs in email delivery, the error message sent from SMTP server can be shown in the designated address. For more information on system registers relevant to mail delivery, see “22 System Registers”.
Error notification	**Turn OFF LB-12053 or LB-12054 before send an e-Mail** When enabled, system registers LB-12053 or LB-12054 will be turned off before sending e-mail, otherwise, when an error occurs in email delivery, the register remains ON.
Test SMTP Settings	Verify SMTP settings in advance by sending a test email to the specified e-mail address.

Click [Recipients] to open the following dialog to edit contacts:



Setting	Description
Contact list	Add or remove contacts from the list.
Group Information	Group up contacts. **No. of groups:** Set no. of contact groups, according to the number, the groups are named from A to P and up to 16 groups can be set. **Current group:** Displays the group that includes the contacts above. **Comment:** Enter a description for the current group.

For more information on sending Event Log, see “7 Event Log”.

For more information on e-mail related system registers, see “22.3.40 e-Mail”.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

5.12. Cellular Data Network

This tab is for setting cellular data. Cellular data network settings may differ from one model to another.

Applicable models:

cMT X Series models

cMT-SVR (OS ver. 20151127 or later)

cMT3071/cMT3072/cMT3090/cMT3103/cMT3151 (OS ver. 20180723 or later)

cMT-HDM/cMT-FHD/cMT-FHDX (OS ver. 20190130 or later)

MT8051iP/MT8071iP/MT8102iP (OS ver. 20201007 or later)

Attaching a 3G/4G USB dongle to the USB port enables the cMT-SVR to connect to 3G/4G networks.

Setting	Description
Use existing contact settings on HMI	If this check box is selected, the cellular data currently in HMI will be used. When no specific settings are required, generally, [PIN code] is “0000”, [APN] is “internet”, and [Username], [Password], [Dial number] are optional.

The relevant registers:

LW-11297: (16 words) : PIN code of SIM card (cellular network)
LW-11313: (16 words) : Access Point Name (cellular network)
LW-11329: (16 words) : username (cellular network)
LW-11345: (16 words) : password (cellular network)
LW-11361: (16 words) : dial number (cellular network)
LW-11377: (16bit) : stop (set 0)/start (set 1) connection (cellular network)
LW-11378: (16bit) : last error code (0:success, 1:incorrect PIN code, 2:no SIM, 3:no device, 4:puk locked, 5:other) (cellular network)
LW-11379: (16bit) : connection status (0:no device, 1:disconnect, 2:connecting, 3:connected) (cellular network)

USB Tethering

By connecting an Android® mobile phone with HMI using a micro USB cable, or any USB data cable that is compatible with the phone, the internet connection of the phone can be shared with HMI. On the Android mobile phone please enable USB Tethering.

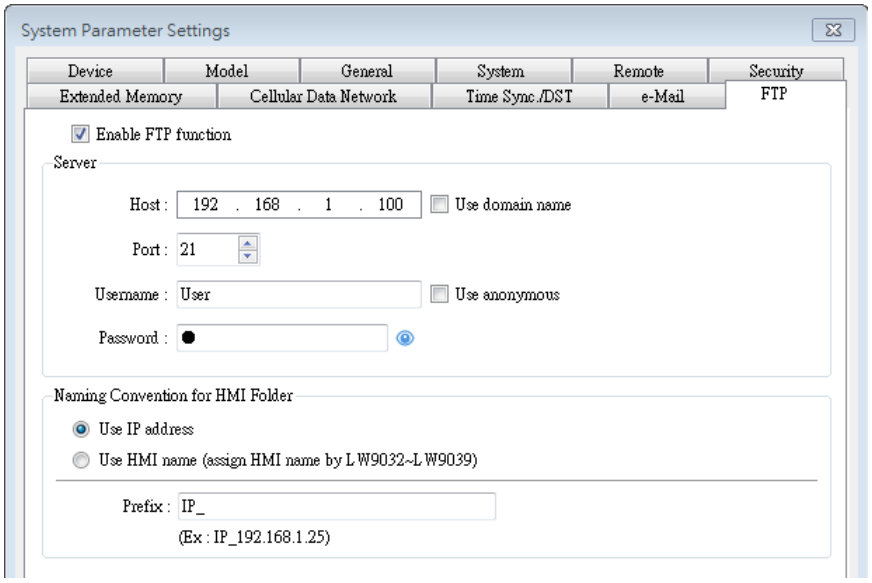
When successfully connected, the following registers can be used to monitor and control the connection status.

LW-11380:	(16bit) : stop (set 0)/start (set 1) connection (USB tethering)
LW-11381:	(16bit) : connection status (0:no device, 1:disconnect, 2:connected, 3:fail, 4:OS not support, 5:HMI not support) (USB tethering)

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

5.13. FTP

This tab is for configuring FTP connection parameters. With [Enable FTP function] selected, files can be backed up to FTP server from HMI. This feature is only supported on cMT / cMT X Series models.



Setting	Description
Host	FTP server's IP address or domain name.
Port	FTP server's port number.
Username	FTP server's username.
Password	FTP server's password.
Naming Convention for HMI Folder	Use IP address or HMI name to name the HMI folder. **Prefix:** Add a prefix to the beginning of the folder name.

6. Window Operations

This chapter describes different types of windows and how to create, set and delete a window.

6.1. Overview 6-2
6.2. Window Types 6-2
6.3. Create, Set, and Delete a Window 6-4
6.4. Window Transparency 6-8
6.5. Editing Tips 6-10

6.1. Overview

A window is a basic element in a project. With a window, all kinds of information like objects, pictures, and texts can be displayed on HMI screen. In total, 1997 windows numbered from 3 ~ 1999 in EasyBuilder Pro can be built and edited.

6.2. Window Types

There are 4 types of windows, each with different functions and usages:

- Base Window
- Fast Selection Window
- Common Window
- System Message Window

6.2.1. Base Window

The most frequently used window, except for main screen, it can also be:

- A background of other windows.
- A keyboard window.
- A pop-up window of Function Key object.
- A pop-up window of Direct Window and Indirect Window objects.
- A screen saver.

Note

- Base Window should be in same size as the HMI screen. Therefore, the resolution of the base window should be set to the resolution of HMI.

6.2.2. Fast Selection Window

Window no. 3 is the default Fast Selection Window. This window can co-exist with base window. Generally, it is used to place the frequently-used buttons on the lower-left side or the lower-right side on the screen. Please create window no. 3 first, and set the relevant properties in [System Parameter Settings] » [General] tab. Apart from showing or hiding fast selection window with the button on the screen, there are system registers to do so:

[LB-9013] hide (set ON)/show (set OFF) Fast Selection window

[LB-9014] hide (set ON)/show (set OFF) Fast Selection button

[LB-9015] hide (set ON)/show (set OFF) Fast Selection window/button

Note

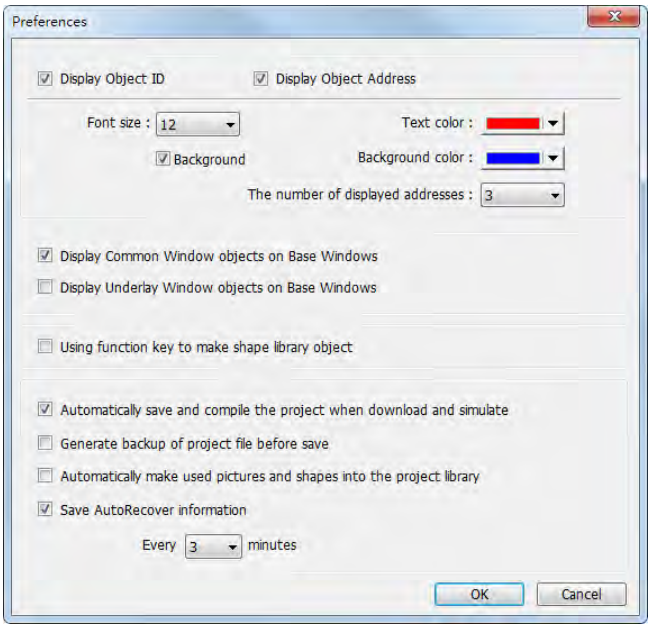
- cMT / cMT X Series does not support Fast Selection Window.

6.2.3. Common Window

Window no. 4 is the default Common Window. Objects in this window will be displayed in other base windows, not including pop-up windows. Therefore, the common objects in different windows are often placed in common window.

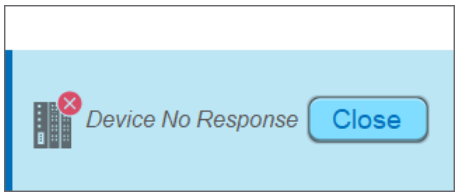
When operating HMI, select [Function Key] » [Change common window] to change the source of common window.

In menu [File] » [Preferences] » [Display] select whether or not to [Display Common Window objects on Base Windows] when editing a project. This can avoid overlapping objects in base window with objects in common window.



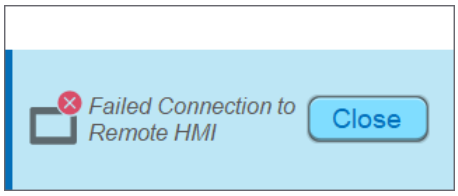
6.2.4. System Message Window

Windows No. 5, 6, 7, 8 are the default System Message Windows:



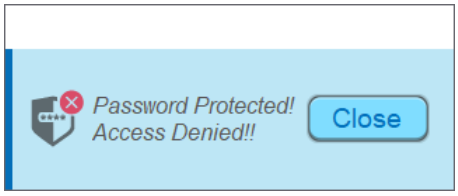
Window No. 5: Device No Response

When the communication between device and HMI is disconnected, this message window will pop up automatically right on the base window currently opened. This window can be disabled by system registers.



Window No. 6: HMI Connection

When failing to connect HMI with a remote HMI, this message window will pop up automatically.



Window No. 7: Password Restriction

When attempting to control an object without authorization, this window may pop up as a warning depending on the settings of the object.

Window No.8: Storage Space Insufficient
When HMI flash memory, USB disk or SD card run out of storage space, this message window will pop up automatically. (When the memory space is under 4 MB)

The following system registers can be used to check the free memory space in HMI, USB disk, or SD card:
[LW-9072] HMI current free space (K bytes)
[LW-9074] SD current free space (K bytes)
[LW-9076] USB disk current free space (K bytes)
To check if there is sufficient storage in the devices, the following system registers can be used. These addresses will set ON when the space is under 4 MB.
[LB-9035] HMI free space insufficiency alarm (when ON)
[LB-9036] SD card free space insufficiency alarm (when ON)
[LB-9037] USB disk free space insufficiency alarm (when ON)

For more information, see “22 System Registers”.

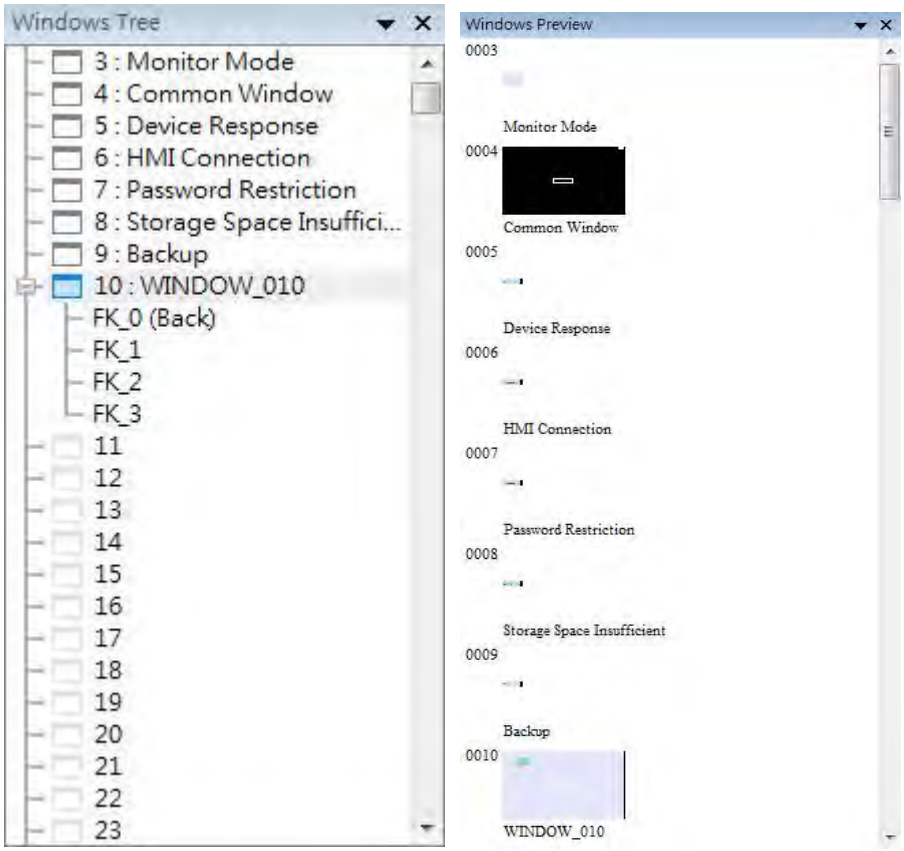
The text shown in windows no. 5 ~ 8 can be edited for easier reference.

Note

- For eMT/iE/XE/mTV models, a screen can display up to 24 pop-up windows simultaneous including System Message Window, Direct Window and Indirect Window. For cMT/cMT X Series, the number of pop-up windows is not limited.
- The system does not allow opening the same window with two Direct (or Indirect) Windows in one base window.
- Windows no. 3 to 9 are used by the system only, and windows no. 10 to 1999 can be edited based on actual usage.

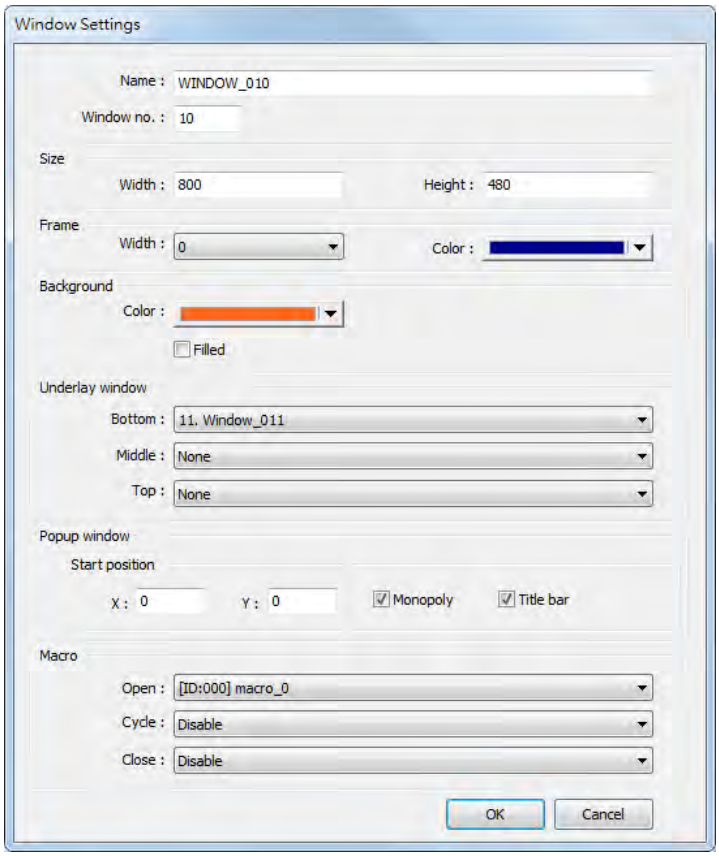
6.3. Create, Set, and Delete a Window

Check the existing windows in [Windows Tree].
[Windows Tree] displays window numbers and window names. Opened windows are marked with icon. Press the (+) sign to see the object ID, address and description of the objects used in this window.
[Window preview] displays the thumbnails of windows.



6.3.1. Creating and Setting a Window

In windows tree right click on a window number then select [New].



Setting	Description
Name	The name appears on the title bar and also in windows tree.
Window no.	Can be 3 to 1999.
Size	Set the window size in accordance with the HMI resolution.
Underlay window	Underlay Window can be seen as an extra Common Window. When designing the project, some commonly used objects are used in some windows but not all. These objects can be placed in underlay window. Each base window can set three underlay windows as background, from [Bottom] to [Top]. The objects in underlay windows are displayed in this order in base window.
Pop-up window	Base window can also be used as a pop-up window. Use [X] and [Y] to set the coordinates indicating where in the screen will this base window pop up. The origin of the coordinates is the upper-left corner of the window.
Monopoly	If the option is selected, when the base window pops up, no operations of other pop-up windows and background windows are allowed until the monopoly window is closed. If a base window is used as a keyboard window, “Monopoly" is automatically enabled.
Title bar	If the option is selected, a title bar appears on a system message window. This feature is only available for system

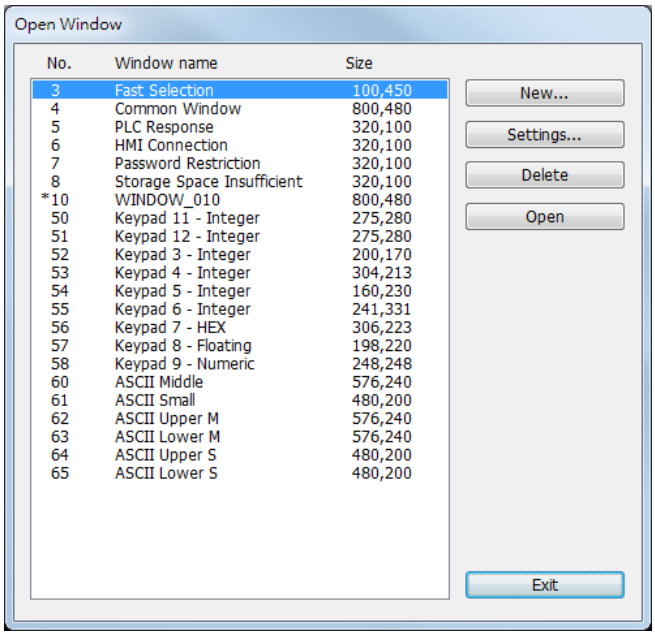
	message windows no. 5 to no. 8.
Macro	Select the macro to be executed when the window opens or closes, or the macro to be executed periodically (each 500 ms). Please build the macro in advance.

Note

- The objects in underlay window cannot be edited from the base window that displays them. To edit those objects, please open the underlay window where they are located.
- When the window number of the underlay window used by the base window is identical to the pop-up window, the pop-up window is disabled.
- When base window and pop-up window use the same underlay window, the objects in the underlay window cannot be displayed in pop-up window.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

Or, from the main menu click [View] » [Open Window] and then click [New] and select the type of the window and click [OK].



Ways to call up [Window Settings] dialog:

- Right click on the window number in the windows tree and select [Settings].
- In [View] » [Open Window] select the window then click [Settings].
- In the window, right click when no object is selected, and select [Attribute].

6.3.2. Open, Close, and Delete a Window

The ways to open an existing window:

- Double click on the window number in the windows tree.

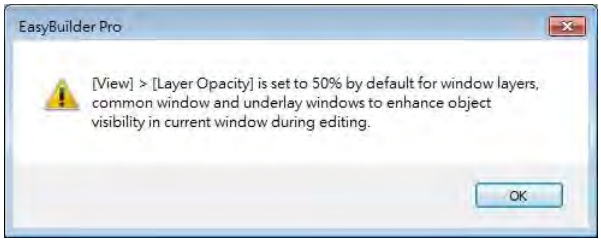
- In the windows tree, select the window number, right click, and then select [Open].
- In [View] » [Open Window] select the window number and then click [Open].

The ways to close or delete an existing window:

- In the windows tree, select the window number; right click, and then select [Close] or [Delete].
- In [View] » [Open Window] select the window number and then click [Delete].

6.4. Window Transparency

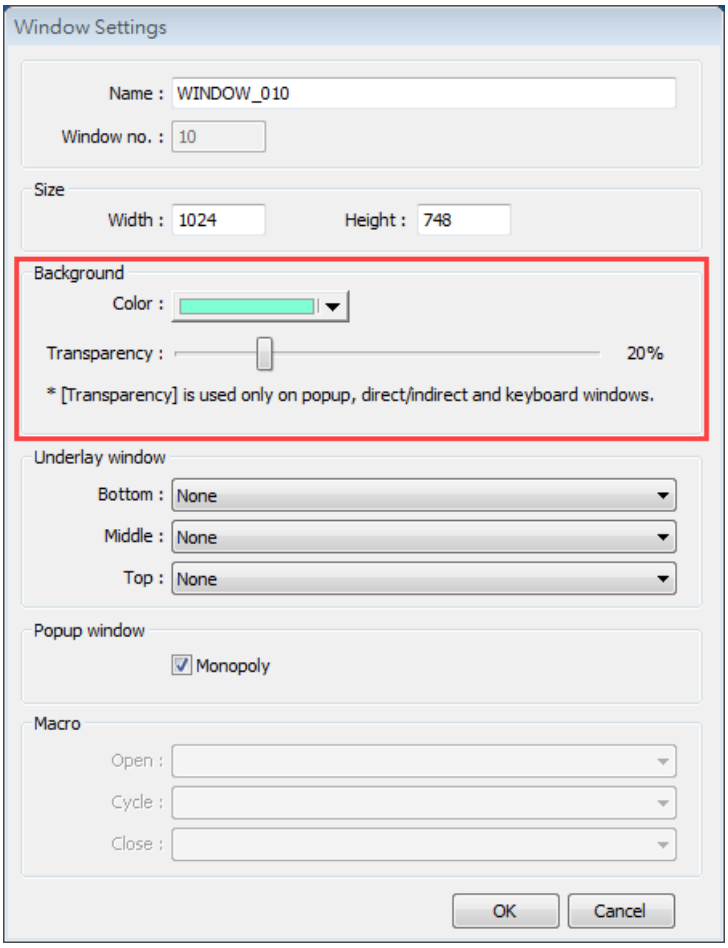
The Opacity setting in the View tab is used to adjust transparency effect of the common window, overlay windows, and window layers.



As shown in the following screenshots, greater Opacity makes the object in the relevant window more visible.

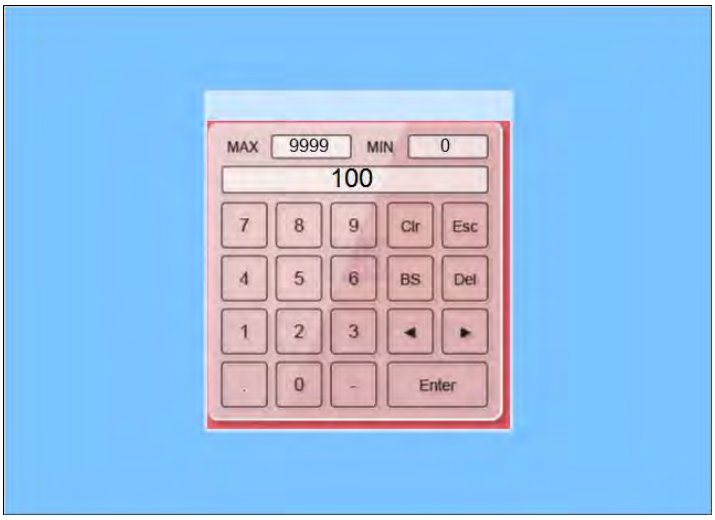
Additionally, the cMT / cMT X Series models support setting window transparency. This setting is applicable for popup windows, Direct / Indirect Windows, and keyboard windows. The settings dialog box and the effect are shown in the following screenshots. Greater transparency

can make the object on the base window more visible when it is covered by the popup window.

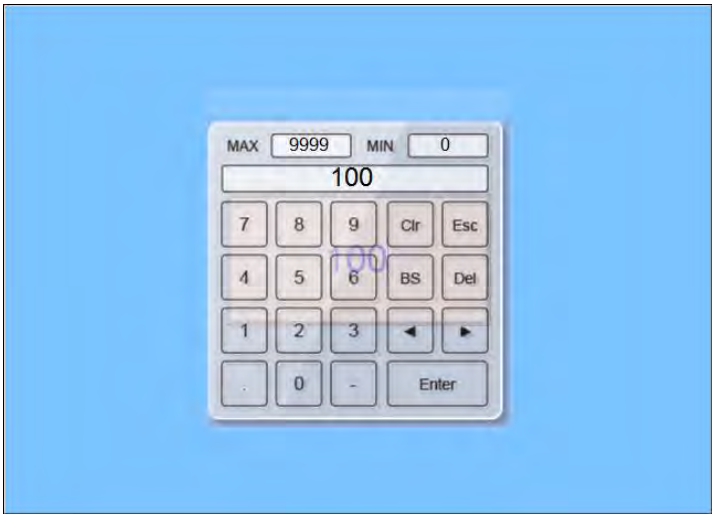


Example

Setting transparency to 40%:



Setting transparency to 90%:

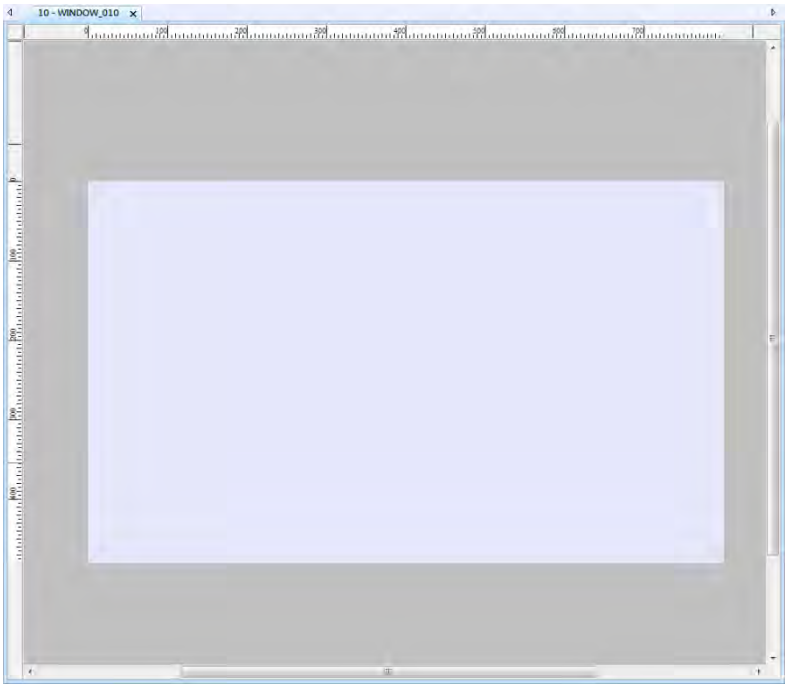


6.5. Editing Tips

EasyBuilder Pro provides a user-friendly interface with which users can easily edit projects. The following are some editing tips.

6.5.1. Non-display Area, Select Tool, and Hand Tool

As shown in the following screenshot, two areas can be found in the editing window: editing area and non-display area. The non-display area refers to the area outside the editing area (the one in the middle). Objects placed in the non-display area will not be visible on HMI, and communication with devices continues. The invisible objects such as Meter, Time-based Data Transfer, or Text objects used for comments can be placed in this area to avoid overlapping with other objects in the editing area, which adds difficulty to project editing. The objects placed in the non-display area remain effective.



Select Tool

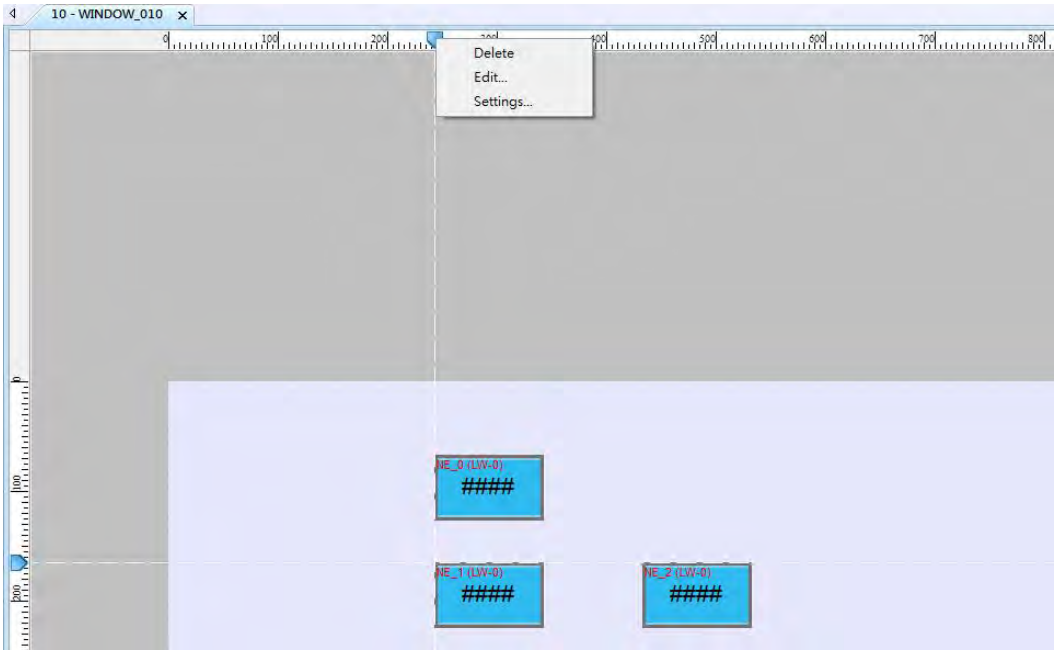
Clicking this icon in the toolbar changes the cursor to the Select Tool. Select a single object by clicking it, or select multiple objects by clicking on an empty part of the editing screen and then drag a box around the objects.

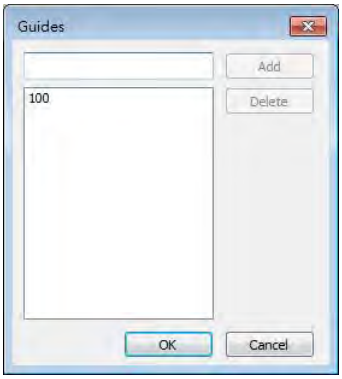
Hand Tool

Clicking this icon in the toolbar changes the cursor to the Hand Tool. The Hand Tool allows moving through the project screen by simply left clicking the mouse and dragging anywhere.

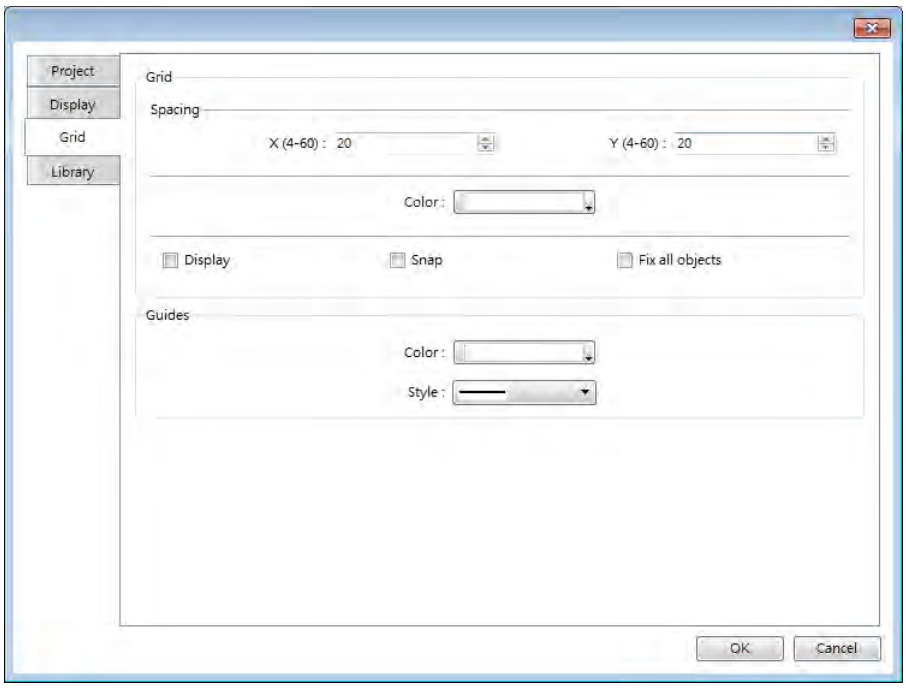
6.5.2. Ruler

Selecting [View] » [Ruler] adds rulers to the top and left of the editing screen. Guides can be added from the ruler by right click on the ruler. Objects will automatically snap to the guides when moved or resized.



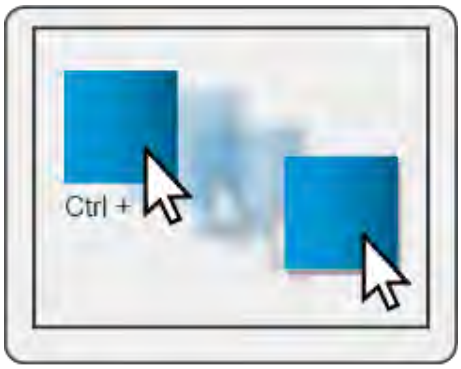
Setting	Description
Delete	Delete the guide.
Edit	Adjust the position of the guide.
Settings	 Add or delete multiple guides.

Selecting [File] » [Preferences] » [Grid] opens the following window for setting the color or style of the Guides.



6.5.3. Quick Copy

Quick copy can be carried out by pressing and holding Ctrl key on the keyboard while an object is selected, and then drag the cursor to a new location where the object will be created.



7. Event Log

This chapter explains how to set and use Event Log.

7.1. Overview 7-2
7.2. Event Log Management 7-2
7.3. Creating a New Event Log 7-9

7.1. Overview

The following are the basic steps to use Event Log:

1. Define event content and trigger condition.
2. Trigger event according to the condition.
3. Save the event log to the specified device.
4. View the process of event by using the relevant objects.

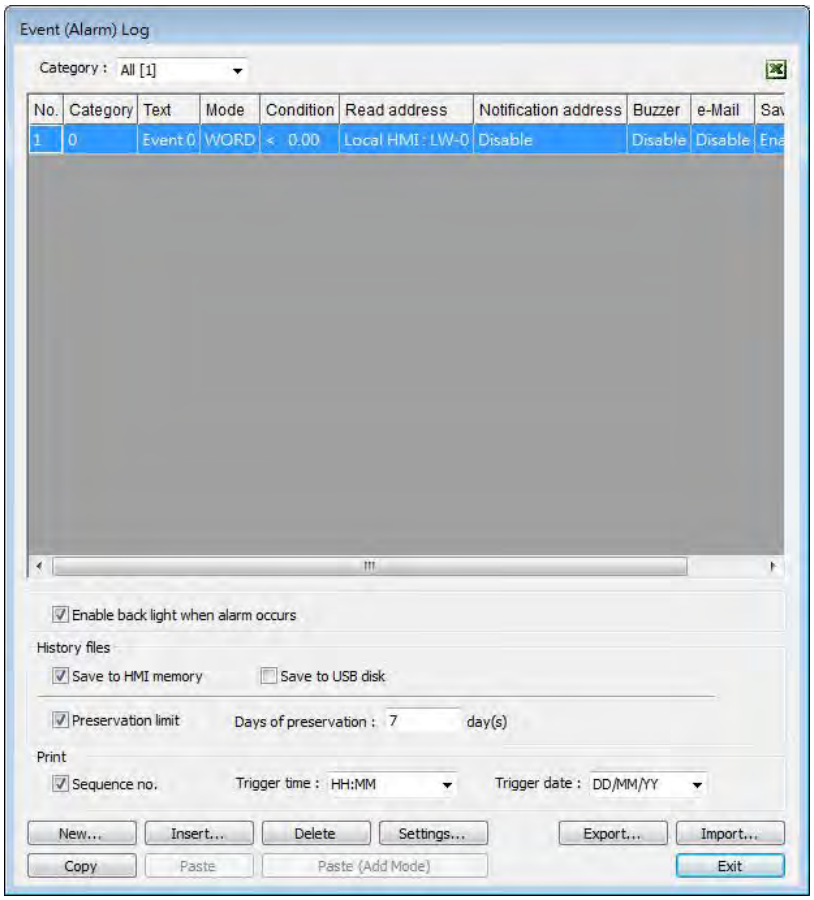
This chapter will explain how to set and use Event Log.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

7.2. Event Log Management

Firstly, define the event content then use Alarm Bar , Alarm Display , Event Display objects to view the process of the whole event from triggering→waiting to be processed→ return to normal. The upper limit for the number of event logs is 10000.

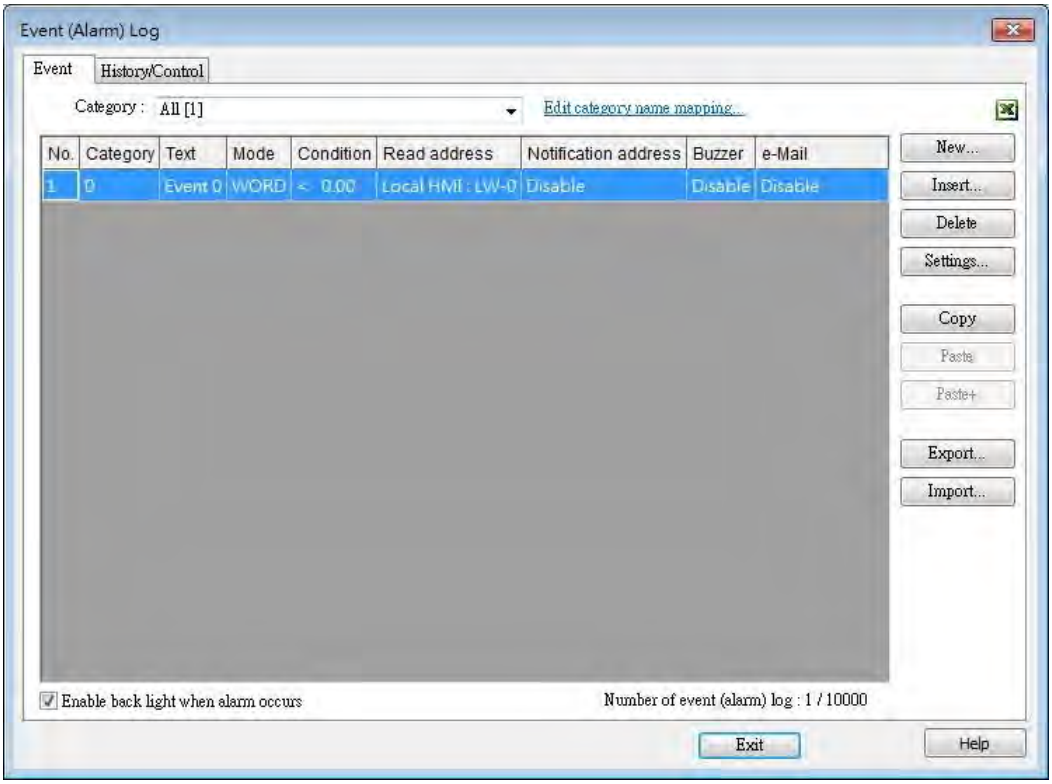
7.2.1. eMT, iE, XE, mTV, iP Series



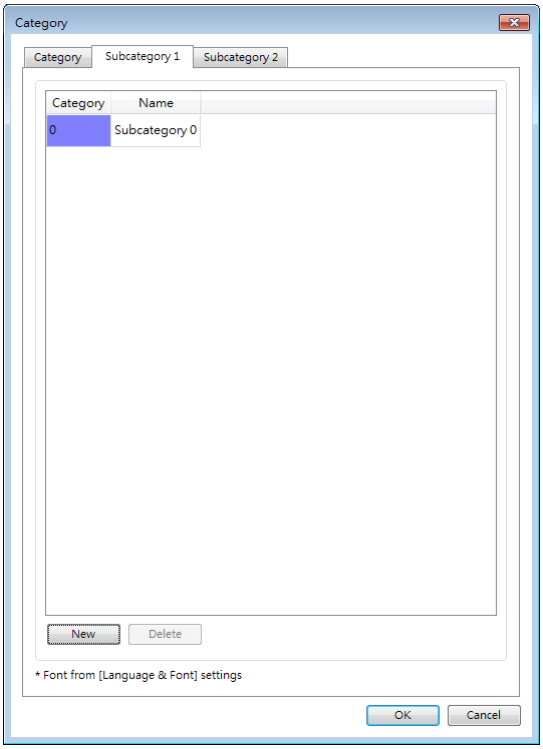
Setting	Description
Category	Classifies events by dividing them into 0 ~ 255 categories. Select one category to add or view event log. In the bracket "[ ]", it shows the number of events are in this category.
History files	Saves event log files to the specified location. Once an event occurs, the HMI immediately saves the history file. When executing On-line or Off-line Simulation on PC, the files will be saved in the HMI_memory / SD_card / USB folder under the installation directory. **Preservation limit** This setting determines the maximum number of Event Log files to be preserved in HMI memory. This does not include the file generated today. That is, if [Days of preservation] is set to 2; the two latest files excluding the file generated today will be kept. The files that are not within the range will be deleted automatically for saving the storage space.

Print	In [System Parameter Settings] » [Model], select a printer and set the printing format.
Copy	Copy the selected item.
Paste	Overwrites the selected item with the new items. A message window will pop up to confirm this operation.
Paste (Add Mode)	Appends as a new entry.

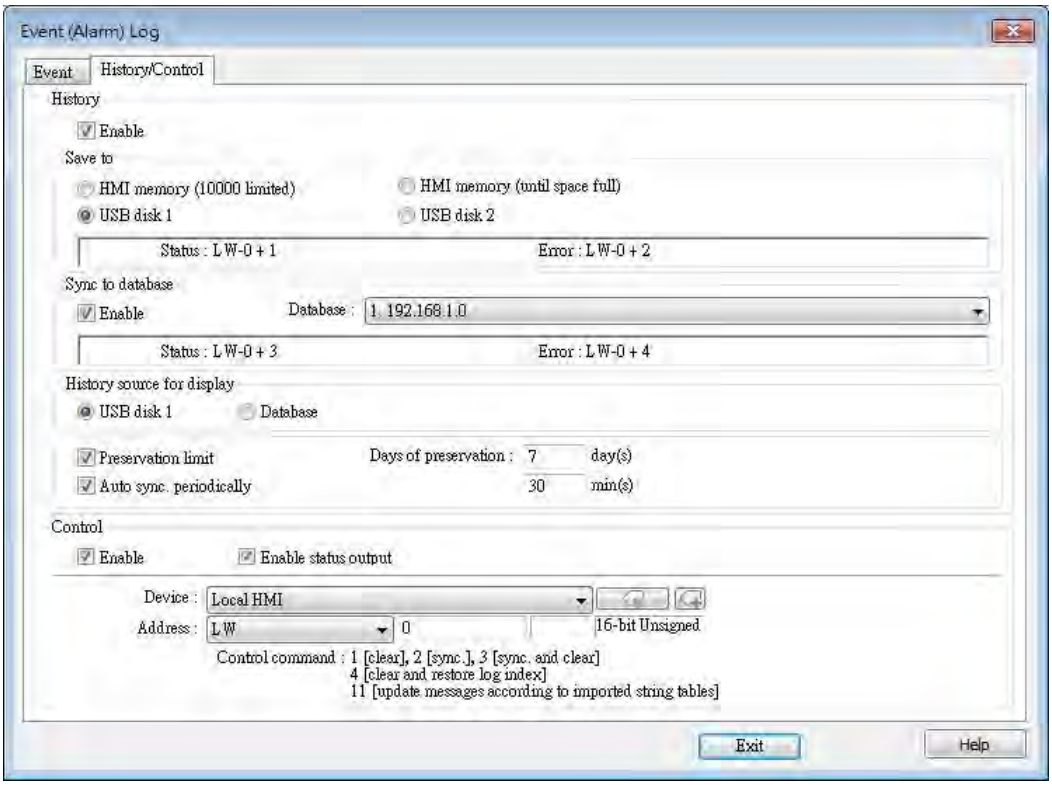
7.2.2. cMT, cMT X Series



Setting	Description
Category	Classifies events by dividing them into 0 ~ 255 categories. Select one category to add or view event log. The number of events in this category is shown in the bracket "[ ]". **Edit category name mapping…** Opens a category name table which allows editing corresponding category names. Subcategories can also be added in this table in addition to the exiting categories. The maximum allowable number of entries in each subcategory is 256 (0~255) entries.



Copy	Copies the selected item.
Paste	Overwrites the selected items with the clipboard contents. A message window will pop up to confirm this operation.
Paste +	Appends the clipboard contents to the end of the list.



Setting	Description
History	Saves the Event Log data to HMI memory (10000 limited or until space full), SD card, USB disk, or database (by synchronizing Event Log file to database server). The rules of saving the data are: ■ When [HMI memory (10000 limited)] is selected, and the number of events reaches 10000, the system will delete the earliest 1000 events on HMI and keep on saving the data to HMI memory. . ■ When [HMI memory (until space full)] is selected, the system will keep on saving the data to HMI memory without deleting the earlier events. In this case the data may not be synchronized to database. ■ When USB disk or SD card is selected, and the number of events reaches 10000, the system will automatically save the data to the external device and delete the earliest 1000 events on HMI. ■ To synchronize the data to database server, select a database that has been configured before. ■ If the external device already contains some events, the new data is appended without overwriting the original data each time in synchronization. ■ When the external device is removed from HMI, or HMI is disconnected from database server, if the connection is recovered before the number of events exceeds 9000, the events occur during disconnection will be saved to HMI. If the number of events exceeds 9000 during disconnection, the earlier data will be deleted, and cannot be synchronized by recovering connection.
History source for display	Event Log displays the history data read from the designated source.
Preservation limit	This setting determines the maximum number of Event Log files to be preserved in HMI memory. This does not include the file generated today. That is, if [Days of preservation] is set to 2; the two latest files excluding the file generated today will be kept. The files that are not within the range will be deleted automatically. The system checks the time of the files

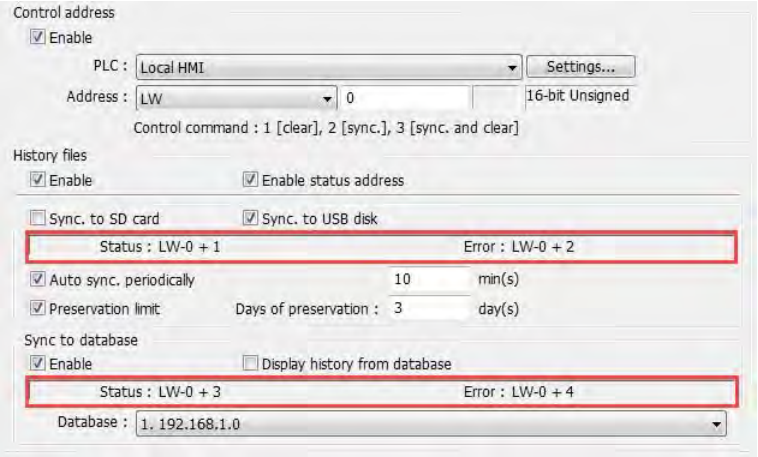
	and deletes earlier files only during synchronization.
Auto sync. periodically	Data will be automatically synchronized to the designated external device in the specified time interval, regardless of the rules explained above. Unit: Minute(s) Range: 1~1440
Control address	If select [Enable] check boxes under both [Control address] and [History files], entering a specific value in the control address sends the corresponding command.

Value	Command
1	Clear the event log on HMI
2	Synchronize event log to the external device
3	Synchronize event log to the external device and then clear the event log on HMI
4	Use the history data stored in USB disk / SD card / database after changing HMI
6	Free unused storage space reserved for event logs (*.db). This command can be used when historical data is saved to HMI memory (until space full).
11	Update event log message contents by reading a new String Table

If none of these values is entered, the system will synchronize data in the same rules as [History] setting.

Status Error

When LW-n is used as the control address, the four consecutive addresses following LW-n (LW-n+1~LW-n+4) will show status and error, please see the prompt in the settings dialog box.



Value	Status address: LW-n+1 and LW-n+3
0	Disconnected from external device or database
1	Connecting with external device or database
2	Connected with external device or database

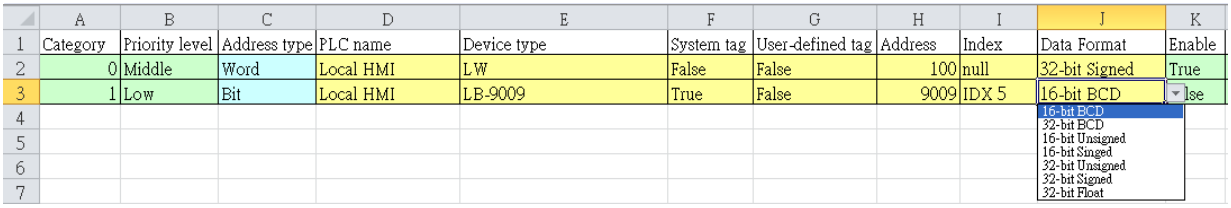
3	Storing records into the archive. When this is done, the value returns to 2.
Value	Error address: LW-n+2 and LW-n+4
0	None
1	Unknown error
2	Failed to connect with external device or database
3	Access denied
4	Wrong database name
5	Inconsistent data format
6	Failed to open table
7	Failed to create table
8	Failed to write table
9	Failed to open database
10	Database is corrupted

Note

- Before removing SD card / USB disk, or disconnecting from database server, please synchronize event log data by using control address.
- When monitoring multiple HMIs by using cMT Viewer, the Start Button on the current HMI screen may flash to indicate that an error has occurred on another HMI that is also being monitored, reminding the user to switch HMI and check the event.

7.2.3. Excel Editing

Click on the Excel icon in Event Log setting dialog box to open the Excel template for a reference of editing. This template is under the installation directory, the file name is EventLogExample.xls. This template includes the ready-made dropdown lists and validation mechanism.



	A	B	C	D	E	F	G	H	I	J	K
1	Category	Priority level	Address type	PLC name	Device type	System tag	User-defined tag	Address	Index	Data Format	Enable
2	0	Middle	Word	Local HMI	LW	False	False	100	null	32-bit Signed	True
3	1	Low	Bit	Local HMI	LB-9009	True	False	9009	IDX 5	16-bit BCD	lse
4										16-bit BCD	
5										32-bit BCD	
6										16-bit Unsigned	
7										16-bit Singed	
										32-bit Unsigned	
										32-bit Signed	
										32-bit Float	

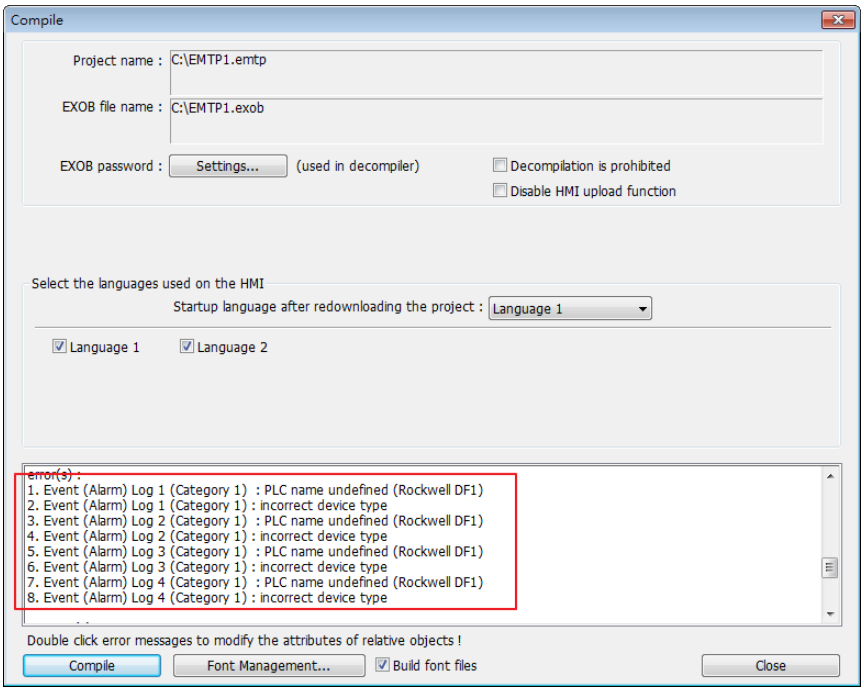
Note

- [System tag] and [User-defined tag] cannot be set to true simultaneously, otherwise, the system will view the User-defined tag to be a System tag, and [User-defined tag] to be false. If setting [Device type] to [User-defined tag], please set [System tag] to false.
- When setting [User-defined tag] to true, if the system compares the [Device type] with the user-defined tag in the system, and no suitable tag is found, the system will set the [User-defined tag] in event log to false
- [Color] format is R:G:B, each should be an integer form 0 to 255.

- Before importing Label Library / Sound Library, please make sure the library names exist in the system.

7.2.4. Quick View of Errors

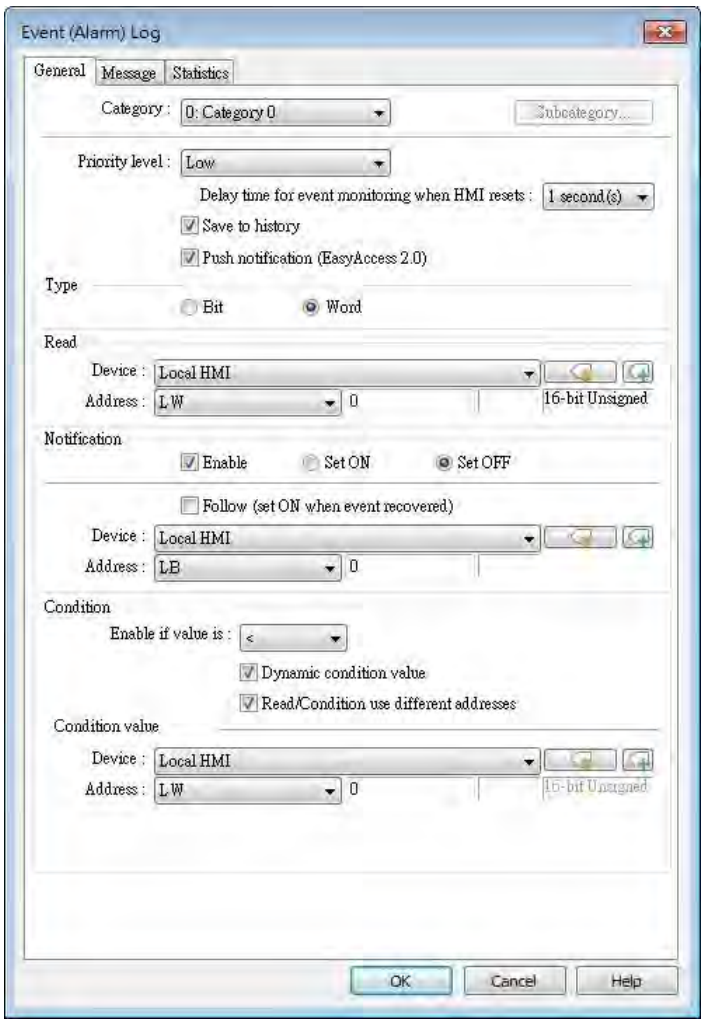
When compiling the project, the errors in Event Log will be displayed in the Compile window. To open Event Log and view the errors, double click on the item in the Compile window.

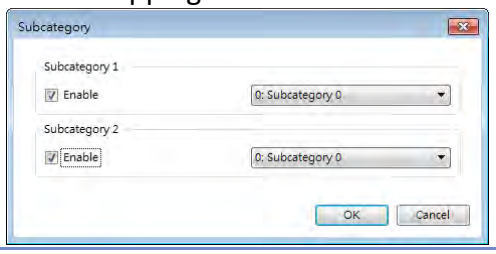


7.3. Creating a New Event Log

General Tab

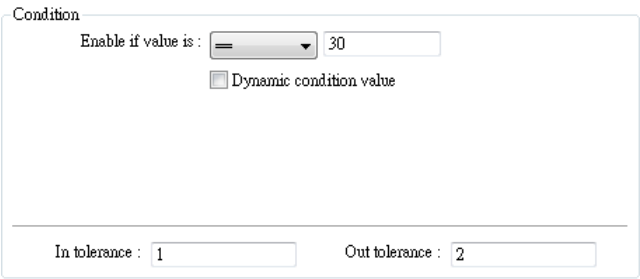
Click [New] in the [Event (Alarm) Log] dialog box.



Setting	Description
Category	Select event category, the range is from 0 to 255.
Subcategory	Add this Event Log into the subcategories in the category name mapping table. 
Priority level	Events in Alarm Bar / Alarm Display are ordered by priority level first and then by time.
Save to history	In Event Log main settings, if [Save to HMI memory] check box under [History files] group box is selected, selecting [Save to history] here determines whether each separate event should be saved as historical file.
Push	When an event occurs, EasyAccess 2.0 push notifications can be sent to iOS®/Android® devices.

notification (EasyAccess 2.0)	
Delay time for event monitoring when HMI resets	This feature is used to set the delay time of Event Log after HMI reboot, in order to avoid false alarm that occurs upon HMI reboot due to uninitialized values. This feature is often used with [Dynamic condition value]. The delay time only occurs once upon HMI reboot.
Read address	The system reads data from this address to check if the event matches the trigger condition.
Notification	When enabled, the system will set the specified address ON or OFF when the event is triggered. **Follow** The notification bit will reset to its original state once the alarm condition returns to normal. For example, when the alarm is triggered, the state of the notification bit turns ON. When the alarm condition returns to normal, and [Follow] check box is selected, the state of the notification bit turns OFF.
Condition	When [Bit] is selected, Event Log will detect the state of a Bit address. When [Word] is selected, Event Log will detect the value of a Word address to check if it is greater than, less than, or equals to a specified value. See Example 1 and Example 2. **Dynamic condition value** Allows online change of the comparison value for trigger condition when the condition is a Word address type. If [Read/Condition use different addresses] is not selected, the source of condition value will be the next consecutive address from [Read address]. **Read/Condition use different addresses** Allows selecting the Word address type to be the source of condition value.

Example 1



The setting above indicates:

When [Read address] value is greater than or equals to 29 (= 30 - 1)

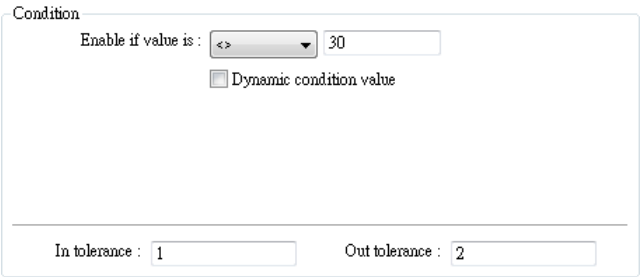
Or less than or equals to 31 (= 30 + 1), the event will be triggered. The trigger condition:

29 ≤ [Read address] value ≤ 31

After the event is triggered, when [Read address] value is greater than 32 (= 30 + 2) or less than 28 (= 30 - 2) the system will return to normal condition:

[Read address] value < 28 or [Read address] value > 32

Example 2



The setting above indicates:

When [Read address] value is less than 29 (= 30 - 1)

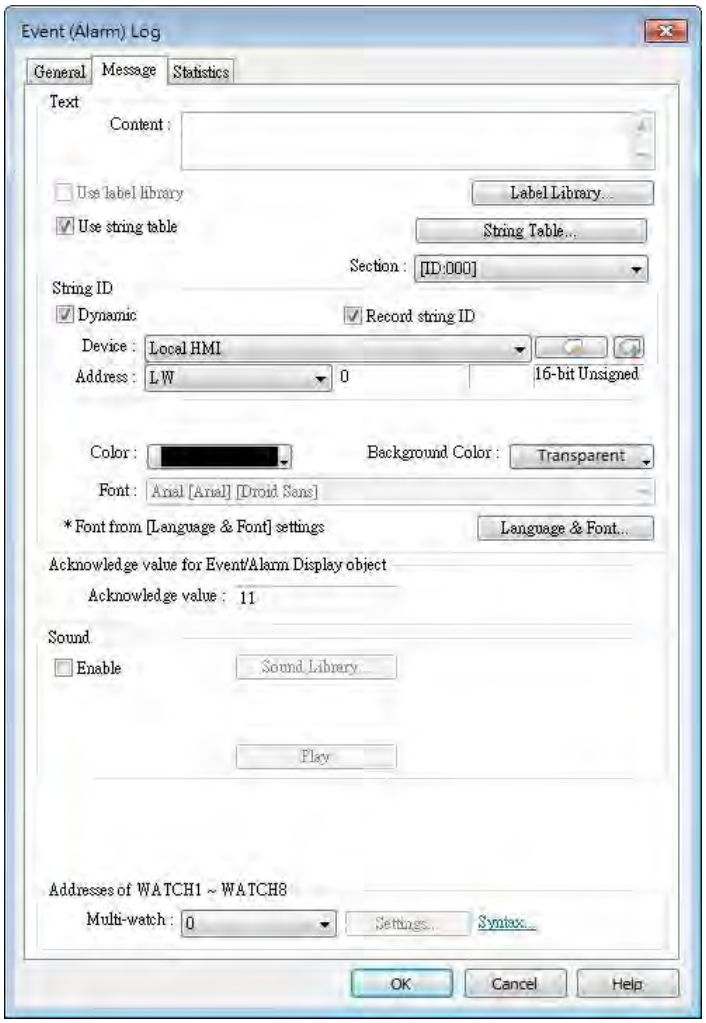
or greater than 31 (= 30 + 1), the event will be triggered. The trigger condition:

[Read address] value < 29 or [Read address] value > 31

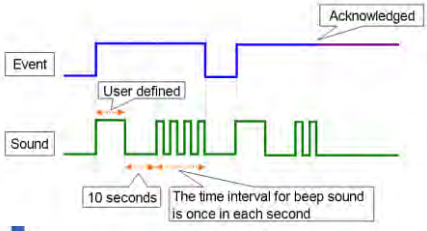
After the event is triggered, when [Read address] value is greater than or equals to 28 (= 30 - 2) or less than or equals to 32 (= 30 + 2) the system will return to normal condition:

28 ≤ [Read address] value ≤ 32

Message Tab

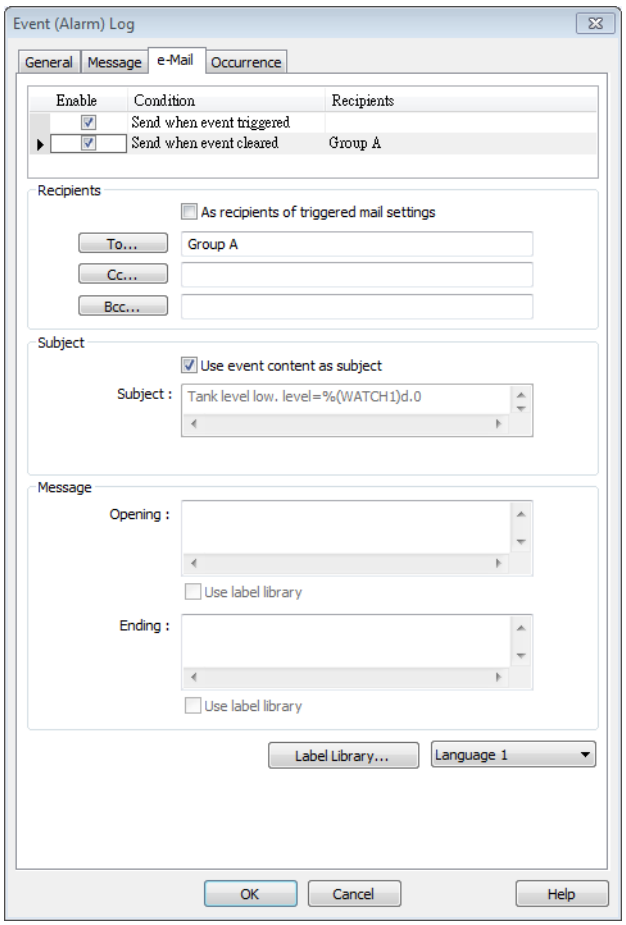


Setting	Description
Content	The text content displayed in [Alarm Bar], [Alarm Display], and [Event Display] objects. Use the formats in the following two examples of WATCH addresses to use register data in content. The content in Label Tag Library and String Table can be used in the Event Log message.
String ID	[Event Display] and [Alarm Display] objects display messages according to the String ID in the designated register. **Record String ID** With this option selected, when an event occurs, the string's ID number will be recorded. Message of the event triggered before can be retained.
Font / Color / Background	The font / color / background color can be set respectively

color	for each event. The font and color settings determine how [Alarm Bar] shows the text, while The font, color, and background color settings determine how [Alarm Display] and [Event Display] show the text. These settings are not available in the [Event Display] under History mode.
Write value for Event/Alarm Display object	When an event in [Event Display] or [Alarm Display] is acknowledged, the value is written to the assigned [Write address]. Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.
Sound	If enabled, the selected sound will be played when an event is triggered. Continuous beep can also be enabled, which only stops when the event is acknowledged or recovered. For continuous beep, a delay time can be set between triggering the alarm and the start of beeping.  Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.
Address of WATCH 1 ~ 8	Users can set how the value is displayed in the designated watch address when an event occurs. Up to 8 watch addresses can be used simultaneously. Click [Syntax] to see how to use the syntax to embed device data in the content of an event log displayed in the watch address. Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

e-Mail Tab

Please enable this function in [System Parameter Settings] » [e-Mail] first.



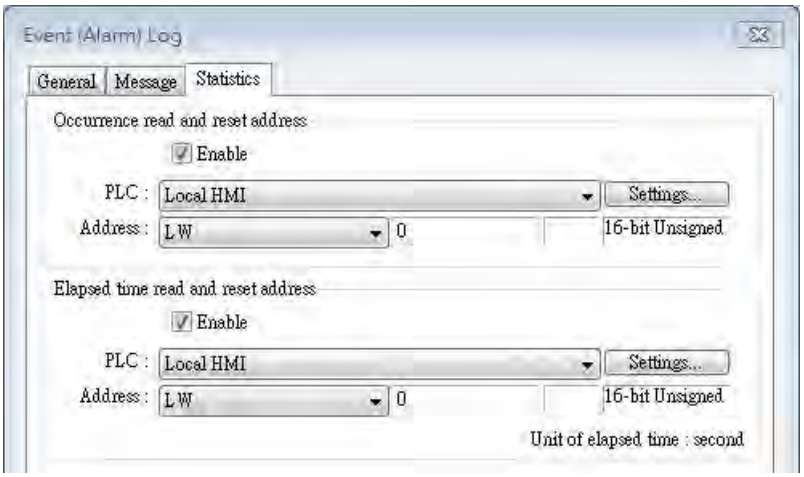
Setting	Description
Recipients	Select the [To], [Cc], and [Bcc] recipients.
Subject	Enter the subject of the e-mail.
Message	Enter the [Opening] and [Ending] content of an e-mail.
Attach	If the [Contains a screenshot of window] check box is selected, the screenshot of the selected window will be attached.

Note

- The priority level of an event determines its importance in e-mail delivery.

Event Priority Level	Email Importance
Emergency	High
High / Normal	Normal
Low	Low

Statistics



Setting	Description
Occurrence read and reset address	If enabled, the number of occurrence of the event after HMI startup will be written to the designated word address. The word address can be read / written.
Elapsed time read and reset address	If enabled, from an event occurs to its recovery, the elapsed time (in seconds) will be written to the designated word address. The word address can be read / written.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

8. Data Sampling

This chapter explains how to set and use Data Sampling.

8.1. Overview 8-2
8.2. Data Sampling Management 8-2
8.3. Creating a New Data Sampling 8-2
8.4. Synchronizing cMT Viewer data and Saving to External Device 8-13
8.5. Checking History Data of a Specific Date on cMT Viewer. 8-13

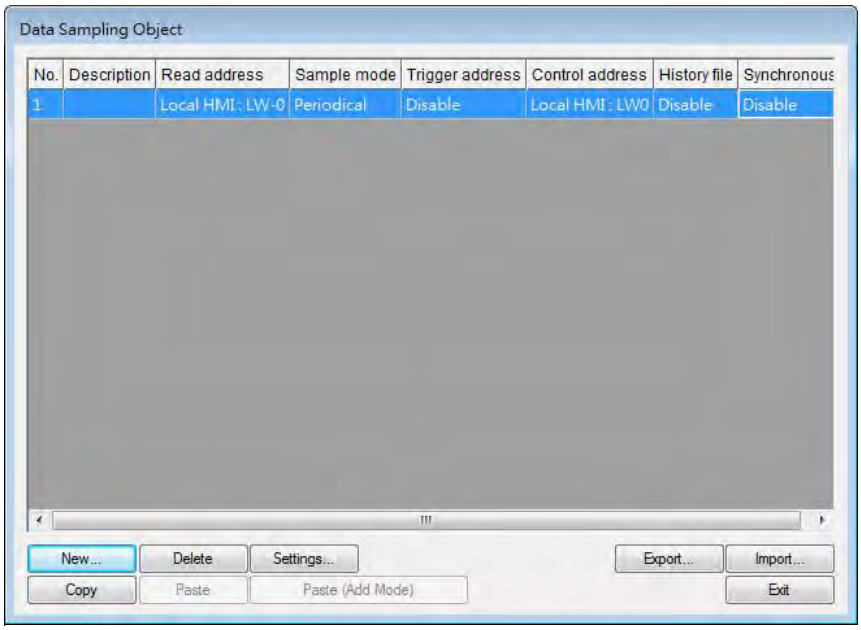
8.1. Overview

After defining how the data is sampled, by sampling time, address, or data length, the sampled data can be saved to the designated location, such as HMI memory, SD card, or USB disk. Trend Display and History Data Display objects can be used to display sampling records.

8.2. Data Sampling Management

Create a new Data Sampling object first by the following steps:

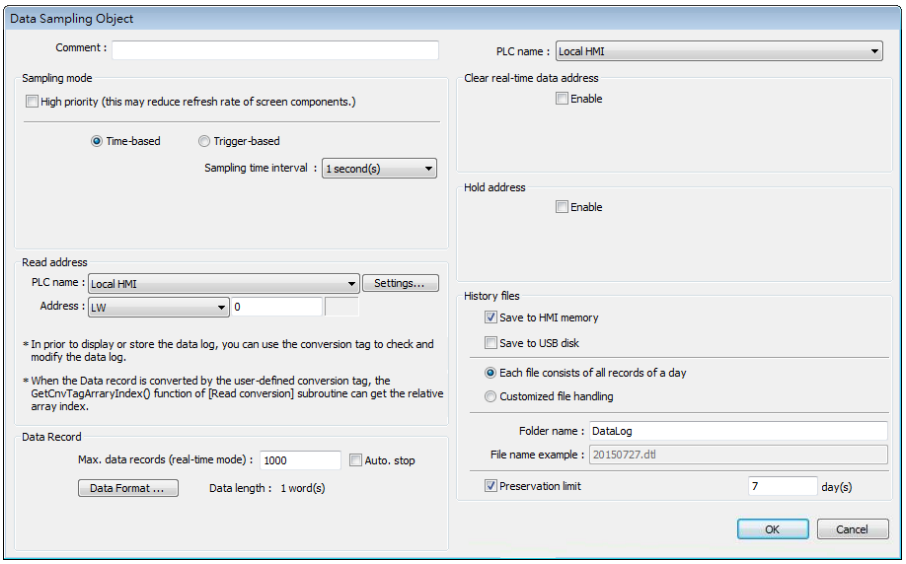
1. From the menu select [Data/History] and click [Data Sampling].
2. Click [New] to finish relevant settings.



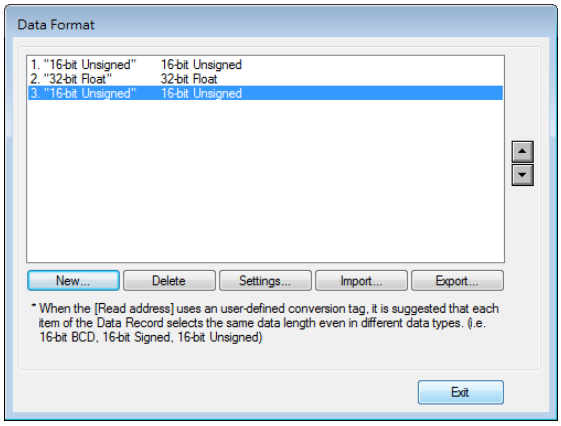
8.3. Creating a New Data Sampling

The following introduces how to set a new Data Sampling.

eMT, iE, XE, mTV, iP Series



Setting	Description
Sampling mode	**High priority** Data sampling processes with this feature enabled will be prioritized. Please note that too many priorities can slow down update rate of other objects. **Time-based** Samples data in a fixed frequency. The [Sampling time interval] can be set from “0.1 second(s) to 120 mins”. **Trigger-based** Triggers data sampling by the status of a designated bit address. **Mode** Conditions to trigger Data Sampling: [OFF -> ON] Triggers sampling when the status of the address changes from OFF to ON. [ON -> OFF] Triggers sampling when the status of the address changes from ON to OFF. [OFF <-> ON] Triggers sampling when the status of the address changes. **Set ON/OFF after triggered** If selected, after triggering Data Sampling, the system will set the designated bit address back to ON/OFF state.
Read address	Specify an address to be the source of Data Sampling. To convert a data record, please use a user-defined conversion tag that allows [Read conversion]. For array addresses, the GetCnvTagArrayIndex function can get the relative array index and then calculate.

Data Record (Real-time)	In Real-time Mode, when [Auto. stop] is not selected, the max. number of data records can be saved on HMI is 86400. When the number of records exceeds 86400, the earliest data will be deleted. **Data Format** Data of different formats in consecutive registers can be sampled. For example, setting: LW-0 (16-bit Unsigned), LW-1 (32-bit Float), and LW-3 (16-bit Unsigned). The upper limit for the number of records is 1000.  **Auto. stop** Data sampling will automatically stop when the number of records reaches [Max. data records]. See “8.3.1 Demonstration of Auto. stop”.
Clear real-time data address	Set when the bit address status changes from [OFF -> ON] or [ON -> OFF], clear the sampled data in Trend Display Real-time Mode. The number of data records returns zero but the data records that are already saved as history files will not be cleared.
Hold address	If the status of the designated address is set ON or OFF, sampling will be paused until the status of the designated address returns.
History files	**Save to HMI memory** Saves Data Sampling to HMI every 10 seconds. Or, use system register [LB-9034] to force storing data. See details on the restrictions of using LB-9034 in the Note below. **Save to SD card / USB disk** Saves Data Sampling to the specified external device. **Each file consists of all records of a day**

	The data sampling file will be saved on a daily basis into the specified folder, and the file name will be *yyyymmdd*.dtl, indicating the date of the file. **Folder name** Specify Data Sampling file name which must be all in ASCII characters. The folder name will be written as: [*Storage Location*] \ [*Folder Name*] \ [*File Name*] **Preservation limit** Determines the number of data sampling files to be preserved. Please note that the current .dtl file is not included in this limit. That is, if this limit is set to 2, apart from the current file, two latest files will be preserved.
Customized file handling	This feature can be used to customize naming and management of data sampling files (*dtl). See “8.3.2 Customized File Handling”.

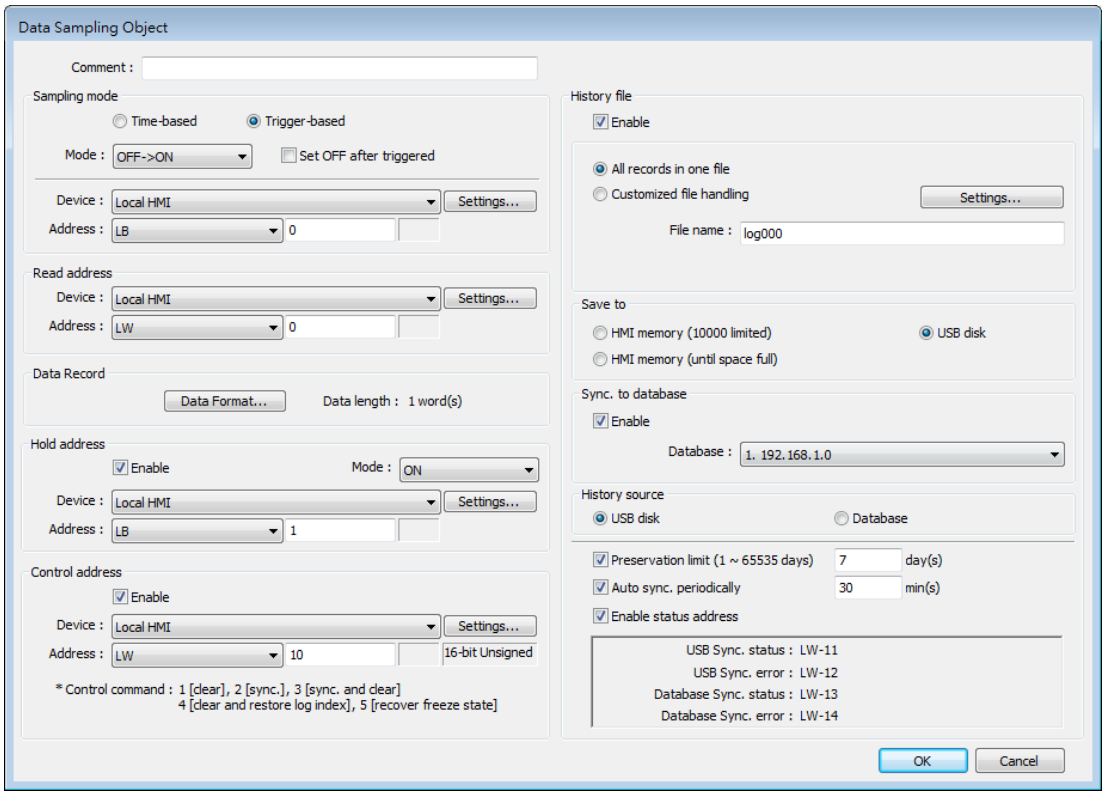
Note

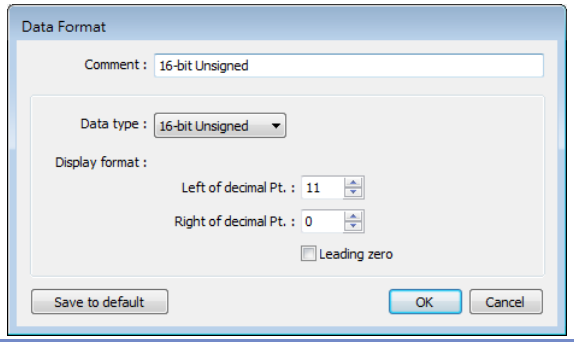
- A Data Sampling may include more than one type of records. Data Sampling can retrieve different types of records at the same time. For example, if define three types of data, 4 words in total, the system retrieves a 4-word data each time from the designated address to be the content in one Data Sampling.
- When using [Each file consists of all records of a day] and set [Preservation limit] to 2 files, the data of yesterday and the day before yesterday will be kept. Data that is not built in this period will be deleted to prevent the storage space from running out.
- When using [Customized file handling] and set [Preservation limit] to 2 files, not only the currently sampled file, another 2 newest files (3 files in total) will be kept. The rest of the data will be deleted to prevent the storage space from running out.
- When running simulation on PC, all data sampling will be saved to the datalog folder in C:\EBPro\[*Storage Location*]\datalog. If you change the data format of data sampling, delete the previous data records in the installation directory to prevent the system from reading the old records.
- When saving files to USB disk or SD card, the capacity of a FAT32 folder depends on the length of the file names. Fewer files can be saved when the file names are longer.
- When using LB-9034, the shortest interval between two successful executions is 2 seconds. That is, after LB-9034 is triggered, it can be effective again 2 seconds later.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

cMT, cMT X Series

The settings are almost similar to eMT, iE, XE, mTV, iP Series. The following highlights the settings that are different.



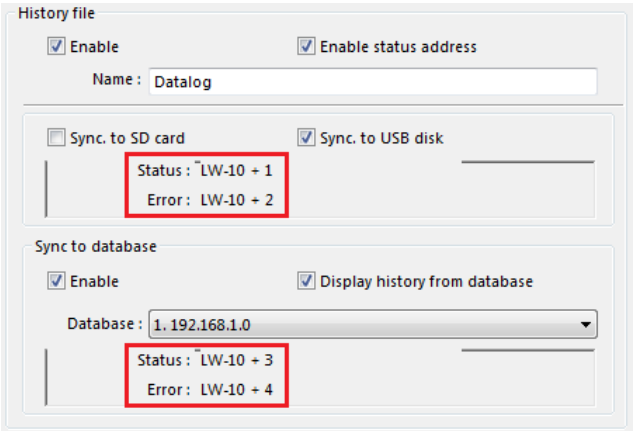
Setting	Description
Data Record	Data of different formats in consecutive registers can be sampled. The upper limit for the number of records is 1000. The display format settings may be used by Trend Displays and History Data Displays which are set to display this data sampling object. 
Control address	Entering a value in the control address sends the corresponding command.

Value	Command
1	Clear the sampled data in HMI
2	Synchronize data to the external device / database
3	Synchronize data to the external device / database and then clear the sampled data in HMI

4	Use the history data stored in USB disk / SD card / database after changing HMI
5	As data sampling stops when HMI storage is full, this command allows restarting of data sampling when there's free space
6	Free up memory space reserved for data logs (*.db). This command can be used when historical data is saved to HMI memory (until space full).

Status & Error address

When LW-n is used as the control address, the four consecutive addresses following LW-n (LW-n+1~LW-n+4) will show status and error, please see the prompt in the settings dialog box.



Value	Status address: LW-n+1 and LW-n+3
0	Disconnected from external device or database
1	Connecting with external device or database
2	Connected with external device or database
3	Storing records into the archive. When this is done, the value returns to 2.
Value	Error address: LW-n+2 and LW-n+4
0	None
1	Unknown error
2	Failed to connect with external device or database
3	Access denied
4	Wrong database name
5	Inconsistent data format
6	Failed to open table
7	Failed to create table
8	Failed to write table

	9	Failed to open database
	10	Database is corrupted

History files

History data can be saved to HMI memory (10000 limited or until space full), USB disk, SD card, and Database (by synchronizing history data to database server.).

The rules of saving the data are:

- When [Save to HMI memory (10000 limited)] is selected and the sampled data reaches 10000 records, the system will delete the earliest 1000 records and keep on saving data to HMI memory.
- When [Save to HMI memory (until space full)] is selected, the system will keep on saving data to HMI memory without clearing the earlier records. In this case, the data may not be synchronized to database.
- When USB disk or SD card is selected, and the number of records reaches 10000, the system will automatically save the sampled data to the external device and delete the earliest 1000 records on HMI.
- To synchronize the data to database server, select a database that has been configured before.

☞ See synchronization rules in “8.4 Synchronizing cMT Viewer data and Saving to External Device”.

☞ See Database Server usage in “13.44 Database Server”.

Customized file handling

This feature can be used to customize naming and management of data sampling files (*db). The current *db file is saved in HMI memory. Please note the following two points when Sync. Destination is USB disk / SD card:

1. When *db file name changes, the *.db file with the former name will be synchronized to USB disk / SD card.
2. If a new *db file is generated when there’s no existing USB disk / SD card on HMI, the earlier *db file will be deleted. For example, if 20161218.db is the current file, when 20161219.db is generated and no external device is inserted to HMI, and then 20161218.db will

	be deleted. When [Customized file handling] is used, enabling [Sync. to database] is not possible. Find Customized File Handling settings dialog box in "8.3.2 Customized File Handling".
Auto sync. periodically	Data will be automatically synchronized to the designated external device in the specified time interval, regardless of the rules explained above. Please note that the timer will reset when a control address is used. See "8.4 Synchronizing cMT Viewer data and Saving to External Device or Database Server".
All record in one file	When Sync. Destination is USB disk / SD card, the preservation limit is 1 to 65535 days.
Customized file handling	When Sync. Destination is USB disk / SD card, the preservation limit is 1 to 65535 files. When Sync. Destination is HMI memory, the preservation limit is 1 to 1000 files.
History source	Data can be read from USB disk, SD card, or Database.

8.3.1. Demonstration of Auto. stop

This feature depends on the arrangement of different objects and modes. (Set [Max. data records] to n.)

Object	[Auto. stop] not selected	[Auto. stop] selected
Trend Display-Real time	Deletes the earlier records and displays the latest number of records (n) in Trend Display. See the following figure.	Stops after reaching the specified number of data records (n).
Trend Display-Historical	Keeps on sampling data and displays all history data in Trend Display.	Stops after reaching the specified number of data records (n).
History Data Display	Keeps on sampling data and displays all history data in History Data Display.	Stops after reaching the specified number of data records (n).
Data Sampling	Keeps on sampling new data.	Stops after reaching the specified number of data records (n).

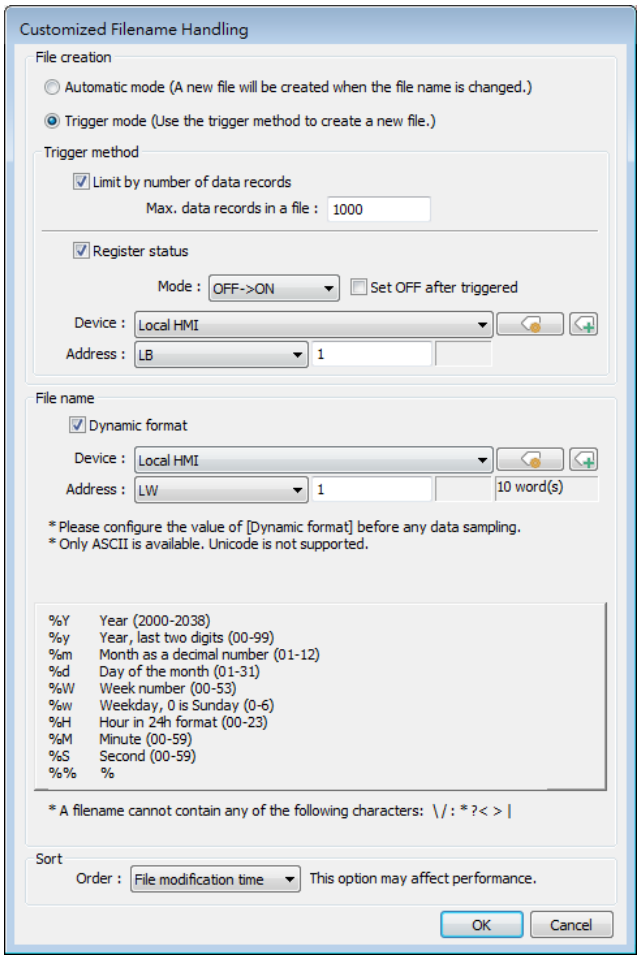
The figure illustrates how the data is sampled in Trend Display – Real Time mode when [Auto. stop check] box is not selected. Set the number of data records to 10, when the 11th data is

generated, the earliest record is deleted and the newest record is added.

Record Number	Data	Not selecting [Auto. stop]
1	101	102
2	102	103
3	103	104
4	104	105
5	105	106
6	106	107
7	107	108
8	108	109
9	109	110
10	110	111
11	111	

8.3.2. Customized File Handling

This feature can be used to customize naming and management of data sampling files (*.dtl, *db).



Setting	Description
File creation	**Automatic mode** A new file will be created when the name of an existing file is changed. **Trigger mode** A new file will be created according to the [Trigger method] settings.
Trigger method	**Limit by number of data records** A new file will be created when the number of data sampling records reaches the specified "Max. data records in a file" **Register status** A new file will be created when the status of a designated bit address meets the specified condition. The condition is specified in Mode field. **Set ON/OFF after triggered** If selected, after the new file is created, the system will set the designated bit address back to ON/OFF state. **Following HMI reboot, generate and write to a new file only after trigger condition has been met.** When the trigger condition has *not* been met following HMI reboot and therefore no new file is generated, the new data will be written to the last file generated before HMI is rebooted. Downloading a new project to HMI is the same as rebooting HMI.
File name	The file name can be an alphanumeric name, and certain half-width symbols are allowed. The file name can also be specified by a file name syntax. **Dynamic format** The file names can be set by a designated word address, or by a file name syntax indicating the current system time. The file name syntax can be specified by selecting time buttons or entering the syntax in Format field. The length limit is from 1 to 25. The following half-width characters are not allowed: \ / : * ? “ < > |
Sort	This setting affects how the system determines the sort

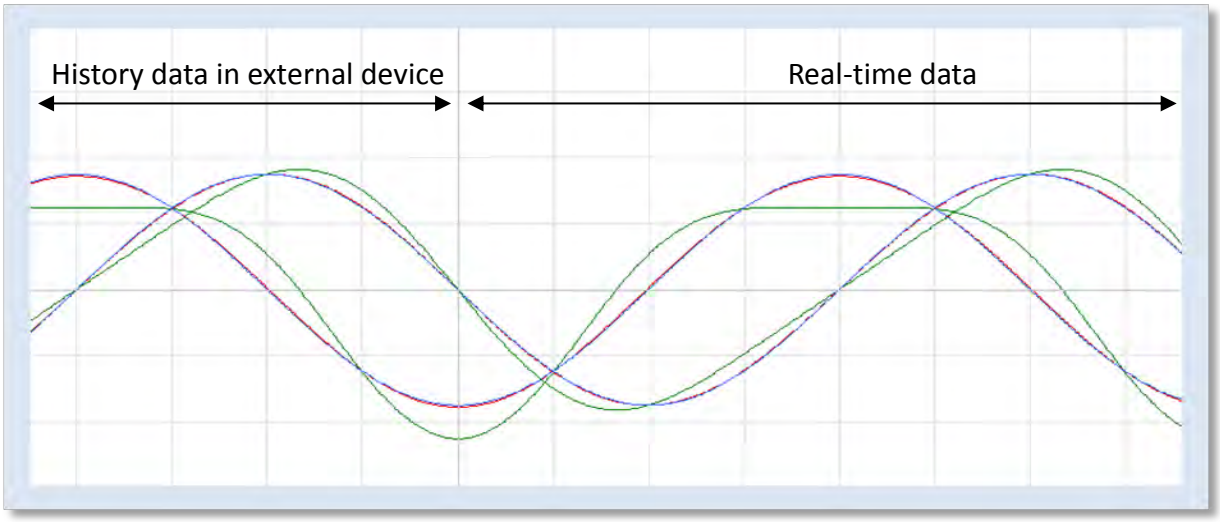
	order of historical files, which affects the file preservation and backup logic. **File name** Files are sorted by their filenames in descending order. **File modification time** Files are sorted by the actual timestamps of the contents. (Because the system has to check the content of the files, this option may affect performance negatively.) Example: Consider the case where files have been generated in this order: filename_05--> filename_04--> filename_03--> filename_02--> filename_01. The earliest file is filename_05 and the latest file is filename_01. With preservation limit set to 5 files, the two sorting methods yield different results when the 6th file is generated: Sorted by Filename: filename_05 is seen as the latest file; therefore, filename_01 will be deleted. Sorted by File Modification Time: filename_01 is seen as the latest file; therefore, filename_05 will be deleted. Note on naming convention: When File Name is selected as the sorting method and the filenames are: filename_1, filename_2, and filename_10, filename_2 will be seen as the latest file, followed by filename_10 and filename_1. To make filename_10 the latest file, please zero-pad the numbers and name the files to: filename_01, filename_02, and filename_10.

Note

- If both [Limit by number of data records] and [Dynamic format] check boxes are selected, before startup HMI, please enter the name in the designated register for Dynamic Format, otherwise, it is impossible the reach the “Max. data records in a file”, and the data sampling file will not be generated.
- When a new file is generated, the system will first detect if the filename already exists. If the file name does exist, the newly sampled data will be appended to the existing file.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

8.4. Synchronizing cMT Viewer data and Saving to External Device or Database



For other series, when displaying the sampled data in Trend Display object, it is necessary to select from Real-time mode or History mode and the two modes cannot simultaneously be displayed in one object.

cMT / cMT X Series allows displaying history data and at the same time updates real-time data in one Trend Display or History Data Display object. The data saved in the external device can be updated.

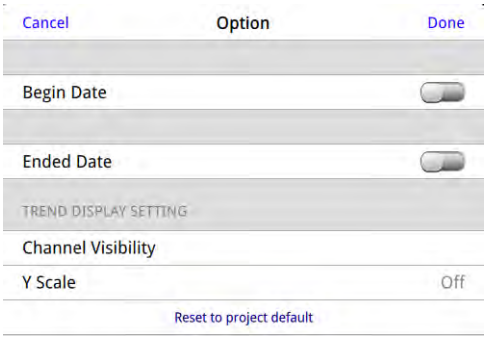
The rule of synchronizing the data saved in the external device:

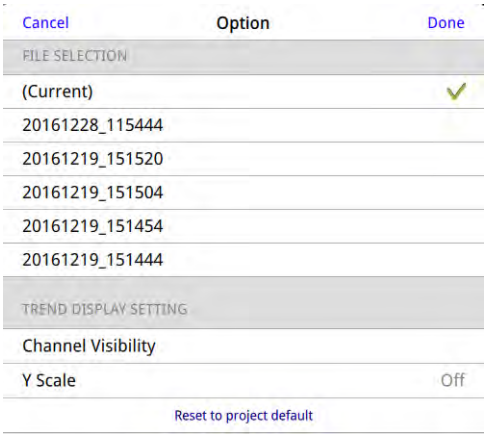
1. When the sampled data reaches 10000 records, HMI will automatically save data to the external device and deletes the earliest 1000 records in HMI.
2. If the external device is removed from HMI and inserted back again at the time when the sampled data is under 9000 records, the data generated during the time the external device is removed is saved in HMI and is not cleared. If the data exceeds 9000 records during the time the external device is removed, the earlier data is cleared and cannot be synchronized even to insert the external device back to the HMI.
3. If there already exists sampled data in the external device, the new data is appended without overwriting the original data each time in synchronization.

8.5. Checking History Data of a Specific Date / File on cMT Viewer.

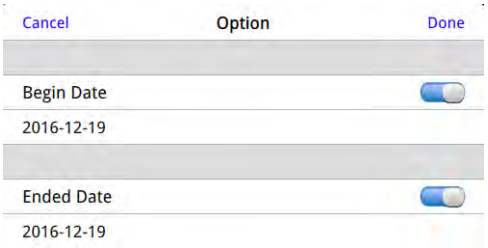
To check the history data, see the following steps (Use Trend Display object as example).

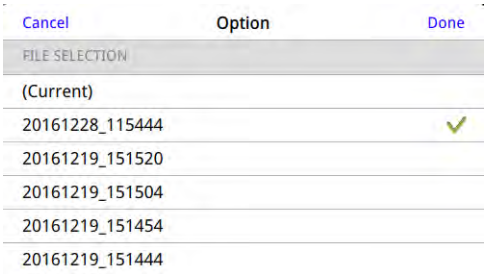
1. Tap the icon in the upper-right corner of the Trend Display object.
2. The following dialog box appears.





3. Specify the date or the file.





4. Tap [Done] button to finish setting.

9. Object General Properties

This chapter explains the basic settings of an object.

9.1. Overview 9-2
9.2. Selecting PLC and Setting Read/Write Address 9-2
9.3. Using Shape Library and Picture Library 9-3
9.4. Setting Label Text 9-7
9.5. Adjusting Profile Size 9-9
9.6. Overlapping Objects (cMT / cMT X Series) 9-10

9.1. Overview

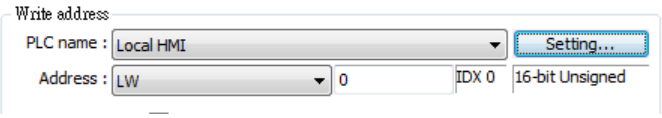
The basic steps to create an object:

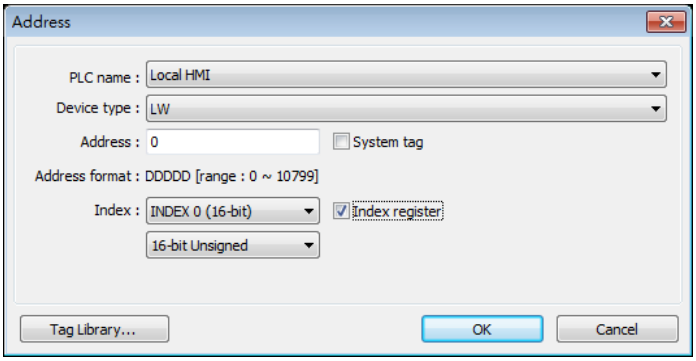
1. Selecting the PLC device and setting the read/write address.
2. Using Shape Library and Picture Library.
3. Setting label text.
4. Adjusting profile size.

This chapter explains the basic settings of an object.

9.2. Selecting PLC and Setting Read/Write Address

Most objects read data from PLC devices, so a properly configured PLC address is needed. Select the PLC to control at [PLC name] which comes from [System Parameters Settings] » [Device List].





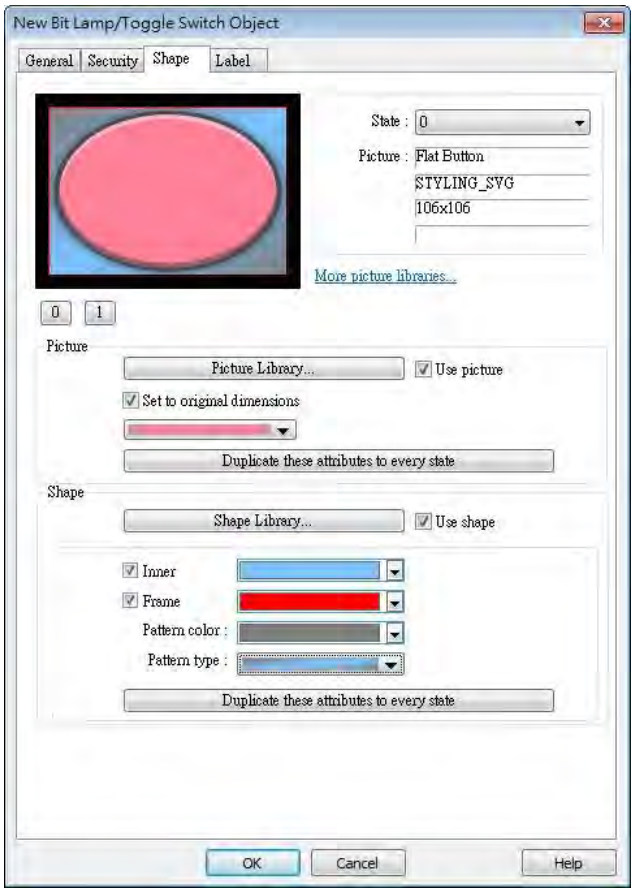
Setting	Description
PLC name	Select the PLC type.
Device type	Different PLCs have different device types.
Address	Set the read/write address.
System tag	Address tags include [System Tag] and [User-defined Tag]. This option allows users to use [System Tag]. [System Tag] consists of the preserved addresses by system for particular purposes. The address tags include bit registers or word registers (LB or LW). After selecting [System tag], not only will the [Device type] field display the chosen tag, but [Address] field will also display the chosen register.

Index register	Select this check box to use the [Index Register].

- For more information about System Tag, see “22 System Reserved Words and Bits”.
- For more information about Index Register, see “11 Index Register”.
- For more information about Tag Library, see “16 Address Tag Library”.

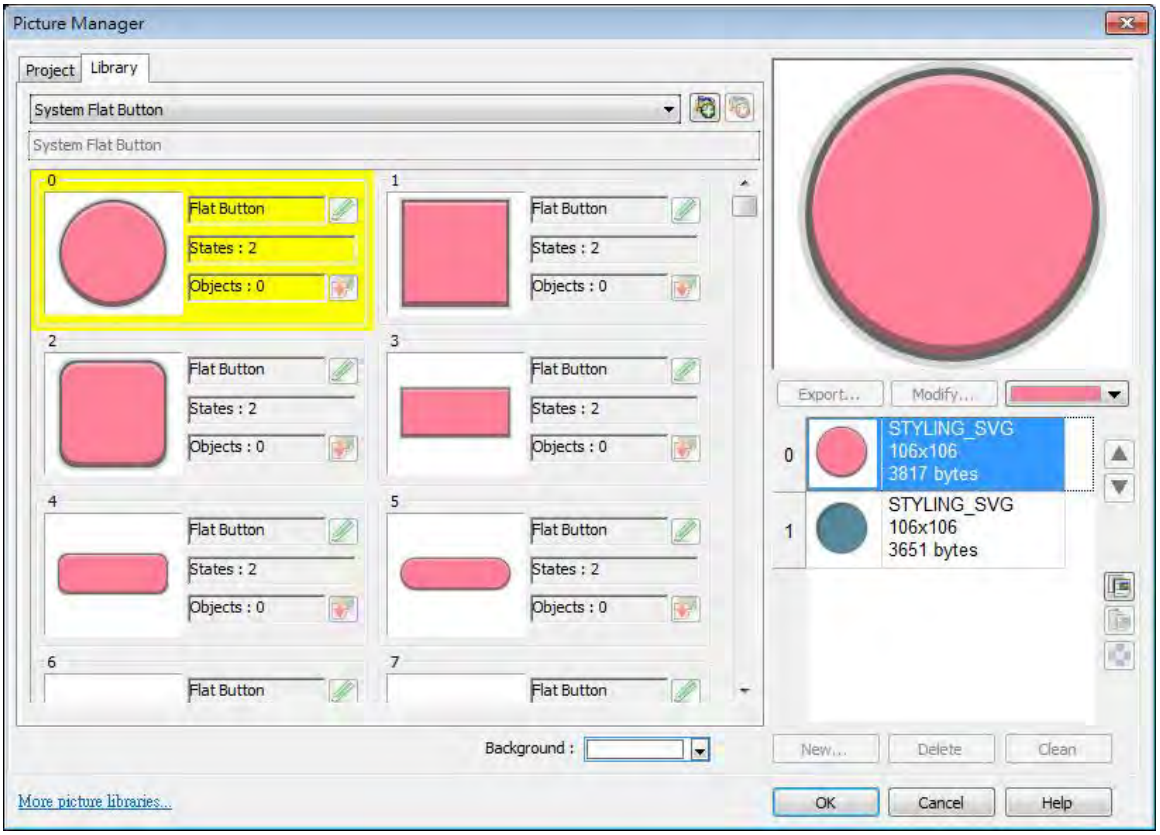
9.3. Using Shape Library and Picture Library

In the settings dialog box of certain objects, “Shape” tab can be found. In this tab, Shape Library and Picture Library can be used to add visual effects on objects. Select Shape tab to use the libraries when creating an object.



9.3.1. Picture

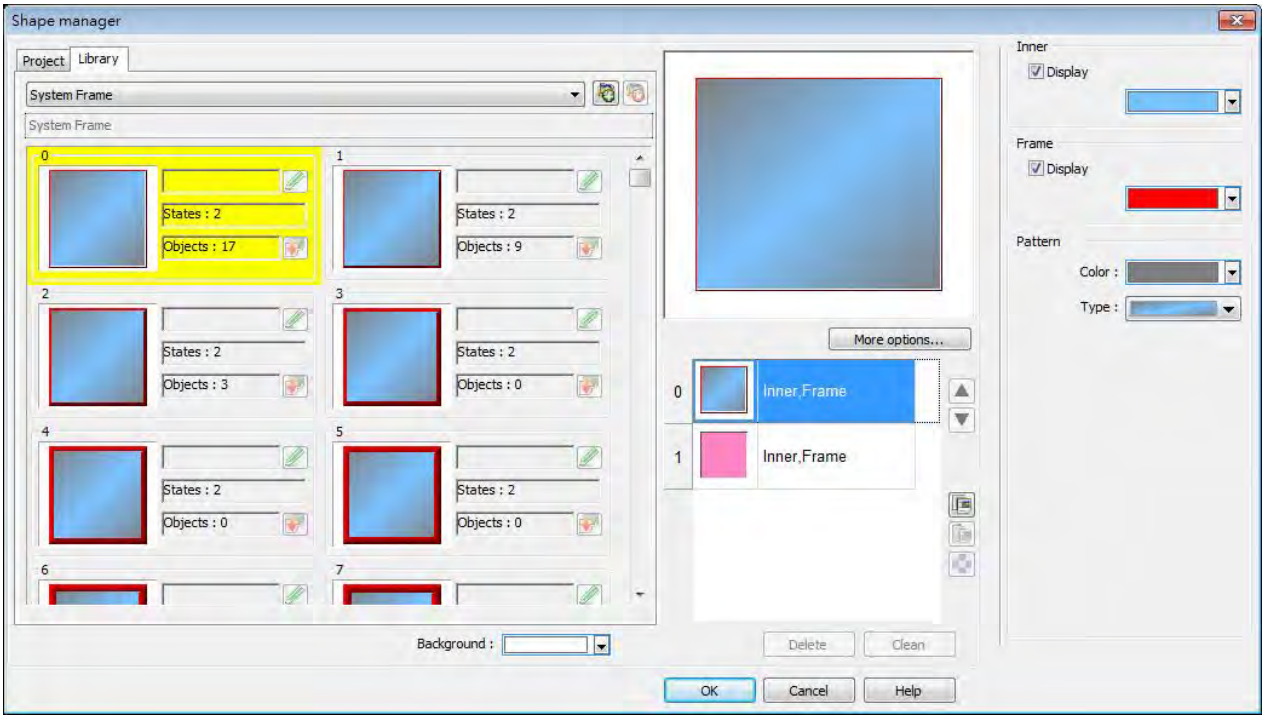
Select [Use picture] and then click [Picture Library] button to open Picture Manager window and select a picture from the window.

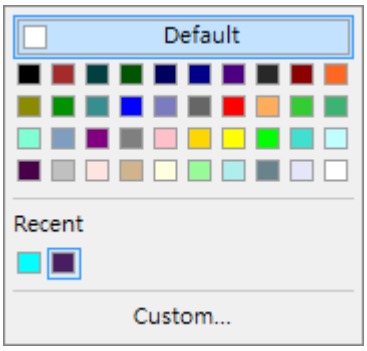


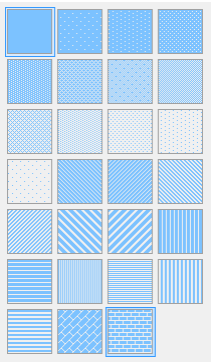
Setting	Description
Picture Library	Click [Picture Library] button in Picture group box to open Picture Manager window.
Set to original dimensions	When this checkbox is selected, EasyBuilder will return the picture to its original size to eliminate the redrawing step.
Duplicate these attributes to every state	This button is available only when a system picture is used. Clicking this button will duplicate the attributes of the current picture to other states.

9.3.2. Shape

Select [Use shape] and then click [Shape Library] button to open Shape Manager window and select a shape from the window.

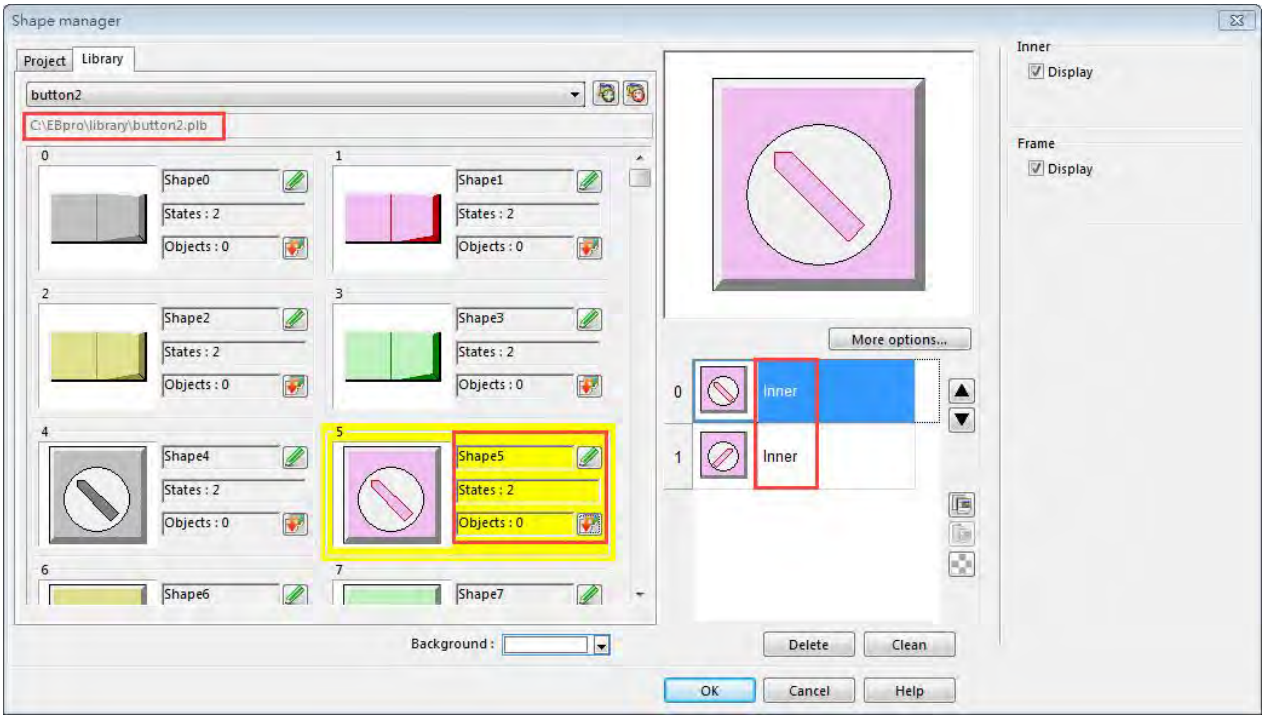


Setting	Description
Inner	Select [Display] to show the inner color of the shape and select a color as the inner color by clicking the drop-down button. 
Frame	Select [Display] to show the frame color of the shape and select a color as the frame color by clicking the drop-down button.
Pattern color	Click the drop-down button to select a color for the interior pattern of the shape.
Pattern type	Click the drop-down button to select a pattern.

	
Set to all states	Duplicate the attributes of the current state to other states.

9.3.3. Shape manager

In Picture Manager and Shape Manager windows, the currently selected picture / shape is highlighted yellow as shown below.



The information about the selected picture / shape:

Directory The directory from which EasyBuilder searches for the pictures / shapes.

Shape5 Name of the shape.

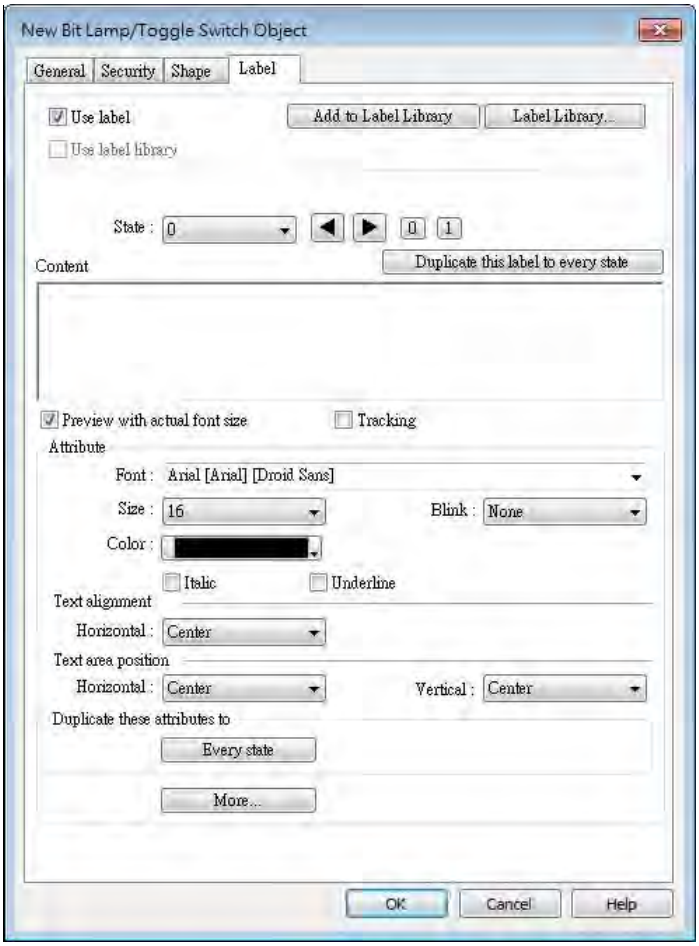
States: 2 Number of states of the shape.

Objects: 1 This shape is used by 0 object in the project.

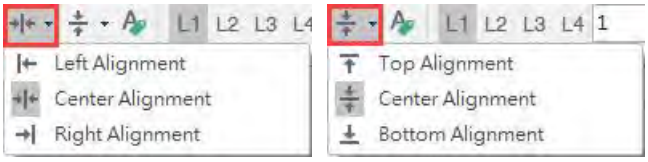
Inner The state 0 and state 1 of this shape contain only “inner” but not “frame.”

For more information, see “14 Shape Library and Picture Library”.

9.4. Setting Label Text



Setting	Description
Use label	Use a label for the object. **Add to Label Library** Add a new label containing the content for all states to the Label Library.
Use label library	Use a label from the Label Library.
Label Library	View all the labels in the Label Library. For more information, see “15 Label Tag Library and Multi Language”.
Content	Set the content of the label. If [Use label library] is selected, the content in the Label Library can be used.
Duplicate this label to every state	Copy the content of the current state to all states.
Tracking	With this option selected, changing the position of the

	text in one state will also change the text position in other states.
Font	Select a font from the list. EasyBuilder Pro supports Windows True-type fonts.
Size	Select a font size.
Color	Select a font color.
Blink	Specify the way the text blinks. Choose [None] to disable this feature or set the blinking interval to [1 second] or [0.5 seconds].
Italic	Use Italic font. *Italic Label*
Underline	Use Underline font. Underline Label
Text Alignment	Multiline text is aligned within the label outline. Below are the corresponding shortcut keys. [Left] [Center] [Right] 111 222222 333333333 / 111 222222 333333333 / 111 222222 333333333
Text area position	Multiline text is aligned within the object outline. Below are the corresponding shortcut keys.  Horizontal Vertical
Duplicate these attributes to every state	Copy the attributes of the current state to all states.
More	**Convert labels to bitmap images (Use bitmap font)** Convert the label text into bitmap format. This setting is only available for non-cMT models. **ON = OFF (use state 0)** The content displayed will not change due to state. **Movement:** **Direction**

	Set the direction of the marquee effect. The directions include: [No movement], [Left], [Right], [Up], [Down].
	Continuous
	Specify how the marquee effect is displayed.
	If this option is not selected, the next text appears only when the previous text disappears completely.
	If this option is selected, the text will be displayed continuously.
	Speed
	Adjust the speed of the text movement.

Note

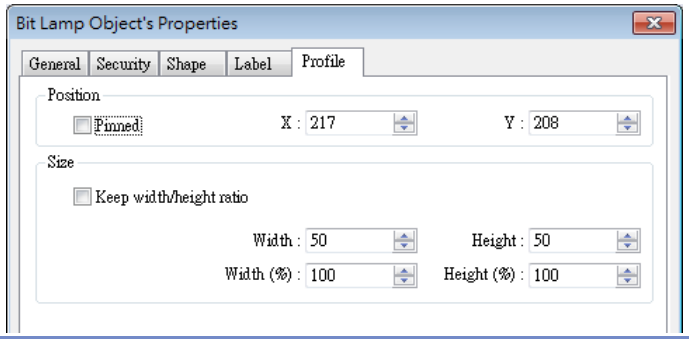
- When Text/Comment object is used with Comment enabled, the text used as comment will not appear on HMI.
- When the texts of the fonts used in the editing window and on the HMI screen are not properly aligned vertically, it is possible to correct it with the FontRules.xml file in "%appdata%\Weintek\EasyBuilder" folder.

  By adding the names of the fonts in question into the FontRules.xml file, their vertical metrics can be adjusted in advance to avoid the misalignment problem.

```
<?xml version="1.0"?>
<FontRules>
    <Version data="1.1"/>
    <Rule data="[\w]*algerian[.\w]*"/>
    <Rule data="[\w]*castellar[.\w]*"/>
    <Rule data="^Wingdings$"/>
    <Rule data="fix_vertical_metrics = ^Gotham Pro Black$"/>
    <Rule data="fix_vertical_metrics = ^Gotham Pro Bold$"/>
    <Rule data="fix_vertical_metrics = ^Gotham Pro$"/>
</FontRules>
```

9.5. Adjusting Profile Size

When an object is created and placed in the editing screen, double click it and select the [Profile] tab to adjust the position and size of the object.



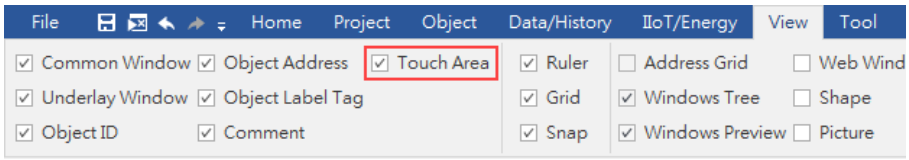
Setting	Description
Position	**Pinned** When this check box is selected, the position and the size of the object cannot be changed. **[X]** and **[Y]** The coordinates of the position of the object in the editing screen.
Size	Adjust the [width] and [height] of the object.

9.6. Overlapping Objects (cMT / cMT X Series)

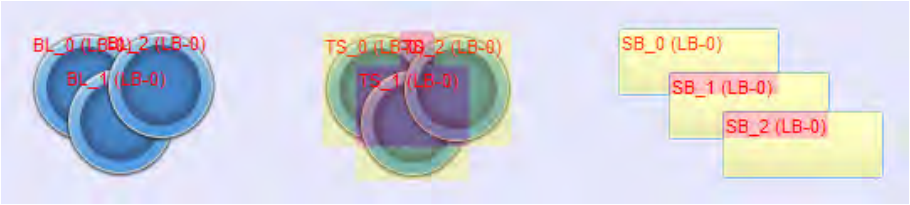
The following explains the ways to find the overlapping objects and convert them into a single Combo Button. After conversion, the general settings of the Combo Button will follow the settings of the frontmost object.

9.6.1. Enable Touch Area

1. Open View tab and select [Touch Area].

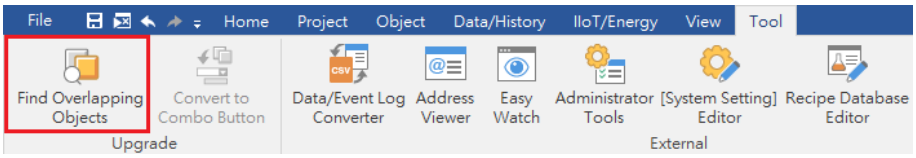


2. All the touchable objects in the proejct will be highlighted in yellow, the overlapping areas will be highlighted in pink, and the objects that are not touchable will remain the same.

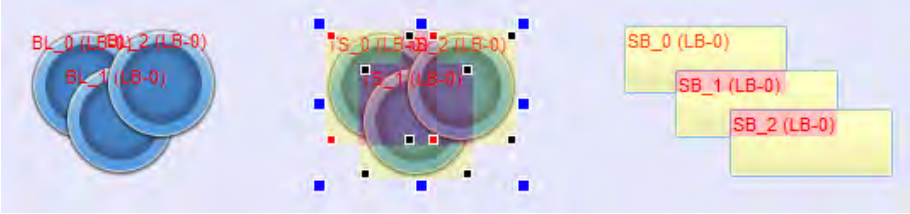


9.6.2. Find Overlapping Objects

1. Open Tool tab and click [Find Overlapping Objects] to find all the touchable overlapping objects in the project.



2. At the first time clicking [Find Overlapping Objects], the system will find the first group of overlapping objects in the current window. To search for the next group, click this button again. When all the overlapping objects in the current window are found, the system will start searching for overlapping objects in the next window, and then open the window in which overlapping objects are found.



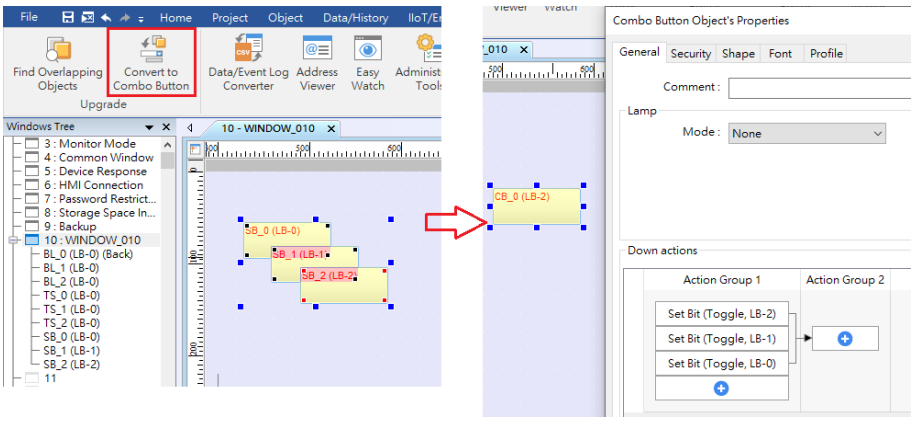
Note

- All layers of a window must be editable before clicking [Find Overlapping Objects].

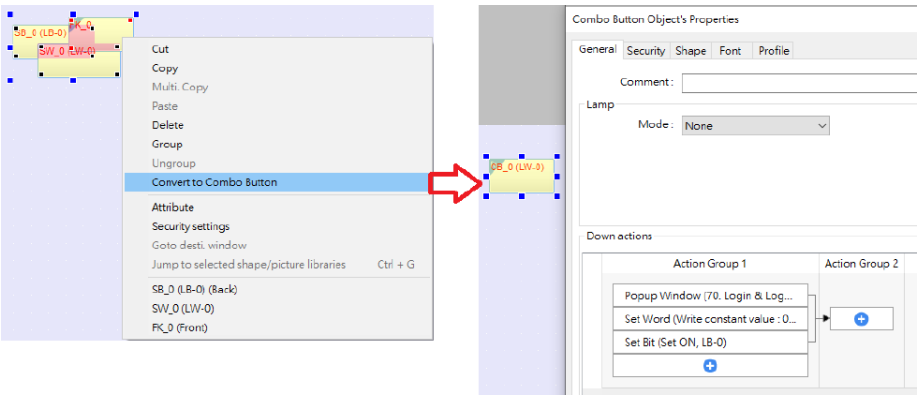
9.6.3. Convert to Combo Button

When the overlapping objects found are Set Bit, Toggle Switch, Set Word, Multi-State Switch and Function Key, they can be converted to a single Combo Button by the following ways.

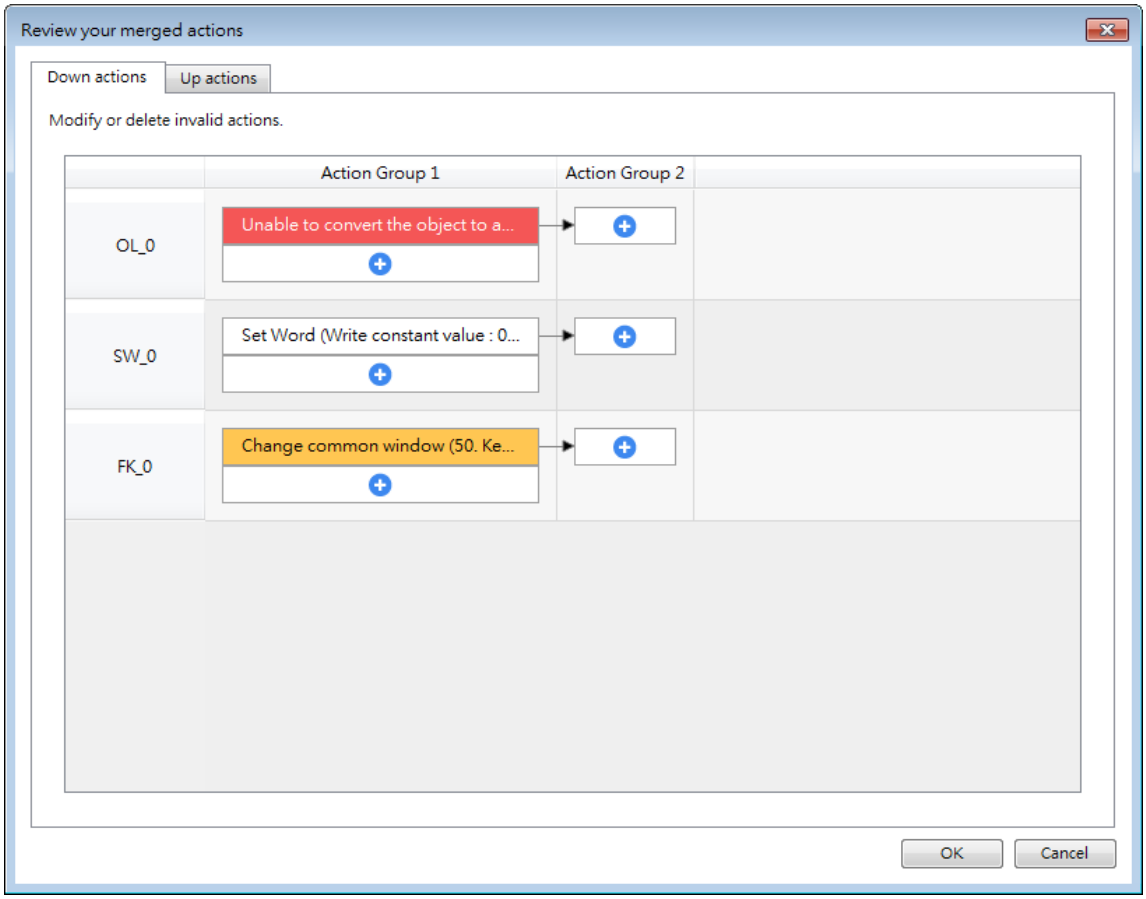
Way 1. Open Tool tab and click [Convert to Combo Button].



Way 2. Select the overlapping objects together, click the right mouse button, and then select [Conver to Combo Button].



After [Convert to Combo Button] is clicked, a review window shows for users to review the actions that can be triggered by the Combo Button. Actions that are not highlighted can be merged; actions that are highlighted in yellow will be automatically adjusted to meet merging condition; and actions that are highlighted in red cannot be merged. The user can move the mouse cursor to the yellow or red area to see explanations.



Note

- The Comment of the generated Combo Button will follow the setting of the frontmost object that has a Comment.

- The Security settings of the generated Combo Button will follow the settings of the frontmost object. If the Security settings of the objects layers down are different from the frontmost object, an alert is shown after conversion.
- The Sound settings of the generated Combo Button will follow the settings of the frontmost object that has Sound settings. If the Sound settings of the objects layers down are different from the frontmost object, an alert is shown after conversion.
- The Picture or Shape, and the Label settings of the generated Combo Button will follow the settings of the frontmost object that has these settings. When there's no object using a picture or a shape, the Combo Button will follow the Label settings of the backmost object that is mergeable.
- The generated Combo Button will have the Vibration setting enabled if any of the overlapping objects has Vibration setting enabled.
- The generated Combo Button will follow the Operation Log settings of the overlapping object that uses Operation Log. When Operation Log is used for multiple overlapping objects, an alert is shown after conversion.

10. User Password and Object Security

This chapter discusses the protection for operations provided by setting up user passwords and security classes.

10.1. Overview 10-2
10.2. User Password and Operable Object Classes 10-2
10.3. Enhanced Security Mode and Control Address 10-6
10.4. Enhanced Security Mode Usage 10-10
10.5. Object Security Settings 10-23
10.6. Example of Object Security Settings 10-28
10.7. Protecting Password Settings from Unauthorized Editing 10-31
10.8. Bulk Changing of Security Settings of Multiple Objects 10-32

10.1. Overview

This chapter discusses the protection for operations provided by setting up user passwords and security classes. Authentication modes are:

- General Mode
- Enhanced Security Mode

In addition, cMT / cMT X series allows the use of LDAP protocols for user authentication.

To set up the protection system, please:

1. Set user password and operable classes.
2. Set object class for objects.

An object belongs only to one security class. Setting the object class to “None” means any user can operate this object.

10.2. User Password and Operable Object Classes

The security parameters can be found in [System Parameter Settings] » [Security].

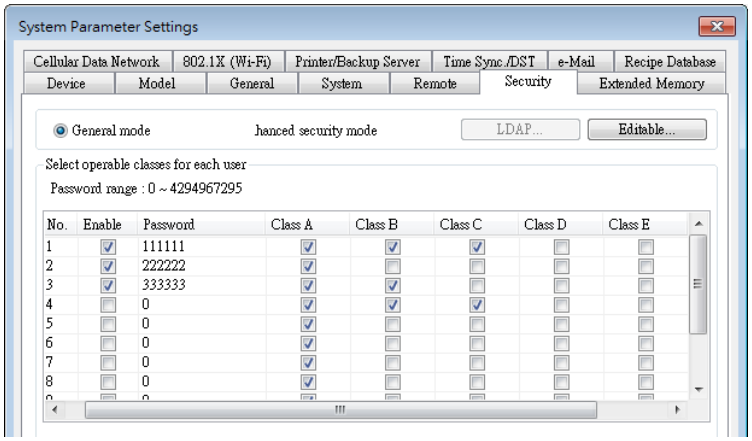
10.2.1. General Mode

Up to 12 sets of user and password are available. A password should be one non-negative integer. There are six security classes: A to F.

Once the password is entered, the objects that the user can operate are classified. As shown below, “User 1” can only operate objects with class A or class C.

Note

- General Mode is not used for cMT / cMT X Series.

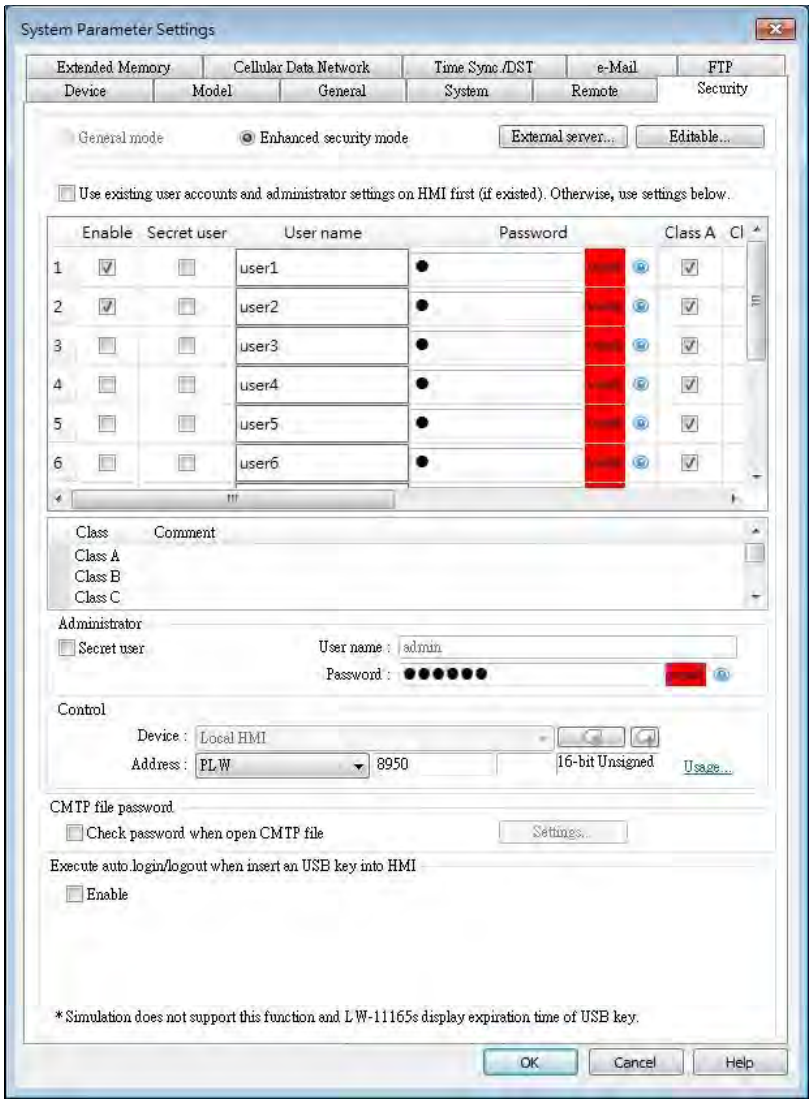


Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

10.2.2. Enhanced Security Mode

Up to 11 users can be set here. In addition, [Administrator] setting is provided. Administrator has all privileges and can operate all object classes. User passwords must be alphanumeric characters and each user can have up to 12 operable classes: A to L. (Up to 127 users can be set in Administrator Tools. Please see “10.4 Enhanced Security Mode Usage” for more details.)

Enhanced Security Mode provides a [Control address] for users to manage the accounts directly on HMI. Please see “10.3 Enhanced Security Mode and Control Address” for more details. Alternatively, use USB Security Key to log in automatically. Insert the USB disk in which the key is saved to log in. Please see 10.4.3 Login / Logout Automatically with USB Security Key” for more details.



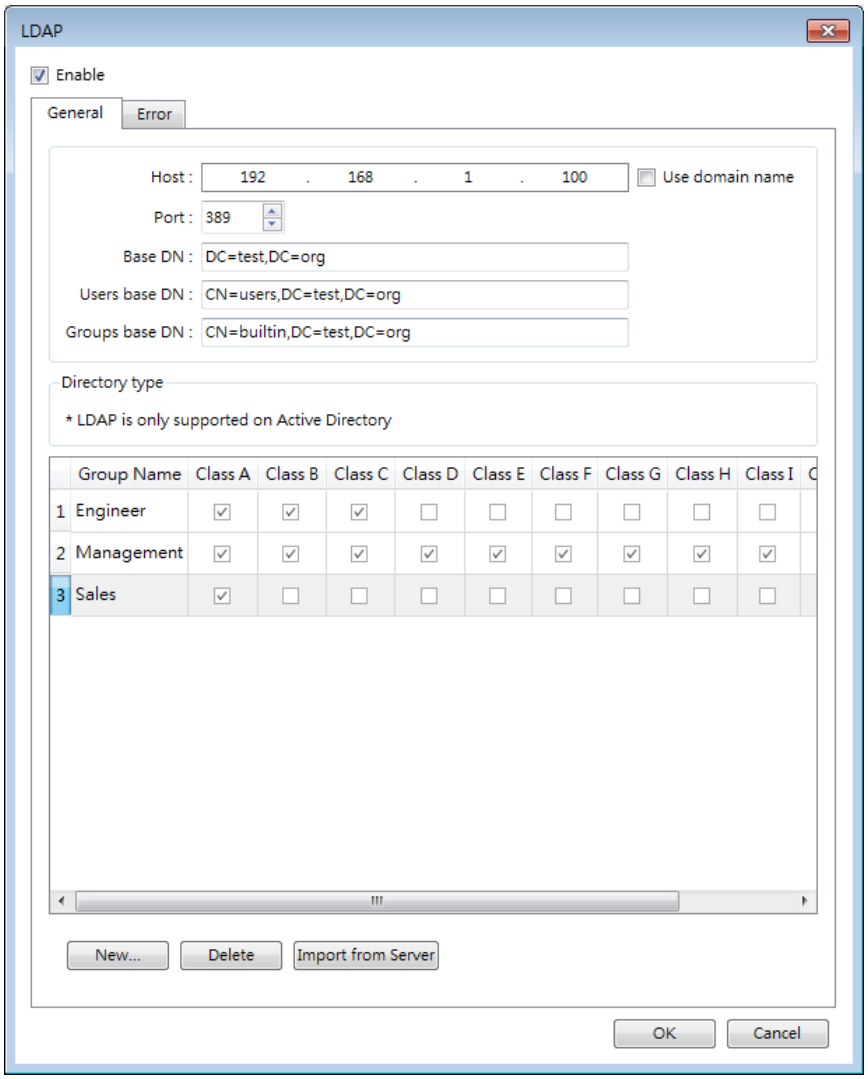
Note

- In Enhanced Security Mode on a cMT / cMT X model, the Control Address can only be assigned to a word register of Local HMI. Please note that security features will work only on HMI when the control address is LW. Remote login on cMT Viewer will not be possible.
- EasyAccess 2.0's HMI Viewer on a eMT/iE/XE/mTV model doesn't support Enhance Security Mode, please use VNC Viewer instead.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

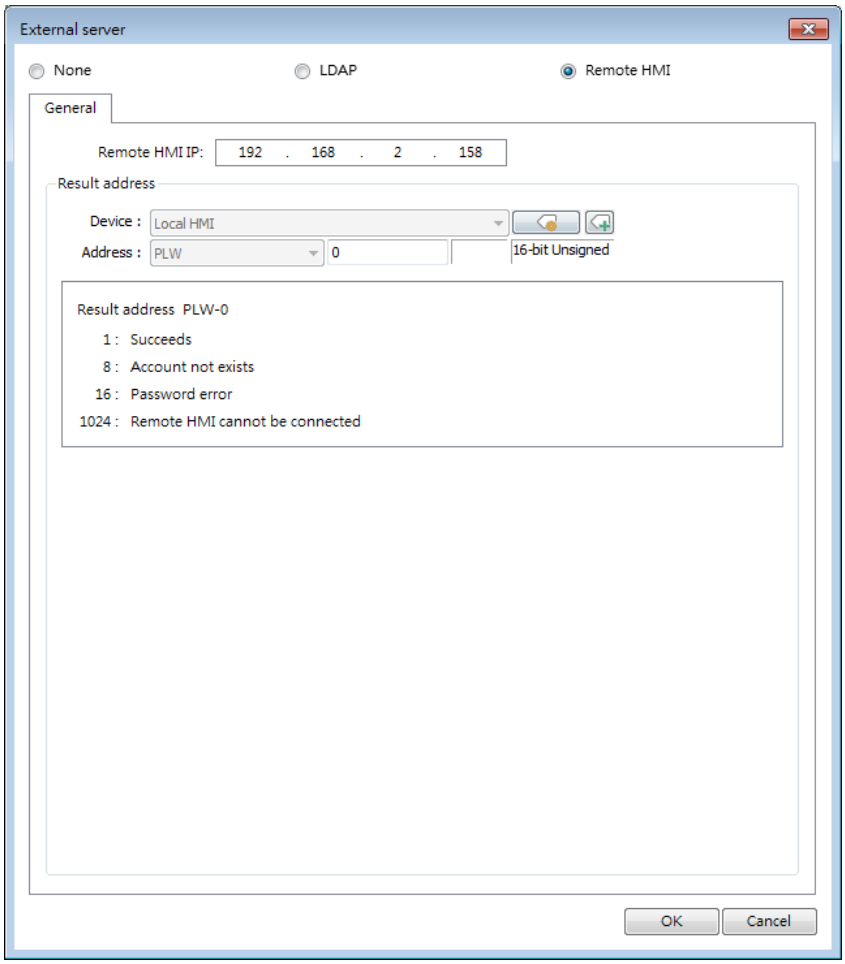
10.2.3. LDAP Mode

LDAP (Lightweight Directory Access Protocol) enables applications to access Directory server providing database-like data structure, and here, the primary use of LDAP is to enable centralized user account management. When using LDAP mode, user account management is up to the Directory server, with HMI validating user login via the LDAP protocol. To have LDAP set up on HMI, users only need to provide necessary information about the directory server and set the operable classes for each group, without the need for managing username/password for each user.



10.2.4. Remote HMI Mode

In this mode, user accounts can be managed on a remote HMI, instead of the local HMI. The accounts on a remote HMI can be used to log in the local HMI; therefore, managing the accounts on the local HMI is not necessary.



10.3. Enhanced Security Mode and Control Address

The Control Address is used for login and account management. The Control Address can only be assigned to LW register on Local HMI, and 20 consecutive registers will be used. To log in using Control Address, please select to log in by [user name] or [user index].
Please set [user name] and [password] in [System Parameter Settings] » [Security] » [Enhanced security mode] in advance.

10.3.1. Control Address Settings

When control address is set to LW-n, where n is an arbitrary number, the following addresses will be designated:

Address	Tag Name	Description
LW-n (1 word)	command	Commands to be executed: Login, Logout, Add/Setting/Delete Accounts, etc.
LW-n + 1 (1 word)	command execution result	Displays the result of executing commands.
LW-n + 2 (1 word)	user index	The index of accounts (used with Option

		List Object).
LW-n + 3 (1 word)	user privilege	Binary value. Level A = bit0, Level B = bit1, …
LW-n + 4 (8 words)	user name	Account name (Case-sensitive and only allows letters and numbers).
LW-n + 12 (8 words)	password	Account password (Case-sensitive and only allows letters, numbers, or special characters).

After setting the [Control address], the relevant addresses can be found in [Address Tag Library] » [User-defined tags].For example, setting [Control address] to LW-0: (UAC stands for User Account Control)

LW-0 → [UAC command]

LW-1 → [UAC command execution result]

LW-2 → [UAC user index]

LW-3 → [UAC user privilege]

LW-4 ~ LW-11 → [UAC user name]

LW-12 ~ LW-20 → [UAC password]

Note

- In Enhanced Security Mode on a cMT / cMT X model, the Control Address can only be assigned to a word register of Local HMI. Please note that security features will work only on HMI when the control address is LW. Remote login on cMT Viewer will not be possible.
- EasyAccess 2.0's HMI Viewer on a eMT/iE/XE/mTV model doesn't support Enhance Security Mode, please use VNC Viewer instead.

10.3.2. Commands

Setting different values in LW-n [command] enables different commands:

Set Value	Command	Corresponding Address
1	Log in by user name	Set [user name] and [password] first. After entering the user name and password, the system will check if they are valid in [System Parameter Settings] » [Security] » [Enhanced security mode].
2	Log in by user index	Set [user index] and [password] first. Please refer to 10.4.4 Enhanced Security Mode with Option List Object.
3	Log out	
4	Change the password of current logged-in user	Set [user name] and [password] first. Please fill in the original password in [user name] and new password in [password].
5	Add an account	Set [user name], [password] and [user privilege] first.

6	Add a temporary account (minutes)	Set [user name], [password], [user privilege], and [user index] first. [user index] is for specifying a time period (in minutes), within this period the account is valid. If 0 is specified, this account stays valid until the HMI is powered off.
7	Delete an existing account by user name	Set [user name] first.
8	Delete an existing account by user index	Set [user index] first.
9	Setting the privilege of an existing account by user name	Set [user name] and [user privilege] first.
10	Setting the privilege of an existing account by user index	Set [user index] and [user privilege] first.
11	Setting the password of an existing account by user name	Set [user name] and [password] first.
12	Setting the password of an existing account by user index	Set [user index] and [password] first.
13	Read the privilege of an existing account by user name	Set [user name] first. If the command succeeds, [user privilege] can be displayed.
14	Read the privilege of an existing account by user index	Set [user index] first. If the command succeeds, [user privilege] can be displayed.
15	Add a temporary account (days)	Set [user name], [password], [user privilege], and [user index] first. [user index] is for specifying a time period (number of days), within this period the account is valid. If 0 is specified, this account stays valid until the HMI is powered off.
16	Add an expiring account (minutes)	Set [user name], [password], [user privilege], and [user index] first. [user index] is for specifying a time period (in minutes), within this period the account is valid. 0 is an invalid value for this setting.
17	Add an expiring account (days)	Set [user name], [password], [user privilege], and [user index] first. [user index] is for specifying a time period (number of days), within this period the account is valid. 0 is an invalid value for this setting.
18	Remaining minutes	Set [user name] first. If succeeded, the

	for user name	remaining time (in minutes) will be displayed in [user index].
19	Remaining minutes for user index	Set [user index] first. If succeeded, the remaining time (in minutes) will be displayed in [user index].
20	Remaining days for user name	Set [user name] first. If succeeded, the remaining time (number of days) will be displayed in [user index].
21	Remaining days for user index	Set [user index] first. If succeeded, the remaining time (number of days) will be displayed in [user index].

Note

- Add a temporary account / expiring account: The difference between temporary accounts and expiring accounts is that temporary accounts are not stored in the system and will be invalid after HMI is turned off. Both temporary accounts and expiring accounts will be automatically deleted when they are expired.
- Delete the existing account: The currently logged in account cannot be deleted.
- Offline/Online Simulation: Simulate using the account settings in the program. Any modifications of the account during simulation will not be reserved for next simulation.
- admin: Default administrator account, cannot be deleted, has all privileges and cannot be changed.
- System Register PLW-10754: Displays current user name. (Only available for cMT / cMT X Series)
- The [user privilege] address does not display the privileges assigned to current user account, please use system register LW-9222 to display the privileges.
- LDAP mode does not support login with [user index].

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

10.3.3. Command Execution Results

After the command is executed, the system will store the result code to control address LW-n + 1. The listed result codes below are shown in hexadecimal format.

Result Codes	Command execution result
(0x001)	Succeeds
(0x002)	Invalid command
(0x004)	Account exists (when adding a new account)
(0x008)	Account not exists
(0x010)	Password error
(0x020)	Deny command

(0x040)	Invalid name
(0x080)	Invalid password character exists
(0x100)	Invalid import data
(0x200)	Out of validity range (when log in by USB Security Key). The [Effective Time] can be set in Administrator Tools.

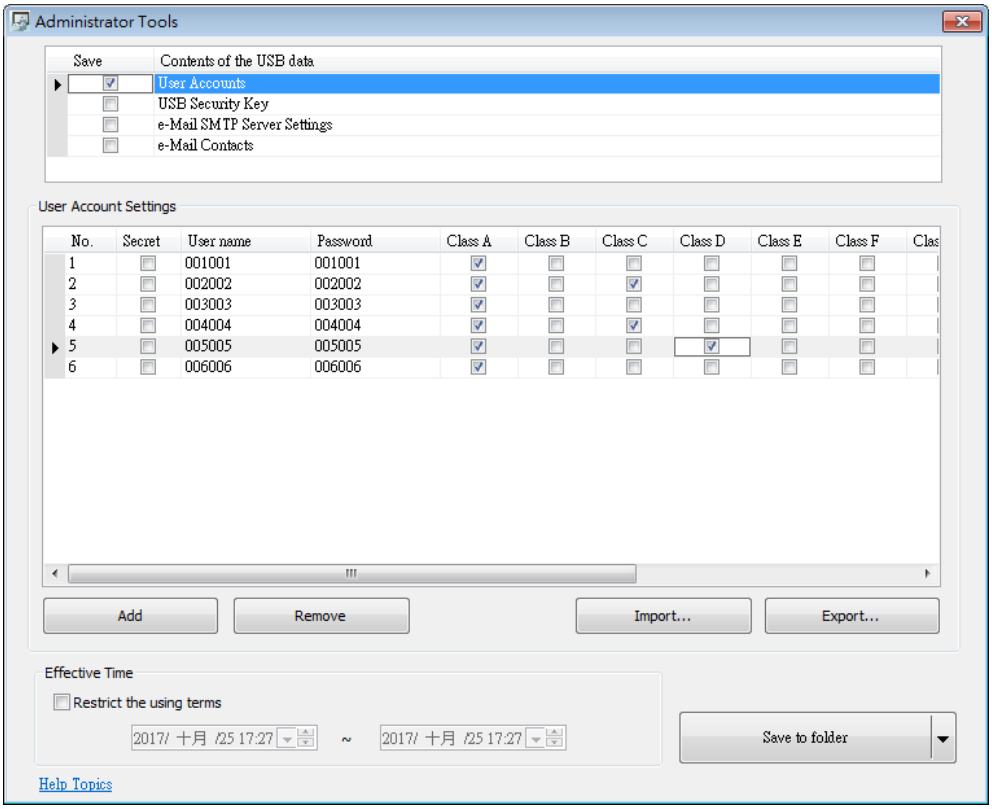
Note

- Users can add a new event in Event (Alarm) Log, and designate the [Read address] to LW-n + 1 [command execution result]. Open [Message] tab » [Text] » [Content] and specify the message to be displayed in Event Display Object for showing command execution result.

10.4. Enhanced Security Mode Usage

10.4.1. Importing User Accounts

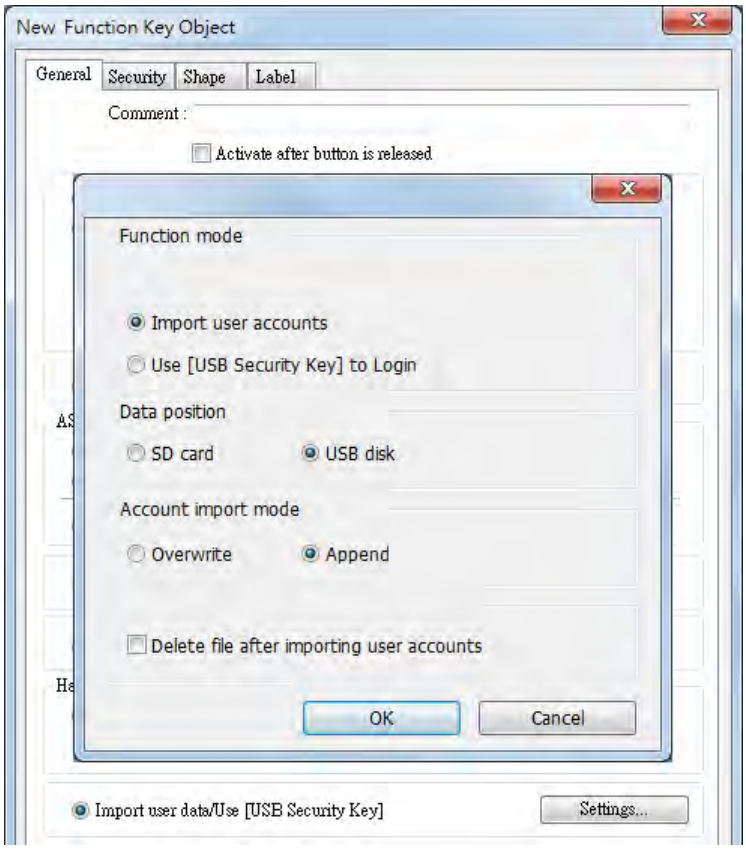
The user accounts can be set using other tools we provide, apart from the settings in [System Parameter Settings] » [Security] tab. Administrator Tools can also be used to set user accounts. Administrator Tools can be found in the installation directory. After the program starts, select the [User Accounts] check box. Up to 127 accounts can be added.



For more information, see “36 Administrator Tools”.

The added accounts can be stored in USB disk or SD card and imported in HMI by a Function

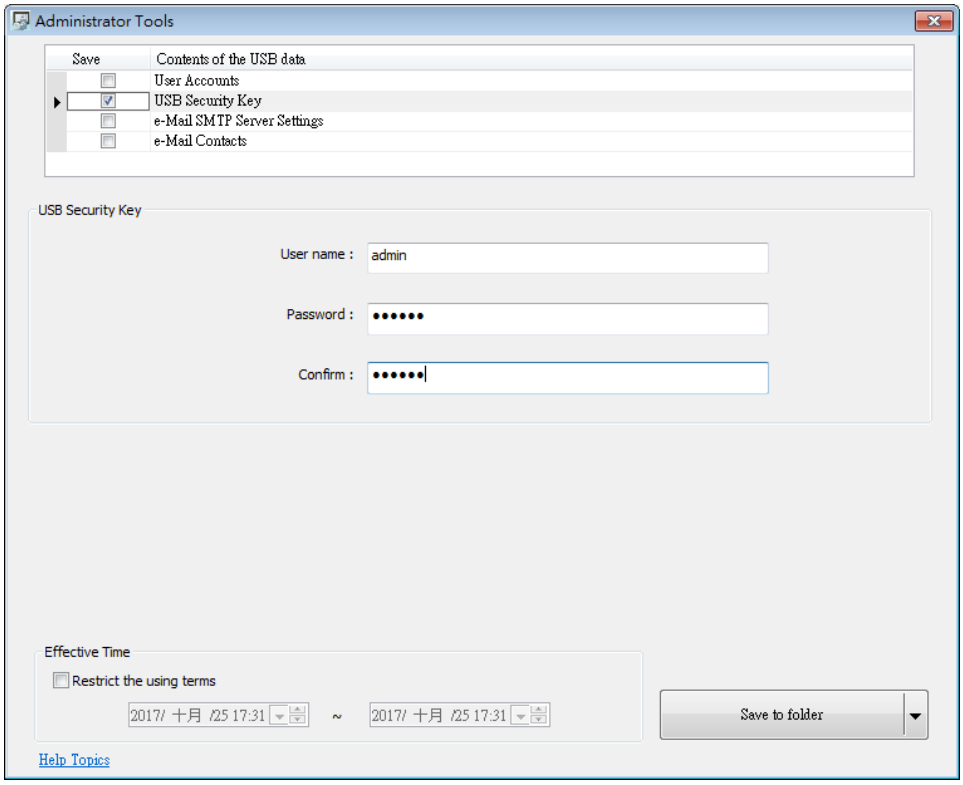
Key Object. To do so, create a Function Key Object, and select [Import user accounts].



When finished, insert the external device to HMI, and press Function Key to import accounts. If [Overwrite] is selected, the existing accounts will be overwritten with new accounts and automatically log out after importing. If select [Delete file after importing user accounts] check box, the system will delete the account data saved in the external device after importing. If the [Effective Time] in Administrator Tools is specified, the importing can only be done in the time limit specified. The imported accounts will not be deleted by system when the effective time ends.

10.4.2. Login with USB Security Key

Instead of entering user name and password to login, a key can be used to do so. In EasyBuilder Pro installation directory, launch Administrator Tools, select [USB Security Key] check box. The account information uses the predefined data in [System Parameter Settings] » [Security].

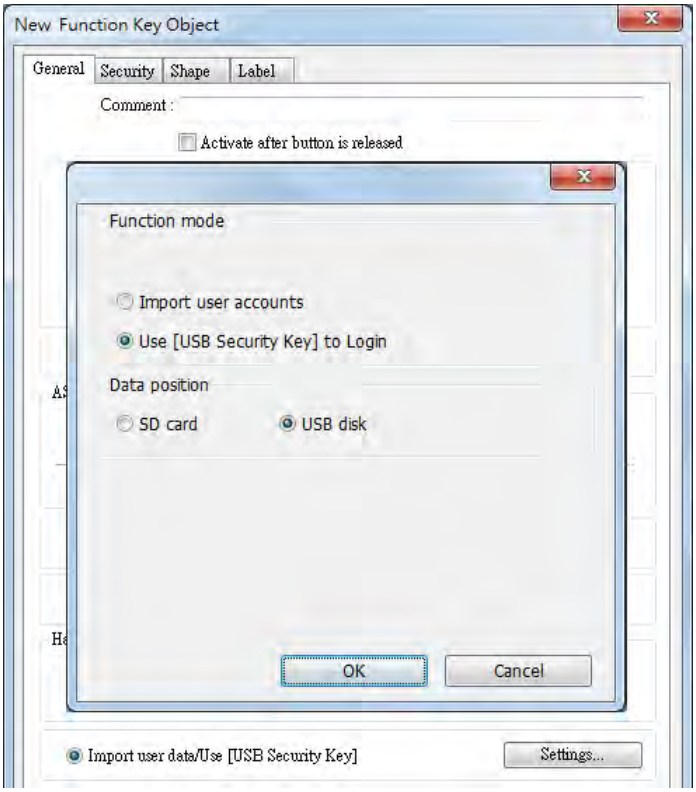


Note

- Please note that the user accounts used for USB Security Key must already exist in HMI.

For more information, see “36 Administrator Tools”.

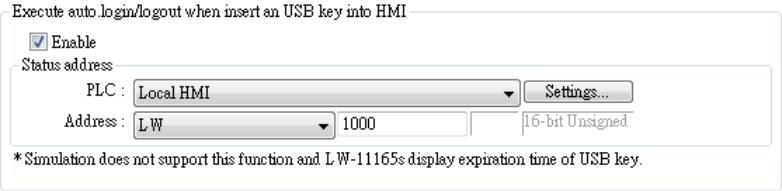
USB Security Key can be stored in USB disk or SD card, and create a Function Key to log in by USB Security Key as shown below:



When finished, insert the external device to HMI, and press Function Key to log in using USB Security Key. If the [Effective Time] in Administrator Tools is specified, the login can only be done in the time limit specified. The system will log out automatically when the key expires.

10.4.3. Login / Logout Automatically with USB Security Key

As shown below, in [System Parameter Settings] » [Security], select [Enable] check box for [Execute auto. login/logout when insert an USB key into HMI].



This function allows automatic login / logout using an USB security key. Insert the USB disk in which the key is saved to HMI to log in, and remove the USB disk to log out. The login / logout status will be written into a designated address, the result codes of login / logout:

0x00: No Action

0x01: Login Succeeds

0x04: Login Fails

0x08: Login Succeeds

0x10: Logout Fails

For more information about USB Security Key, see “36 Administrator Tools”.

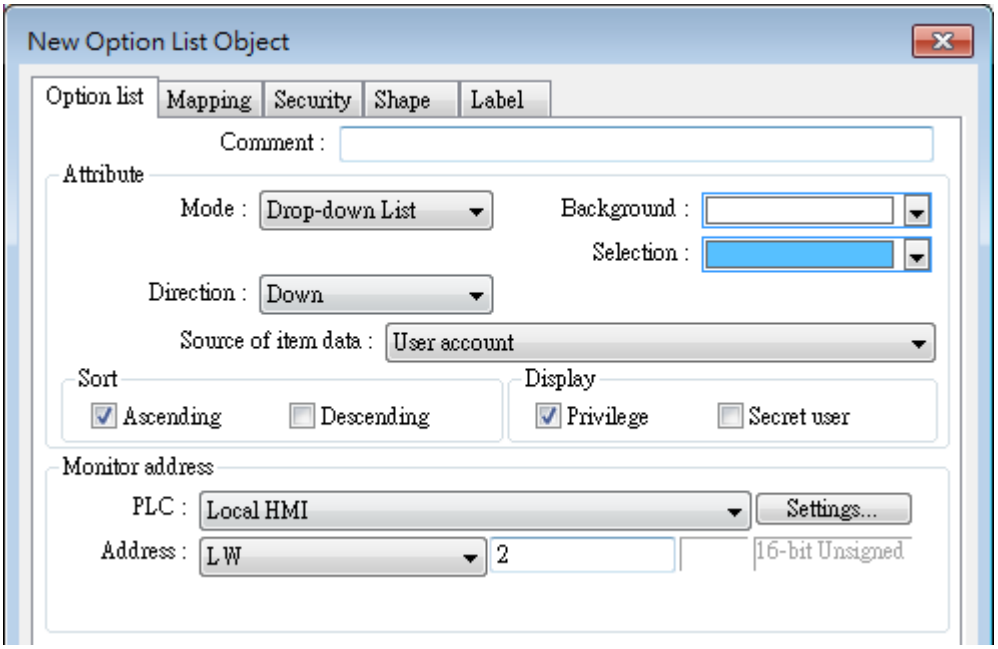
Note

- When Auto Login / Logout is enabled, log in by [Function Key] object is not possible, but it is still possible to log in / out with a designated control address.
- This function does not support On-line / Off-line simulation.
- Only the USB Security Key saved in USB disk is valid.

Click the icon to download the demo project that explains how to use USB Security Key to log in / out. Please confirm your internet connection before downloading the demo project.

10.4.4. Enhanced Security Mode with Option List Object

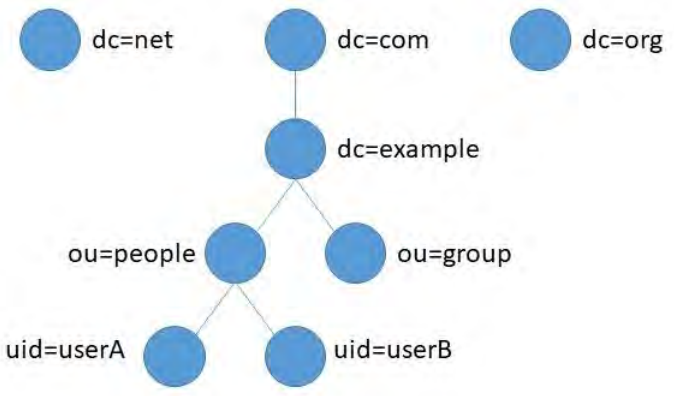
Enhanced Security Mode uses Control Address LW-n + 2 as account index. With Option List Object, account names and privileges can be displayed. Users can select whether or not to display the account privileges and secret users in Option List. Secret users are set to be hidden in [System Parameter Settings] » [Security] » [Enhanced Security Mode]; their account names will be hidden in Option List if [Secret user] check box is not selected. If the control address is set to LW-0, the monitor address for index of Option List is designated to LW-2.



10.4.5. LDAP Mode

LDAP (Lightweight Directory Access Protocol) enables applications to access Directory server providing database-like data structure, and here, the primary use of LDAP is to enable centralized user account management. When using LDAP mode, user account management is up to the Directory server, with HMI validating user login via the LDAP protocol. To have LDAP

set up on HMI, users only need to provide necessary information about the directory server and set the operable classes for each group, without the need for managing username/password for each user.



The control addresses used by LDAP Mode are the same as the control addresses used by Enhanced Security Mode. Please see chapter 10.3 in this user manual for more information on the control address. Please note that obtaining LDAP user name using Option List object is not possible; therefore, [Log in by user index] is not supported.

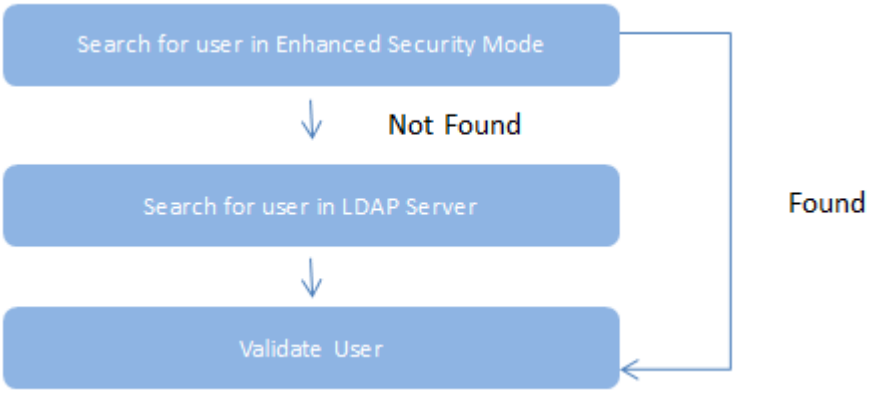
Note

- A user may be a member of multiple groups; in this case, the user has permission to operate all classes assigned for all the groups the user is in. As shown in the following figure, if a user is a member of both Engineer and Sales groups, the user can operate classes A~F.

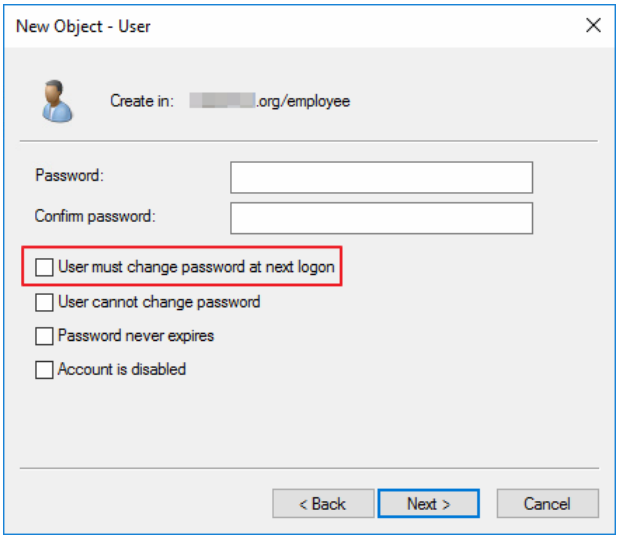
	Group Name	Class A	Class B	Class C	Class D	Class E	Class F
1	Engineer	☑	☑	☑	☐	☐	☐
2	Sales	☐	☐	☐	☑	☑	☑

- The credentials in the list in Enhanced Security Mode can also be managed and validated in LDAP mode. Please note that when a username exists in the lists of both modes, the system will only validate user using Enhanced Security Mode. As shown in the following figure, in the case where username Angela exists in the user list in both LDAP server and Enhanced Security Mode, the HMI will validate user under Enhanced Security Mode.

LDAP Server		Enhanced Security Mode				
Name	Type	No.	Enable	Secret user	User name	Password
Angela	User	1	☑	☐	Angela	1
Bella	User	2	☑	☐	Gina	2
Cindy	User	3	☑	☐	Helen	3
Dora	User					
Elly	User					
Fanny	User					

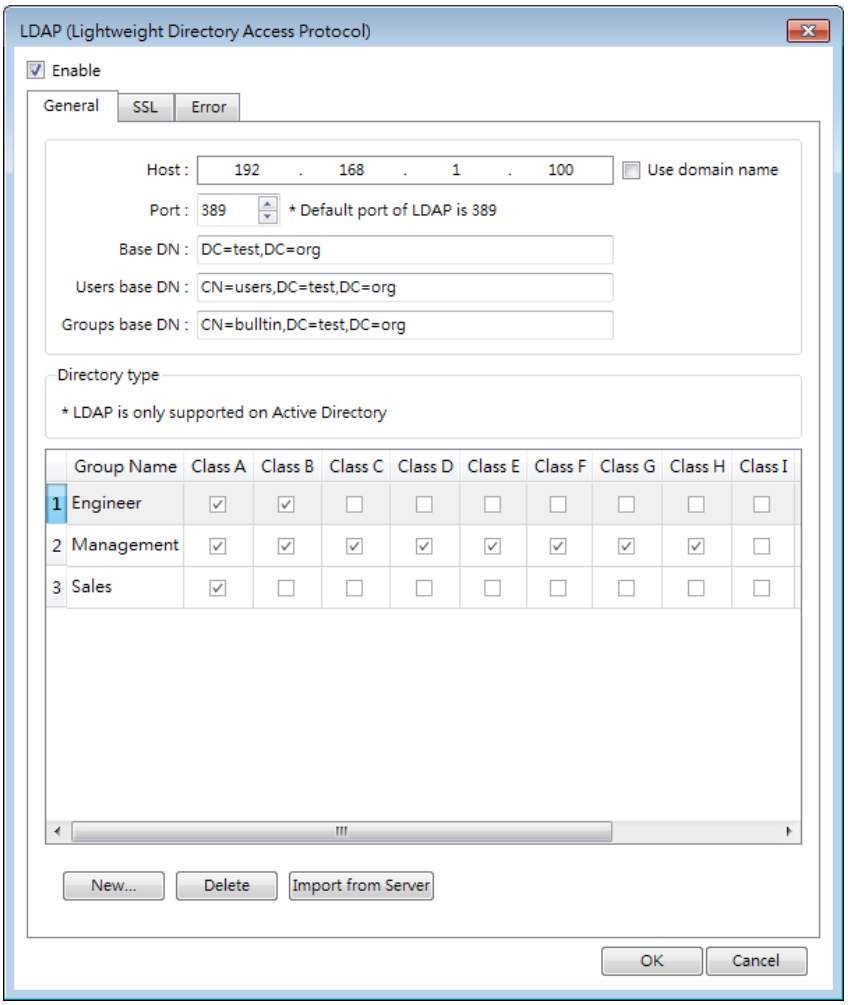


- LDAP Mode does not support login with [user index].
- LDAP is only supported on Active Directory.
- HMI cannot change user's password; therefore, when adding a new user in LDAP server, please do not select [User must change password next logon].



10.4.5.1. General Tab

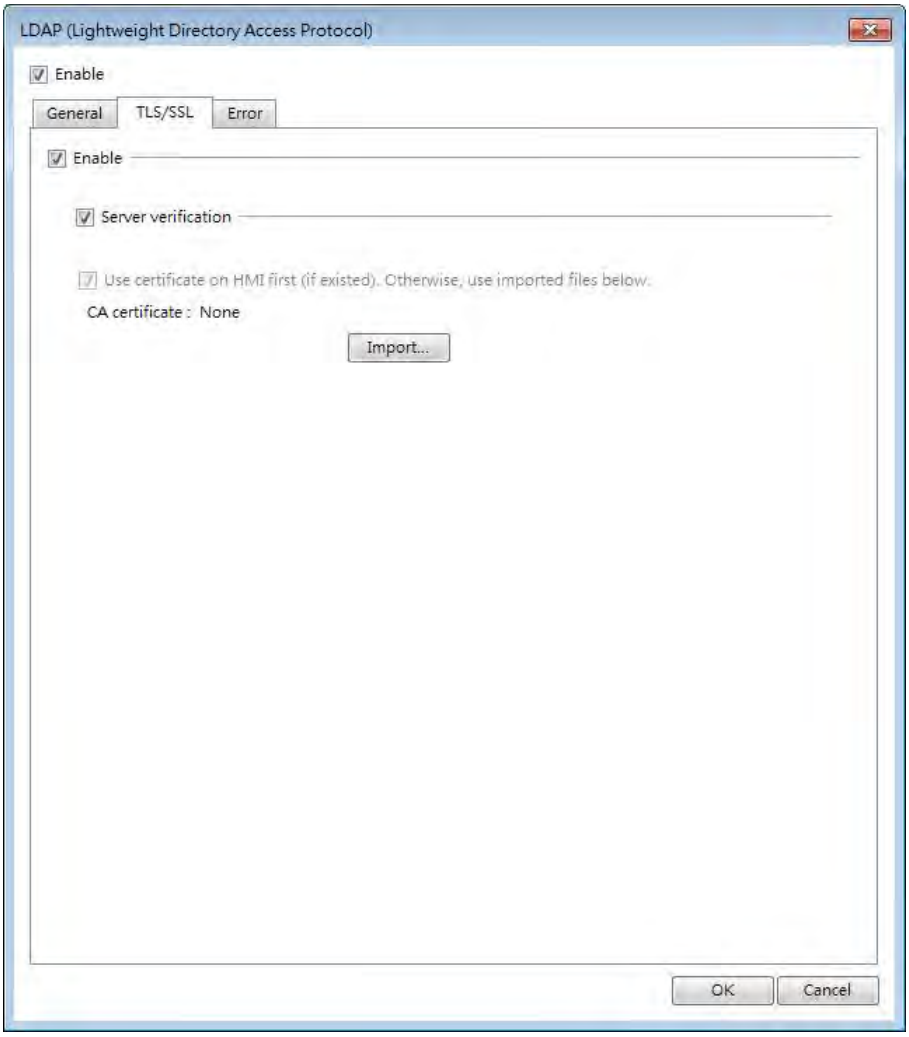
Set LDAP server and operable classes for each group.



Setting	Description
Host	Set the IP address of the host or use domain name.
Port	By default the port number is: LDAP: 389 LDAPS: 636
Base DN	LDAP server's domain name (DN).
User base DN	Organizational units (OU) that hold users.
Group base DN	Organizational units (OU) that hold groups.
New	Add a new group.
Delete	Delete a group.
Import from Server	Log in LDAP server using user credentials to import all allowable groups.
Group Name and Class	Select the operable classes for each group. The group name can be 64 words in maximum, case-sensitive, and allows letters / numbers / symbols / Unicode.

10.4.5.2. TLS/SSL Tab

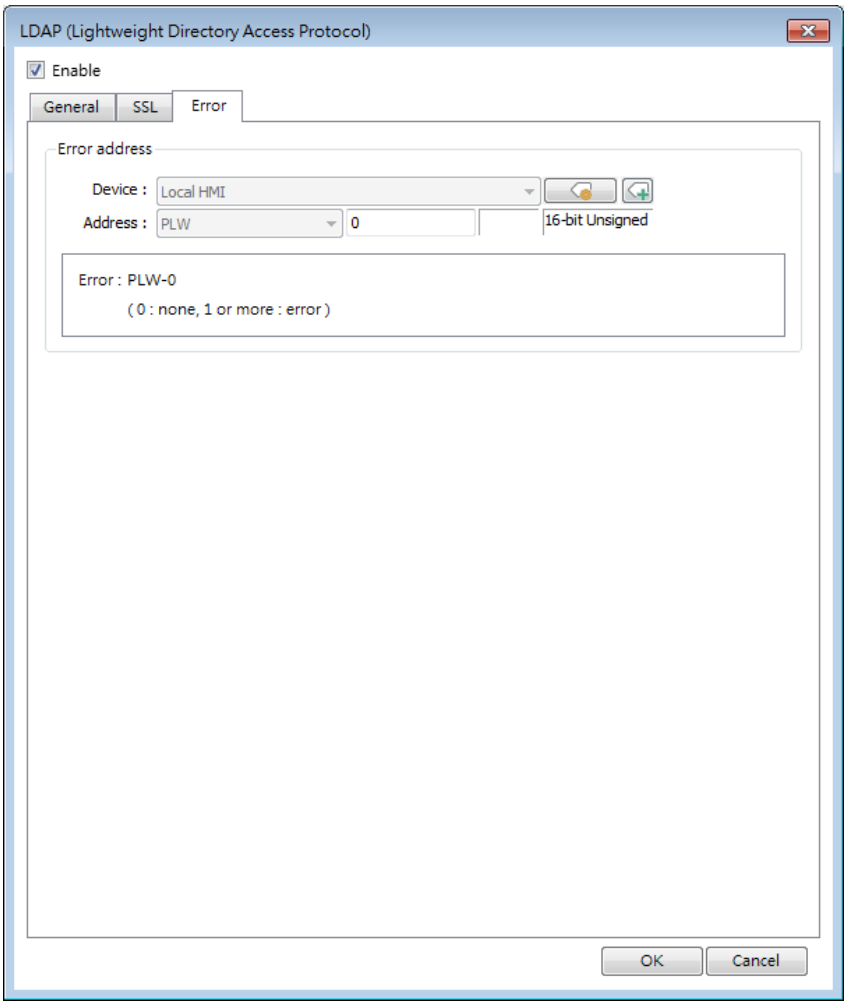
Enable settings in this tab for LDAPS (LDAP over SSL) connection with the AD server.



Setting	Description
Enable	Enable TLS/SSL security for secured LDAP communication.
Server verification	When establishing connection, the HMI will verify whether the certificate supplied by the server matches the one stored on HMI.
Use certificate on HMI (if existed)…	Use current certificate on HMI or import a new certificate.

10.4.5.3. Error Tab

When LDAP server cannot be connected, an error code shows in the designated address.

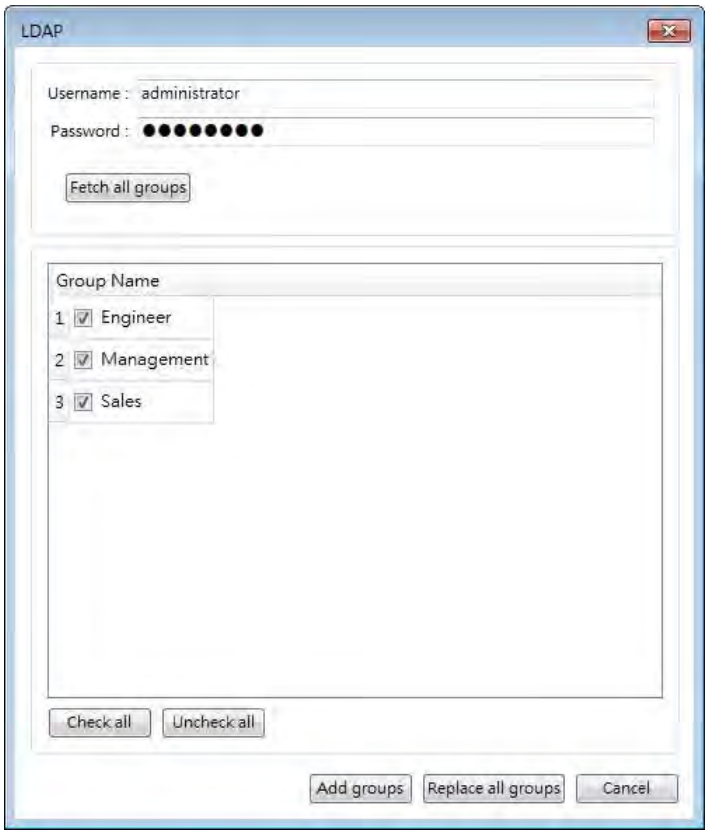


Setting	Description
Error address	The result of login is output to this address.

Value	Description
0	No error
1	Error on LDAP server or no password is entered.
2	Unknown error
257	Remote LDAP server cannot be connected.
258	Wrong username or password.
259	Verification failed
512	Unknown TLS
513	Domain name does not match CN.

10.4.5.4. LDAP Settings (Import from Server)

Get group information from LDAP server.



Setting	Description
Username	Log in LDAP Server using username.
Password	Log in LDAP Server using password.
Fetch all groups	Fetch all groups of the DN in LDAP server.

Error Message	Description
Can't contact LDAP server	LDAP server cannot be connected.
Invalid Credentials	Wrong username or password used for login LDAP server.
Unknown	Error on LDAP server or no password is entered.

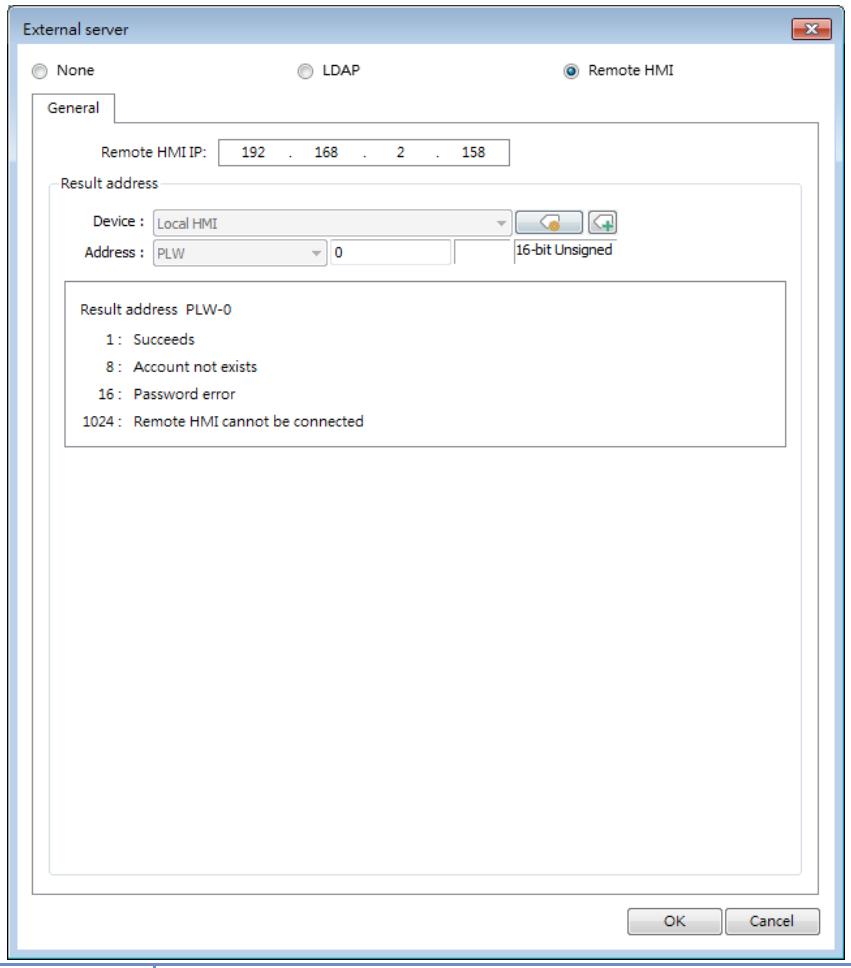
Note

- The maximum number of groups allowable in LDAP mode is 128 groups. When importing from LDAP server, the system will check the number of groups in LDAP server first, exceeding 128 groups will result in unsuccessful import.

- Importing duplicate group name will not clear the operable classes of that group.

10.4.6. Remote HMI Mode

In this mode, user accounts can be managed on a remote HMI, instead of the local HMI. The accounts on a remote HMI can be used to log in the local HMI; therefore, managing the accounts on the local HMI is not necessary.



Setting	Description
Remote HMI IP	The IP address of the remote HMI that holds the user accounts.
Result address	When an error occurs while trying to authenticate the account or connect to the remote HMI, the corresponding error code will be output to the designated result address.

Note

- The accounts can be authenticated via both local and remote HMI. When the same account exists on both local and remote HMI, the authentication is done via the local HMI, instead of the remote HMI. As shown below, user Angela will be authenticated via local

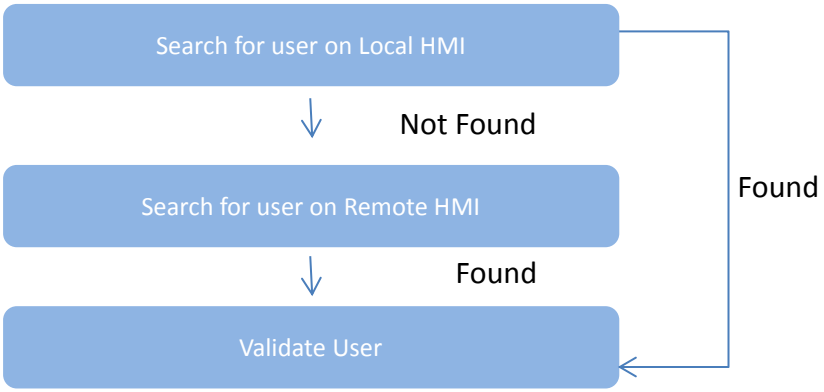
HMI.

Accounts on Remote HMI

	Enable	Secret user	User name	Password	
1	☑	☐	Angela	111	weak
2	☑	☐	Bella	222	weak
3	☑	☐	Gigi	333	weak

Accounts on Local HMI

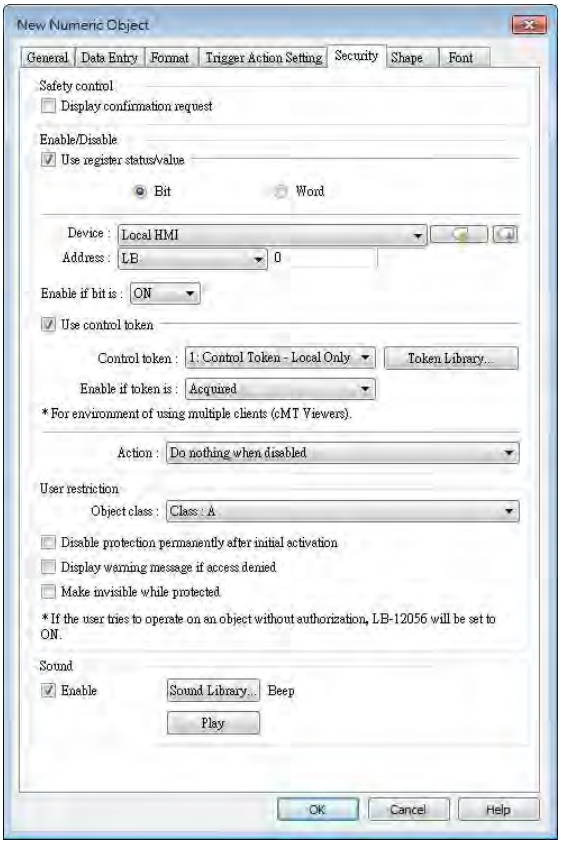
	Enable	Secret user	User name	Password	
1	☑	☐	Angela	1	weak
2	☑	☐	Amy	2	weak
3	☑	☐	Allen	3	weak



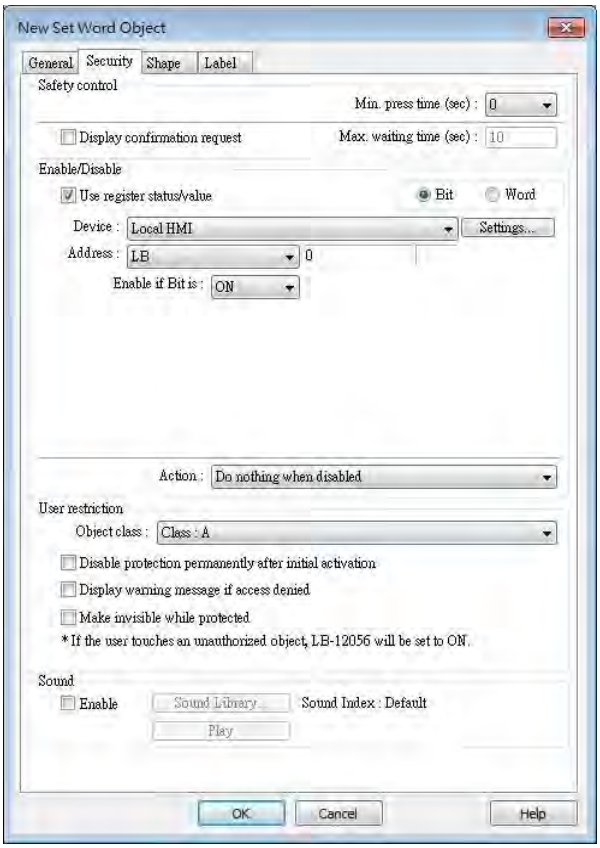
10.5. Object Security Settings

Settings in the Security tab allow users to configure conditions so that the object is operable when the condition is met. The sound emitted when operating the object can be selected.

cMT, cMT X Series

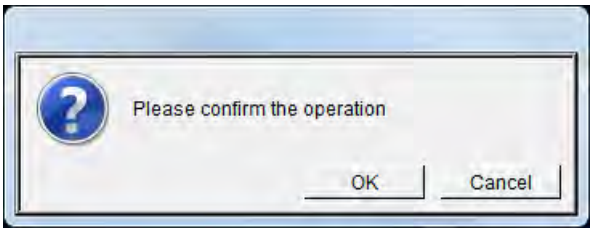


eMT, iE, XE, mTV Series



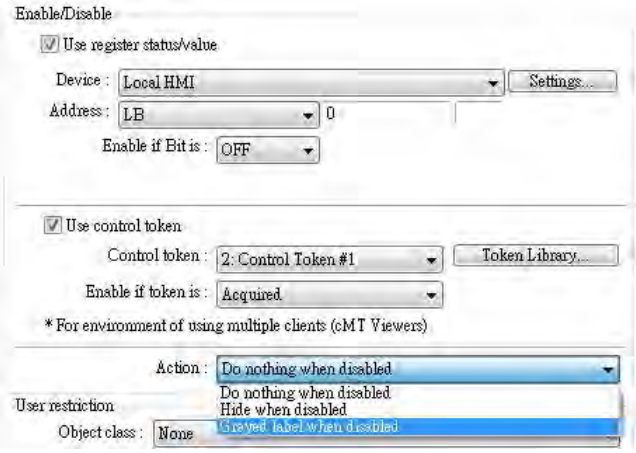
10.5.1. Security Tab

Setting	Description
Min. press time (sec)	Press and hold the object for longer than the [Min. press time] set here to activate the object.
Display confirmation request	After pressing the object, a dialog appears for operation confirmation. If the response to this dialog comes later than the set [Max. waiting time (sec)], this dialog disappears automatically and the operation will be canceled.

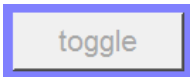
	

10.5.2. Enable/Disable

When [Use register status/value] or [Use control token] is selected, whether the object is operable is determined by the status of the designated address or acquisition the control token, respectively. As shown in the following figure, only when LB-0 is in OFF state and "2: Control Token" is acquired will this object be operable.

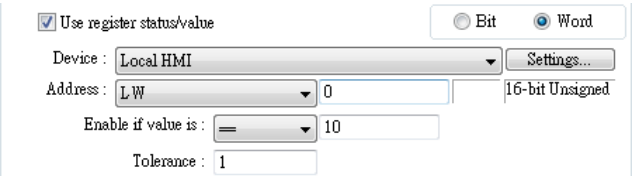


The following table describes the action this object will take when it's token is not acquired.

Setting	Description
Do nothing when disabled	When the control token is not acquired, the object is displayed.
Hide when disabled	When the control token is not acquired, the object is hidden.
Grayed label when disabled	When the control token is not acquired, the label of the object turns gray. 

10.5.2.1. Use Register Status/Value

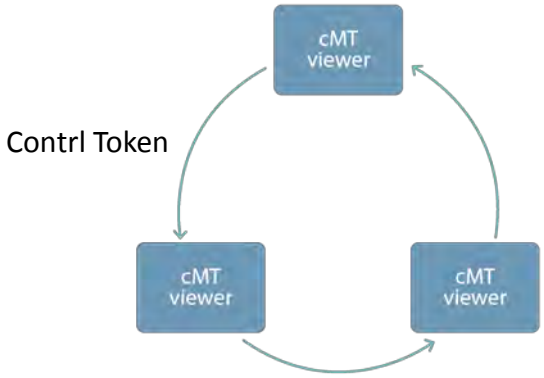
When selected, the status of the designated bit/word address determines whether the object is operable.

Setting	Description
Bit	The object is operable when the designated bit is in On/Off state.
Word	When [Use Register Status/Value] and [Word] are both selected, the status of a designated word address determines whether the object is operable. **Enable if value is:** >, <, ==, <>, >=,<= When the value in the word address reaches the condition specified here, the object is operable. **Tolerance:** This setting is available for <> and ==. **<>**: The object will be operable when: value in address > [value in address + tolerance] or value in address < [value in address - tolerance] **==**: The object will be operable when: value in address is between [value in address + tolerance] and [value in address - tolerance] (including value in address ± tolerance) For example:  When the value in the designated word address is between 9~11, the object is operable. Please note that [Word] option is only available for Set Word and Numeric objects.

Note

- Word objects supported on cMT/cMT X Series include: Set Word, Numeric, ASCII, Combo Button.
  Word objects supported on iE/XE/eMT/mTV Series include: Set Word, Numeric.

10.5.2.2. Control Token



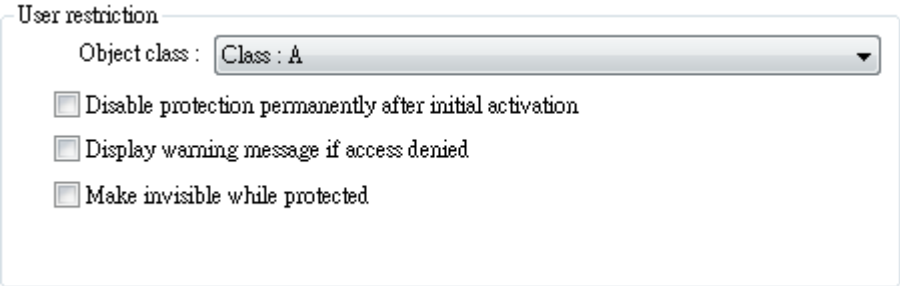
One cMT / cMT X HMI can be simultaneously controlled by multiple cMT Viewer clients. To ensure system safety by preventing an object to be controlled by multiple clients simultaneously, a control token can be set. Only one cMT Viewer client can acquire the control token at a time, and only the cMT Viewer client that acquires control token can operate the object. The rest of the clients can acquire the token one by one when the token is not occupied.

The applicable objects include: Combo Button, Numeric, ASCII, Direct Window, and Indirect Window.

Setting	Description
Control Token	Select a control token for the object.
Token Library…	Add/delete control token. For more information, please see “Chapter 34 Control Token” in this user manual.
Enable if Token is	When [acquired] is selected, only the device that obtains the control token can operate the object. When [unacquired] is selected, only the devices that do not obtain the control token can operate the object.

10.5.3. User Restriction

Set the security class of the object to be operated by an authorized user.

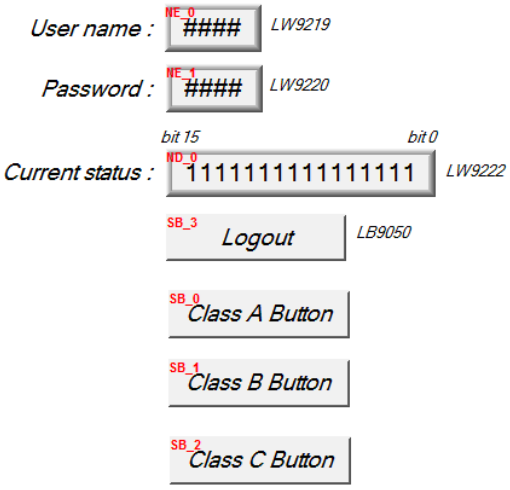


Setting	Description
Object class	"None" means any user can operate this object. Only account "admin" can operate "Administrator" object class.
Disable protection permanently after initial activation	Once the permitted class of the user matches that of the object, the system will stop checking the security class permanently, that means, any user can operate this object freely after it is unlocked.
Display warning message if access denied	When an unauthorized user attempts to operate the object, a warning dialog (Window no. 7) appears. The content of the message in the dialog can be modified.
Make invisible while protected	When the user's privilege does not match the object class, the object will be hidden.

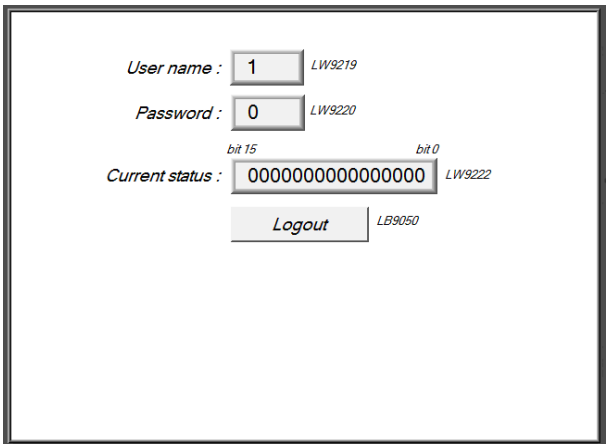
10.6. Example of Object Security Settings

The following shows an example of setting object security class:

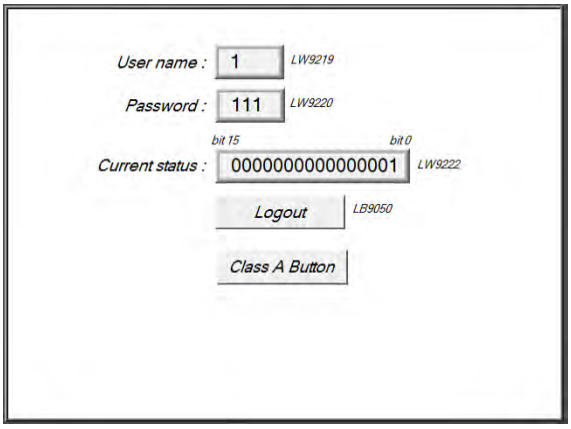
1. Create a project, go to [System Parameter Settings] » [Security] » [General] to enable 3 users:
   User 1 = Operable class: A
   User 2 = Operable class: A, B
   User 3 = Operable class: A, B, C
2. Design Window no. 10 as shown:



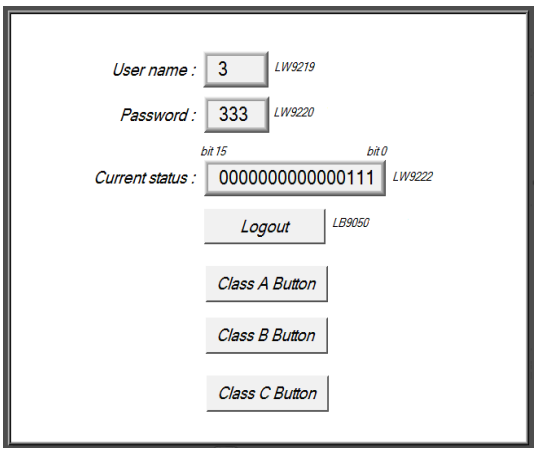
   Create two [Numeric Input] objects:
   [LW-9219] User no. (1~12), Length = 1word
   [LW-9220] For entering user password. Length = 2 words
   Create a [Numeric Display] object:
   [LW-9222] Displays the operable object class of current user. (16-bit Binary)
   Create a [Set Bit] object
   [LB-9050] logout
   Create three [Set Bit] objects:
   Each set to different classes but all select [Made invisible while protected].
3. After setting, please save and compile the project and execute off-line simulation. The below shows how it works when simulating.



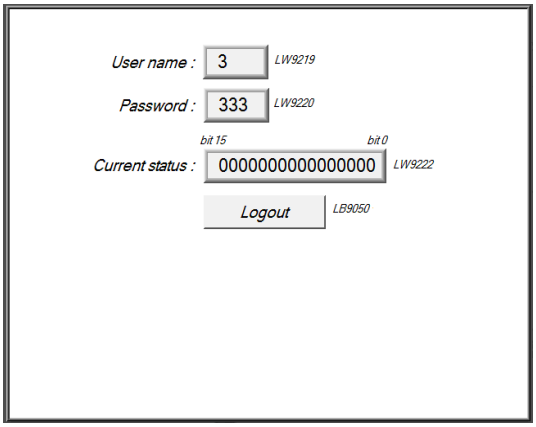
Before entering the password, it displays “0000000000000000”, which means that the user operable object class is “None”. [Class A Button] ~ [Class C Button] objects are classified from “A” to “C” and selected [Made invisible while protected]; therefore they are hidden at this moment.



Enter User 1 password “111”.
Since User 1 is only allowed to operate class A objects, [Class A Button] object appears for operating. [LW-9222] bit 0 turns to “1” means that user can operate class A objects.



Enter User 3 password “333”.
Since User 3 is allowed to operate class A, B, C objects, [LW-9222] bit 0 ~ bit 2 turns to “1”, means that user can operate class A ~ C objects.



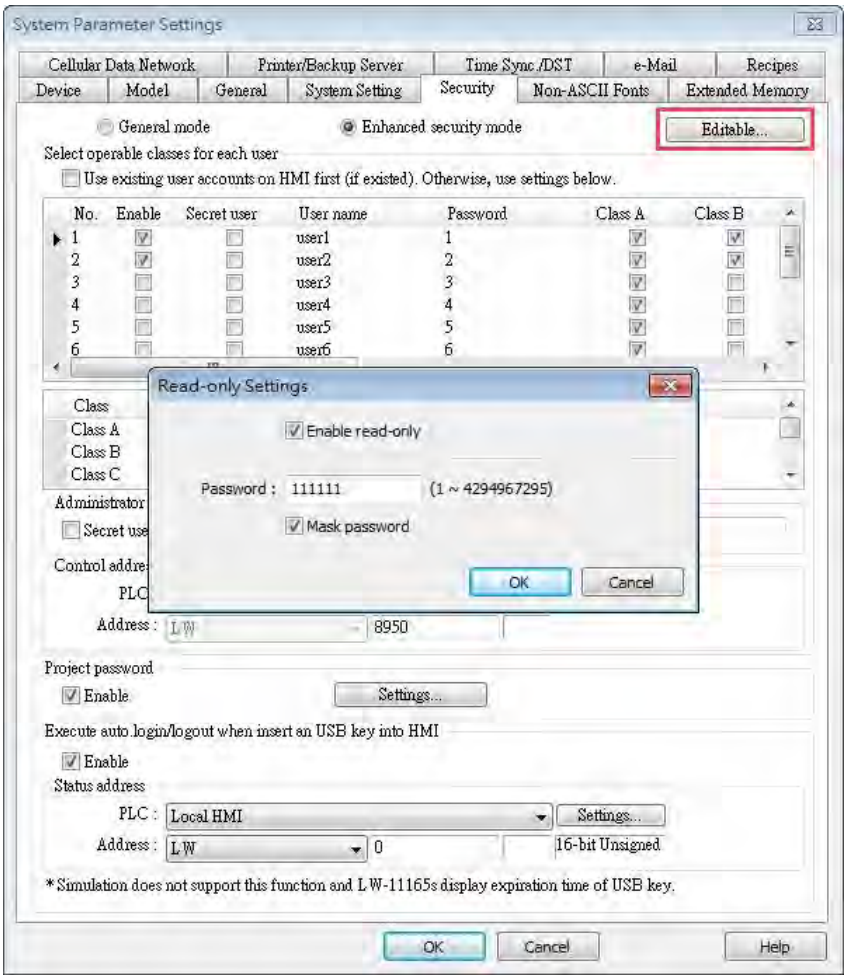
Click [Logout] button to log out, the system will return to the initial state, and current user can only operate class “None” objects.

Note

- Password input: If the password is incorrect, [LB-9060] will be ON; if the password is correct, [LB-9060] will be OFF. All user passwords (User 1 to User 12) can be obtained from system registers [LW-9500] ~ [LW-9522], 24 words in total.
- Changing password directly on HMI: When [LB-9061] is set ON, the system will read data in [LW-9500] ~ [LW-9522] to update user password. The new password will be used in future operations. Please note that the user operable object classes will not be changed due to the change of password.

10.7. Protecting Password Settings from Unauthorized Editing

Before sending the project to others who may edit the project afterwards, it is recommended to click [Editable] button in Security settings tab to open read-only mode. This mode can protect password settings from unauthorized editing.



When [Enable read-only] is selected, a password will be required for changing security settings in the project.

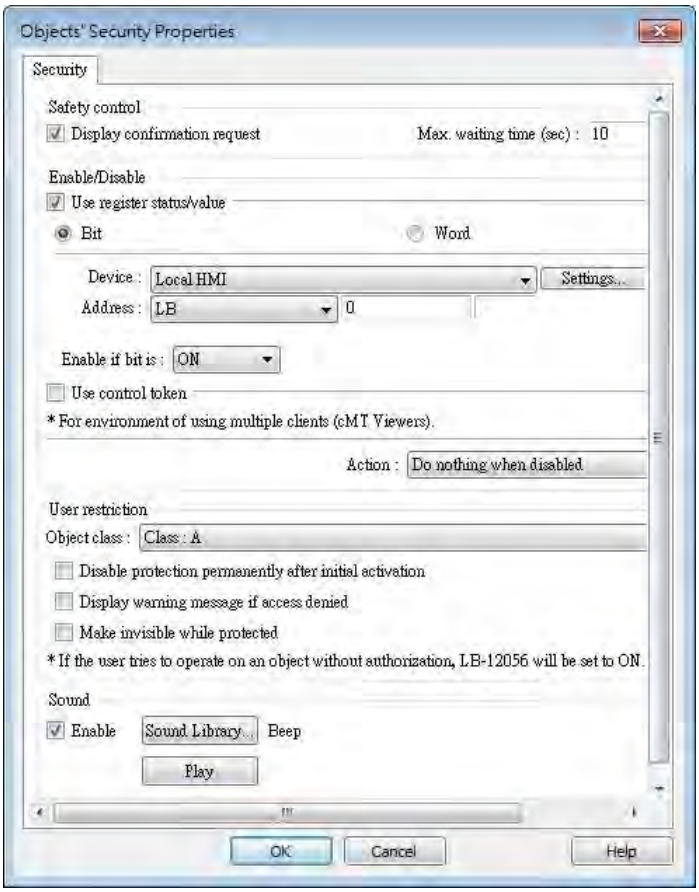
When [Mask password] is selected, passwords will be masked by asterisks (*).

Note

- The protected projects cannot be decrypted since they are encrypted by users, therefore, please remember your password.

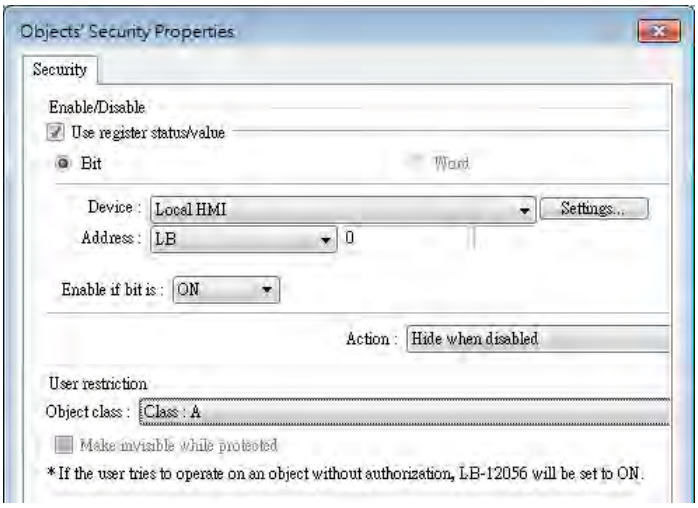
10.8. Bulk Changing of Security Settings of Multiple Objects

Selecting multiple objects in a group and then selecting [Security settings] in the right-click menu can open an Objects' Security Properties window that allows users to change the security settings of all the selected objects at a time.



Note

- When the selected objects have different security settings pages, Objects' Security Properties will automatically adjust and show the settings that users are allowed to change. The rest of the settings will be hidden or greyed out. The following is a window that shows when selecting a Bit Lamp and a Numeric object.


Objects' Security Properties
Security
Enable/Disable
Use register status/value
Bit
Word
Device :
Local HMI
Settings...
Address :
LB
0
Enable if bit is :
ON
Action :
Hide when disabled
User restriction
Object class :
Class : A
Make invisible while protected
* If the user tries to operate on an object without authorization, LB-12056 will be set to ON.

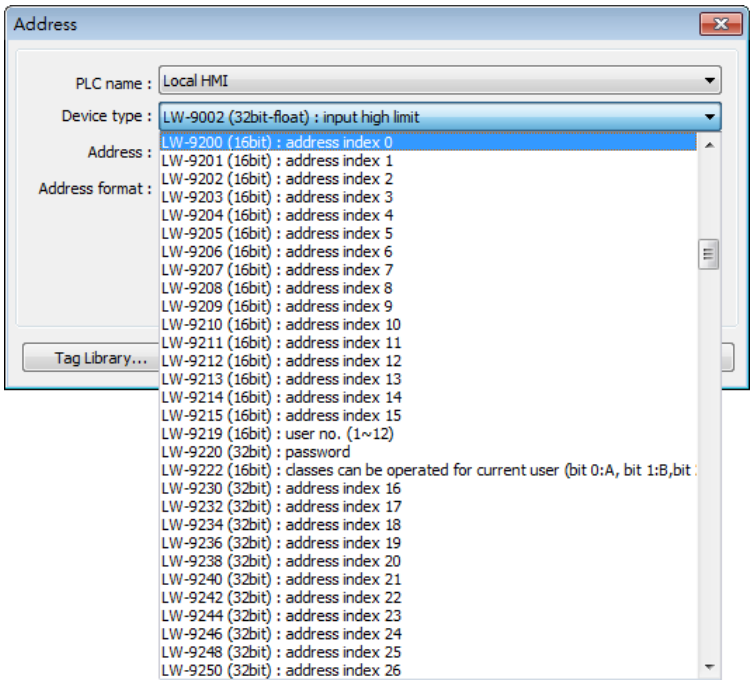
11. Index Register

This chapter explains how to use Index Register.

11.1. Overview 11-2
11.2. Examples of Index Register 11-3
11.3. Examples of Using Tag-Based PLC and Index Registers 11-5

11.1. Overview

EasyBuilder Pro provides Index Registers for changing addresses flexibly. With Index Registers, user can change the object's read/write address directly on HMI without changing its settings. There are 32 Index Registers, divided into 16-bit and 32-bit. cMT / cMT X models offer 32 Private Index Registers, also divided into 16-bit and 32-bit.



The corresponding address of 16-bit Index Register 0 to 15: LW-9200 (16bit) to LW-9215 (16bit)
The maximum offset range is 65536 words.
The corresponding address of 32-bit Index Register 16 to 31: LW-9230 (32bit) to LW-9260 (32bit)
The maximum offset range is 4294967296 words.
The corresponding address of 16-bit Private Index Register 0 to 15: PLW-9200 (16bit) to PLW-9215 (16bit)
The maximum offset range is 65536 words.
The corresponding address of 32-bit Private Index Register 16 to 31: PLW-9230 (32bit) to PLW-9260 (32bit)
The maximum offset range is 4294967296 words.

When using [Index register], the address is designated by the following equation:
The constant set in [Address] + the value in the chosen Index Register.

The way to use Private Index Registers is the same as that of Index Registers. The difference is

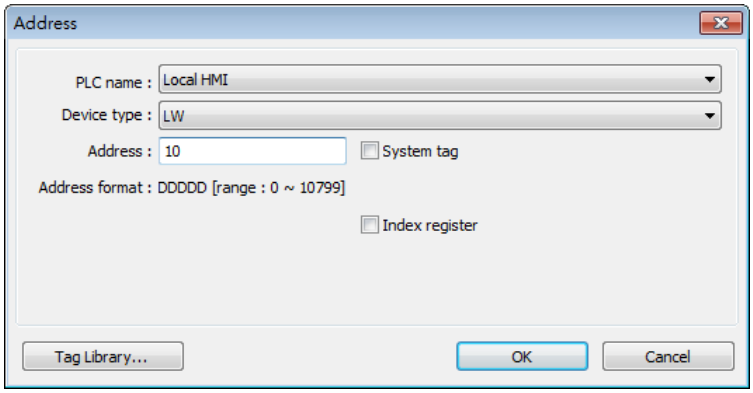
that Private Index Registers work for objects in cMT Viewer app, which runs independently on each client device; therefore, operations in this app will not make the HMI show the same screen.

Note

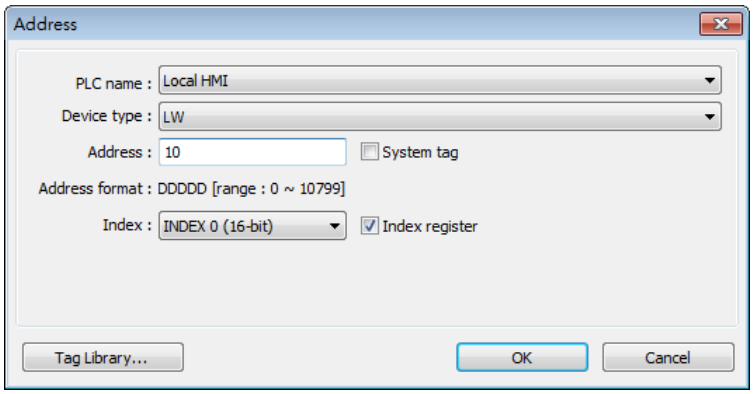
- Index Registers work for the **Word** registers. For **Bit** registers, adding 1 to the value in the Index Register, the offset is 16 bits.
- Index Registers do not work for PLB and PLW registers.

11.2. Examples of Index Register

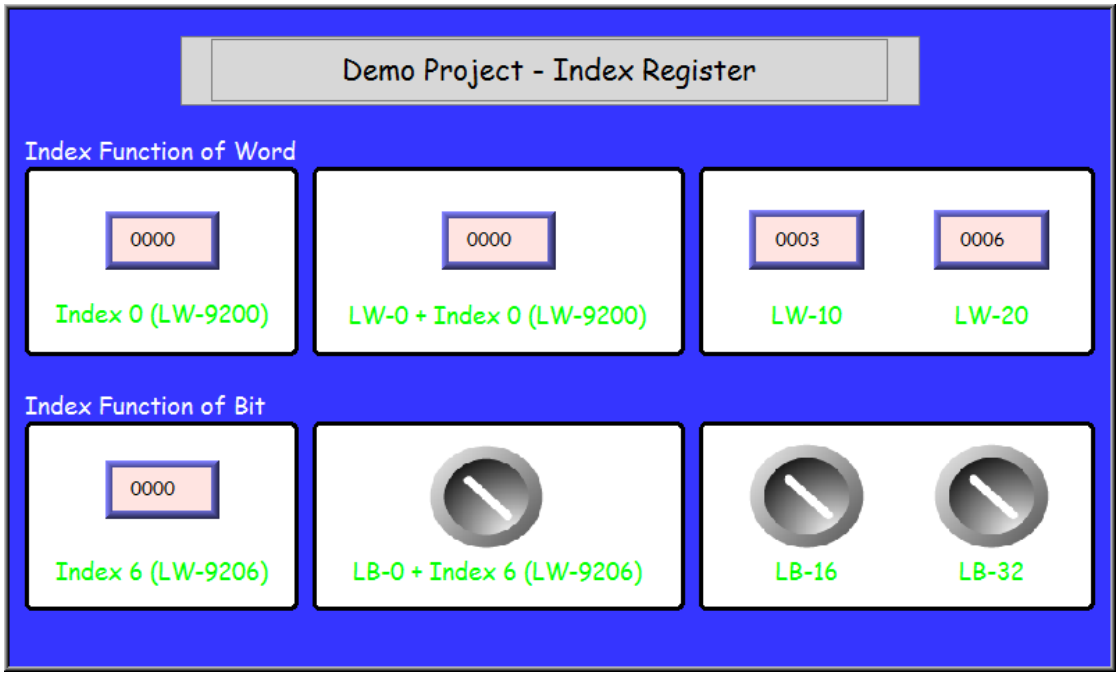
The following explains the way to designate the register while Index Register is used.
If **not** selecting **[Index register]** check box and set address to [LW-10]. The system will directly read / write LW-10.



If select **[Index register]** check box and set [Index] to [INDEX 0 (16-bit)], the system will read / write [LW(10 + value in Index Register 0)].
If the data in [LW-9200] is “5”, the designated address is [LW(10+5)] = [LW-15].

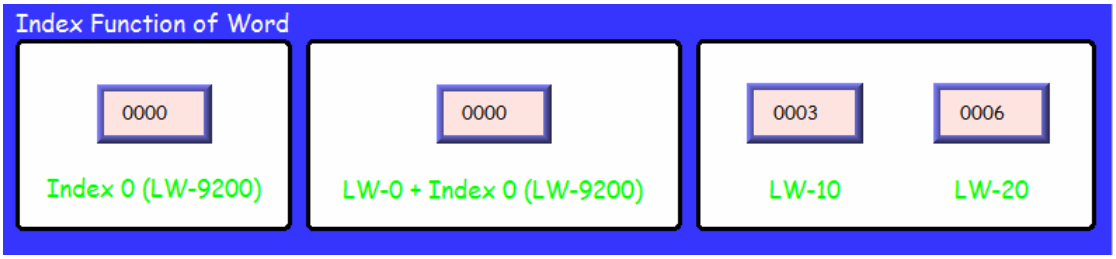


Here’s a demo project shown as an example:

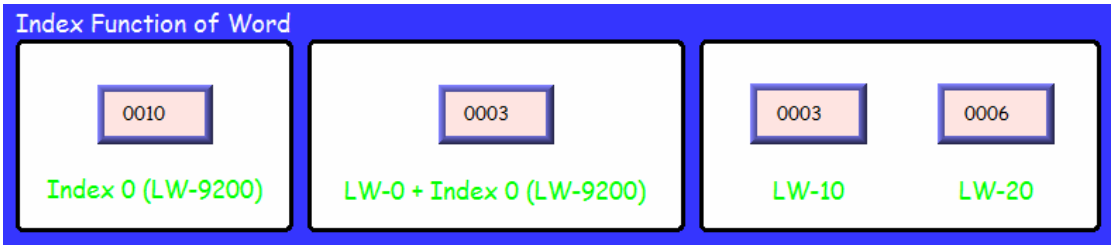


Example 1

The following shows an example of using a Word register and select [Index register]. If the value in [LW-0] is 0, in [LW-10] is 3, and in [LW-20] is 6, the result is:



If the value in Index 0 (LW-9200) is 0, then [LW0 + Index 0] = read [LW-0].



If the value in Index 0 (LW-9200) is 10, then [LW0 + Index 0] = read [LW-10] = 3.

Example 2

The following shows an example of using a Bit register and select [Index register].

If the state of [LB-16] is ON, and the state of [LB-32] is OFF.

Since 1 Word equals to 16 Bit, adding 1 in Index Register, the offset is 16 bits.

If Index 6 (LW-9206) is set to 1, then switch [LB-0 + Index6] reads LB-16 which is in ON state.

If Index 6 (LW-9206) is set to 2, then switch [LB-0 + Index6] reads LB-32 which is in OFF state.

Note

- When using Index Registers for Bit register, the offset is 16 bits. For example, if the Bit register is LB-0, and set the value in Index Register to 1, then LB-16 will be activated. If set the value in Index Register to 2, then LB-32 will be activated.

Click the icon to download the demo project. Please confirm your internet connection.

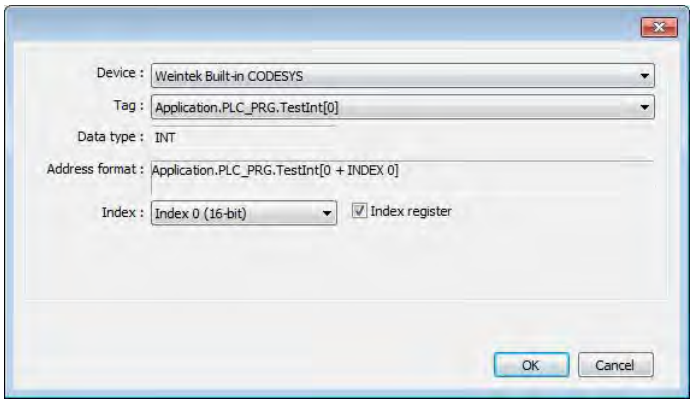
11.3. Examples of Using Tag-Based PLC and Index Registers

Index Registers only support one-dimensional array variables.

Example 1: INT

If **[Index register]** is enabled and [Index] is set to [Index 0 (16-bit)], the system will read / write "Application.PLC_PRG.TestInt[0 + value in Index Register 0]".

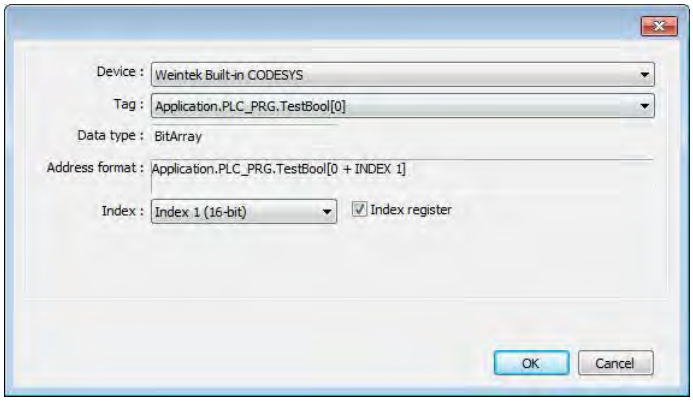
If the data in [LW-9200] is "5", the actual address is "Application.PLC_PRG.TestInt[5]".



Example 2: Bit

If **[Index register]** is enabled and [Index] is set to [Index 1 (16-bit)], the system will read / write "Application.PLC_PRG.TestBool[0 + value in Index Register 1]".

If the data in [LW-9201] is "7", the designated address is "Application.PLC_PRG.TestBool[7]".



Note that, unlike its non-tag counterpart, the 16-bit offset is not necessary for Bit array tags.

12. Keyboard Design and Usage

This chapter explains how to design and use keyboard in EasyBuilder Pro.

12.1. Overview 12-2
12.2. Steps to Design a Popup Keyboard 12-2
12.3. Steps to Design a Keyboard with Direct Window 12-5
12.4. Steps to Design a Fixed Keyboard on Screen 12-6
12.5. Steps to Design a Unicode Keyboard 12-7

12.1. Overview

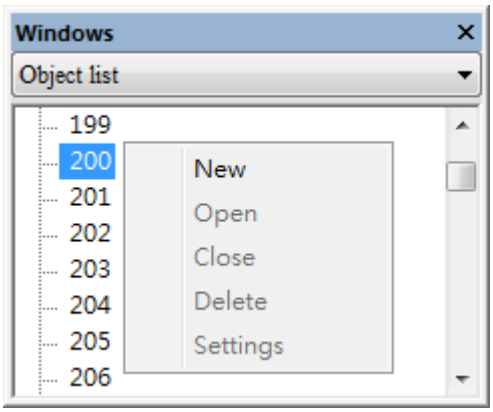
Numeric Input and ASCII Input objects need keyboard as an input tool. Both numeric keyboard and ASCII keyboard are created with Function Key object. Apart from the keyboards provided by EasyBuilder Pro, you can create the keyboard if needed. System keyboard is available on cMT / cMT X Series models.

The types of the keyboards are:

- Popup Keyboard (with or without title bar)
- Fixed Keyboard
- Unicode Keyboard
- System Keyboard (available on cMT / cMT X models)

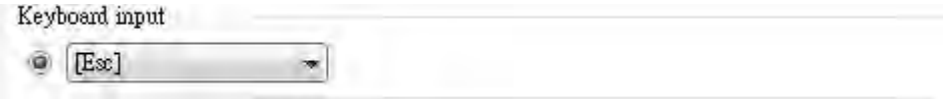
12.2. Steps to Design a Popup Keyboard

1. Create and open a window for the new keyboard. For example, set to “window no. 200”.

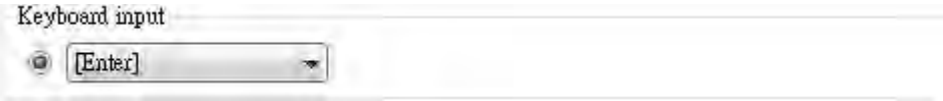


2. Adjust the height and width of “window no. 200” and create a variety of Function Key objects in [ASCII/Unicode mode].

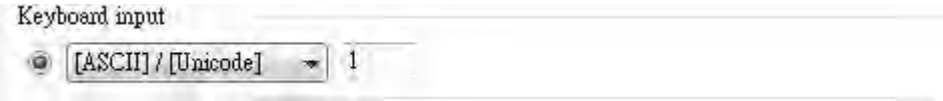
Set one of the Function Key objects as the [Esc] key.



Set another Function Key object as the [Enter] key.



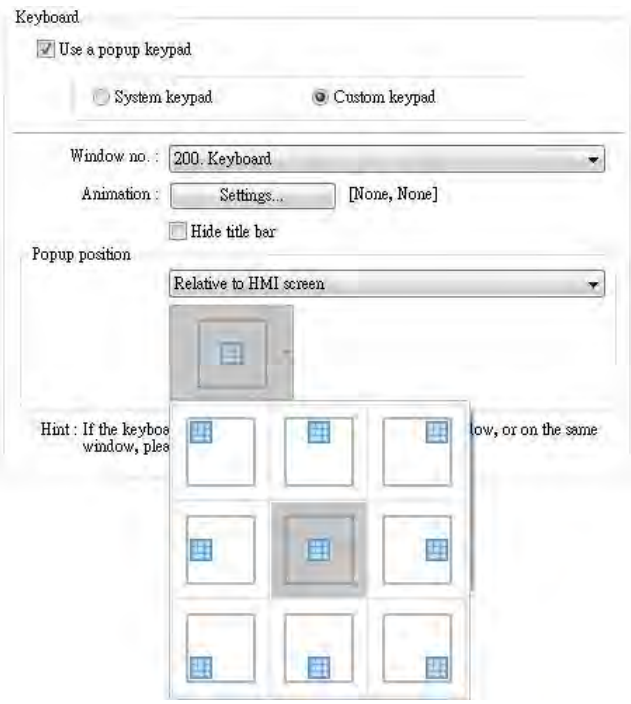
The rest are mostly used to enter numbers.



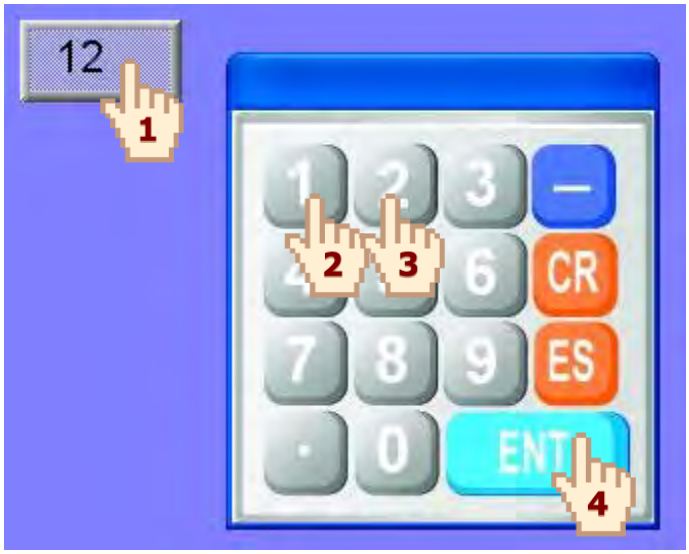
3. Select a suitable picture for each Function Key object.

4. Select [System Parameter Settings] » [General] » [Keyboard] » [Add] to add “window no. 200”. Up to 32 keyboards can be added.

5. After the keyboard window is added, when you create Numerical Input and ASCII Input objects, “200. Keyboard” can be found in [Data Entry] » [Keyboard] » [Window no.]. The [Popup position] is for designating the display position of the keyboard on the HMI screen. The system divides the screen into 9 areas.

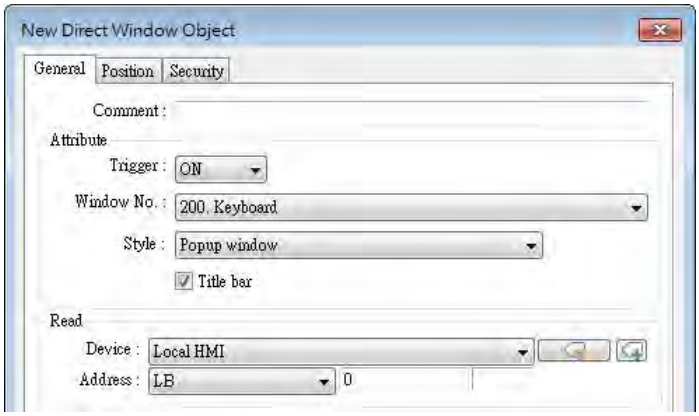


6. Select “200.Keyboard”. When you press Numerical Input or ASCII Input objects on the screen, “window no. 200” will pop up. You can press the keys on the keyboard to enter data.

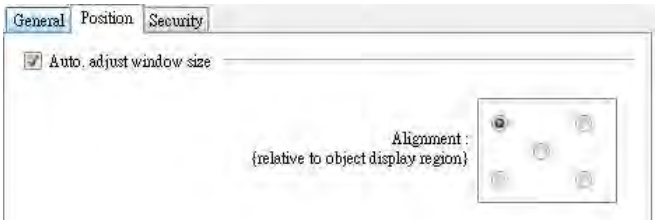


12.3. Steps to Design a Keyboard with Direct Window

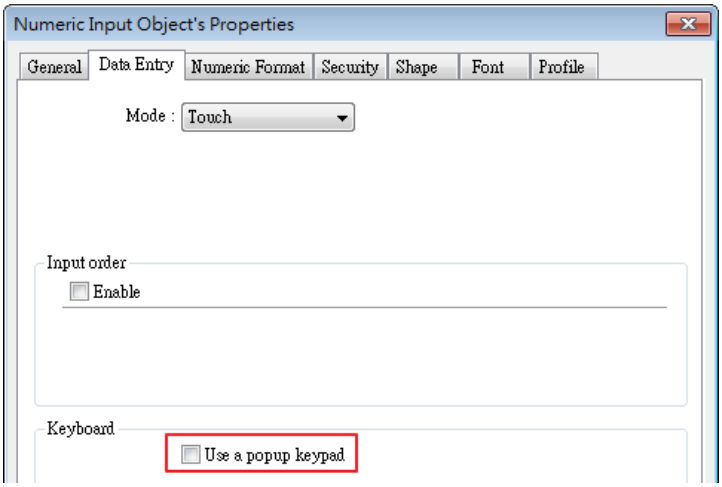
1. Create a Direct Window object and set a read address to activate it.
   In [General] » [Attribute] select [Title bar] and the correct [Window No.].



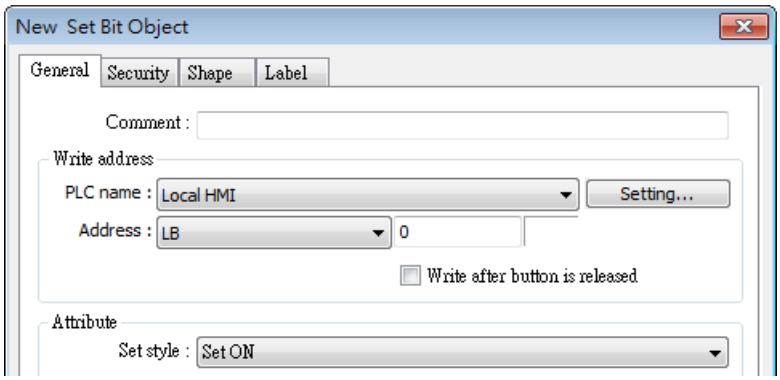
2. Open the [Position] tab and select [Auto. adjust window size].



3. Create a Numeric Input object, and don't select [Use a popup keypad] check box.



4. Create a Set Bit object, set address to [LB-0] and set [Set style] to [Set ON]. Overlay it on the Numeric Input object. Pressing the Numeric Input object will open the keyboard window.



5. Add Set Bit objects on the [Enter] and [ESC] Function Key objects respectively. Set address to [LB-0] and [Set style] to [Set OFF]. In this way when pressing either [Enter] or [ESC] key will close the keyboard window.

12.4. Steps to Design a Fixed Keyboard on Screen

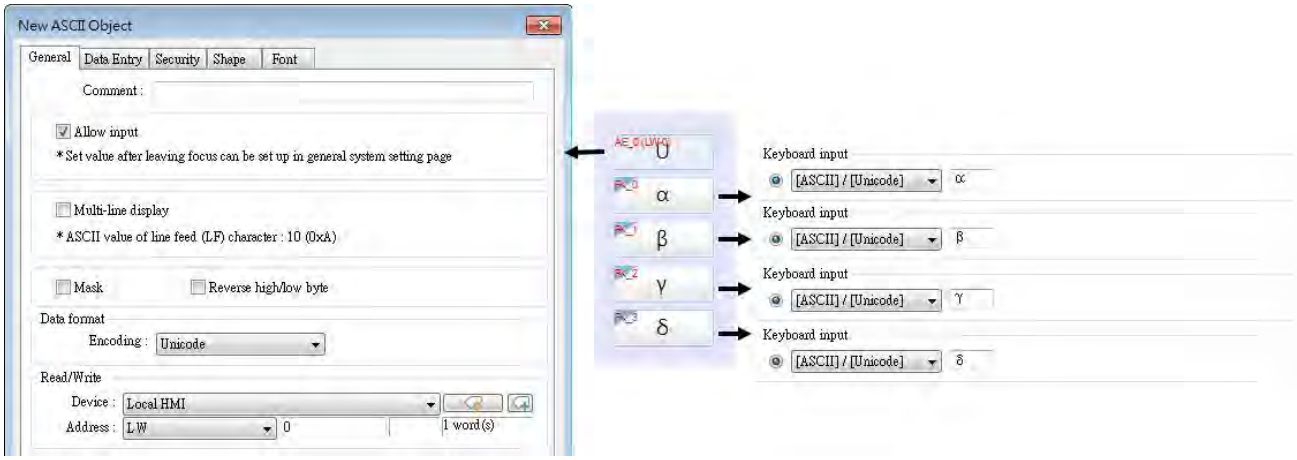
You can also place a fixed keyboard on the screen instead of popup keyboard or Direct Window. This type of keyboard can't be moved or closed.

1. Create a Numeric Input object, in [Data Entry] » [Keyboard] don't select [Use a popup keypad] check box.
2. Use Function Key objects to design the keyboard and place it on the screen.
3. Press the Numeric Input object and enter a value with Function Key objects directly.

12.5. Steps to Design a Unicode Keyboard

The following steps explain how to create a Unicode keyboard with Function Key objects.

1. Place an ASCII Input object on the window and select [Unicode].
2. Create Function Key objects as shown in the following figure. A simple UNICODE keyboard is created.



Note

- You can group up the elements of the designed keyboard and save to Group Library for future use.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13. Objects

This chapter explains how to use different objects.

13.1. Bit Lamp 13-3
13.2. Word Lamp 13-5
13.3. Set Bit 13-10
13.4. Set Word 13-13
13.5. Function Key 13-22
13.6. Toggle Switch 13-29
13.7. Multi-State Switch 13-31
13.8. Slider 13-35
13.9. Numeric 13-42
13.10. ASCII 13-63
13.11. Indirect Window 13-67
13.12. Direct Window 13-75
13.13. Moving / Rotating Shape 13-81
13.14. Animation 13-87
13.15. Bar Graph 13-91
13.16. Meter Display 13-100
13.17. Trend Display 13-108
13.18. History Data Display 13-130
13.19. Data Block Display 13-139
13.20. XY Plot 13-147
13.21. Alarm Bar and Alarm Display 13-154
13.22. Event Display 13-162
13.23. Data Transfer 13-175
13.24. Backup 13-183
13.25. Media Player 13-191
13.26. BACnet Schedule 13-197
13.27. PLC Control 13-201
13.28. Scheduler 13-208
13.29. Option List 13-218
13.30. Timer 13-226
13.31. Video In 13-231
13.32. System Message 13-238

13.33. Recipe View 13-240
13.34. Flow Block 13-248
13.35. Operation Log 13-253
13.36. Combo Button 13-266
13.37. Circular Trend Display 13-274
13.38. Picture View 13-284
13.39. File Browser 13-287
13.40. Import/Export 13-291
13.41. Pie Chart 13-296
13.42. Barcode 13-301
13.43. String Table 13-311
13.44. Database 13-313
13.45. Dynamic Scale 13-331
13.46. Dynamic Drawing 13-334
13.47. PDF Reader 13-340
13.48. Table 13-342
13.49. VNC Viewer 13-344
13.50. Contacts Editor 13-349
13.51. Event Bar Chart 13-352
13.52. Action Trigger 13-357
13.53. Calendar 13-364
13.54. Touch Gesture 13-366
13.55. PLC Web Browser 13-368
13.56. Condition 13-372

13.1. Bit Lamp

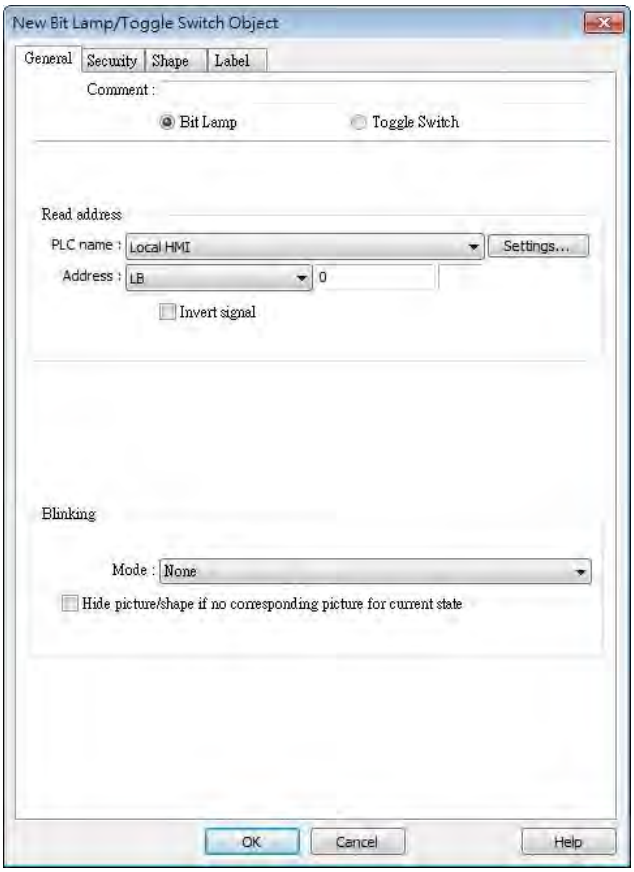
13.1.1. Overview

Bit Lamp object displays the state of a designated bit address. If the bit state is OFF, the State 0 shape will be displayed. If the bit state is ON, the State 1 shape will be displayed.

13.1.2. Configuration

Click [Object] » [Bit Lamp] icon on the toolbar to open a Bit Lamp object property dialog box. Set up the properties, press OK button, and a new Bit Lamp object will be created.

General Tab



Setting	Description
Comment	User can describe the information of the object. **Bit Lamp / Toggle Switch** Switch between Bit Lamp and Toggle Switch features.
Read address	Click [Setting] to select the [PLC name], [Address], [Device type], [System tag], [Index register] of the bit device that controls the [Bit Lamp] object. Users can also set address in [General] tab while adding a new object. **Invert signal** Reverses the display of ON / OFF states. For example, if [Invert signal] check box is selected, when the designated bit is OFF, the object displays ON state.
Blinking	The appearance of the object may alternate between states when the bit is ON or OFF. **Mode:** **None** No blinking. **Alternating image on state 0** The appearance of the object alternates between State 0 and 1 when the bit is OFF. **Alternating image on state 1** The appearance of the object alternates between State 0 and 1 when the bit is ON. **Blinking on state 0** The State 0 appearance of the object will blink when the bit is OFF. **Blinking on state 1** The State 1 appearance of the object will blink when the bit is ON. **Hide picture/shape if no corresponding picture for current state** If selected, when there are not enough pictures to represent all the states, hides the picture. Otherwise, displays the last state.

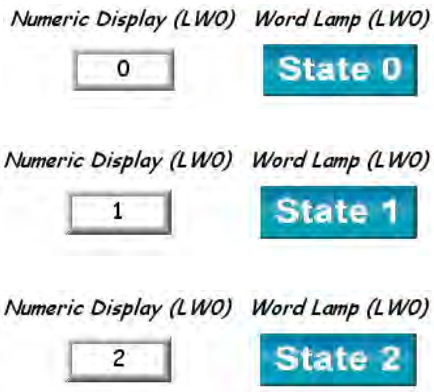
Note

- In [Label] tab, if select [ON=OFF (use state 0)] check box, both state 0 and 1 follow the settings of state 0.

13.2. Word Lamp

13.2.1. Overview

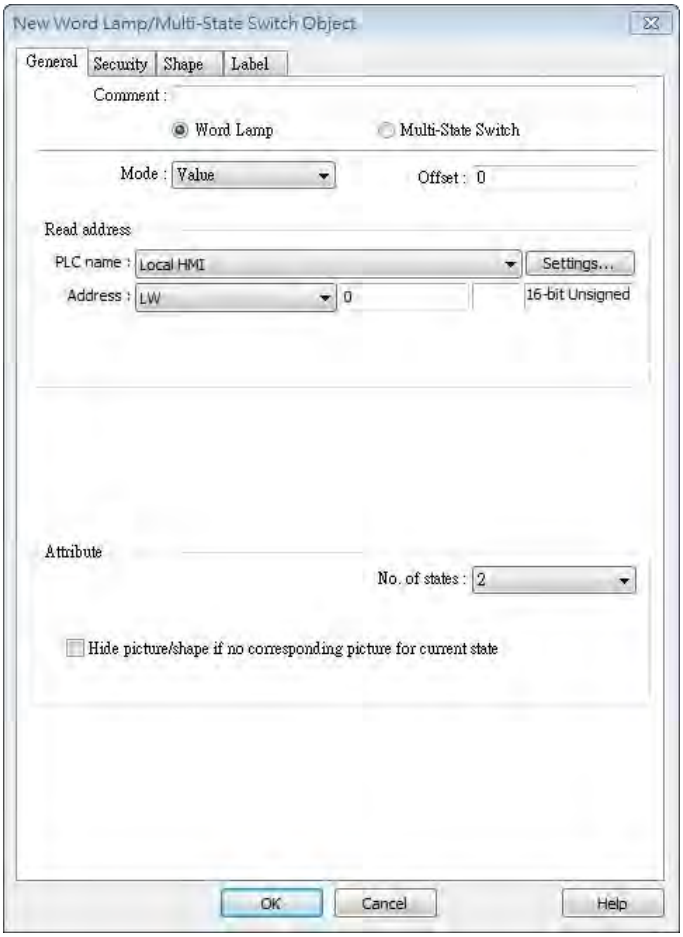
Word Lamp object displays the state according to the value of a designated word register. Up to 256 states are available. When the value of the register is 0, State 0 appearance of the object is displayed, and with the register value being 1 the object displays State 1, and so on.



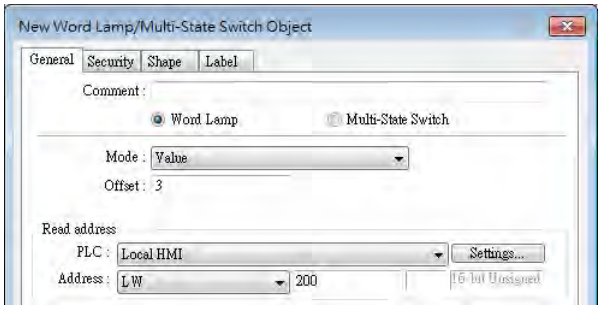
13.2.2. Configuration

Click [Object] » [Word Lamp] icon on the toolbar to open a Word Lamp object property dialog box. Set up the properties, press OK button, and a new Word Lamp object will be created.

General Tab



Setting	Description
Comment	User can describe the information of the object. **Word Lamp / Multi-State Switch** Switch between Word Lamp and Multi-State Switch features.
Mode / Offset	Word Lamp object offers the following three modes: **Value** The state is displayed according to the value in the designated word address and plus the [Offset]. As shown below, if the value within LW-200 is 3, since the offset is set to 3, the shape of state 0 is displayed. (value 3 - offset 3)



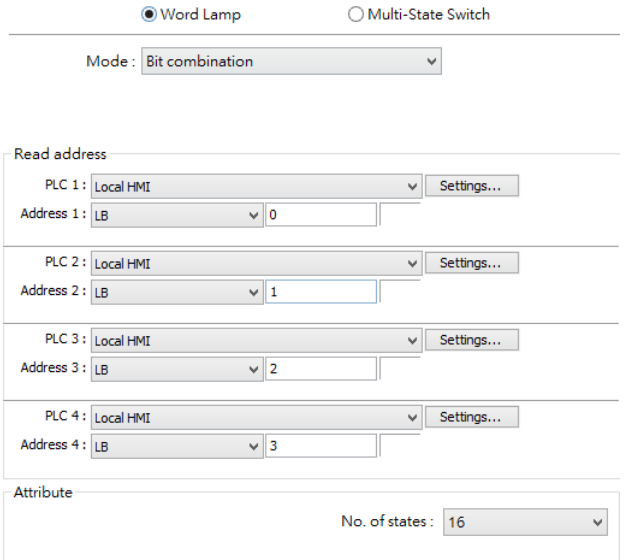
LSB

Convert the value from decimal to binary. The least significant active bit in a binary data word selects the state displayed.

Decimal	Binary	Displayed state
0	0000000000000000	State = 0 when all bits are 0.
1	0000000000000001	The lowest non-zero bit is bit 0, state = 1.
2	0000000000000010	The lowest non-zero bit is bit 1, state = 2.
3	0000000000000011	The lowest non-zero bit is bit 0, state = 1.
4	0000000000000100	The lowest non-zero bit is bit 2, state = 3.
7	0000000000000111	The lowest non-zero bit is bit 0, state = 1.
8	0000000000001000	The lowest non-zero bit is bit 3, state = 4.
16	0000000000010000	The lowest non-zero bit is bit 4, state = 5.
32	0000000000100000	The lowest non-zero bit is bit 5, state = 6.
64	0000000001000000	The lowest non-zero bit is bit 6, state = 7.
128	0000000010000000	The lowest non-zero bit is bit 7, state = 8.
256	0000000100000000	The lowest non-zero bit is bit 8, state = 9.
512	0000001000000000	The lowest non-zero bit is bit 9, state = 10.
1024	0000010000000000	The lowest non-zero bit is bit 10, state = 11.
2048	0000100000000000	The lowest non-zero bit is bit 11, state = 12.
4096	0001000000000000	The lowest non-zero bit is bit 12, state = 13.
8192	0010000000000000	The lowest non-zero bit is bit 13, state = 14.
16384	0100000000000000	The lowest non-zero bit is bit 14, state = 15.
32768	1000000000000000	The lowest non-zero bit is bit 15, state = 16.

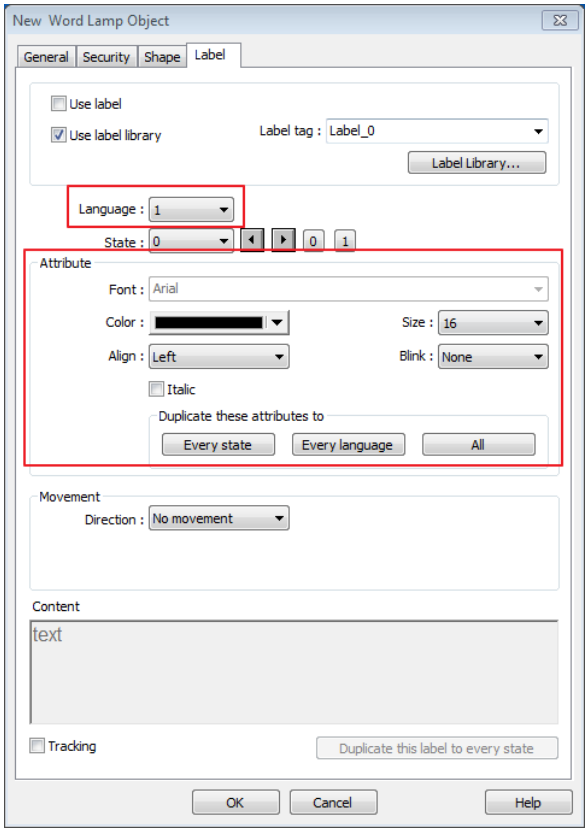
Bit combination

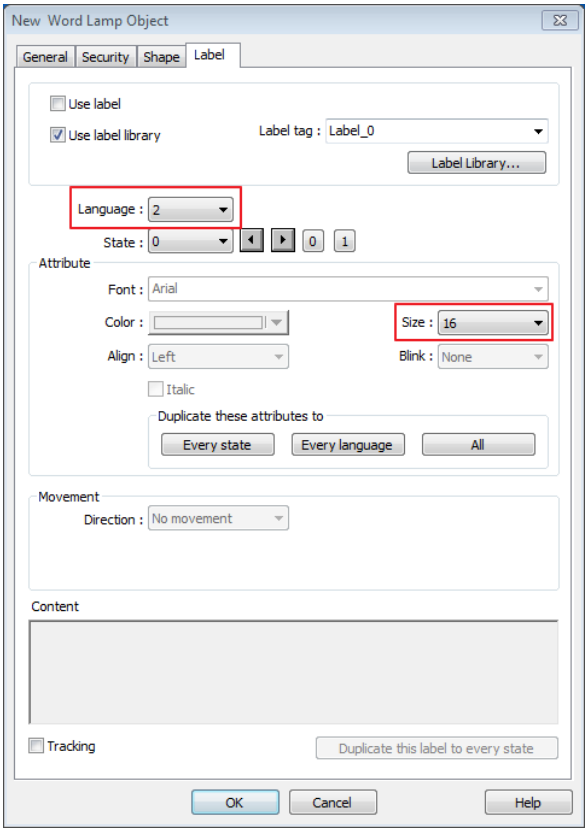
Lamp state depends on the states of bit combinations, where PLC 1

	represents the least significant bit (LSB), PLC 2 represents the next LSB, and so on. Maximum number of bit is 4, for a total of 16 states. Changing [No. of states] in Attribute group box changes the number of read addresses.  **Change state by time** The state displayed changes on a time basis. The frequency can be set.
Read address	Click [Setting] to select the [PLC name], [Address], [Device type], [System tag], [Index register] of the word device that controls the [Word Lamp] object. Users can also set address in [General] tab while adding a new object.
Attribute	**No. of states** The number of states is utilized by the object. The state is numbered from 0, so the number of states minus 1 will be the state number. If the value within the word register is ≥ [No. of states] defined in Attribute, the highest state will be displayed. If the number of states is set to 8, the valid states will be 0, 1, 2, …, 7. In this case if the word value is 8 or higher, the system will display the state 7 shape. **Hide picture/shape if no corresponding picture for current state** If selected, when there are not enough pictures to represent all the states, hides the picture. Otherwise, displays the last state.

Note

- In [Label] tab, Language 1 determines the relevant settings of the font. For Language 2~8, only the font size can be changed and other settings follow Language 1.


New Word Lamp Object
General
Security
Shape
Label
Use label
Use label library
Label tag : Label_0
Label Library...
Language : 1
State : 0
0
1
Attribute
Font : Arial
Color :
Size : 16
Align : Left
Blink : None
Italic
Duplicate these attributes to
Every state
Every language
All
Movement
Direction : No movement
Content
text
Tracking
Duplicate this label to every state
OK
Cancel
Help


New Word Lamp Object
General
Security
Shape
Label
Use label
Use label library
Label tag : Label_0
Label Library...
Language : 2
State : 0
0
1
Attribute
Font : Arial
Color :
Size : 16
Align : Left
Blink : None
Italic
Duplicate these attributes to
Every state
Every language
All
Movement
Direction : No movement
Content
Tracking
Duplicate this label to every state
OK
Cancel
Help

13.3. Set Bit

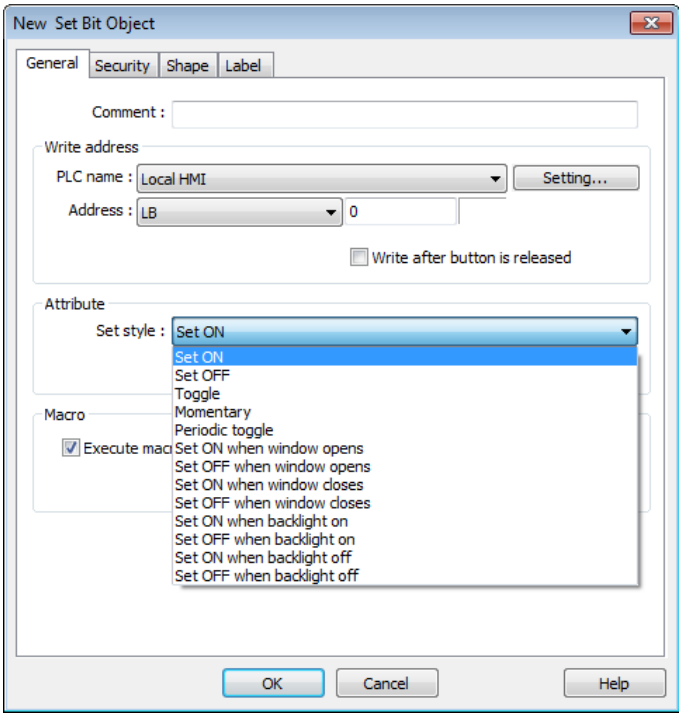
13.3.1. Overview

The Set Bit object provides two operation modes: manual or automatic. Manual mode can trigger a designated bit address to change the state between ON and OFF when the object is touched. In automatic mode, the bit is automatically activated when a pre-defined condition occurs; touching the button will not be effective.

13.3.2. Configuration

Click [Object] » [Set Bit] icon on the toolbar to open a Set Bit object property dialog box. Set up the properties, press OK button, and a new Set Bit object will be created.

General Tab



Setting	Description
Write address	Click [Setting] to select the [PLC name], [Address], [Device type], [System tag], [Index register] of the bit device that controls the Set Bit object. Users can also set address in [General] tab while adding a new object. **Write after button is released**

	If this function is selected, the action is delayed till button is released; otherwise, the action is executed once the button is pressed. This function does not work with momentary buttons.
Mode / Offset	

Set style	Description
Set ON	Set ON the designated bit of the device.
Set OFF	Set OFF the designated bit of the device.
Toggle	Alternates the bit state each time pressed.
Momentary	Holds the bit ON only while button is pressed.
Periodical toggle	Set a designated bit ON and OFF at a set time interval. Time interval can be selected; the range is from 0.1 to 25.5 seconds.
Set ON when window opens	Set ON the bit within the window when the window opens.
Set OFF when window opens	Set OFF the bit within the window when the window opens.
Set ON when window closes	Set ON the bit within the window when the window closes.
Set OFF when window closes	Set OFF the bit within the window when the window closes.
Set ON when backlight on (N/A for cMT-SVR)	Set the bit ON when the backlight is turned ON.
Set OFF when backlight on (N/A for cMT-SVR)	Set the bit OFF when the backlight is turned ON.
Set ON when backlight off (N/A for cMT-SVR)	Set the bit ON when the backlight is turned OFF.
Set OFF when backlight off (N/A for cMT-SVR)	Set the bit OFF when the backlight is turned OFF.

Macro	Set Bit object can trigger the start of a Macro routine when the Macro has been created in advance. For more information, see “18 Macro References”.
Trigger mode	If [Set style] is set to [Toggle], there is a further selection to make of whether the macro operates after Off to ON, ON to OFF transition, or at both of the changes of state.

Note

- In [Label] tab, if select [ON=OFF (use state 0)] check box, both state 0 and 1 follow the settings of state 0.

- Using address types other than PLB or PLW_Bit for Set Bit objects with [Periodic Toggle] attribute is not supported by cMT-SVR.

13.4. Set Word

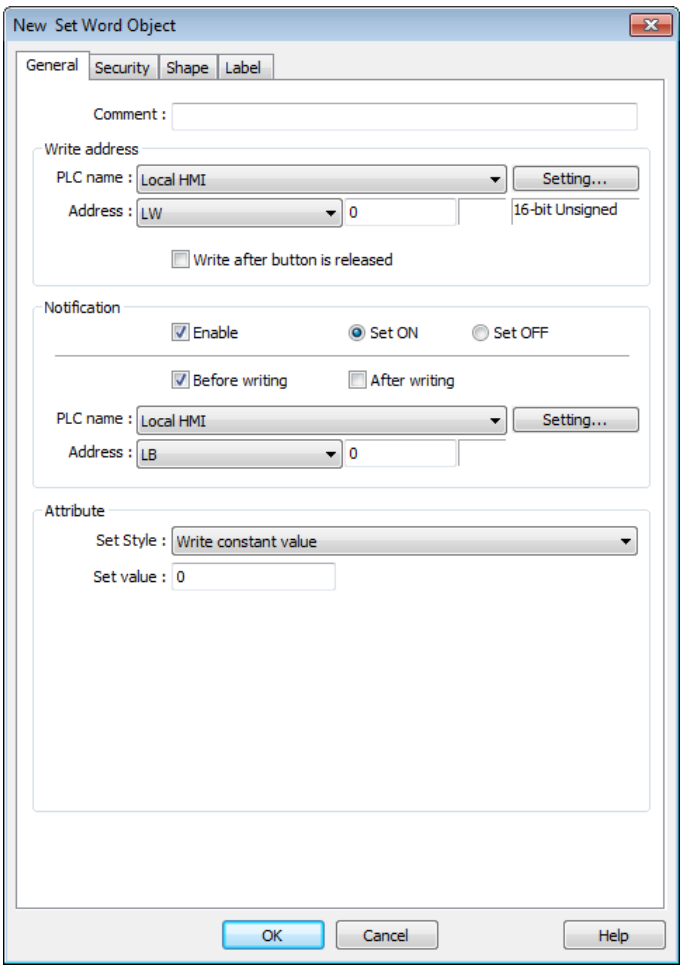
13.4.1. Overview

The Set Word object provides two operation modes: manual or automatic. Manual mode can change the value in a designated word address when the object is touched. In automatic mode, the word register is automatically activated when a pre-defined condition occurs; touching the button will not be effective.

13.4.2. Configuration

Click [Object] » [Set Word] icon on the toolbar to open a Set Word object property dialog box. Set up the properties, press OK button, and a new Set Word object will be created.

General Tab



Setting	Description
Write address	Click [Setting] to select the [PLC name], [Address], [Device type], [System tag], [Index register] of the word device that controls the Set Word object. Users can also set address in [General] tab while adding a new object. **Write after button is released** If this function is selected, the action is delayed till button is released; otherwise, the action is executed once the button is pressed.
Notification	If this check box is selected, it will notify a designated bit address (setting ON or OFF). **Before writing / After writing** Set the state of the designated bit address before or after the manual operation.
Attribute	**Set Style** Select the button action from the drop down list, see Example 2. **Dynamic limits** Set the [Bottom limit] and [Upper limit] by a designated register, see Example 1.

Example 1

Set the [Bottom limit] and [Upper limit] by a designated register. When Dynamic Address is LW-n, where n is an arbitrary number, the rule of setting Upper / Bottom limit is:

Content	16-bit	32-bit	64-bit (cMT / cMT X only)
Dynamic address	LW-n	LW-n	LW-n
Bottom limit	LW-n	LW-n	LW-n
Upper limit	LW-n+1	LW-n+2	LW-n+4

When Dynamic Address is LW-100, the rule of setting Upper / Bottom limit is:

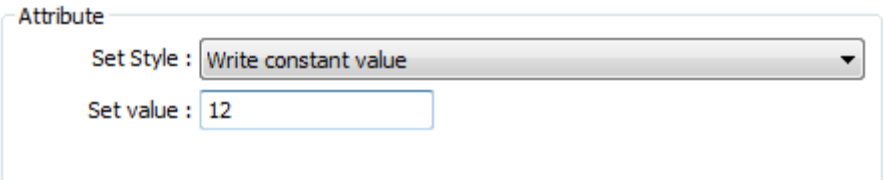
Content	16-bit	32-bit	64-bit (cMT / cMT X only)
Dynamic address	LW-100	LW-100	LW-100
Bottom limit	LW-100	LW-100	LW-100
Upper limit	LW-101	LW-102	LW-104

Example 2

The available button actions are:

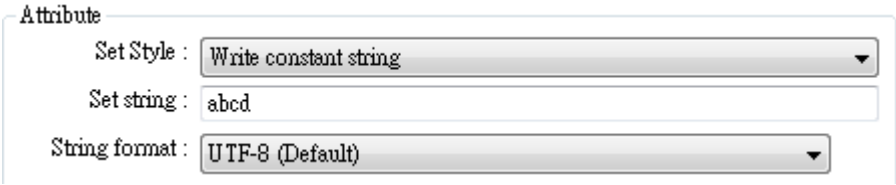
- Write constant value

Write a preset value. Each time the button is pressed, the value in [Set value] is written to the designated register. Data format is as set by the [Write address] above; it can be 16-bit BCD, 32-bit BCD, …64-bit Double. In the following figure, when the button is pressed, the number 12 is written to the designated register.



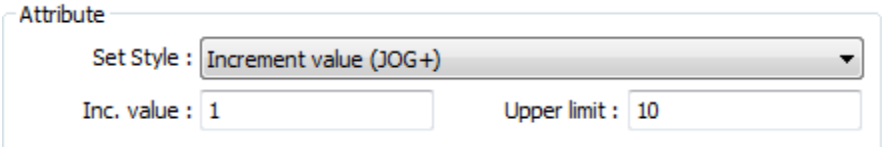
- Write constant string (cMT / cMT X only)

Write a preset string. Each time the button is pressed, the string in [Set string] is written to the designated register. The string format can be: UTF-8, Unicode, Latin-1…etc. In the following figure, when the button is pressed, the string “abcd” is written to the designated register.



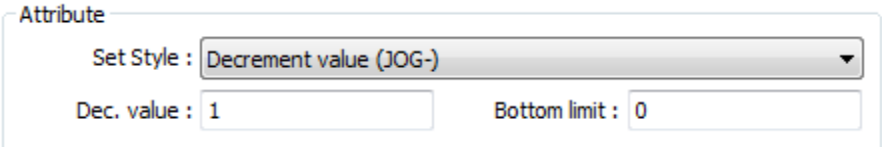
- Increment value (JOG+)

Increase value in register by a set amount in [Inc. value], each time when the button is pressed, up to the [Upper limit]. As shown below, each button press increases the value in the register by 1 until the value is 10.



- Decrement Value (JOG-)

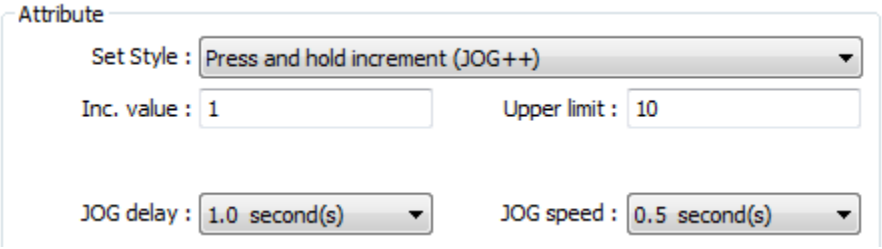
Decrease value in register by a set amount in [Dec. value], each time when the button is pressed, down to the [Bottom limit]. As shown below, each button press decreases the value in the register by 1 until the value is 0.



- Press and hold increment (JOG++)

When the button is held longer than a set time in [JOG delay], it will increase the value in a register by a set amount :[Inc. value] at a set rate :[JOG speed], to the [Upper limit].

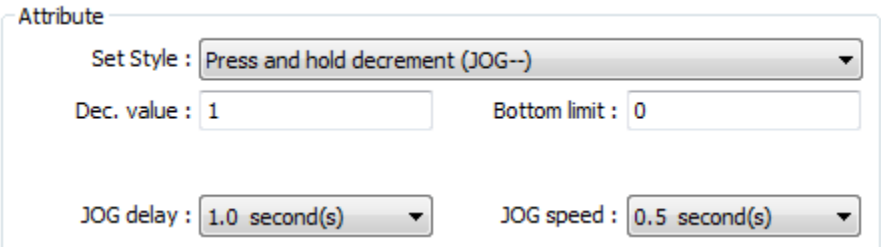
As shown below, when the button is pressed, it increases the value in the designated register by 1. When the button is held longer than 1 second, it increases the value in register by 1 every 0.5 second, till the value is 10.



- Press and hold increment (JOG--)

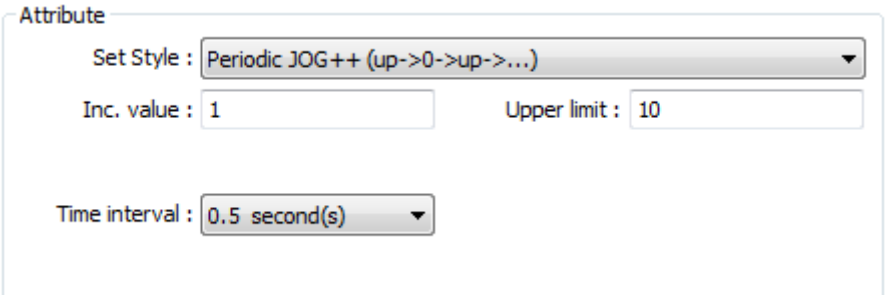
When the button is held longer than a set time in [JOG delay], it will decrease the value in a register by a set amount: [Dec. value] at a set rate: [JOG speed], to the [Bottom limit].

As shown below, when the button is pressed, it decreases the value in the designated register by 1. When the button is held longer than 1 second, it decreases the value in register by 1 every 0.5 second, till the value is 0.



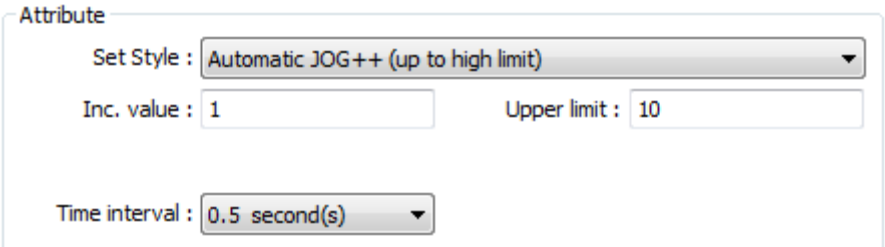
- Periodic JOG++

This automatic function increases the value in the register by a set amount: [Inc. value], at a set rate: [Time interval], to the [Upper limit]. As shown below, the system will automatically increase the value in the register by 1 every 0.5 second, till the value is 10. Then the value returns to 0 and add 1 every 0.5 second again.



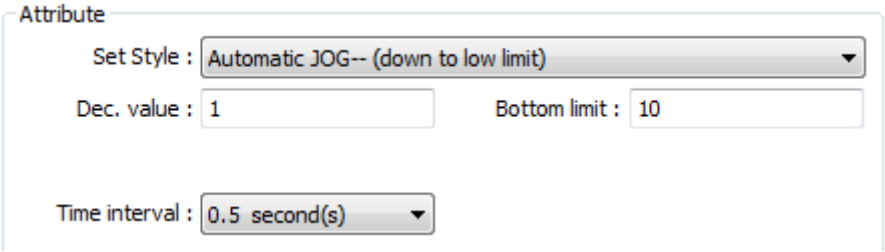
- Automatic JOG++

This automatic function increases the value in the register by a set amount: [Inc. value], at a set rate: [Time interval], to the [Upper limit].then holds this value. As shown below, the system will automatically increase the value in the register by 1 every 0.5 second, till the value is 10, and then stop.



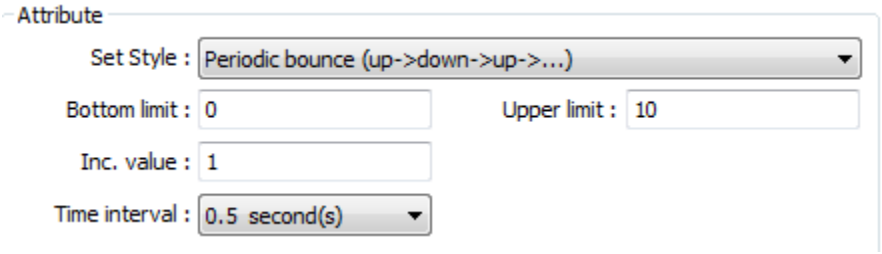
- Automatic JOG--

This automatic function decreases the value in the register by a set amount: [Dec. value], at a set rate: [Time interval], to the [Bottom limit].then holds this value. As shown below, the system will automatically increase the value in the register by 1 every 0.5 second, till the value is 10, and then stop.



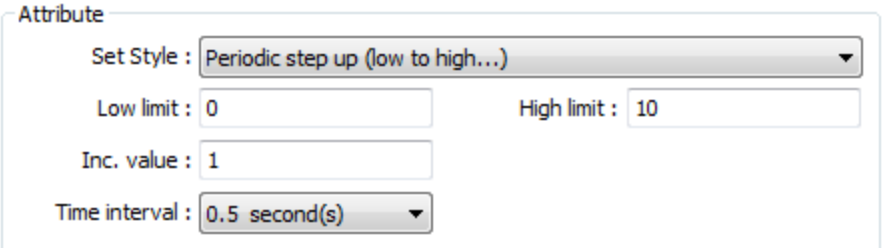
- Periodic bounce

Increases the word address value to the [Upper limit] by a [Inc. value] at a set rate in [Time interval], then decreases to the [Bottom limit] by the same value at the same rate. As shown below, the system will increase the value in the designated register by 1 every 0.5 second, till the value is 10, and then decrease the value by 1 every 0.5 second till the value is 0 whenever the screen is active.



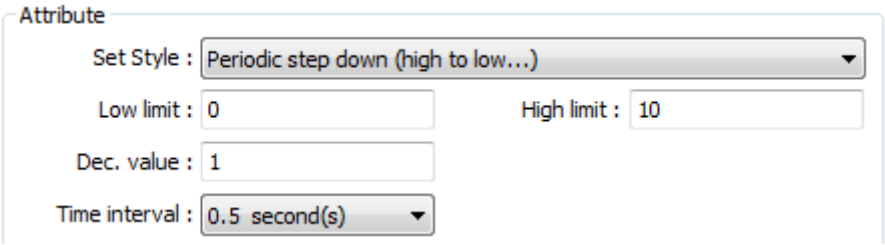
- Periodic step up

Step up to the [High limit] by [Inc. value] at a set rate in [Time interval], then reset immediately to the [Low limit]. The action repeats whenever the screen is active. As shown below, the system will increase the value in the designated register by 1 every 0.5 second, till the value is 10, and then reset to 0 and increase again, and the action repeats.



- Periodic step down

Step down to the [Low limit] by [Dec. value] at a set rate in [Time interval], then reset immediately to the [High limit]. The action repeats whenever the screen is active. As shown below, the system will decrease the value in the designated register by 1 every 0.5 second, till the value is 0, and then reset to 10 and decrease again, and the action repeats.



- Set when window opens / Set when window closes

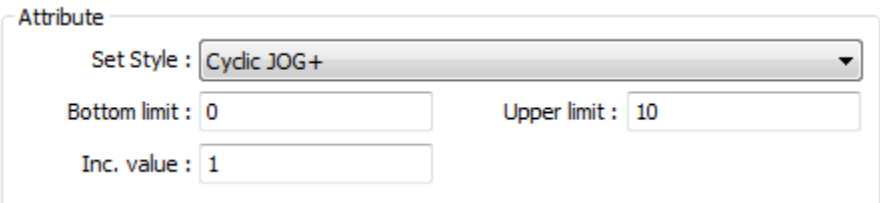
Automatic function occurs whenever the screen is active. The value entered in [Set value] is set into the word address when the action occurs. If [Set value] is set to 5, when the window opens / closes, the system enters 5 into the designated register.

- Set when backlight on / Set when backlight off (Not supported on cMT-SVR)

Automatic function occurs whenever the backlight is active. The value entered in [Set value] is set into the word address when the action occurs. If [Set value] is set to 5, when the backlight turns ON / OFF, the system sets 5 into the designated register.

- Cyclic JOG+

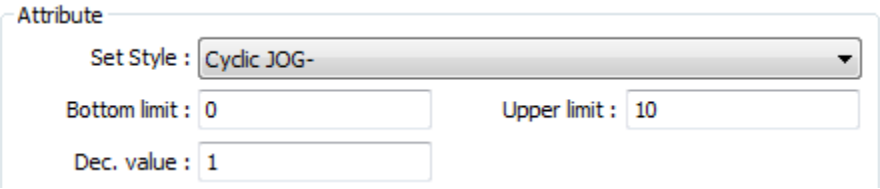
Each time when the button is pressed, increases the word address value to the [Upper limit] by [Inc. value] then reset to the [Bottom limit]. As shown below, each time when pressing the button, the system will increase the value in the designated register by 1, till the value is 10, and then reset to 0 and increase again by pressing the button.



- Cyclic JOG-

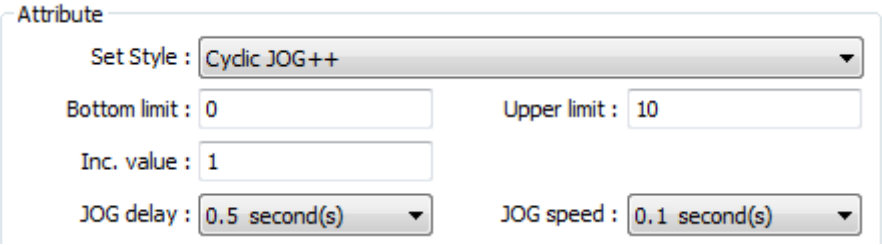
Each time when the button is pressed, decrease the word address value to the [Bottom limit] by [Dec. value] then reset to the [Upper limit]. As shown below, each time when pressing the

button, the system will decrease the value in the designated register by 1, till the value is 0, and then reset to 10 and decrease again by pressing the button.



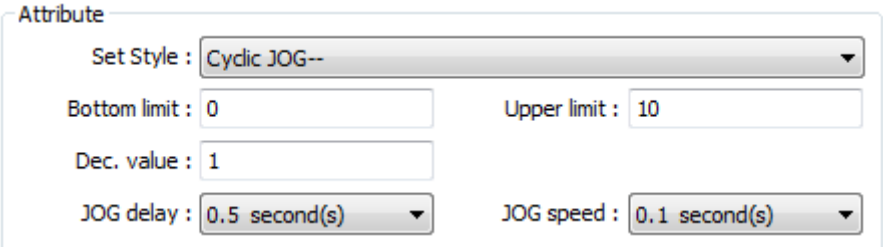
- Cyclic JOG++

When the button is held longer than a set time in [JOG delay], it increases the value in a register by a set amount in [Inc. value] at a set rate in [JOG speed], to the [Upper limit], then reset to the [Bottom limit]. As shown below, when the button is held longer than 0.5 second, increase the value in the designated register by 1 every 0.1 second, till the value is 10, and then reset to 0 and increase again by holding the button.



- Cyclic JOG- -

When the button is held longer than a set time in [JOG delay], decrease the value in a register by a set amount in [Dec. value] at a set rate in [JOG speed], to the [Bottom limit], then reset to the [Upper limit]. As shown below, when the button is held longer than 0.5 second, decrease the value in the designated register by 1 every 0.1 second, till the value is 0, and then reset to 10 and decrease again by holding the button.

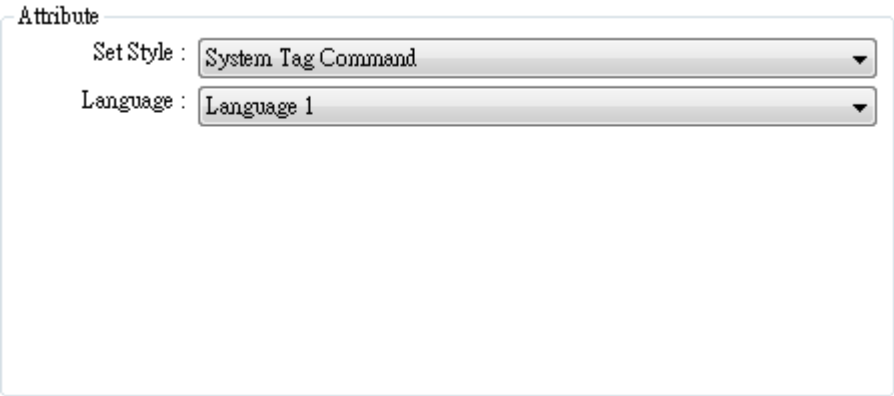


- Object Control Command (cMT / cMT X only)

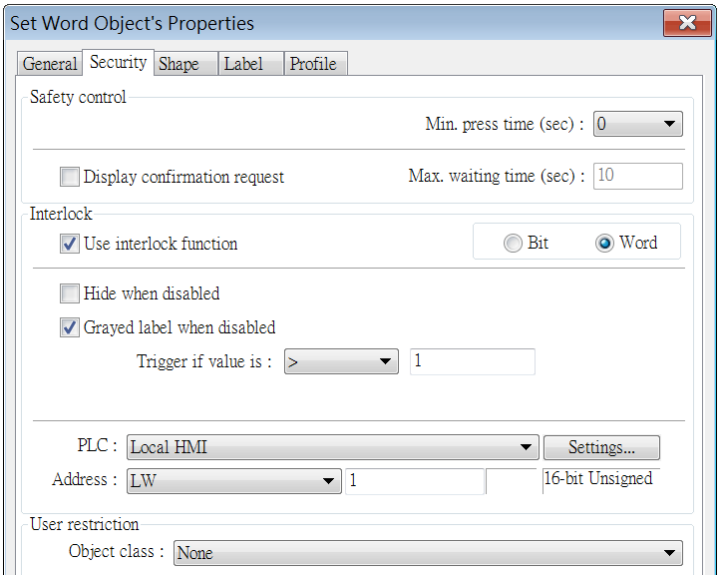
Listed for selection are commands (with their corresponding command values) available for the control address of various functions. For a given function, with write address set to its control address, Set Word writes the command value and achieves function control.

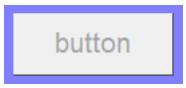
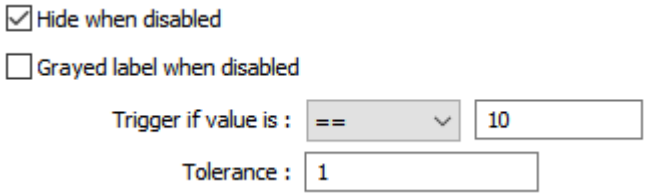
- System Tag Command (cMT / cMT X only)

When a system tag is set as write address, for example, LW-9134: Language Mode, the commands relating to the system tag can be selected.



Security Tab



Setting	Description
Interlock	**Use interlock function** When this option is enabled and [Word] is selected, whether the object is operable depends on the condition of a word address specified in [Trigger if value is:]. In the settings above, the object is operable only when the value in LW-1 is greater than 1. **Hide when disabled** The object is hidden when the specified condition does not occur in the specified word address. **Grayed label when disabled** The label of the object turns gray when the specified condition does not occur in the specified word address.  **Trigger if value is:** This setting is for specifying a trigger condition. The available options are: >, <, ==, <>, >=, and <=. A tolerance value can be set for conditions == and <>. For example:  When the value is the specified word address is greater than or equal to 11, or smaller than or equal to 9, the object will be hidden and is not operable.

Note

- Using address types other than PLW for Set Word objects with automatic attributes such as [Periodic set up], [Periodic set down], [Automatic JOG++], [Automatic JOG--], and [Periodic JOG++]…etc, is not supported by cMT-SVR.

13.5. Function Key

13.5.1. Overview

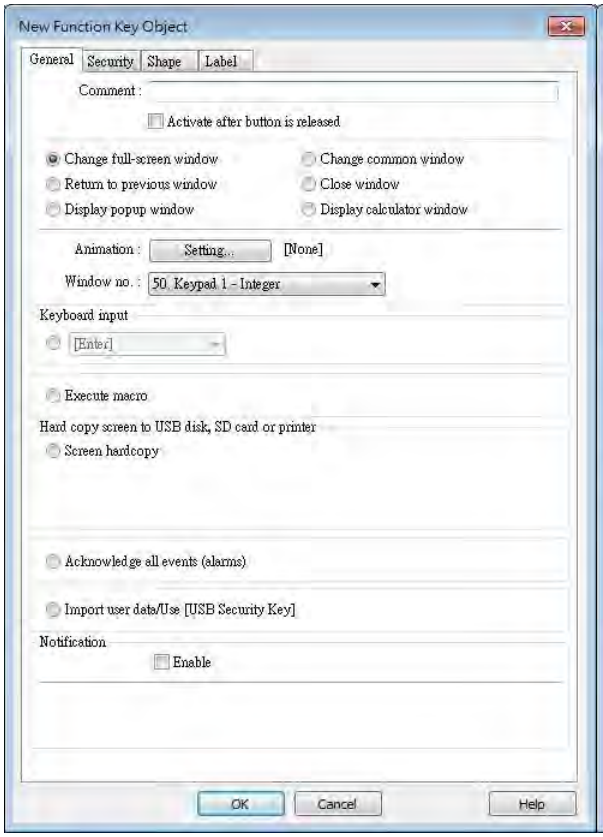
The Function Key object can be used for several tasks, such as switching between windows, keypad design, Macro execution, screen hardcopy, and setting USB security key.
Function Keys with [Screen hardcopy] or [Import user data/Use [USB Security Key]] selected do not work remotely on cMT Viewer.

13.5.2. Configuration

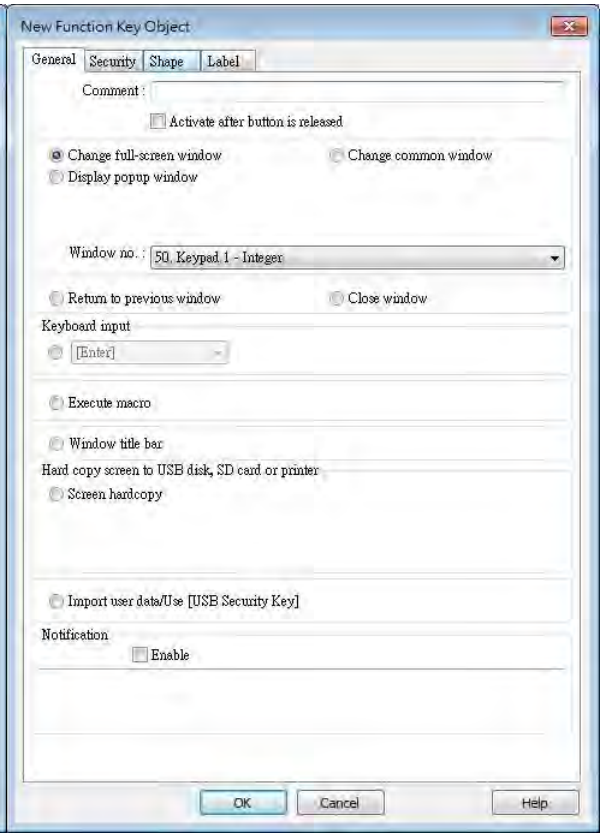
Click [Object] » [Function Key] icon on the toolbar to open a Function Key object property dialog box. Set up the properties, press OK button, and a new Function Key object will be created.

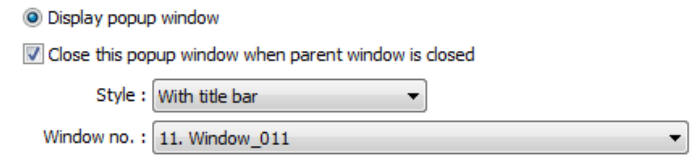
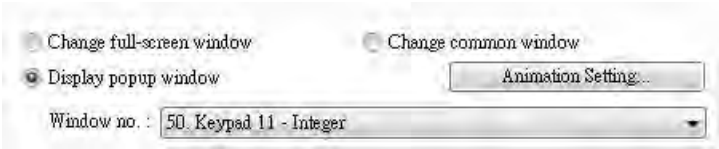
General Tab

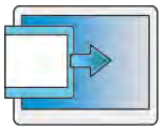
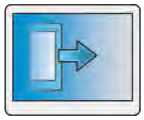
cMT / cMT X Series

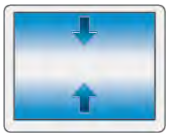
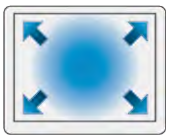
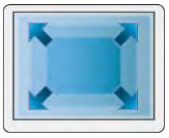


eMT, iE, XE, mTV Series



Setting	Description
Activate after button is released	If this function is selected, the action is delayed till button is released; otherwise, the action is executed once the button is pressed.
Change window	**Change full-screen window:** Change to another base window. **Change common window:** Change common window. **Display popup window:** A pop-up window displays in the base window. If [Close this popup window when parent window is closed] check box is selected, the pop up window will be closed when change the base window to another window. Otherwise, a function key in the pop up window is needed to close it.  **(cMT / cMT X Series) Animation Setting:** cMT / cMT X Series allows using transition effects for opening popup windows using Function Keys. The settings can be opened by clicking [Animation Setting].  The effects are shown below. Different effects may be used for Start (window appears) and End (window disappears).

Effect Name	Transition
Fade	
Fly	
Float	

Wipe	
Split	
Circle	
Clock	
Zoom	
Turn	
Push	

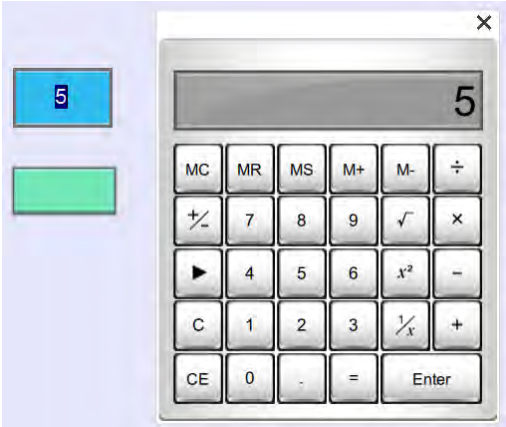
[Duration] specifies how many milliseconds (ms) a transition effect takes to complete.

[Direction] The direction of the transition.

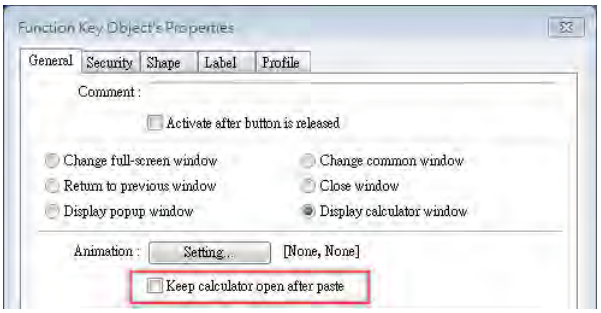
Return to previous window: If this is selected, the Function Key will change from the current screen to the previous one displayed. For example, when window no. 10 is changed to window no. 20, press the function key to return to window no. 10. This function is only available for base window.

Close window: Close any active pop-up windows, message windows included.

Display calculator window(cMT / cMT X Series): If this option is selected, clicking this Function Key can open a calculator with an unchangeable appearance. This calculator allows operators to do simple calculation and enter the result to a destination object by pressing the Enter key.



Selecting this option opens [Keep calculation open after paste] checkbox, which keeps the calculator displayed after the value is entered to the object.



Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

Keyboard Input

Configures the button as a keypad key, and the character it enters, via [Numeric] or [ASCII] objects.

Enter: Same as the keyboard's "Enter" function.

Backspace: Same as the keyboard's "Backspace" function.

Clear: Clear the value in the word register.

Esc: Same as the [Close window] function; it is used to close the keyboard window.

Delete: Same as the keyboard's "Delete" function, deletes the number or character on the right side of the text cursor.

Left: Same as the keyboard's "←" key moves the text cursor to the left side of the previous number or character.

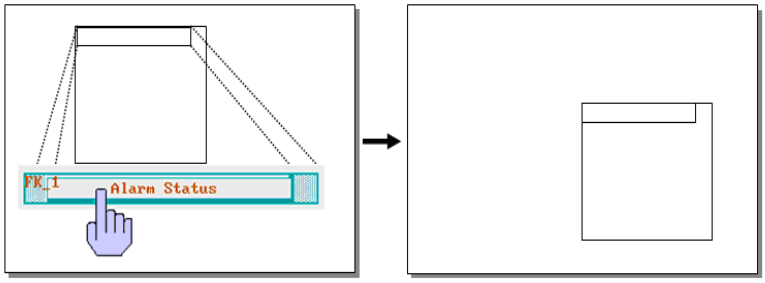
Right: Same as the keyboard's "→" key moves the text cursor to the left side of the next number or character.

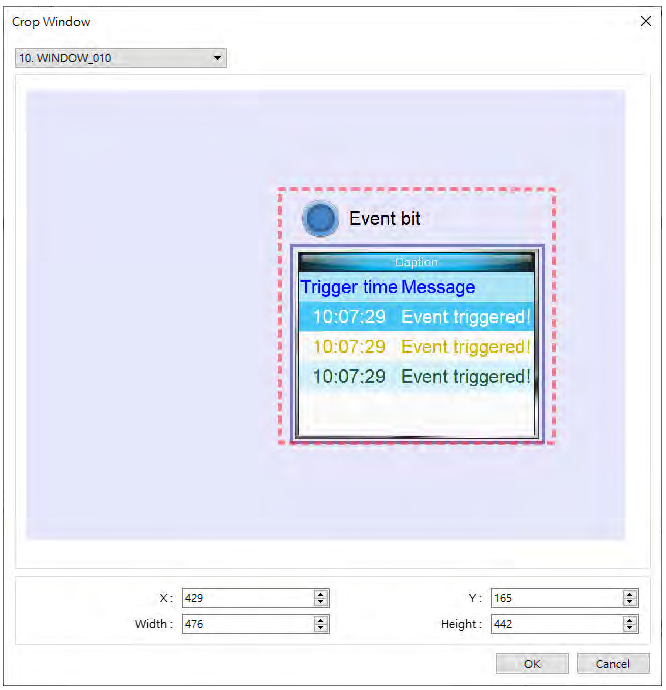
Line feed: Move the cursor down to the next line.

Inc: Increment by 1.

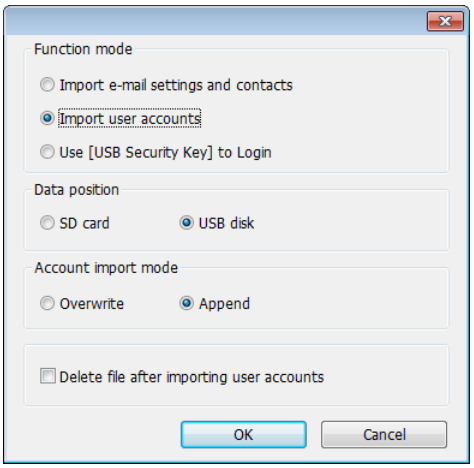
Dec: Decrement by 1.

ASCII/UNICODE: Specify the character to be entered by this key.

Execute Macro	Select this check box to execute one of the Macros from the drop down list that has already been configured by users. For more information, see “18 Macro References”. Execute macro Macro : [ID:000] macro_0
Window title bar	Function Key defined can be used to move a pop-up window which has no [window title bar] to a preferred position on screen. Select the pop-up window and then click on a preferred position, the window will be moved.  Select the window title bar firstly. Touching the screen for the new position the popup window will be moved. Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.
Hard copy screen to USB disk, SD card or printer	Print the current window. Before using this function, choose a printer model in [System Parameter Settings] » [Model] » [Printer]. If a monochrome printer is used, selecting [grayscale] can provide a better print result, but the text may not be clearly printed. To improve text printing, avoid using [grayscale]. Hard copy screen to USB disk, SD card or printer Screen hard copy Printer : HP PCL Series (USB) Rotate image 90 degrees Mode : color **Crop window** This option can be selected for taking a cropped screenshot.



Acknowledge all events (alarms) (cMT / cMT X Series)	Acknowledge all events once by pressing the Function Key.
Import user data / Use [USB Security Key]	A Function Key can be used to import the e-mail contacts or user accounts set, also, to log in using USB Security Key.



Data Position

Select the external device to store data from [SD card] or [USB disk].

Account import mode

If [Overwrite] is selected, the existing accounts will be overwritten with new accounts in the external device after importing. If [Append] is selected, HMI will append more accounts while the old accounts

	still exist. **Delete file after importing user accounts** If select this check box, the system will delete the account data saved in the external device after importing, this can prevent the account data from leaking out.
Notification	If this selection is enabled, it will notify a designated bit address to set ON or OFF, each time the button is pressed.

Note

- [Overwrite] is the only option when importing the e-mail contacts. This means that all existing contacts will be removed first, and then the new contacts are added.

For more information, see “6 Window Operations”, “12 Keypad Design and Usage”, “36 Administrator Tools”.

13.6. Toggle Switch

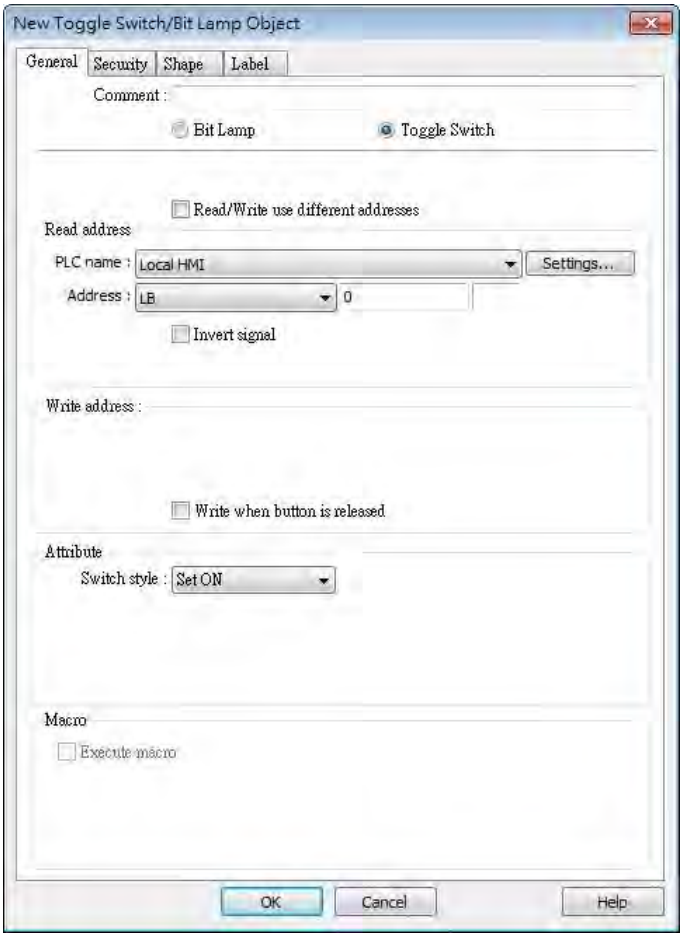
13.6.1. Overview

Toggle Switch object is a combination of Bit Lamp object and Set Bit object. The appearance of the object is controlled by the ON / OFF state of the read bit address. As well, pressing the button sets the value in the bit address according to the settings.

13.6.2. Configuration

Click [Object] » [Toggle Switch] icon on the toolbar to open a Toggle Switch object property dialog. Set up the properties, press OK button, and a new Toggle Switch object will be created.

General Tab



Setting	Description
Comment	User can describe the information of the object. **Bit Lamp / Toggle Switch** Switch between Bit Lamp and Toggle Switch features.
Read/Write use different addresses	Different addresses can be used to read data and write data.
Read address	Click [Setting] to select the [PLC name], [Address], [Device type], [System tag], [Index register] of the bit device that controls the [Toggle Switch] object. Users can also set address in [General] tab while adding a new object. **Invert signal** Reverses the display of ON / OFF states. For example, if [Invert signal] check box is selected, when the designated bit is OFF, the object displays ON state. When [Read/Write use different addresses] option is not selected, the title of this group box will be “Read/Write address”.
Write address	Click [Setting] to select the [PLC name], [Address], [Device type], [System tag], [Index register] of the bit device that controls the [Toggle Switch] object. Users can also set address in [General] tab while adding a new object. The address can be the same or different from [Read address]. **Write after button is released** If this function is selected, the action is delayed till button is released, otherwise, the action is executed once the button is pressed. This function does not work with momentary buttons.
Attribute	**Set style** / **Description** Set ON: Set ON the designated bit of the device. Set OFF: Set OFF the designated bit of the device. Toggle: Alternates the bit state each time pressed. Momentary: Holds the bit ON only while button is pressed.
Macro	Toggle Switch object can trigger the start of a Macro routine when the Macro has been created in advance. ☞ For more information, see “18 Macro References”.

13.7. Multi-State Switch

13.7.1. Overview

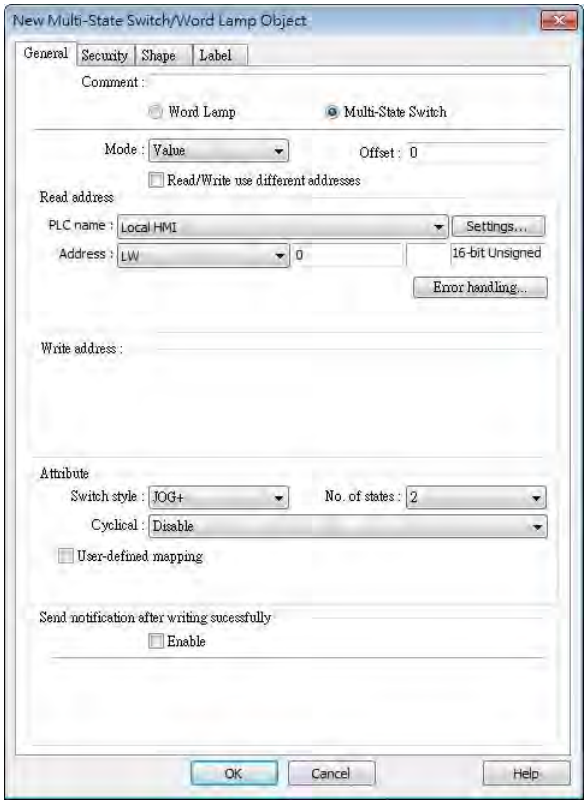
Multi-state Switch object is a combination of Word Lamp object and Set Word object. The appearance of the object is controlled by the value of the read word address. As well, pressing the button sets the value in the word address according to the settings.

13.7.2. Configuration

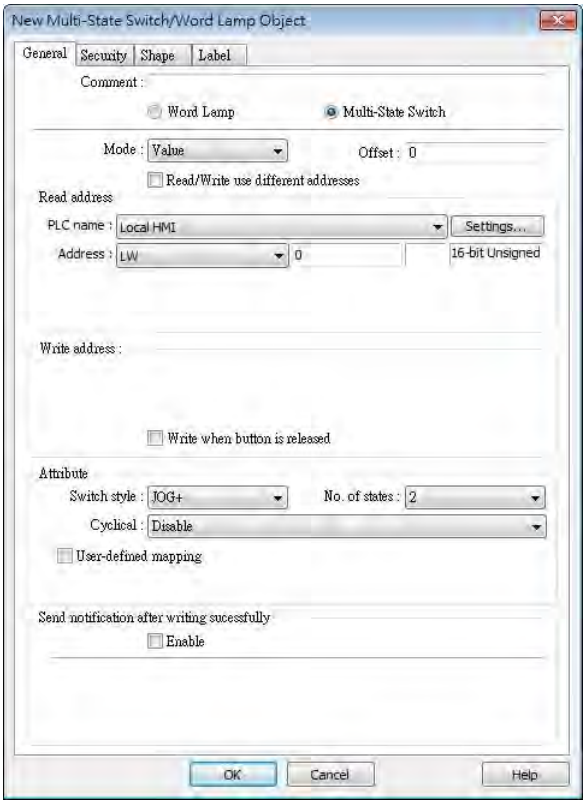
Click [Object] » [Multi-State Switch] icon on the toolbar to open a Multi-State Switch object property dialog box. Set up the properties, press OK button, and a new Multi-State Switch object will be created.

General Tab

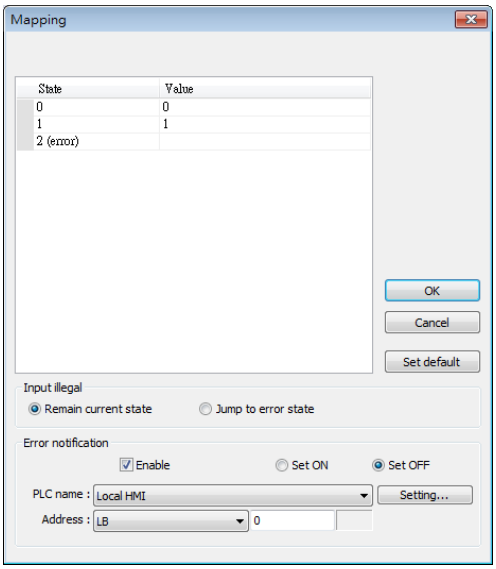
cMT / cMT X Series



eMT, iE, XE, mTV Series



Setting	Description
Comment	User can describe the information of the object. **Word Lamp / Multi-State Switch** Switch between Word Lamp and Multi-State Switch features.
Model / Offset	Different modes can be selected: [Value], [LSB]. For more information, see “13.2 Word Lamp”.
Read/Write use different addresses	Different addresses can be used to read data and write data.
Read address	Click [Setting] to select the [PLC name], [Address], [Device type], [System tag], [Index register] of the word device that controls the Multi-state Switch object. Users can also set address in [General] tab while adding a new object. When [Read/Write use different addresses] option is not selected, the title of this group box will be “Read/Write address”.
Write address	Click [Setting] to select the [PLC name], [Address], [Device type], [System tag], [Index register] of the word device that controls the Multi-state Switch object. Users can also set address in [General] tab while adding a new object. **Write after button is released** If this function is selected, the action is delayed till button is released; otherwise, the action is executed once the button is pressed.
Attribute	**Switch style** Select the object’s operation mode, see Example 1. **User-defined mapping** The value placed in the write register of each selection, the action taken when an illegal value is entered, and error notification to a designated bit address can be set.



Remain current state

If an illegal value is entered, Multi-state Switch will remain at the current state.

Jump to error state

If an illegal value is entered, Multi-state Switch will jump to the error state.

Error notification

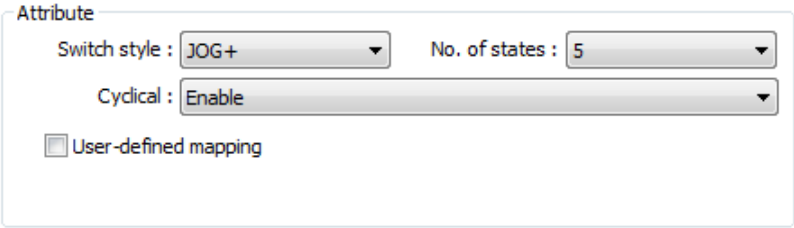
If an illegal value is entered, automatically set the value placed in the designated register.

Send notification after writing successfully	After the system successfully writes data to PLC, the designated bit address will be set On/Off.
Error handling (cMT / cMT X Series)	The action taken when an illegal value is entered or notify a designated bit address. This is similar to [User-defined mapping]; the difference is the value corresponding to each state need not to be preset.

Example 1

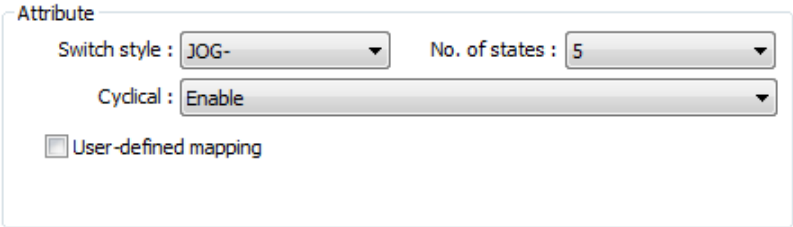
- JOG+

Increase the value of a designated register by 1 each time when pressing the button, till the value equals to [No. of states]. A cyclic action can be enabled. As shown below, each time when pressing the button, the state number will add 1 start from state 0, till state 4 ([no. of state]-1), and returns to 0 and step up again.



- JOG-

Decrease the value of the designated register by 1 each time when pressing the button, till the value equals to 0. A cyclic action can be enabled. As shown below, each time when pressing the button, the state number will minus 1 start from state 4 ([no. of state]-1), till state 0, and returns to state 4 and step down again.



13.8. Slider

13.8.1. Overview

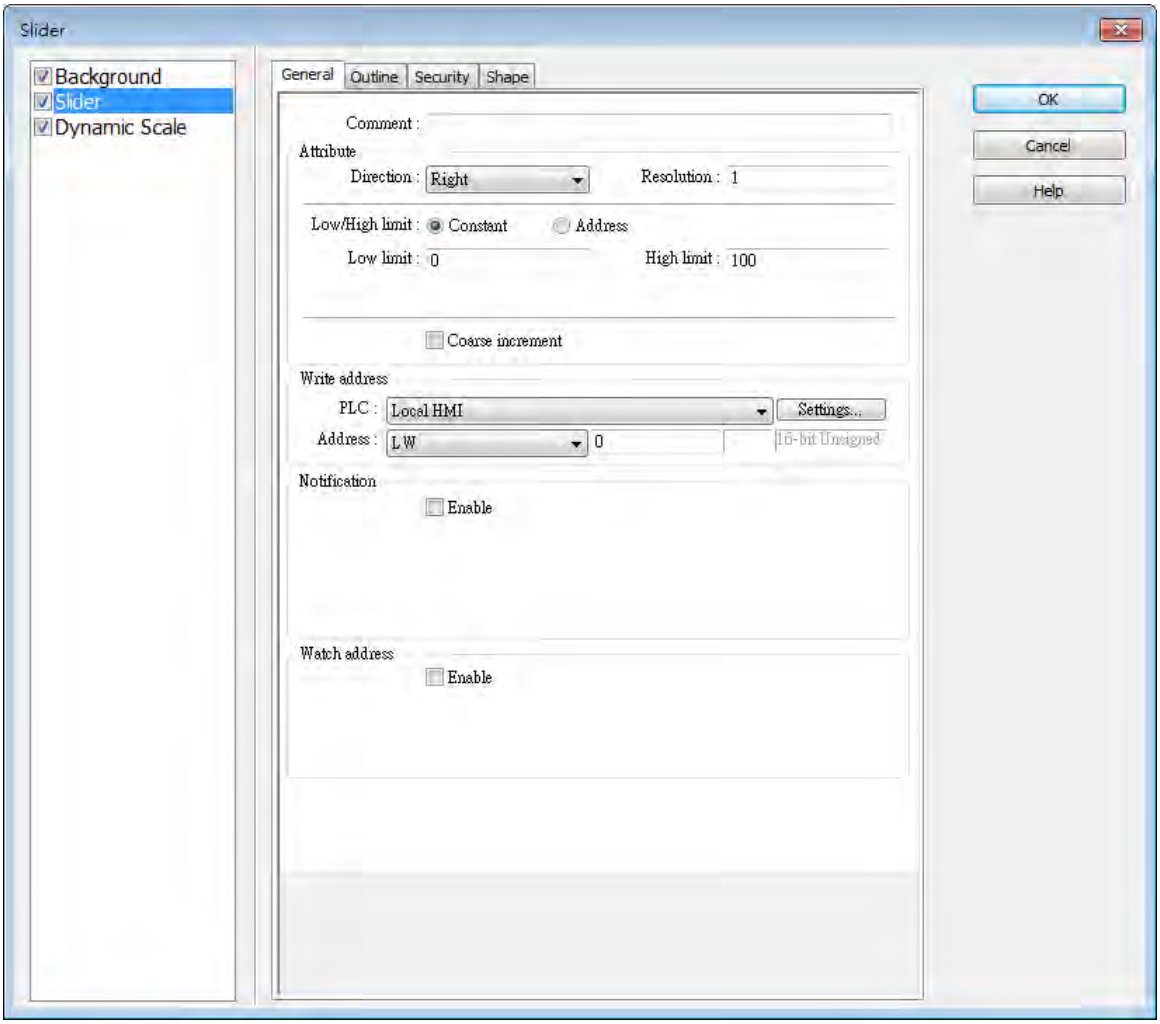
Slider object is used to change the value in a designated word register address by moving the slide on the screen.

13.8.2. Configuration

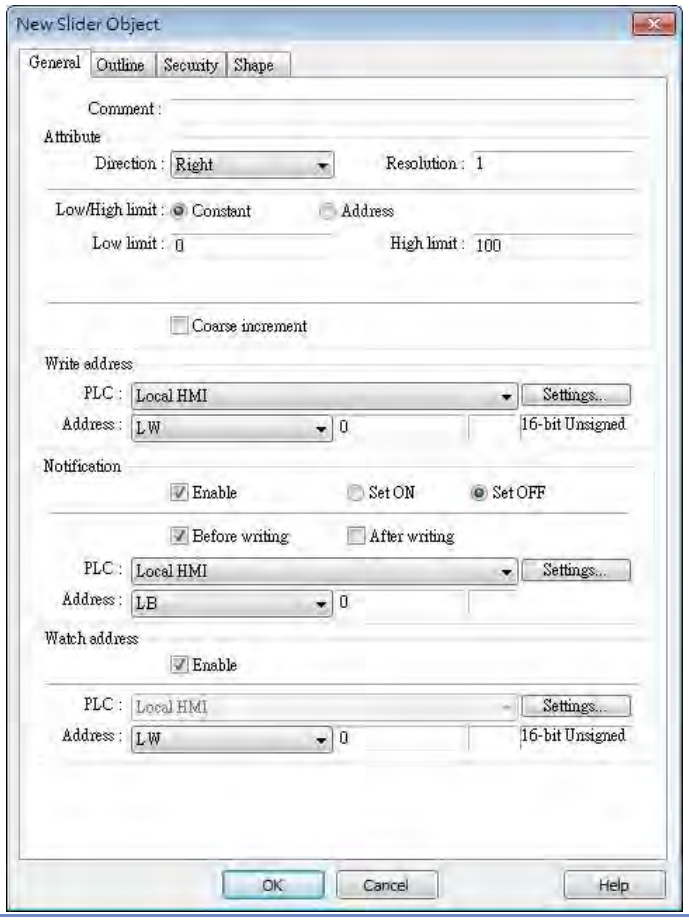
Click [Object] » [Slider] icon on the toolbar to open a Slider object property dialog box. Set up the properties, press OK button, and a new Slider object will be created.

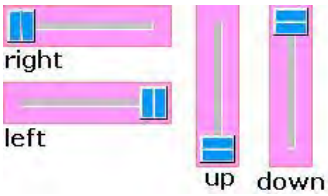
General Tab

cMT / cMT X



eMT, iE, XE, mTV



Setting	Description
Attribute	**Direction** Select the direction of the slider. (Right, Up, Left, Down)  **Resolution** Sets the value change in the word register for each step of the Slider. For example, if set to 10, the register value changes by 10 points for each increment or decrement on the Slider. **Constant** Sets the range of the Slider. For example, If set [Low limit] to 5, and [High limit] to 100, the Slider will enter values between 5 and 100. **Address** Set the [Low/High limit] by a designated register, see Example 1. **Coarse increment** Apart from moving the roller to change the value as in [Resolution],

	if this option is selected, the word value will increase / decrease by the [Increment] value each time the object is touched.
Write address	Click [Setting] to select the [Device], [Address], [Device type], [System tag], [Index register] of the word device that controls the Slider object. Users can also set address in [General] tab while adding a new object.
Notification	If enabled, the state of a designated bit address will be set to ON or OFF before or after writing. Click **[Setting]** to select the [Device], [Address], [Device type], [System tag], [Index register] of the bit device that controls the notification settings. Users can also set address in [General] tab while adding a new object. **[Before writing] / [After writing]** Change the state of a designated bit register before, or after the slider is slid.
Watch address	When moving the roller, the new value written to the word register address can be displayed in real time.

Example 1

Set the low or high limit by a designated register. When write address is LW-n, where n is an arbitrary number, the rule of setting limits is:

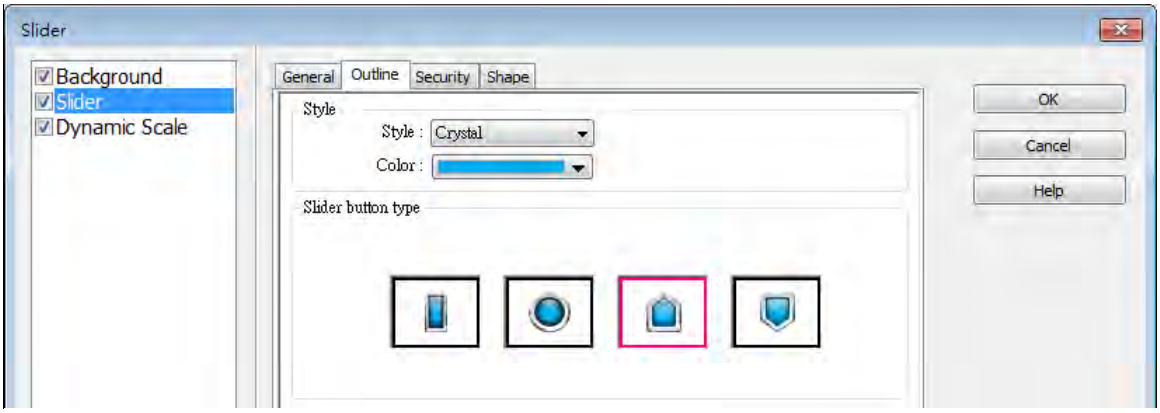
Content	16-bit	32-bit	64-bit (cMT / cMT X only)
Address	LW-n	LW-n	LW-n
Low limit	LW-n	LW-n	LW-n
High limit	LW-n+1	LW-n+2	LW-n+4

When address is LW-100, the rule of setting limits is:

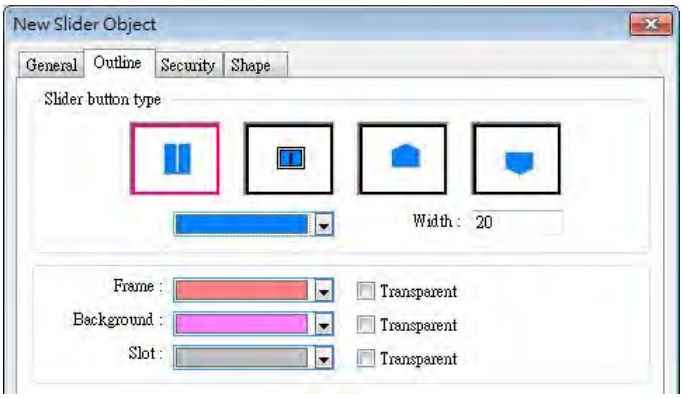
Content	16-bit	32-bit	64-bit (cMT / cMT X only)
Address	LW-100	LW-100	LW-100
Low limit	LW-100	LW-100	LW-100
High limit	LW-101	LW-102	LW-104

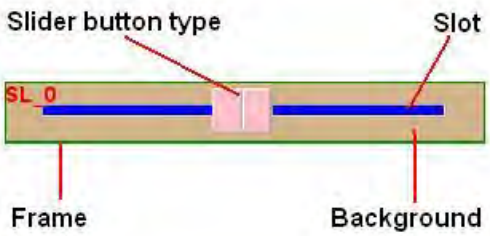
Outline Tab

cMT / cMT X Seires



eMT, iE, XE, mTV Seires

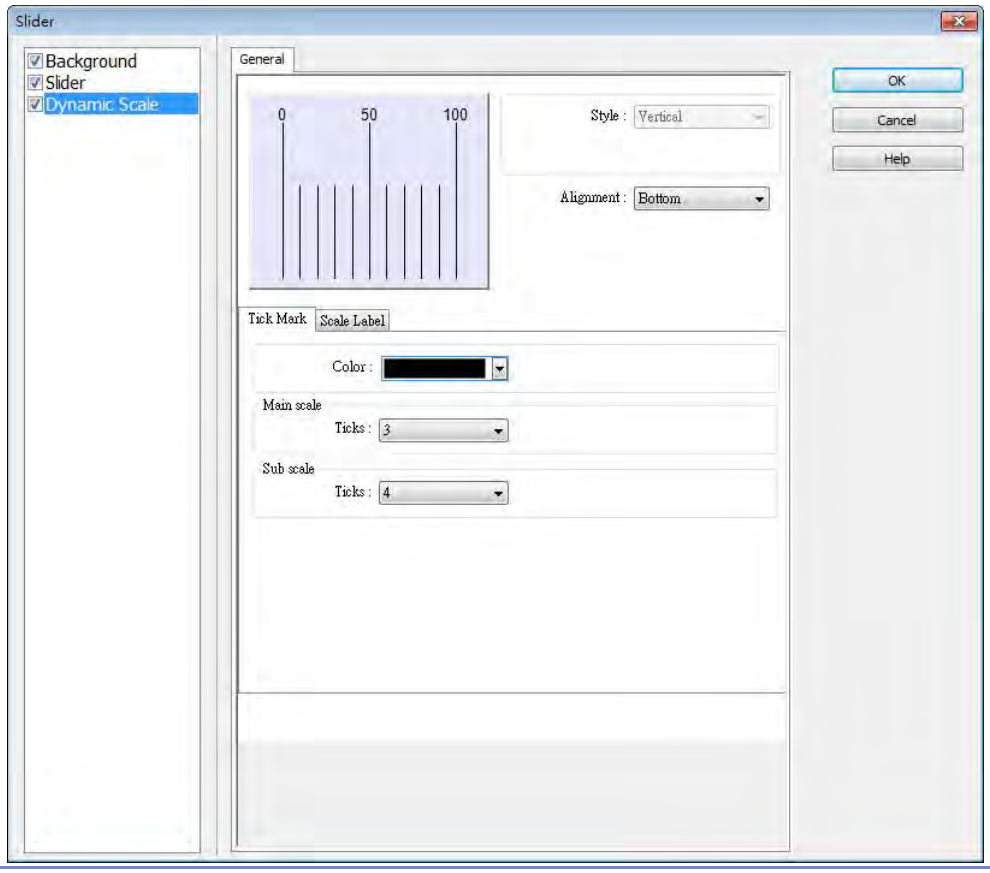


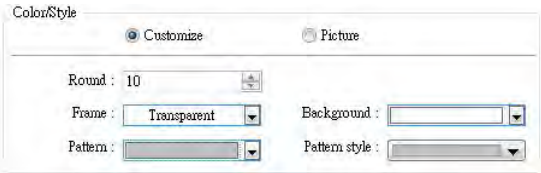
Setting	Description
Slider button type (cMT / cMT X)	Select slider button type, by default, a picture selected from the Picture Library can be the slider button.
Slider button type (eMT, iE, XE, mTV)	Four default styles are offered, and the width and color of the Frame, Background, Slot can be set. 

13.8.3. Combo Setting

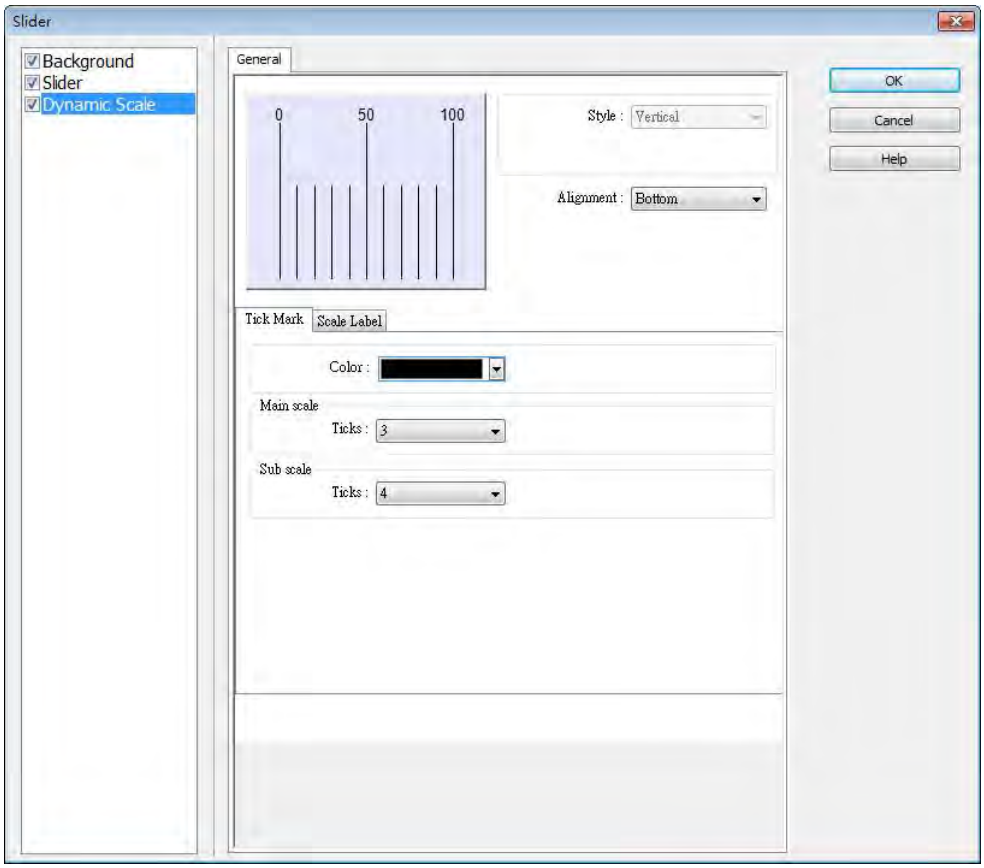
cMT / cMT X Series HMI support combo setting for Slider, which allows setting of multiple related objects at a time. Slider can be set with Background and Dynamic Scale.

Background

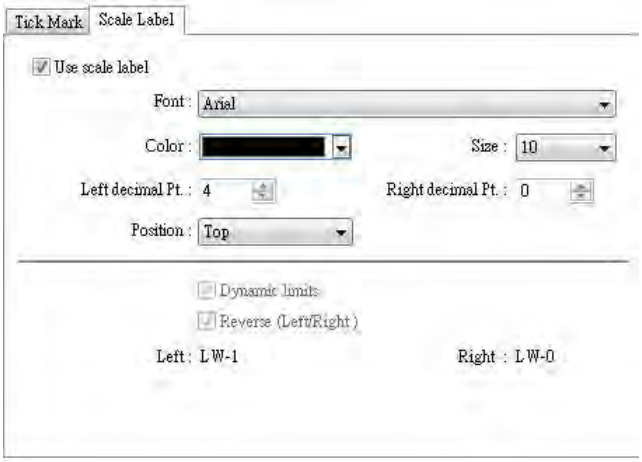


Setting	Description
Margin	Specify the space between the background edge and the objects.
Color/Style	**Customize**  Select a suitable background pattern and color. **Picture** Use the default picture or choose a picture from Picture Library.

Dynamic Scale



Setting	Description
Style	The scale style will follow the Slider.
Alignment	The position of the scale bar along the Slider.
Tick Mark	Configure the number of tick marks for main and sub scales, and the color of tick marks.
Scale Label	Configure the font, font color, font size and other attributes of scale label. In Slider settings if [Address] is selected for Low/High limit, then Scale Label's [Dynamic limits] will be set automatically. In Slider settings if [Left] is selected for Direction, then Scale Label's [Reverse (Left/Right)] will be set automatically.


Tick Mark
Scale Label
Use scale label
Font : Arial
Color :
Size : 10
Left decimal Pt. : 4
Right decimal Pt. : 0
Position : Top
Dynamic limits
Reverse (Left/Right)
Left : LW-1
Right : LW-0

13.9. Numeric

13.9.1. Overview

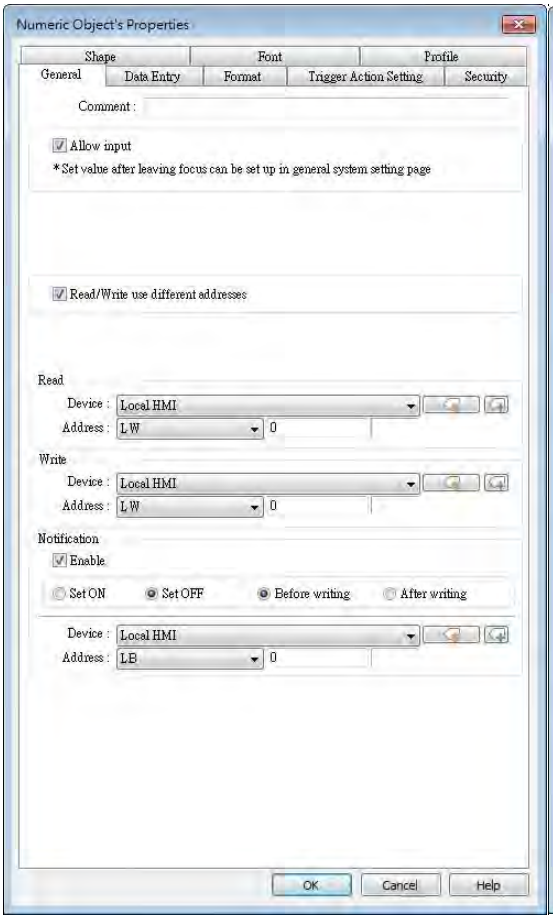
Numeric object can be used to input or display the value of a designated word register.

13.9.2. Configuration

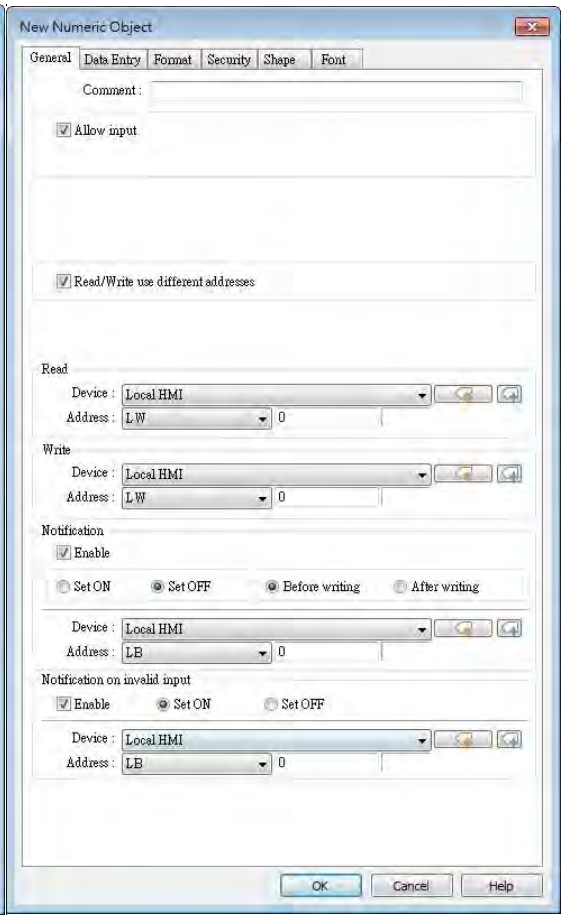
Click [Object] » [Numeric] icon on the toolbar to open a Numeric object property dialog box. Set up the properties, press OK button, and a new Numeric object will be created.

General Tab

cMT, cMT X Series



eMT, iE, XE, mTV Series

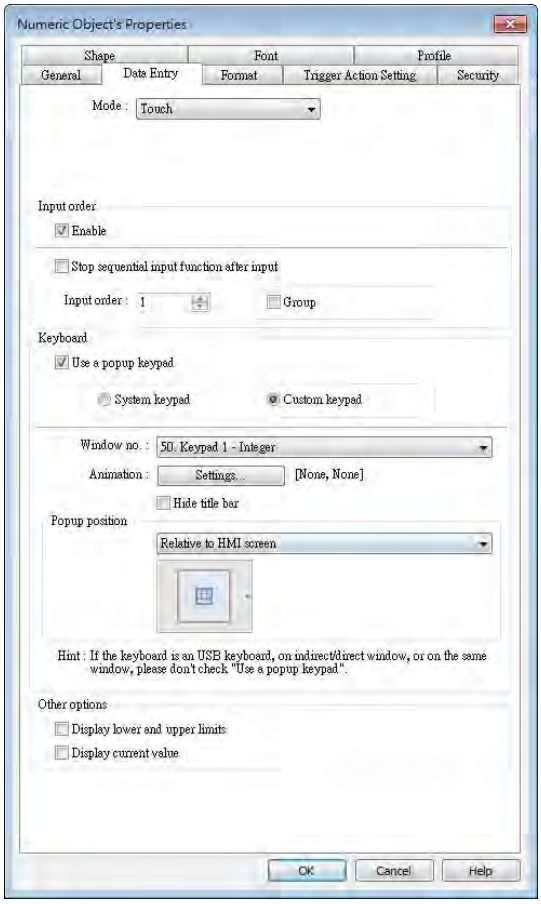


Setting	Description
Allow input	If selected, the input features and relevant settings are enabled.
Read / Write	Different addresses can be used to read data and write data.

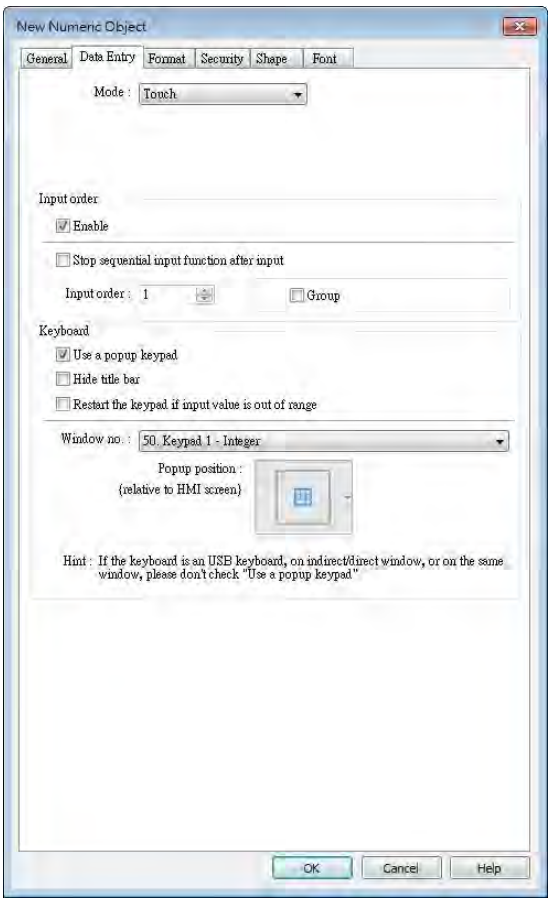
use different address	
Read address	Click [Setting] to select the [Device], [Address], [Device type], [System tag], [Index register] of the word device that displays the value. Users can also select a tag defined in Address Tag Library. When [Read/Write use different addresses] option is not selected, the title of this group box will be “Read/Write address”.
Write address	Select the [Device], [Device type], [Address] of the word device that system writes to.
Notification	With notification enabled, the state of the designated bit address can be set on / off, before / after writing. Notification works differently comparing cMT / cMT X Series with eMT/iE/XE/mTV Series. **cMT/cMT X Series:** Trigger the object to open a keyboard window » Notification before writing » Enter a value and press ENTER » Start writing operation » Notification after writing. **eMT/iE/XE/mTV Series:** Trigger the object to open a keyboard window » Enter a value and press ENTER » Notification before writing » Start writing operation » Notification after writing.
Notification on invalid input	If an illegal value is entered, automatically set the state of a designated register.

Data Entry Tab

cMT, cMT X

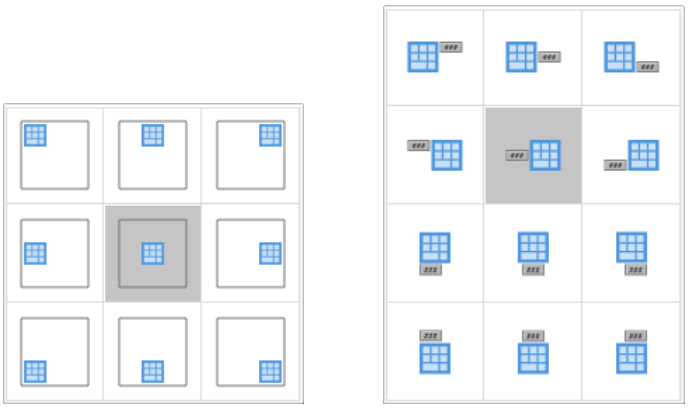


eMT, iE, XE, mTV Series



Setting	Description
Mode	**Touch** Used when data entry is initiated by touching the screen object. **Bit control** Used when data entry is enabled by turning ON a designated bit, and entry ends when the bit goes OFF.
Input control	Specify a bit address that enables or ends data entry. The order of data entry is specified in [Input order] and an external USB keyboard is needed for data entry. For cMT-SVR, use cMT Viewer's keyboard.
Input order	Perform continuous input by setting [Input order] and [Group]. The criterion of searching the next input object: ● The range of [Input order]: 1 ~ 511, range of [Group]: 1 ~ 15. ● If [Group] is not selected, its input order is 0. ● The system only searches for the objects within the same

	Group. ● The lower number of order is entered before the higher number of order. ● For multiple objects within the same group and with the same input order, the object placed in the lower layer is entered first. Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.
Keyboard	**Use a popup keypad** If selected: A pre-designed pop-up keypad can be chosen by selecting a check box and selecting the relative position on the HMI screen. When data entry is enabled, the pop-up keypad displays in the selected position, and closed when data entry ends. If not selected: When data entry is enabled, the pop-up keypad is not displayed. Users may: ● Create a custom design on the same screen window. ● Use a USB keyboard. **System keypad** cMT / cMT X Series model has its own system keypad, select this checkbox to use the system keypad, or select [Custom keypad] to set up a customized keypad. **Animation Setting** When using a cMT / cMT X Series model with [Custom keypad] selected, the transition effect of the keypad window can be selected. See Chapter 13.5 Function Key in this manual for the list of effects. **Hide title bar** Use a keypad without the title bar. **Popup position** When using a cMT / cMT X Series model with [Custom keypad] selected, the position where the keypad pops up can be selected. The position can either be relative to HMI screen or relative to object. **Relative to HMI screen** **Relative to object**



Restart the keypad if input value is out of range

When entering data, if the value entered is not within the valid range, the system will automatically restart the keypad.

Other options (For cMT / cMT X Series)	**Display lower and upper limits** If selected, when entering a value, the range is displayed near the object. **Display previous value** If selected, when entering a value, the value before update is displayed near the object.	

To create a keyboard in current window, see “12 Keypad Design and Usage”.

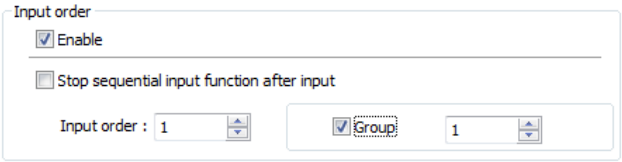
Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

Example 1

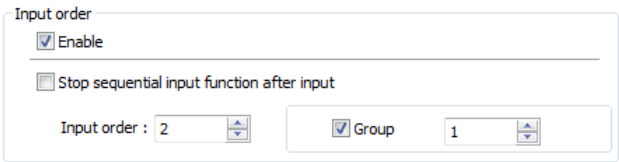
This example demonstrates how to use [Input Order] and [Group] to perform continuous input in several Numeric objects. After entering data in one object, entry will be passed to the next input order object which is in the same group.

1. Create three Numeric objects, and set [Input order] to 1, 2, and 3 respectively. Include the three objects in [Group 1] as shown in the following figure.

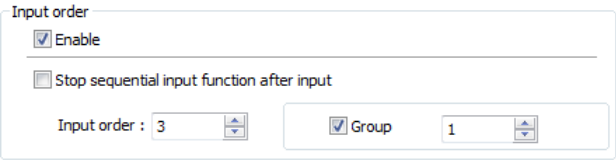
   LW-0

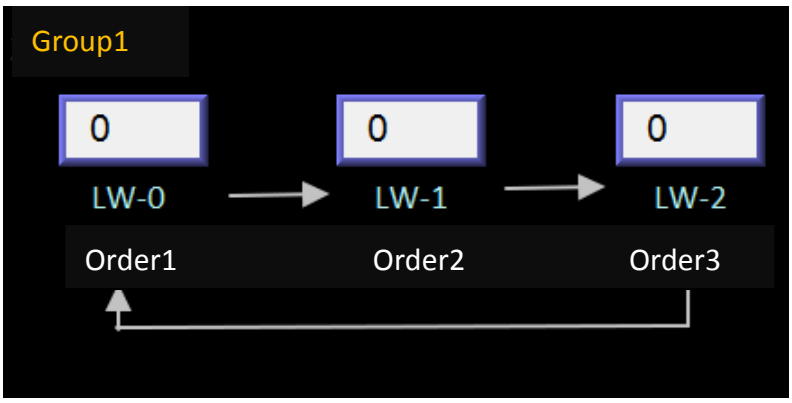


   LW-1

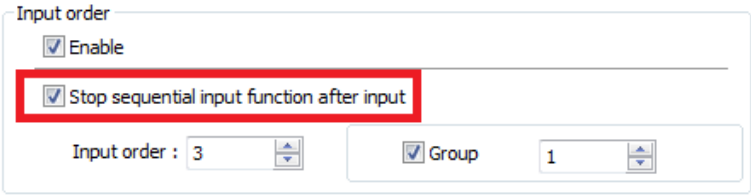


LW-2

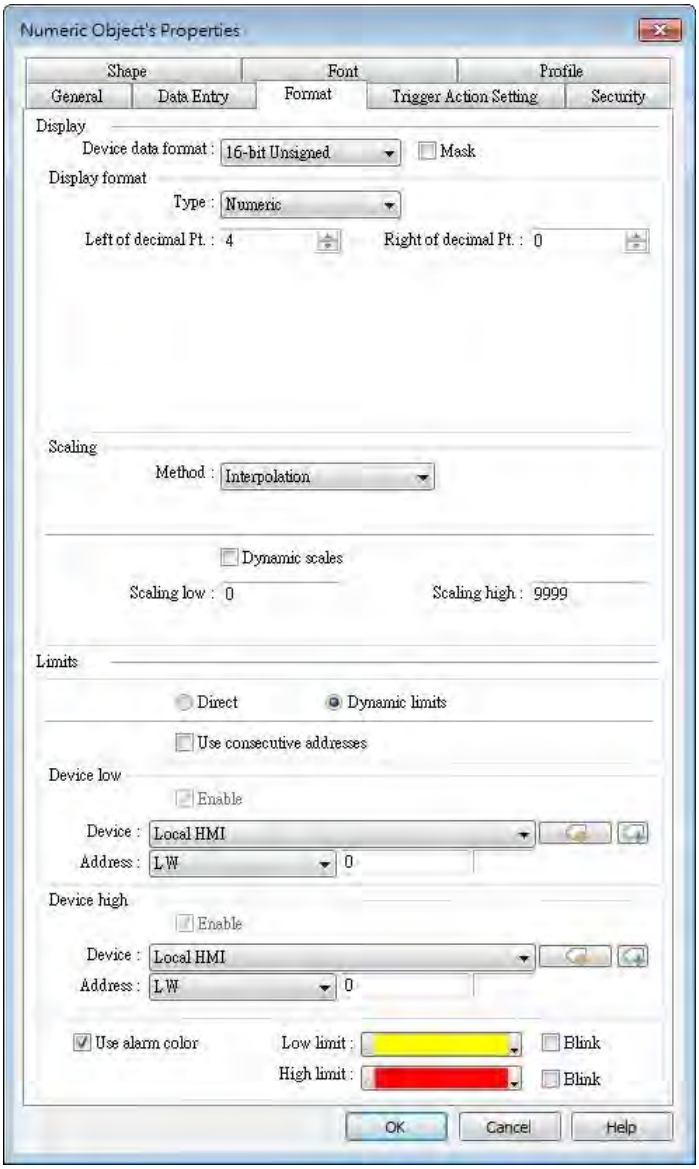




2. When finish entering data in the last object, to end data entry of all objects, please select [Stop sequential input function after input] check box.



Format Tab



Setting	Description
Device Data Format	Set the data format of a designated word register. Options include: BCD, Binary, Unsigned, Signed, HEX, and Float. 16-bit uses 1 word where 32-bit uses two words. **Mask** If selected, any values entered will be hidden by displaying them as asterisks “*”.
Display format	**Numeric** General Numeric formats. **Customized** Each “*” sign represents each integer digit, and each “#” sign

represents each fraction digit that will be displayed in the Numeric object. Float and Double formats are only supported on cMT / cMT X models.

Apart from “*” signs, extra texts can be entered in the [Format] field, for example: kg. The available alignment options are: [Left], [Center], [Right], [Leading zero].

The numeric value represented by “*” sign will be displayed from the highest digit to the lowest in the Numeric object.

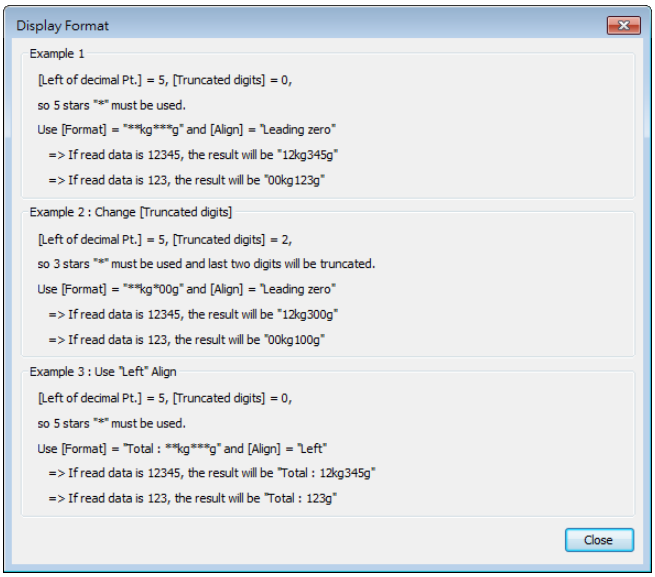
Truncated digits

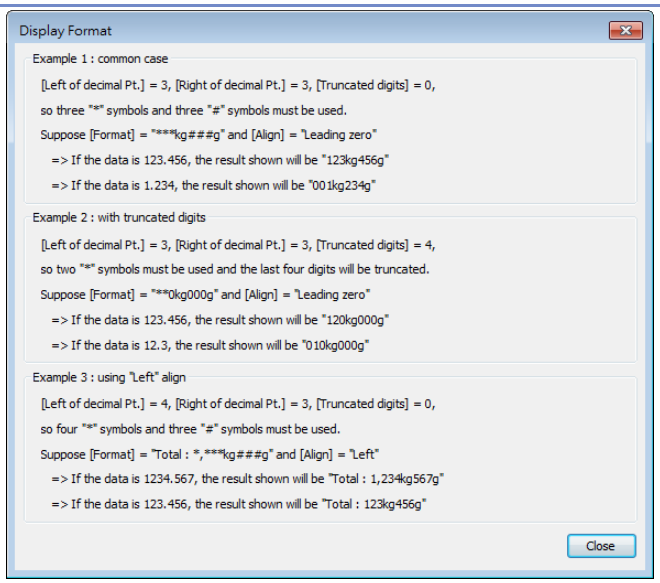
Specify the number of digits to be truncated, from the lowest digit to the highest.

The number of “*” signs = [Left of decimal Pt.] - [Truncated digits] - [Right of decimal Pt.].

The number of “#” signs = [Right of decimal Pt.] - [Truncated digits].

Please see the following examples:





When leading zero is not used, the text enclosed in two “*” signs will not show, for example:

[Left of decimal Pt.] = 5, [Truncated digits] = 0, and Display Format is “Total=**kg***g”

If the data read is 255, the result would be: "Total=255g"

If the data read is 1000, the result would be: "Total=1kg000g"

Scientific Notation

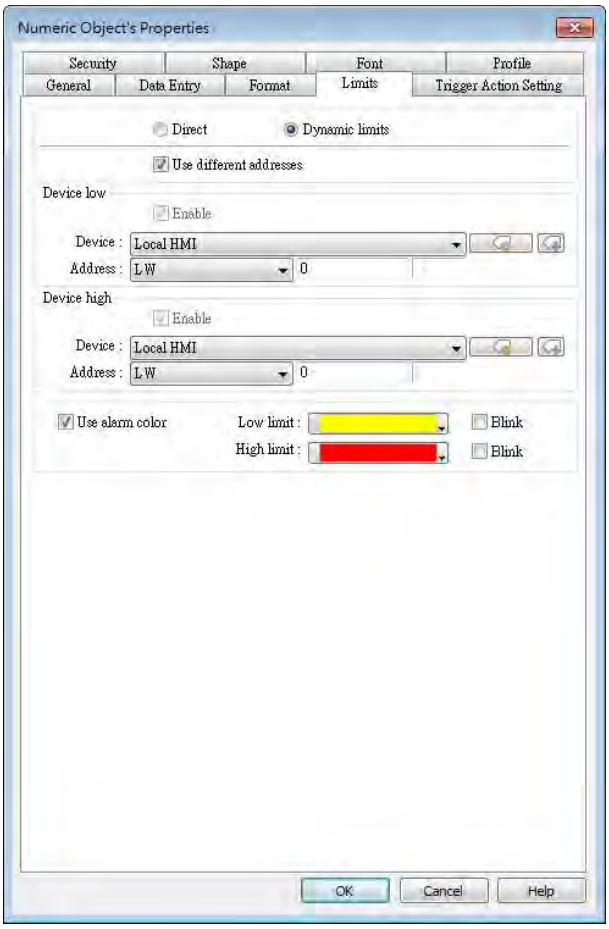
Express numerical data using scientific notation; for example, number 100 is displayed as 1e+2. Scientific notation is not available for device data formats: BCD, HEX, or Binary.

Number of digits	**Left of decimal Pt.** The number of digits before the decimal point. **Right of decimal Pt.** The number of digits after the decimal point.
Scaling	**Interpolation** If this check box is selected, [Engineering low] and [Engineering high] boxes appear. Values entered in these boxes correspond to the display range required. The setting also requires [Input low] and [Input high] in the limits section. See Example 2. Test: Preview the result of Interpolation. See Example 2. Dynamic scales: Set the [Engineering low] and [Engineering high] by a designated register. See Example 4. **Macro subroutine** The value read from or written to the register can be computed by macro subroutines selected in [Read conversion] and [Write conversion]. The macro subroutines should be defined in Macro

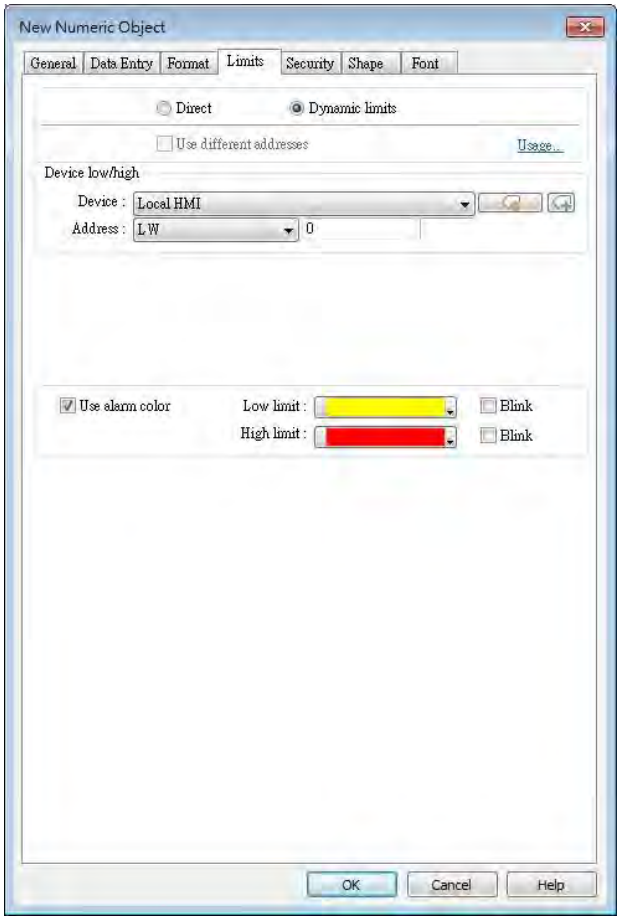
	Function Library. To use this feature, see “13.9.2.1 The rule of using Macro subroutine”.
Limits	This section allows users to apply display limits to the values held in the input register. The color when the register value is outside limits can be set. **Direct** Sets the limits by entering values in [Input low] and [Input high]. If the value entered is outside the limits, the value in the register cannot be changed. **Dynamic limits** Set the limits by a designated register. To use consecutive addresses, see Example 5. On a cMT / cMT X model, the [Use consecutive addresses] option can be deselected so that users can designate different addresses to set the high and low limits respectively. When the scaling method used is interpolation, both high and low limit addresses are required.
Use alarm color	**Low limit** When the value in the register is outside the [Low limit], display digits by the color set. **High limit** When the value in the register is outside the [High limit], display digits by the color set. **Blink** When the value in the register is outside either limit, the digits flash.

Limit Tab

cMT, cMT X



eMT, iE, XE, mTV Series

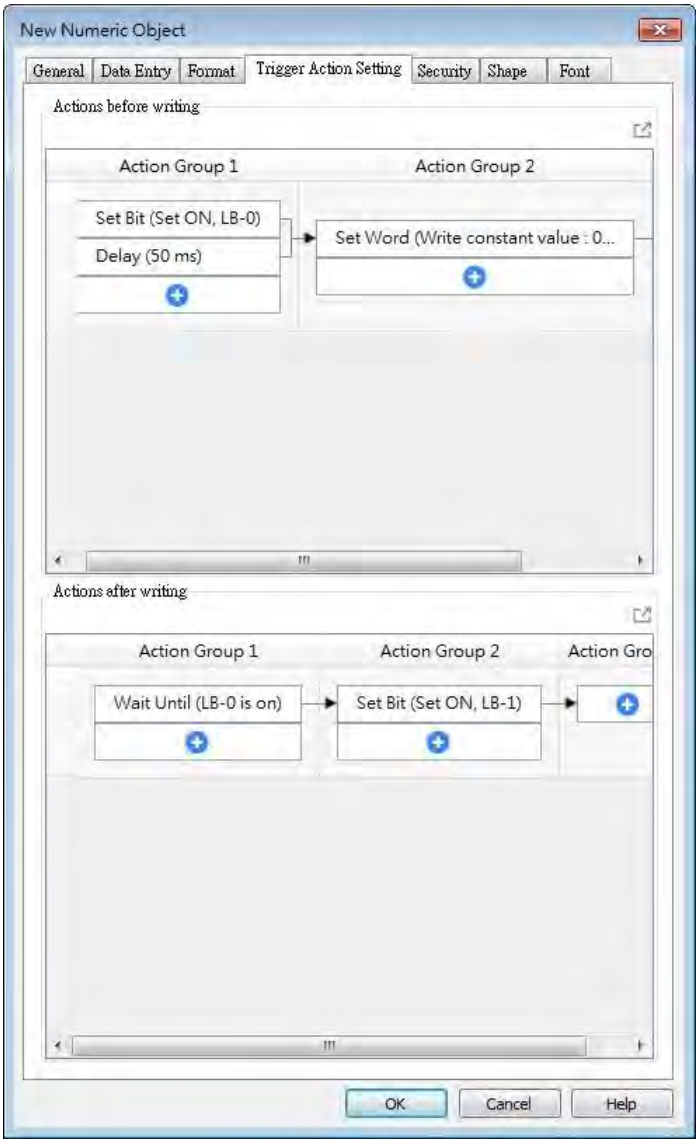


Setting	Description
Limits	This section allows users to apply display limits to the values held in the input register. The color when the register value is outside limits can be set. **Direct** Sets the limits by entering values in [Input low] and [Input high]. If the value entered is outside the limits, the value in the register cannot be changed. **Dynamic limits** Set the limits by a designated register. To use consecutive addresses, see Example 5. On a cMT / cMT X model, with [Use different addresses] option selected, the high and low limits can be set respectively by designating different addresses. When the scaling method used is

	interpolation, both high and low limit addresses are required.
Use alarm color	**Low limit** When the value in the register is outside the [Low limit], display digits by the color set. **High limit** When the value in the register is outside the [High limit], display digits by the color set. **Blink** When the value in the register is outside either limit, the digits flash.

Trigger Action Setting (cMT / cMT X)

Actions executed before/after writing can be classified into action groups. The groups are put in sequence. The actions within the same group are executed at the same time. When all the actions within the same group are completed, the actions in the next group are then executed. Compared to the Notification actions selected in General tab, Trigger Action Setting offers more flexible combination of actions. For more details about these settings, please also see Action Trigger in Chapter 13.



Setting	Description
Actions before writing	The predefined action groups will be executed once the write operation starts, and will not wait for it to be completed.
Actions after writing	The predefined action groups will be executed when the write operation is completed.

13.9.2.1. The rule of using Macro Subroutine

- There must be a return value and exactly one parameter.

Examples:

sub char test (short a) // (Correct)

sub test (char a) // (Incorrect, no return value.)

sub char test (char a, char b) // (Incorrect, two parameters.)

- Use the Macro data type that corresponds to the object's data format.

The mapping is as follows:

Macro Data Type	Numeric Object Data Format
short	16-bit Signed
Int	32-bit Signed
unsigned short	16-bit BCD, 16-bit HEX, 16-bit Binary, 16-bit Unsigned
unsigned int	32-bit BCD, 32-bit HEX, 32-bit Binary, 32-bit Unsigned
float	32-bit Float
long	64-bit Signed (cMT / cMT X only)
unsigned long	64-bit Unsigned (cMT / cMT X only)
double	64-bit Double (cMT / cMT X only)

For example, if the data format of the numeric object is 16-bit Unsigned, only the corresponding Macro data type: unsigned short, is available.

Examples:

sub char test(unsigned short a) // (Correct)

sub char test(char a) // (Incorrect)

- Supports only the local HMI address.

Examples:

GetData(var, "Local HMI", LB, 0, 1) // (Correct)

GetData(var, "MODBUS RTU", 0x, 0, 1) // (Incorrect)

- The following system defined functions are unable to be invoked:

ASYNC_TRIG_MACRO, SYNC_TRIG_MACRO, DELAY, FindDataSamplingDate, FindDataSamplingIndex, FindEventLogDate, FindEventLogIndex, INPORT, INPORT2, OUTPORT, PURGE, TRACE

- The following statements are not supported:

For-Next, While-Wend

Example 2

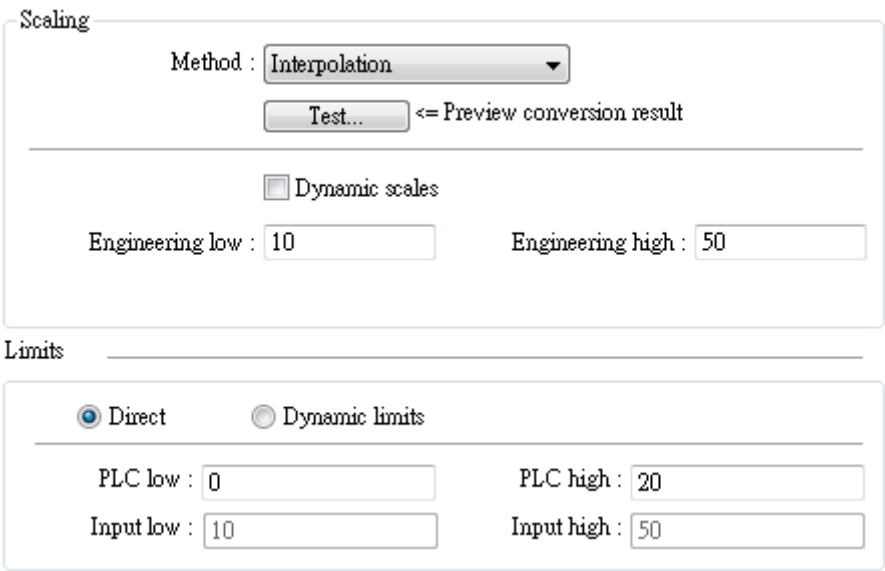
If [Interpolation] is selected, the scaling equation is as the following:

If A indicates the original data and B indicates the displayed data:

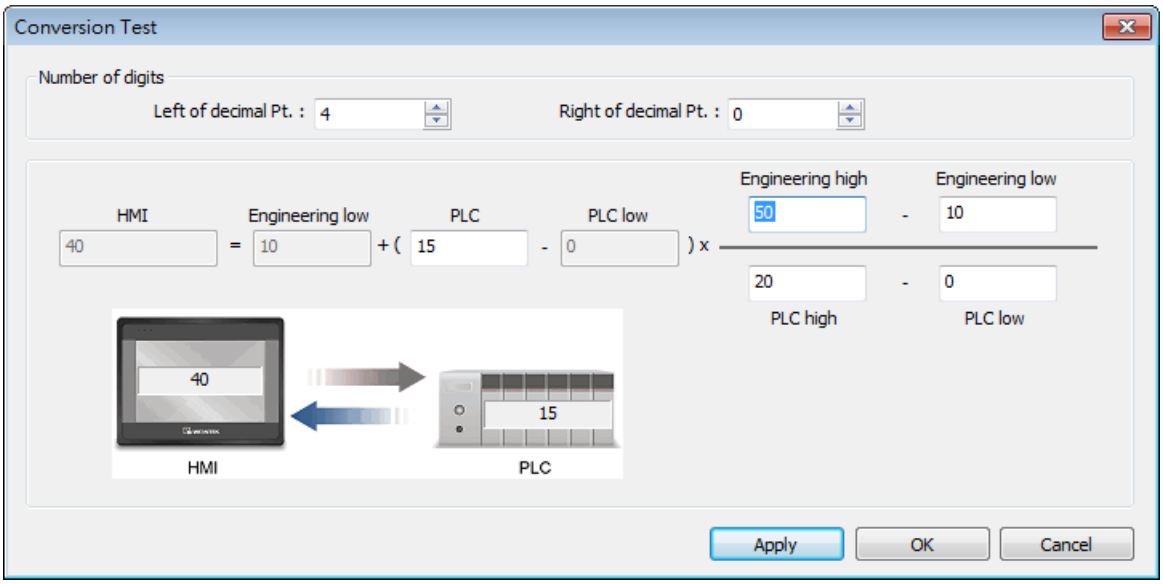
B = [Engineering low] + (A - [PLC low]) × Ratio

where, Ratio = ([Engineering high] - [Engineering low]) / ([PLC high] - [PLC low])

As shown below, the original data is 15, after conversion, 40 will be displayed.



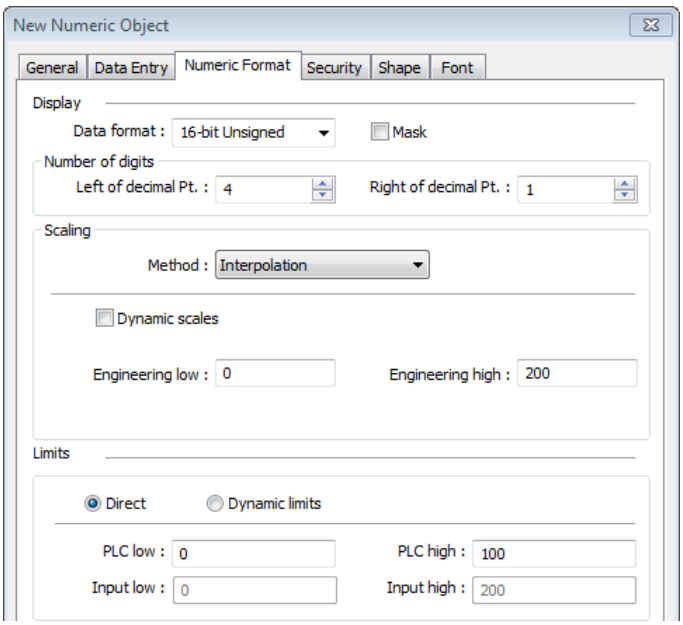
Click [Test] button to preview the result of Interpolation. Enter a value in [PLC] field as shown in the following figure, for example, enter value 15, and the result, which is 40, will be displayed.



Example 3

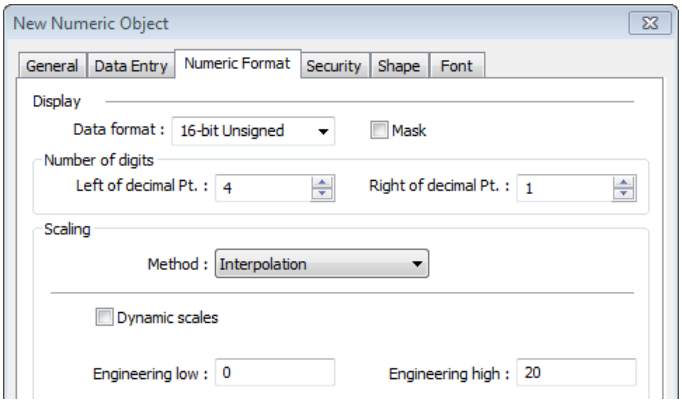
If the numeric format selected is not Float and decimal point is used, the decimal place of the converted result will not be adjusted automatically, please adjust [Engineering high] to correctly place the decimal point of the result gained in [Interpolation] mode. Please see the illustration below.

1. Create two Numeric objects, set [Right of decimal Pt.] to 1 and select [Interpolation] method for one of the objects as shown in the following figure.



2. Enter value “123”, the object set to [Interpolation] displays “246.0” instead of “24.6”.

3. To move the decimal point one place to the left, adjust [Engineering high] as shown in the following figure.



Example 4

If [Interpolation] is selected, set the [Engineering low] and [Engineering high] by a designated register. When Dynamic Address is LW-n, where n is an arbitrary number, the rule of setting [Engineering low] and [Engineering high] is:

Content	16-bit	32-bit	64-bit (cMT / cMT X only)
Dynamic address	LW-n	LW-n	LW-n
Engineering low	LW-n	LW-n	LW-n
Engineering high	LW-n+1	LW-n+2	LW-n+4

When address is LW-100, the rule of setting limits is:

Content	16-bit	32-bit	64-bit (cMT / cMT X only)
Dynamic address	LW-100	LW-100	LW-100
Engineering low	LW-100	LW-100	LW-100
Engineering high	LW-101	LW-102	LW-104

Example 5

Set the limits by a designated register. When [Address] is LW-n and [Use consecutive addresses] is selected, the rule of setting limits is:

Content	16-bit	32-bit	64-bit (cMT / cMT X only)
Address	LW-n	LW-n	LW-n
Low limit	LW-n	LW-n	LW-n
High limit	LW-n+1	LW-n+2	LW-n+4

When address is LW-100, the rule of setting limits is:

Content	16-bit	32-bit	64-bit (cMT / cMT X only)
Address	LW-100	LW-100	LW-100
Low limit	LW-100	LW-100	LW-100
High limit	LW-101	LW-102	LW-104

Example 6

The following demonstrates how to use [Macro subroutine] for scaling when configuring Numeric object.

The following two macros are used, one for [Read conversion] and one for [Write conversion].

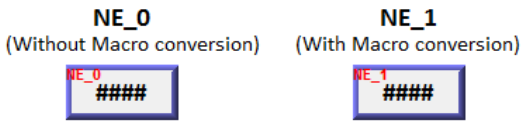
Read Conversion

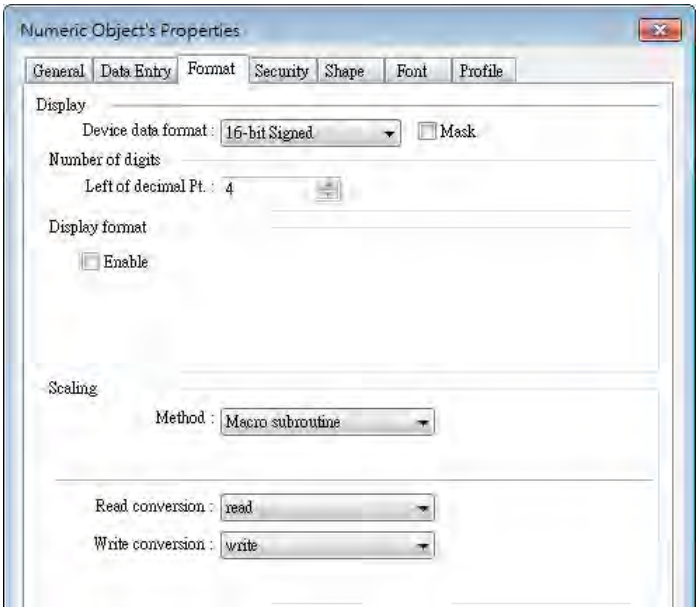
```
sub short read(short a)

short b
b = a + 10
return b

end sub
```

Write Conversion

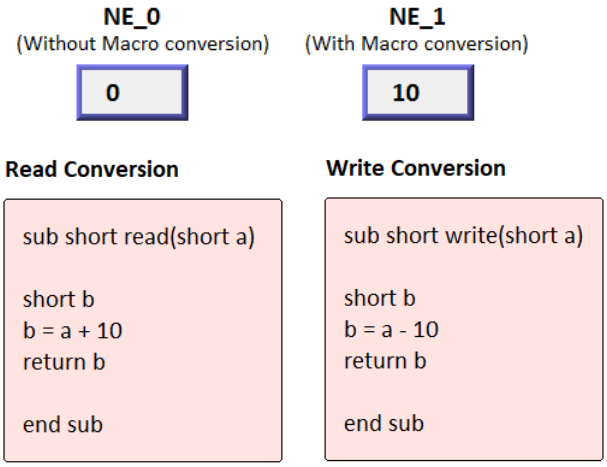
```
sub short write(short a)

short b
b = a - 10
return b

end sub
```

1. Create two Numeric objects: NE_0 and NE_1 and use the same control address. Select [Macro subroutine] for NE_1.

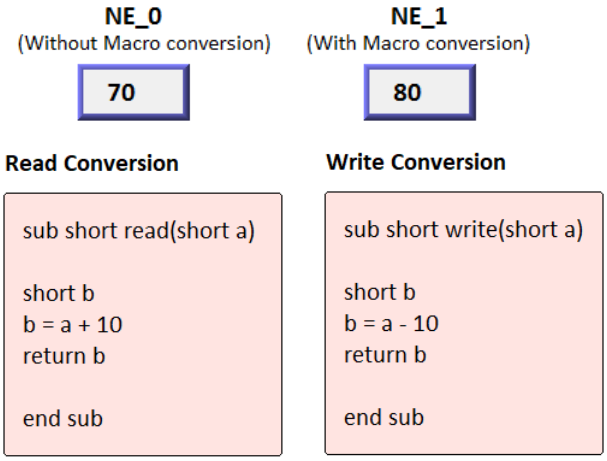




2. Enter 0 in NE_0 then NE_1 will execute [Read conversion]. The value gained will be 10.



3. Enter 80 in NE_1, [Write conversion] is executed and the value gained will be 70. NE_0 displays 70.

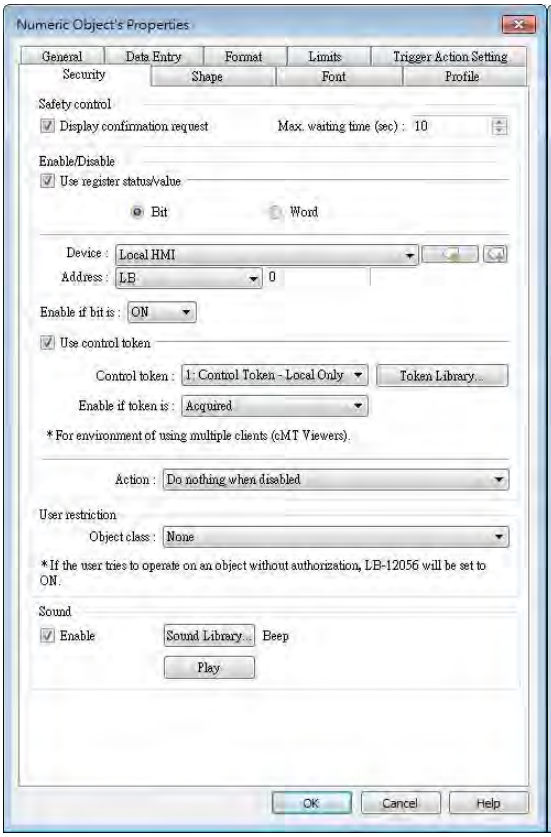


Note

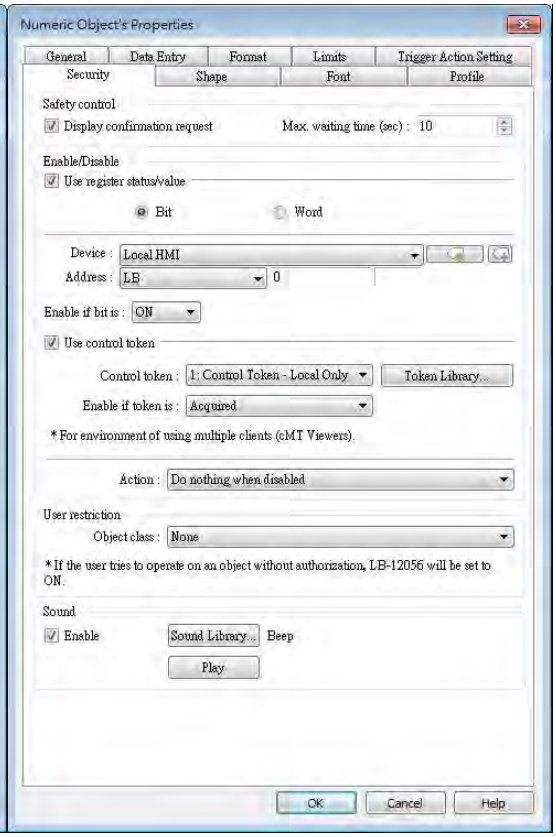
- If executing [Read conversion] and [Write conversion] by the same numeric object, the value entered in this object is computed by the Macro subroutine of [Write conversion] first, and then the result is computed by the Macro subroutine of [Read conversion]. In Example 5, if the subroutine of [Write conversion] is set to b=a-20, then entering 80 in NE_1 will get 60 after [Write conversion] and then the object displays 70 after [Read conversion].

Security Tab

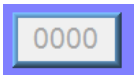
cMT, cMT X



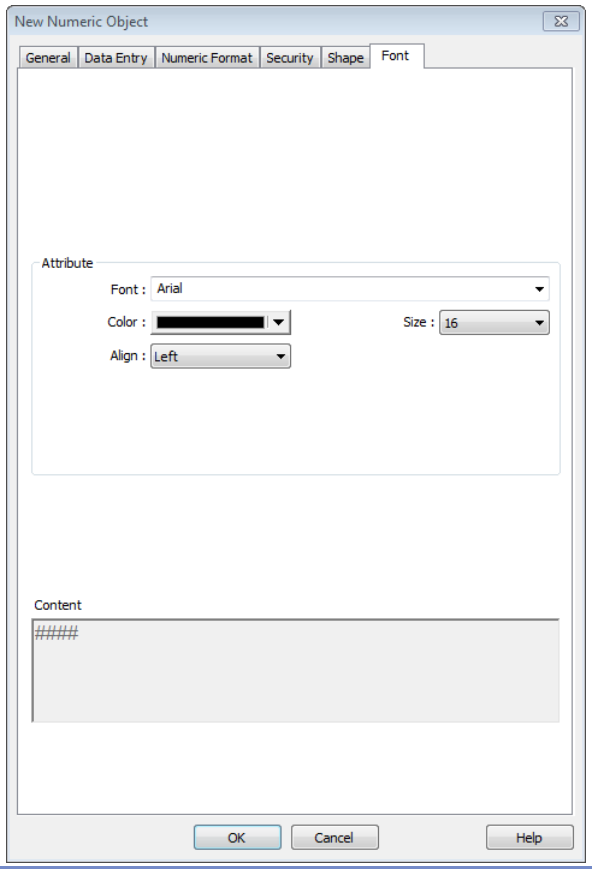
eMT, iE, XE, mTV Series



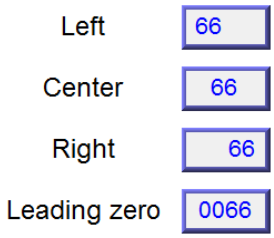
Setting	Description
Safety control	Displays a confirmation request window before executing a command.
Enable/Disable	**Use register status / value** With this option selected, whether an object is operable depends on whether the specified condition occurs in the designated address. **Use control token** With this option selected, whether an object is operable depends on whether a control token is acquired or unacquired.

	Action modes: **Do nothing when disabled** The object is not operable, and its status remains unchanged when the specified condition does not occur in the designated address. **Hide when disabled** The object is hidden when the specified condition does not occur in the designated address. **Grayed label when disabled** The value in the object turns gray when the specified condition does not occur in the designated address. 
User restriction	Set the security class of the object to be operated by an authorized user.
Sound	When enabled, the selected sound will be played when the object is touched.

Font Tab



Setting	Description
Color	When the value is within the limits, display digits using color set in

	this tab.
Align	**Left:** Align the number to the left. **Center:** Align the number to the center. **Right:** Align the number to the right. **Leading zero:** The number is preceded with leading zeros when the number of digits is less than that set. 
Size	Set the font size.

13.10. ASCII

13.10.1. Overview

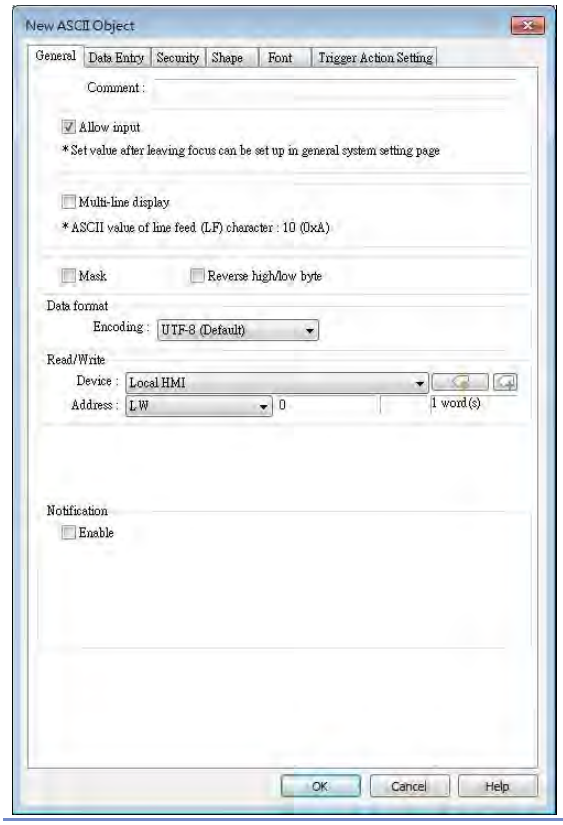
ASCII object can be used to input or display ASCII or UNICODE characters held in designated word registers.

13.10.2. Configuration

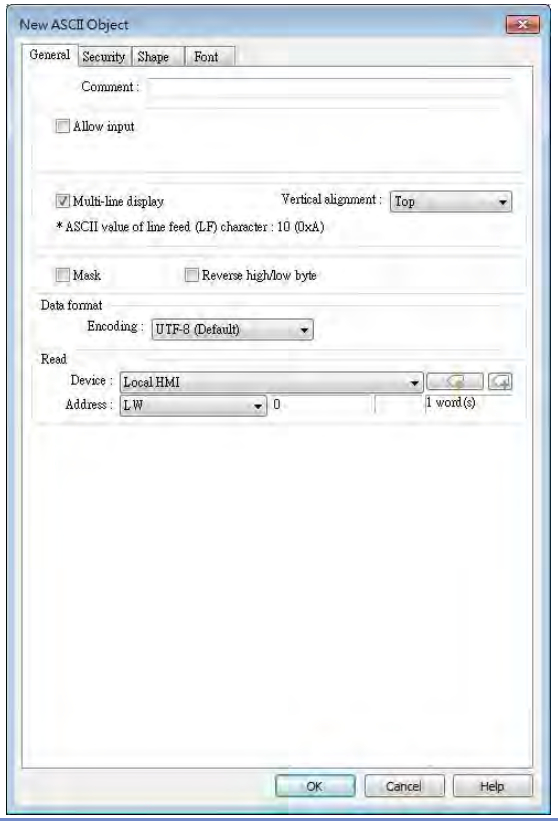
Click [Object] » [ASCII] icon on the toolbar to open an ASCII object property dialog box. Set up the properties, press OK button, and a new ASCII object will be created.

General Tab

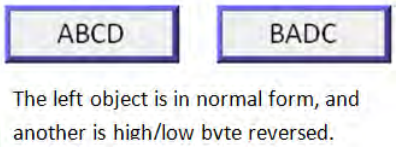
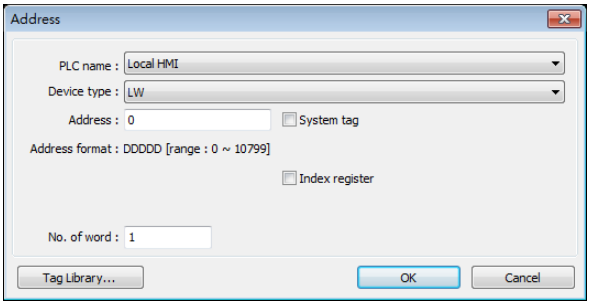
cMT, cMT X



eMT, iE, XE, mTV Series



Setting	Description
Allow input	If selected, the input features and relevant settings are enabled.
Multi-line display	If selected, the ASCII object can display multi-lined text. If a line feed character LF (0xA) is used in the string, a newline will be created.

Vertical alignment	When [Multi-line display] is enabled, the method to vertically align multiple lines of text can be selected.
Mask	If selected, any values entered will be masked by asterisks (*)..
Reverse high/low byte	Normally an ASCII code is displayed in “high byte”, “low byte” order. Reverse selection makes the system display ASCII characters in “low byte”, “high byte” order.  The left object is in normal form, and another is high/low byte reversed.
Data format	Select encoding from UTF-8 (default), Unicode, or Latin-1.
Read address	Click [Setting] to select the [PLC name], [Address], [Device type], [System tag], [Index register] of the word device that displays characters. Users can select a defined address tag from Address Tag Library, or set address in [General] tab while adding a new object.  **No. of words** Select the maximum number of words to be displayed. cMT / cMT X Series: max. 160 words eMT/XE/mTV/iP: max. 64 words
Notification	With notification enabled, the state of the designated bit address can be set on / off, before / after writing. Notification works differently comparing cMT/cMT X Series with eMT/iE/XE/mTV Series. **cMT/cMT X Series:** Trigger the object to open a keyboard window » Notification before writing » Enter a value and press ENTER » Start writing operation » Notification after writing. **eMT/iE/XE/mTV Series:** Trigger the object to open a keyboard window » Enter a value and press ENTER » Notification before writing » Start writing operation

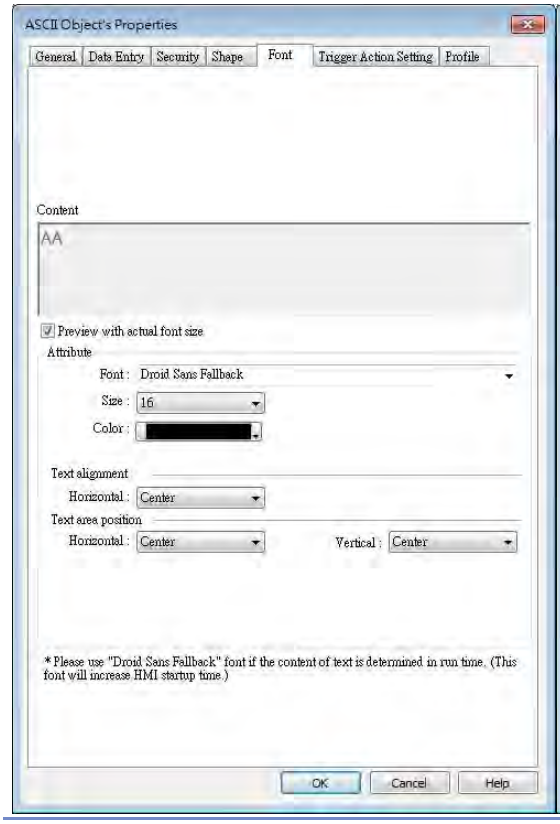
» Notification after writing.

Note

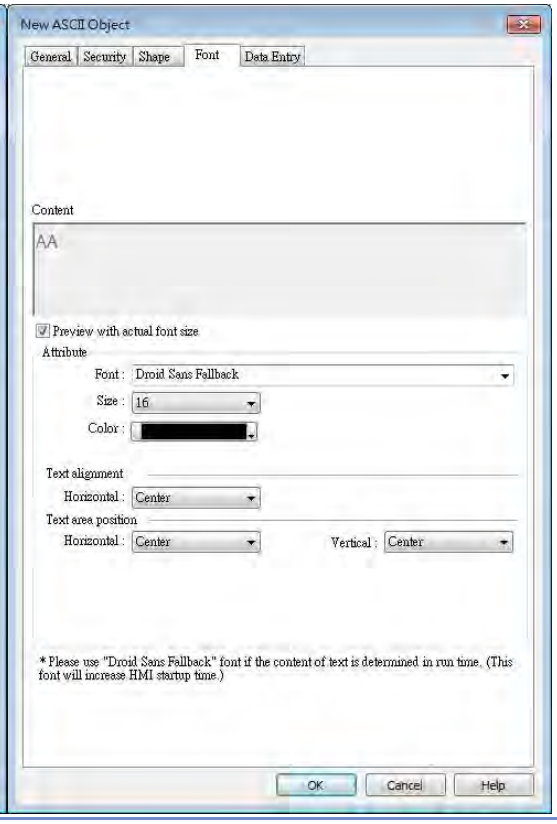
- A UNICODE character uses 1 word, and an ASCII character uses 1 byte. Therefore 1 word can be used as 1 UNICODE character or 2 ASCII characters. (1 word equals to 2 bytes)

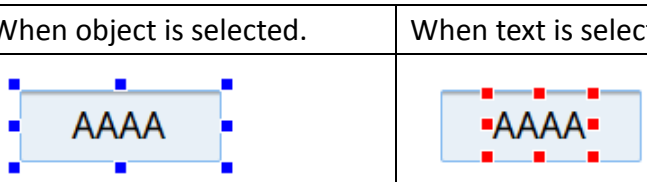
Font Tab

cMT, cMT X



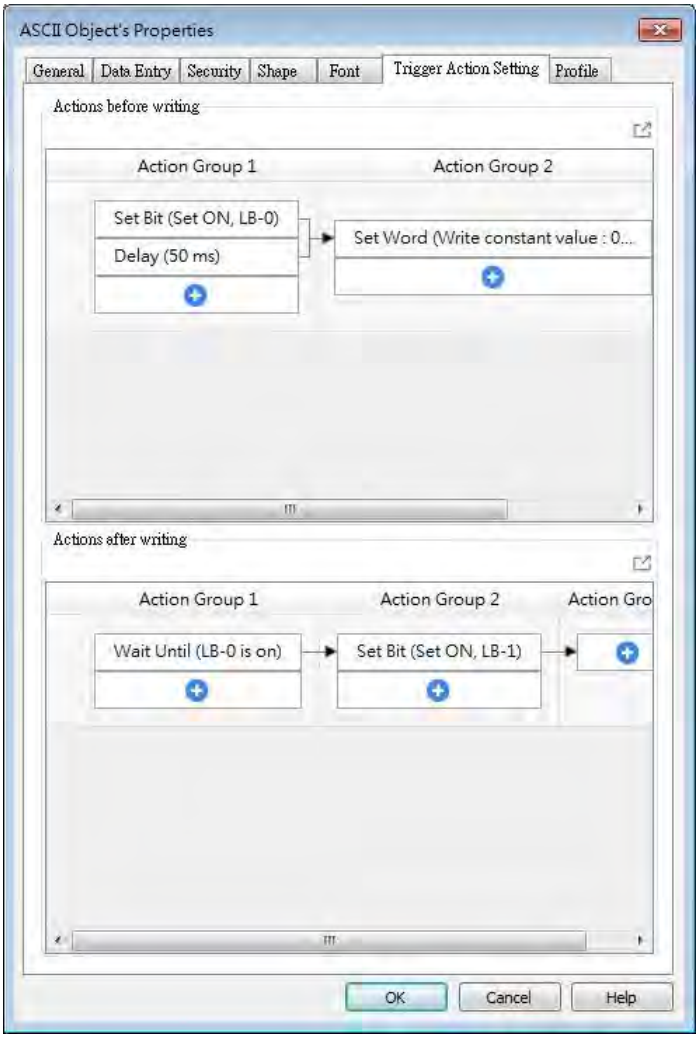
eMT, iE, XE, mTV Series



Setting	Description
Attribute	The font, size, color, and alignment can be set.
Text Alignment	Multiline text is aligned within the label outline. The method to align multiple lines of text can be selected.
Text area position	Multiline text is aligned within the object outline. The label outline is highlighted by red dotted line and the object outline is highlighted by blue dotted line. Please note that this feature is ineffective when [Multi-line display] is enabled in General tab. 

Trigger Action Setting (cMT / cMT X)

Actions executed before/after writing can be classified into action groups. The groups are put in sequence. The actions within the same group are executed at the same time. When all the actions within the same group are completed, the actions in the next group are then executed. Compared to the Notification actions selected in General tab, Trigger Action Setting offers more flexible combination of actions. For more details about these settings, please also see Action Trigger in Chapter 13.



Setting	Description
Actions before writing	The predefined action groups will be executed once the write operation starts, and will not wait for it to be completed.
Actions after writing	The predefined action groups will be executed when the write operation is completed.

13.11. Indirect Window

13.11.1. Overview

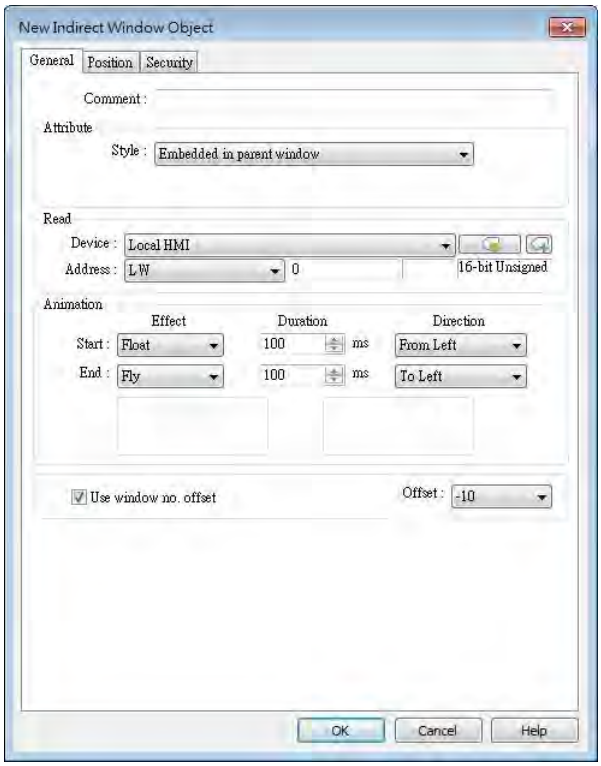
Indirect Window object opens or closes a pop-up window assigned by a designated word register. There are two ways to use Indirect Window object: The first is to use the profile of Indirect Window object, and let the pop-up window be resized and displayed in the defined profile; the second is to automatically resize the window according to the size of the pop-up window to be displayed. To close the pop-up window, assign 0 to the designated word register. The difference between Direct Window and Indirect Window is that Direct Window is controlled by a bit register, while Indirect Window is controlled by a word register.

13.11.2. Configuration

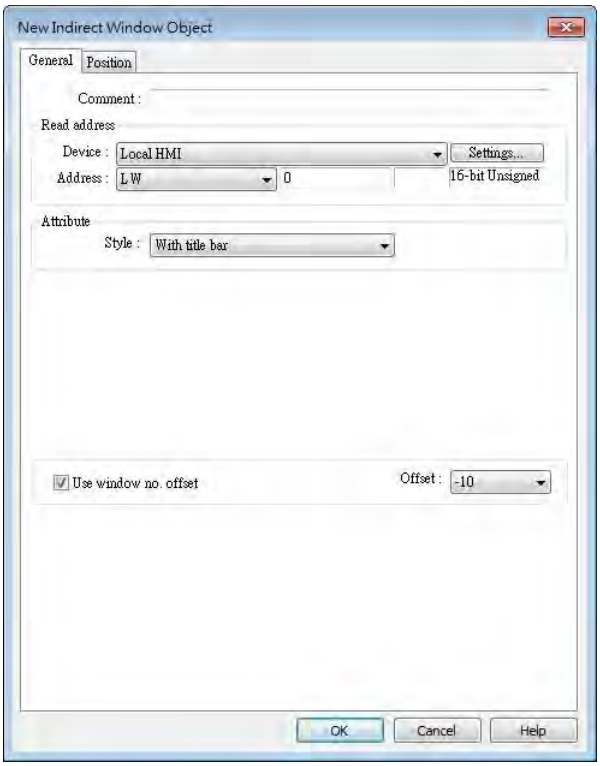
Click [Object] » [Embed Window] » [Indirect Window] icon on the toolbar to open the object property dialog box. Set up the properties, press OK button, and a new Indirect Window object will be created.

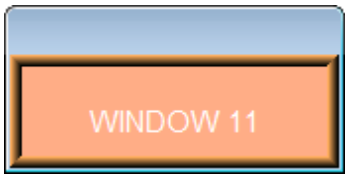
General Tab

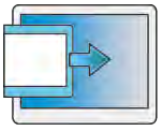
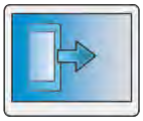
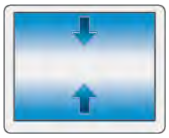
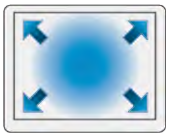
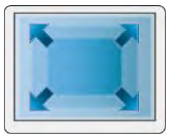
cMT / cMT X Series



eMT, iE, XE, mTV Series



Setting	Description
Style	This setting is available only on cMT / cMT X Series. When a base window contains an underlay window, this setting determines the display style of an Indirect Window placed in the underlay window. Please see Example 2 below. **Embedded in parent window** The Indirect Window is displayed in its parent window. When an Indirect Window is placed in an underlay window, the Indirect Window is displayed in the underlay window when it is called in the base window. **Popup window** The Indirect Window is displayed in the base window. When an Indirect Window is placed in an underlay window, the Indirect Window is displayed in the base window when it is called in the base window.
Read address	Click [Setting] to select the [PLC name], [Address], [Device type], [System tag], [Index register] of the word device that controls the pop-up window. Users can also set address in [General] tab while adding a new object.
Attribute	**Style** Set the display style of the pop-up window. There are two styles: ● No title bar The pop-up window has no title bar and cannot be dragged. ● With title bar The pop-up window has a title bar that can be dragged to move the window. 
Animation (cMT / cMT X Series)	**Effect** Different effects may be used for Start (window appears) and End (window disappears).

Effect Name	Transition
Fade	
Fly	
Float	
Wipe	
Split	
Circle	
Clock	
Zoom	
Turn	
Push	

Duration

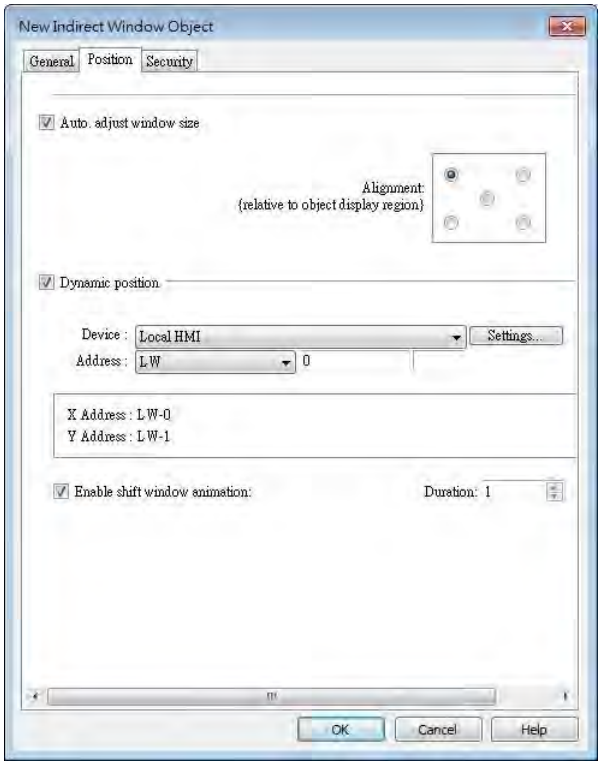
Specifies how many milliseconds (ms) a transition effect takes to complete.

Direction

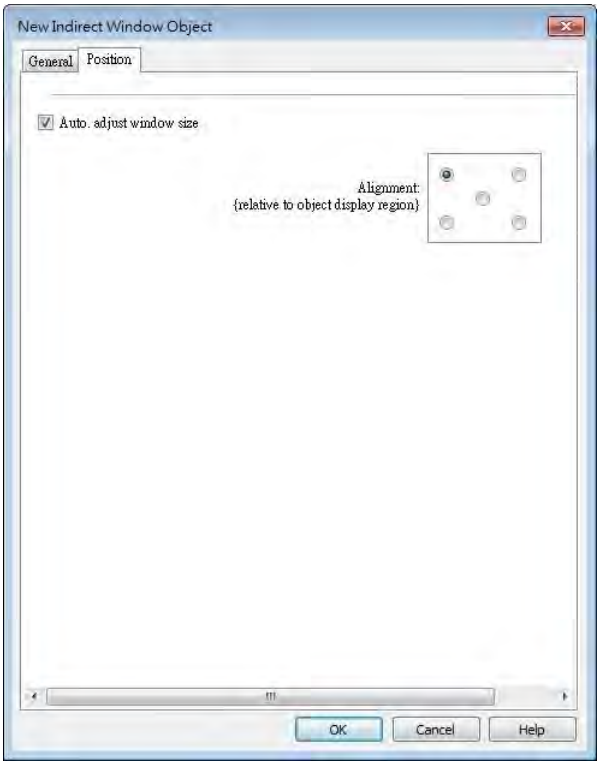
	The direction of the transition.
Use window no. offset	Sets the offset of the window number for selecting the pop-up window. The window number of the pop-up window is calculated by the value in the word register added to the offset. For example, assume the value in the register is 20 and offset is 5, the pop-up window number will be 25.

Position

cMT / cMT X Series



eMT, iE, XE, mTV Series



Setting	Description
Auto. adjust window size	Automatically resizes the Indirect Window and align the pop-up window to the preset region. **Alignment** Sets a reference point of the pop-up window from one of the five positions on the screen; for example, if the lower-right region is selected, the lower-right corner of the pop-up window is aligned to the lower-right region of the Indirect Window. See Example 1.
Dynamic position	With this option selected, the position of the window on the screen can be changed dynamically by the values in the designated address. **Enable shift window animation** With this option selected, the window changes position with transition effect.

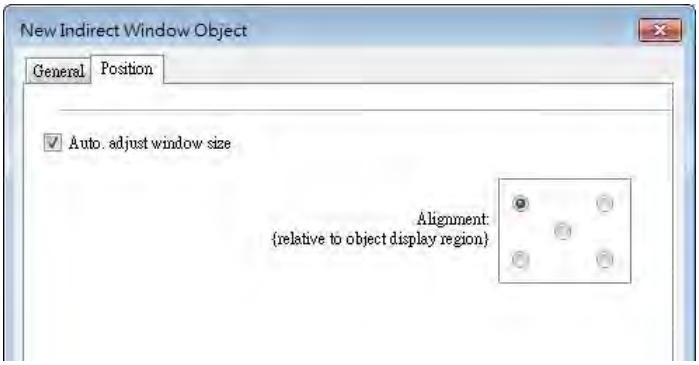
Example 1

Here is an example of using Indirect Window. The setting is shown in the following figure, set the address to LW-0 which assigns the window number. Create window no. 11 and 12 first.

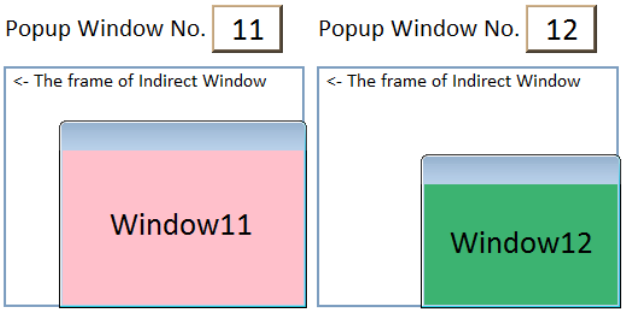
1. Create an Indirect Window object, set address to LW-0, and select [Auto. adjust window

size].

2. Select the region where the window is to be displayed.



3. Enter value 11 in LW-0, the pop-up window displayed is window no. 11.
4. Enter value 12 in LW-0, the pop-up window displayed is window no. 12.
5. Enter value 0 in LW-0, the pop-up window is closed.

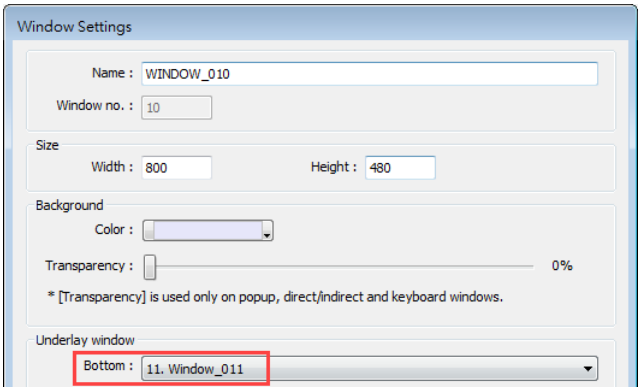


To close the pop-up window, apart from entering 0 in the designated word register, another way is to place a Function Key object in the pop-up window, and set the key to [Close window].

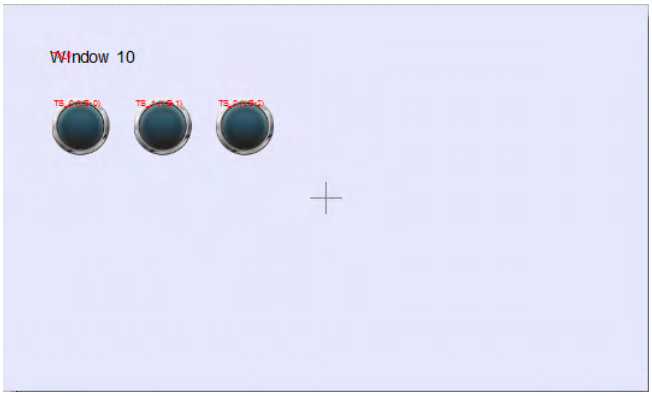
Example 2

Here is an example showing the difference between these two styles: [Embedded in parent window] and [Popup window]. The style setting determines how an Indirect Window placed in an underlay window is displayed when it is called in the base window.

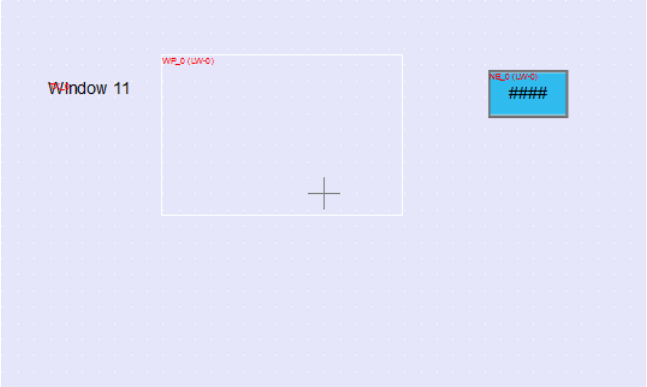
1. Add two base windows: Window no. 10 and Window no. 11. Add an underlay window in Window no. 10 and set Window no. 11 as the underlay window.



2. Create several objects in Window no. 10.



3. In Window no. 11 create an Indirect Window object and a Numeric Input object. Set both read addresses to LW-0.



4. By entering a window number in the Numeric Input object, the Indirect Window is displayed according to the selected style.

Embedded in parent window

Popup window

Note

- For eMT/iE/XE/mTV models, a screen can display up to 24 pop-up windows simultaneous including System Message Window, Direct Window and Indirect Window. For cMT / cMT X Series, the number of pop-up windows is not limited.
- The system does not allow opening the same window with two Direct (or Indirect) windows in one base window.
- If the pop up window has monopoly property enabled, then when the window pops up, all background windows may not be operated until the monopolizing window has been closed.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

13.12. Direct Window

13.12.1. Overview

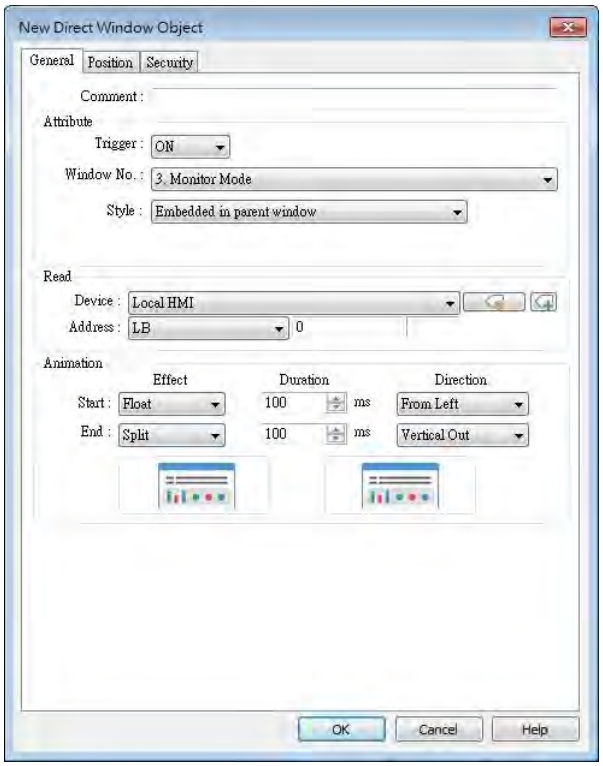
Direct Window object opens or closes a pop-up window assigned by a designated bit register. When the state of the bit register changes, the pop-up window appears at the predefined location. The display area for the pop-up window is limited by the size of predefined location. Returning the state of the bit register closes the pop-up window. The difference between Direct Window and Indirect Window is that Direct Window is controlled by a bit register, while Indirect Window is controlled by a word register.

13.12.2. Configuration

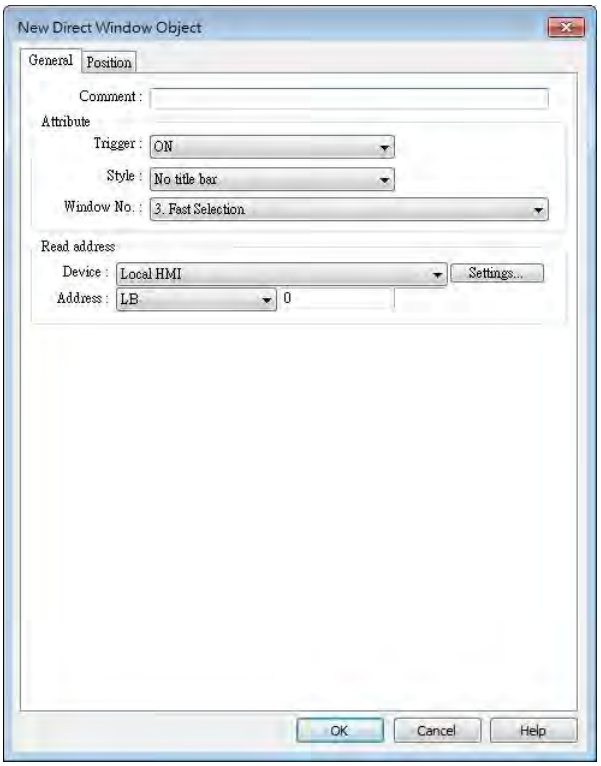
Click [Object] » [Embed Window] » [Direct Window] icon on the toolbar to open a Direct Window object property dialog box. Set up the properties, press OK button, and a new Direct Window object will be created.

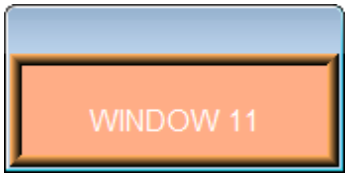
General Tab

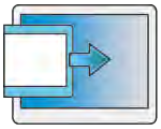
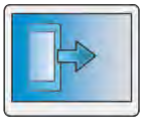
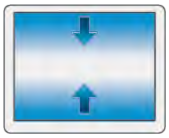
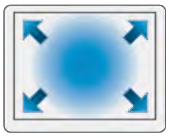
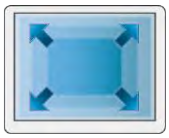
cMT / cMT X Series



eMT, iE, XE, mTV Series



Setting	Description
Style	This setting is available only on cMT / cMT X Series. When a base window contains an underlay window, this setting determines the display style of a Direct Window placed in the underlay window. Please see Example 2 in Chapter13.11 in this manual. **Embedded in parent window** The Direct Window is displayed in its parent window. When a Direct Window is placed in an underlay window, the Direct Window is displayed in the underlay window when it is called in the base window. **Popup window** The Direct Window is displayed in the base window. When a Direct Window is placed in an underlay window, the Direct Window is displayed in the base window when it is called in the base window.
Read address	Click [Setting] to select the [PLC name], [Address], [Device type], [System tag], [Index register] of the bit device that controls the pop-up window. Users can also set address in [General] tab while adding a new object.
Attribute	**Style** Set the display style of the pop-up window. There are two styles: ● No title bar The pop-up window has no title bar and cannot be dragged. ● With title bar The pop-up window has a title bar that can be dragged to move the window.  **Window no.** Set the pop-up window number.
Animation (cMT / cMT X Series)	**Effect** Different effects may be used for Start (window appears) and End (window disappears).

Effect Name	Transition
Fade	
Fly	
Float	
Wipe	
Split	
Circle	
Clock	
Zoom	
Turn	
Push	

Duration

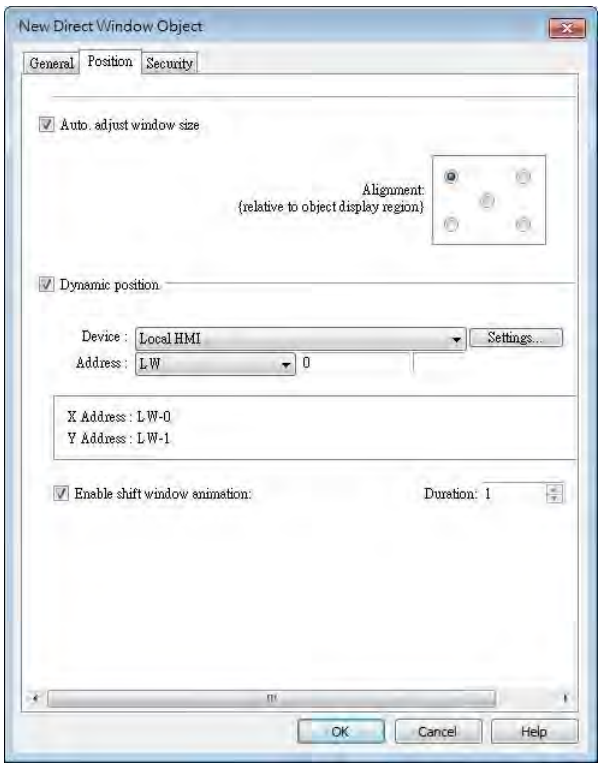
Specifies how many milliseconds (ms) a transition effect takes to complete.

Direction

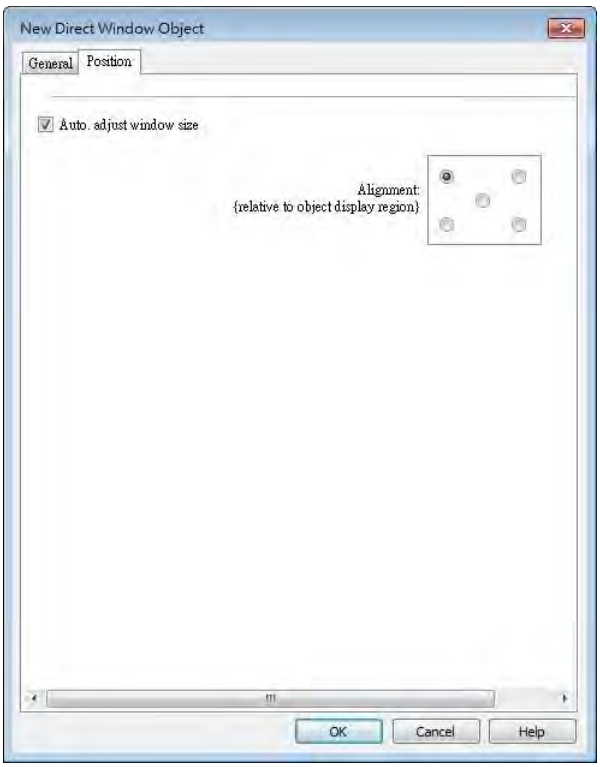
	The direction of the transition.

Position

cMT / cMT X Series



eMT, iE, XE, mTV Series

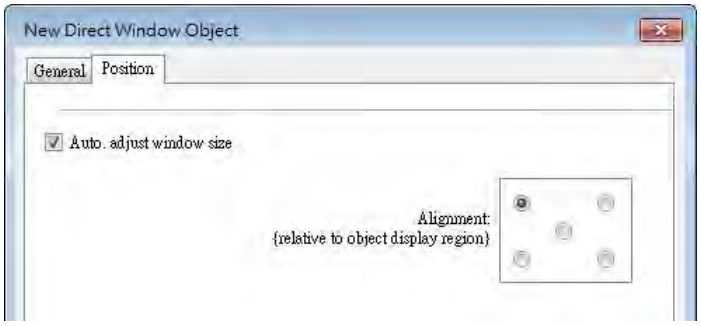


Setting	Description
Auto. adjust window size	Automatically resizes the Direct Window and align the pop-up window to the preset region. **Alignment** Sets a reference point of the pop-up window from one of the five positions on the screen; for example, if the lower-right region is selected, the lower-right corner of the pop-up window is aligned to the lower-right region of the Direct Window. See Example 1.
Dynamic position	With this option selected, the position of the window on the screen can be changed dynamically by the values in the designated address. **Enable shift window animation** With this option selected, the window changes position with transition effect.

Example 1

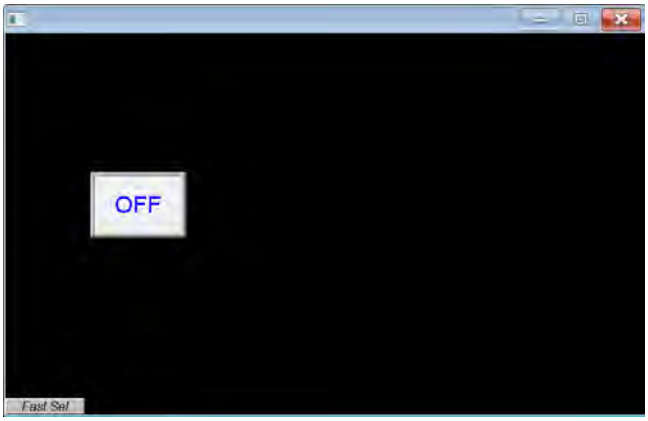
Create window no. 11 which can be controlled by a Toggle Switch with address LB-0.

1. Create a Direct Window object and set read address to LB-0.
2. In this example, the reference point for alignment is set to the lower-right region.



3. When LB-0's state is ON, window no. 11 will show.

4. When LB-0's state is OFF, window no. 11 will be hidden.



Note

- For eMT/iE/XE/mTV models, a screen can display up to 24 pop-up windows simultaneous including System Message Window, Direct Window and Indirect Window. For cMT / cMT X Series, the number of pop-up windows is not limited.
- The system does not allow opening the same window with two Direct (or Indirect) Windows in one base window.
- If the pop up window has monopoly property enabled, then when the window pops up, all background windows may not be operated until the monopolizing window has been closed.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

13.13. Moving / Rotating Shape

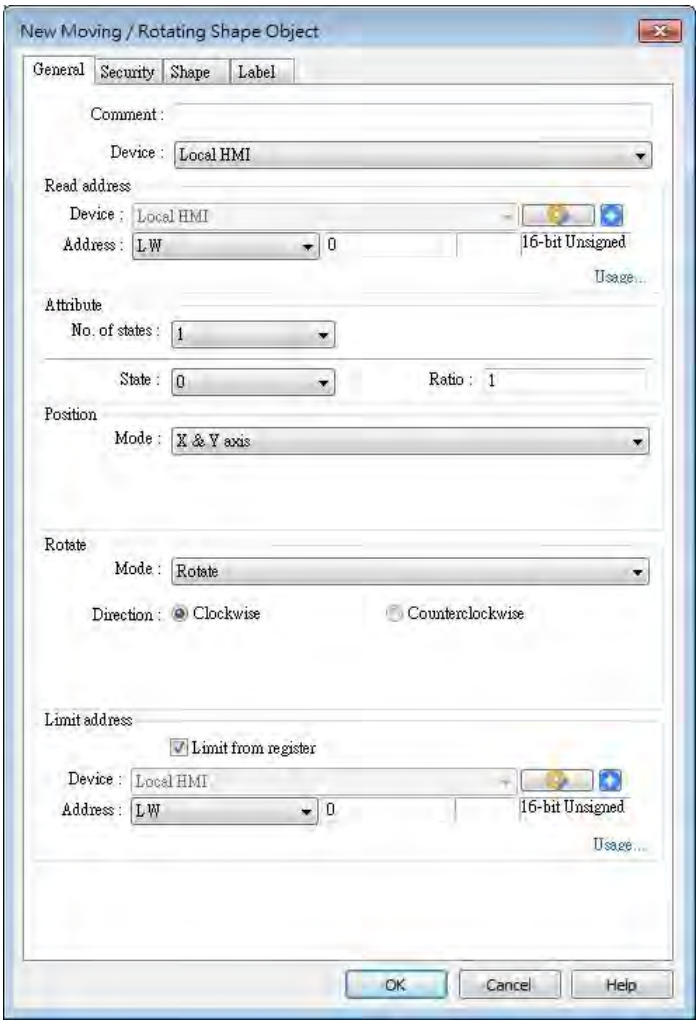
13.13.1. Overview

Moving / Rotating Shape is an object that changes its state, moves, and/or rotates according to the user-defined parameters. The state, moving distance, and rotation direction/angle are determined by values in consecutive registers.

13.13.2. Configuration

Click [Object] » [Animation] » [Moving / Rotating Shape] icon on the toolbar to create the object, set up the properties, press OK button, and a new Moving / Rotating Shape object will be created.

General Tab



Setting	Description
Read address	Click [Setting] to configure the [Device], [Device type], [Address], [System tag], or [Index register] of the word devices that control the display of object's state and moving distance. Users can also set the address in [General] tab while adding a new object.
Attribute	Select the object's movement mode and range. See "13.13.2.1 Illustration of Modes" in the following part.
Display ratio	The size of shape in different states can be set individually as shown in the following figure.
Rotate	Select Clockwise or Counterclockwise for Direction and define the

	angle range (unit in degree) within which the object may rotate. The allowable range is 0~360 degrees, and the Max. angle should be greater than Min. angle. For more information please see “13.13.2.2 Rotation Modes”.
Limit address	The object’s moving range can be set by adjusting the data in the designated register, see Example 1.

Example 1

Supposed that the object’s moving range is limited by register LW-n, the addresses in the following table are used to limit the moving / rotating range.

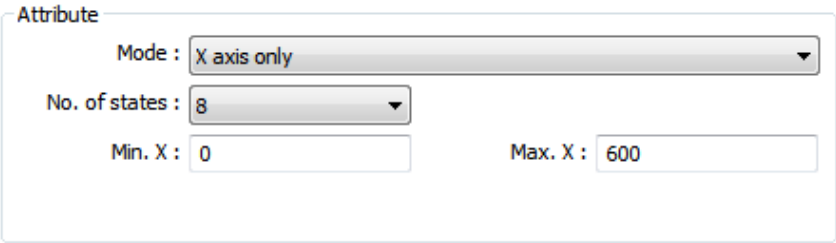
Data format	16-bit	32-bit	64-bit (cMT / cMT X only)
Position - [Min. X] address	LW-n	LW-n	LW-n
Position - [Max. X] address	LW-n+1	LW-n+2	LW-n+4
Position - [Min. Y] address	LW-n+2	LW-n+4	LW-n+8
Position - [Max. Y] address	LW-n+3	LW-n+6	LW-n+12
Rotate – Input Low	LW-n+4	LW-n+8	LW-n+16
Rotate – Input High	LW-n+5	LW-n+10	LW-n+20
Rotate – Scaling Low	LW-n+6	LW-n+12	LW-n+24
Rotate – Scaling High	LW-n+7	LW-n+14	LW-n+28

13.13.2.1. Illustration of Modes

Available modes are: (Assume Read Address is LW-n)

- X axis only

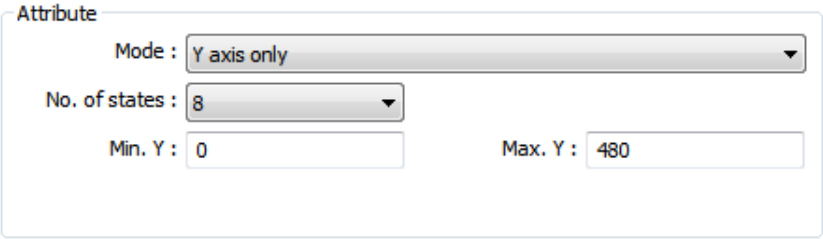
The object is only allowed to move along the X-axis. The moving distance ranges from [Min. X] to [Max. X].



Data format	16-bit	32-bit	64-bit (cMT / cMT X only)
Object state	LW-n	LW-n	LW-n
Moving distance on X-axis	LW-n+1	LW-n+2	LW-n+4

- Y axis only

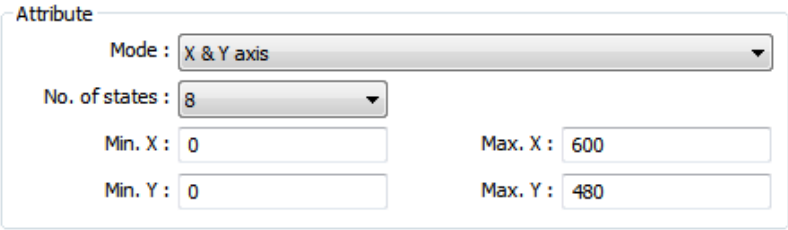
The object is only allowed to move along the Y-axis. The moving distance ranges from [Min. Y] to [Max. Y].



Data format	16-bit	32-bit	64-bit (cMT / cMT X only)
Object state	LW-n	LW-n	LW-n
Moving distance on Y-axis	LW-n+1	LW-n+2	LW-n+4

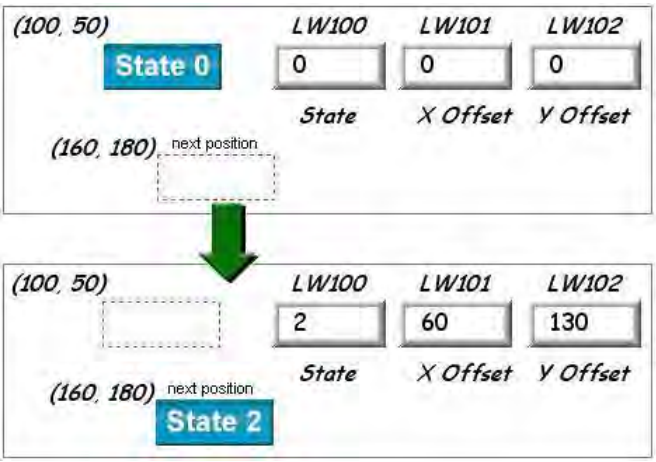
- X & Y axis

The object is allowed to move along the X-axis and Y-axis. The moving range in X and Y directions is defined by [Min. X], [Max. X] and [Min. Y], [Max. Y] respectively.



Data format	16-bit	32-bit	64-bit (cMT / cMT X only)
Object state	LW-n	LW-n	LW-n
Moving distance on X-axis	LW-n+1	LW-n+2	LW-n+4
Moving distance on Y-axis	LW-n+2	LW-n+4	LW-n+8

For example, if the object's read address is LW-100 and the data format is [16-bit Unsigned], LW-100 is used to control the object's state, LW-101 is used to control the object's moving distance on the X-axis, and LW-102 is used to control the object's moving distance on the Y-axis. The following figure shows that the object's read address is LW-100 and initial position is (100, 50). To move the object to the position (160,180) and change its state to State 2, assign 2 to LW-100, 160-100 = 60 to LW-101, 180-50 = 130 to [LW102].



- X axis w/ scaling

The object moves in X-axis only with scaling. Suppose that the value of the designated register is DATA, the system uses the following equation to calculate the moving distance on the X-axis.

$$\text{Displacement}=(\text{Data-[Input low]})\times\frac{\text{[Scaling high]-[Scaling low]}}{\text{[Input high]-[Input low]}}$$

Data format	16-bit	32-bit	64-bit (cMT / cMT X only)
Object state	LW-n	LW-n	LW-n
Moving distance on X-axis	LW-n+1	LW-n+2	LW-n+4

- Y axis w/ scaling

The object is for Y axis movement with scale, and the equation to calculate the moving distance on the Y-axis is the same as the one in [X axis w/ scaling].

Data format	16-bit	32-bit	64-bit (cMT / cMT X only)
Object state	LW-n	LW-n	LW-n
Moving distance on Y-axis	LW-n+1	LW-n+2	LW-n+4

- X axis w/ reverse scaling

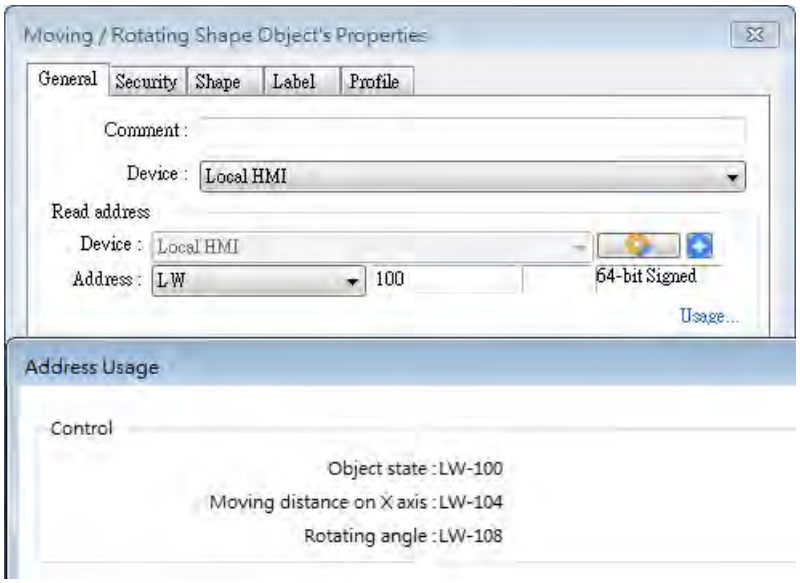
This works in the way as [X axis w/ scaling], but the moving direction is in reverse.

- Y axis w/ reverse scaling

This works in the way as [Y axis w/ scaling], but the moving direction is in reverse.

13.13.2.2. Rotation Modes

The addresses that control the moving / rotating shape are consecutive addresses starting from Read Address (LW-n), and they may vary depending on the selected mode. Please click [Usage…] to open the window that shows the control addresses relating to rotation parameters.



- Rotate

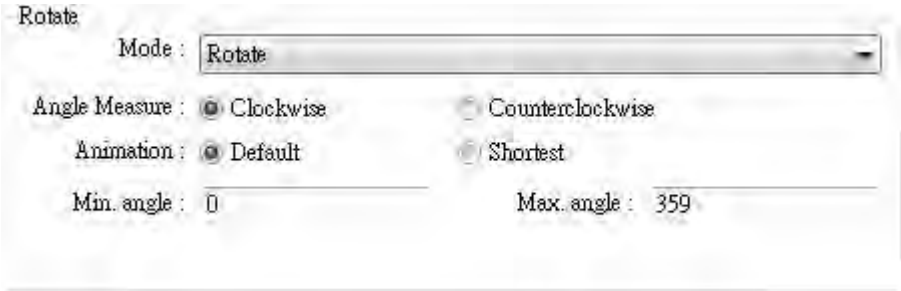
  Select rotating direction from clockwise or counterclockwise and set the Min. and Max. angle range.

  When rotation animation is enabled, two modes can be selected: Default and Shortest.

  e.g. When the angle measure is set to clockwise and the rotating angle is changed from 10 degrees to 350 degrees, these two modes give different results:

  Default: rotates 340 degrees clockwise (linear interpolation between the two numbers)

  Shortest: rotates 20 degrees counterclockwise (in the direction that produces the shortest animation path)



- Rotate w/ scaling

  Select rotating direction from clockwise or counterclockwise. The angle of rotation is calculated by the formula as shown below.

$$\text{Angle of Rotation} = (\text{Data-[Input low]}) \times \frac{\text{[Scaling high]-[Scaling low]}}{\text{[Input high]-[Input low]}}$$

- Rotate w/ reverse scaling

  Similar to [Rotate w/ scaling] but reverses the object from clockwise to counterclockwise and vice versa.

13.14. Animation

13.14.1. Overview

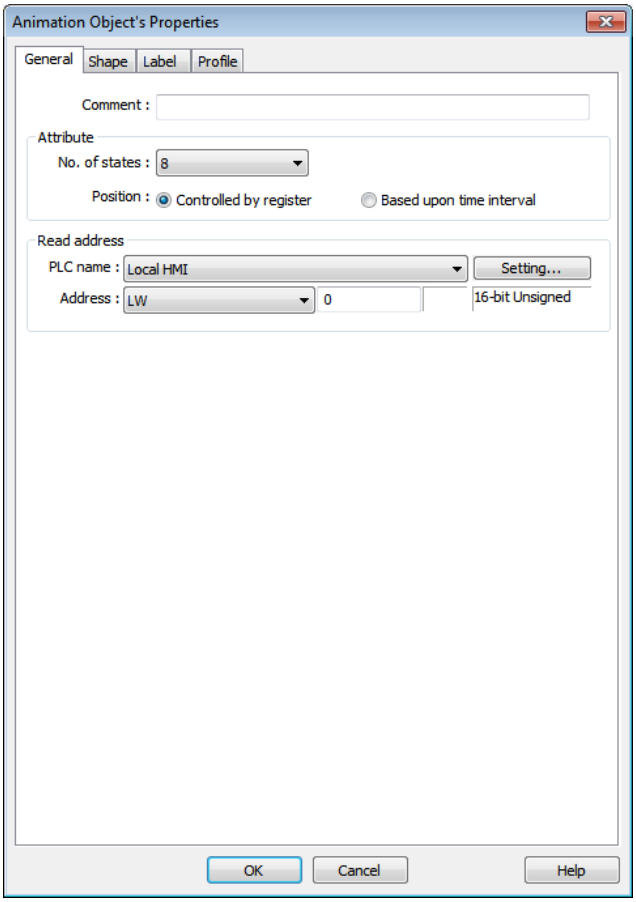
Animation object is defined by a pre-defined point set and states. Animation object will then move to a given point in a given state defined by designated registers. The object state and position depend on current value of two consecutive registers. The first register controls the state of the object and the second register controls the position along the predefined path.

13.14.2. Configuration

Click [Object] » [Animation] » [Animation] icon on the toolbar. First, create the pre-defined path. Move the mouse to each moving position, and click the left button to define positions one by one. When it is done, right click on the screen, set up the properties, press OK button, and a new Animation object will be created.

To change the object's attributes, double click on the object to open Animation Object's Properties dialog box.

General Tab



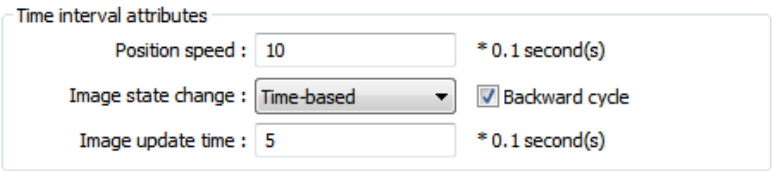
Setting	Description
Attribute	**No. of states** Configure the number of states for this object. **Controlled by register** Use the designated registers to control the object's state and position. See Example 1. **Based upon time interval** The object's state and position will change from time to time. [Time interval attributes] is used to set the time interval for states and positions.  **Position speed:** The speed of movement. The unit is 0.1 second. Supposed that [Speed] is set to 10, the object's position will change each second.

Image state change: Determines how state changes, either [Position dependent] or [Time-based]. If [Position dependent] is selected, the object state will change when position changes. If [Time-based] is selected, the object position will change based on [Position speed] and the object state will change based on [Image update time].

Backward cycle: Assumed the object has four positions: position 0, position 1, position 2, and position 3, and [Backward cycle] is not selected. When the object moves to the last position (position 3), the next position will be back to the initial position 0, and repeat. The moving path is shown as follows:

position 0 → position 1 → position 2 → position 3 → position 0 → position 1 → position 2…

If [Backward cycle] is selected, when the object moves to the last position (position 3), it will move backwards to position 2, position 1 and then the initial position 0, and start over again. The moving path is shown as follows.

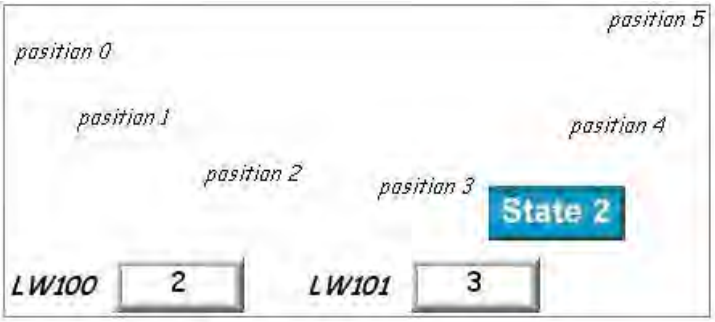
position 0 → position 1 → position 2 → position 3 → position 2 → position 1 → position 0…

Example 1

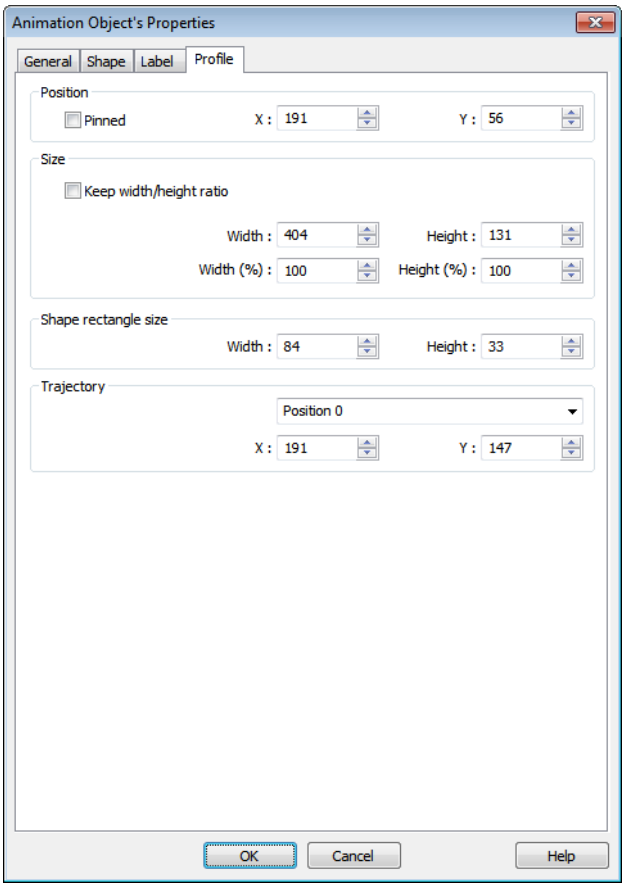
The object's state and position are determined by the registers, and the addresses must be configured correctly, as in the following table:

Data format	16-bit	32-bit	64-bit (cMT / cMT X only)
Object state	LW-n	LW-n	LW-n
Object position	LW-n+1	LW-n+2	LW-n+4

For example, if the designated register is LW-100 and the data format is [16-bit Unsigned], then LW-100 represents object's state, LW-101 represents position. In the picture below, LW-100 = 2, LW-101 = 3, so the object's state is 2 and position is 3.



Profile Tab



Setting	Description
Shape rectangle size	Set the size of the shape.
Trajectory	Set the position of each point on the moving path.

Note

- Since multiple pictures might be used by an [Animation] object, [Set to original dimension] will not return all pictures to the original size.

13.15. Bar Graph

13.15.1. Overview

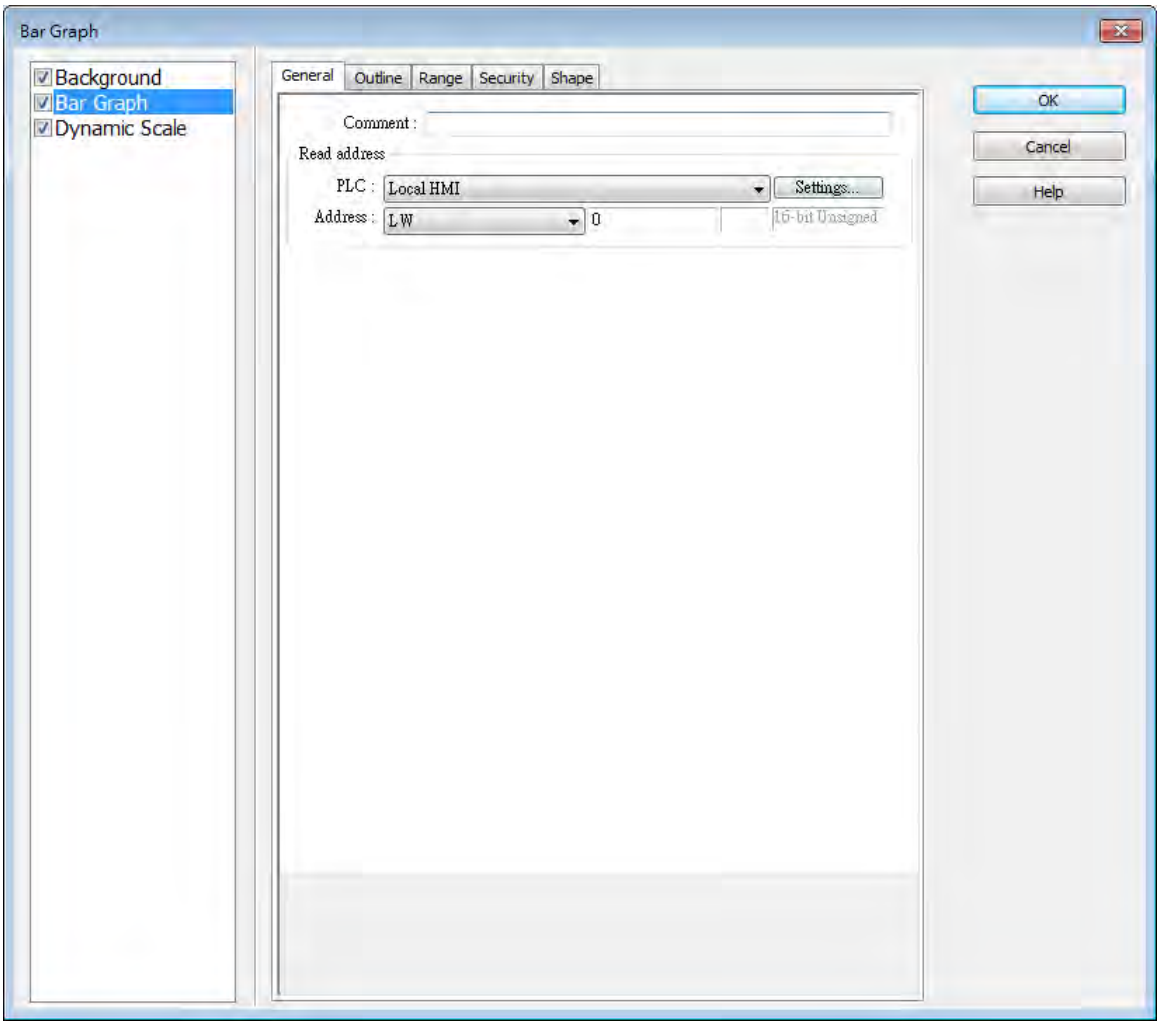
Bar Graph object displays data as a bar graph for visualization.

13.15.2. Configuration

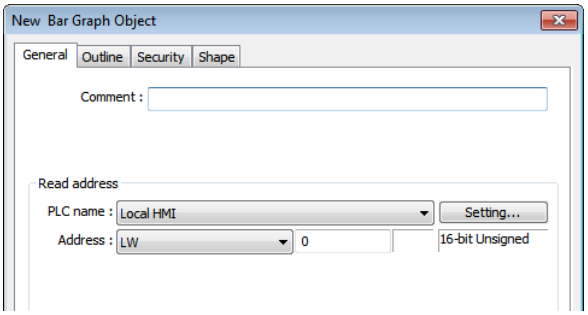
Click [Object] » [Chart] » [Bar Graph] icon on the toolbar to open Bar Graph dialog box. Select properties, click OK button, a new Bar Graph object is created.

General Tab

cMT / cMT X



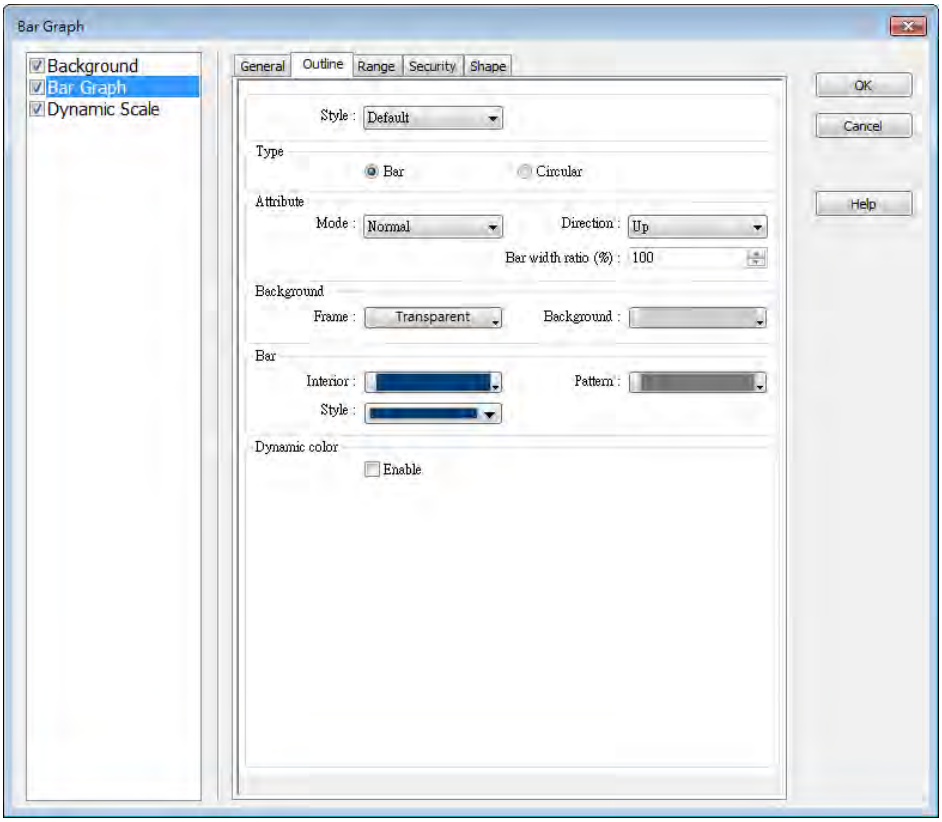
eMT, iE, XE, mTV



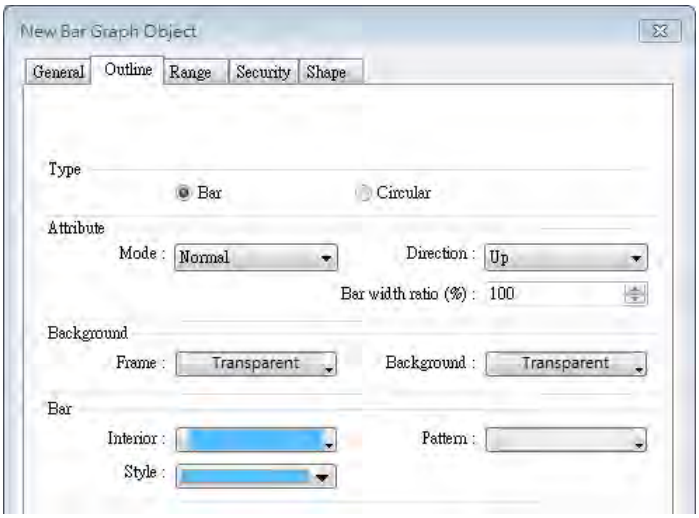
Setting	Description
Read address	Click [Setting] to Select the [Device], [Device type], [Address], [System tag], and [Index register] of the word devices that controls how the bar graph displays.

Outline Tab

cMT / cMT X



eMT, iE, XE, mTV

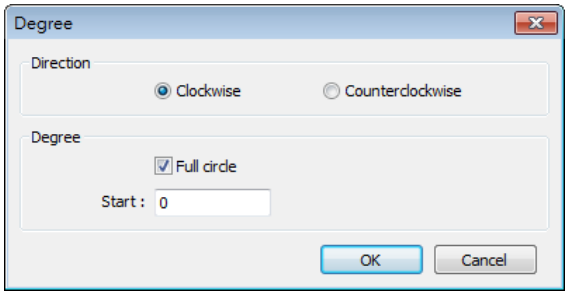


Setting	Description
Type	Choose either [Bar] or [Circular].
Attribute	**Mode** Choose either [Normal] or [Offset]. If [Offset] is selected, an original value [Origin] must be entered for reference.

Direction / Degree

Bar: Determine the bar graph direction. Available options are [Up], [Down], [Right], and [Left].

Circular: Determine the circular bar graph direction. Available options are [Clockwise] and [Counter clockwise].

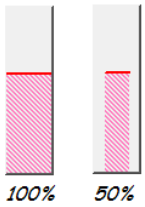


If [Full circle] is selected, set the start degree.

If [Full circle] is not selected, set the start and end degree.

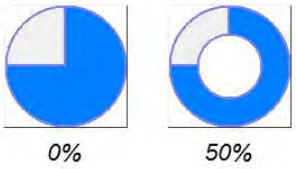
Bar width ratio (%)

The ratio of bar to object width. The figure below shows two ratios, 100% and 50%.



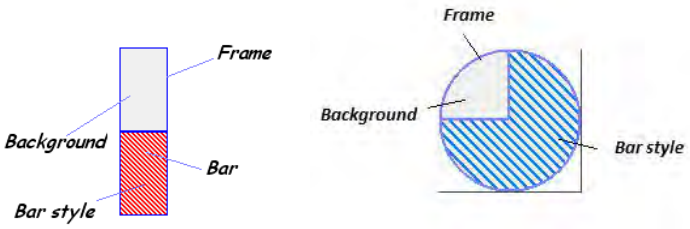
Hole Radius (%)

The ratio of the radius of the hole to the radius of the whole circular bar graph. The figure below shows two ratios, 0% and 50%.

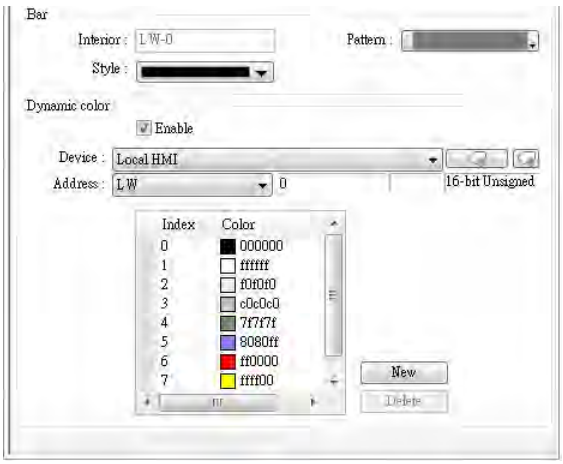


Color/Style

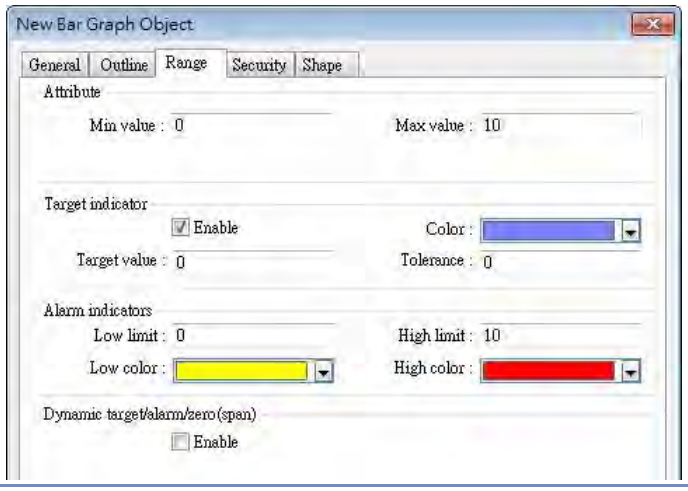
Set the bar's frame and background color, bar style, and bar color. See the picture below.



Dynamic color	(For cMT / cMT X Series models only) With this option selected, the interior color of the bar is determined by the value in the designated address.



Range Tab



Setting	Description
Zero / Span	The percentage of filling can be calculated by the formula, see Example 1.
Target indicator	When the register value meets the condition, the color of filled area will change to the target color, see Example 2.
Alarm indicators	If the register value is larger than [High limit], the color of filled area will change to [High color]. If the register value is smaller than [Low limit], the color will change to [Low color]. Please note that when [Dynamic color] is enabled, the Target indicator and Alarm indicator settings will not be present and their colors are determined by the designated addresses.
Dynamic taget/alarm	When [Enable] is selected, the [Low limit] and [High limit] of [Alarm indicator] and the [Target Value] of [Target indicator] will use

/zero(span)	designated registers, which is shown in their respective fields see Example 3.

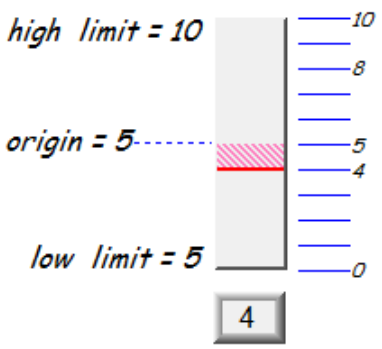
Example 1

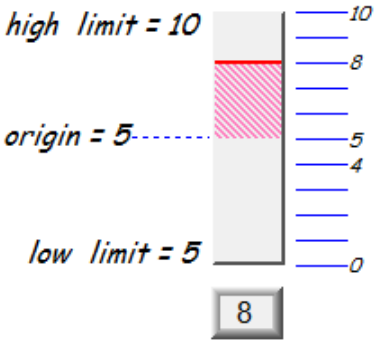
The percentage of filling can be calculated by the following formula:

$$\text{Percentage of filling} = \frac{\text{Register value} - [\text{Zero}]}{[\text{Span}] - [\text{Zero}]} \times 100\%$$

Assume [Offset] is selected. If (Register value – [Zero]) is greater than 0, the bar will fill up from [Origin]. If (Register value – Zero) is less than 0, the bar will be drawn below [Origin].

For example, [Origin] is 5, [Span] is 10, and [Zero] is 0.
For different value in read address, it will display as below:
If the value at read address is 4: If the value at read address is 8:

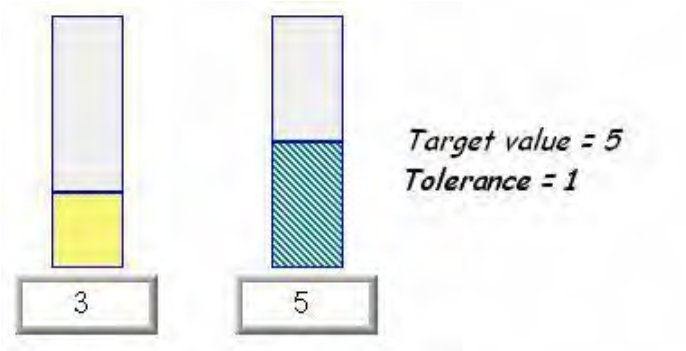




Example 2

When the register value meets the following condition, the color of filled area will change to the target color.
[Target Value] - [Tolerance] ≤ Register value ≤ [Target Value] + [Tolerance]

Assume [Target Value] is 5 and [Tolerance] is 1. As shown below, if the register value is equal to or larger than 4 (=5-1) and equal to or less than 6 (=5+1), the filled area's color of the bar will change to the target color.



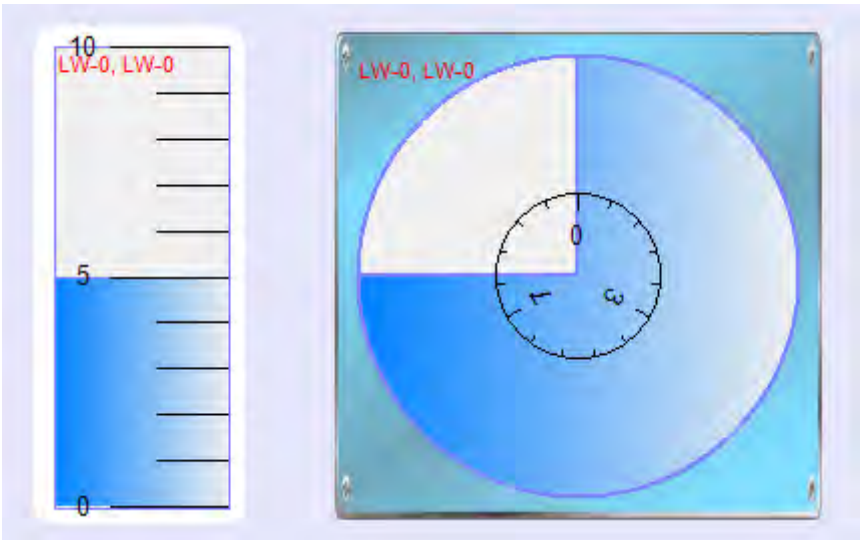
Example 3

If [Dynamic target/alarm] is enabled, [Low limit] and [High limit] of [Alarm indicator] are defined by designated registers as shown in the following table. Furthermore, if [Dynamic zero/span] is used, [Zero], [Span] and [Origin] will be defined by designated registers. Assume the address is LW-n, the limits are:

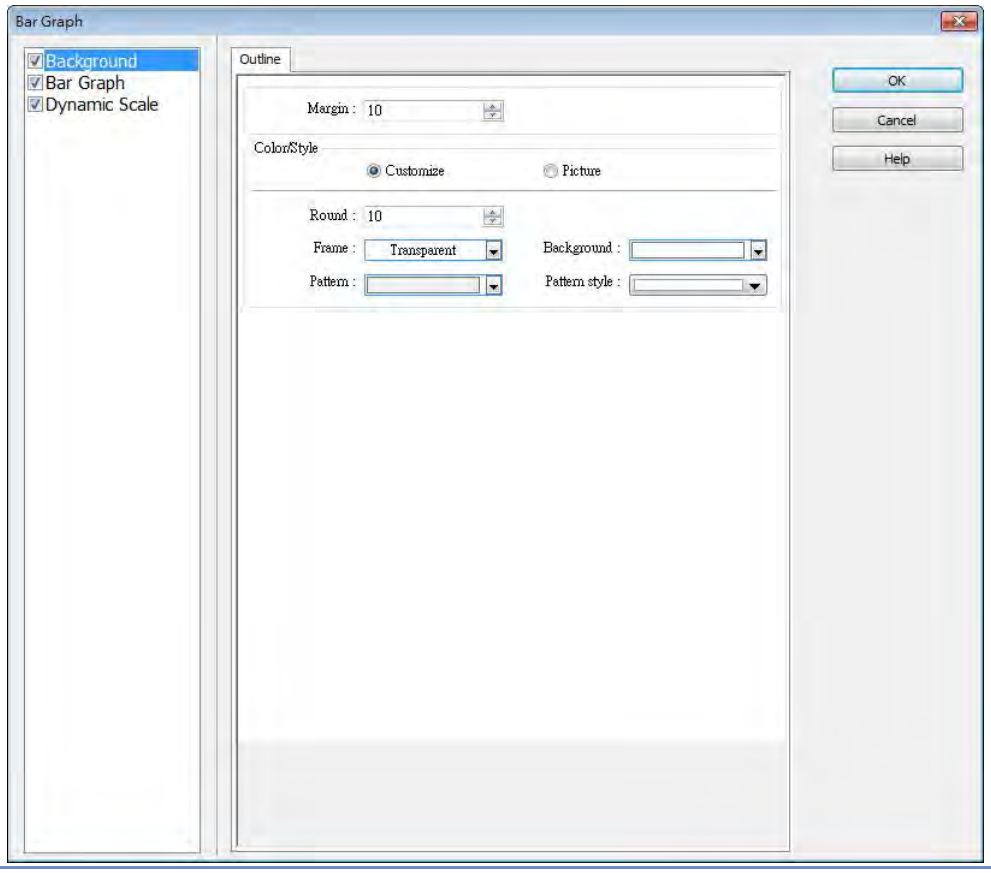
Data format	16-bit	32-bit	64-bit (cMT / cMT X only)
Alarm Low Limit	LW-n	LW-n	LW-n
Alarm High Limit	LW-n+1	LW-n+2	LW-n+4
Target	LW-n+2	LW-n+4	LW-n+8
Zero	LW-n+3	LW-n+6	LW-n+12
Span	LW-n+4	LW-n+8	LW-n+16
Origin	LW-n+5	LW-n+10	LW-n+20

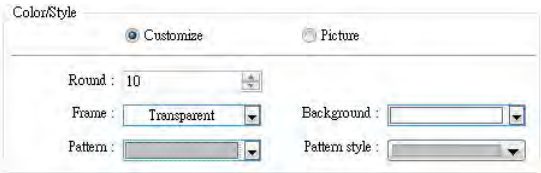
13.15.3. Combo Setting

cMT / cMT X Series HMI support combo setting for Bar Graph, which allows setting of multiple related objects at a time. Bar Graph can be set with Background and Dynamic Scale.

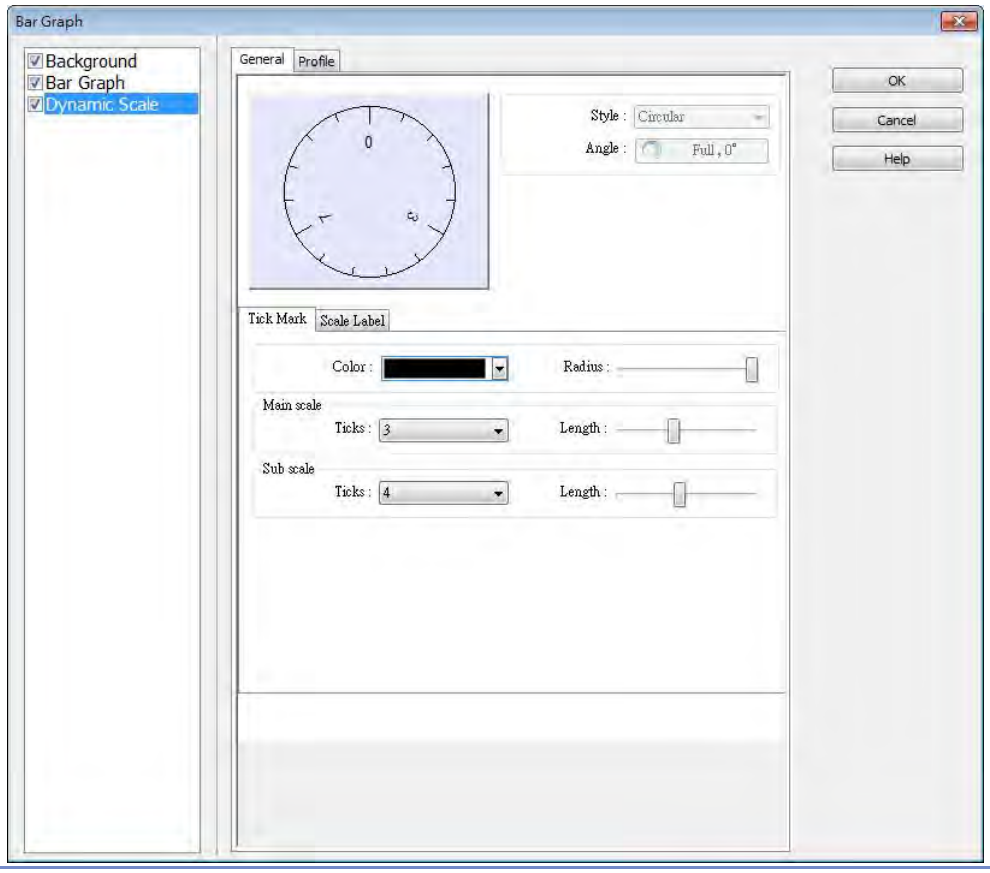


Background



Setting	Description
Margin	Specify the space between the background edge and the objects.
Color/Style	**Customize**  Select a suitable background pattern and color. **Picture** Use the default picture or choose a picture from Picture Library.

Dynamic Scale



Setting	Description
Style	The scale style will follow the bar type.
Tick Mark	Configure the number of tick marks for main and sub scales. If the style is circular, the radius and tick mark length can be specified.
Scale Label	Configure the font, font color, font size and other attributes of scale label.

13.16. Meter Display

13.16.1. Overview

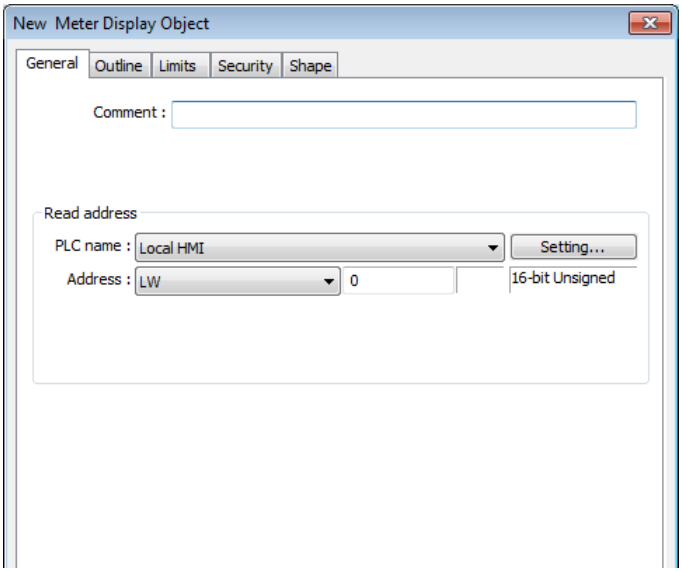
Meter Display object displays the value of word register with a meter.

13.16.2. Configuration

Click [Object] » [Chart] » [Meter Display] icon on the toolbar to open the Meter Display dialog box. Set the object's attributes and then click OK to create a new Meter Display object.

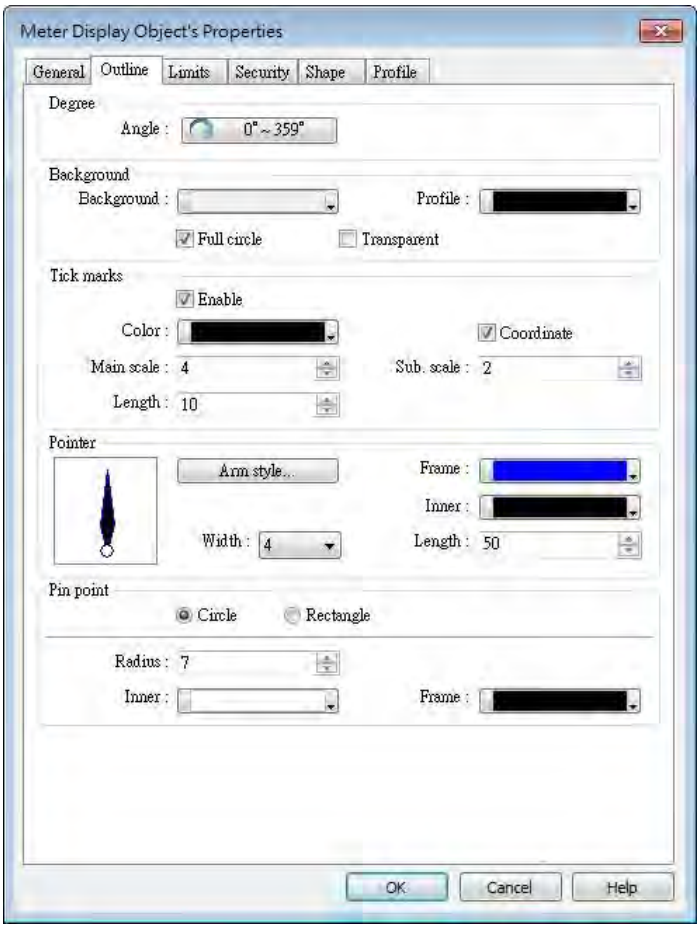
13.16.2.1. eMT, iE, XE, mTV Series

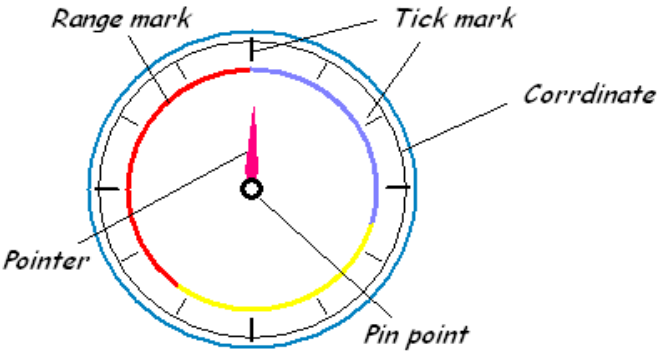
General Tab



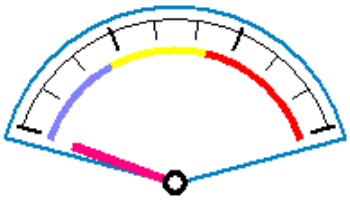
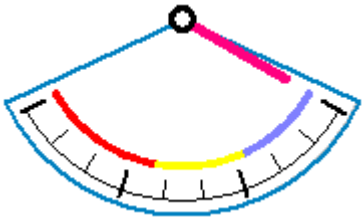
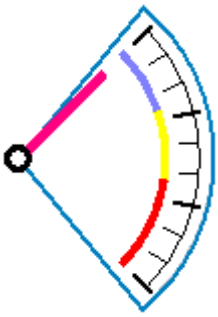
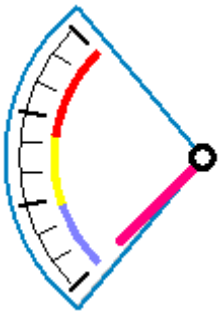
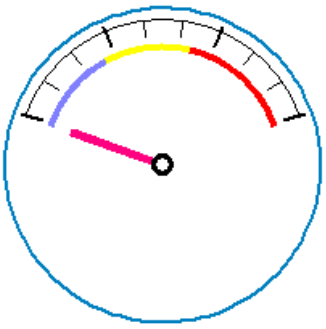
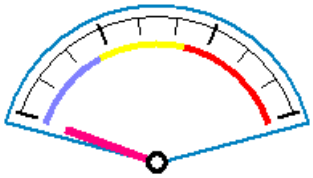
Setting	Description
Read address	Click [Setting] to select the [Device], [Device type], [Address], [System tag], and [Index register] of the word devices that controls the Meter Display object.

Outline Tab

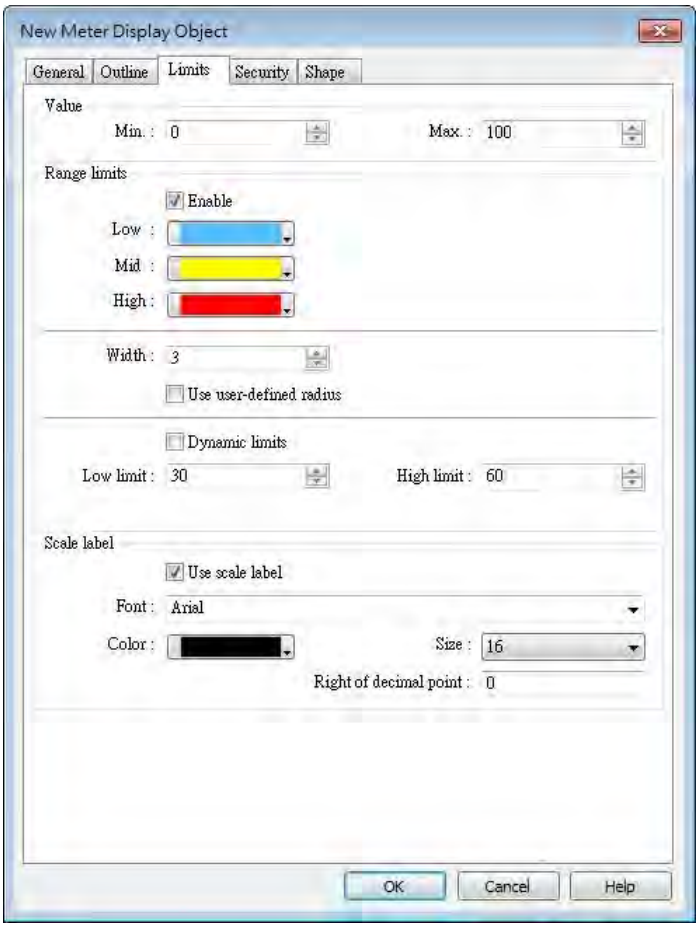


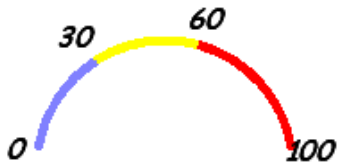


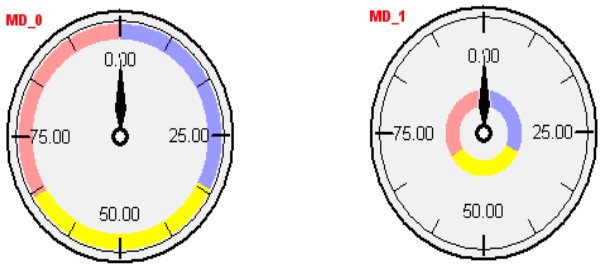
Setting	Description
Degree	Set the pointer to go around the meter clockwise or counterclockwise. Sets the object's start degree and end degree measured clockwise from the 12 o'clock position. The angle range is 0 to 360 degrees. The following shows meters of different settings.

	• [Start degree] =290° [End degree] = 70°  • [Start degree] = 120° [End degree] = 240°  • [Start degree] = 40° [End degree] = 140°  • [Start degree] = 225° [End degree] = 315° 
Background	Sets the object's background color and profile color. **Full circle** When selected, the object will display the whole circle. Otherwise, the object will only display a partial circle in the defined degree range, as shown in the following figure.  *Full circle*  *non-full circle* **Transparent** When selected, the object will not display the background and profile color.
Tick marks	Configures the number of tick mark and color.
Pointer	Configures pointer's style, length, width, and color.
Pin point	Configures the style, radius, and color of the pin point.

Limits Tab



Setting	Description
Value	Sets the object's display range. Meter Display object will use the value of [Zero] and [Span] and the value of register to calculate the pointer's position. See Example 1.
Range limits	Configures the values of [Low limit], [High limit], their corresponding display colors, and the width.  **Use user-defined radius** Configures the radius to display range limits. For example, set to 80: Set to 30:

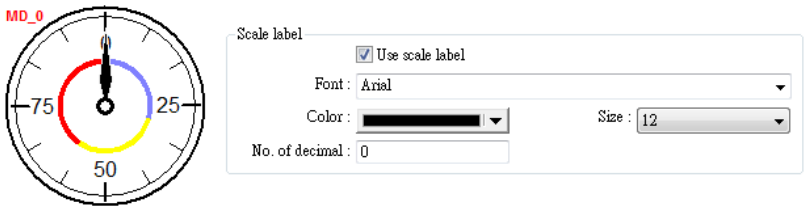


Dynamic Limits

The low limit and high limit are set by registers. See Example 2.

Scale label

Select the attribute of scale label on Meter Display.



Example 1: Pointer position calculation

Set object's display range. Meter Display object will use the value of [Zero] and [Span] and the value of register to calculate the pointer's position. For example, supposed that [Zero] is 0, [Span] is 100, when the value of register is 30, [Start degree] is 0, and [End degree] is 360, then the degree indicated by the pointer is:

{ (30 – [Zero]) / ([Span] – [Zero]) } * ([End degree] - [Start degree]) =

{(30 – 0) / (100 - 0)} * (360 – 0) = 108

Pointer will be pointing at 108 degrees.

Example 2: Dynamic Limits

The low limit and high limit are set by the register.

Suppose the address is LW-n, the following table shows the read address of low limit and high limit:

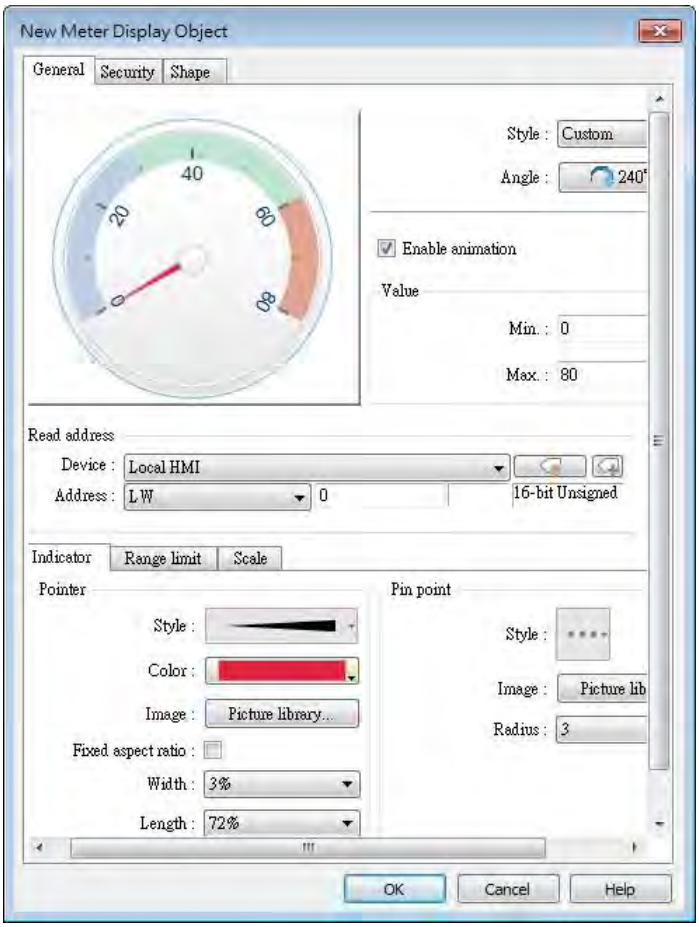
Content	16-bit	32-bit	64-bit (cMT / cMT X only)
Low limit	LW-n	LW-n	LW-n
High limit	LW-n+1	LW-n+2	LW-n+4

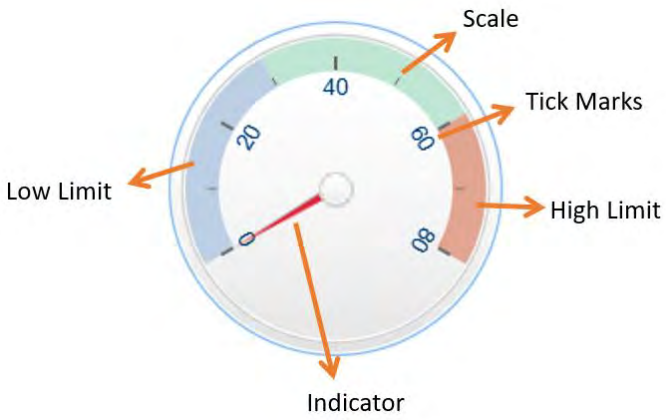
For instance, when address is LW-100, the rule of setting limits is:

Content	16-bit	32-bit	64-bit (cMT / cMT X only)
Low limit	LW-100	LW-100	LW-100
High limit	LW-101	LW-102	LW-104

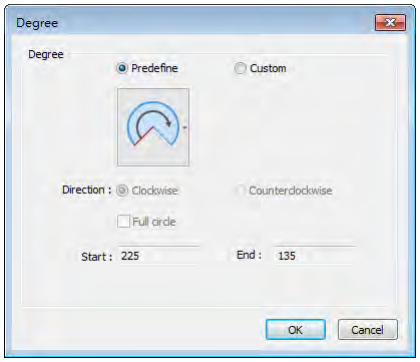
13.16.2.2. cMT / cMT X Series

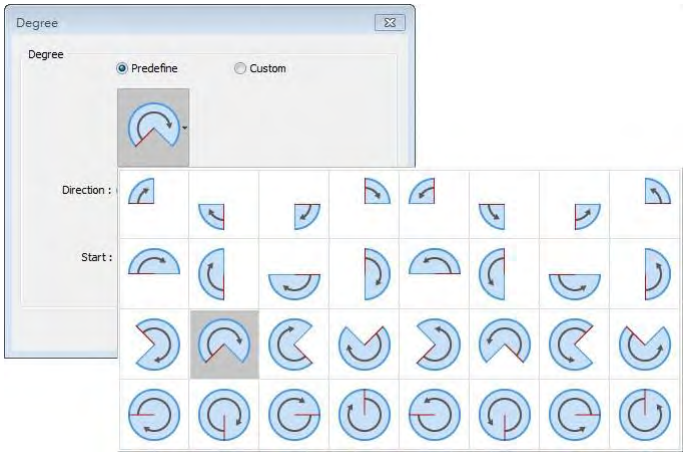
General Tab





Setting	Description
Style	Select a style from the drop-down list. Available options include: [Custom], [Style 1], [Style 2], and [Classical]. If [Custom] is selected, properties such as the width / length of the pointer or the radius of the pin point, etc must also be manually set. The following background-related settings are available using [Style 1], [Style 2], or [Classical]: **Color** Sets the color of the background picture of meter. **Outline** The following is the outline of Style 1 when [Full circle], [Half circle], or [Quarter circle] is selected. The following is the outline of Style 2 when [Full circle], [Half circle], or [Quarter circle] is selected. **Rotation** Rotates the background picture clockwise according to the angles set.
Angle	**Direction** Sets the range to label the scale, using twelve o'clock direction as the 0° reference. **Degree** Available options include: [Predefined] and [Custom]. With [Predefine] selected, pick from the thumbnails configure directions and angle ranges with ease.





	Full circle If selected, the full circle is drawn according to the selected direction and the start angle. The limits are determined by the value set in [Minimum] and [Maximum] field under [Value].
Enable animation	If selected, the pointer slides to the designated position when the read value changes; if not selected, the pointer directly points to the designated position when the read value changes.
Value	Sets the lower and upper limits of the meter.
Read address	Displays the value in meter according to the value in the designated word register.
Indicator	Sets the style of pointer and pin point. If [Custom] is selected, the direction of the pointer must points upward to correctly display.
Range limit	Sets the colors to indicate different ranges. **Dynamic limits** The low limit and high limit are decided by the register. See Example 2 in the previous section.
Scale	Sets the number of main and sub scale, the color of tick marks and scale label.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.17. Trend Display

13.17.1. Overview

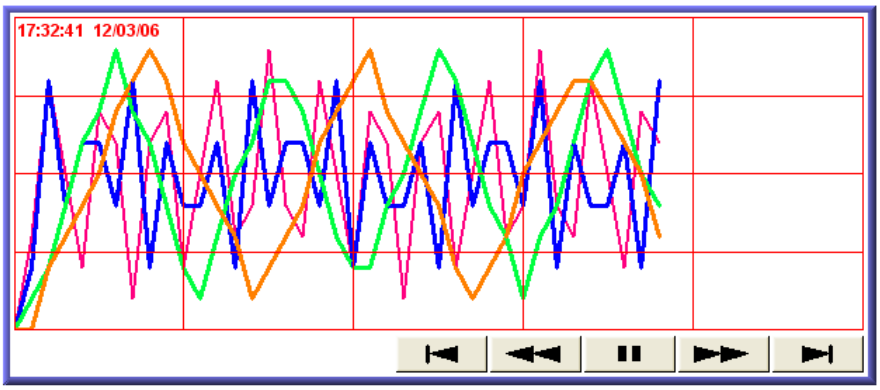
Trend display objects draw curves of the data recorded by Data Sampling object.

13.17.2. Configuration

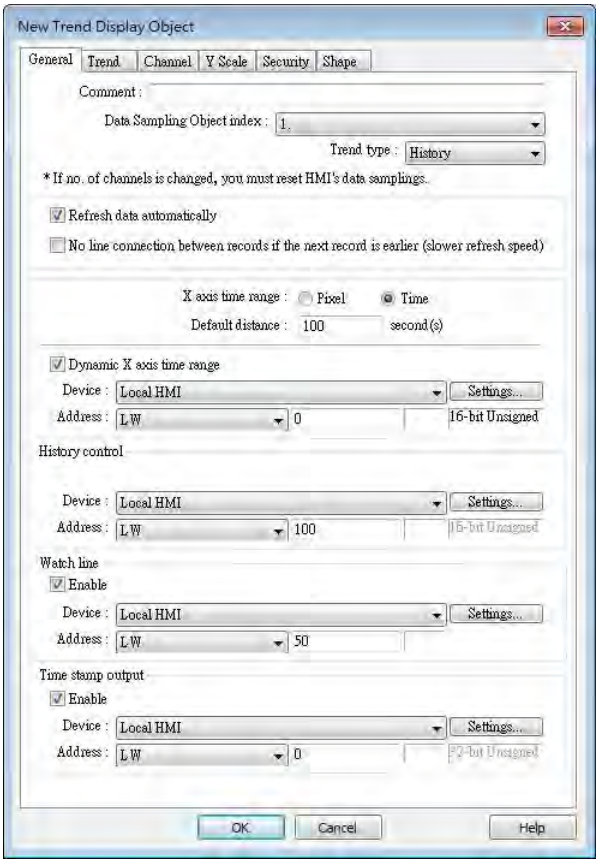
Click [Data/History] » [Trend Display] icon on the toolbar to open a Trend Display object property dialog box. Set up the properties, press OK button, and a new Trend Display object will be created.

13.17.2.1. eMT, iE, XE, mTV Series

General Tab



Button	Description
⏮	Go to the earliest sampling data.
◀◀	Go to the previous time interval.
⏸	Click to stop auto-scrolling. When the new sampling data is generated, the display does not scroll, nor is the new data outside the display range displayed.
▶	Click to start auto-scrolling. The display scrolls as the new sampling data is generated.
▶▶	Go to the next time interval.
⏭	Go to the latest sampling data.



Setting	Description
Data Sampling Object index	Select a Data Sampling object as the source data.
Refresh data automatically	In history mode, when this option is selected, Trend Display will be automatically refreshed every 10 seconds. If this option is not selected, Trend Display can only be refreshed by changing window.
No line connection between records if the next record is earlier	When HMI time is adjusted to an earlier time, and data sampling keeps going on, selecting this option can prevent the system from drawing a line to connect the gap between current trend curve (earlier in time axis) and former trend curve (later in time axis). This can slow down refresh speed.
Trend type	Select the mode of data source, either [Real-time] or [History]. **Real-time** In this mode, the display object shows all sampled data since the HMI started. The maximum number of records that can be sampled is set in [Max.data records] (Real-time mode) of the Data Sampling object. When the sampling data exceed this setting, the earlier data will be deleted. To show older data, use [History] mode. [Hold control]: Suspends the update of Trend Display. However, it

does not stop the sampling process of Data Sampling object.

History

In this mode, the data comes from the history data files stored on HMI. . The history data files are sorted by dates, and each is given an index. The system uses [History control] to select the history data files that are created on different dates.

The system sorts the history data of sampling data by date; the latest file is record 0 (typically the data sampled today), the second latest file is record 1, and so on. If the value of designated register in [History control] is n, the Trend Display object will display data record n.

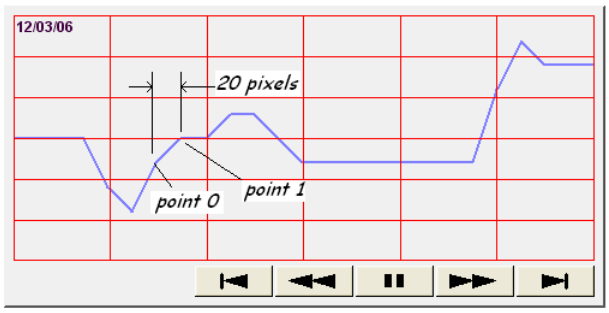
Here is an example to explain [History control]. If the designated register is LW-0, and the sampling data files available are pressure_20061120.dtl, pressure_20061123.dtl, pressure_20061127.dtl, and pressure_20061203.dtl, and it is 2006/12/3 today, based on the value of LW-0, the sampling data file which will be selected by [Trend Display] is shown as follows:

Value of LW-0	Selected sampling history data
0	pressure_20061203.dtl
1	pressure_20061127.dtl
2	pressure_20061123.dtl
3	pressure_20061120.dtl

If use with Option List object and select data source as [Dates of historical data], the history data will be sorted by date and displayed in Option List object, see “13.29 Option List”.

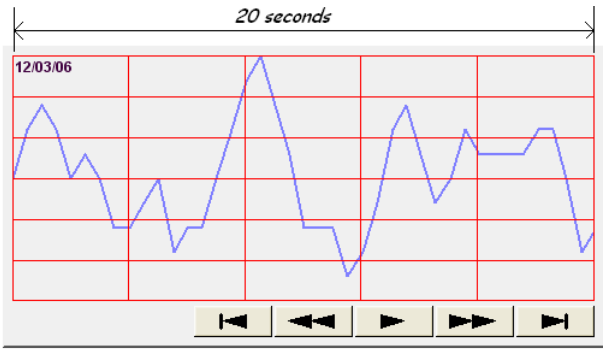
Pixel

[Distance] is used to set the distance between two sampling points, as shown in the following figure.



Time

[Distance] is used to set the X-axis in unit of time, as shown in the following figure.



Select [Time] for [X axis time range] and go to [Trend] » [Grid] and enable [Time scale]. Please refer to [Time scale] in the later section.

Dynamic distance between data samples/ Dynamic X axis time range

Designate a 32-bit word register for adjusting the distance between two sampling points (select Pixel), or the time unit represented by X-axis (select Time). If no value is entered, the default value will be used.

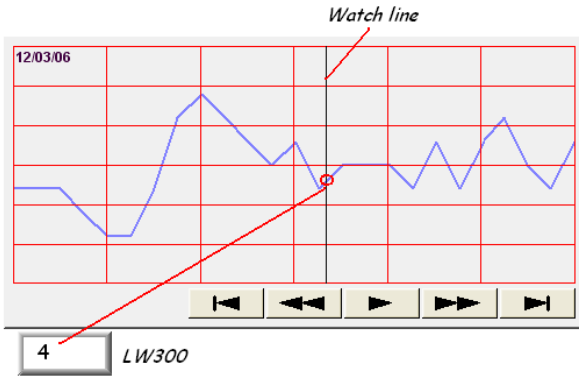
Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Refresh data automatically

If selected, every time when opening the window on which the Trend Display object in history mode is placed, the display is automatically refreshed every 10 seconds. Please note that:

- The refresh status can be observed from the control button of Trend Display object.
  Showing button: The automatic refresh feature is enabled.
  Showing button: The automatic refresh feature is disabled.
- When scrolling to the previous data, the automatic refresh feature is disabled, the button is shown.
- If [Refresh data automatically] check box is selected, when change back to the window, the display is refreshed, ignoring the control buttons. For example, select [Refresh data automatically], and scroll to the previous data, the automatic refresh feature is disabled. In this case, changing to another window and then change back will still refresh the display.
- If [Refresh data automatically] check box is not selected when building the project, the feature can still be enabled by pressing button on HMI. In this case, the automatic refresh feature is disabled, that is, even when change back to the current window, the display will not be refreshed.

Hold control	When the register is set ON, suspend the update of Trend Display. It does not stop the sampling process of Data Sampling object. This setting is available only in Real-time mode.
Watch line	Use the [Watch line] function to display a "watch line" when user touches the Trend Display object. It will also export the sampling data at the position of watch line to the designated word device and use Numeric objects to display the results, as shown in the following figure.



[Watch line] can also export sampling data with multiple channels. The system will consecutively write each channel to the specified address and the following addresses, in the same order as in [Data Sampling] object. The address assigned to [Watch line] is the start address, and sampling data for each channel will be exported to the word devices starting from "start address." If the data format of each channel is different, the corresponding address of each channel is arranged from the first to the last. If the watch register is LW-300, watch function will export each channel's data to the following addresses:

Register	Channel	Data format
LW-300	0	16-bit Unsigned (1 word)
LW-301	1	32-bit Unsigned (2 words)
LW-303	2	32-bit float (2 words)
LW-305	3	16-bit Signed (1 word)

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Time stamp output	Time stamp output Suppose the address is set to LW-n, then: If enabled, the system will use the time of the first sampling data as "time origin", and write the time stamp of the most recent sampled

	data (relative to “time origin”) to [LW-n+2]. When clicking on the curve, the time stamp of the closest sampled point will be written to [LW-n]. Clear real-time data address (Data Sampling object) will clear the time origin as well. Time stamp is recorded in seconds.

Trend Tab

Setting	Description
Transparent / Frame / Background	Select the color of frame and background.
Show scroll controls	Enable or disable the scroll control as shown in the following figure.
Grid	Set the number of dividing lines and the line color. The number of divisions depends on the setting in General tab » [Distance

between data samples] / [X axis time range].

X-axis interval

The number of vertical grid lines.

- Select [Distance between data samples] in General tab:

Select how many sampling points will be included between two vertical grid lines.

- Select [X axis time range] in General tab:

Select the time range between two vertical grid lines.

X-axis division

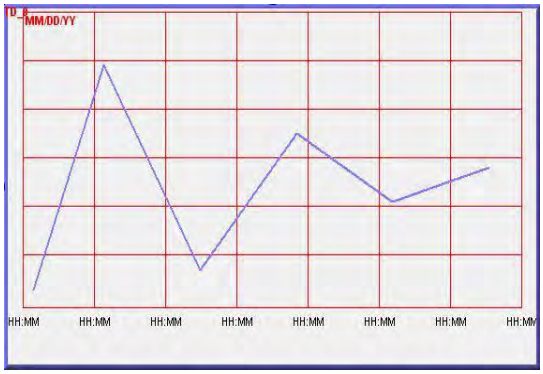
The number of vertical grid lines.

Y-axis division

The number of horizontal grid lines.

Time scale

Select [Time] / [Date] check box to display the time scale along the x axis.



Font / Color / Size

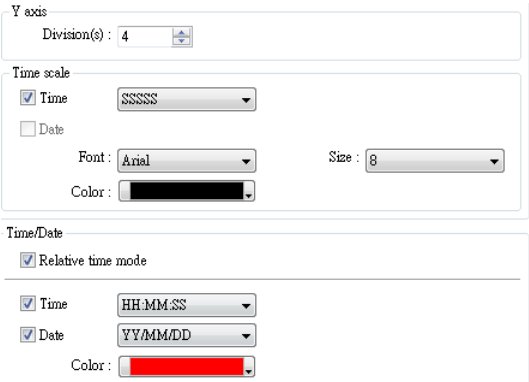
Select the font, font color, and font size of the time scale.

The default font size is 8.

Time / Date

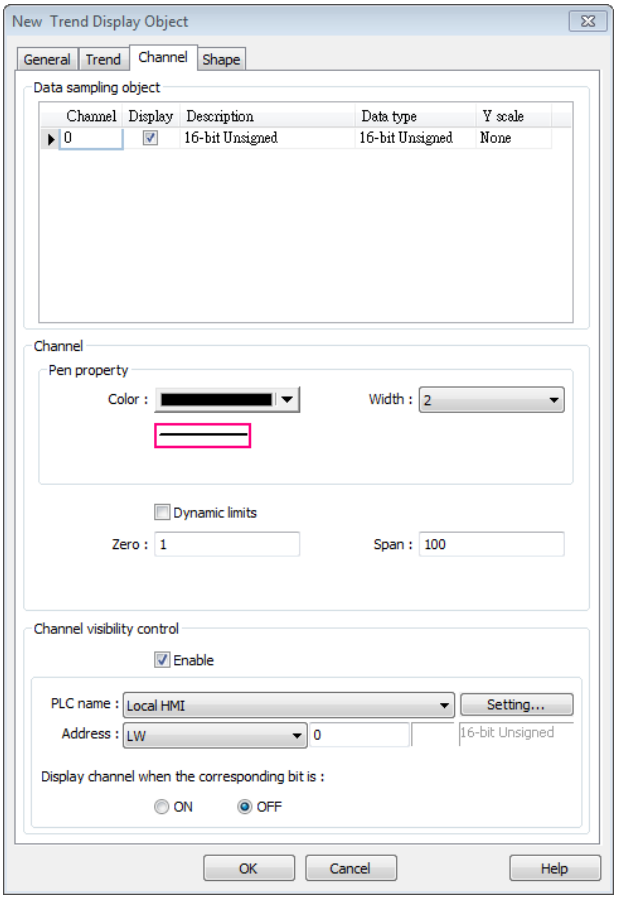
Relative time mode

When this mode is selected, the time of the earliest sampling data will be the start time from which to count a relative time. This mode works with time scale [SSSSSS] and under this mode, time scale [Date] cannot be used.



	The time of the latest sampling data will be marked on the top left corner of the object. This group box is used to set the time / date display format and font color.

Channel Tab



Setting	Description
Y scale	Set Y-axis to be Main Axis or Aux. Axis. See “Y Scale Tab” for more information.
Channel	Configure each sampling line’s format and color. At most 64 channels could be configured. **Dynamic limits** ● Not selected: [Zero] and [Span] are used to set the low limit and high limit of sampling data. If the low limit is 50 and the high limit is 100 for one sampling line, [Zero] and [Span] must be set as [50] and [100], so that all the sampling data can be displayed in the trend display object.

- Selected

The low limit and the high limit are read from the designated word devices, as shown below. When address is LW-n, the register's address:

Data Format	16-bit	32-bit	64-bit (cMT / cMT X only)
Low limit	LW-n	LW-n	LW-n
High Limit	LW-n+1	LW-n+2	LW-n+4

For example, if LW-100 is used here, the low limit and the high limit will be read from:

Data Format	16-bit	32-bit	64-bit (cMT / cMT X only)
Low limit	LW-100	LW-100	LW-100
High Limit	LW-101	LW-102	LW-104

A typical usage of this is to zoom in and zoom out of Trend Display. (Not available for cMT / cMT X Series). See Example 1.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Channel visibility control

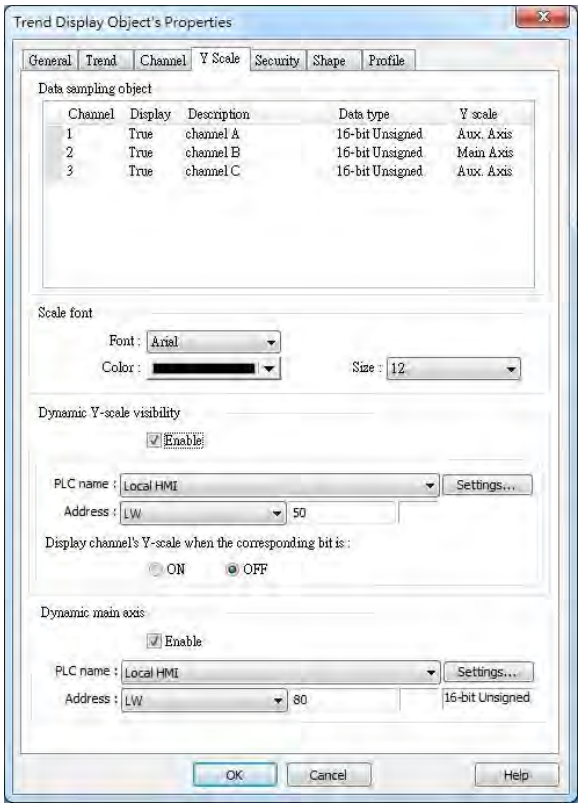
If [Enable] is selected, the bits of the assigned word register will be used to show/hide each channel. The first bit controls the first channel, and the second bit controls the second channel, and so on. For example, suppose there are 5 channels and LW-0 is used, channels which will be shown given the states of the control bits are:

Channel	Control Bit	State	Displayed
1	LW_bit-000	OFF	YES
2	LW_bit-001	ON	NO
3	LW_bit-002	ON	NO
4	LW_bit-003	OFF	YES
5	LW_bit-004	OFF	YES

Note on using this feature: Each control bits are not reserved for the channel. If a particular channel is not displayed, the control bit is assigned to the next displayed channel. For example, if the third channel of the 5 channels is not displayed, only 4 channels will be displayed in Trend Display, and the used control bits will only be: LW_bit-000~003.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Y Scale Tab



Setting	Description
Y scale	Show whether Y-axis is Main Axis or Aux. Axis. Y-axis will not be displayed when [none] is selected for Y scale in Data Sampling Object group box. At most 32 Y axes can be displayed, including one main axis and multiple aux. axes.
Scale font	Select the font, font color, and font size of the scale.
Dynamic Y-scale visibility	To show or hide Y-scale. If the control address is LW-50, then the first axis is controlled by LW_Bit 5000, and the second axis is controlled by LW_Bit 5001, and so on.
Dynamic main axis	To change the main axis. If writing 1 into LW-80, the main axis will be Channel 1; if writing 2 into LW-80, the main axis will be Channel, and so on.

Note

- A Trend Display on an eMT/iE/XE/mTV model can display 6MB of data in maximum.

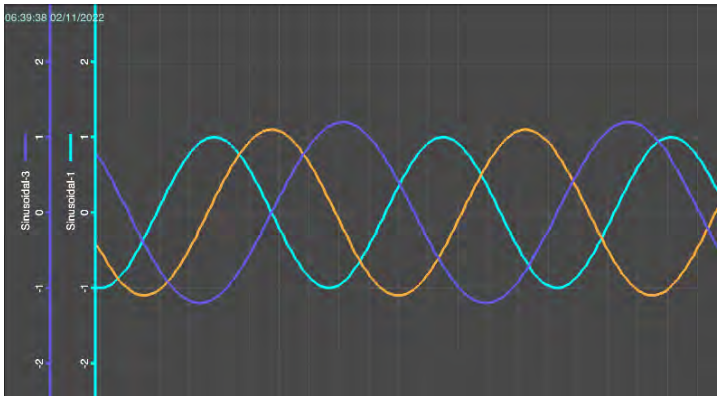
Click the icon to watch the demonstration film. Please confirm your internet connection

before playing the film.

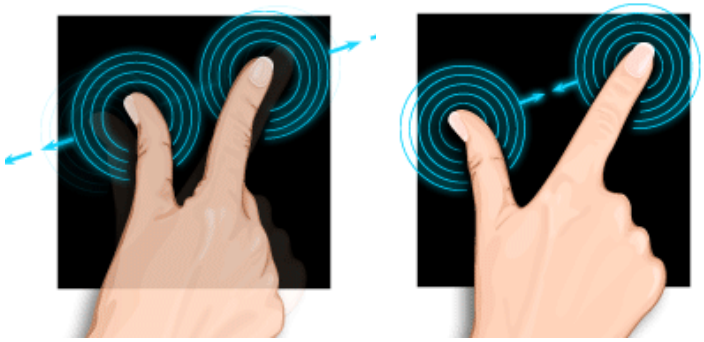
Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.17.2.2. cMT, cMT X Series

General Tab

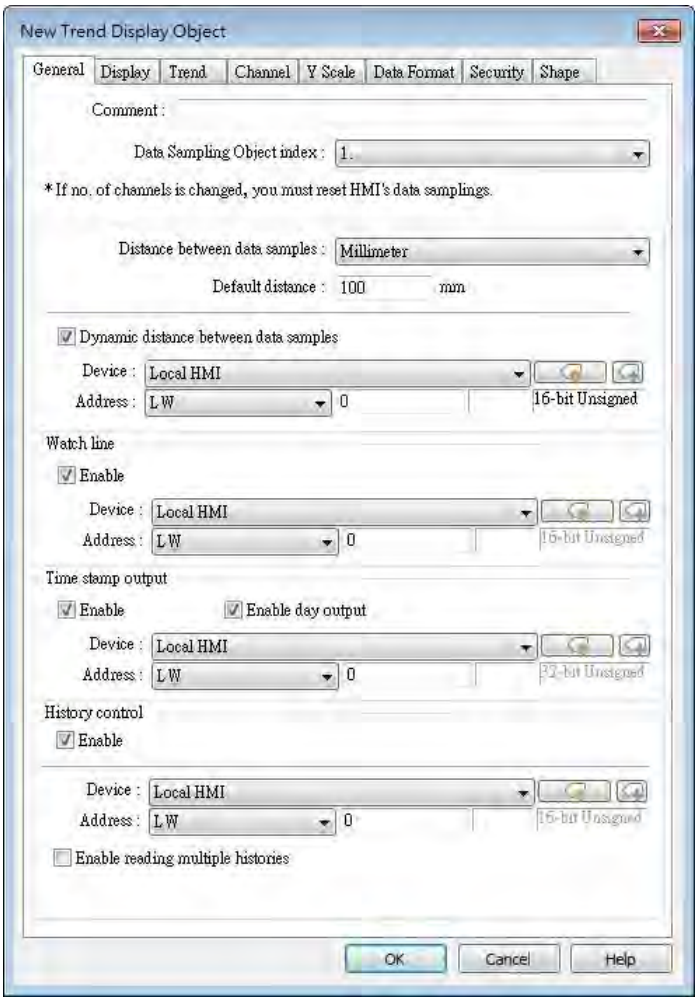


The Trend Display on cMT / cMT X Series combines Real-time mode and History mode. Drag left to scroll the Trend Display to view history data and drag right to view the latest sampling data. Pinch two fingers together to zoom out Trend Display or spread them apart to zoom in.



Zoom In Zoom Out

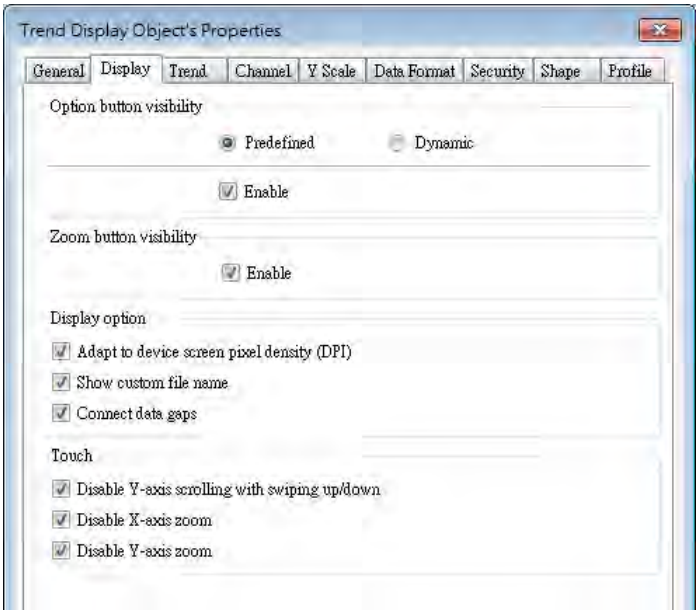
For more information about how sampling data is saved, see “8 Data Sampling”.



Setting	Description
Data Sampling Object index	Select a [Data Sampling] object as the source data.
Millimeter	Same as eMT, iE, XE mTV Series. The upper limit is 500 mm.
Time	Same as eMT, iE, XE mTV Series.
Pixel	Same as eMT, iE, XE mTV Series. The upper limit is 1920 pixels.
Dynamic distance between data samples	Same as eMT, iE, XE mTV Series.
Watch line	Same as eMT, iE, XE mTV Series.
Time stamp output	When enabled, the number of seconds passed from the first sampling data can be calculated. Suppose the address is set to LW-n, then: [LW-n]: The number of seconds passed from the first sampling data to the closest sampled point. [LW-n+2]: The number of seconds passed from the first sampling

	data to the most recent sampling data. **Enable day output** With this option selected, the number of days passed since 1970/1/1 can be calculated. Suppose the address is set to LW-n, then: [LW-n+4]: The number of days passed since 1970/1/1 to the closest sampled point. [LW-n+6]: The number of days passed since 1970/1/1 to the most recent sampling data.
History Control	Same as eMT, iE, XE mTV Series. **Enable reading multiple histories:** In the following two modes, days are calculated backward. **Number of days** Suppose the address is set to LW-n, then: [LW-n]: The start day index from which to calculate number of days. 0: today, 1: yesterday, and so on. [LW-n+2]: The total number of days before the start day. **Specific days** Suppose the address is set to LW-n, then: [LW-n]: The start day index from which to calculate number of days. 0: today, 1: yesterday, and so on. [LW-n+2]: The end day index, the specified value must be greater than that of the start day index. For example, if start is 4 and end is 7, the specified days will be 4 days before to 7 days before.

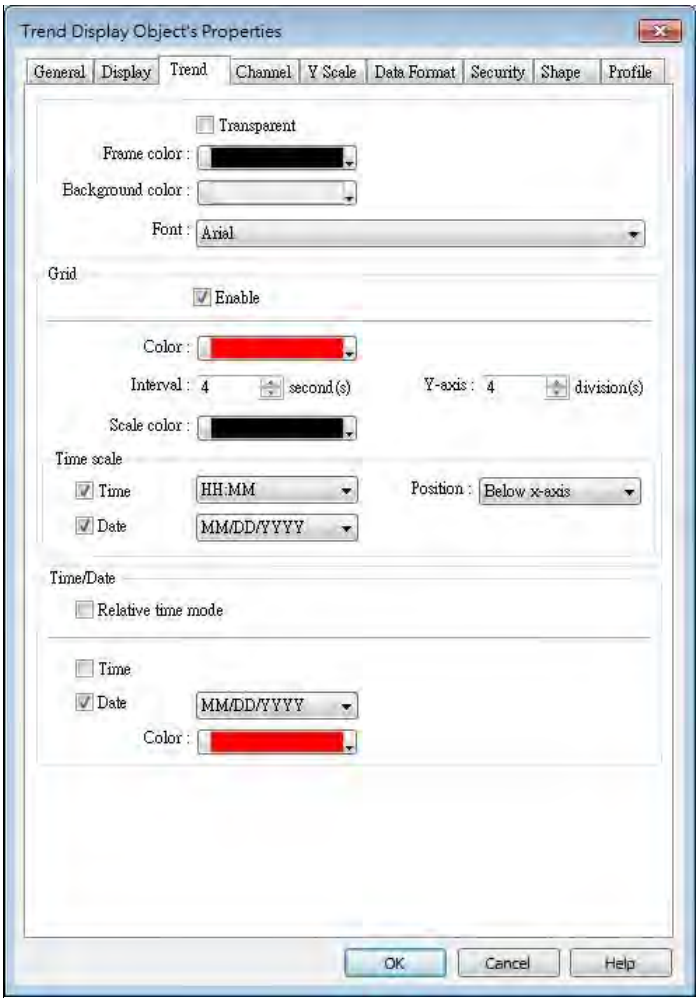
Display Tab



Setting	Description
Option button visibility (cMT Series)	Show or hide the option button of the object. **Predefined** Select [Enable] to show the option button. **Dynamic** Choose a bit address to control option button visibility.
Zoom button visibility	Show or hide the zoom button of the object.
Display option	**Adapt to device screen pixel density (DPI)** With this option selected, text size, position, and the spacing between Y axes will be adjusted according to the DPI of the screen of the connected device. **Show custom file name** This option appears when [Customized file handling] is selected in Data Sampling. With this option selected, the customized file name is shown as caption in Trend Display. **Connect data gaps**

	Data sampling stops when PLC is disconnected, and after PLC is reconnected, the trend curve of the disconnected period is plotted with a gap. With this option selected, the system will draw a line to connect the gap between the former trend curve (before disconnected period) and the current trend curve (after disconnected period).
Touch	**Disable Y-axis scrolling with swiping up/down** With this option selected, dragging the object with your finger will only scroll X-axis range while Y-axis remains being placed in the middle of the object even if it is zoomed in/out. **Disable X axis zoom** With this option selected, dragging the object with your finger will only scroll Y-axis range. **Disable Y axis zoom** With this option selected, dragging the object with your finger will only scroll X-axis range.

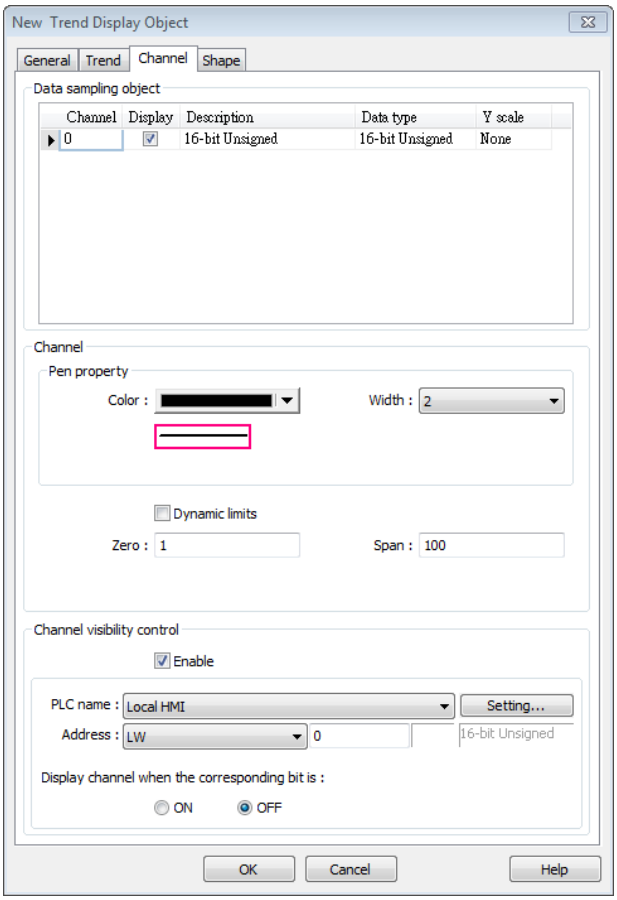
Trend Tab



Setting	Description
Transparent / Frame / Background	Select the color of frame and background.
Grid	Set the number of dividing lines and the line color. The number of divisions depends on the setting in General tab » [Distance between data samples] / [X axis time range]. **X-axis interval** The number of vertical grid lines. ● Select [Distance between data samples] in General tab: Select how many sampling points will be included between two vertical grid lines. ● Select [X axis time range] in General tab: Select the time range between two vertical grid lines. **X-axis division**

	The number of vertical grid lines. **Y-axis division** The number of horizontal grid lines.
Time scale	Select [Time] / [Date] check box to display the time scale along the x axis. The position of the time scale can be selected. The default font size is 8.
Time / Date	**Relative time mode** When this mode is selected, the time of the earliest sampling data will be the start time from which to count a relative time. This mode works with time scale [SSSSSS] and under this mode, time scale [Date] cannot be used. The time of latest sampling data will be marked on the top left corner of the object. This group box is used to set the time / date display format and font color.

Channel Tab



Setting	Description
Y scale	Set Y-axis to be Main Axis or Aux. Axis. See “Y Scale Tab” for more information. At most 32 Y axes can be displayed, including one

main axis and multiple aux. axes.

Channel	Display	Description	Data type	Y scale
1	True	channel A	16-bit Unsigned	Aux. Axis
2	True	channel B	16-bit Unsigned	Main Axis
3	True	channel C	16-bit Unsigned	Aux. Axis

Channel

Configure each sampling line's format and color. At most 64 channels could be configured.

Dynamic limits

- Not selected:

[Zero] and [Span] are used to set the low limit and high limit of sampling data. If the low limit is 50 and the high limit is 100 for one sampling line, [Zero] and [Span] must be set as [50] and [100], so that all the sampling data can be displayed in the trend display object.

- Selected

The low limit and the high limit are read from the designated word devices, as shown below. When address is LW-n, the register's address:

Data Format	16-bit	32-bit	64-bit (cMT / cMT X only)
Low limit	LW-n	LW-n	LW-n
High Limit	LW-n+1	LW-n+2	LW-n+4

For example, if LW-100 is used here, the low limit and the high limit will be read from:

Data Format	16-bit	32-bit	64-bit (cMT / cMT X only)
Low limit	LW-100	LW-100	LW-100
High Limit	LW-101	LW-102	LW-104

Channel visibility control

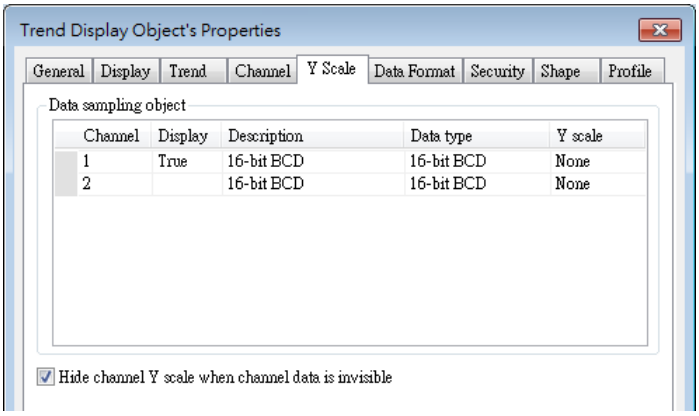
If [Enable] is selected, the bits of the assigned word register will be used to show/hide each channel. The first bit controls the first channel, and the second bit controls the second channel, and so on. For example, suppose there are 5 channels and LW-0 is used, channels which will be shown given the states of the control bits are:

Channel	Control Bit	State	Displayed
1	LW_bit-000	OFF	YES
2	LW_bit-001	ON	NO
3	LW_bit-002	ON	NO

4	LW_bit-003	OFF	YES
5	LW_bit-004	OFF	YES

Note on using this feature: Each control bits are not reserved for the channel. If a particular channel is not displayed, the control bit is assigned to the next displayed channel. For example, if the third channel of the 5 channels is not displayed, only 4 channels will be displayed in Trend Display, and the used control bits will only be: LW_bit-000~003.

Y Scale Tab

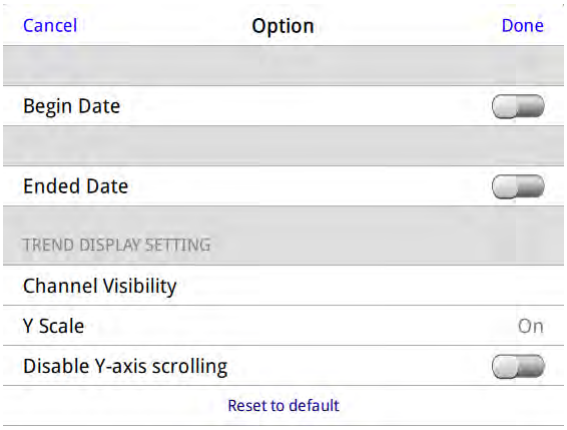


The scale along the Y axis of a specific channel can be displayed. To enable Y Scale, [Grid] should first be enabled in [Trend] tab.

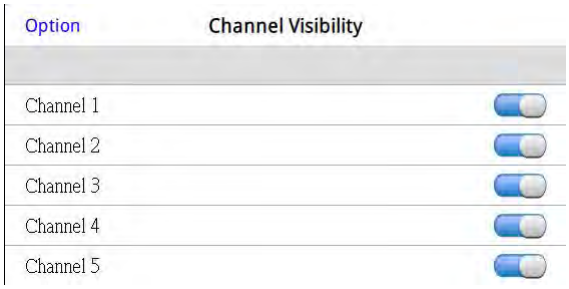
[Hide channel Y scale when channel data is invisible] option is selected by default. With this option selected, whether channel Y scale is shown or hidden is determined by the settings in Channel Visibility Control.

Y Scale can be configured on the in cMT Viewer as shown in the following steps.

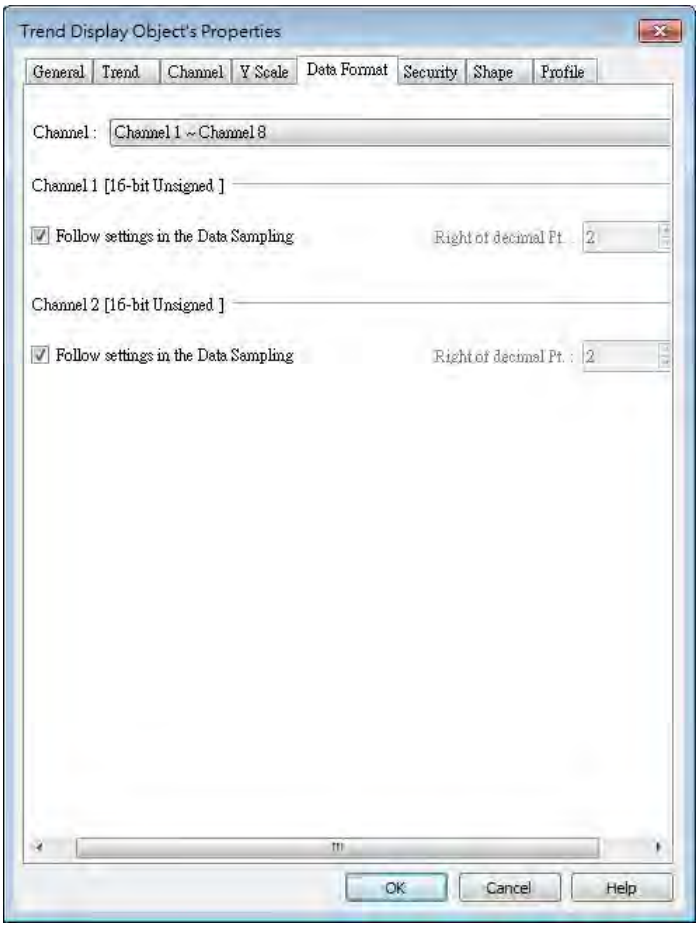
1. Tap the button on the upper right corner of Trend Display object.
2. Tap [Trend Display Setting] » [Y Scale].



3. Set channel visibility.



Data Format Tab



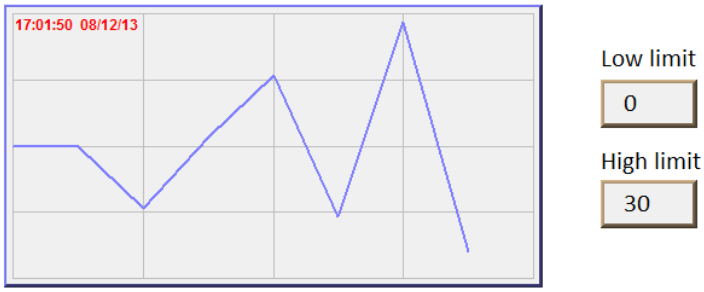
Setting	Description
Follow settings in the Data Sampling	Use the [Right of decimal Pt.] setting in Data Sampling.
Right of decimal Pt	The number in this field determines the number of decimal places of the value displayed; for example, when 1 is entered in this field, and the original value obtained by Data Sampling is 45, then 4.5 will be displayed in Trend Display as a result.

Example 1

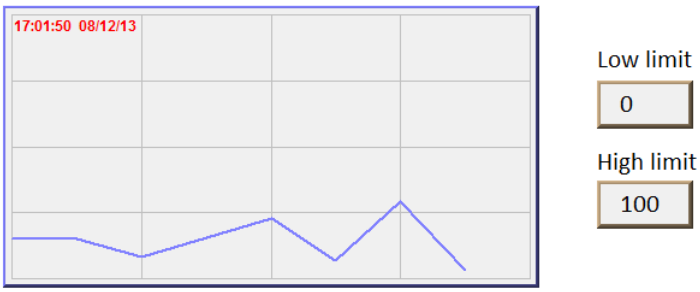
The example explains how to zoom in or zoom out Trend Display. The feature described is not available for cMT / cMT X Series.

In Channel tab select [Dynamic limits] check box. If the [Address] is set to LW-n, then LW-n controls the low limit where LW-n+1 controls the high limit.

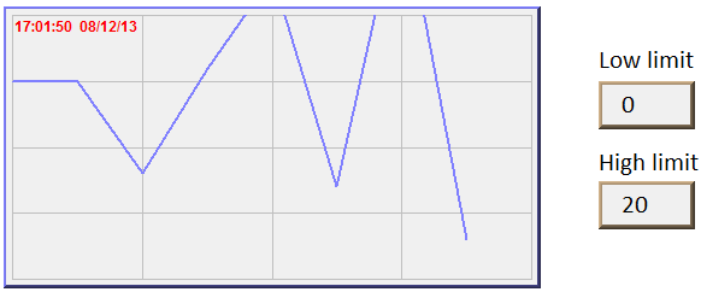
Set [Address] to LW-0 and create two Numeric objects for entering the low / high limit. The address that controls the low limit is LW-0; the address that controls the high limit is LW-1. Let's suppose the data is between 0 and 30; set the [Low limit] to 0 and the [High limit] to 30, the trend curve is displayed as shown in the following figure.



To zoom out the Trend Display, enter a value greater than 30 in [High limit] as shown in the following figure.



To zoom in the Trend Display, enter a value less than 30 in [High limit] as shown in the following figure.



Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.18. History Data Display

13.18.1. Overview

History Data Display object displays data stored by Data Sampling object. It differs from Trend Display in that History Data Display object uses a table to display data. The following is an example of a history data display object.

No.	Time	Date	Ch.0	Ch.1	Ch.2
3577	21:52	16/09/07	0	0	0
3576	21:52	16/09/07	0	0	0
3575	21:52	16/09/07	0	0	0
3574	21:52	16/09/07	0	0	0
3573	21:52	16/09/07	0	0	0
3572	21:52	16/09/07	0	0	0
3571	21:52	16/09/07	0	0	0
3570	21:52	16/09/07	0	0	0
3569	21:52	16/09/07	0	0	0
3568	21:52	16/09/07	0	0	0

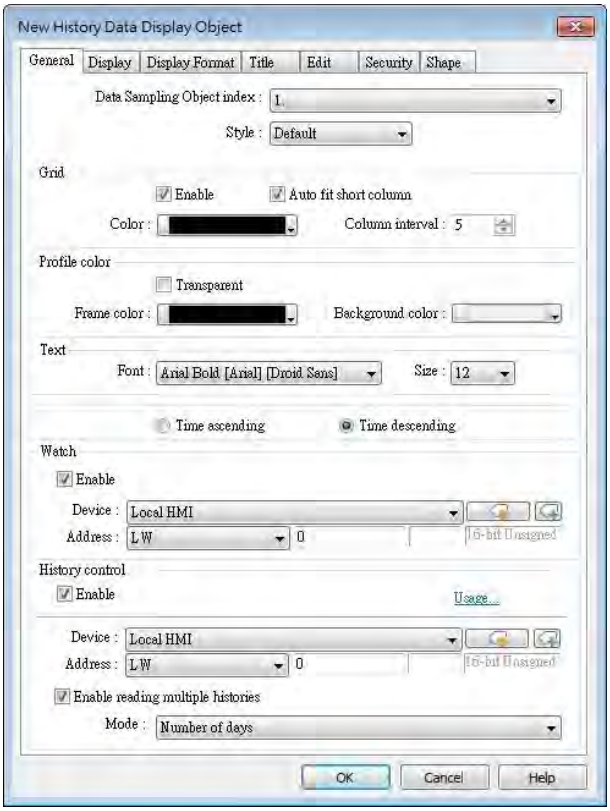
A History Data Display on an eMT/iE/XE/mTV model can display 6MB of data in maximum. The data will not be displayed after exceeding the limit.

13.18.2. Configuration

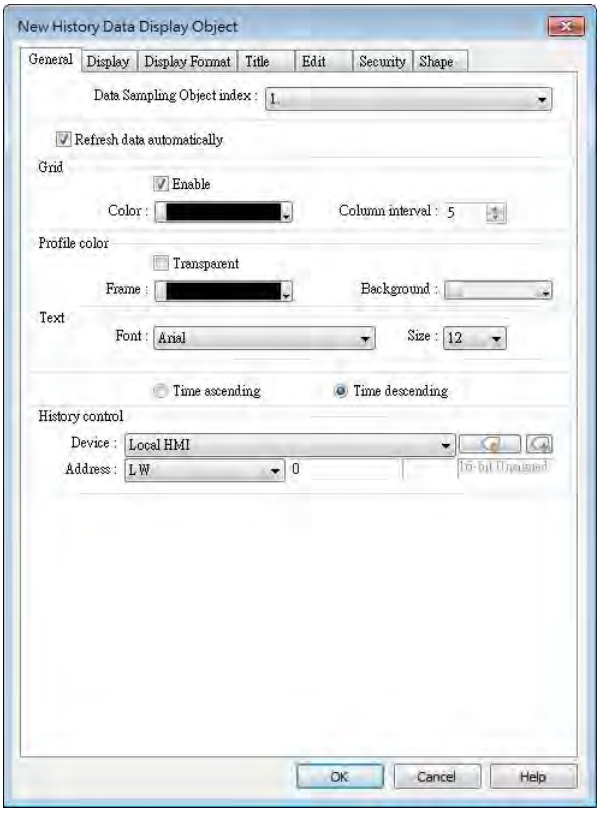
Click [Data/History] » [History Data Display] icon on the toolbar to open a History Data Display object property dialog box. Set up the properties, press OK button, and a new History Data Display object will be created.

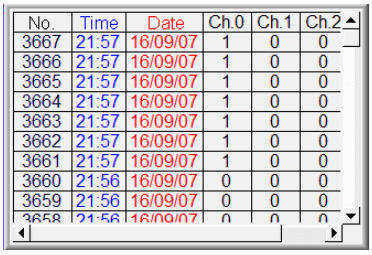
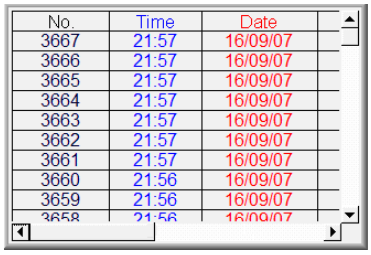
General Tab

cMT, cMT X



eMT, iE, XE, mTV

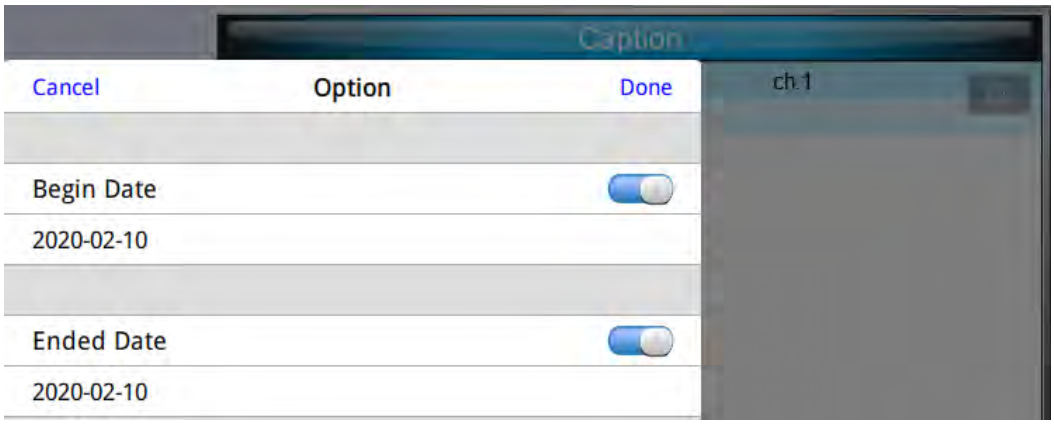


Setting	Description
Data Sampling object index	Select a Data Sampling object as the source data.
Refresh Data Automatically	The system will refresh data every 10 seconds. When this option is not selected, data can be freshed only by changing windows.
Style	Select History Data Display object's style.
Grid	Shows grids between rows and columns. **Color** Change the color of grids. **Column interval** Change the width of each column. The figures below are the examples.  

Profile color	Change the color of frame and background. Use [Transparent] to hide frames and background.
Text	Change the font and font size.
Time / Date	**Time ascending** Put earliest data at the top and the latest data at the bottom. **Time descending** Put the latest data at the top and the earliest data at the bottom.
History Control (eMT, iE, XE, mTV Series)	The history files are sorted by date and each file is given an index. The latest one is assigned index 0 (in most cases: today), the second latest file is assigned index 1, and so on. [History Control] is used to specify the history data to be shown.
Watch (cMT / cMT X Series)	By tapping on a record in History Data Display object, data in the selected row can be output to the designated addresses.
History Control (cMT / cMT X Series)	Same as eMT, iE, XE mTV Series. **Enable reading multiple histories:** In the following two modes, days are calculated backward. **Number of days** Suppose the address is set to LW-n, then: [LW-n]: The start day index from which to calculate number of days. 0: today, 1: yesterday, and so on. [LW-n+2]: The total number of days before the start day. **Specific days** Suppose the address is set to LW-n, then: [LW-n]: The start day index from which to calculate number of days. 0: today, 1: yesterday, and so on. [LW-n+2]: The end day index, the specified value must be greater than that of the start day index. e.g. start: 4, end: 7, the specified days will be 4 days before to 7 days before.

Note

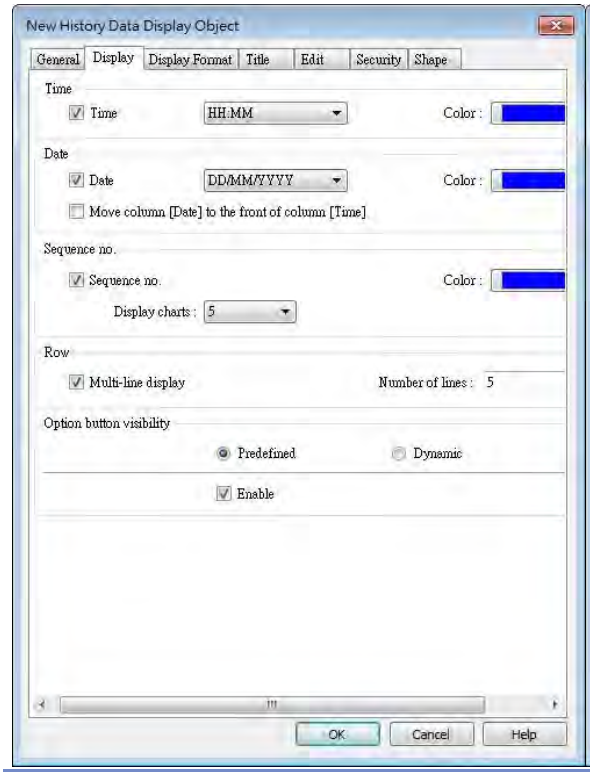
- When using cMT / cMT X Series, use the gear icon in the upper-right corner of History Data Display object in cMT Viewer to select the date and display the data.



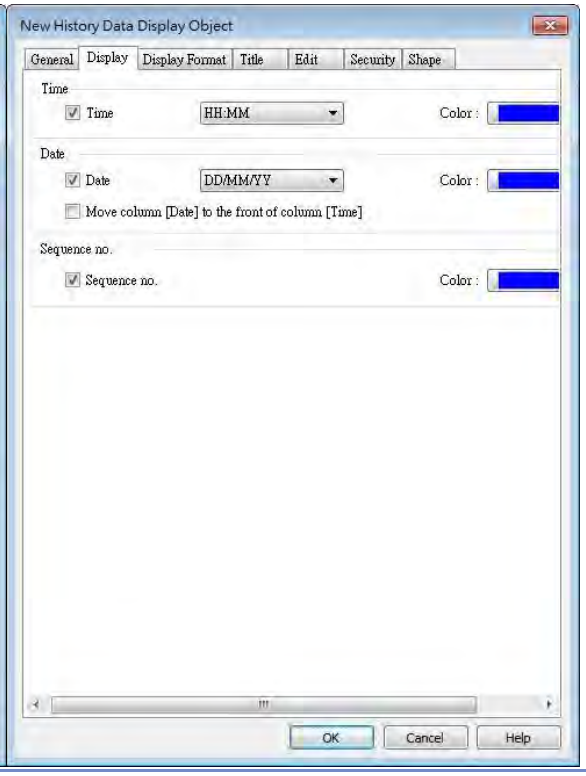
Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

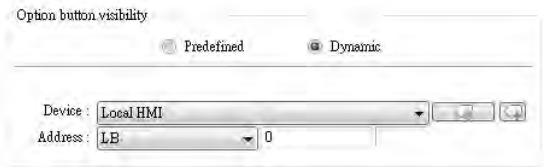
Display Tab

cMT, cMT X



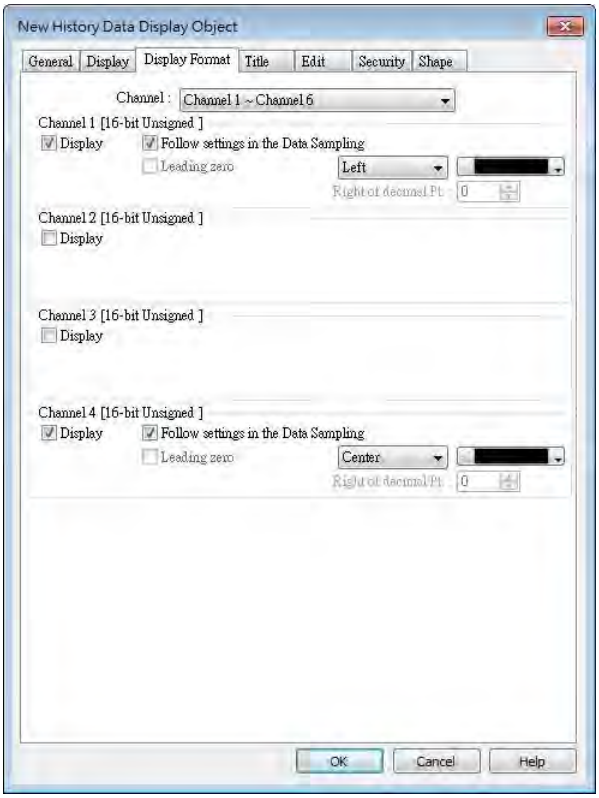
eMT, iE, XE, mTV



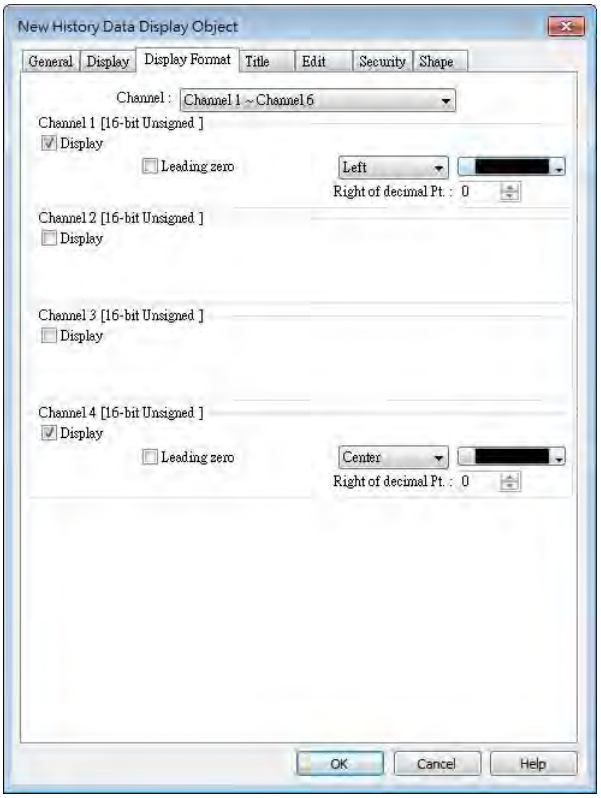
Setting	Description
Time / Date	Enable or disable showing the time and date and configure its format and color. **Move column [Date] to the front of column [Time]** Swap the position of column [Date] and column [Time].
Sequence no.	Show the sequence number of all records.
Row	**Multi-line display** Display a specified number of lines of data. Data will be displayed as specified regardless of the actual number of lines of data.
Option button visibility (cMT / cMT X Series)	Show or hide the option button . **Predefined** Select [Enable] to show the option button. **Dynamic**  Designate a bit address to control option button visibility.

Data Format Tab

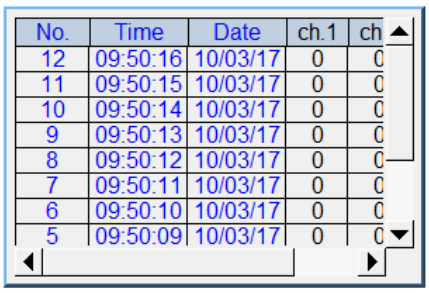
cMT / cMT X Series



eMT, iE, XE, mTV Series

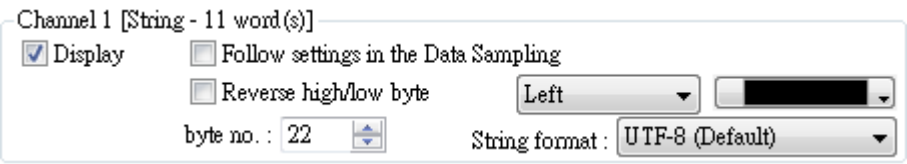


Setting	Description
Channel	Each History Data Display object can display up to 64 channels. Check [Display] to select the channels to be shown on the screen. In the figure above, there are 4 channels (channel 1 to channel 4) in the Data Sampling object, and only Ch.1 and Ch.4 are selected. The data formats are shown next to channel name. The data format of each channel is decided by the corresponding Data Sampling objects. The result is shown below:



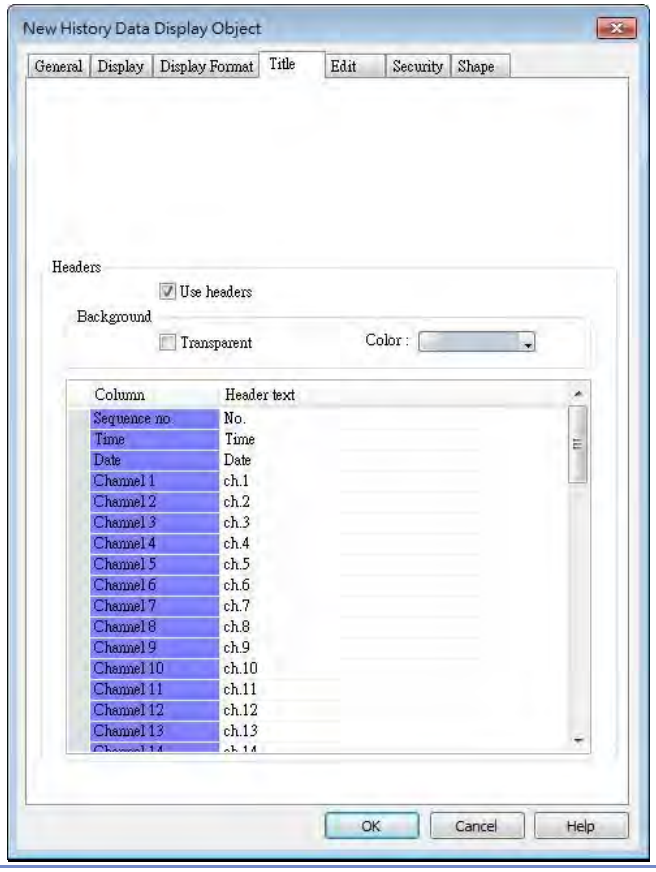
No.	Time	Date	ch.1	ch
12	09:50:16	10/03/17	0	0
11	09:50:15	10/03/17	0	0
10	09:50:14	10/03/17	0	0
9	09:50:13	10/03/17	0	0
8	09:50:12	10/03/17	0	0
7	09:50:11	10/03/17	0	0
6	09:50:10	10/03/17	0	0
5	09:50:09	10/03/17	0	0

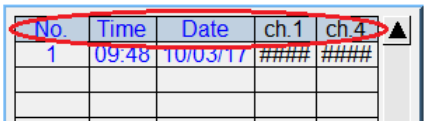
When using cMT / cMT X models, Data Sampling's Data Format settings can be applied to History Data Display object.
Two display modes are available when displaying [String] format in History Data Display object:

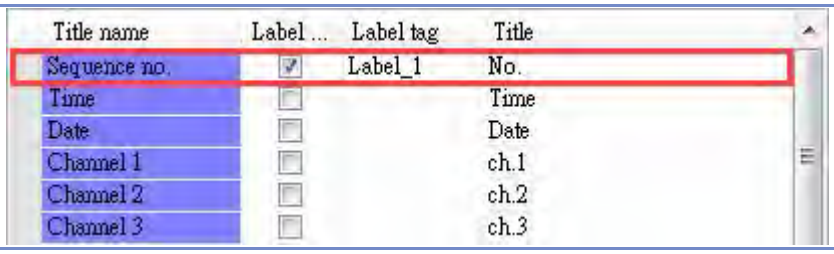


- Follow settings in the Data Sampling.
- Reverse high byte and low byte data and then display.

Title Tab



Setting	Description
Use headers	Enable or disable title, which is marked as shown below: 
Background	**Transparent** When selected, hide the background for title area. **Color** Set the background color of title.
Setting	Defines the text to be shown on the title. The text can be edited in Label Tag Library. Build the Label Tag Library first, and in History Data Display settings dialog select the checkbox in the Label Library column, and then select the label tag.

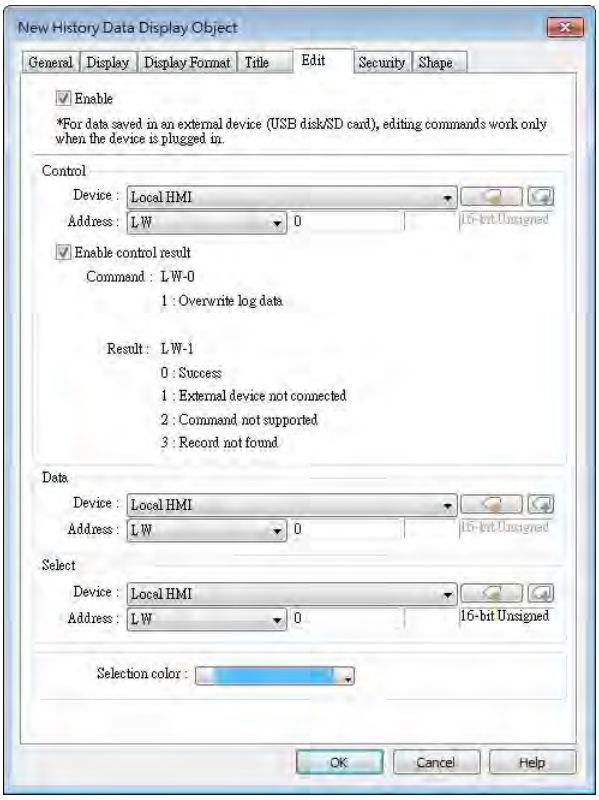


Note

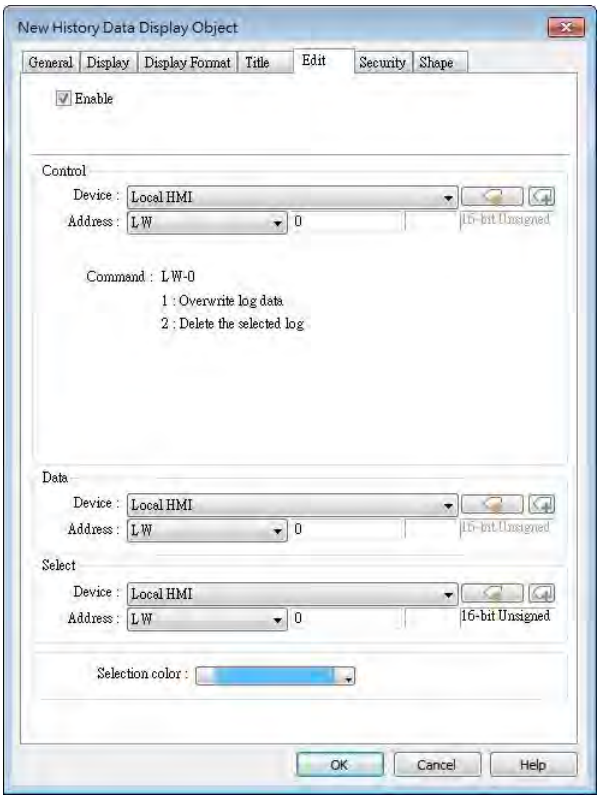
- After running simulation on PC, to run simulation again using the same project that contains some changed data, please find the HMI_memory, SD_card, or usb1 folder in EasyBuilder Pro installation, and then delete the old data sampling records in it, to prevent the system from reading old data in the second simulation.

Edit Tab

cMT / cMT X Series



eMT, iE, XE, mTV Series



Setting	Description
Control address	LW-n: Command Value: 1 – The data in Data Address will overwrite the data in data log. Value: 2 – The selected data log will be deleted. LW-n+1: Result (cMT / cMT X Series)

Value	Description
0	The command is executed successfully.
1	The external device is not connected.
2	The command is not supported.
3	The record cannot be found.

Data address

The system will start reading row by row from the selected one in History Data Display, and change the data log accordingly.
Please note that the settings in this address must be identical to that in Data Sampling.

Select address

By changing the value in Select Address, the corresponding row in History Data Display can be selected.
Please note that:

1. When the value in Select Address is 0, no row will be selected, and the value in Data Address will stay the same as that of the previously selected row number.
2. When the value in Select Address exceeds the total number of rows in History Data Display, the last row (the one with largest row number) will be selected.

Note

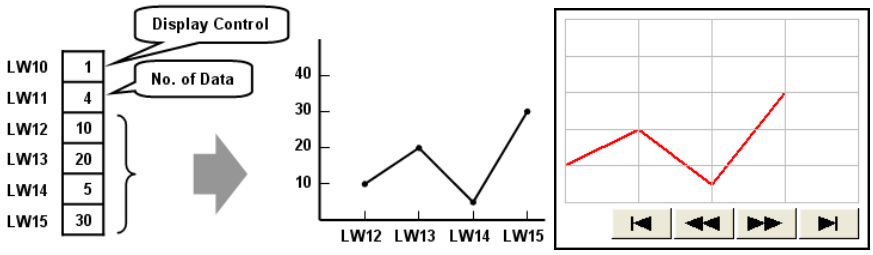
- When an external device is used to save data log, removing the external device will make Edit function ineffective. Editing can keep on when the external device is inserted again and the system starts reading historical data.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

13.19. Data Block Display

13.19.1. Overview

Data Block is a combination of several word devices with continuous address, where the X axis of Data Block Display object represents the address and the numbers on the Y axis represent the data values in the corresponding address. Data Block Display object can display multiple data blocks. For example, it can display two data blocks LW-12~LW-15 and RW-12~RW-15 in trend curves simultaneously. It is very useful to observe and compare the difference of trend curves.

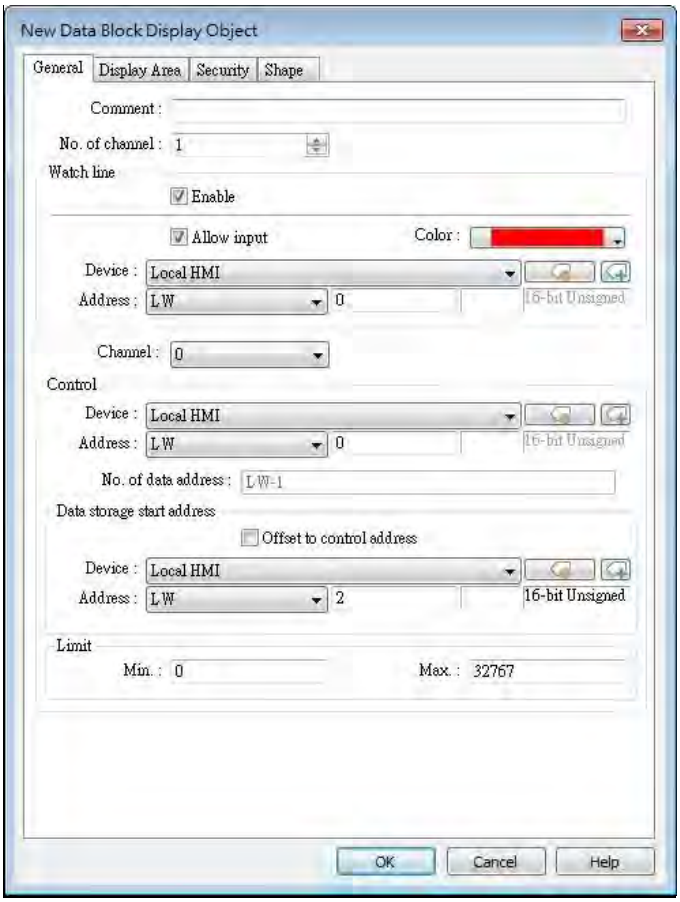


The display result

13.19.2. Configuration

Click the [Object] » [Chart] » [Data Block Display] icon on the toolbar to open the property dialog box. Set up the properties, press OK button, and a new Data Block Display object will be created.

General Tab



Setting	Description
Comment	Description of the object.
No. of channel	Set the no of channel for this object. Each channel represents one data block. The maximal number of channels is 12.
Watch Line	If enabled, when user touches the [Data Block Display] object, it will display a vertical cursor line on it, and store the data on the line to the designated registers. See Example 1. **Allow input (cMT / cMT X Series)** With this checkbox selected, entering a value in the specified address for watch line can move the verticle watch line to the desired position.
Channel	Select the channel to be configured.
Control address	Specify the control address also the data source. Control address is used to control and clear the drawn curve. After executing the operation below, the system will reset the control word to zero. Enter “0” = No action (default) Enter “1” = Draw (Without clear first) Enter “2” = Clear

	Enter “3” = Redraw **No. of data address** If control address is LW-n, then LW-n+1 stores the number of word devices in each data block, i.e. the number of data. The maximum value is 1024. **Data storage start address** If [Offset to start address] is enabled, the [Offset value storage address] will be set as [Control address] + 2. If select 16-bit data format, the address for each data will be start address, start address + 1, start address + 2 and so on. If select 32-bit data format, the address for each data will be start address, start address + 2, start address + 4 and so on. For more information about control address, see Example 2 to 5.
Limit	Set the minimum and maximum limit for the curve.

Note

- The system can draw at most N curves, where N = 32 divided by the number of channel.

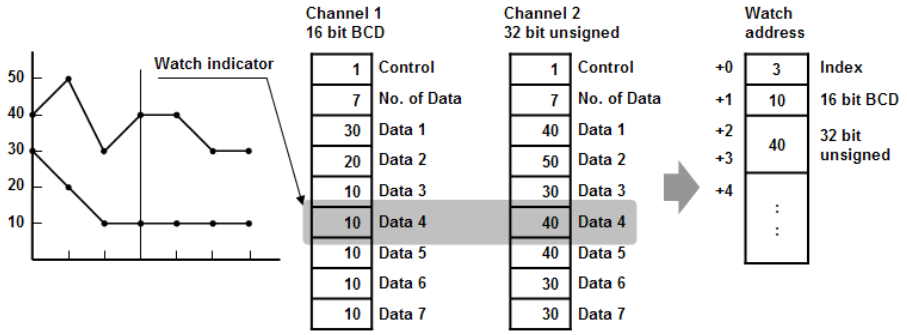
Example 1

How to use watch (Cursor Line) feature

Use “Watch” function to check the value of any point of the curve. When the user touches [Data Block] object, it will display a “cursor line”, and the system will write the index and value of that data on the cursor line to the designated address.

Data Format	Index Value	Channel 1 Value	Channel 2 Value
16-bit	Address	Address + 1	Address + 2
32-bit	Address	Address + 2	Address + 4
64-bit (cMT / cMT X only)	Address	Address + 4	Address + 8

When watch address is set to LW-n, the value written into LW-n represents the channel index number to be called up. (Start form 0)

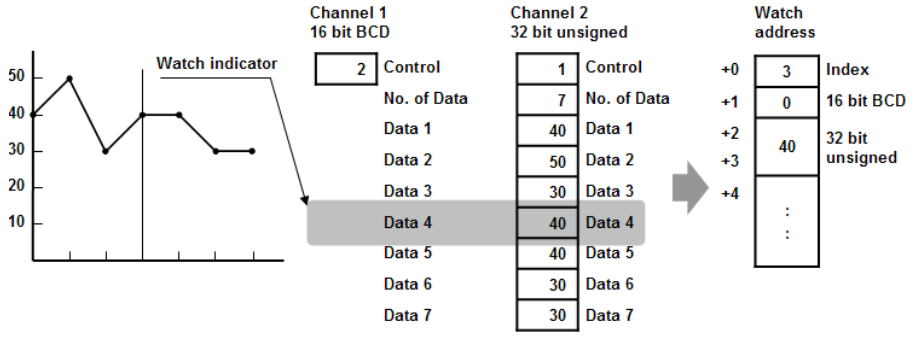


Note

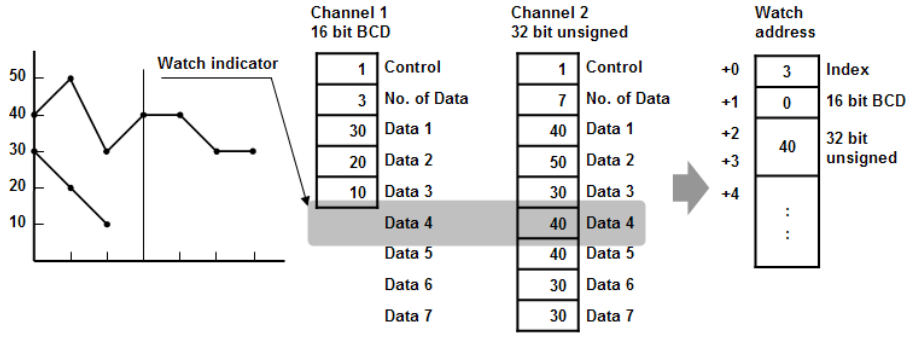
- [Data Index] is a 16 bit unsigned integer. When the designated register of cursor line is 32

bit device, it will be stored in the bit 0-15.

- If the channel to be viewed has no data, “0” will be displayed, as shown below. In the example, there is no data in channel 1, when the cursor points at Data 4, “0” will be displayed as shown below.



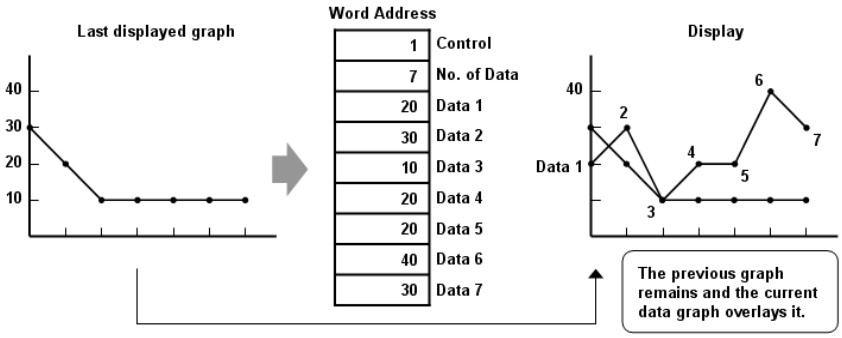
- If there is less data in Channel 1, “0” will be displayed, as shown below.



Example 2

How to show a data block

1. Write the number of data to [No. of data address], i.e. “Control address+1”
2. Store the data consecutively beginning at [Data storage start address].
3. Write “1” to [Control address] to draw the curve without cleaning the plot. All previous curves will not be erased.
4. The system will write “0” to [Control address] after marking the plot.



Note

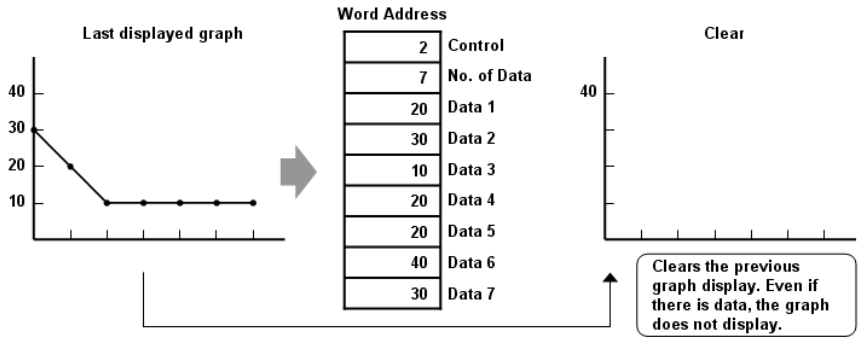
- Do not change the content of [Control address], [No. of data address] and [Data storage start address] between step 3 and step 4 above as doing so might cause error for the

trend curve plot.

Example 3

How to clear the graph

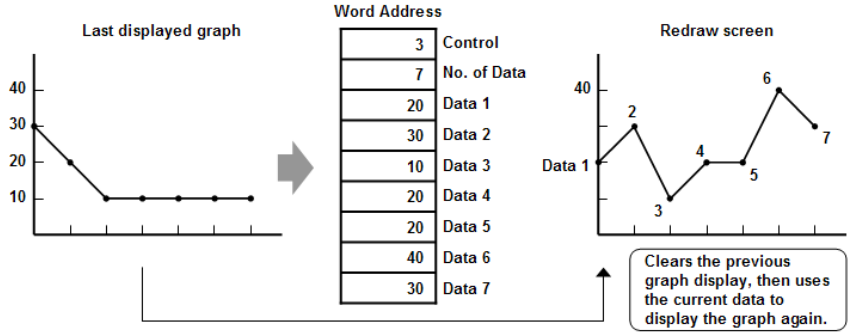
1. Write “2” to [Control address], all the trend curves will be cleared.
2. The system will write “0” to [Control address] after the trend curve is cleared.



Example 4

How to clear the previous trend curve and display new one

1. Write the number of data to [No. of data address], i.e. “control address+1”
2. Store the data consecutively beginning at [Data storage start address].
3. Write “3” to [Control address], the previous trend curves will be cleared and the new content in data block will be plotted on the screen.
4. The system will write “0” to [Control address] after the trend curve has been plotted.



Example 5

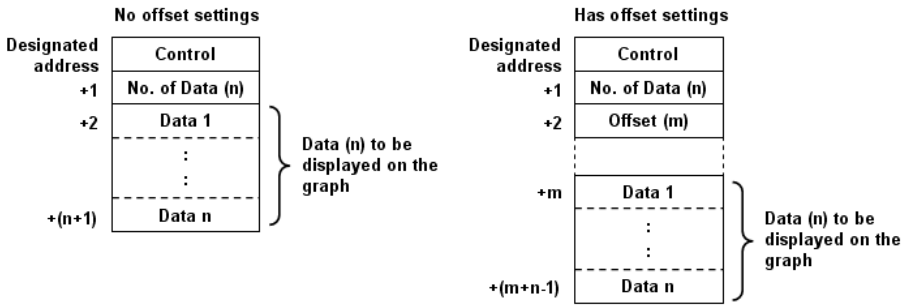
How to use offset mode

If [Offset to start address] is selected, [Control address], [No. of data address], and [Offset value storage address] will use 3 consecutive addresses.

For example, assume the total number of channels is 3 (start from 0 to 2), and the [Control address] are LW-0, LW-100, and LW-200, respectively. Then, the other addresses will be set as follows: (In the example, format 16-bit Unsigned is used and [Offset value storage address] are all m).

Item	Channel 0	Channel 1	Channel 2
Control Address	LW-0	LW-100	LW-200
No. of data address	LW-1	LW-101	LW-201
Offset value storage address	LW-2 (=m)	LW-102 (=m)	LW-202 (=m)
Data 1	LW-0+m	LW-100+m	LW-200+m
Data 2	LW-1+m	LW-101+m	LW-201+m
…	…	…	…

The following figure on the left shows the result when offset mode is not used while the figure on the right shows the result when offset mode is used.

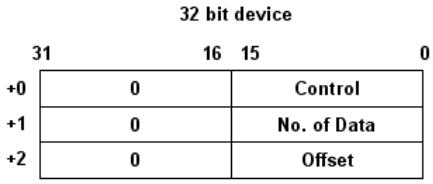


Note

- When [Control address] is set to LW-n, [No. of data address] and [Offset value storage address] are as follows:

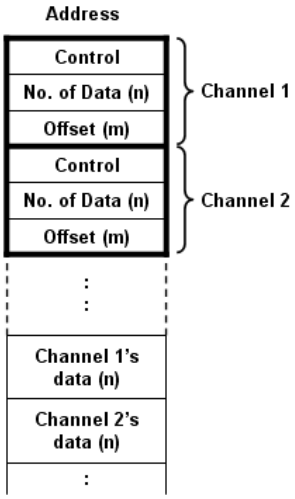
Data Type	16-bit	32-bit	64-bit (cMT / cMT X only)
Control address	LW-n	LW-n	LW-n
No. of data address	LW-n+1	LW-n+2	LW-n+4
Offset value storage address	LW-n+2	LW-n+4	LW-n+8

- If the control registers are 32-bit devices, only bit 0-15 will be used for control purpose, bit 16-31 will be ignored. (as illustrated below)

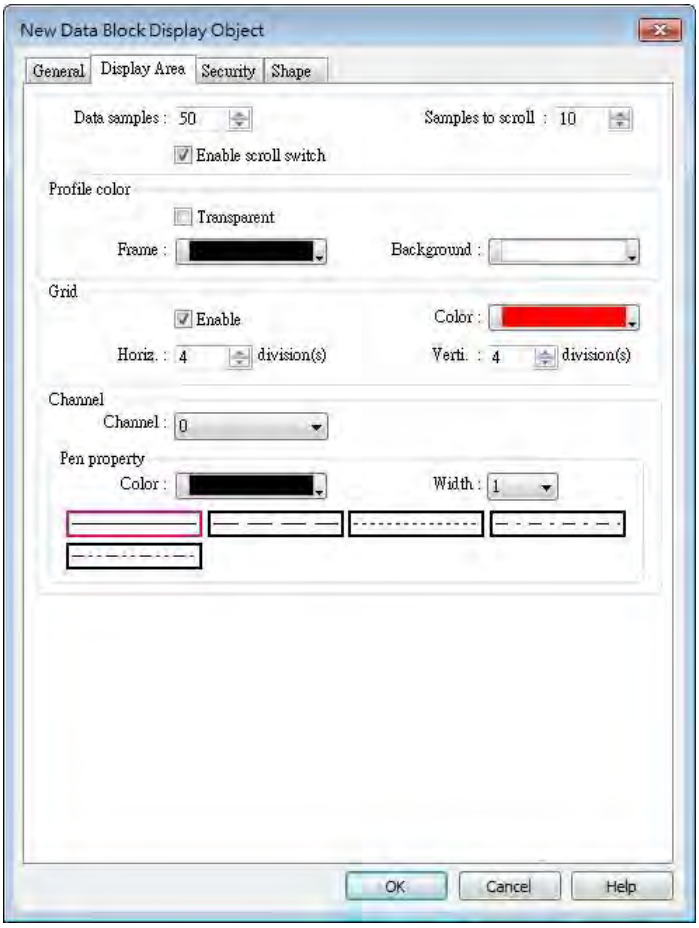


- When the value in [Control address] is not zero, the system will read [No. on data address] and [Offset value storage address].
- It is recommended to use [Offset to start address] for data block display with multiple

channels and the same device type. As shown in the following figure, The control words of channel 1 is located from LW-n, the control words of channel 2 is located from LW-n+3, and so on.



Display Area



Setting	Description
Description	**Data samples**

	Configure the maximal number of data samples (points) to be displayed. **Samples to scroll** Configure the number of data samples being scrolled. **Enable scroll switch** Clicking displays the previous or next data point. Clicking displays the first or the last data point.
Profile	Set the color of the frame and background of the object. **Transparent** Hides the background. Color selection will not be available.
Grid	Set the number of horizontal and vertical divisions shown by grid.
Channel	Set the color, width and style of each curve.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.20. XY Plot

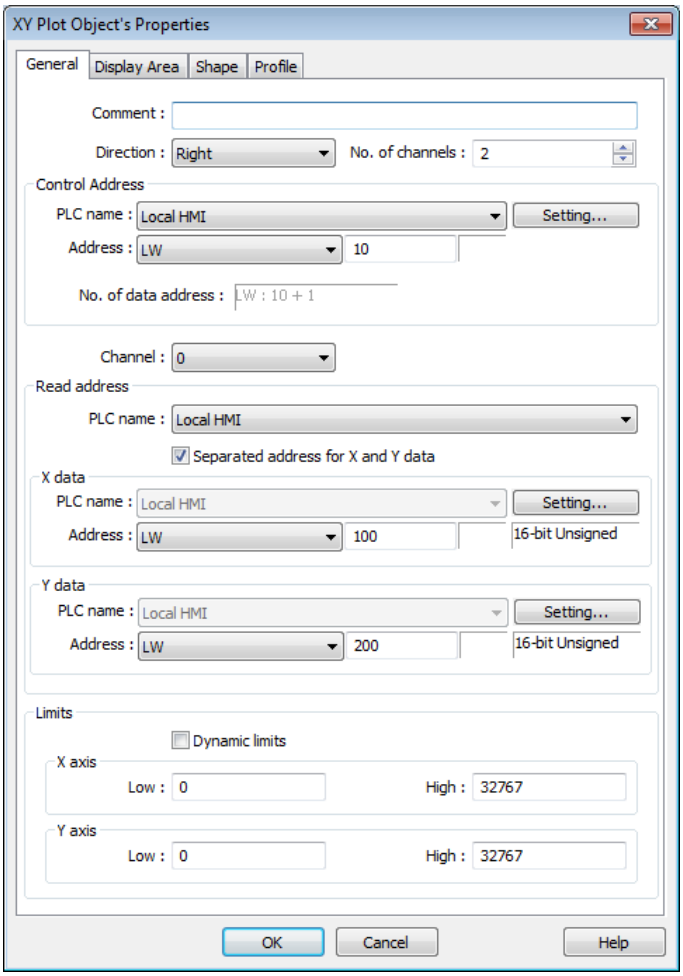
13.20.1. Overview

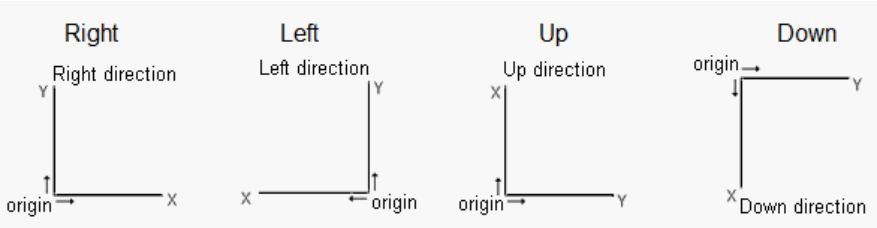
XY Plot object is used to display values for two variables (x,y) for a set of data, where the data comes from word registers. Up to 16 channels can be displayed simultaneously. This object facilitates data observation and analysis. Additionally, negative numbers can be displayed as well.

13.20.2. Configuration

Click [Object] » [Chart] » [XY Plot] icon on the toolbar to open a [XY Plot] object property dialog box.

General Tab



Setting	Description
Direction	There are four selections, right, left, up or down. 
No. of channels	Set the number of channels for observation.
Control address	Controls the operation of all channels simultaneously. When the [Control address] is LW-n, assigning values to LW-n will issue commands to XY plot according to the table below. Meanwhile, LW-n+1 l controls the number of data points plotted. After operation, the [Control address] will be reset to 0. **Control address** / **Value** / **Result** LW-n / 1 / Plots point on XY curve. (The plotted points are kept.) LW-n / 2 / Clears all XY curves. LW-n / 3 / Clears then plots new XY curve. LW-n+1 / Any number / Controls the number of data points plotted. **No. of data address** Controls the number of data points. Each channel can plot up to 1023 points.
Channel	Select a channel to configure.
Read Address	**Device** Select a PLC which will be the source of [X data] and [Y data] and designate a read address. The format of the data register blocks used for the display channels depends on whether [Separated address for X and Y data] and/or [Dynamic limits] has been selected. See Example 1.
Dynamic limits	● When **not** selected (See Example 2) The Low and High limits can be set by entering constants. The Low and High limits are used for calculating X and Y range in percentage. ● When selected (See Example 3) A zoom effect can be created by changing the Low / High Limits.

Example 1

The format of the data register blocks used for the display channels depends on whether [Separated address for X and Y data] has been selected, and if [Dynamic limits] has been selected. The following explains the situations where 16-bit register is used:

- If [Separated address for X and Y data] is **not** selected, and set [Read address] to LW-n:

	Select [Dynamic limits]		**Not** select [Dynamic limits]	
	X data	**Y data**	**X data**	**Y data**
Low Limit	LW-n	LW-n+2	Constant	Constant
High Limit	LW-n+1	LW-n+3	Constant	Constant
1st data	LW-n+4	LW-n+5	LW-n+0	LW-n+1
2nd data	LW-n+6	LW-n+7	LW-n+2	LW-n+3
3rd data	LW-n+8	LW-n+9	LW-n+4	LW-n+5
4th data	LW-n+10	LW-n+11	LW-n+6	LW-n+7

- If [Separated address for X and Y data] is selected, and set [X data] to LW-m, [Y data] to LW-n:

	Select [Dynamic limits]		**Not** select [Dynamic limits]	
	X data	**Y data**	**X data**	**Y data**
Low Limit	LW-m+0	LW-n+0	Constant	Constant
High Limit	LW-m+1	LW-n+1	Constant	Constant
1st data	LW-m+2	LW-n+2	LW-m+0	LW-n+0
2nd data	LW-m+3	LW-n+3	LW-m+1	LW-n+1
3rd data	LW-m+4	LW-n+4	LW-m+2	LW-n+2
4th data	LW-m+5	LW-n+5	LW-m+3	LW-n+3

Example 2

When [Dynamic limits] is not selected, the Low and High limits can be set. The Low and High limits are used for calculating X and Y range in percentage.

$$\text{Scale (\%)} = \frac{\text{Read Address Value} - \text{Low Limit}}{\text{High Limit} - \text{Low Lmit}}$$

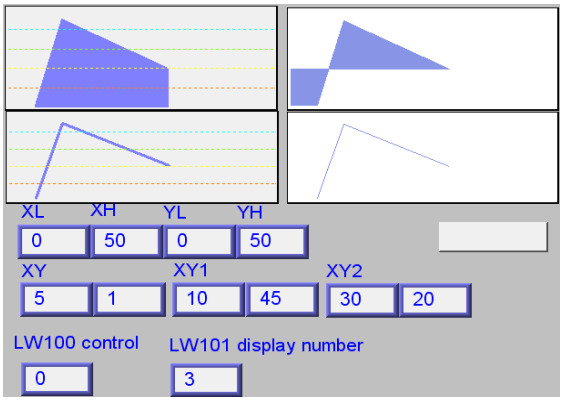
If [Separated address for X and Y data] is **not** selected and the address is LW-n, the corresponding limits are retrieved from the addresses as shown in the following table.

Data format	16-bit	32-bit	64-bit (cMT / cMT X only)
X axis low limit	LW-n	LW-n	LW-n
X axis high limit	LW-n+1	LW-n+2	LW-n+4
Y axis low limit	LW-n+2	LW-n+4	LW-n+8
Y axis high limit	LW-n+3	LW-n+6	LW-n+12

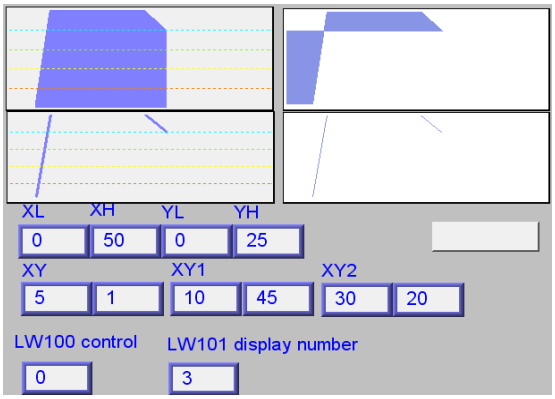
Example 3

If **[Dynamic limits]** is selected, a zoom effect can be created by changing the setting of Low / High Limits.

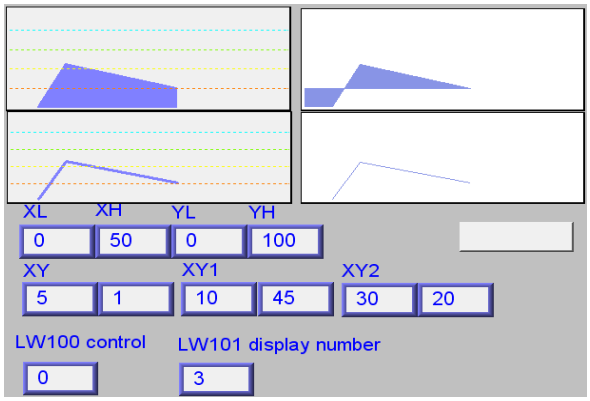
In the following example, XL=X low limit, XH=X high limit, YL=Y low limit, YH=Y high limit, and XY, XY1, XY2 are three XY data. When changing the high limits of X and Y axis, the result is shown below:



Original



Change the high limit of Y axis to 25. (zoom in)



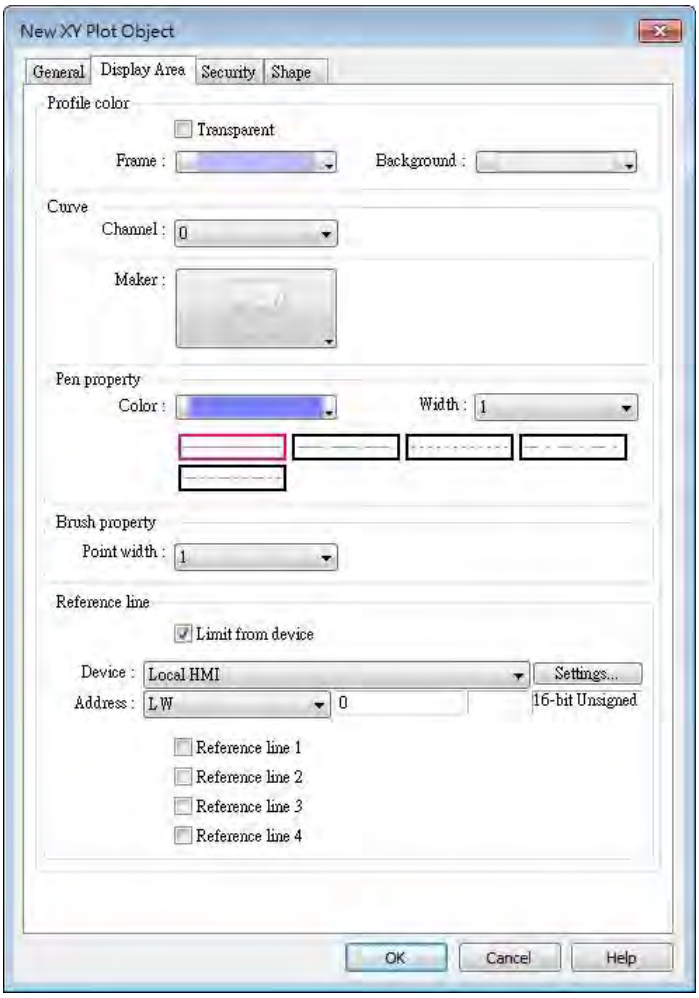
Change the high limit of Y axis to 100 (zoom out)

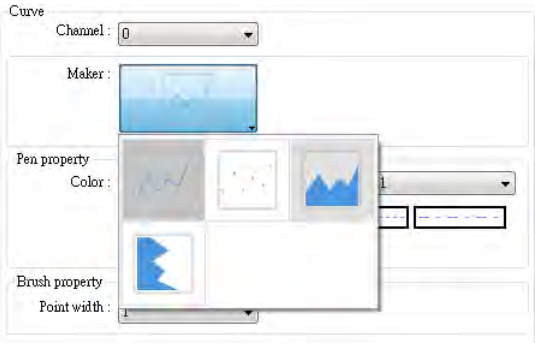
For more information, see “13.17 Trend Display”.

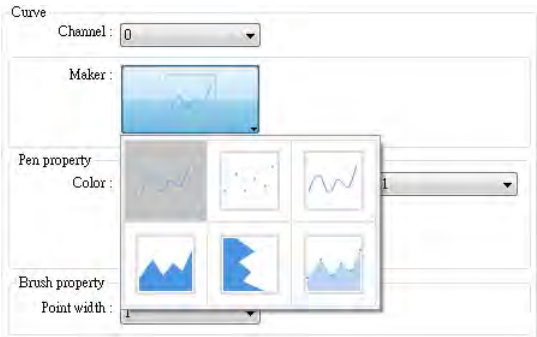
Note

- X and Y data can be set to different formats. For example: If X data uses 16-bit unsigned, Y data uses 32-bit signed, please note the address setting.
- When using a Tag PLC, such as AB tag PLC, X and Y must be in the same format. When using different formats a warning will be shown.

Display Area Tab



Setting	Description
Profile color	Select the color of the frame and the background, or select [Transparent] check box to hide the frame and background.
Curve	For each channel select the properties of color, width, and line style.
Maker	Select the style of XY Plot. For eMT/iE/XE/mTV series, there are four different types of XY plot as shown below: 

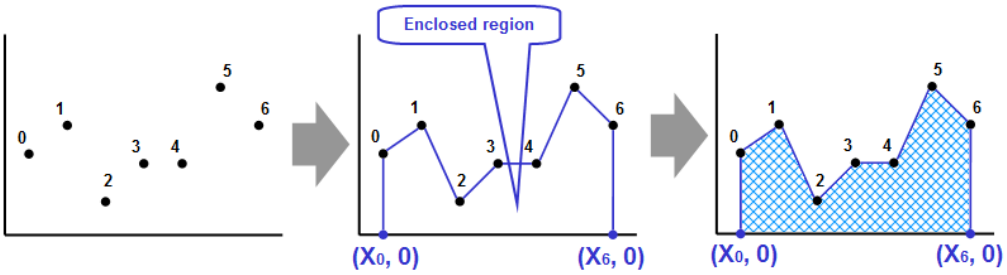
	For cMT / cMT X Series, , there are six different types of XY plot as shown below:  See Example 4.
Pen property	Select color, width and style for the line in XY plot.
Brush property	Select point width.
Reference line	Up to 4 horizontal reference lines can be shown on the graph. Fill in high, low limits and Y axis percentage values. Different colors can be selected for each reference line. If [Limit from PLC] is selected, designate a register to be the read address of reference line.

Example 4

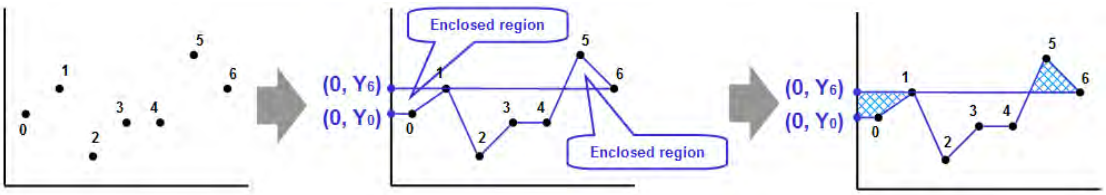
The curve shown in the following figure is drawn with 7 points numbered from P0 to P6.

The steps the system draws the X-axis Projection are:

1. Calculates the two points in X-axis – (X_0, 0) and (X_6, 0).
2. Link all the points in the order of (X_0, 0), P0… P6, (X_6, 0) and returns to (X_0, 0) at last.
3. Fill out all enclosed areas.



Similarly for Y-axis projection:



Note

- XY Plot can be drawn repeatedly up to 32 times:
  1 channel➔32 times
  2 channels➔16 times
  The way to calculate: 32 divided by the number of channels.

13.21. Alarm Bar and Alarm Display

13.21.1. Overview

Alarm Bar and Alarm Display objects are used to display alarm messages which are defined in Event (Alarm) Log objects. When the trigger conditions are met, events or alarms will be displayed as they occur in chronological order in Alarm Bar or Alarm Display object.
Alarm Bar scrolls all alarm messages in one single display line, whereas Alarm Display shows alarm messages in multiple lines.

For more information, see “7 Event Log”.

Alarm Bar - Displays alarm messages in one scrolling line.

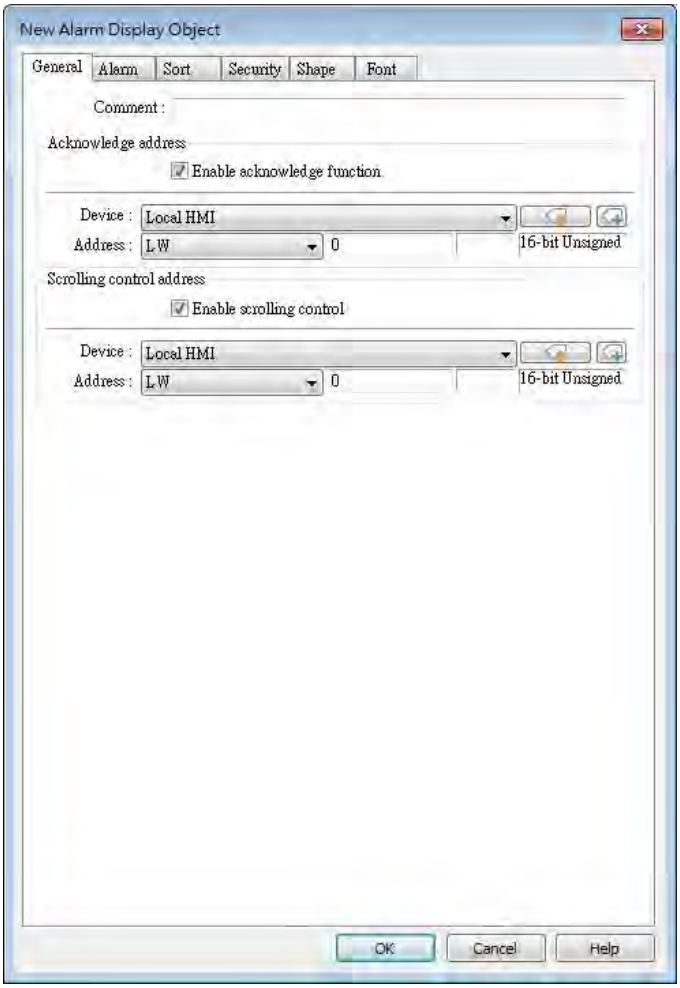
Alarm Display – Displays alarm messages in multiple lines.

13.21.2. Configuration

Click [Data/History] » [Alarm Display] or [Alarm Bar] icon on the toolbar to open the object property dialog box. Set up the properties, press OK button, and a new object will be created.

General Tab

The difference between these two objects is that Alarm Display allows an [Acknowledge address] and a [Scrolling control address] to be set.



Setting	Description
Enable acknowledge function	If selected, the [Acknowledge value] selected for the associated event, specified in Alarm (Event) Log » Message tab will be written to the [Acknowledge address] designated in Alarm Display. For more information, see “7 Event Log”. Acknowledge value for Event/Alarm Display object Acknowledge value : 11
Scrolling control address	If enabled, the value in the designated control address indicates the number of lines to be scrolled down. The minimal value is 0, which indicates the first line. Please note that with this address enabled, the horizontal and vertical scroll bars will not be available on the object; please make sure that the width of the object is enough for completely showing the content.

Note

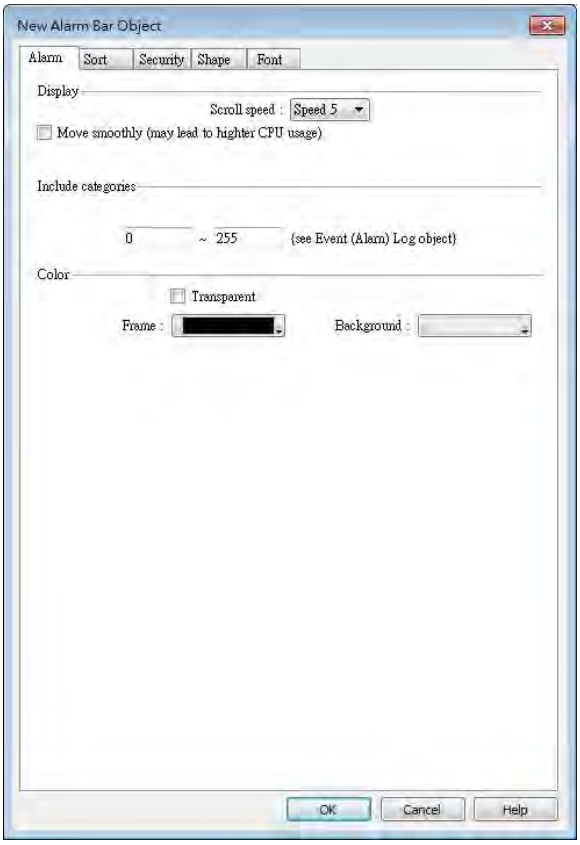
- When using a cMT / cMT X Series model, press and hold the event on the screen to

acknowledge an event; drag a finger on the screen to scroll.

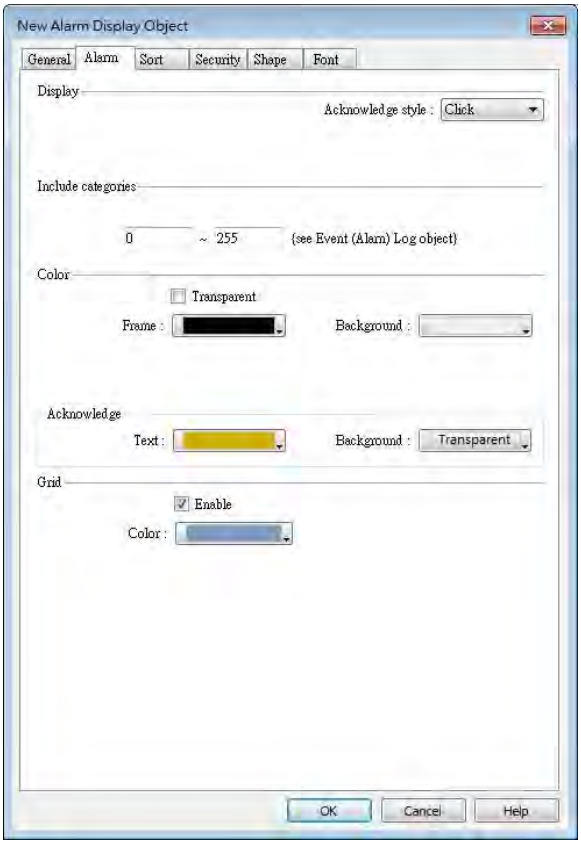
- When using a cMT / cMT X Series model, a default minimal row height exists in Alarm Display objects that have enabled Acknowledge function. The minimal row height prevents the rows to be touch by mistake. On a small-sized HMI, the row height may be slightly taller when Acknowledge function is enabled.

Alarm Tab

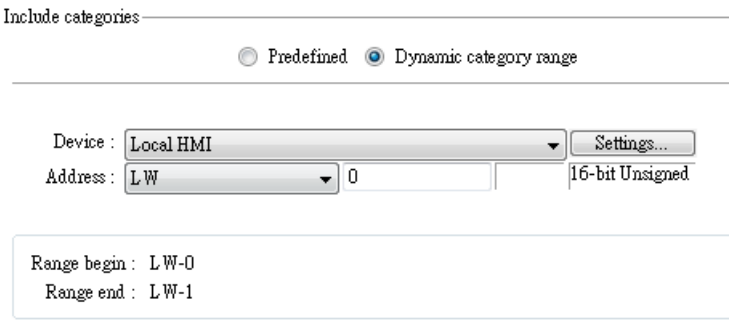
Alarm Bar



Alarm Display

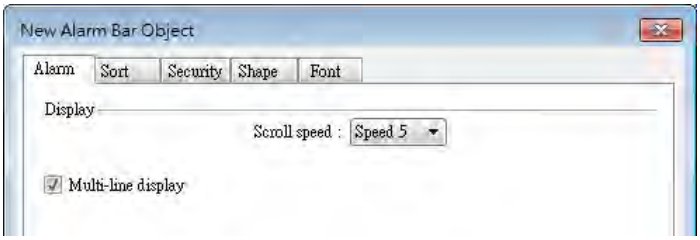


Setting	Description
Include categories	Events in the selected category will be displayed. The categories are set in Event (Alarm) Log object. For example, if the category is set to “2 to 4” here, only events in categories 2, 3, 4 will be displayed. For more information, see “7 Event Log”. **cMT / cMT X Series** Dynamic Category Range is an option available only for cMT Series models. With this option selected, an address can be designated for dynamically selecting event categories to be displayed.



Display

Display settings are only available for Alarm Bar.



Scroll speed

Select one of the speed settings at which the messages scroll.

Multi-line display

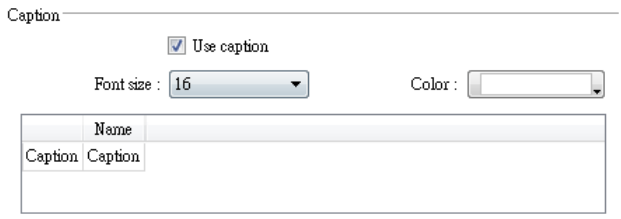
Messages can be displayed in multiple lines with this option selected.

Move smoothly

When this option is selected, the message will move along the alarm bar more smoothly. Please note that enabling this may lead to high CPU loading.

Caption

Caption settings are only available for Alarm Display on cMT / cMT X Series.



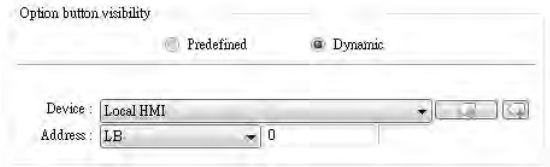
Option button visibility (cMT / cMT X Series)

Show or hide the option button .

Predefined

Select [Enable] to show the option button.

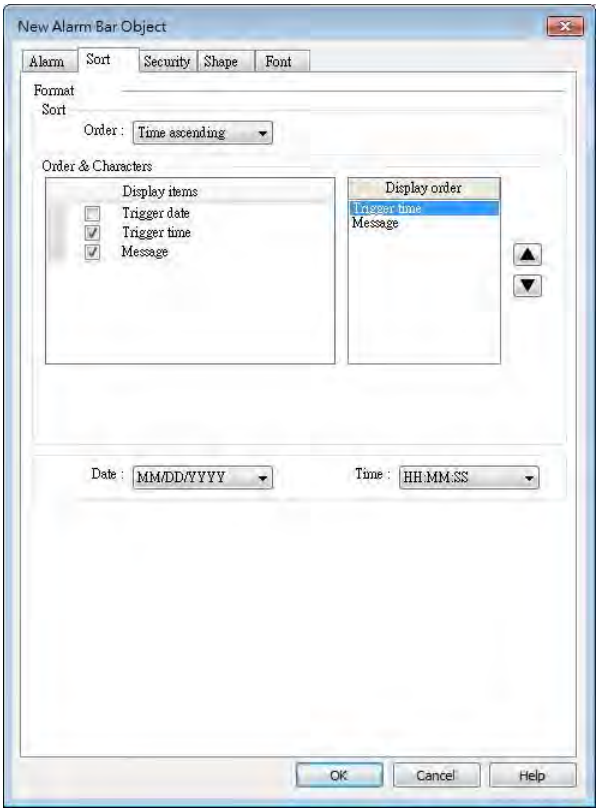
Dynamic



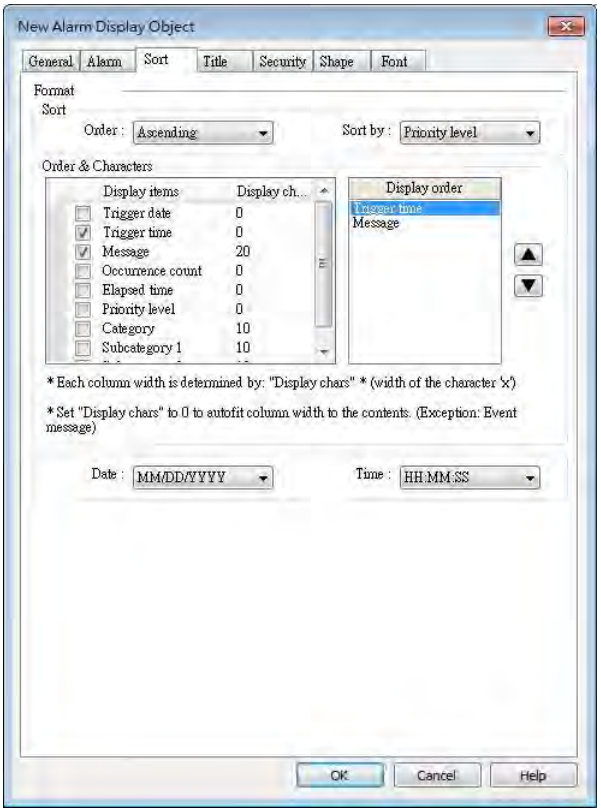
Designate a bit address to control option button visibility.

Sort Tab

Alarm Bar



Alarm Display



Setting	Description
Order (Alarm Bar)	**Time ascending** Latest event is placed last in the list (the bottom). **Time descending** Latest event is placed first in the list (the top).
Order (Alarm Display)	**Sort by:** **Ascending – Priority Level (cMT / cMT X Series)** Event with highest priority level is placed last in the list (the bottom). **Descending – Priority Level (cMT / cMT X Series)** Event with highest priority level is placed first in the list (the top). **Ascending – Occurrence Count** Event with highest occurrence count is placed last in the list (the bottom). **Descending – Occurrence Count** Event with highest occurrence count is placed first in the list (the top). **Ascending - Time**

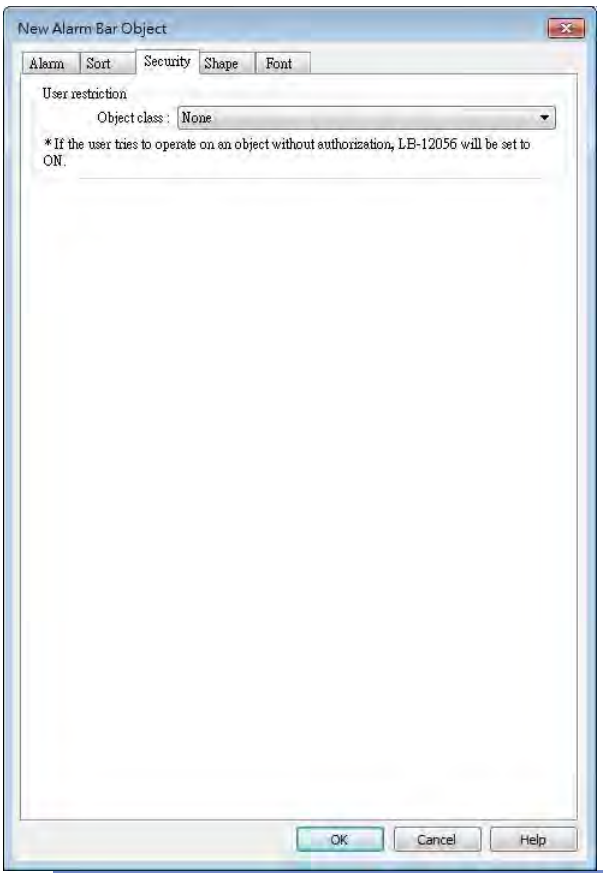
	Latest event is placed last in the list (the bottom). **Descending - Time** Latest event is placed first in the list (the top).
Display order	Select the items to be displayed and use the up and down arrow buttons to adjust the display order of the alarms.
Date	Displays the date tag with each alarm message. The four formats of date tag: MM/DD/YY、DD/MM/YY、DD/MM/YY、YY/MM/DD
Time	Displays the time tag with each alarm message. The four formats of time tag: HH:MM:SS、HH:MM、DD:HH:MM、HH

Title Tab

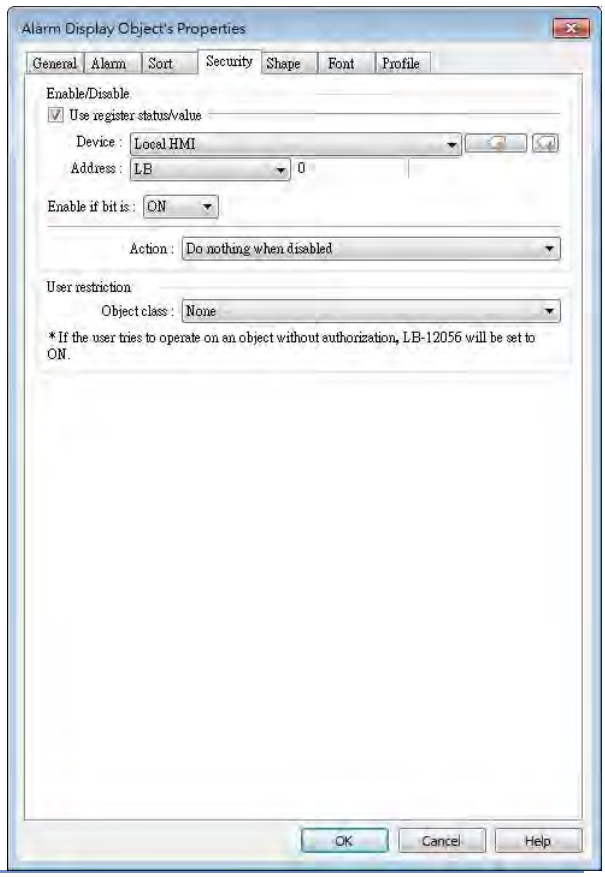
Title settings are available on cMT Series. With [Use headers] selected, the text color and column headers can be configured.

Security Tab

Alarm Bar



Alarm Display

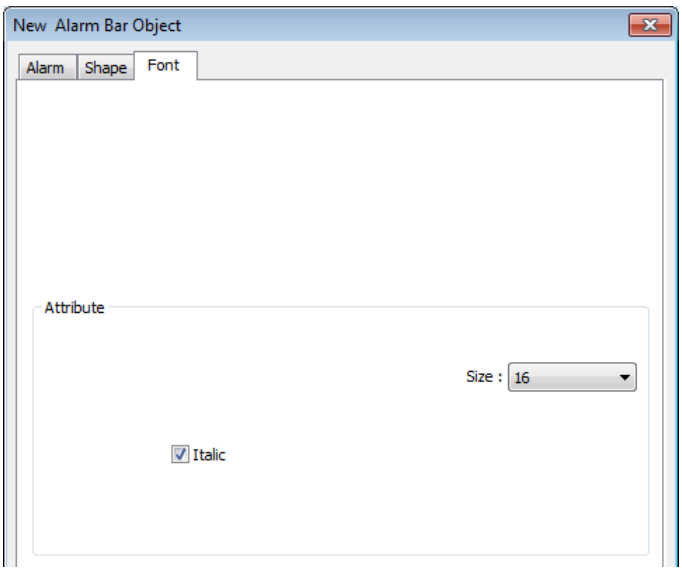


Setting	Description
Interlock	When [Use interlock function] check box is selected, whether the object is operable is determined by the state of a designated Bit address. As shown in the above settings, if LB-0 is ON, the object is operable. **Hide when disabled** When the designated Bit is OFF, the object will be hidden.
User restriction	Set the security class of the object to be operated by an authorized user. **Object class** “None” means any user can operate this object. Only account “admin” can operate “Administrator” object class. **Make invisible while protected** When the user's privilege does not match the object class, the object will be hidden. When this check box is deselected in Alarm Display

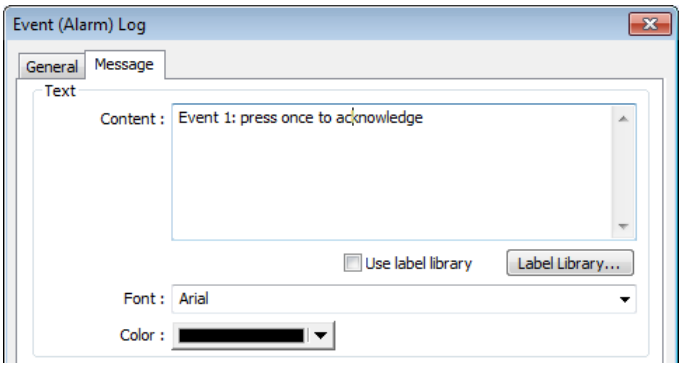
	object settings, the unauthorized user can see the Alarm Display object, but cannot trigger the object or make any change. This check box is greyed out in Alarm Bar object settings.

Font Tab

Set the font size or select [Italic].



The font, color, and content of the alarm messages displayed in Alarm Bar and Alarm Display objects are set in Alarm (Event) Log object:

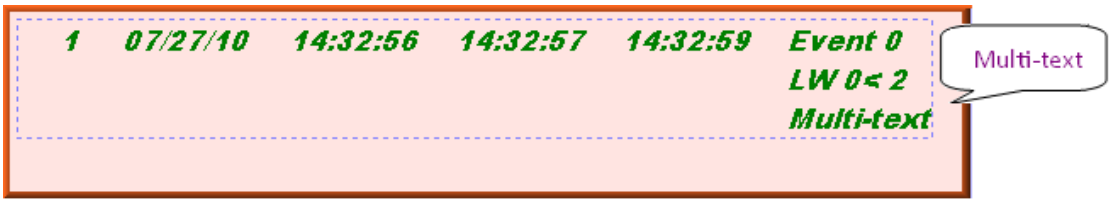


Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.22. Event Display

13.22.1. Overview

Event Display object is used to display event messages which are defined in Event (Alarm) Log and have met a trigger condition. The triggered events are displayed in the chronological order. Event Display object displays: the date and time the event occurs, the time the event is acknowledged, the time the event returns to normal, the event message, the occurrence count, and the elapsed time. Multi-lined messages can also be displayed.

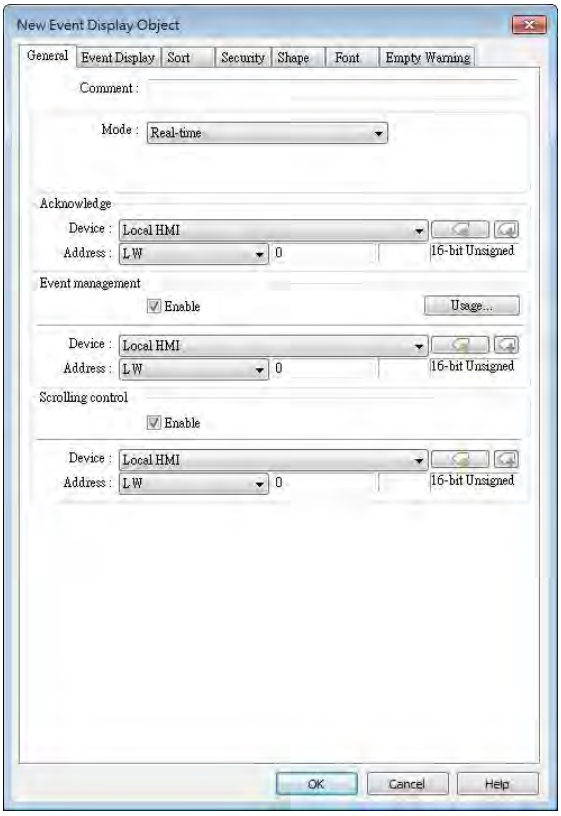


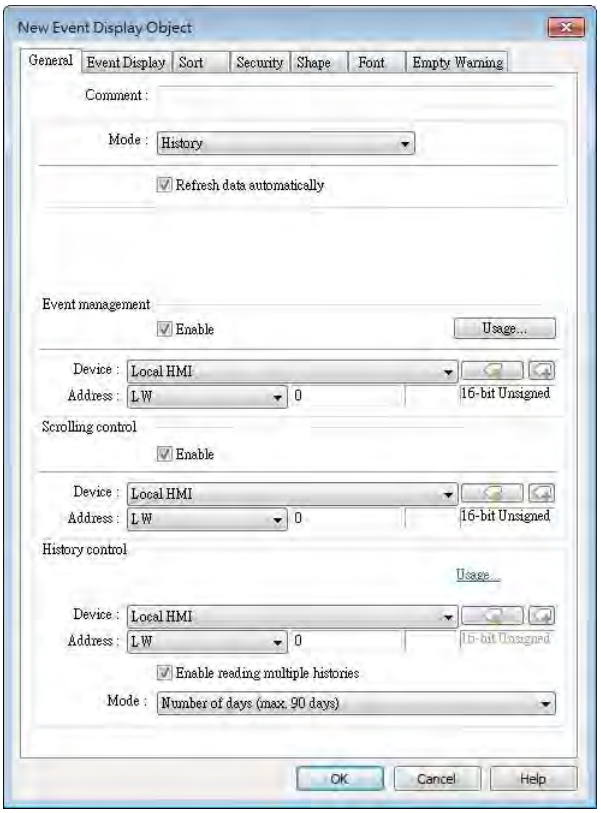
13.22.2. Configuration

Click [Data/History] »[Event Display] icon on the toolbar to open an Event Display object property dialog box. Set up the properties, press OK button, and a new Event Display object will be created.

General Tab

13.22.2.1. eMT, iE, XE, mTV Series





Setting	Description
Mode	The available modes are: [Real-time] and [History]. ● Real-time All the events triggered since HMI starts up are displayed. ● History The system reads the event log in HMI memory and displays them. The content can be updated by changing window. In case when the trend display shows history data from today, the display will refresh once per second.
Acknowledge	When in Real-time mode and an event is acknowledged by touching an active display line, the [Acknowledge value] specified in Event (Alarm) Log object, Message tab, is output to the [Acknowledge address] of Event Display object. For more information, see “7 Event Log”. Acknowledge value for Event/Alarm Display object Acknowledge value : 11
History Control	When in History mode and if: ● [Enable reading multiple histories] is **not** selected

The history files are sorted by date and each file is given an index. The latest one is assigned index 0 (in most cases: today), the second latest file is assigned index 1, and so on. The history control address is used to specify the history file to be displayed.
If control address is LW-100, and four files exist with dates: EL_20100720.evt, EL_20100723.evt, EL_20100727.evt, and EL_20100803.evt.
The index value given to each file is shown below:

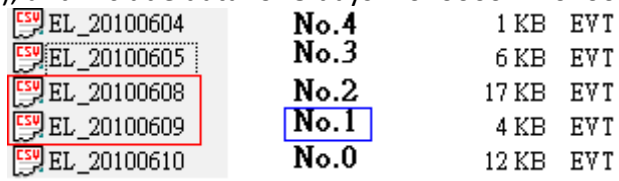
Value in LW-100	The corresponding file
0	EL_20100803.evt
1	EL_20100727.evt
2	EL_20100723.evt
3	EL_20100720.evt

- [Enable reading multiple histories] is selected

Displays a list of events triggered in multiple days. The display range is formed by two consecutive addresses. Please note that the history control address is defined differently in “Number of days” mode and in “Index of the last history” mode.

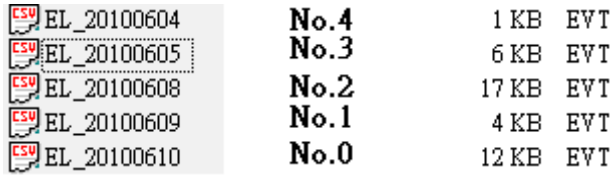
Number of days

Suppose the history control address is set to LW-n, then:
[LW-n]: The start day index (which is expressed as the number of days before today. 0: today, 1: yesterday, and so on.)
[LW-n+1]: The total number of days to display. The days are calculated backward from the start day.
Example: If today were 2010/6/10, and LW-n = 1, LW-n+1 = 3, then the data range will start from 1 day before today, which is 20100609 (yesterday), and include data for 3 days: 20100607~20100609.



Index of the last history

Suppose the history control address is set to LW-n, then:
[LW-n]: The index value of the file to start. 0: the latest file, 1: the second latest file, and so on.
[LW-n+1]: The index value of the file to end. The specified value must be greater than that of the start index.
Example: If LW-n = 1 and LW-n+1 = 3, then file No.1, No.2, No.3 will be displayed.



The maximum size of data that can be displayed is 4MB. Data exceeding the limit will not be displayed.

The following shows how data will be displayed according to data size.
5 history data, each is 0.5MB → Data displayed: 8 x 0.5MB
5 history data, each is 1MB → Data displayed: 4 x 1MB
5 history data, each is 1.5MB → Data displayed:
2 x 1.5MB+1 x 1MB (partial)

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Event management

Enable event management

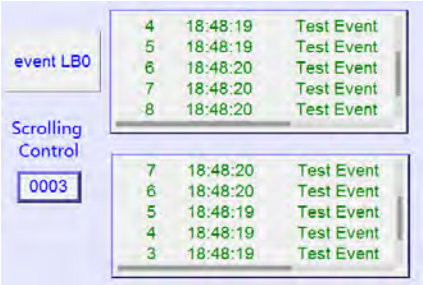
If this check box is selected, writing a specific value into register LW-n and LW-n+1, where n is an arbitrary number, will control [Event Display] object with different commands as shown below:

Address	Value	Command
LW-n	0	Display all events.
	1	Hide [Confirmed] events.
	2	Hide [Recovered] events.
	3	Hide [Confirmed] or [Recovered] events.
	4	Hide [Confirmed] and [Recovered] events.
LW-n+1	1	In real-time mode, delete a single selected event.

Scrolling control

If enabled, the value in the designated control address indicates the number of lines to be scrolled down. The minimal value is 0, which indicates the first line.

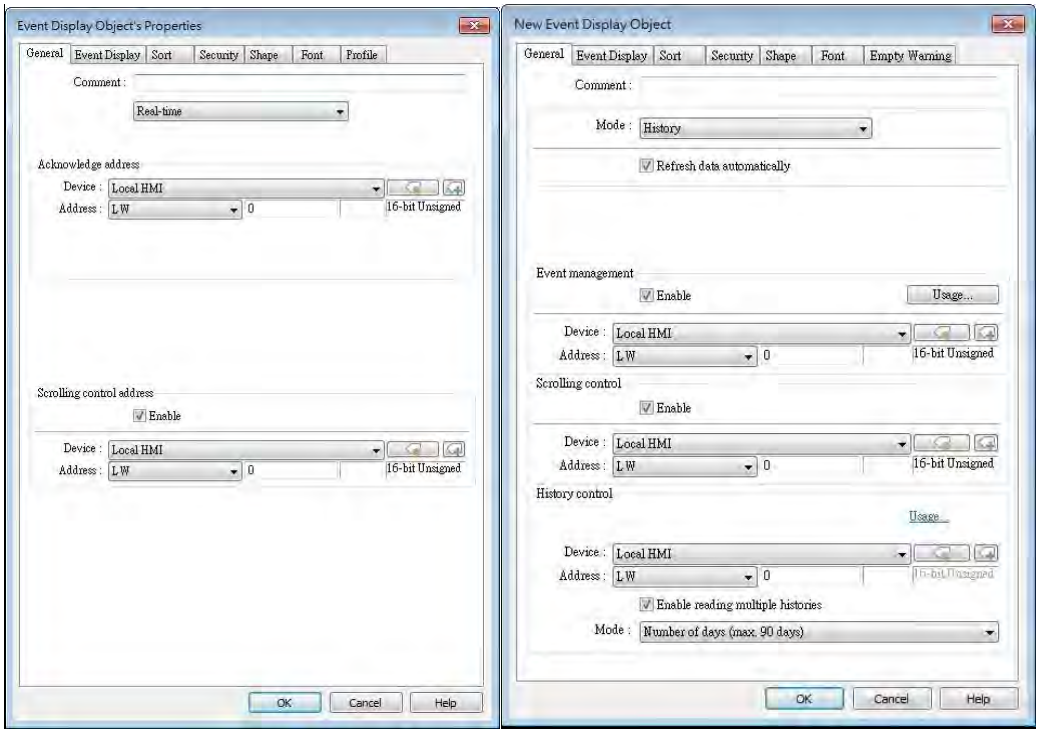
In the following figure, there are 10 events recorded in the object, and value 3 is set at the control address. The upper event object displays the events in time ascending order, and begins at the 4th event; on the other hand, the lower one displays the events in time descending order, and begins at the 7th event.



If [Scrolling control address] is enabled, the scroll bar cannot be used for scrolling, but still shows the relative position of the content. If the control address holds a value that is larger than the total number of lines, the display will stroll to the end. Please note that with this address enabled, the horizontal and vertical scroll bars will not be available on the object; please make sure that the width of the object is enough for completely showing the content.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

13.22.2.2. cMT, cMT X Series

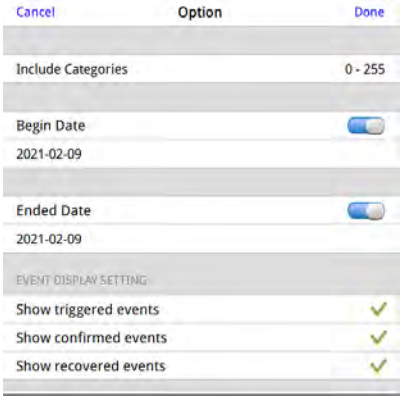


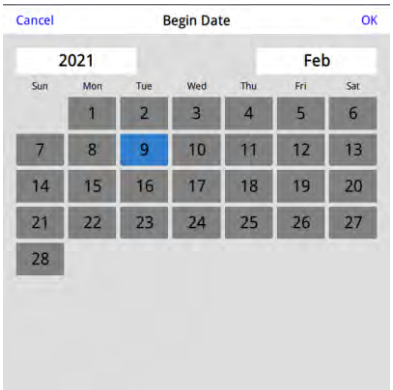
Setting	Description
Mode	The available modes are: [Real-time] and [History]. ● Real-time All the events triggered since HMI starts up are displayed. ● History All events are displayed.
Scrolling control address	If enabled, the value in the designated control address indicates the number of lines to be scrolled down. The minimal value is 0, which indicates the first line.
History Control	When in History mode and if: ● [Enable reading multiple histories] is **not** selected The history files are sorted by date and each file is given an index which is expressed as the number of days before today. 0: today, 1: yesterday, and so on. The history control address is used to specify the history file to be displayed. ● [Enable reading multiple histories] is selected Displays a list of events triggered in multiple days. The display range is formed by two consecutive addresses. Please note that the history control address is defined differently in “Number of days” mode and

	in “Specific days” mode. **Number of days** Suppose the history control address is set to LW-n, then: [LW-n]: The start day index (which is expressed as the number of days before today. 0: today, 1: yesterday, and so on.) [LW-n+1]: The total number of days to display. The days are calculated backward from the start day. **Specific days** Suppose the history control address is set to LW-n, then: [LW-n]: The start day index. 0: today, 1: yesterday, and so on. [LW-n+1]: The end day index. The specified value must be greater than that of the start day index. Example: When start = 4 and end = 7, the specified days will be 4 days before to 7 days before.

For cMT / cMT X Series, all the events that occurred are displayed and updated in real-time.

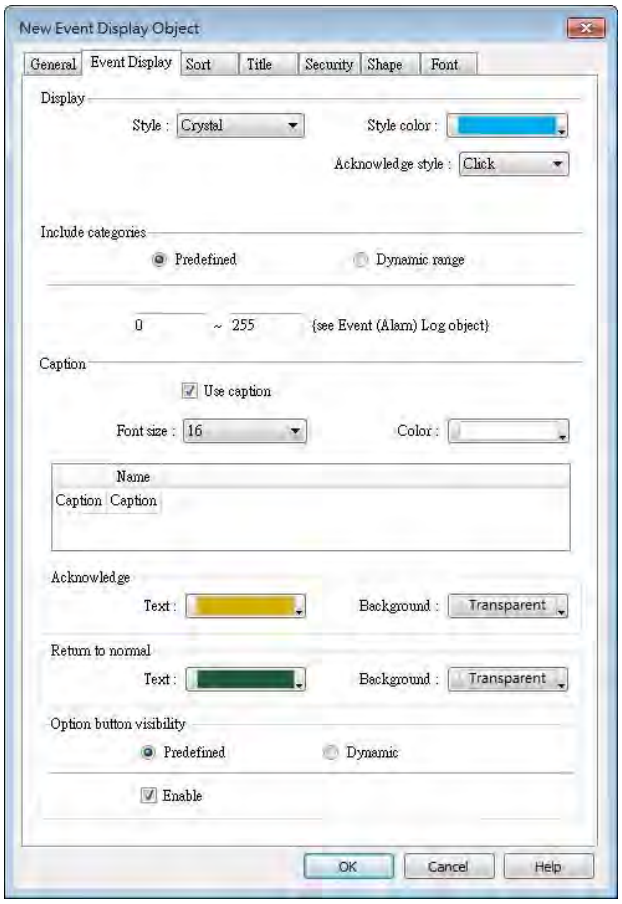
Press the Option Button in the upper-right corner of the object and set the start and end date. If the dates are not set specifically, all the events are displayed.



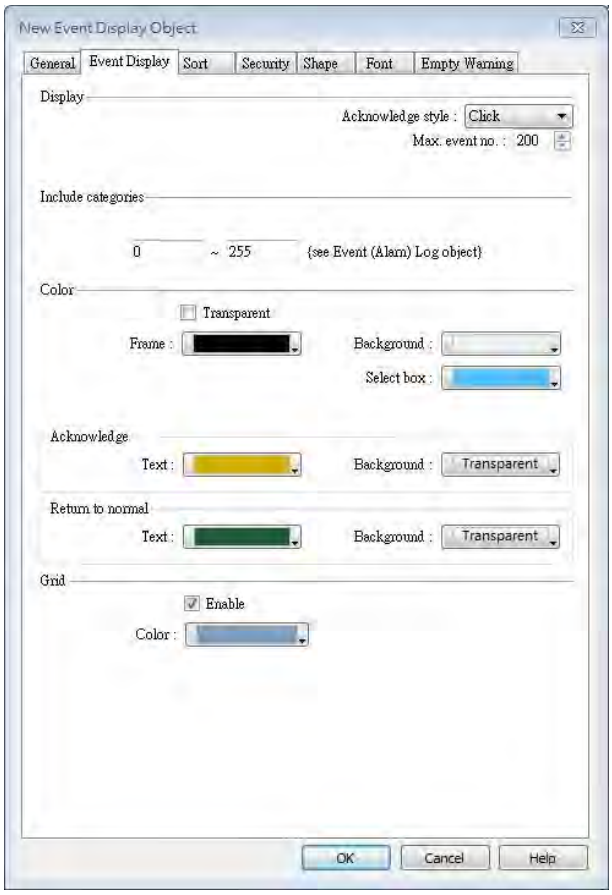


Event Display Tab

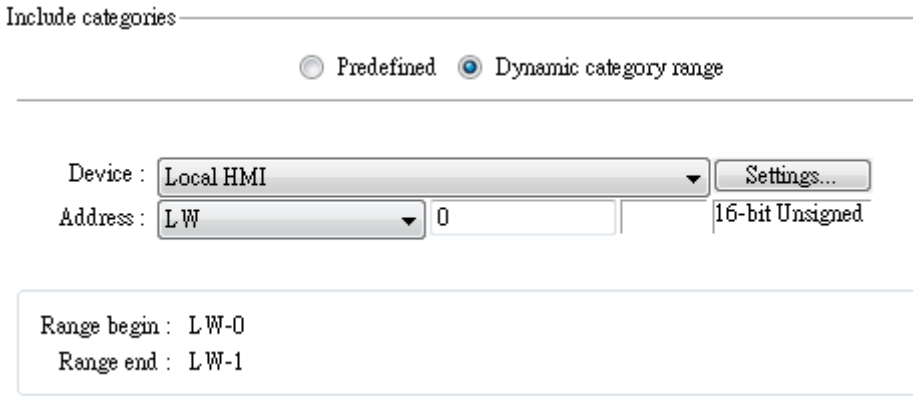
cMT / cMT X Series

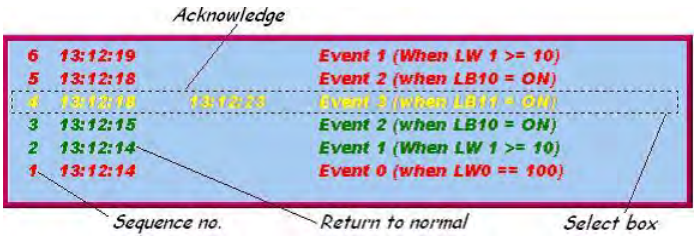


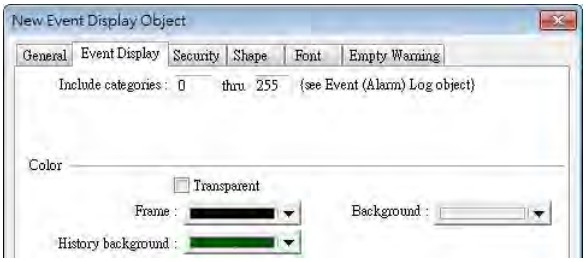
eMT, iE, XE, mTV Series

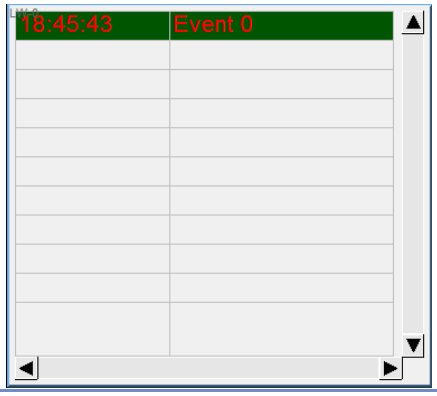


Setting	Description
Include categories	Events in the selected category will be displayed. The categories are set in Event (Alarm) Log object. For example, if the category is set to 2 to 4 here, only events in categories 2, 3, 4 will be displayed. For more information, see “7 Event Log”. **cMT / cMT X Series** Dynamic Category Range is an option available only for cMT Series models. With this option selected, an address can be designated for dynamically selecting categories to be displayed.



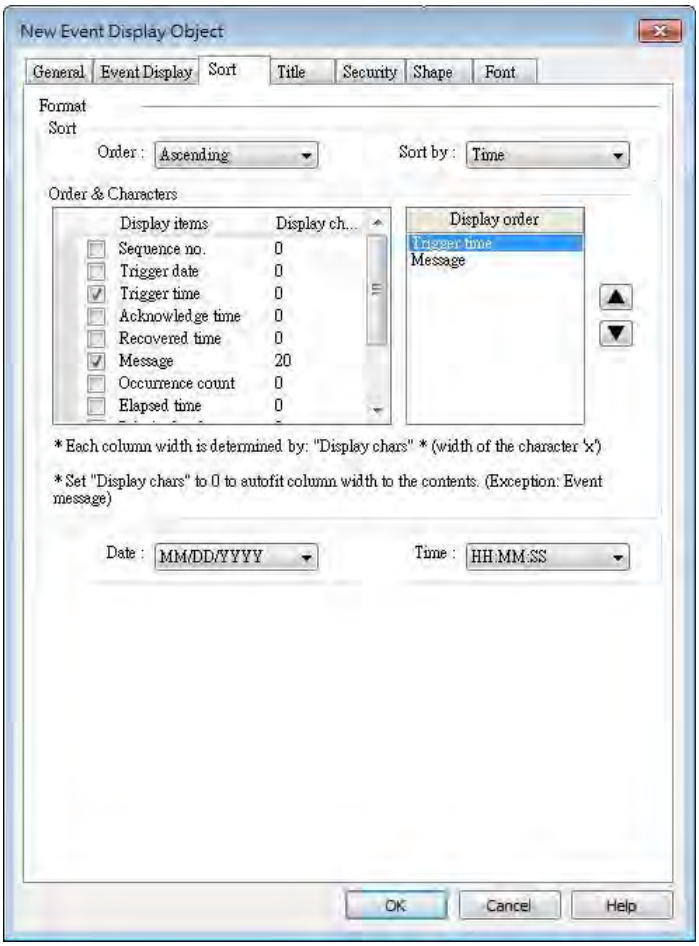
Acknowledge style	Select [Click] or [Double Click] to acknowledge each single event. When an event occurs the user can tap the event line once or twice to acknowledge the new event. When acknowledged, the text color of the event will change to the selected color, and the acknowledge value associated with that event will be sent to the register designated in [Acknowledge address]. If the address is set to LW-100, and the acknowledge value is set to 31, when user acknowledges the event, value 31 is written to LW-100. This can be used in conjunction with Indirect Window object so that when an event is acknowledged, the corresponding message window is displayed.
Max. event no. (eMT, iE, XE, mTV Series)	The maximum number of events to be displayed in this Event Display object. When the number of the displayed events equals to the set maximum number, the new coming event will overwrite the latest event.
Color	Different colors indicate different event states, such as acknowledged, returns to normal, or selected. The system draws a highlight box around the latest selected event.  **History background (eMT, iE, XE, mTV)** When using Event Display object and select History mode, the background color of the history record can be customized.

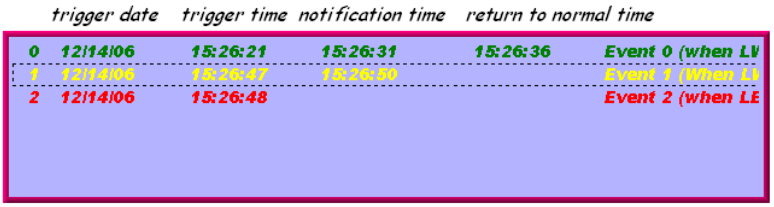




Grid	Displays a grid of rows and columns in the object. The color of the grid lines can be selected. **Auto fit short column** (cMT / cMT X Series Default style) The column width automatically adjusts to the size of the content.
Caption (cMT / cMT X Series)	Available styles are: Default, Crystal, and Flat. With [Use caption] enabled, the font size, color, and name of the caption can be specified for Recipt View object.
Option button visibility (cMT / cMT X Series)	Show or hide the option button . **Predefined** Select [Enable] to show the option button. **Dynamic** Designate a bit address to control option button visibility.

Sort Tab





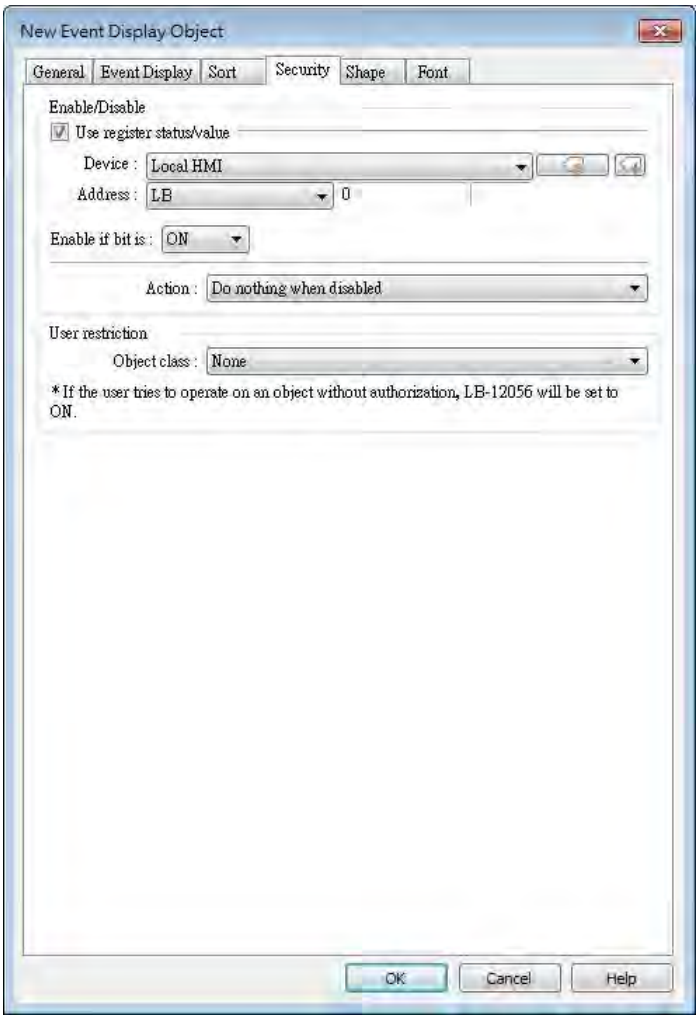
Setting	Description
Order	**Sort by:** **Ascending – Priority Level (cMT / cMT X Series)** Event with highest priority level is placed last in the list (the bottom). **Descending – Priority Level (cMT / cMT X Series)** Event with highest priority level is placed first in the list (the top). **Ascending – Occurrence Count** Event with highest occurrence count is placed last in the list (the bottom). **Descending – Occurrence Count** Event with highest occurrence count is placed first in the list (the

	top). **Ascending - Time** Latest event is placed last in the list (the bottom). **Descending - Time** Latest event is placed first in the list (the top).
Display order	Select the items to be displayed and use the up and down arrow buttons to adjust the display order of the events.
Date	Displays the date tag with each event message. The four formats of date tag: MM/DD/YY、DD/MM/YY、DD/MM/YY、YY/MM/DD
Time	Displays the time tag with each event message. The four formats of time tag: HH:MM:SS、HH:MM、DD:HH:MM、HH

Title Tab

Title settings are available on cMT / cMT X Series. With [Use headers] selected, the text color and column headers can be configured.

Security Tab

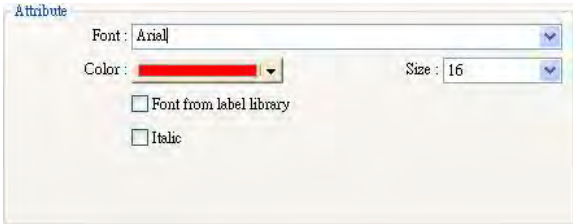


Setting	Description
Interlock	When [Use interlock function] check box is selected, whether the object is operable is determined by the state of a designated Bit address. As shown in the above settings, if LB-0 is ON, the object is operable. **Hide when disabled** When the designated Bit is OFF, the object will be hidden.
User restriction	Set the security class of the object to be operated by an authorized user. **Object class** “None” means any user can operate this object. Only account “admin” can operate “Administrator” object class.

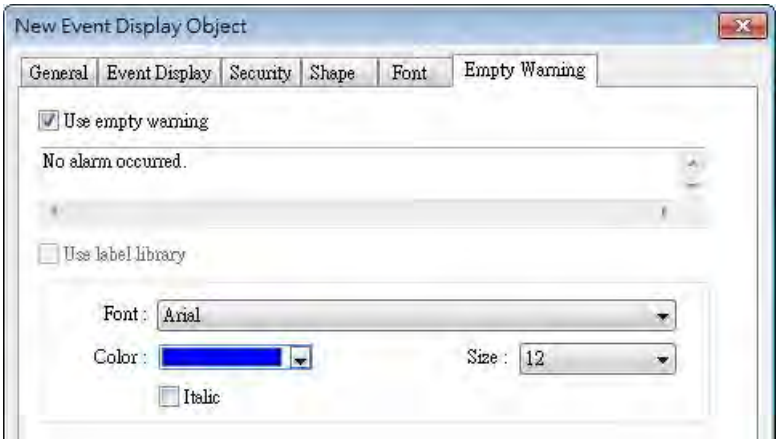
	Make invisible while protected When the user's privilege does not match the object class, the object will be hidden. When this check box is deselected, the unauthorized user can see the Event Display object, but cannot trigger the object or make any change.

Font Tab

In Real-time mode: Users may select Italic font and set the font size. The font is displayed according to the setting in Event Log object.
In History mode: Users may select Italic font and set the font size, font and color, or tick the [Font from label library] check box.



Empty Warning



When [Use empty warning] is enabled, the text displayed when no event has occurred can be specified. This feature is not supported on cMT / cMT X Series models.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.23. Data Transfer

13.23.1. Data Transfer (Per-page)

13.23.1.1. Overview

Data Transfer (Per-page) object can transfer values from the source address to the destination address when the window in which this object is placed opens. The data transfer operation can be activated by manually pressing the object or by detecting the change of state of the designated bit.

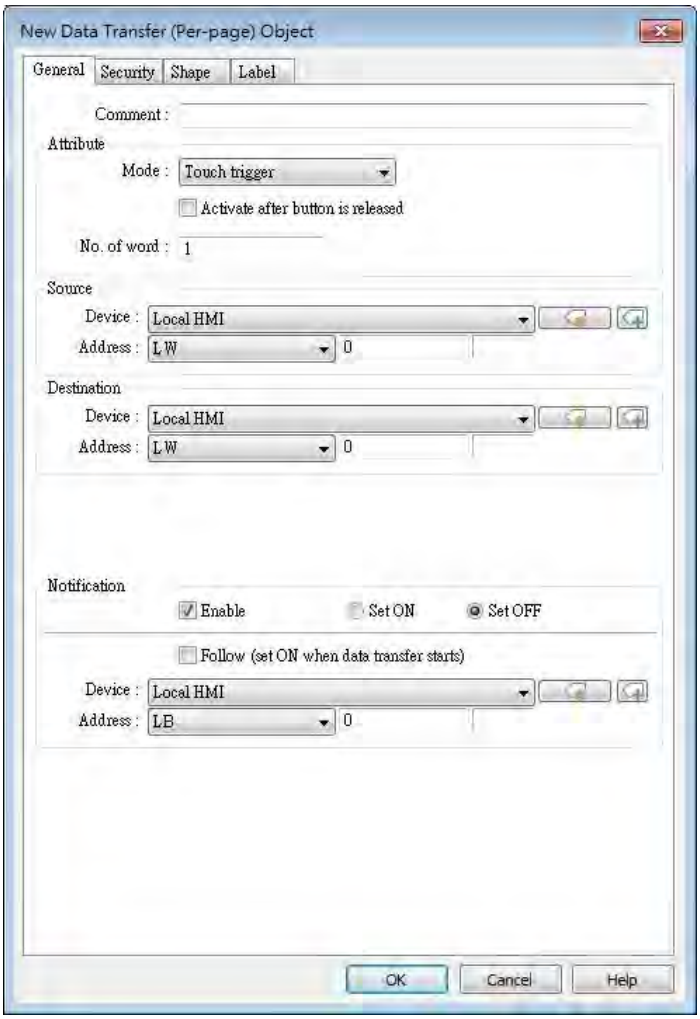
Note

- When Data Transfer (Per-page) object in Bit-Trigger mode is placed in the common window, it activates data transfer operation once the specified condition is met, regardless of the current base window.

13.23.1.2. Configuration

Click [Object] » [Data Transfer] » [Data Transfer (Per-page)] icon on the toolbar to open the property dialog box. Set up the properties, press OK button, and a new Data Transfer (Per-page) object will be created.

General Tab



Setting	Description
Source address	Data Transfer object reads the data from [Source Address].
Destination address	Data Transfer object writes the data to [Destination Address].
Attribute	**No. of word** Enter the number of words to be transferred. Unit: word. **Mode** **Touch trigger** Press the object to activate data transfer operation. **External trigger** The data transfer operation is activated when the state of the designated bit address changes. There is a further selection to make of whether the data transfer

	operation is activated after OFF to ON, ON to OFF transition, or at both of the changes of state.
Trigger address	Specify a bit address for [External trigger] mode.
Notification	When enabled, the system will set the designated address ON or OFF when it's ready for data transfer. **Follow** The notification bit will reset to its original state once the system starts data transfer.

13.23.2. Data Transfer (Global): Time-based

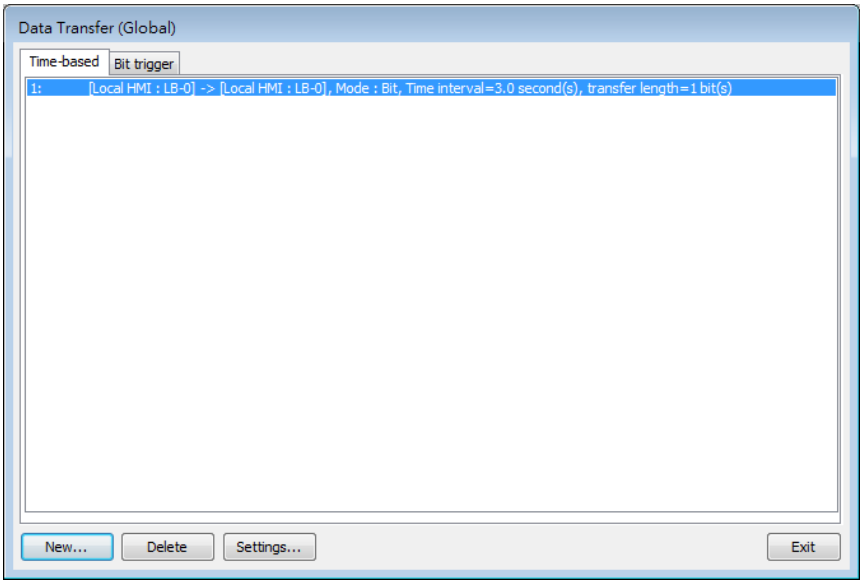
13.23.2.1. Overview

Data Transfer (Global) object in Time-based mode can transfer data from the source address to the destination address based on a time schedule.

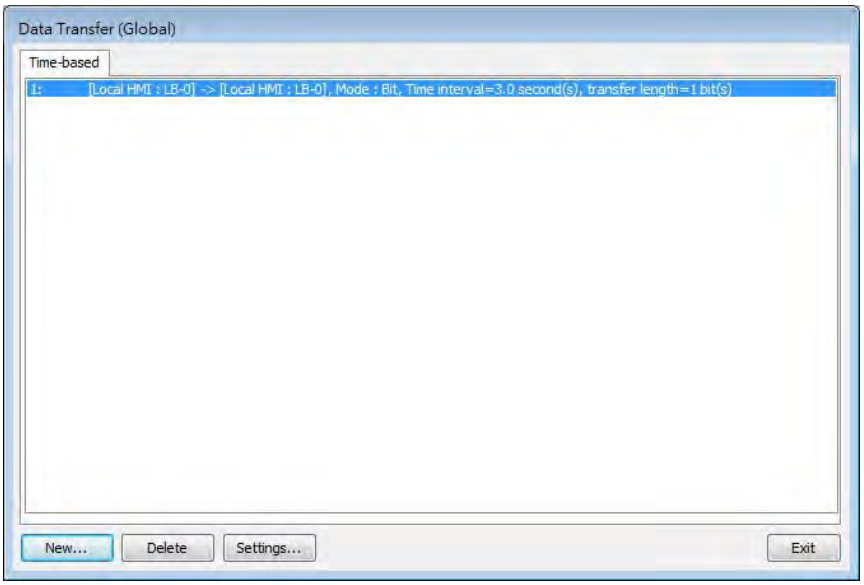
13.23.2.2. Configuration

Click [Object] » [Data Transfer] » [Data Transfer (Global)] icon on the toolbar and open the Time-based tab. Click [New] and set up the properties, press OK button, and a new Data Transfer (Global) object will be created.

cMT / cMT X Series

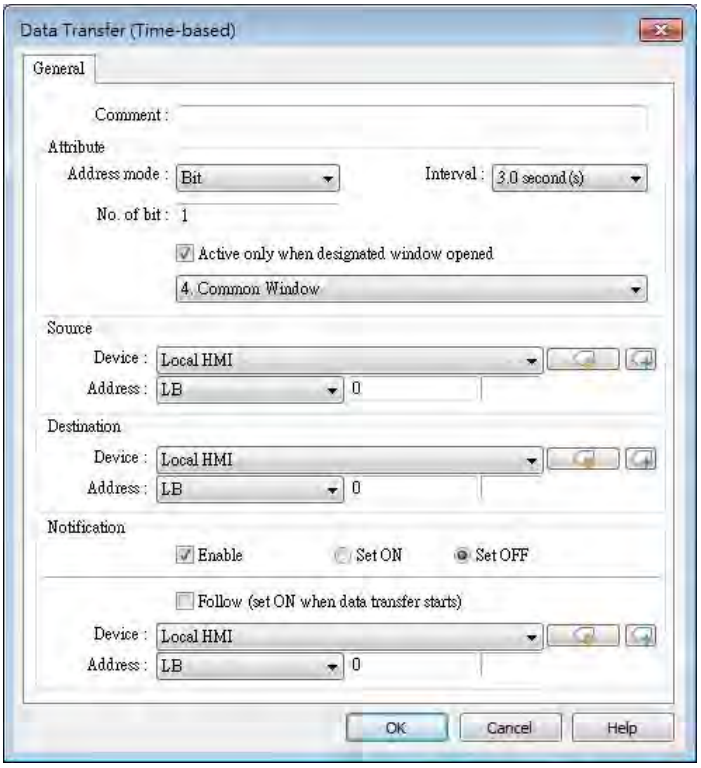


eMT, iE, XE, mTV Series



General Tab

Click the [New] button in the Data Transfer management dialog box.



Setting	Description
Attribute	**Address mode** Select the data type to be transferred from [Bit] or [Word]. **No. of bit /No. of word** When [Bit] is selected in [Address mode], set the number of bits transferred each time when data transfer is triggered. When [Word] is selected in [Address mode], set the number of words transferred each time when data transfer is triggered. **Interval** Select the time interval of data transfer; for example, when 3 seconds is set, the system will transfer data every 3 seconds. Specifying a shorter time interval or a greater amount of data to transfer may decrease overall system performance; therefore, a longer time interval or a smaller amount of data to transfer is recommended. When a short interval is inevitable, please at least set an interval longer than the time needed for data transfer operation; for example, when the data transfer operation takes 2 seconds, the interval must be longer than 2 seconds.

Activate only when designated window opened	Data Transfer object transfers data only when the designated window is opened.
Source address	Data Transfer object reads the data from [Source Address].
Destination address	Data Transfer object writes the data to [Destination Address].
Notification	When enabled, the system will set the designated address ON or OFF when it's ready for data transfer. **Follow** The notification bit will reset to its original state once the system starts transferring data.

13.23.3. Data Transfer (Global): Bit Trigger

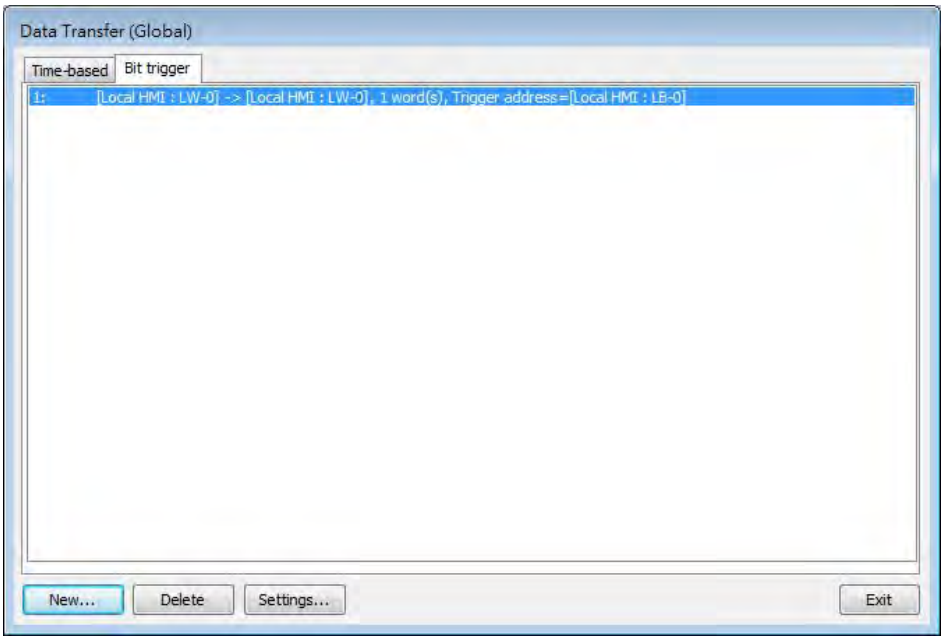
13.23.3.1. Overview

Data Transfer (Global) object in Bit Trigger mode transfers data from the source address to the destination address when HMI detects that the state of the designated bit address changes. This feature is not supported on eMT, iE, XE, mTV models.

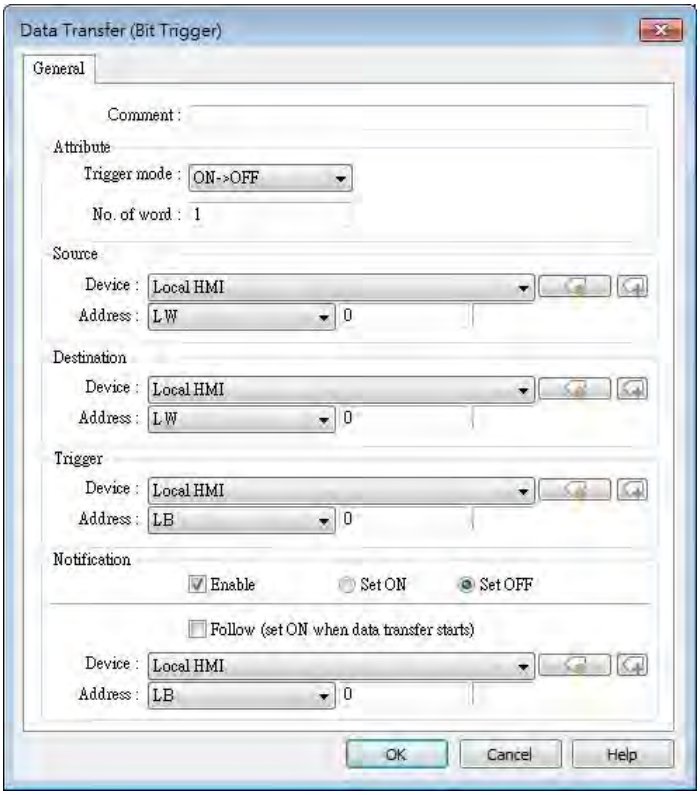
13.23.3.2. Operation

Click [Object] » [Data Transfer] » [Data Transfer (Global)] icon on the toolbar and open the Bit Trigger tab. Click [New] and set up the properties, press OK button, and a new Data Transfer (Global) object will be created.

cMT / cMT X Series



General Tab



Setting	Description
Source address	Data Transfer object reads the data from [Source Address].
Destination address	Data Transfer object writes the data to [Destination Address].
No. of word	Set the number of words transferred each time when data transfer is triggered.
Trigger address	Set the address that controls data transfer and select the trigger mode. **Trigger mode** Trigger data transfer when the state of the designated register changes from OFF to ON, ON to OFF, or at both of the changes of state.
Notification	When enabled, the system will set the designated address ON or OFF when it's ready for data transfer. **Follow** The notification bit will reset to its original state once the system starts data transfer.

13.24. Backup

13.24.1. Overview

Backup (Trigger-based) object and Backup (Global) object can transmit recipe data (RW, RW_A), event log, recipe database, sampling data, and operation log to an external device (SD card, USB disk), in a specified time range or format. For example, when the event log is saved in a SD card, a USB disk can be inserted when HMI power is still ON, and use Backup object to copy the data into USB disk from SD card, and then remove USB disk without turning off HMI power. The data saved in USB disk can be used on PC for analyzing. When the system is backing up, the state of system register [LB-9039] is set ON. With [e-Mail] option, information can be sent to configured email contacts.

Backup (Trigger-based) object is triggered by pressing the object on the screen while Backup (Global) object runs in the background regardless of screens being viewed. Backup (Global) object is available only on cMT / cMT X Series models.

13.24.2. Configuration

On the toolbar click [Object] » [File Operation] to find [Backup (Trigger-based)] and [Backup (Global) objects.

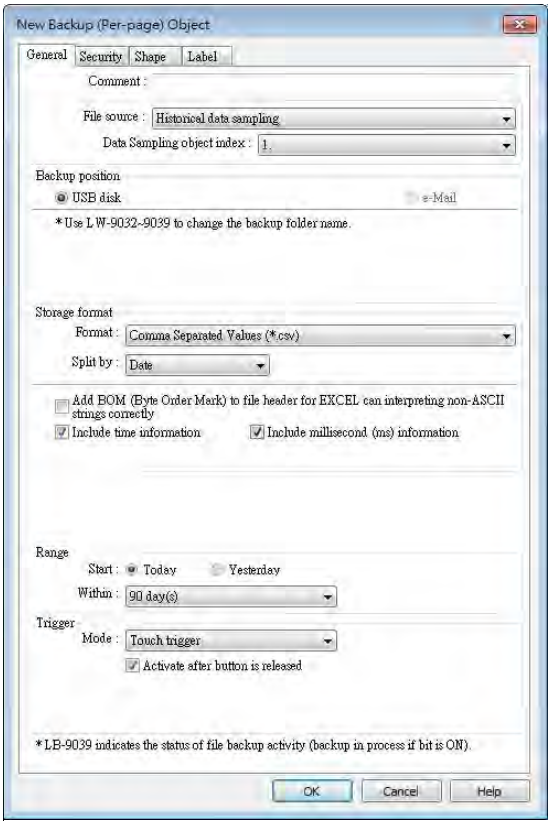
Open [Backup (Trigger-based)] object property dialog box, set up the properties, press OK button, and a new Backup (Trigger-based) object will be created.

When using a cMT / cMT X Series model, [Backup (Global)] object is available. Open [Backup (Global)] object managing dialog box, click [New] to open the object property dialog box, set up the properties, press OK button, and a new Backup (Global) object will be created.

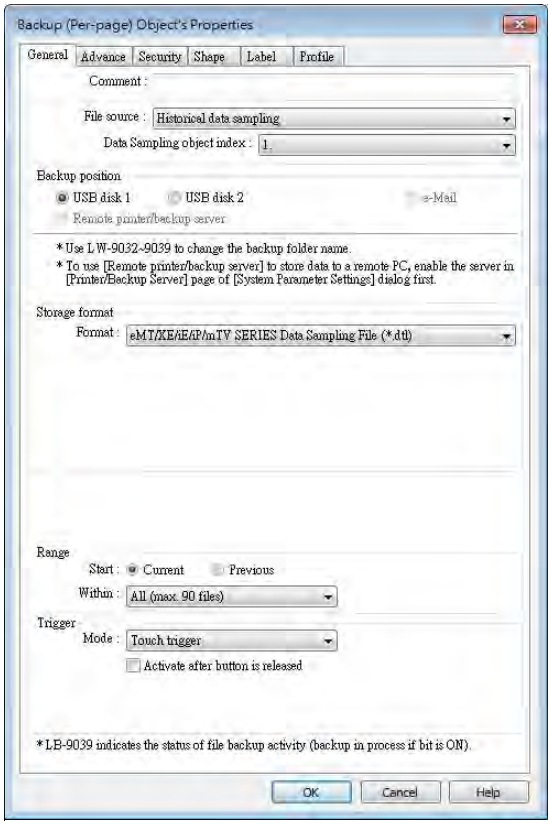
General Tab

Backup (Trigger-based)

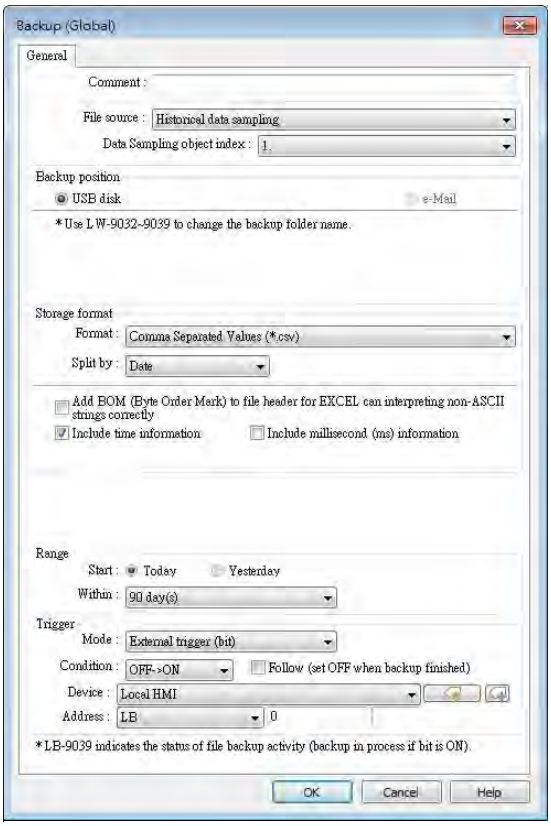
cMT / cMT X Series

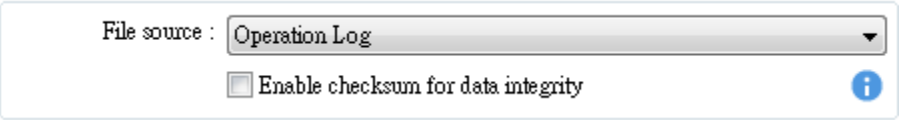


eMT, iE, XE, mTV Series



Backup (Global) cMT / cMT X Series



Setting	Description
Source	**[RW], [RW_A], [Recipe database], [Historical event log], [Historical data sampling], [Operation log], [SQL Query]** Select one from the above for the source. When backing up [Historical data sampling], use [Data Sampling object index] to select the one to back up. Options other than RW, RW_A will be available only when they are used in the project file. **cMT / cMT X Series:**  When Operation Log is selected as file source, [Enable checksum for data integrity] option is provided for using Backup object to generate a backup file with checksum. EasyConverter can then be used to examine data integrity.
Backup position	Select the destination where the source files will be saved to. **SD card / USB disk** The external device connected to HMI. **Remote printer/backup server** (eMT, iE, XE, mTV Series) To select this, enable MT remote printer/backup server at: [Menu] » [Edit] » [System Parameters] » [Printer/Backup Server]. Please note that [Operation log] backup can only be saved to Remote printer/backup server. To save into a SD card or USB drive, please use the control address of Operation Log object. For more information, see “26 EasyPrinter”. **E-mail** To use e-mail, go to [System Parameters] » [e-Mail] tab to configure first. And then go to Backup object » [e-Mail] tab to configure the recipient address, subject, and message.
Save format	Select the desired format to back up the file. **eMT, iE, XE, mTV Series:** • HMI Event Log File (.evt) / HMI Data Log File (.dtl) • Comma Separated Values (.csv) The [Event] column is included in the backup file to indicate the type of the event.

	A	B	C	D	E
1	Event	Category	Date	Time	Message
2	0	1	2013/7/4	16:12:11	Event A
3	2	1	2013/7/4	16:12:12	Event A
4	0	0	2013/7/4	16:12:33	Event B
5	2	0	2013/7/4	16:12:36	Event B
6	0	0	2013/7/4	16:12:37	Event B
7	1	0	2013/7/4	16:12:37	Event B
8	2	0	2013/7/4	16:12:39	Event B
9	0	0	2013/7/4	16:12:40	Event B

0 = Event is triggered

1 = Event is acknowledged

2 = Event returns to normal

EasyConverter can be used to easily convert HMI Event Log File (.evt) and HMI Data Log File (.dtl) to .xls or .csv format.

- SQLite Database File (.db)

cMT / cMT X Series:

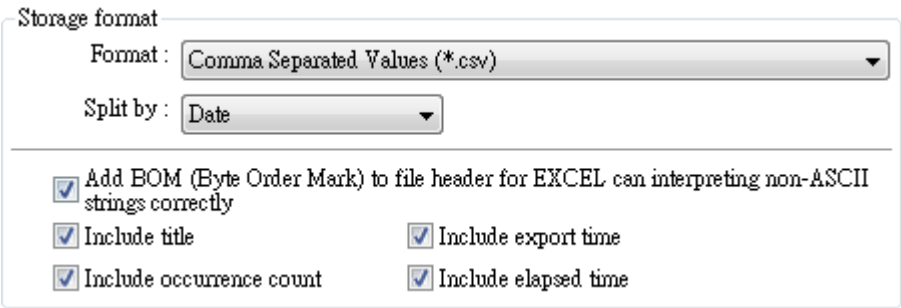
- SQLite Database File (.db) (Only for backup to e-mail)
- Comma Separated Values (.csv)

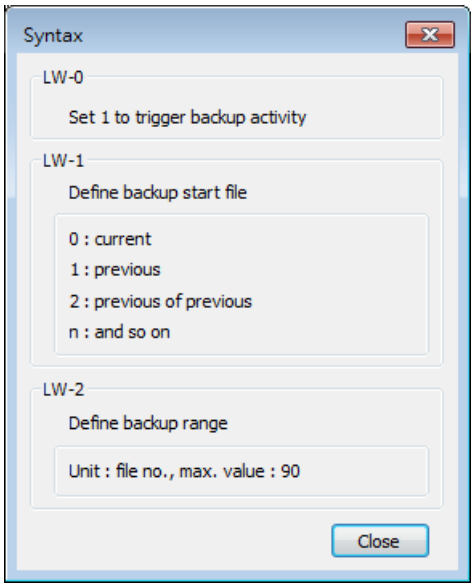
Split by

No split	Backup all historical data to the same file.
Date	Files are separated according to the date of historical data. When [Customized file handling] is used in Data Sampling, the backup files are separated in the same way they are by the rules of customized file handing.
Number of rows	Files are separated according to the number of rows set.

Add BOM (Byte Order Mark) to file header for EXCEL can interpreting non-ASCII strings correctly.

When back up event log in .csv format, open the csv file in EXCEL. The BOM (Byte Order Mark) can be added to the file header so that the .csv file containing non-ASCII strings can directly be opened in EXCEL. The title, export time, occurrence count, and elapsed time can be included or omitted in the backed-up .csv file.

	
Event category range	This groupbox will be available only when backing up a historical event log in CSV format. Two options can be found in this group box: All and Partial. Selecting partial and entering “3, 5, 8” in the field will backup events in categories 3, 5, and 8. Selecting partial and entering “3-8” will backup events in categories 3 to 8.
Range	**Historical data sampling** Select a number of files. For example, if [Start] is set to [Current], and [Within] is set to [5 file(s)], the latest five files in memory will be backed up. **Historical eventlog** Select a number of days. For example, if [Start] is set to [Yesterday], and [Within] is set to [2 day(s)], the files obtained yesterday and the day before yesterday will be backed up. Select [All] to save all files, and the maximum is 90 days.
Trigger	**Mode** There are three ways to activate Backup function. **Touch trigger** Touch the Backup object to activate backup operation. **External trigger (bit)** Register a bit device to trigger the backup operation. Select whether the backup operation is activated after Off to ON, ON to OFF transition, or at both of the changes of state. Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project. **External trigger (word)** Users can specify the number of days to backup data using [Trigger address]. [Trigger address] usage (suppose LW-n is used): LW-n: Will start to back up when the value changes from 0 to 1. LW-n+1: The start date of backup. LW-n+2: The number of days for backup. (The maximum: 90 days).



On cMT-SVR, Backup (Trigger-based) object only supports [Touch trigger] mode but not [External trigger (bit)] and [External trigger (word)] modes.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Trigger address	When the state of the designated register is set ON, the backup operation is activated. When the backup operation is done, the state of the designated register is set OFF.

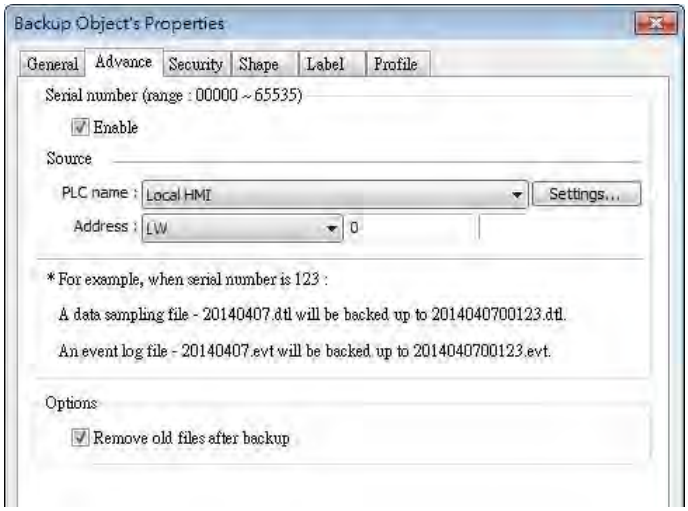
Note

- All history files should have been saved in memory, either HMI memory, USB disk or SD card. Otherwise, the Backup object will not work.
- The maximum number of days for backup is 90 days.
- When saving files to USB disk or SD card, the capacity of a FAT32 folder depends on the length of the file names. Fewer files can be saved when the file names are longer.

For cMT / cMT X Series, see “7 Event Log” and “8 Data Sampling” that explain the mechanism of synchronizing data to external device.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Advance Tab (eMT, iE, XE, mTV)

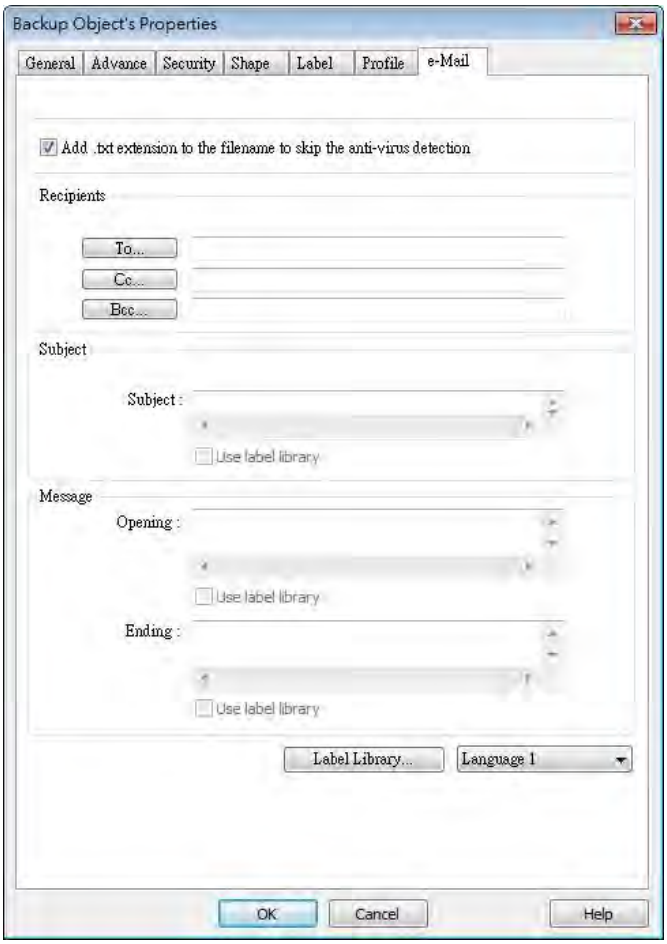


Setting	Description
Serial number	If enabled, when backing up history files, a user-defined, 5-digit serial number can be appended to the end of the file name of the history data backup. The serial number is determined by the value in the designated source address. After backup, the value of this LW address will automatically increment by 1. The range of the serial number is 0~65535 For example, if the serial number is 123, the appended 5 digits will be 00123. A data sampling file -20140407.dtl will be backed up as 2014040700123.dtl. An event log file -20140407.evt will be backed up as 2014040700123.evt.
Options	**Remove old files after backup** If selected, the old history files will be removed after backup.

Note

- cMT / cMT X Series does not support Advance settings.

e-Mail Tab



Setting	Description
Add .txt extension to the filename to skip the anti-virus detection	If selected, when sending backup data as an email attachment, the filename extension .txt will be added to the file name. This prevents the mail server or anti-virus software from blocking emails.
Recipients, Subject, Message	Back up the email address of the recipients, the subject of the email, and the message content.

13.25. Media Player

13.25.1. Overview

Media Player object plays video files on HMI. It can, for example, be useful in providing better instructions onsite to facilitate maintenance.

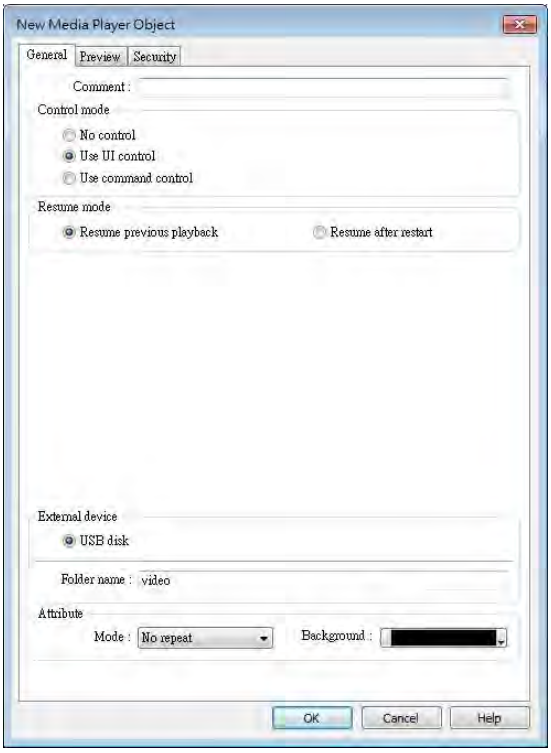
This object does not work remotely on cMT Viewer.

13.25.2. Configuration

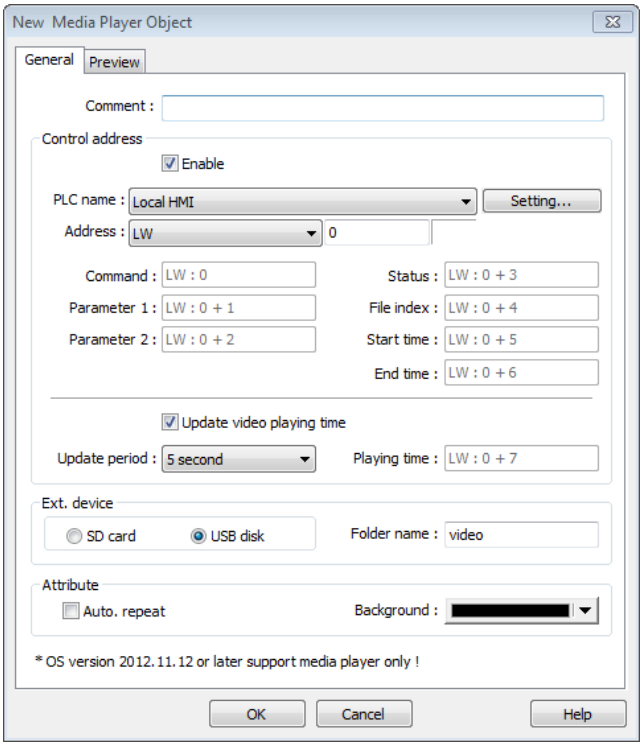
Click [Object] » [Media Player] icon on the toolbar to open a Media Player object property dialog box. Set up the properties, press OK button, and a new Media Player object will be created.

General Tab

cMT / cMT X (excluding cMT-SVR)



eMT, XE



Setting	Description
Control mode	**No control:** Disable manual control by user. Video playback starts as soon as the window opens.

	Use UI control: Use built-in user interface to control media playback. **Use command control:** Use control addresses to control media playback.
Resume mode	Defines where to resume video playback when changing back to a window with media player. Options include **Resume previous playback** and **Resume after restart**.
Control address	● Selected Designate a word register to control the object operations. ● Not selected No manual control. Video will be played automatically when the designated window opens. **Command** (control address + 0) Enter a value in the Command register to designate which action is executed. **Parameter 1** (control address + 1) Enter a value in Parameter 1 associated with each command action. **Parameter 2** (control address + 2) Enter a value in Parameter 2 associated with each command action. **Status** (control address + 3) Indicates the status or errors. **File index** (control address + 4) The file number in the designated folder. It is recommended to file the video name with a number. **Start time** (control address + 5) The start time of the video (second). 0, normally. **End time** (control address + 6) The end time of the video (second). (The time length of the video) **Update video playing time** If enabled, the elapsed playing time of video will be written into [Playing time] register at a rate set by [Update period] in seconds. **Update period** Update period of [Playing time], range from 1 to 60 (second). **Playing time** (control address + 7) The elapsed playing time of video (Second). Normally between start time and end time.
Ext. device	Play video files in SD card / USB disk. **Folder name** The folder name of video files stored in SD card or USB disk. Files must be stored in root directory. Subdirectories won't be accepted. (For example, "example\ex" is an invalid directory.) [Folder name] cannot be empty, must be alpha-numeric, and all in

	ASCII character.
Attribute	**Auto. repeat** When finish playing all the video files, replay from the first file. Ex: Video 1 > Video 2 > Video 1 > Video 2 **Background** The background color of the object.

Note

- The data format for control address is 16-bit Unsigned or 16-bit Signed. If using 32-bit Unsigned or 32-bit Signed, only the previous 16 bits will be effective.

Control command

The following are the settings of different commands.

- **Play index file**

  [Command] = 1

  [Parameter 1] = file index

  [Parameter 2] = ignore (set 0)

Note

- Files are stored with file names in ascending order.
- If the file cannot be found, [Status] bit 8 is set ON.
- Please stop the playing video before switching to another.

- **Play previous file**

  [Command] = 2

  [Parameter 1] = ignore (set 0)

  [Parameter 2] = ignore (set 0)

Note

- If [File index] is zero, the same file is replayed.
- If the file cannot be found, [Status] bit 8 is set ON.

- **Play next file**

  [Command] = 3

  [Parameter 1] = ignore (set 0)

  [Parameter 2] = ignore (set 0)

- If there are no more files, the index 0 file is played.
- If the file cannot be found, [Status] bit 8 is set ON.

- **Pause / Play Switch**

[Command] = 4
[Parameter 1] = ignore (set 0)
[Parameter 2] = ignore (set 0)

- **Stop playing and close file**
  [Command] = 5
  [Parameter 1] = ignore (set 0)
  [Parameter 2] = ignore (set 0)
- **Start playing from the designated time**
  [Command] = 6
  [Parameter 1] = target time (second)
  [Parameter 2] = ignore (set 0)

Note

- Parameter 1 (target time) must be less than the ending of time or it plays the last second.

- **Forward**
  [Command] = 7
  [Parameter 1] = target time (second)
  [Parameter 2] = ignore (set 0)

Note

- Going forward to the designated second in [Parameter 1]. If the video is paused, the forwarding action will be started by playing.
- When the designed time is later than the end time, it plays the last second.

- **Backward**
  [Command] = 8
  [Parameter 1] = target time (second)
  [Parameter 2] = ignore (set 0)

Note

- Going Backward to the designated second in [Parameter 1], if the video is paused, the backward action will be started by playing.
- When the designed time is earlier than the beginning time, it plays from beginning.

- **Adjust volume**
  [Command] = 9
  [Parameter 1] = volume (0 ~ 128)

[Parameter 2] = ignore (set 0)

Note

- Default volume is 128.

- **Set video display size**

  [Command] = 10

  [Parameter 1] = display size (0 ~ 16)

  [Parameter 2] = ignore (set 0)

Note

- [Parameter 1 = 0] ： Fit video image to object size.
- [Parameter 1 = 1 ~ 16]：Magnification from 25% ~ 400% in 25% increments where 1 = 25%, 2 = 50%, 3 = 75% and so on.

- **Status (control address + 3)**

When playing a video the system sets [File Open (bit00)] and [File Playing (bit01)] to ON. If the file cannot be found, or an invalid command is entered, the Command Error bit 08 is set ON. If the file format is not supported, or a disk I/O error occurs, during playback, the File Error bit 09 is set ON.

15	09	08		02	01	00	bit
Reserved (all 0)	0	0			0	0	

00: File Opened / Closed (0 = closed, 1 = opened)

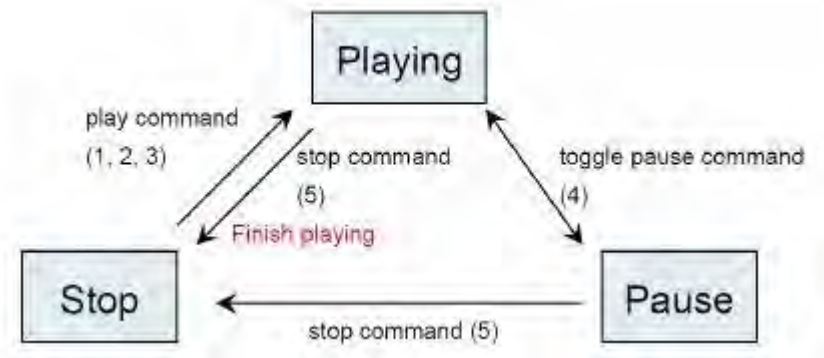
01: File Playing (0 = not playing, 1 = playing)

08: Command Error (0 = accepted, 1 = incorrect)

09: File Error (0 = accepted, 1 = incorrect)

Note

- The figure shows the status value associated with each state:
  Stop = 0, Pause = 1, Playing = 3



- [Command], [Parameter 1], and [Parameter 2] are write addresses. All others are read

only.

Preview Tab

Users can test whether the video format is supported by using the preview function.

Setting	Description
Forward << / Backward >>	Go forward or backward of the video. (in minutes)
Play / Pause	Select to start playing video or pausing.
Stop	Stop playing and close the video file. To test another video, please stop playing the current video first.
Load	Select a video to preview.

Note

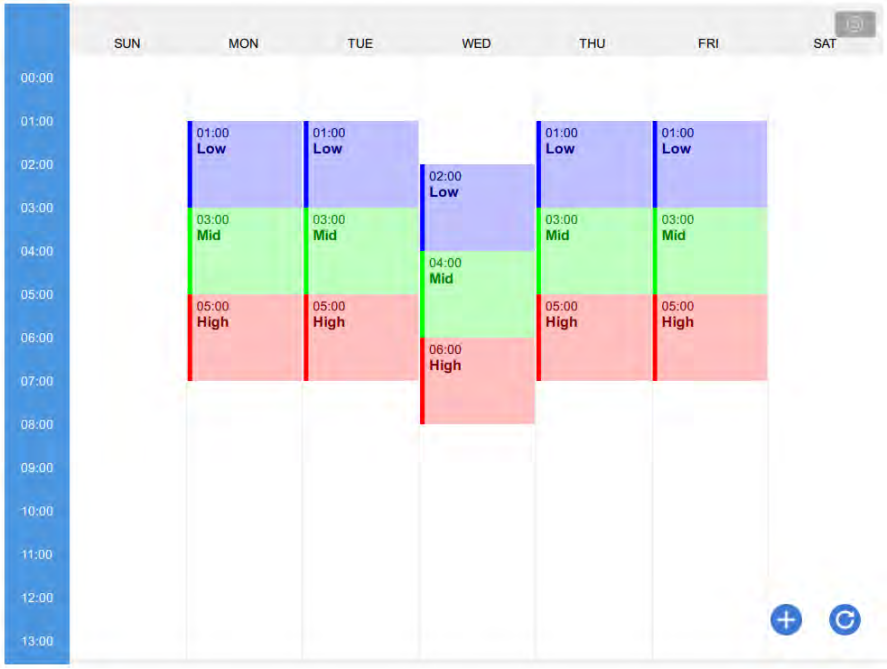
- Only one video file can be played at one time.
- If [control address] is not enabled and [Auto. repeat] is not selected, after finish playing the first file, the system will stop playing and close the video file.
- If [control address] is not enabled, the system will find the first file in the designated folder and start to play (in ascending order of the file name).
- If the file can be previewed, the format is supported. If the video image quality is poor, please adjust the resolution.
- The supported formats: mpeg4, xvid, flv…etc.
- The recommended frame rate for videos is 30fps.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.26. BACnet Schedule

13.26.1. Overview

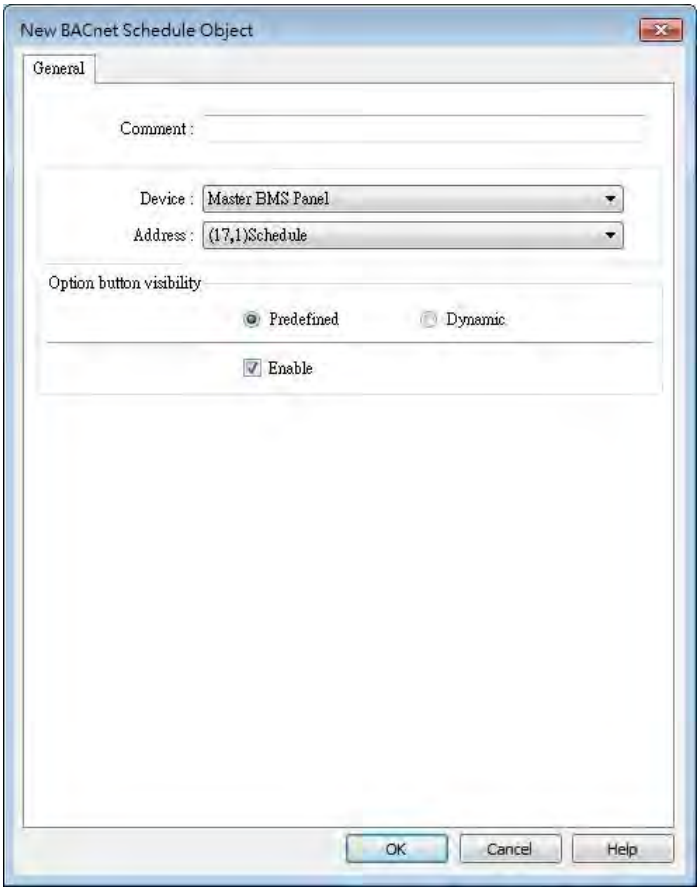
BACnet Schedule helps visualize and access schedule data (Schedule) in BACnet IP controller with a graphical UI. This feature is only applicable for cMT / cMT X Series HMI.



13.26.2. Configuration

Add BACnet/IP driver into the device list in System Parameter Settings and check whether a Scudule is contained in the scanned BACnet/IP device or the imported tag file.
Click [Object] » [Time Related] »[BACnet Schedule] icon on the toolbar to open the BACnet Schedule property dialog box. Configure the parameters and then click OK, a BACnet Schedule object is created.

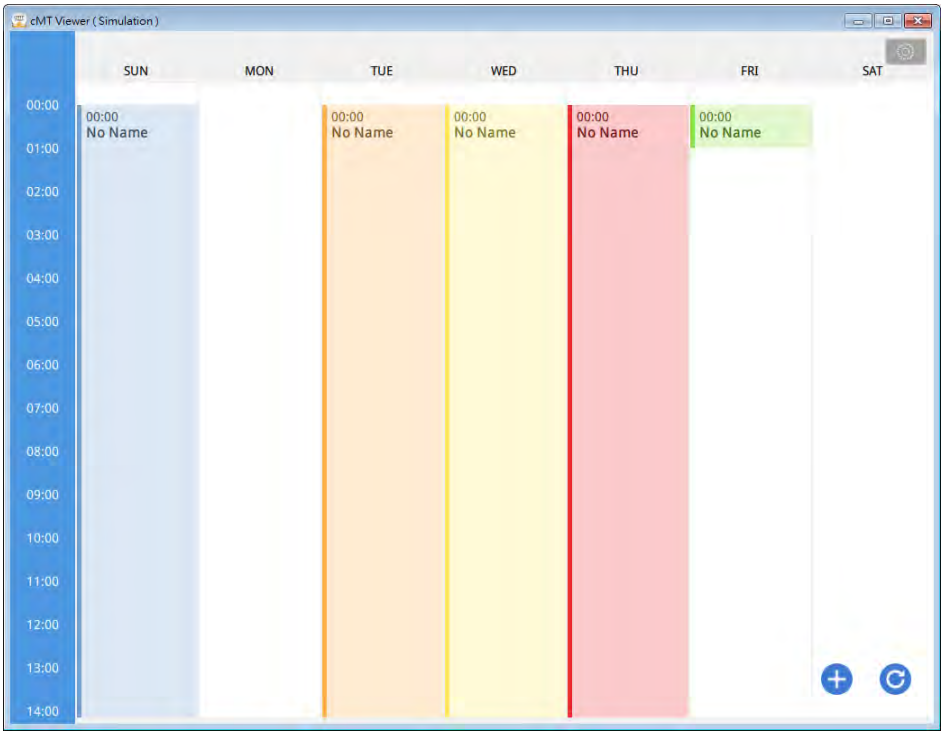
General Tab



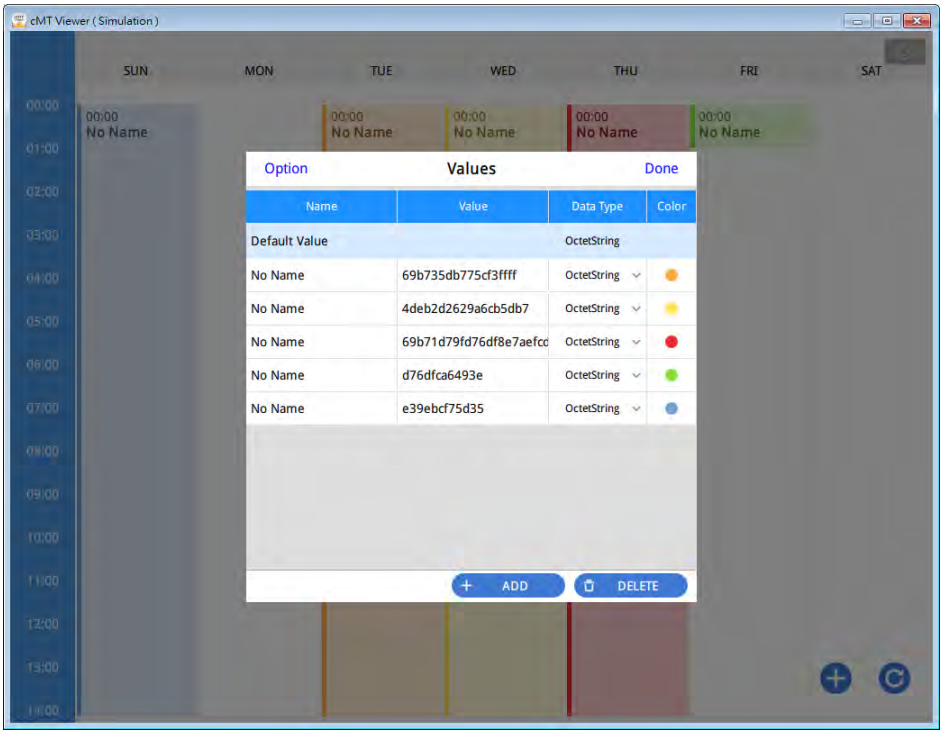
Setting	Description
Device	Select the BACnet/IP device to be read.
Address	Select the address tag of the Schedule to be read.
Option button visibility (cMT / cMT X Series)	Show or hide the option button . **Predefined** Select [Enable] to show the option button. **Dynamic** Designate a bit address to control option button visibility.

Example 1

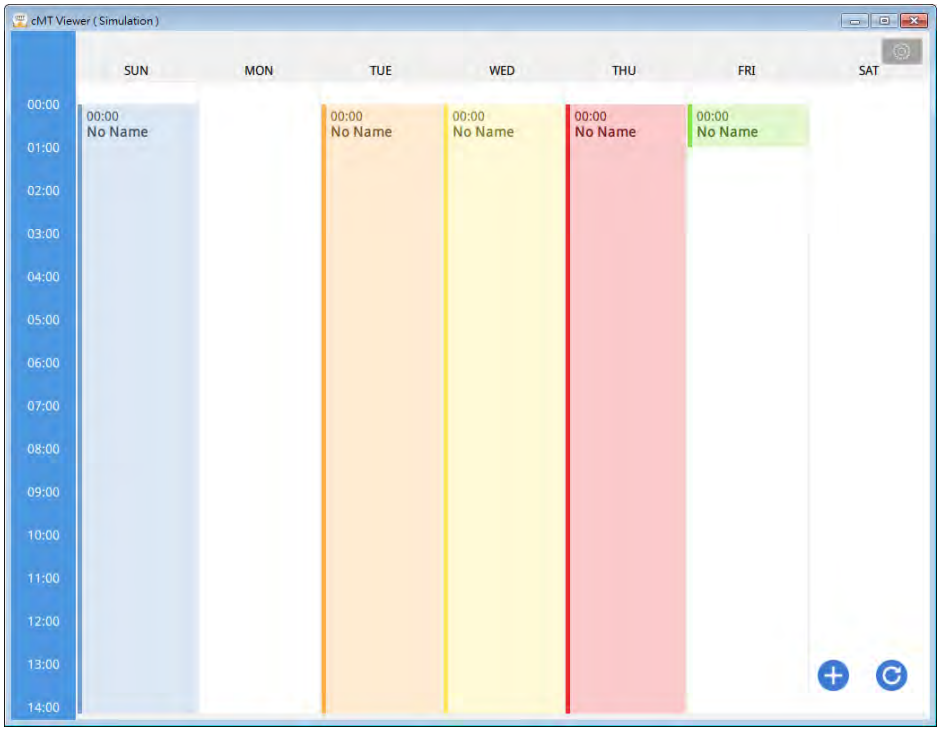
After downloading the project to HMI, the HMI will automatically read the schedule settings in the BACnet IP device as shown below.



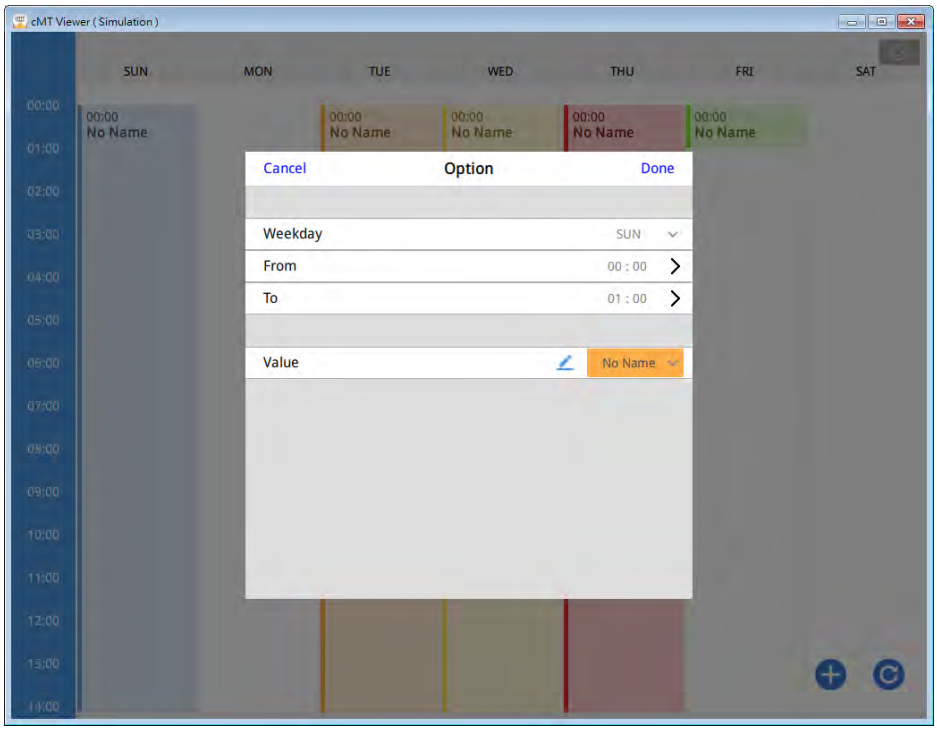
Click and then select Value Editor to configure the Values table containing these columns: Name, Value, Data Type, and Color, as shown below.



Click to add a schedule setting. Click to refresh schedule data by reading from the BACnet device.



As shown in the following screenshot, in a schedule settings window, the Weekday, start time (From), end time (To), and Value should be configured. Value can be selected from the value table. (Clicking open Value Editor for editing the value table as well.)



After the time settings are completed, the values will be written to the corresponding schedule tags in BACnet device.

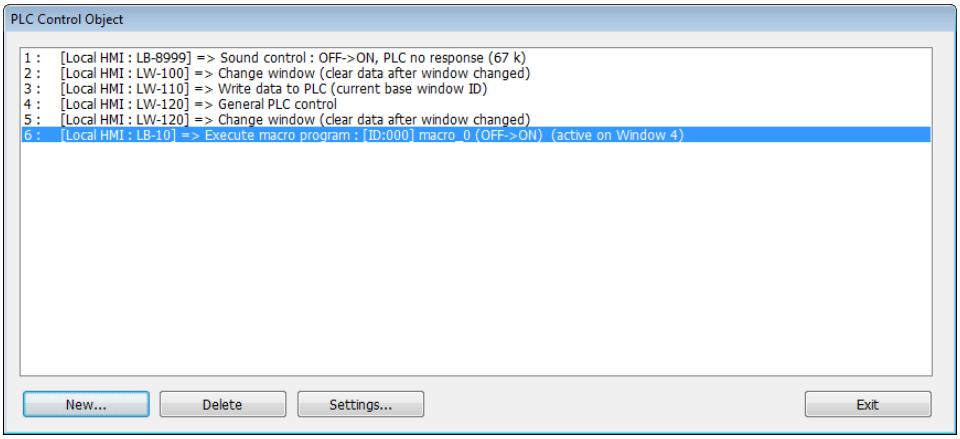
13.27. PLC Control

13.27.1. Overview

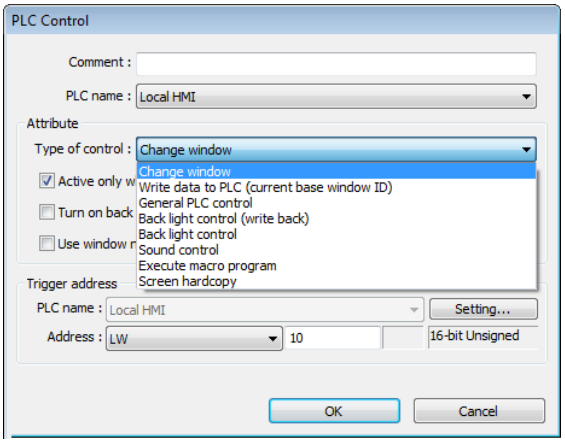
PLC Control object can execute commands when it is triggered.

13.27.2. Configuration

Click [Object] » [PLC Control] icon on the toolbar to open the PLC Control Object management dialog box. To add a PLC Control object, click [New], set up the properties, press OK button and a new PLC Control object will be created.



Click [New] and the following dialog box appears. See “13.27.2.1 Type of Control”.



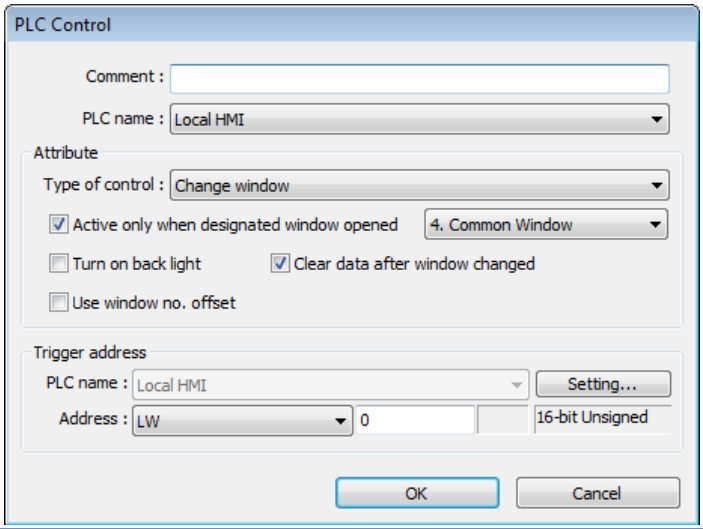
Note

- The [General PLC Control] option is not available for cMT / cMT X Series.
- Triggering [Screen hardcopy] in cMT Viewer will result in files being saved into the USB disk / SD card connected to the cMT / cMT X HMI. Likewise, the [Back light control] option in cMT Viewer will control the backlight of the HMI itself.

- Selecting [Change window] or [General PLC Control] as [Type of control] will require more than one trigger word (consecutive). Using a user-defined tag PLC will require declaring an array datatype.

13.27.2.1. Type of Control

- **Change window**



Setting	Description
Active only when designated window opened	Allow this operation only if a particular screen is displayed.
Turn on back light	The backlight is turned ON when the window is changed.
Clear data after window changed	Reset the value at trigger address to zero when the window object is changed. If [Use window no. offset] is selected, this option will only show when a negative offset is used.
Use window no. offset	Select the check box and select a window offset, the new window no. to change to will be the value in [Trigger address] plus the offset. For example, if [Trigger address] is LW-0 and offset is set to -10. When the value in LW-0 is 25, the system will change to window no. 15 (25-10=15). The range of the offset is -1024 to 1024.

Note

- If [LB-9017] is set ON, the write-back function will be disabled, the new window number is not written back into a designated address.

Place a valid window number in the designated trigger address to change the base screen to the new window number. The new window number is written back into the designated

address.

For example, if current window is window no. 10, and [Trigger address] is set to LW-0, When LW-0 is changed to 11, the system will change the current window to window no. 11, and then write 11 to LW-1.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

When the window is changed, the new window number is written back into the address that is calculated by [Trigger address] and the data format, as shown in following table.

Data Format	Trigger address	Write address
16-bit BCD	Address	Address + 1
32-bit BCD	Address	Address + 2
16-bit Unsigned	Address	Address + 1
16-bit Signed	Address	Address + 1
32-bit Unsigned	Address	Address + 2
32-bit Signed	Address	Address + 2
64-bit Unsigned	Address	Address + 4
64-bit Signed	Address	Address + 4

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

- **Write data to PLC (current base window)**

Each time the base window is changed, the new window number will be written into the [Trigger address]. If [Use window no. offset] is selected, the window number of the base window plus the window number offset will be written into the [Trigger address].

- **General PLC Control (eMT, iE, XE, mTV)**

Transfer word data blocks from PLC to HMI, and vise-versa, and the transfer direction is controlled by the value in the [Trigger address].

Value in [Trigger address]	Action
1	Transfer data from PLC register → HMI RW register
2	Transfer data from PLC register → HMI LW register
3	Transfer data from HMI RW register → PLC register
4	Transfer data from HMI LW register → PLC register

Four consecutive word registers are used as described in the following table:

Address	Purpose	Description
[Trigger address]	Determine the direction of data transfer	The valid values are listed in the above table. When a new control code is written into the register, HMI will start to transfer. After data transfer is finished, the value will be set to 0.
[Trigger address] +1	The size of data to transfer.	The unit is word.
[Trigger address] +2	Offset to the start address of PLC register	Assume the value is "n", where n is an arbitrary number, the start address of PLC register is [Trigger address + 4 + n]. Take an OMRON PLC as an example: If [Trigger address] uses DM-100, [Trigger address + 2] will be DM-102. If the value in DM-102 is 5, the start address of data source would be DM-109 (100 + 4 + 5 = 109).
[Trigger address] +3	Offset to the start address of LW or RW memory in HMI	Take OMRON PLC as an example: If set [Trigger address] to DM-100, [Trigger address + 3] will be DM-103. If the value in DM-103 is 100, the start address of memory in HMI is RW-100 or LW-100.

Example 1

To use PLC Control object to transfer 16 words data in OMRON PLC, starting from address DM-100, to the HMI address, starting from RW-200. The setting is shown below:

1. Firstly, create a PLC Control object, set [Type of control] to [General PLC control], and set [Trigger address] to DM-10, that is, to use the four sequential registers start from DM-10 to control data transfer.
2. Confirm the data size and the offset addresses.
   Set DM-11 to 16, since the number of words to transfer is 16 words.
   Set DM-12 to 86, which indicates the address of data source is DM-100 (100=10+4+86).

Set DM-13 to 200, which indicates the destination address is RW-200.

3. Set DM-10 according to the direction of data transfer.
   If set DM-10 to 1, the data will be transferred from PLC to HMI RW register,
   If set DM-10 to 3, the data will be transferred from HMI RW register to PLC.

- **Back light control (write back)**

When [Trigger address] is turned ON, HMI backlight will be turned ON/OFF and [Trigger address] will be turned OFF. Any touch on the screen will turn the backlight on.

- **Back light control**

When [Trigger address] is turned ON, HMI backlight will turn ON/OFF and the state of [Trigger address] will not be changed.

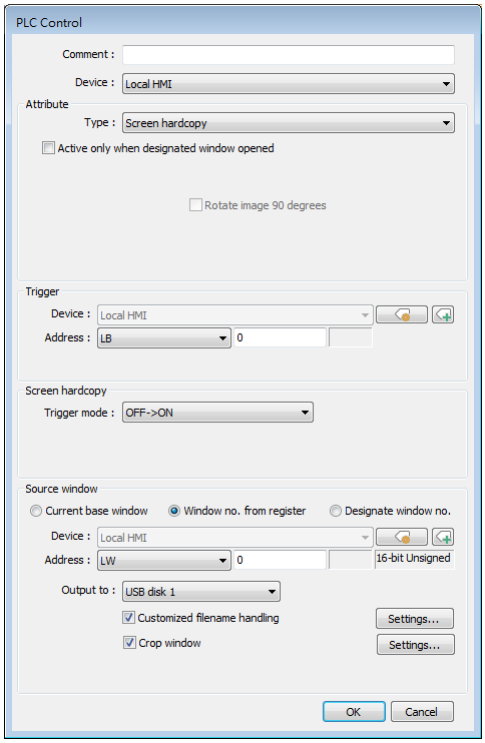
- **Sound control**

When the state of the designated [Trigger address] changes, the HMI will play the sound selected from the sound library. There is a further selection determines whether the sound is played after Off to ON, ON to OFF transition, or at both of the changes of state.

- **Execute macro program**

Select a pre-defined Macro from the drop-down list. When the state of the designated [Trigger address] changes, the selected Macro is executed. There is a further selection determines whether the Macro is executed after Off to ON, ON to OFF transition, or at both of the changes of state. If select [Always active when ON], the macro will be executed repeatedly. (The shortest time interval between runs is 0.5 second.)

- **Screen hardcopy**



When the state of the designated [Trigger address] changes, print the selected screen. If select [SD card] or [USB disk] as [Printer], a “hardcopy” folder will be generated in the selected external device for saving the printed screen in JPG format. The name of the JPG files starts from yyyymmdd_0000.

To print the screen using a printer, go to System Parameter Settings » Model tab and set the printer.

To print the screen using a remote printer, go to System Parameter Settings » Printer/Backup Server tab and configure the parameters.

There are three options to specify the source window for hardcopy:

Current base window

Print the base window currently opened.

Window no. from register

Print the window designated by the value in a designated word address.

Designate window no.

Directly select a window to be printed.

Customized File Handling

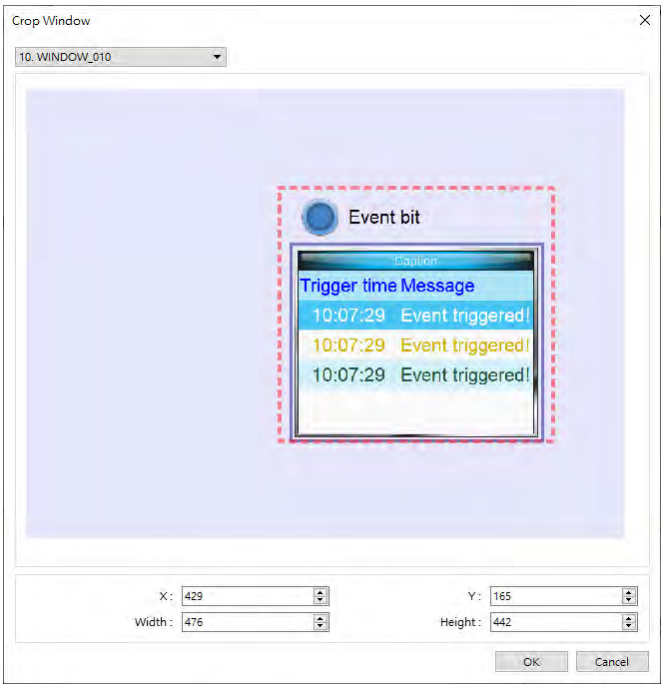
This feature can be used to customize naming of the folders and the JPG files.

Setting	Description
Folder name	The folder name can be an alphanumeric name, and certain half-width symbols are allowed: !@#$%^&()_+{}`-=;',. The folder name can also be specified by a naming syntax. **Dynamic format** The folder names can be set by a designated word address, or by a naming syntax indicating the current system time. The syntax can be specified by selecting time buttons or entering the syntax in Format field. The length limit is from 1 to 25. **Note:** Up to 10 layers of folders can be created. The exceeding layers will be ignored.
File name	The way to specify a file name is similar to specifying a folder name. **Note:** If the file name already exists, the system will add “_0001” to the file name as a serial number. The numbers of the later files add up until “_9999”. The files after “_9999” will be ignored. For example, if the three file names exist: “A.jpg”, “A_0001.jpg”, “A_0003.jpg”. When trigger screen

	hardcopy with the same file name, the coming files will be named in this order: “A_0002.jpg”, “A_0004.jpg”, ”A_0005.jpg”, and so on.

Crop window

With this option selected, taking a cropped screenshot is possible.



Note

- A background printing procedure is performed when the printed window is not the current base window.
- If the hard-copied window is a background window, its [Direct Window] and [Indirect Window] objects will not be printed.
- When using a dynamic format name, the system will use a “_” sign as a substitute for invalid symbols.
- When using a dynamic format name, if screen hardcopy is triggered without specifying the format first, the system will save the file in the default directory, which is: hardcopy\yymmdd_0000.JPG
- On cMT / cMT X models, a triggered bit address is reset to its original state automatically, whereas on eMT/iP/iE/mTV models, the state of a triggered bit address is retained.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

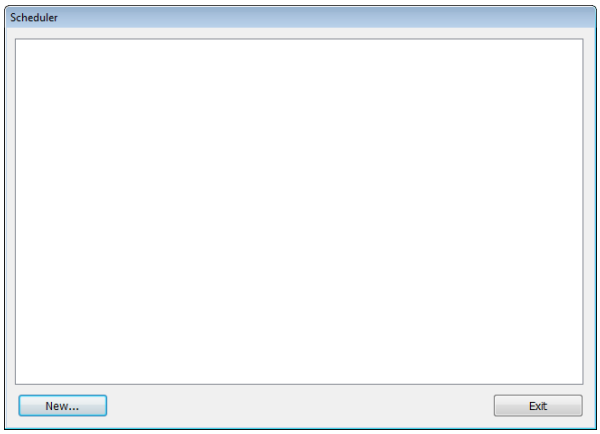
13.28. Scheduler

13.28.1. Overview

Scheduler object turns bits ON/OFF, or writes values to word registers at designated start times. It works on a weekly basis.

13.28.2. Configuration

Click [Object] » [Time-related] » [Scheduler] icon on the toolbar to open the Scheduler management dialog box, click [New] to open the Scheduler property setting dialog box.



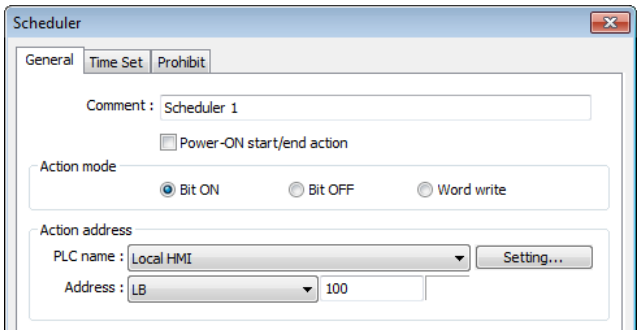
The following two demonstrations explain the usage of Scheduler.

Example 1

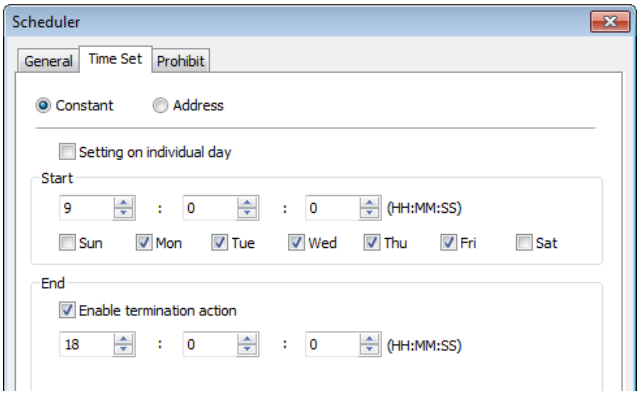
A motor is scheduled to power - ON at 9:00 and power – OFF at 18:00, Monday to Friday.

We are using LB-100 to control the motor state. LB-100 will be set ON at 9:00 and OFF at 18:00.

1. Click the Scheduler icon on the toolbar to open the Scheduler management dialog box, click [New].
2. In [General] tab, select [Bit ON] in [Action mode] and set [Action address] to LB-100.



3. In [Time Set] tab, select [Constant].

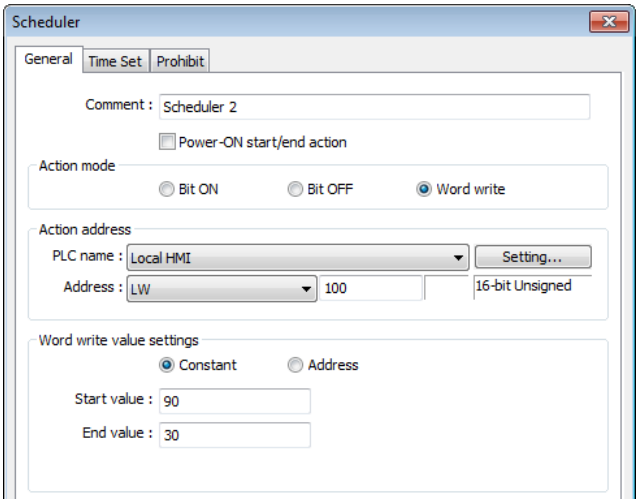


4. Enter [Start] time as 9:00:00 and select Monday to Friday. Do not select [Setting on individual day].
5. Enter [End] time as 18:00:00 and select [Enable termination action] check box.
6. Click [OK], a new Scheduler object will be created on the [Scheduler] list.

Example 2

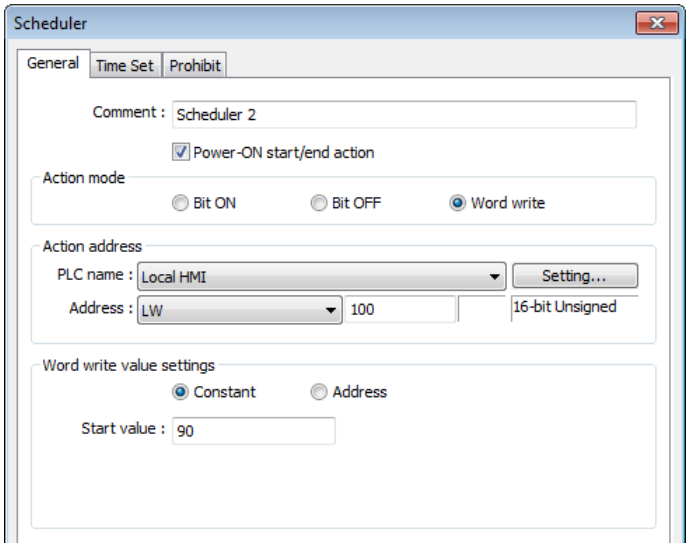
A thermal heater is scheduled to heat up to 90°C at 08:00 and cool down to 30°C at 17:00, Monday to Friday. LW-100 is used to store the set point value.

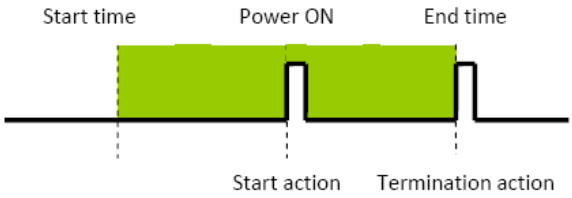
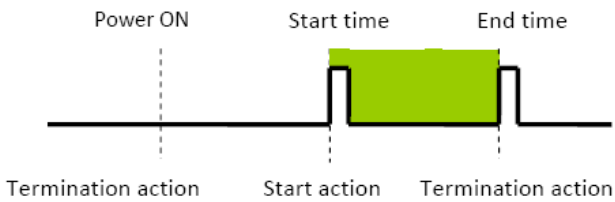
1. Click the Scheduler icon on the toolbar to open the Scheduler management dialog box, click [New].
2. In [General] tab, select [Word write] in [Action mode] and set [Action address] to LW-100.
3. Select [Constant] for [Word write value settings] and enter 90 in [Start value].



4. In [Time set] tab select [Constant].
5. Enter [Start] time as 8:00:00 and select Monday to Friday. Do not select [Setting on individual day].
6. Enter [End] time as 17:00:00 and select [Enable termination action] check box.
7. Return to [General] tab and enter 30 in [End value].
8. Click [OK], a new schedule object will be created on the [Scheduler] list.

General Tab



Setting	Description
Power ON start/end action	Execute the defined action when the HMI is powered ON. ● Enabled When HMI is powered ON within the scheduled time range, the start action will be performed automatically. When HMI is powered ON outside the scheduled time range, the termination action will be executed. **Inside scheduled range**  **Outside scheduled range**  ● Disabled When the HMI is powered ON at a time later than the start time, the start action will not be performed, but the termination action will be performed. When the termination action is not defined, the scheduled range is not recognized and no action is performed.

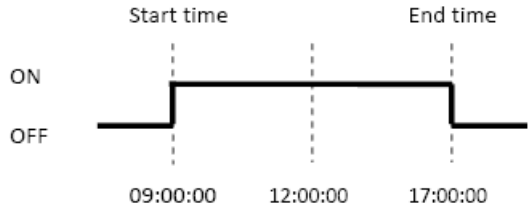
Action mode

Choose the action to do at the given time.

Bit ON

At the start time, set the designated bit ON. At the end time, set the designated bit OFF.

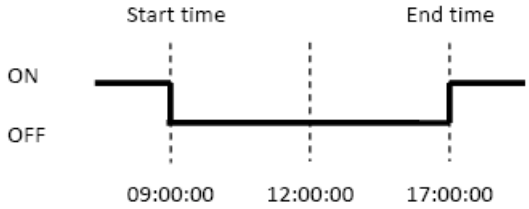
Example: Start time : 09:00:00 End time : 17:00:00



Bit OFF

At the start time, set the designated bit OFF. At the end time, set the designated bit ON.

Example: Start time : 09:00:00 End time :17:00:00



Word write

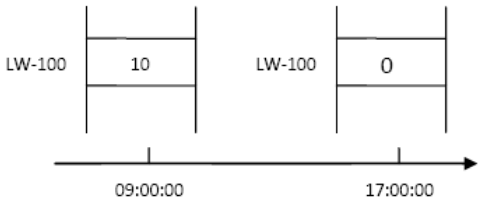
The [Write start value] entered here is transferred to the designated [Action address] word register at the start time. At end time, the [Write end value] entered here is written to the [Action address]. The valued can be entered manually or be set by using [Address] mode. In [Address] mode, the value in the specified address is the start value where the value in [Address + 1] is the end value.

Example: Device address: LW-100

Start time: 09:00:00 End time: 17:00:00

Write start value: 10 Write end value: 0

Use register: If control address is LW-n, then enter 10 in LW-n and enter 0 in LW-(n+1).

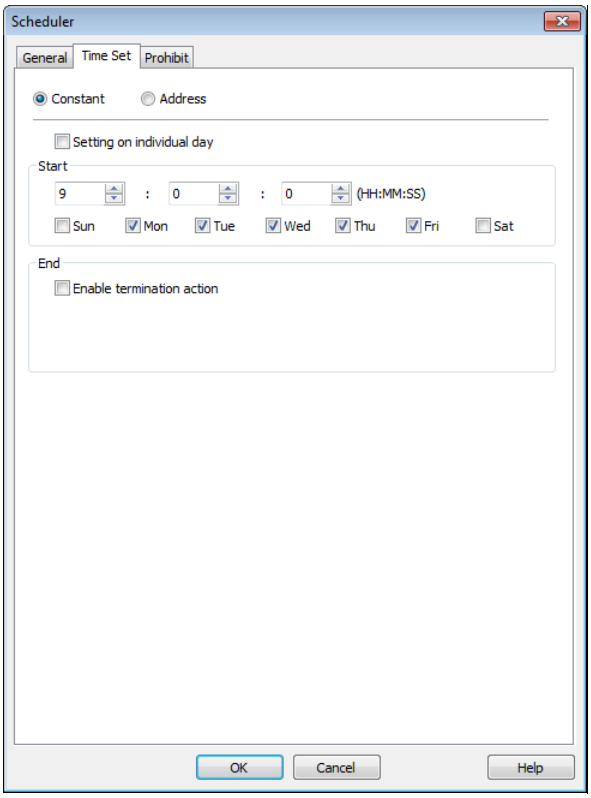


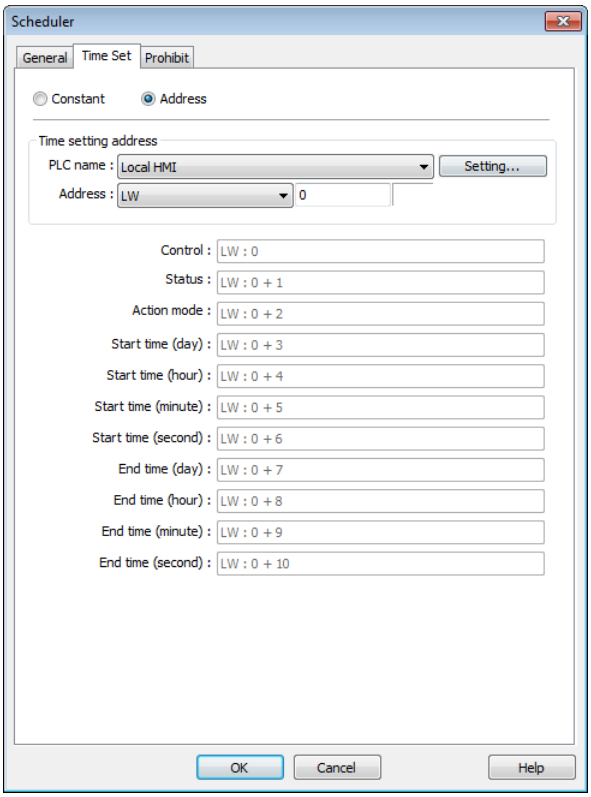
Note

- Only is an [End time] is set in the [Time set] tab will the [Write end value] box appear.

Time Set

Specify start time and end time. [Constant] allows specifying a date or period and time. [Address] allows controlling the time by the designated address.



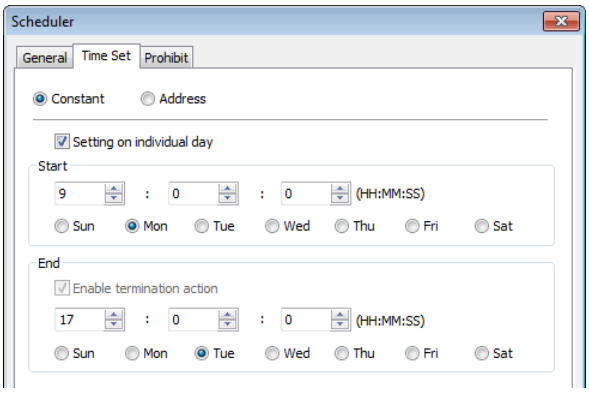


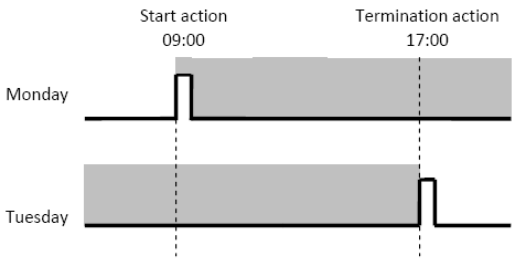
- Constant

Setting on individual day

If [Setting on individual day] is selected

The same start time and end time can be assigned to different days of the week.



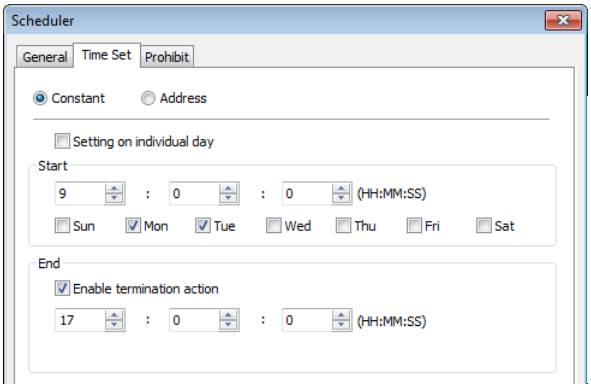


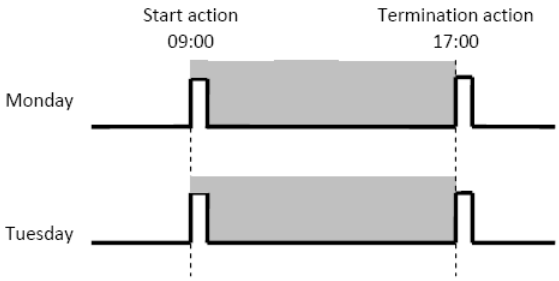
Note

- Start and end time must be entered.
- Start and end time must be on a different time, or same time but different day.

If [Setting on individual day] is not selected

Start time and end time entered must start and end within 24-hours.

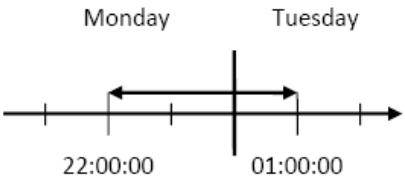




Note

- Start time and end time must be on a different time, different day.
- If an end time is earlier than a start time, the end action will occur in the next day.

Start date: Monday
Start time: 22:00
End time: 01:00



- Address

The scheduler object retrieves the start/end time and day of week information from word registers, enabling all parameters to be set and changed under PLC or user control.

Designated as the top address in a block of 11 sequential registers which are used to store time setting data.

The format of the 11 word registers should normally be 16-unsigned integer. If a 32-bit word address is chosen, only bits 0-15 are effective, and bits 16-31 should be written as zero.

The following describes each register.

Control (Time setting address + 0)

When [Control] bit is ON, the HMI will read and update [Action mode], [Start time], and [End time] values.

15	0	Bit
Reserved (0 fixed)	0	

Bit 0: no action 1: read times/action mode

Note

- HMI will not regularly read the data from [Action mode] (address + 2) to [End time] (address + 10). Please turn [Control] ON when the settings are changed.

Status (Time setting address + 1)

When the read operation is completed, Bit00 of this register turns ON. If time data read is out of range or incorrect in any way Bit01 turns ON.

15	02	01	00
Reserved (0 fixed)		0	0

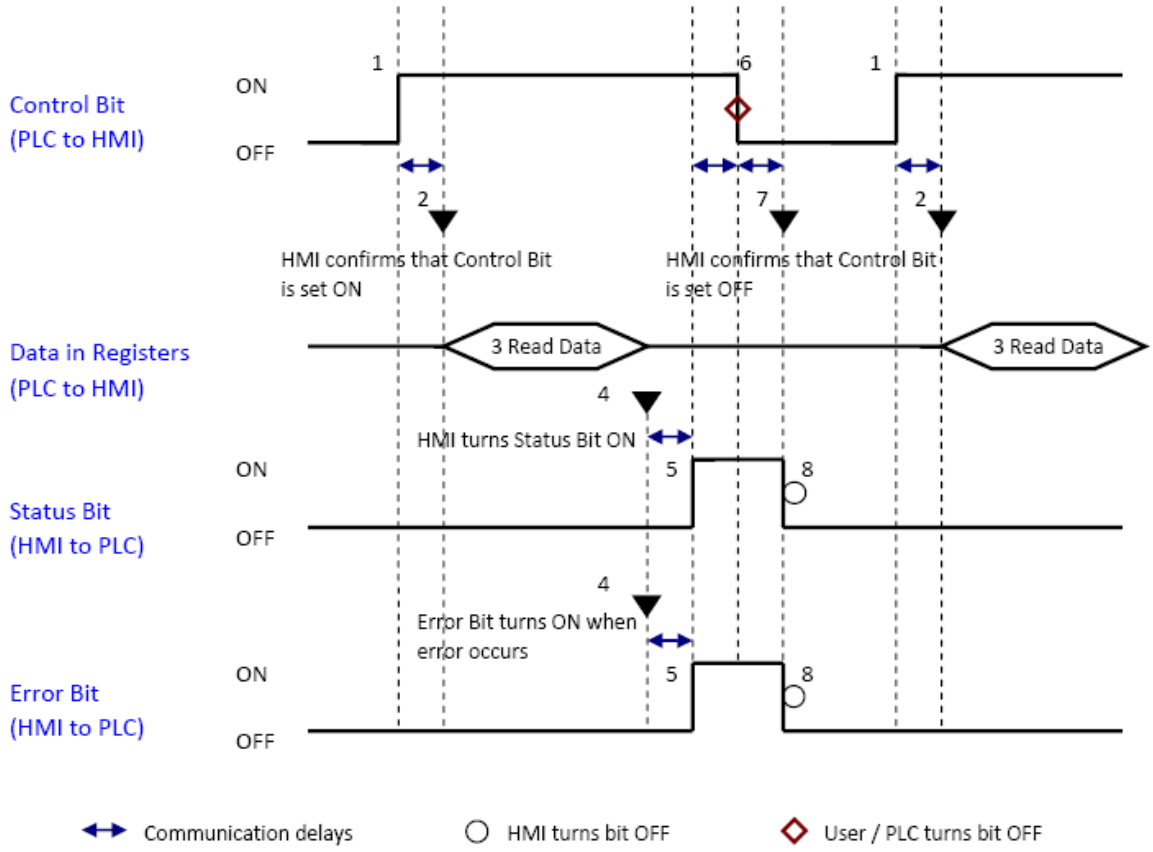
Bit

Bit 00: Status bit: Read operation completed. (0: reading or reading not started. 1: reading completed.)

Bit 01: Error bit: Start or end time format incorrect. (0: corrected 1: error)

Note

- After the scheduler reads the data and the status is turned ON (The value in [Address + 1] = 01), the control bit must be turned OFF (address = 0). The status bit and error bit will be turned OFF (1→0) at the same time.



Action mode (Time setting address + 2)

Enable/disable [Enable termination action] and [Setting on individual day]. Whatever the [Enable termination action] bit is, all the time data, from [Control] to [End time (second)], will be read.

15	02	01	00	Bit
Reserved (0 fixed)		0	0	

Bit 00 Enable termination action (0: Disabled 1: Enabled)

Bit 01 Setting on individual day (0: Disabled 1: Enabled)

Note

- If [Enable termination action] is OFF, all 11 registers are still read but end time is ignored.
- If [Setting on individual day] is ON, make sure that all start end times are entered. If more than one start / end day bit is ON, and error will occur.

Start/End Day (Start Day: Time setting address + 3, End Day: Time setting address + 7)

Designates which day of week is used to trigger the start or end action.

15	07	06	05	04	03	02	01	00	Bit
Reserved (0 fixed)		Sat	Fri	Thu	Wed	Tue	Mon	Sun	

Bit 00 Sunday (0: not used 1: used)

Bit 01 Monday (0: not used 1: used)

Bit 02 Tuesday (0: not used 1: used)

Bit 03 Wednesday (0: not used 1: used)

Bit 04 Thursday (0: not used 1: used)

Bit 05 Friday (0: not used 1: used)

Bit 06 Saturday (0: not used 1: used)

Start/End Time (Start Time: Time setting address + 4 to + 6, End Time: Time setting address + 8 to + 10)

Hour: 0 – 23 Minute: 0 – 59 Second: 0 - 59

Values outside these ranges will cause error.

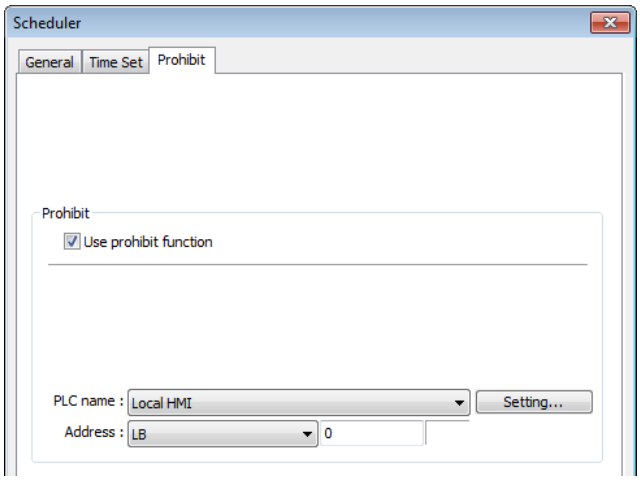
Note

- 16-bit unsigned integer format must be used; BCD format is not supported here.
- In [Address] mode, [Control] bit should be set after HMI reboots to update scheduler time.
- When using RW address, [Control] bit should be set after HMI reboots to update scheduler time. Placing a Set Bit object with [Set ON when window opens] selected in the common window is recommended, this can retain last settings after HMI reboots.

- End time depends on [Action mode] (address + 2). [Enable termination action] (Bit 00) and [Setting individual day] (Bit 01) are related:

Setting individual day	Enabled	Disabled	
Enable termination action	Enabled	Enabled	Disabled

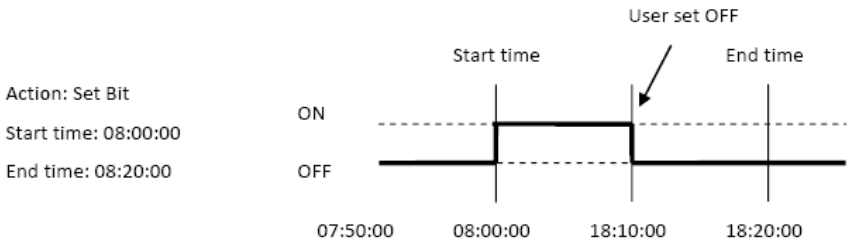
Prohibit tab



Before the scheduled action is performed, the HMI will read the specified bit state. If it is ON, the scheduled start/end action will be skipped. Otherwise, it will be performed normally.

Note

- The maximum number of Scheduler objects in a project is 64.
- A time schedule applies one action only when the start time is reached.



- [Write start/end value] and [Prohibit] bit is read only once before start action. After that, even to change the state of [Prohibit] bit or [Write start/end value], the end action and the value written will not be affected. Also, to read data of [Write start/end value] and [Prohibit] bit, there is a delay of start action due to the communication.
- Each time RTC data is changed, scheduler list entries that possess both start and end times will be checked for in-range or out-range conditions. For in-range, the start action will occur. If the end action is not set, the new range is not recognized, the action will not occur.
- If several Scheduler objects are set to the same start time or end time, the action is performed in ascending order of the schedule number.

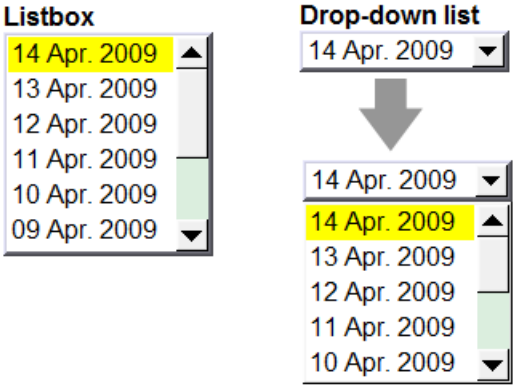
- In [Time Set] » [Address] mode, the system will read [Control] word regularly. The length of the period depends on the system.
- In [Time Set] » [Address] mode, when start time and end time is out- range, error occurs in the set action time. (Note: BCD is not an acceptable format)
- In [Time Set] » [Address] mode, the action will not start up until the first time the time data is successfully updated.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.29. Option List

13.29.1. Overview

Option List object displays a list of items that the user can view and select. Once the user selects an item, the corresponding data will be written to a word register.
There are two forms of this object – [List box] and [Drop-down list]. The [List box] lists all items and highlights the selected one. The [Drop-down list] normally displays only the selected item. Once the object is pressed, the system will display a list (which is similar to list box) as shown in the following figure.

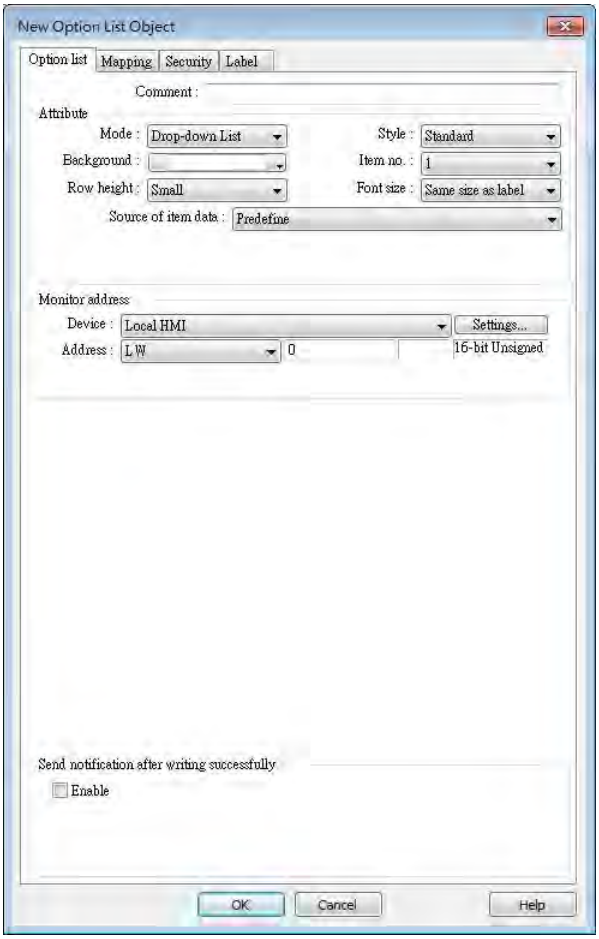


13.29.2. Configuration

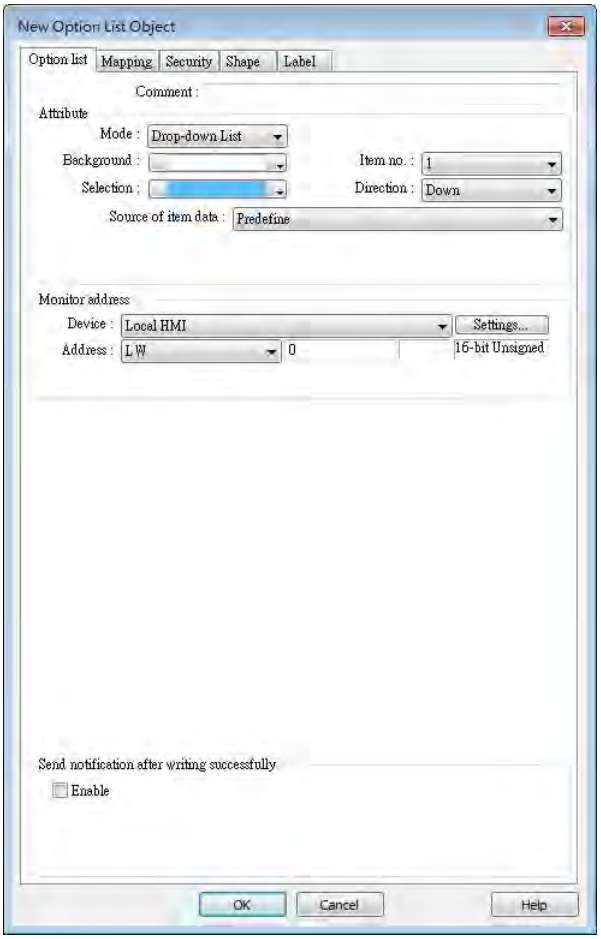
Click [Object] » [Option List] icon on the toolbar to open an Option List object property dialog box. Set up the properties, press OK button, and a new Option List object will be created.

Option List Tab

cMT / cMT X Series



eMT, iE, XE, mTV Series



Setting	Description
Attribute	**Mode:** The list style, either [List box] or [Drop-down list]. **Style:** This option is available only when the selected mode is [Drop-down list]. Available styles are: [Standard] and [Classic]. The appearance of the obejects will be shown in cMT / cMT X Style when [Standard] is selected, and shown in iE/eMT/XE style when [Classic] is selected. When the project is converted from iE/eMT/XE to cMT / cMT X, the default style will be [Classic]. **Item no.:** Set the number of items for the object. Each item represents a state displayed in the list and the corresponding value will be written to the [Monitor address]. **Background:** Set background color. **Selection:** Set background color for the selected item. **Row height:** Set the row height for the drop-down list.

	Font size: When [Same size as label] is selected, the font size set in Label tab will be used for the drop-down list. When [Fixed size] is selected, the system default font size will be used for the drop-down list. **Source of item data:** There are 4 sources available: [Predefine], [Dates of historical data], [Item address], and [User account]. See 13.29.2.1..
Monitor address	The corresponding value of the selected item will be written to [Monitor address]. **Write when button is released** If this check box is selected, the selected item value will be written to [Monitor address] after the button is released.
Send notification after writing successfully	Set On/Off the designated bit address after successfully writing data to PLC.

Note

- For cMT / cMT X Series, the [Dates of historical data] and the [write when button is released] selections are not available.

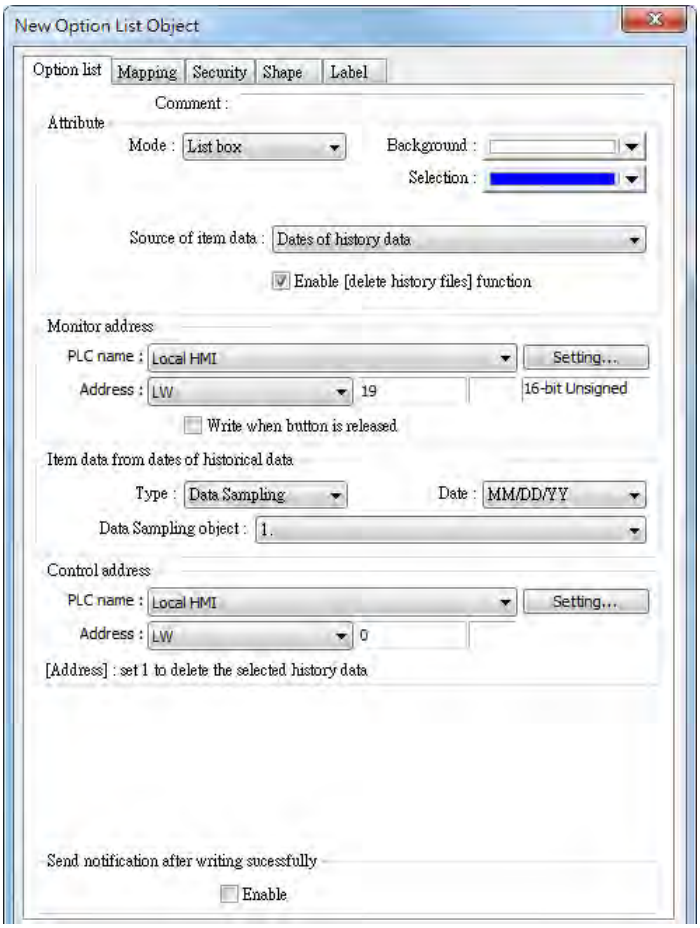
13.29.2.1. Source of item data

- **Predefine**

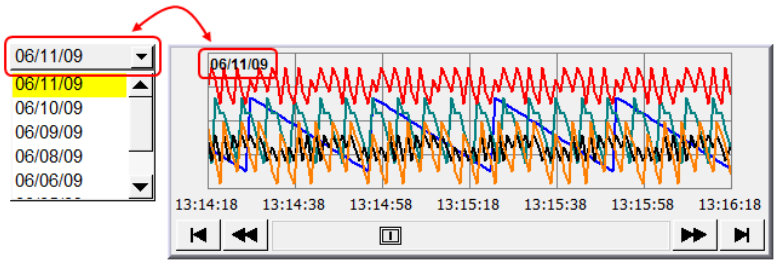
The list is manually defined in [Mapping] tab.
The number of items can be adjusted by [Item no.], and each item represents one state. Each item has a corresponding value which will be written to [Monitor address].

- **Dates of historical data**

This selection is not available for cMT / cMT X Series.



Option List object can be used with historical data display objects, such as Trend Display object, History Data Display object and Event Display object to control which history file should be shown. The figure below is an example of Option List used with Trend Display.

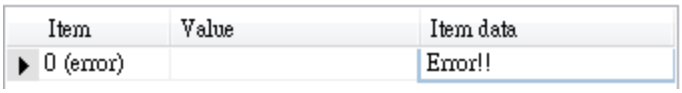


Setting	Description
Type	Two options are available: [Event (Alarm) log] and [Data sampling].
Date	Set the date format. YYYY means a four digits year (EX: 2012), YY means a two digits year (EX: 12), MM means month and DD means day.
Data Sampling object	Select which Data Sampling object is displayed when [Type] is [Data Sampling], and it should be the same as the [Data sampling object index] configured in [Trend Display] or [History Data Display].

Enable [delete history data] function	If selected, a control address can be set. Writing “1” to this address will delete the history data of the specified date.

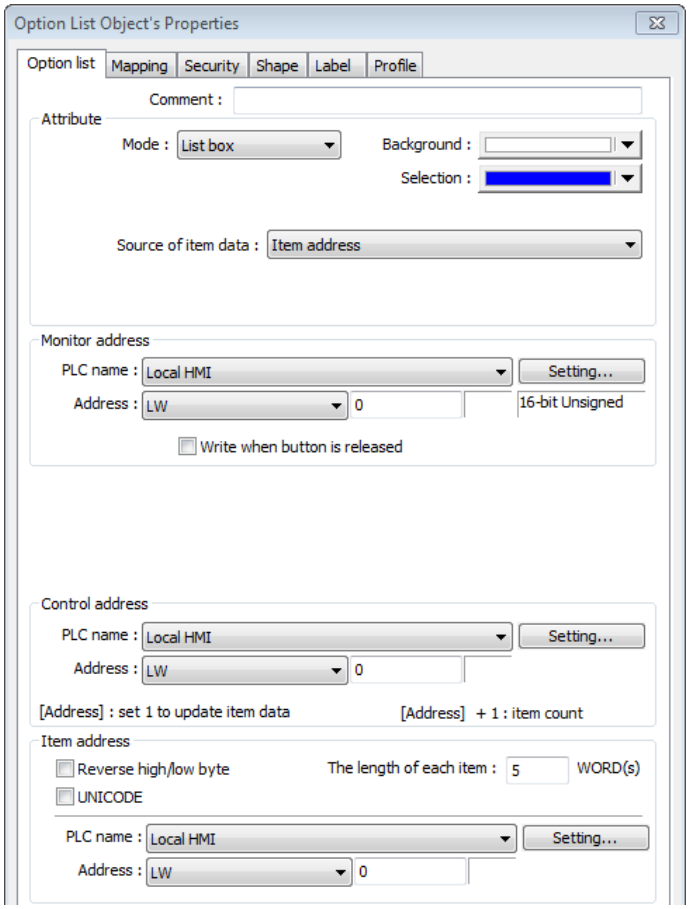
Note

- In [Dates of historical data] mode, since the system automatically reads the historical data and finds the date information, it is not necessary to configure in the [Mapping] tab.
- The error message displayed in Option List can be modified in [Mapping] tab.



- **Item address**

The list will be read from the given [Item address] and controlled by [Control address]. The following options will be available:



Setting	Description
Control address	[Address]: If the value at this address is changed to 1, the option list would be replaced by items defined at [Item address]. After updating, the value will be restored to 0. [Address + 1]: Define the number of items in [Item address].

Item address	Assign the item address **UNICODE** The item will use UNICODE characters, such as Chinese characters. **The length of each item** Define the number of letters for each item, the unit is Word.

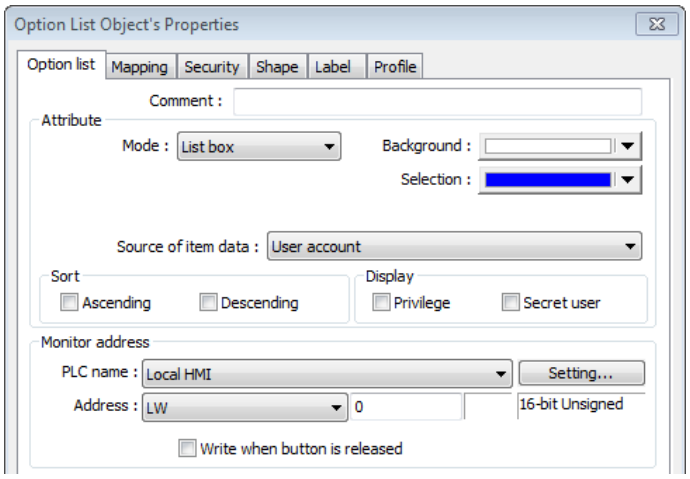
Note

- The UNICODE characters used here should be used by Text object, so that EasyBuilder Pro will compile the needed fonts and download these fonts to HMI, then the UNICODE letters can be correctly displayed.
- [The number of items] multiplied by [The Length of each item] must be less than 4096 words.
- The system automatically disables [Mapping] tab in [Item address] mode.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

- **User account**

If [Enhanced Security] mode is enabled, [User account] would appear in the [Source of item data] and it lists the names of users.



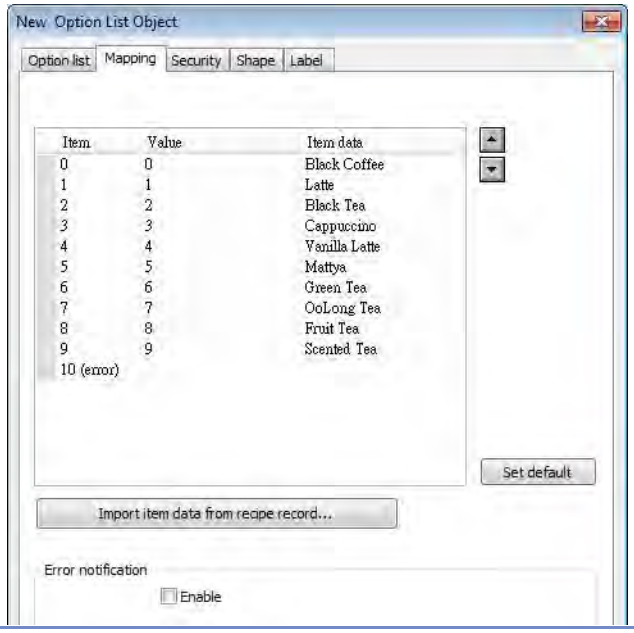
Setting	Description
Sort	Select the sorting method from [Ascending] of [Descending].
Display	If [Privilege] is selected, the privileges for each user will be displayed in option list. If [Secret user] is selected, even though it is defined to be hidden in [System parameter settings] » [Security] » [Enhanced Security], the users will still be displayed in [Option List].

Note

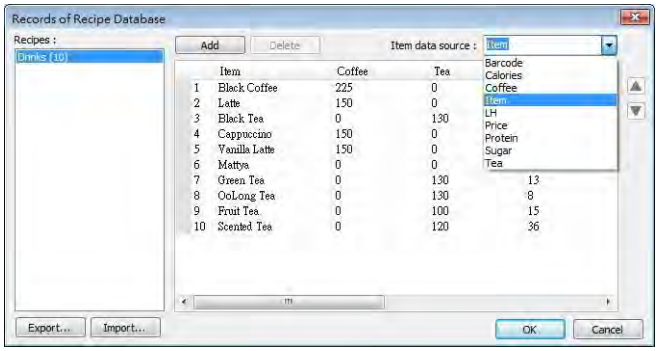
- The address that controls user index is [Control Address +2 (LW-n+2)] which is set in [System Parameters] » [Security] » [Enhanced Security].

Mapping Tab

This table displays all available states/items, their item data and values. To change the number of available items, please go to [Option list tab] » [Attribute] » [Item no.].



Setting	Description
Item	The system lists all available items. Each item represents a state that will be displayed in the list. This field is read-only.
Value	Here user can assign value for each item, basing on the following two criteria: For reading: If the value in [Monitor address] is changed, the object selects the first-matched item. If no item is matched, the status goes to error state and signals the notification bit register (if requested). For writing: The system writes this value to [Monitor address] when user selects an item.
Item data	Text displayed for each item. The Option List object displays the text of all items in the list for users to review and select.
Import item data from recipe record	This feature is enabled when select [Recipe-Selection] as [Monitor address]. Click [Import item data from recipe record] to open the [Records of Recipe Database] setting dialog box. Select [Item data source], the data belonging to the selected column will all be imported to Option List object.



Before importing, the number of items in Option List changes according to the number of items defined in Records of Recipe Database.

After importing, modifying Records of Recipe Database will not change the content of Option List.

Error state	On error state, the list box removes the highlight to represent no item is selected and the drop-down list displays the data of error state. Only the drop-down list uses error state, list box is not able to use error state. For example, item number 8 is the error state when specifying 8 in [Item no.]. (The first item number is 0)
Set default	Reset all values or states to default. That is, set 0 for item 0, 1 for item 1, and so on.
Error notification	The system will set ON/OFF to the specified bit register when error is detected. The signal of the bit register could be used to trigger a procedure for correcting the error by using objects such as Event Log, Alarm Bar, or pop-up window.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.30. Timer

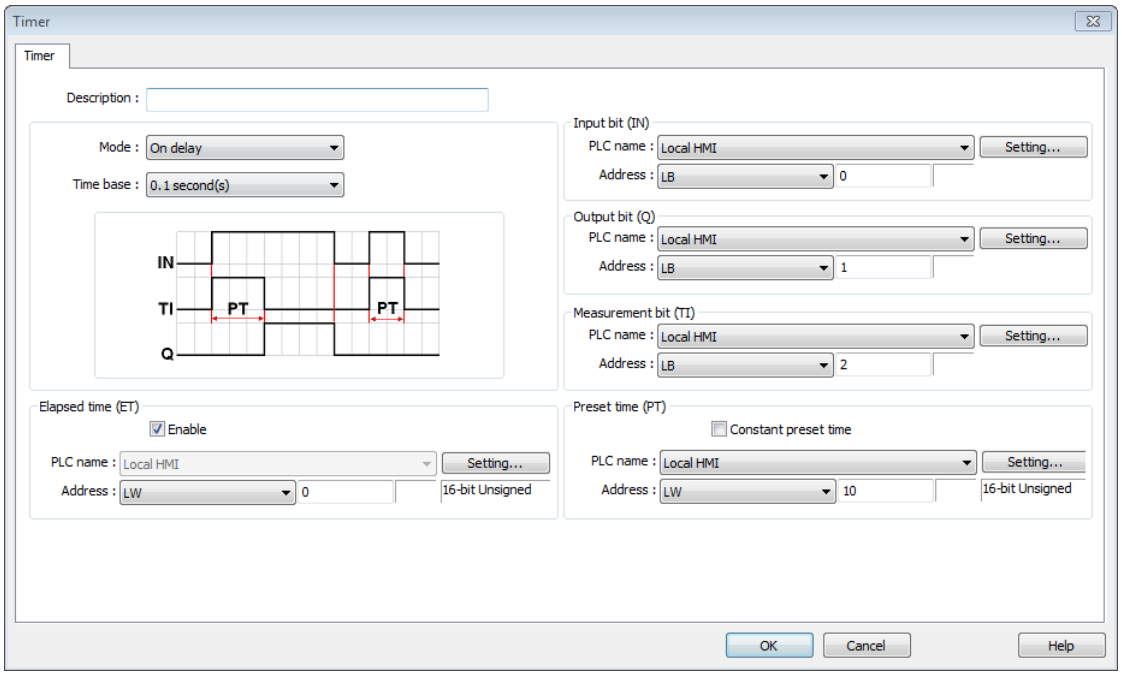
13.30.1. Overview

Timer object is a switch that can be used to control the mode to count time. The modes are explained later. Timer object uses the following 6 variables:

Timer Variable	Type	Description
Input bit (IN)	Bit	The main switch of Timer.
Measurement bit (TI)	Bit	Turns ON when the Timer begins counting time.
Output bit (Q)	Bit	Activated when the Timer finishes counting time.
Preset time (PT)	Word	Presets a time before the Timer begins counting time.
Elapsed time (ET)	Word	Displays the elapsed time.
Reset bit (R)	Bit	Resets the elapsed time (ET) to 0.

13.30.2. Configuration

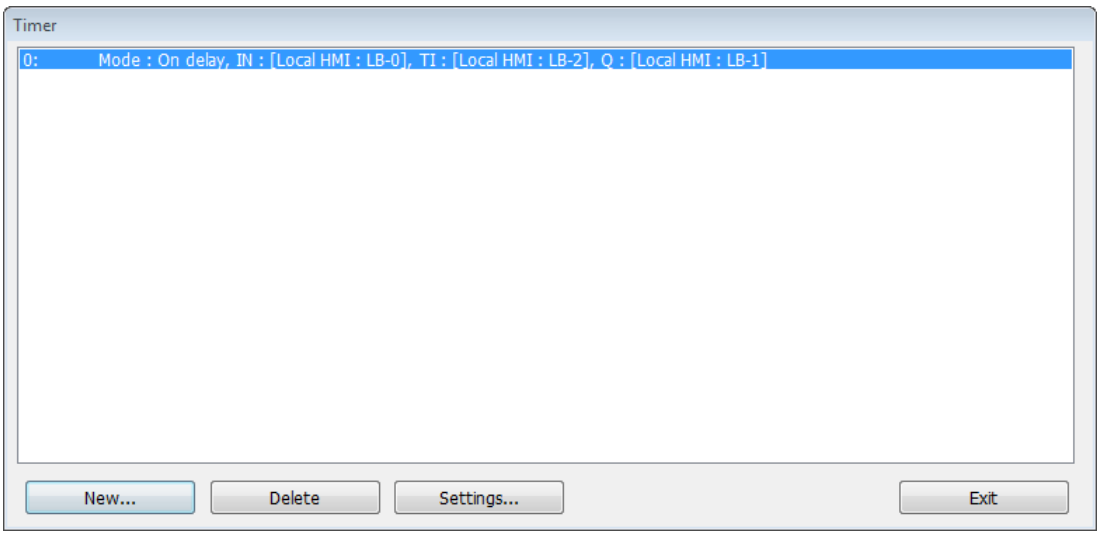
Click [Object] » [Time-related] » [Timer] icon on the toolbar to open the property dialog box as shown in the following figure.



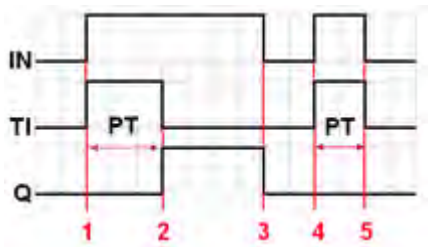
Note

■ [Constant preset time] is only available for cMT / cMT X series.

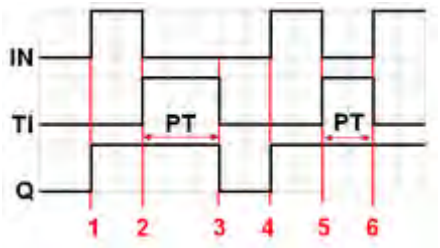
If use cMT / cMT X Series, clicking the Timer icon on the toolbar will open the Timer managing window, click [New] to configure.



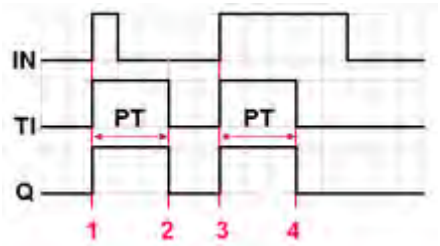
● **On delay**

Mode	Register
	**Input bit (IN):** The main switch of Timer. **Measurement bit (TI):** Turns ON when the Timer begins counting time. **Output bit (Q):** Turns ON when the Timer finishes counting time. **Preset time (PT):** Presets a time before the Timer begins counting time. **Elapsed time (ET):** Displays the elapsed time.
Description	
Period 1: When the IN turns ON, TI turns ON and the ET starts counting. The Q remains OFF. **Period 2**: When the ET equals to the PT, the TI turns OFF and the Q turns ON. **Period 3**: When the IN turns OFF, the Q turns OFF and the ET is reset to 0. **Period 4**: When the IN turns ON, the TI turns ON and the ET starts counting. The Q remains OFF. **Period 5**: Turns IN OFF before the ET reaches the PT, the TI turns OFF, and the ET is reset to 0. Since the ET doesn't reach the PT, the Q remains OFF.	

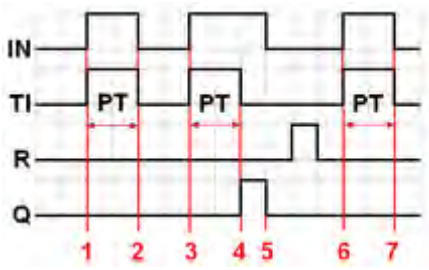
- **Off delay**

Mode	Register
	**Input bit (IN):** The main switch of Timer. **Measurement bit (TI):** Turns ON when the Timer begins counting time. **Output bit (Q):** Turns OFF when the Timer finishes counting time. **Preset time (PT):** Presets a time before the Timer begins counting time. **Elapsed time (ET):** Displays the elapsed time.
Description	
Period 1: When the IN turns ON, the TI remains OFF and the Q turns ON, the ET is reset to 0. **Period 2**: When the IN turns OFF, the TI turns ON and the Q remains ON, the ET starts counting. **Period 3**: When the ET equals to the PT, the Q and TI turn OFF. **Period 4**: When the IN turns ON, the TI remains OFF and the Q turns ON, the ET is reset to 0. **Period 5**: When the IN turns OFF, the TI turns ON and the Q remains ON, the ET starts counting. **Period 6**: Turns the IN to ON before the ET reaches the PT, the TI turns OFF, the Q remains ON, and the ET is reset to 0,.	

- **Pulse**

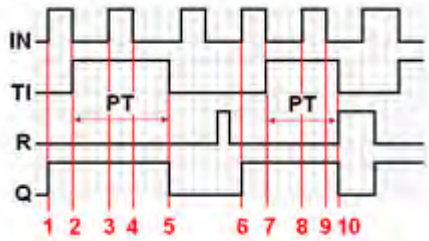
Mode	Register
	**Input bit (IN):** The main switch of Timer. **Measurement bit (TI):** Turns ON when the Timer begins counting time. **Output bit (Q):** Turns ON when the Timer begins counting time and turns OFF when the Timer finishes counting time. **Preset time (PT):** Presets a time before the Timer begins counting time. **Elapsed time (ET):** Displays the elapsed time.
Description	
Period 1: When the IN turns ON, the TI and Q turn ON simultaneously, and the ET starts counting. **Period 2**: When the ET equals to PT, the TI and Q turn OFF simultaneously. Since IN is turned OFF when counting time, the ET is reset to 0. **Period 3**: When the IN turns ON, the TI and Q turn ON simultaneously, and the ET starts counting. **Period 4**: When the ET equals the PT, the TI and Q turn OFF simultaneously.	

- **Accumulated ON delay**

Mode	Register
	**Input bit (IN):** The main switch of Timer. **Measurement bit (TI):** Turns ON when the Timer begins counting time. **Output bit (Q):** Turns ON when the Timer finishes counting time. **Preset time (PT):** Presets a time before the Timer begins counting time. **Elapsed time (ET):** Displays the elapsed time. **Reset bit (R):** Resets ET to 0

Description
Period 1: When the IN turns ON, the TI turns ON and the elapsed time ET starts counting, the Q remains OFF. **Period 2**: When the IN turns OFF, if the ET doesn't reach the PT, the TI turns OFF, and at the same time the Q remains OFF. The ET is in the retentive state. **Period 3**: When the IN turns ON, the TI turns ON. The timer measurement starts again and the ET starts counting from the kept value. The Q remains OFF. **Period 4**: When the ET reaches the PT, the TI turns OFF and the Q turns ON. **Period 5**: When the IN turns OFF, the Q turns OFF. Turning ON the reset bit R will reset the ET to 0, and then the reset bit turns OFF.

- **Accumulated OFF delay**

Mode	Register
	**Input bit (IN):** The main switch of Timer. **Measurement bit (TI):** Turns ON when the Timer begins counting time. **Output bit (Q):** Turns OFF when the Timer finishes counting time. **Preset time (PT):** Presets a time before the Timer begins counting. **Elapsed time (ET):** Displays the elapsed time. **Reset bit (R):** Resets ET to 0

Description
Period 1: When the IN turns ON, the TI remains OFF and the Q turns ON. **Period 2**: When the IN turns OFF, the TI turns ON and the Q remains ON. The ET starts counting. **Period 3**: When the IN turns ON, the TI and Q remain ON, and the ET is in the retentive state. **Period 4**: When the IN turns OFF again, the ET starts counting from the kept value. **Period 5**: When the ET equals to the PT, the TI and Q turn OFF simultaneously. Turning ON the reset bit R will reset the ET to 0, and then the reset bit turns OFF.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.31. Video In

13.31.1. Overview

HMI provides the video In feature that enables users to monitor sites with cameras. The images/video can also be stored in external devices for further analysis. Sampling applications include site monitoring, or vehicle/building surveillance…etc.

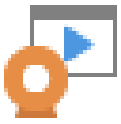
The following tables show the video input and image capturing behaviors for different models.

Model		iP	iE	eMT	mTV	XE	cMT-SVR	cMT-FHD cMT-FHDX cMT-HDM	Other cMT/cMTX
Supported device type	IP Camera	N/A	N/A	Y	N/A	Y	N/A	Y	Y
	USB Camera	N/A	N/A	Y	Y	Y	N/A	Y	Y
	Video Input	N/A	N/A	eMT3121A eMT3151A	N/A	N/A	N/A	N/A	N/A

Model		Non-cMT Series – Image Capture	cMT / cMT X Series – Image Capture/ Video Recording
Image Capturing	IP Camera	One image is captured at the trigger moment.	Image Capture: one image is captured at the trigger moment. Video Recording: video is recorded, for a pre-determined duration, around the trigger moment.
	USB Camera		
	Video Input	Several images are captured around the trigger moment.	

This object does not work remotely on cMT Viewer.

13.31.2. Configuration

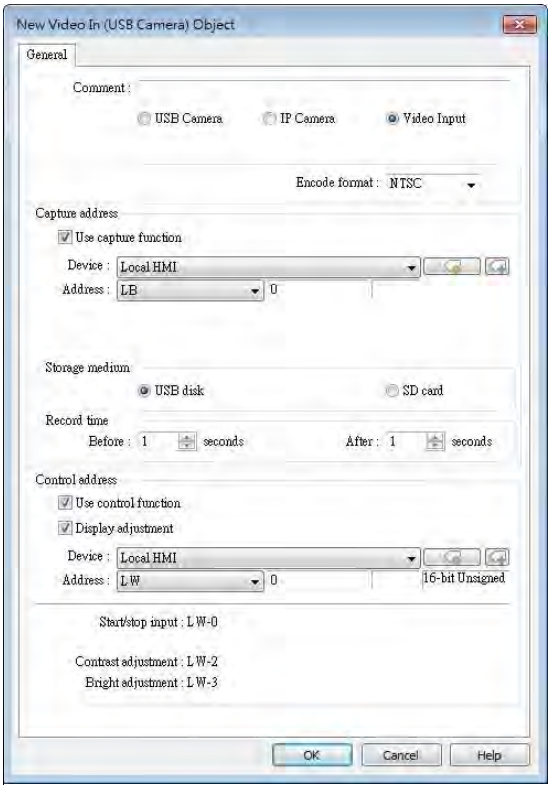


Click the Video In icon on the toolbar to open the property dialog box. Set up the properties, press OK button, and a new Video In object will be created.

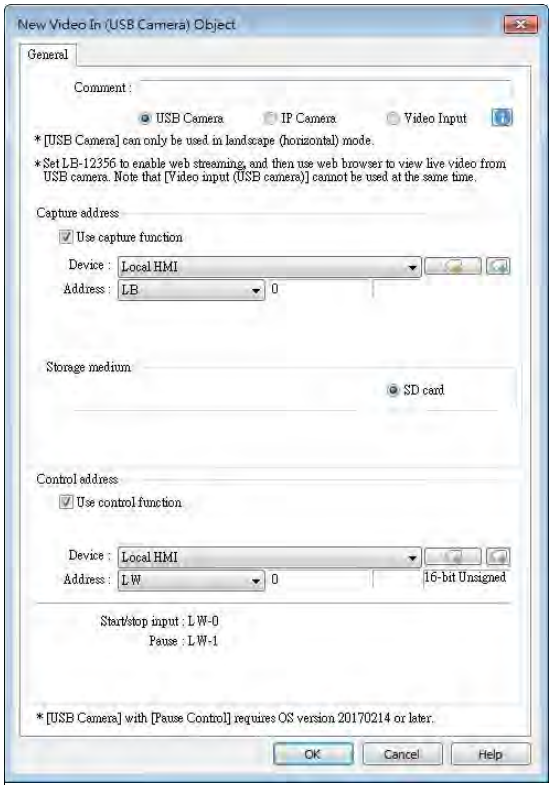
General Tab

eMT, iE, XE, mTV Series

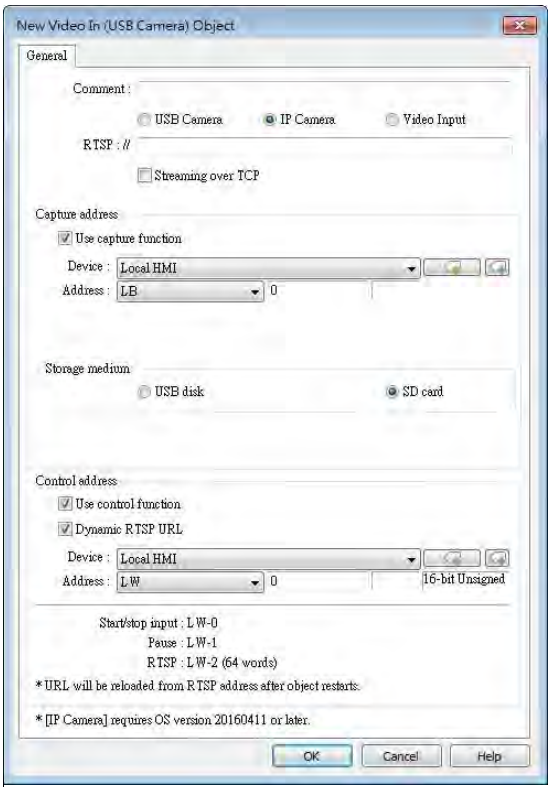
Video Input



USB Camera

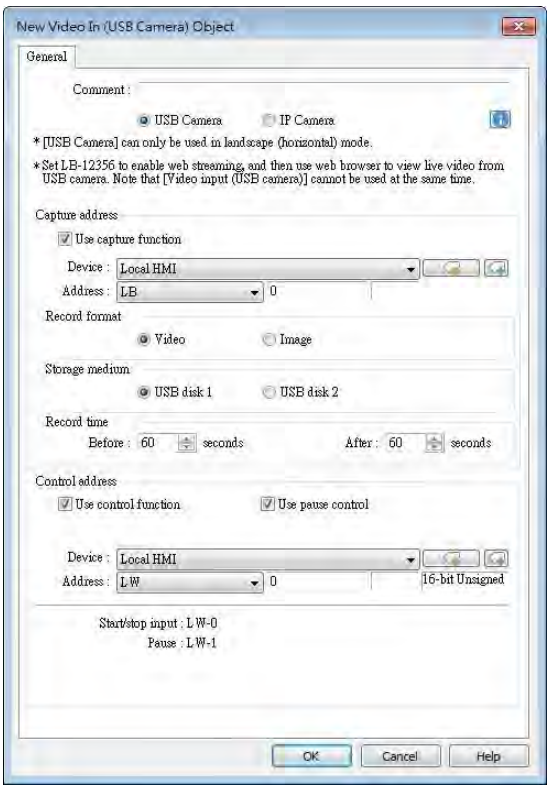


IP Camera

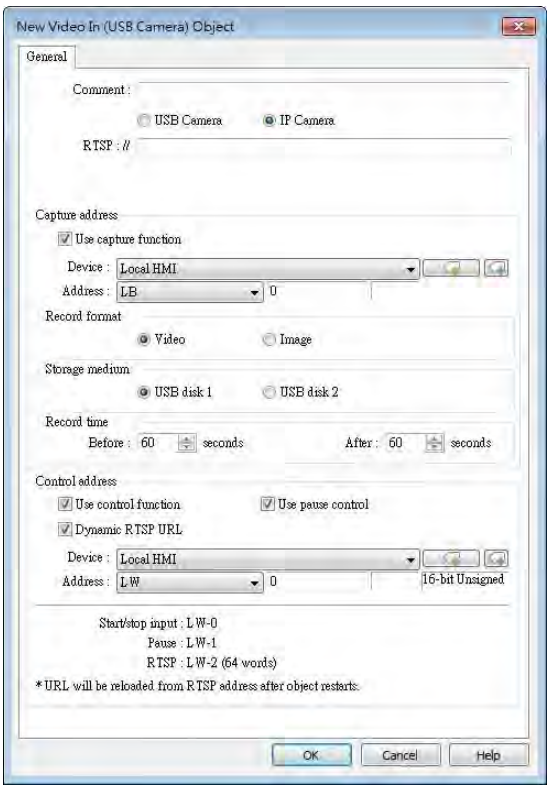


cMT, cMT X Series

USB Camera



IP Camera



Setting	Description
Input channel	Select the Video Input channel from channel 1 or channel 2. (Analog video systems)
Encode format	Select the format from NTSC or PAL. (Analog video systems)
RTSP	Enter the IP camera's RTSP address. When account and password are required for accessing the IP Camera, an RTSP address with credentials embedded can be used. For example: admin:admin@192.168.1.119:554/cam/realmonitor?channel=1&subtype=0 **Streaming over TCP** Select this check box to stream video over TCP.
Capture address	Select [Use capture function] check box and configure the settings. A non-cMT model can capture images while a cMT / cMT X model not only can capture images but also can record an avi video from n seconds before to n seconds after the triggering time of the designated address. **Capture address** Designate the address that triggers image capturing.

Storage medium

Designate the storage device.

Information about storage type and input type:

Input Type	Storage Type
Analog Input	SD card or USB disk.
USB Camera	SD card only. Exception: cMT3092X
IP Camera	SD card or USB disk.

Record time (cMT / cMT X Series)

The camera records the video from and to the specified recording time (n seconds before and after the triggering time of the designated address). The video is then saved to the specified storage device. Please note that after the video is recorded, the HMI may take a while to compress the video. The time needed for compressing the video may differ between models depending on CPU performance. A red dot appears in the object indicating that recording or compressing is in progress. The HMI will not record the next video during compression.

Record time (eMT Series)

Set a period of time to capture the images.

System	Method
Analog video system	• The longest period can be set from 10 seconds before triggering [Capture address] to 10 seconds after triggering. • The time interval of image capturing is once every second. • The captured .jpg file will be named in the following format: Before or after [Capture address] is triggered: YYYYMMDDhhmmss.jpg The moment that [Capture address] is triggered: YYYYMMDDhhmmss@.jpg For example, set [Record time] “Before” and “After” to “5” seconds. When the state of [Capture address] changes from OFF to ON, the system will start capturing one image per second, from 5 seconds before the triggering time to 5 seconds after the triggering time, which is 11 images in

	total including the one captured at the triggering moment.
USB Camera	Only the image of the triggering moment is captured. The name format: YYYYMMDDhhmmss.png.
IP Camera	Only the image of the triggering moment is captured. The naming format: YYYYMMDDhhmmss.png.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Control address

If enabled, enter certain value to the control address and the following addresses can control Video Input object. For example, if the designated control address is LW-n (n is any address), enter certain value to the designated addresses will execute commands as the following table.

Analog Video Systems

Address	Value	Command
LW-n	0	Stop displaying image
	1	Open channel 1 and display the image on HMI
	2	Open channel 2 and display the image on HMI
	3	Open channel 1 but don't display the image on HMI (Capture function operable)
	4	Open channel 2 but don't display the image on HMI (Capture function operable)
LW-n+1	1	Pause / resume the video
LW-n+2	1~100	Adjust the contrast ratio (Analog video systems only)
LW-n+3	1~100	Adjust the brightness (Analog video systems only)

USB Camera / IP Camera

Address	Value	Command
LW-n	0	Stop displaying image
	1	Start displaying image
LW-n+1	1	Pause / resume the video
LW-n+2	string	RTSP address for IP camera (64

		words)

- After changing the value in [Control address (LW-n)], the system will keep the new value.
- After changing the value in [Control address + 1 (LW-n+1)], the system will execute the command and then reset the value to 0.
- If [Use control function] check box is not selected, the system will play the image of the selected channel.
- If [Display adjustment] check box is selected, the contrast ratio and brightness can be adjusted. (Analog video systems)

Note

- Only one channel can be opened at a time, regardless of the type of camera used.

About analog video systems:

- Real-time images can still be captured when Video In is paused.
- Recommended analog video systems and resolutions:

	1:1	50%
NTSC	720 x 480	360 x 240
PAL	720 x 576	360 x 288

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

About cMT / cMT X Series Video Recording

- Compressing the recorded video may take a period of time depending on the video length.
- When HMI is compressing a video, it will not record new video even if the designated address is triggered.

About USB Camera:

- When the USB Camera is disconnected during video playing, video will not resume when camera is connected again. If [Control address] is used in the project, please use the control address to stop and restart video. If [Control address] is not used, switch to another screen and then return, or restart HMI in order to resume the video.
- The maximum size of Video In object of eMT3070A is 340*240, as for eMT3105P, eMT3120A, eMT3150A, XE, and mTV Series, the maximum size of Video in object is 640*480.
- When using a USB Camera, the display resolution of the run-time video is determined by the resolution supported by the USB Camera that is closest to the size of the object. The resolution supported by the USB Camera may not be identical to the size of the object. Therefore, it is good practice to adjust the size of object according to the actual resolution of the video.

- When using a USB Camera, the right and bottom edge of the Video In object will keep a distance of 50 pixels away from the window edge to prevent the run-time video from exceeding the window.
- When using a USB Camera, the background color of Video In object is black. If the resolution of the run-time video is smaller than the object, the empty area is colored black. Therefore, it is good practice to adjust the size of object according to the actual resolution of the video. The tested and available USB Cameras are: Logitech C170, Logitech C310, Logitech C910, LifeCam VX-2000.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

About IP Camera:

- A RTSP address is required for using IP camera. The RTSP address can be found in the configuration tool of the IP camera, or possibly from an online repository.
- When the IP Camera is disconnected during video streaming, video will not resume when the camera is connected again. If [Control address] is used in the project, please use the control address to stop and restart video. If [Control address] is not used, switch to another screen and then return, or restart HMI in order to resume the video.
- When using an IP camera, the display resolution of the run-time video will be same as that of the object drawn in the EasyBuilder Pro project. If the actual video resolution is different from the size of the object, the resolution will be adjusted to fit the object, which may cause distortion of video image. Therefore, it is good practice to adjust the size of object according to the actual resolution of the video.
- Use the IP cameras that comply with ONVIF standard, and use RTSP for video streaming.
- To ensure the quality and smoothness of the video, and not to affect HMI's general performance, adjust the settings of IP camera when video lags or high CPU loading is observed. As the project complexity and hardware specs vary from one case to another, please fine-tune the video parameters accordingly.
- Recommended video specification:

Resolution	960x544 (max.)
Format	H.264, MJPEG
Refresh Rate	15 fps
Transmission Speed	800kbps (max.)

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

13.32. System Message

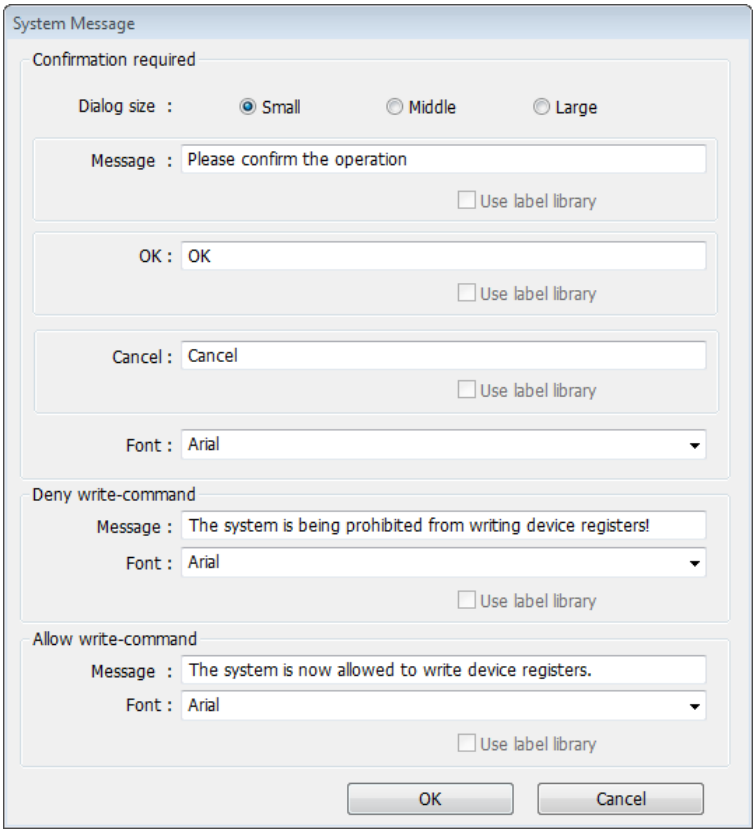
13.32.1. Overview

If objects use [Display confirmation request] or [local HMI supports monitor function only] is turned on/off, the corresponding messages configured here will be displayed in pop-up message boxes.

13.32.2. Configuration

Click the System Message icon on the toolbar to open the setting dialog box.

System Message



Setting	Description
Dialog Size	Select the size for pop-up window and texts.
Confirmation required	If an object uses [Display confirmation request], this message would pop up when the object is used. [Message] shown on

	confirmation dialog box, and the text label of the 2 buttons, [OK] and [Cancel], can be set. Please use the same font for the labels of [Message], [OK] and [Cancel]. Additionally, only when selecting [Label Library] for [Message], the use of Label Library for [OK] and [Cancel] buttons can be enabled.
Deny write-command	Displays when system tag LB-9196 (local HMI supports monitor function only) is turned ON.
Allow write-command	Displays when system tag LB-9196 (local HMI supports monitor function only) is turned OFF.

Note

- cMT / cMT X Series does not support adjusting dialog size and using system tag LB-9196.

13.33. Recipe View

13.33.1. Overview

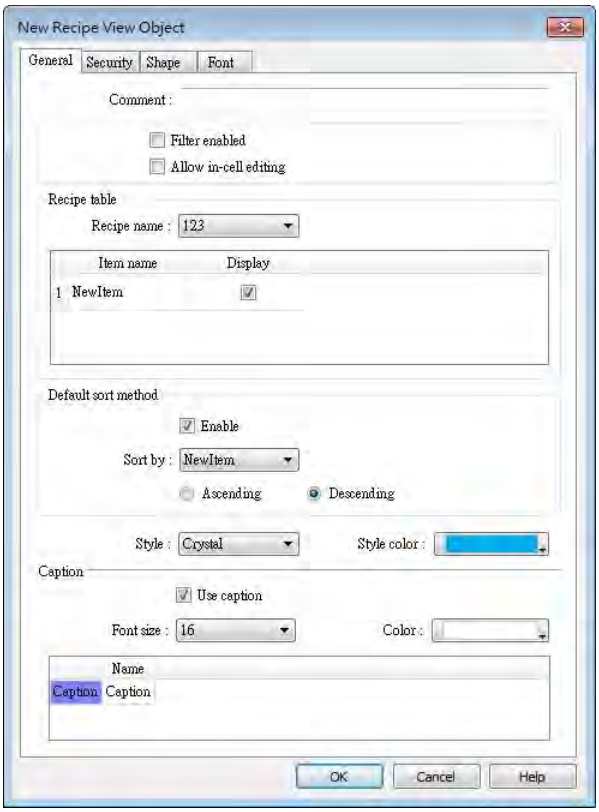
Recipe View object can be used to display a specific recipe. All items and values of the recipe can be viewed by using this object.

13.33.2. Configuration

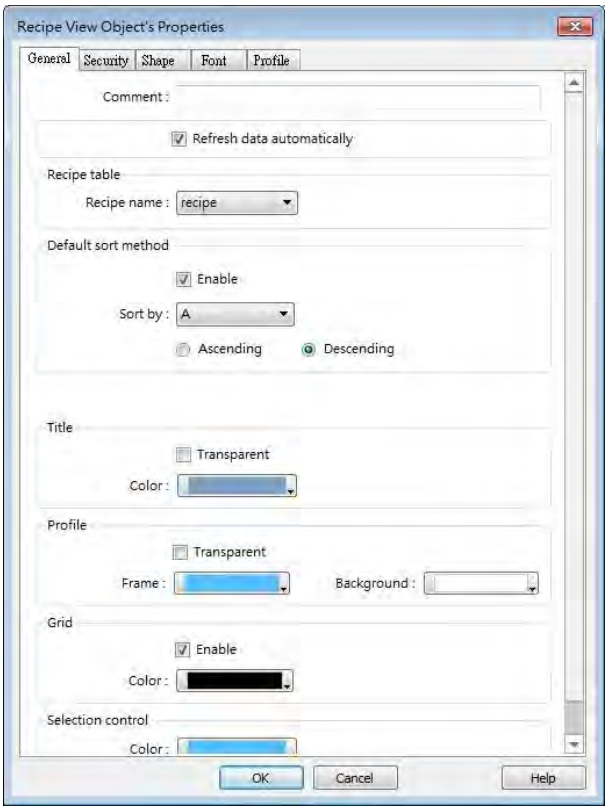
Click the Recipe View icon on the toolbar to open a Recipe View object property dialog box. Set up the properties, press OK button, and a new Recipe View object will be created.

General Tab

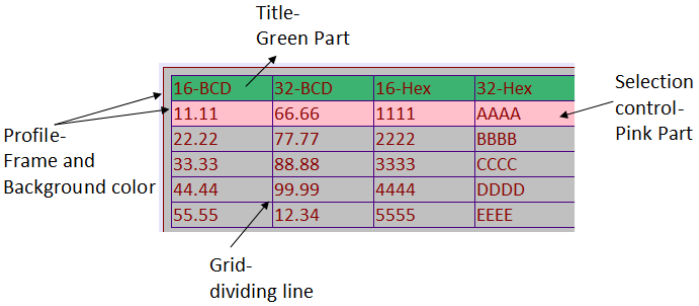
cMT / cMT X Series



eMT, iE, XE, mTV Series



The name of each part of the Recipe View object is shown in the following figure.



Setting	Description
Filter enabled	When selected, entering keywords in Recipe View to search for specific text is possible.
Allow in-cell editing	When selected, editing Recipe Database directly in Recipe View is possible.
Refresh data automatically	When selected, the system will automatically refresh Recipe View when recipe is changed; otherwise, Recipe View will be refreshed after window change.
Recipe table	**Recipe name** Choose the recipe name or look for other recipes from the drop-down list. **Display** Choose an item to be displayed by selecting its checkbox.
Default sort method	Configure how the items are sorted. [Ascending] and [Descending] can be selected.
Style (cMT / cMT X Series only)	Available styles are: Default, Crystal, and Flat.
Caption (cMT / cMT X Series only)	With [Use caption] enabled, the text, font size, color, and name of the caption can be specified. (Use caption is only available when the selected style is Crystal or Flat.)
Title	The item name assigned in [Data/History] » [Recipe Database]. **Transparent** If selected, the title row has no shading; the color selection is not available.
Profile	The frame and background color of the object can be set. **Transparent** Select to hide the background, the color selection is not available.
Grid	The dividing lines between columns and rows. **Enable** Select to show the grid. **Auto fit short column** (cMT / cMT X Series Default style) The column width automatically adjusts to the size of the content.

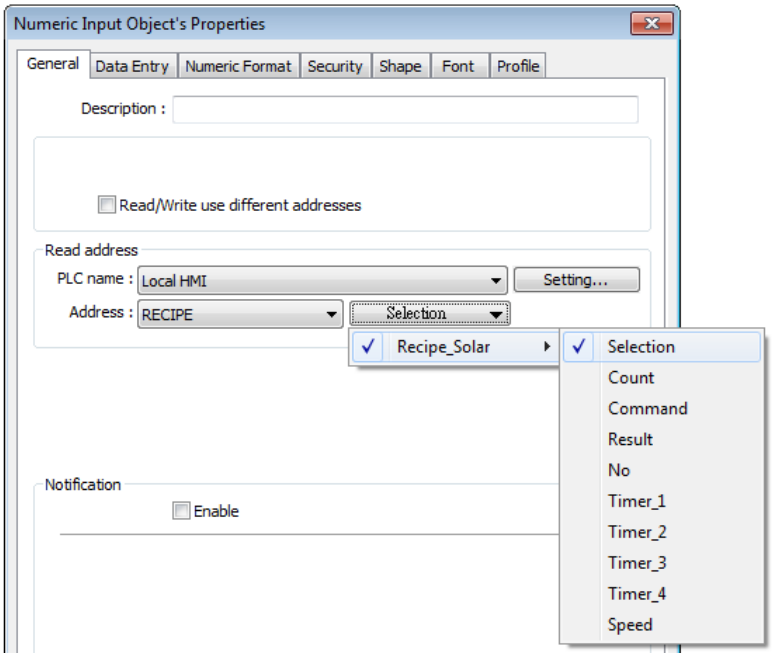
Selection Control	Shading color of the selected row.

Note

- There are 4 system registers that can be used to view/update/add/delete recipe database:

  Selection

  Current selection of record in Recipe View object, and it is numbered from zero. If the first record is chosen, the value of Selection will be 0. When the value of Selection is changed, the corresponding values will be updated, such as “No”, “Timer_1”, “Timer_2”, as shown in the following figure.



  Count

  Show the number of records in current recipe.

  Command

  Enter certain value will send command to the selected record.

  Enter “1”: Add a new recipe record.

  Enter “2”: Update the selected recipe record.

  Enter “3”: Delete the selected recipe record.

  Enter “4”: Delete all recipe records.

  Enter “5”: Write the selected recipe record to PLC.

  Enter “6”: Update the recipe record selected from PLC.

  Result

  View the result of executing commands.

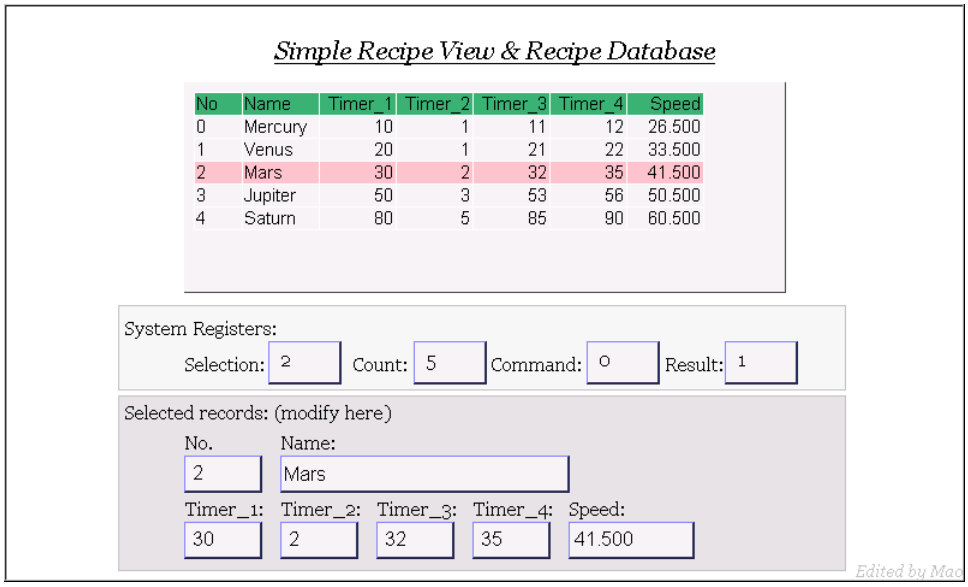
  Displays “1”: Command successfully executed.

Displays “2”: The selected record does not exist.
Displays “4”: Unknown command.
Displays “8”: Records reach limit (10000 records), no new records can be added.
Display “16”: Another command is being executed.
Display “32”: Transfer command failed.

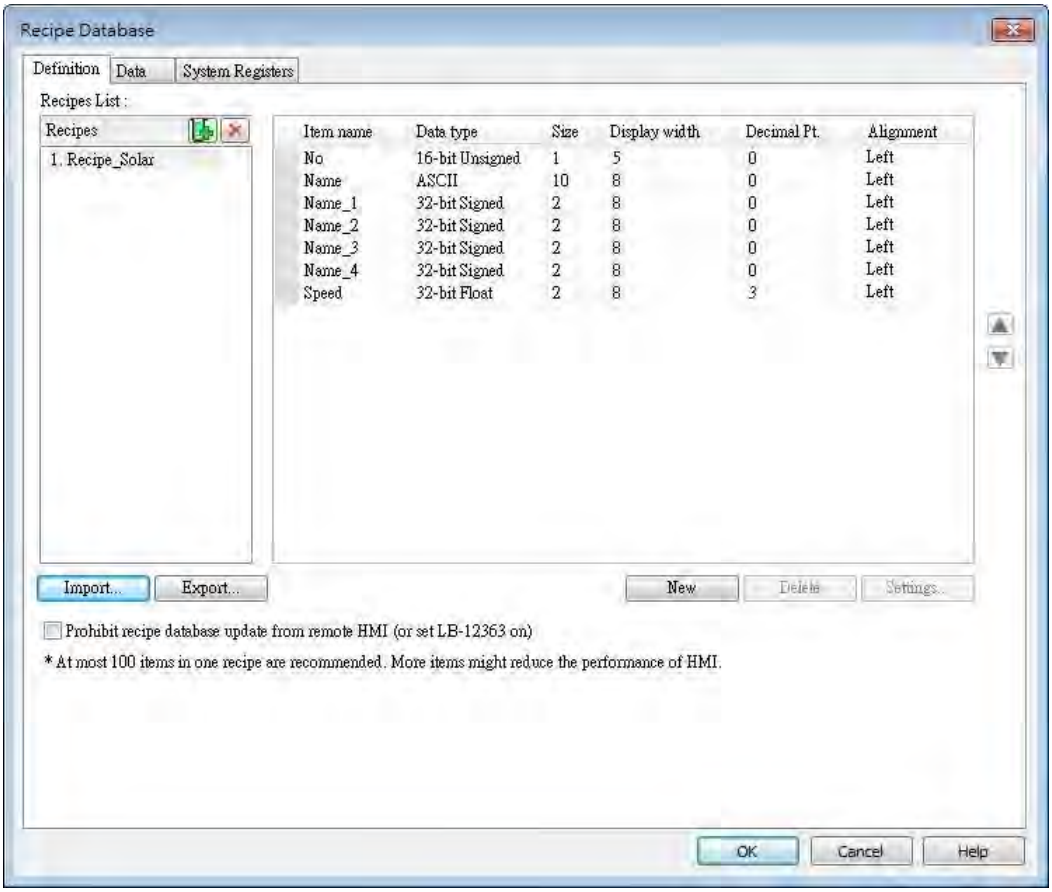
Please go to [Data/History] » [Recipe Database] tab to create the recipe data before using Recipe View object. See “24 Recipe Editor”.

Example 1

In this example, a recipe database is created to be displayed by Recipe View object. When you select a recipe record on Recipe View object, the value of [Selection] and the corresponding values will change accordingly. When finish designing, you can modify the recipe database by entering a value in [Command].

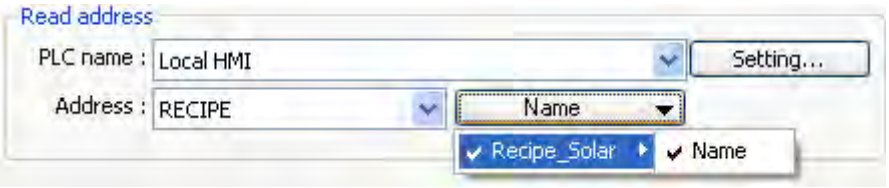


1. Create a recipe as shown in the following figure.



2. In Data tab create a number of records as shown in the following figure.

3. Create a Recipe View object and use the recipe database created in the preceding steps.
4. Create 4 Numeric objects using registers “Selection”, “Count”, “Command”, and “Result”.
5. Create corresponding input objects for “No”, “Name”, “Timer_1”, …, “Timer_4”, “Speed”. For example, “Name” is an ASCII item with size “10”. Create an ASCII object and set device type to “RECIPE” » “Name”.



6. The project is then completed.
7. As shown above, “Mars” is selected and the corresponding items are also updated. There are 5 records so the “Count” displays “5”. Try selecting different rows of the Recipe View object. Fields “Name”, “Timer_1”, …will change accordingly.
8. Try the following operations:

- Add:

  To add current data as a new record, enter “1” in “Command”.
- Update:

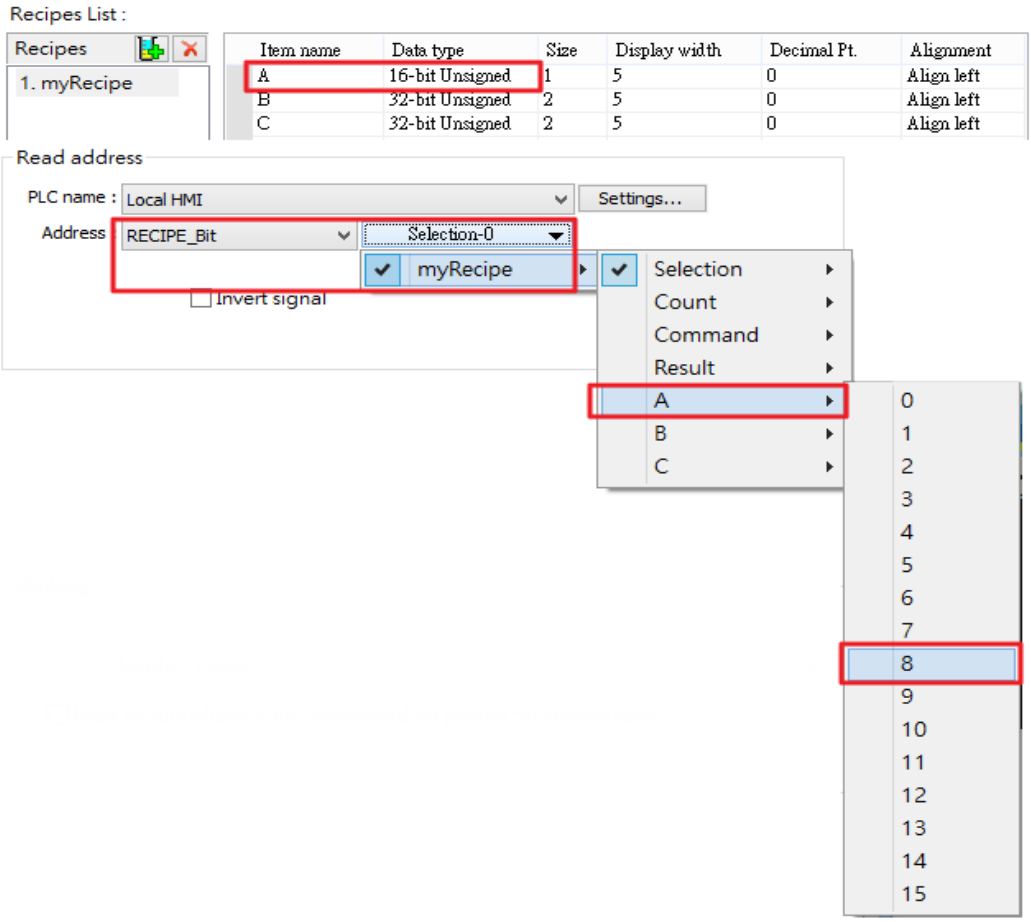
  To update recipe database, enter “2” in “Command”.

- Delete:
  To delete the selected record, enter “3” in “Command”.
- Sort the items.
- Click the title to change the order.

Example 2

In this example, [RECIPE_Bit] can be used to read / write individual bits of Recipe data. Although BOOL type items cannot be added to Recipe Database, individual bit access of 16bit / 32bit data is possible.

As shown in the following figure, select [RECIPE_Bit] for the read address of Bit object and point to the target item, and then the available Bit selections will be displayed. In this manner, Recipe Database can be used to record, read, and write bit data.



13.34. Flow Block

13.34.1. Overview

Flow Block object displays the flow status of the blocks in the pipe or the status of the transportation lines. Unlike Moving Shape object which requires a precise measurement between two points when drawing a straight line provided by users, the blocks flow at a fixed interval in a horizontal or vertical straight line. For cMT / cMT X Series models, drawing non-horizontal and non-vertical lines is possible.

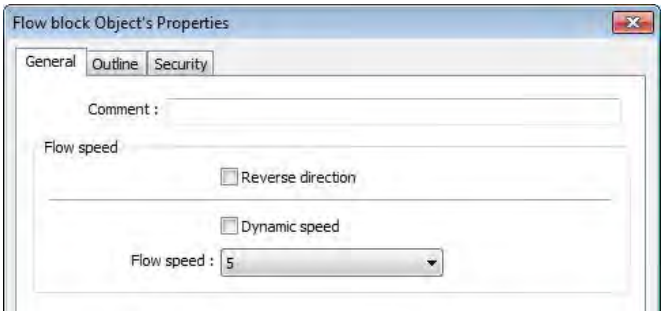
The features of Flow Block:

- Each section of the Flow Block must be a horizontal or vertical straight line and the blocks flow at a fixed interval within it. For cMT / cMT X Series models, drawing non-horizontal and non-vertical lines is possible (angle limit: 5 degrees).
- Dynamic speed and direction adjustment (Speed and direction can be controlled by a designated register.
- Security mechanism (Interlock), which hides Flow Block when the status of designated bit is invalid.

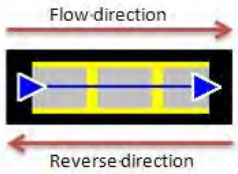
13.34.2. Configuration

Click on the Flow Block icon on the toolbar or select [Objects] » [Animation] » [Flow Block] to create object.

General Tab

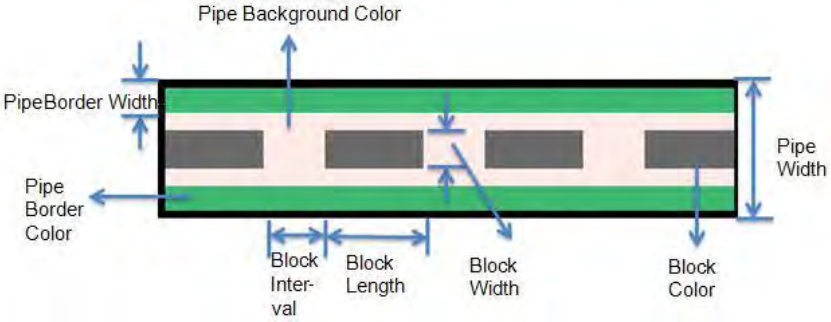


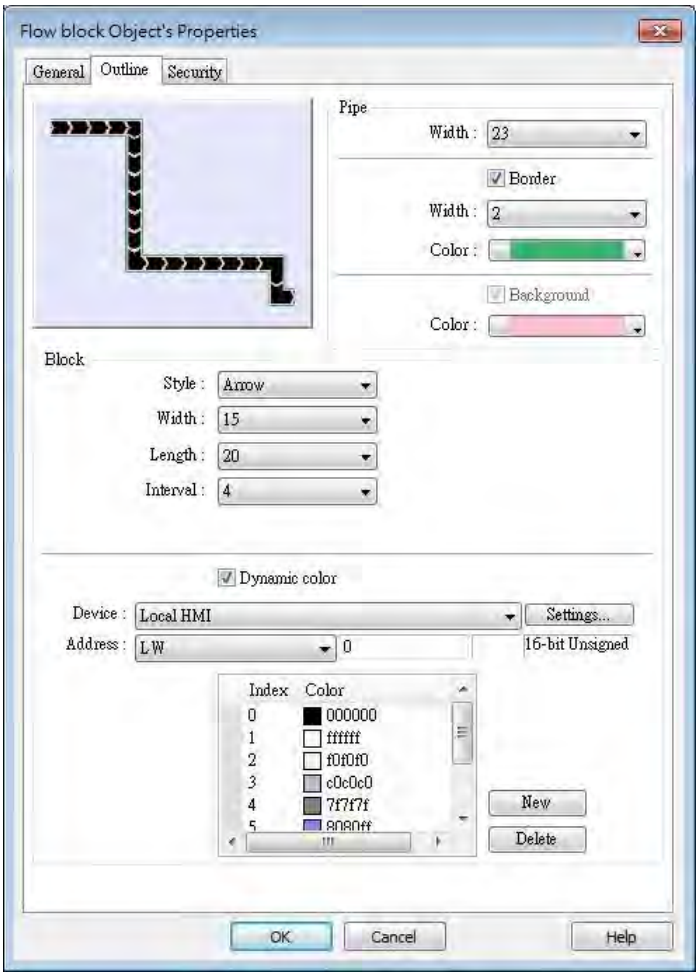
Setting	Description
Reverse direction	The blocks flow in the direction the object is drawn (the blue arrow). If select this check box, the blocks flow in the opposite direction.

	
Dynamic speed	**Read address** The direction and speed at which the blocks flow can be controlled by a designated register. The valid rage is -25 to 25. When a negative value is entered, the blocks flow in a reversed direction. **Setting** Displays the address and format of the designated register. [System register], [Index register], and [Tag Library] can be set here.
Flow speed	25 flow speed levels, the valid range is 0 to 25 when [Dynamic speed] is not selected. A larger value indicates a faster speed.

Outline Tab

For setting the outline property of Flow Block. The following illustration shows each item.





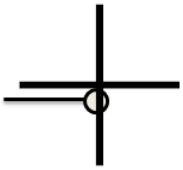
Setting	Description
Pipe	Sets the properties of the pipe within which the blocks flow. The background color, border width and color can be set. When the [Border] check box is selected, the background color must be set.
Block	Sets the properties of blocks. Style, width, length, interval and color can be set. Available styles are Rectangle and Arrow. The direction in which the arrows point to indicates the direction of the flow block. Rectangle: Arrow:
Dynamic color	Allows dynamic color change with 256 customized colors, numbered from 0 to 255 allowed. The color is selected by entering a value in the designated register. Entering a value greater than the largest color number changes the Flow Block to the last color in the list.

Note

- If both [Reverse direction] and [Dynamic speed] check boxes are selected in [General] tab,

when entering a negative value in the designated register of dynamic speed, the blocks flow in the direction the object is drawn.

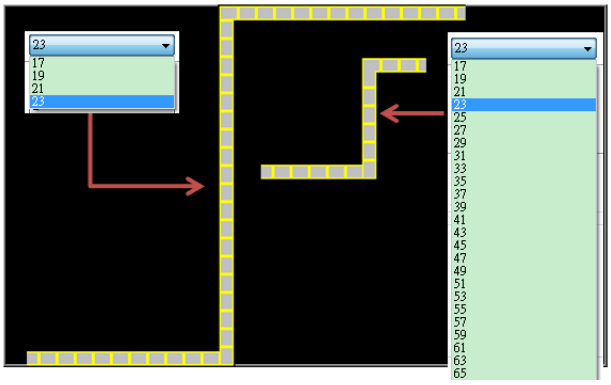
- When both [Arrow] and [Dynamic speed] are selected, the arrow will only show when a value is given to the designated address.
- To avoid the pipe lines from overlapping when drawing a turn, there is a minimum width planned at each turn. As shown in Fig. 34.1, the sign on the cross cursor defines the minimum width. Fig. 34.2 demonstrates that each turn is drawn in the minimum width.



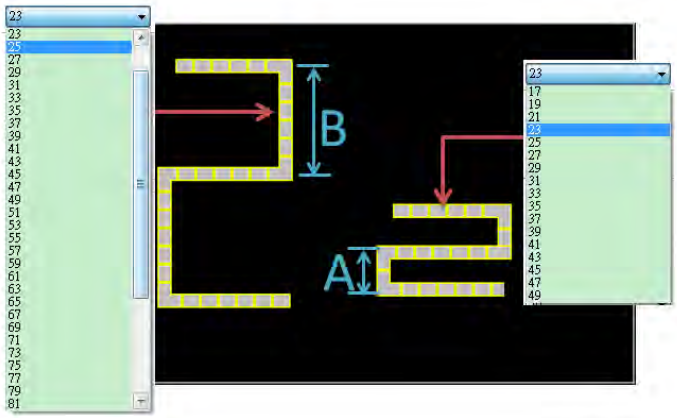
(Fig. 34.1) (Fig. 34.2)

The valid range of the length, width, and height of the Flow Block can be adjusted according to the size of the object drawn and the size of the window.

As shown in the following figure, when the size of the Flow Block is larger, the valid range is restricted to prevent the flow block from exceeding the window size. When the size of the object is smaller, the adjustment range will be larger.



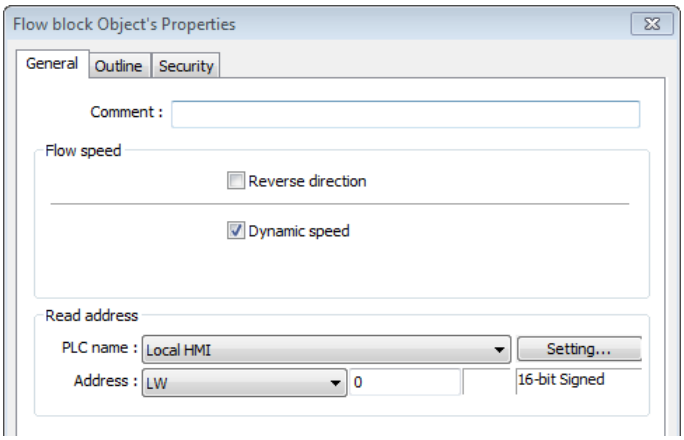
To prevent the flow block from overlapping itself, when the distance between two lines is shorter (Section A), the valid range is restricted. When the distance is longer (Section B), the adjustment range will be larger.



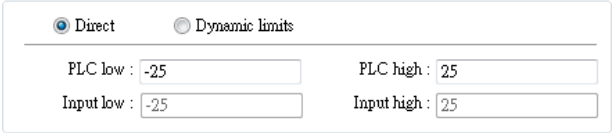
Example 1

The demonstration below shows how to use [Dynamic speed] to control the direction and speed of Flow Block by a designated word register.

1. Create a Flow Block object and select [Dynamic speed] check box. Set [Address] to LW-0, and set the format to 16-bit Signed.



2. Create a Numeric object, set [Address] to LW-0. The high limit is 25, and the low limit is -25. The format is 16-bit Signed.



3. Execute simulation or download the project to HMI. When entering a positive value in LW-0, the blocks flow in the direction the section is drawn. A larger value indicates a faster speed. When a negative value is entered, the blocks flow in a reversed direction, and the smaller value indicates a faster speed. When 0 is entered, it stops flowing.

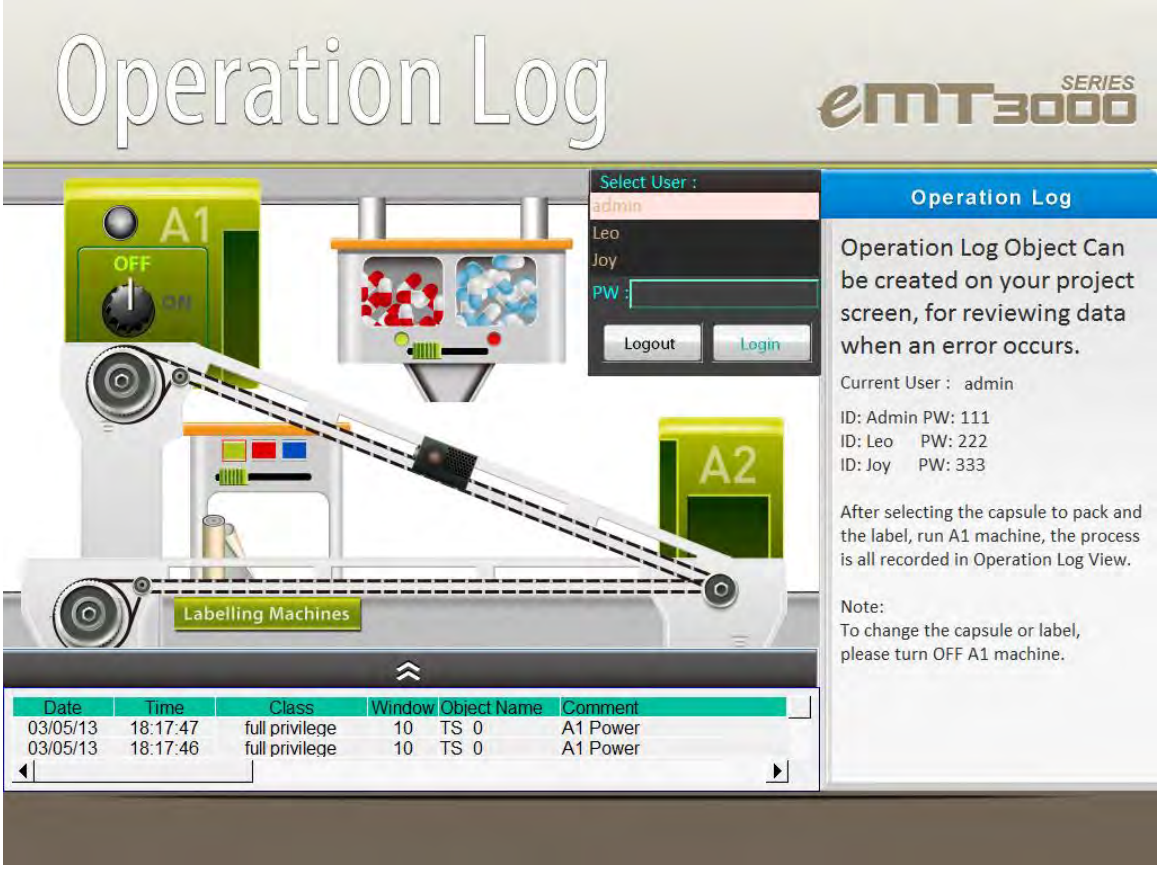
Click the icon to download the demo project. Please confirm your internet connection.

13.35. Operation Log

13.35.1. Operation Log Settings

13.35.1.1. Overview

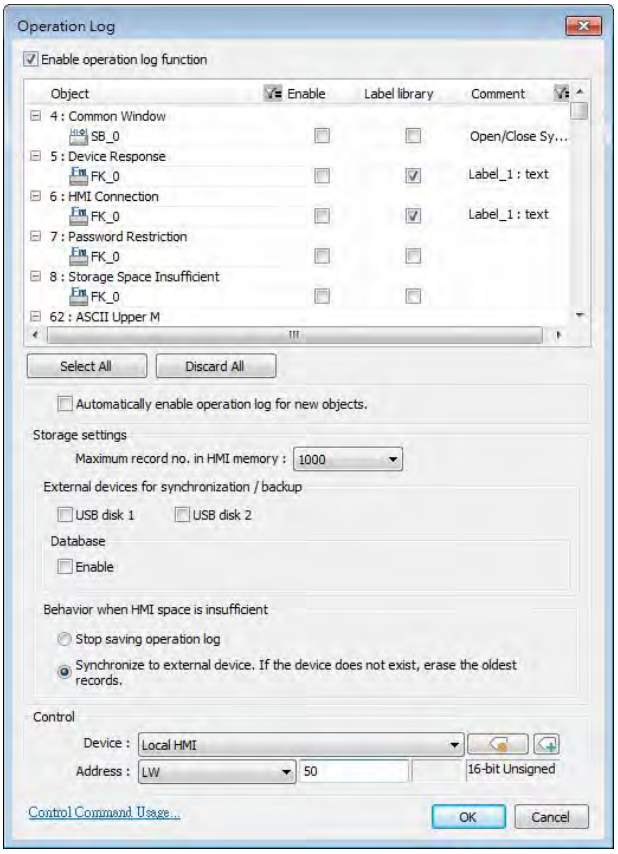
Operation Log records user's operation steps and displays the record in real-time. When an error occurs, use operation log to analyze the problem. The backup tables can be used to review the process in order to resolve the errors. Operation log, if configured properly, may also be used to satisfy the requirement for audit trail and electronic signature.



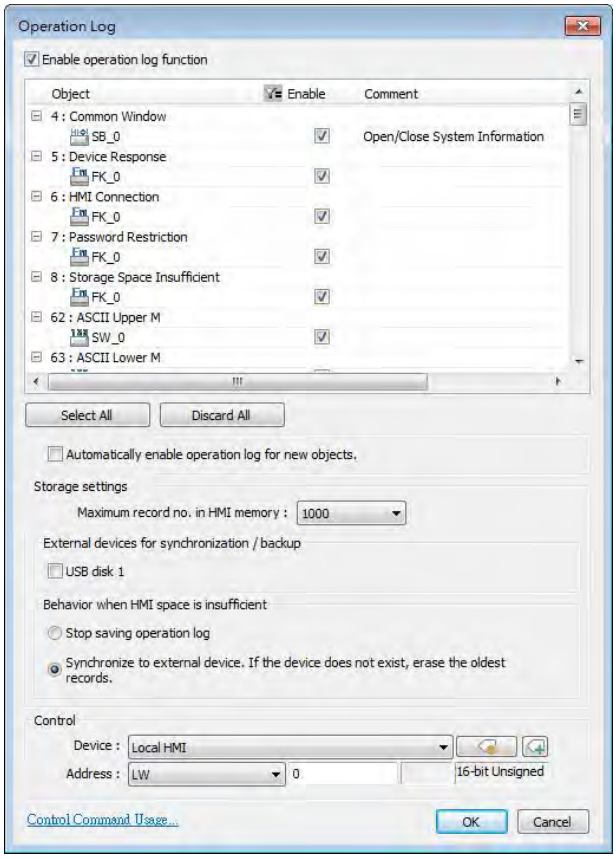
13.35.1.2. Configuration

Select the objects to be recorded. Click [Data/History] » [Operation Log Setting], and then select [Enable operation log function] check box.

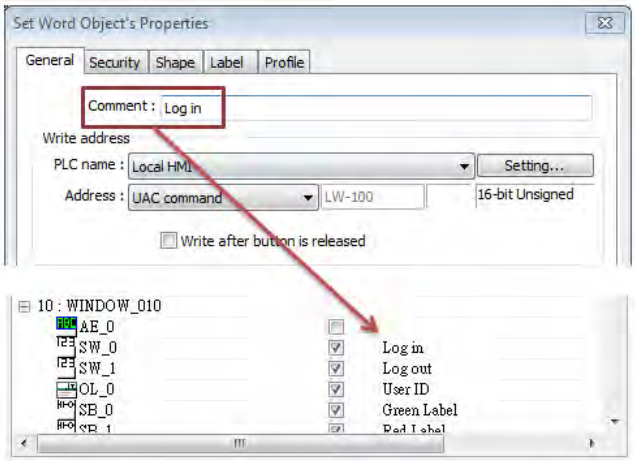
cMT, cMT X Series



eMT, iE, XE, mTV Series



Setting	Description
Object	When Operation Log is enabled, the objects that can be recorded are listed in the setting dialog box sorted by window numbers. [Filter]: By clicking icon, all recordable objects are listed. Users can use the filter to more easily locate the desired objects.
Enable	The selected objects are recorded by Operation Log.
Label Library	Use the labels in Label Library to display the comments in different languages.
Comment	The description of the object. The selected label will be shown in this column, and if Label Library is not used, the comment in the object's settings will be shown as below.



By clicking icon, all objects whose comments contain the string are listed.

Select all	Selects all the listed objects. If [Filter] is used, clicking [Select all] only selects the objects in the list.
Discard all	Discards all the selected objects. If [Filter] is used, clicking [Discard all] only discards the objects in the list.
Automatically enable operation log for new objects	With this option selected, new objects created will automatically be enabled for Operation Log.
Storage settings	Sets the way the records are stored. **Maximum record no. in HMI memory** Sets the maximum number of records that can be stored in HMI memory. **External devices for synchronization / backup** Stores backup data to SD card or USB disk. Backup data can be synchronized to database (cMT / cMT X Series). **Behavior when HMI space is insufficient** When HMI memory space is insufficient, two options are provided: [Stop saving operation log]: Stops saving new records in order to keep the earlier records. [Synchronize to external device]: Stores the Operation Log to the external device. When the device does not exist, the HMI clears the oldest records in its memory.
Control address	Entering different values in the control address sends corresponding commands to Operation Log and returns the result of executing the command. If control address is LW-n (where n is an arbitrary number), the

	address that returns the result of executing the command is LW-n+1. Control address (LW-n): (1): Clear all records. (2): Copy the records to the USB disk. (3): Copy the records to the SD card. (4): Copy the records to the USB disk and clear the records in HMI memory. (5): Copy the records to the SD card and clear the records in HMI memory. (6): Enable Operation Log. (7): Disable Operation Log. (8): Use history data stored in USB disk after changing HMI. (9): Use history data stored in SD card after changing HMI. (10): Copy the records to the database server. (cMT / cMT X Series) (11): Copy the records to the database server and clear the records in HMI memory. (cMT / cMT X Series) (12): Use existing historical data in database after changing to another HMI. (cMT / cMT X Series) Execution result (LW-n+1): (0): Processing. (1): Execution succeeded. (2): The device does not exist. (3): The record does not exist. (4): Unknown error.

Note

- Operation Log can only record the operation of the objects that are manually triggered. Objects that cannot be manually triggered are not recorded, such as Time Based Data Transfer object.
- When running off-line or on-line simulation, Operation Log is stored under EasyBuilder installation directory: HMI_memory\operationlog\operationlog.db
- Triggering Macro with a Set Bit object generates two records, the triggering of bit and the triggering of Macro.

13.35.2. Operation Log View

13.35.2.1. Overview

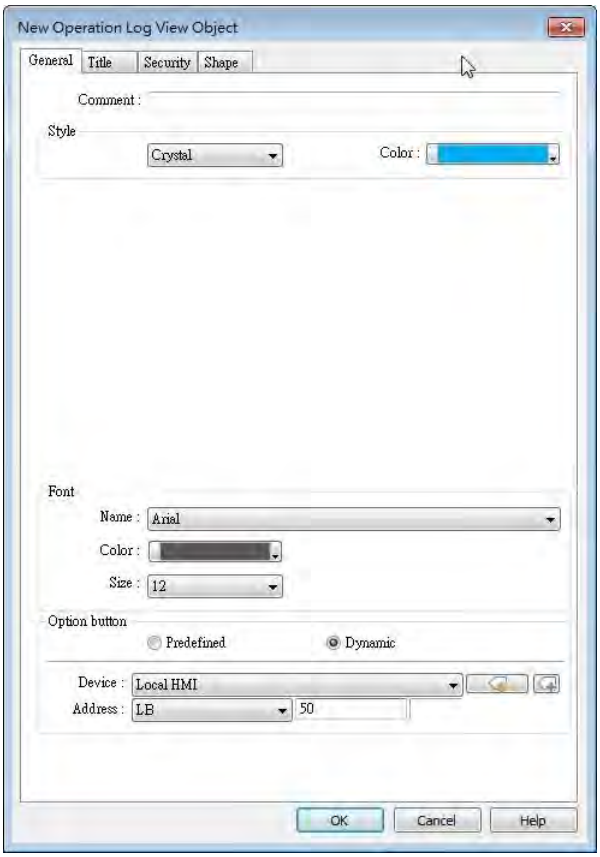
Operation Log View can be used to review the Operation Log.

13.35.2.2. Configuration

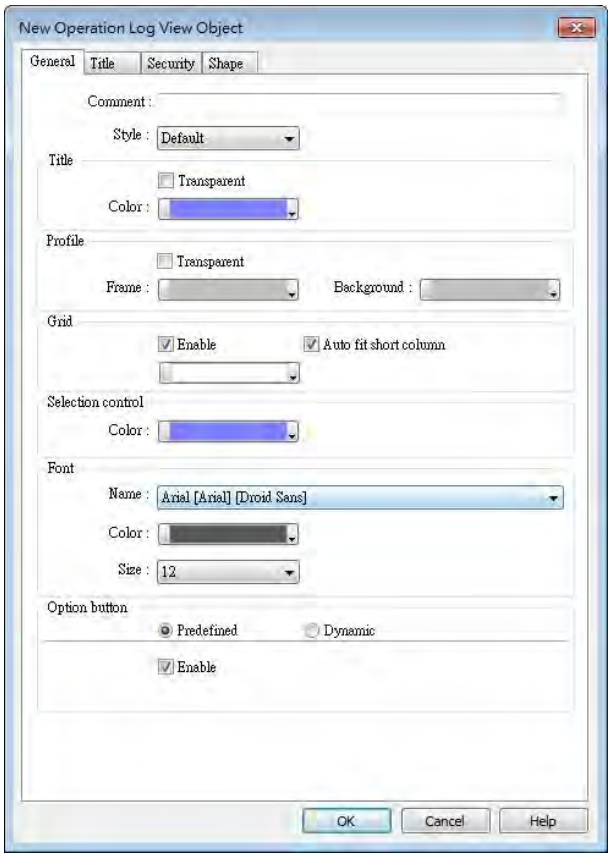
Before using Operation Log View, please follow the steps described in the preceding part to finish Operation Log Settings. Click [Data/History], and then click [Operation Log View].

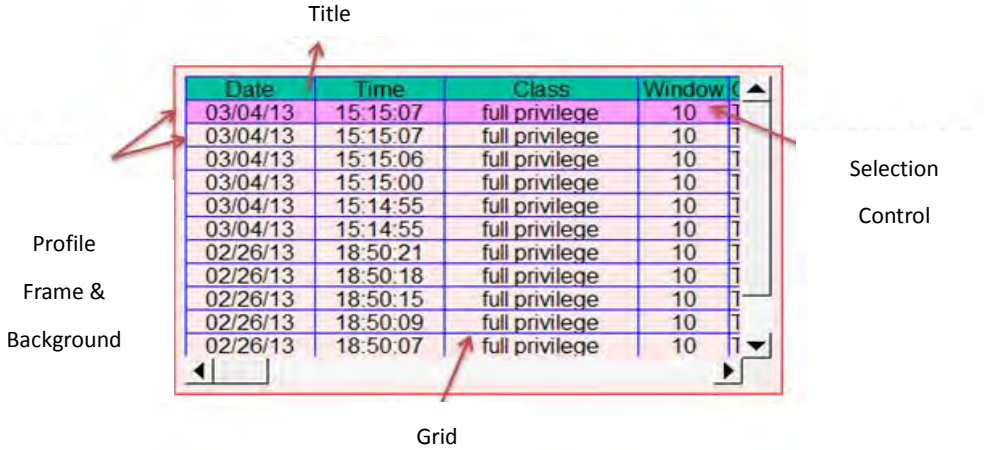
General Tab

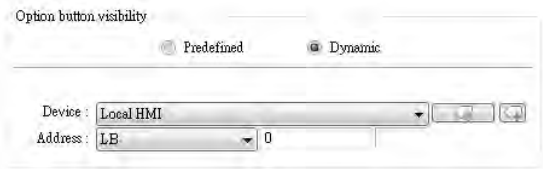
cMT, cMT X Series



eMT, iE, XE, mTV Series

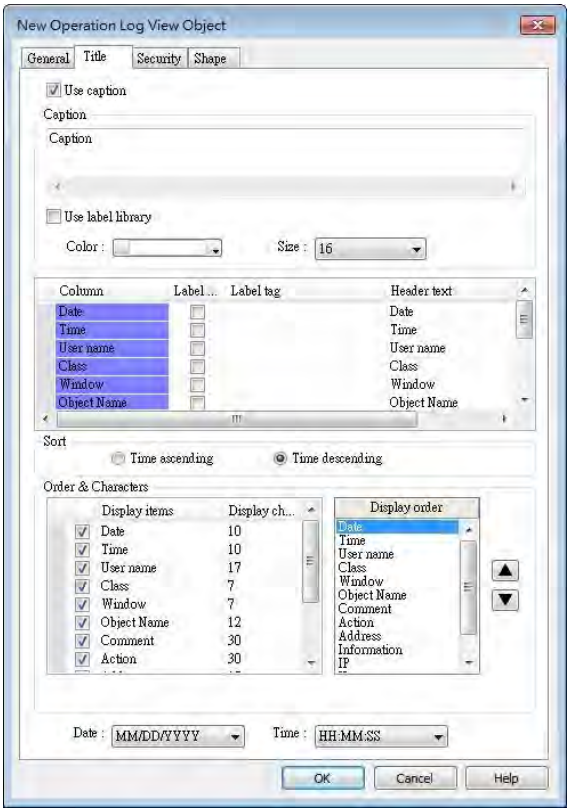




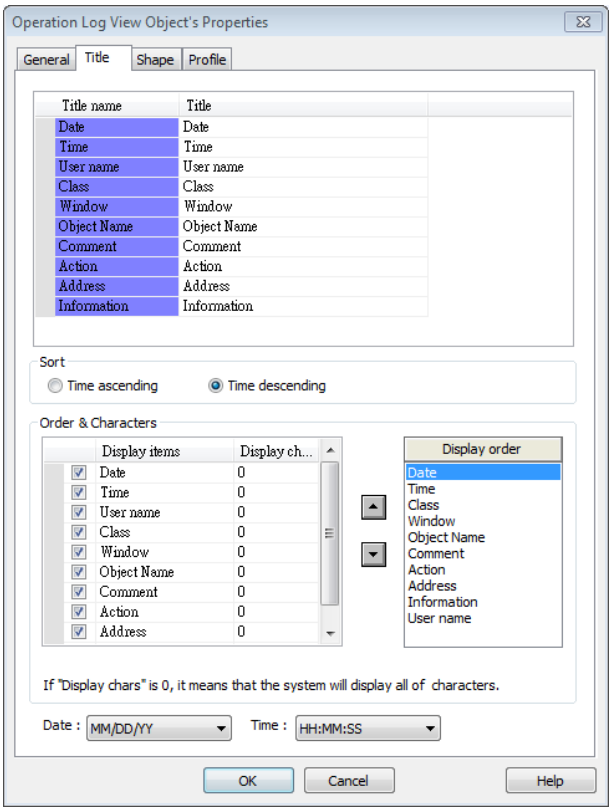
Setting	Description
Style	The style of Operation Log View object can be selected from Default, Crystal, and Flat.
Title/Profile/Grid/ Selection control	These attributes can be configured when the chosen style is Default.
Font	Sets the color, font, and font size of the text displayed in Operation Log View object.
Option button visibility (cMT / cMT X Series)	Show or hide the option button . **Predefined** Select [Enable] to show the option button. **Dynamic**  Designate a bit address to control option button visibility.

Title Tab

cMT, cMT X Series



eMT, iE, XE, mTV Series



Setting	Description
Use Caption	Sets the caption displayed in Operation Log View object.
Title	Sets the title name of the columns displayed in Operation Log View object.
Sort	Sorts the records in time ascending or descending order.
Display order	Sets the order of the displayed item. If [Display chars.] is 0, all characters are displayed.
Date / Time	Sets the format of date and time displayed in Operation Log View object.

13.35.3. Operation Log Printing

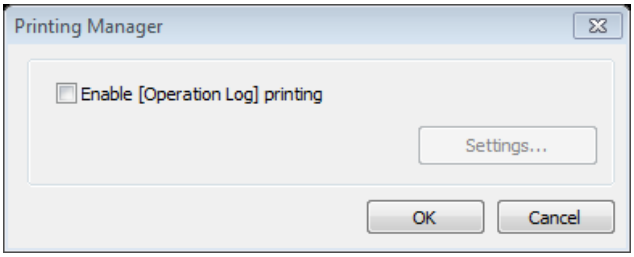
13.35.3.1. Overview

Operation Log Printing can generate an Operation Log sheet by printing out using a printer or by saving as JPEG file into an external device. Before using this function, please go to Operation Log Settings to finish the settings.

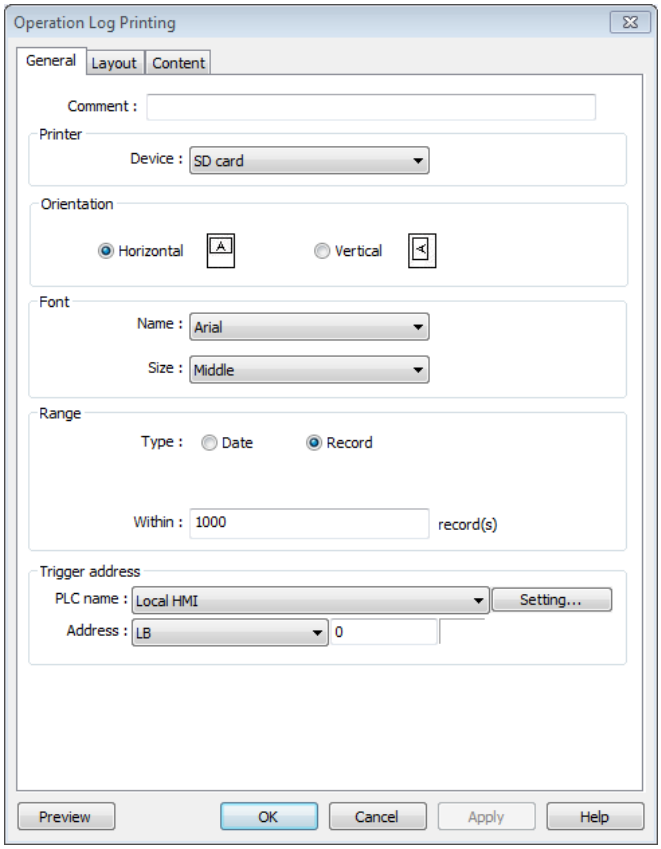
Operation Logs printed using cMT Viewer will be saved to the USB disk / SD card connected to the cMT / cMT X HMI.

13.35.3.2. Configuration

Select “Enable [Operation Log] printing” check box and click [Settings] button to open the Operation Log Printing dialog box.



General Tab



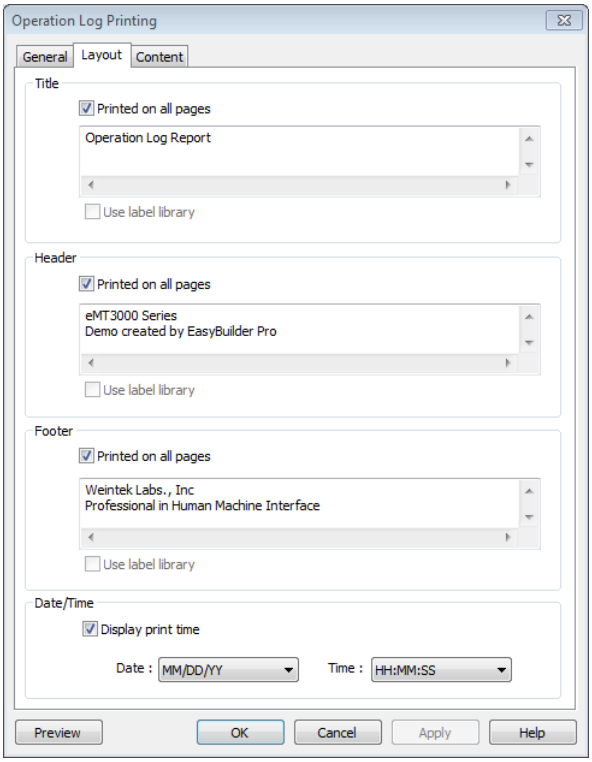
Setting	Description
Printer	Select the device to save the Operation Log sheet. If a printer is selected, the paper size should be A4. If an external device is selected, the Operation Log sheet is saved as a JPEG file. The system generates a folder named "operationlogsheet", and the files

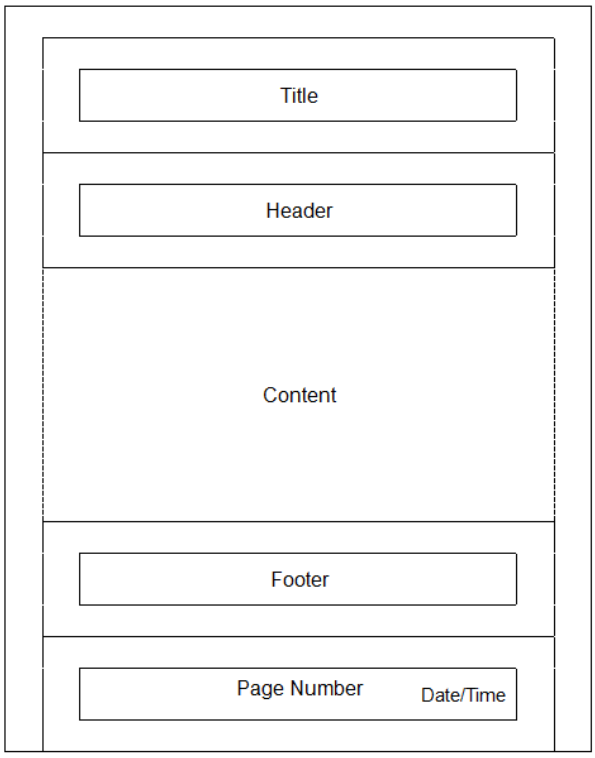
	saved in the folder are named "*print date_sequence number*". For example, the first JPEG file saved on 2013/05/08 is named 130508_0000 and so on.
Orientation	Sets the layout of the Operation Log sheet to be horizontal or vertical.
Font	Sets the font and the font size to of the Operation Log sheet. The following table lists the corresponding size.

Size	Title	Content
Large	20 pt.	16 pt.
Middle	16 pt.	12 pt.
Small	12 pt.	8 pt.

Range	Sets the range of the Operation Log to be included in the sheet. **Date** Sets the range by date, counted from the start day through the number of days entered. The maximum available range is 30 days. **Record** Sets the range by the number of records. The maximum available range is 10000 records.
Trigger address	Sets the register to control Operation Log Printing. When the register is set ON, it starts printing. When the printing is done, the register is set OFF automatically.
Preview	Preview the result before generating the Operation Log sheet.

Layout Tab

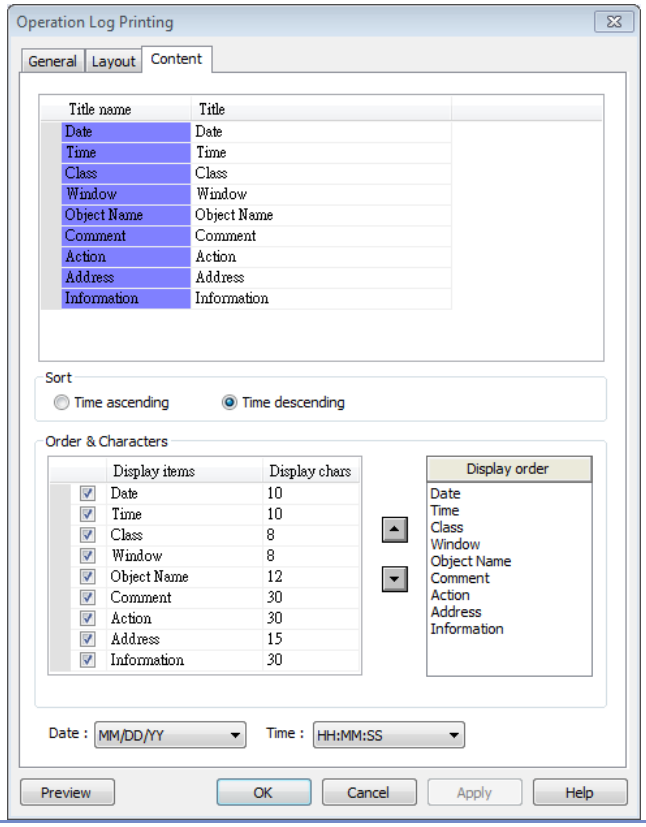




The layout of each part is shown in the above figure.

Setting	Description
Title	Sets the content of the title. The title is limited to one line. **Printed on all pages** If selected, the title is shown on each page; otherwise, the title is shown on the first page.
Header	Sets the content of the header. The header can have 5 lines in maximum. **Printed on all pages** If selected, the header is shown on each page; otherwise, the header is shown on the first page.
Footer	Sets the content of the footer. The footer can have 5 lines in maximum. **Printed on all pages** If selected, the footer is shown on each page; otherwise, the footer is shown on the last page.
Date/Time	If selected, the date/time the in the sheet is shown on the lower-right corner of each page; otherwise, the date/time is not shown.
Page number	Shown on each page.

Content Tab



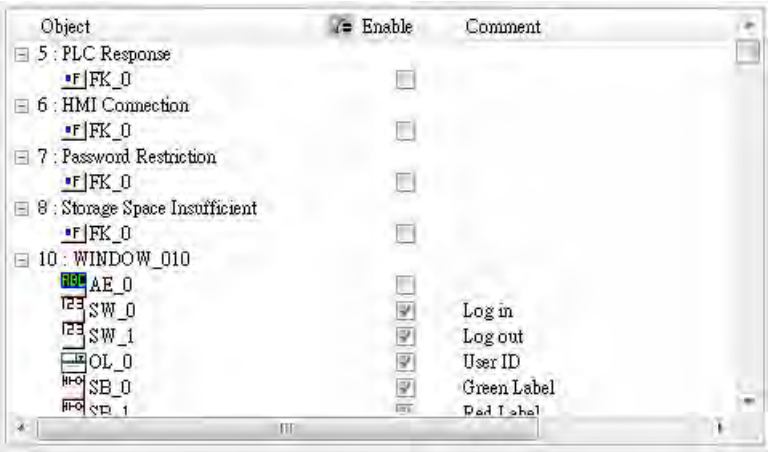
Setting	Description
Title	Sets the title displayed.
Sort	**Time ascending** The latest record is placed at the bottom. **Time descending** The latest record is placed at the top.
Date/Time	Sets the format of date and time displayed.

13.35.3.3. Demonstration

Example 1

The following demonstration explains how to create an Operation Log project.

1. Create a Toggle Switch object and a Numeric object on window number 10.
2. Go to Operation Log Settings; enable the Toggle Switch object and Numeric object on window number 10.



3. Create an Operation Log View object and finish relevant settings.
4. Run off-line simulation, trigger Toggle Switch and Numeric object. Operation Log is displayed by Operation Log View object.

Date	Time	Object ID	Action	Address	Information
04/05/17	13:48:22	TS 0	Toggle	Local HMI : LB-0	bit set OFF->ON
04/05/17	13:48:19	NE 0	Set word	Local HMI : LW-0	write 0->0

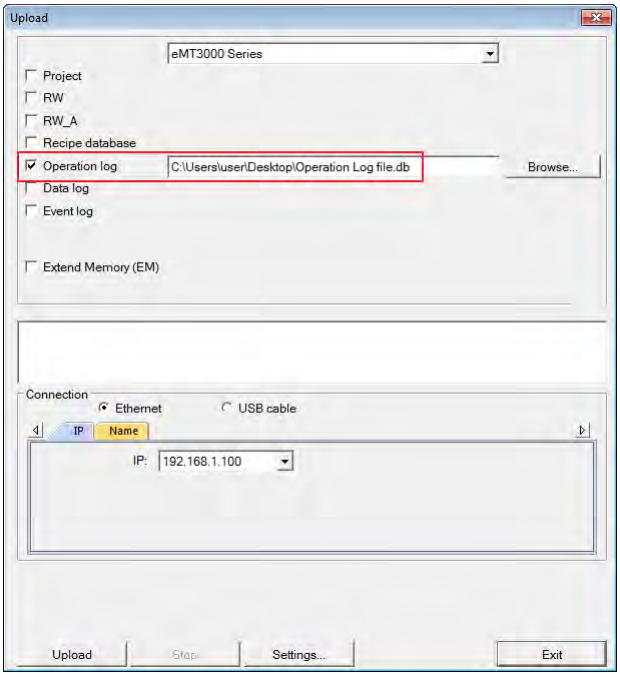
Click the icon to download the demo project. Please confirm your internet connection.

Example 2

Upload Operation Log to PC by using Utility Manager or use Backup object to send the file by email.

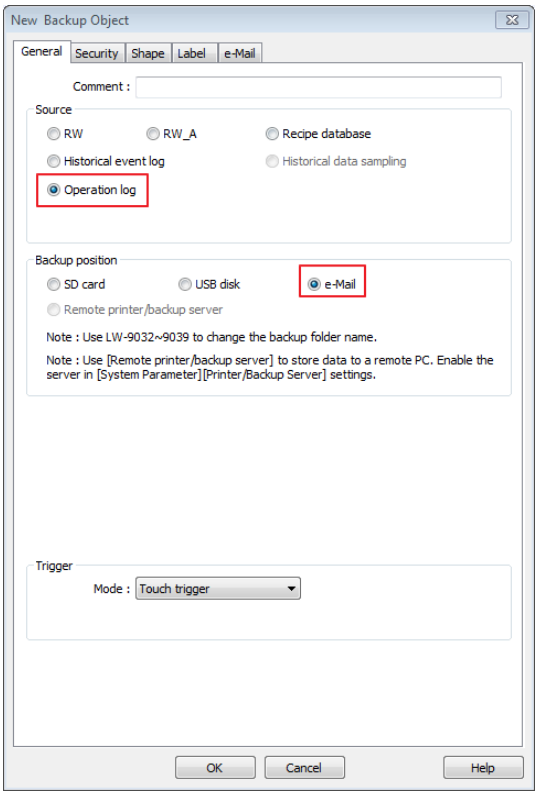
- Upload by Utility Manager

1. Open Utility Manager, click [Upload].
2. Select [Operation log], enter file name and HMI IP, and then click [Upload].



- Send the sheet by e-mail

1. Open [System Parameter Settings] » [e-Mail] tab. Set e-mail server and the address of recipient and sender.
2. Create a Backup object, under [Source] select [Operation log], and under [Backup position] select [e-Mail].



For more information about e-Mail settings, see “5 System Parameter Settings”.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.36. Combo Button

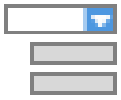
13.36.1. Overview

Combo Button can execute multiple commands. The former way was to overlay multiple objects in the same position, and the commands are executed in the order of the layer of the objects. This takes time to test the order when planning the project. Combo Button allows users to easily set multiple commands with one object, and freely adjust the order of executing commands.

The following are the features of Combo Button:

- Executes multiple commands.
- Allows adjusting the order of executing multiple commands.
- Displays the state in Bit or Word Lamp.

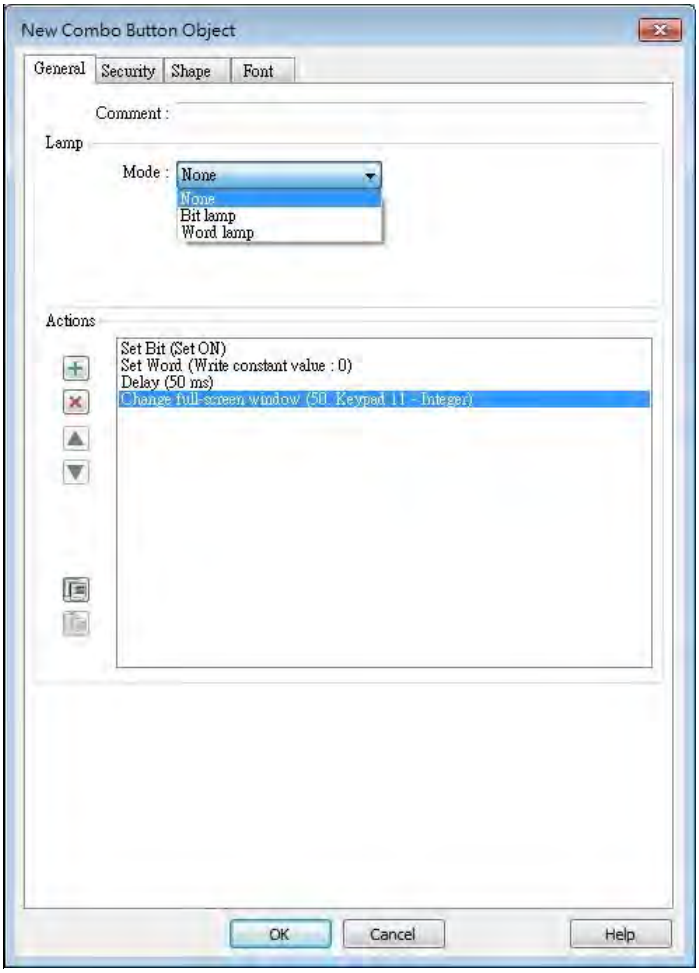
13.36.2. Configuration



Click on the Combo Button icon on the toolbar or select [Objects] » [Combo Button] to create object. Configure object properties, click OK, a new Combo Button object will be created.

13.36.2.1. eMT, iE, XE, mTV

General Tab



Setting	Description
Lamp	The mode to display the state of a designated bit or word register. **None:** Not using lamps to show states. **Bit Lamp** Displays the state of a designated bit address. [Invert Signal] Reverses the display of ON / OFF states. For example, if [Invert signal] check box is selected, when the designated bit is OFF, the object displays ON state. **Word Lamp** Displays the state according to the value of a designated word register. [No. of state]: The number of states used by the object. The state is numbered from 0, so the number of states minus 1 will be the state number. If the value in the word register is ≥ [No. of states] defined

in Attribute, the highest state will be displayed.
If the number of states is set to 8, the valid states will be 0, 1, 2, …, 7. In this case if the word value is 8 or higher, the system will display the state 7 shape.

Actions

A combo button can execute up to 20 actions.

Key	Description
+	Add actions.
×	Delete the selected action.
▲ ▼	Change the order of the actions.
	Copy the selected action.
	Paste the copied action.

Add

Delay

Delays the action for a few milliseconds. A combo button can set one [Delay] action only.

Set Bit

Sets the designated bit ON or OFF.

Set style	Description
Set ON	Set ON the designated bit of the device.
Set OFF	Set OFF the designated bit of the device.
Toggle	Alternates the bit state each time pressed.

Set Word

Sets the value in the designated register.

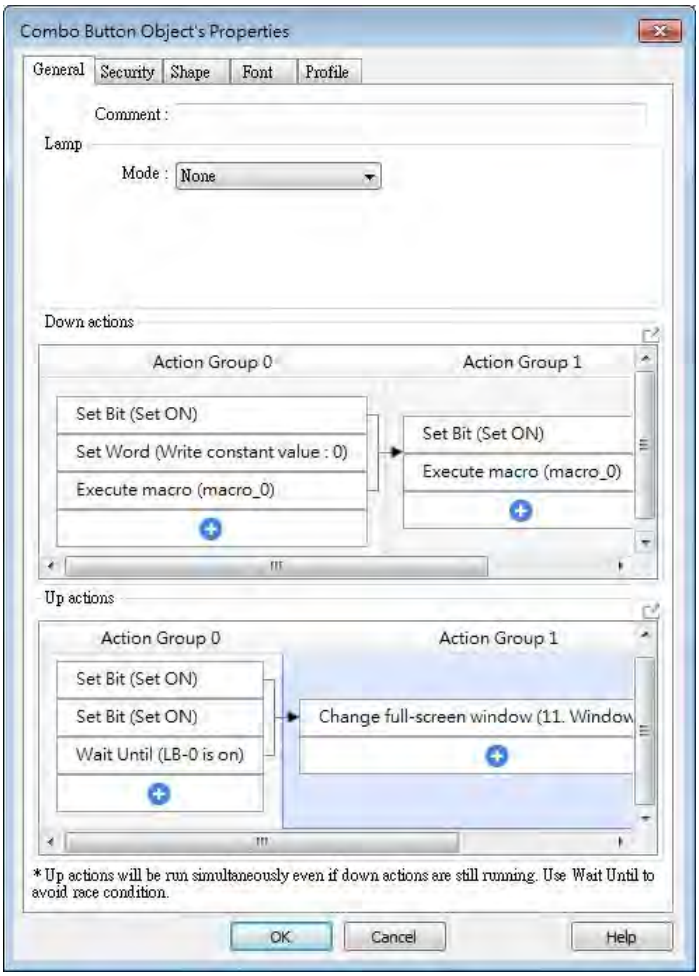
Set style	Description
Write Constant Value	Writes a constant value to the designated register.
JOG+	Increases value in register by a set amount in [Inc. value] each time when the button is pressed, to the [Upper limit].
JOG-	Decreases value in register by a set amount in [Dec. value] each time when the button is pressed, to the [Bottom limit].
Dynamic limits	Sets the Upper / Bottom limit by a designated register. When Dynamic Address is LW-n, where n is an arbitrary number, set upper limit when using [JOG+], and bottom limit when using [JOG-].

Change Window

Switch to the designated window.

13.36.2.2. cMT / cMT X Series

General Tab



Bit/Word Lamp actions are explained in 13.36.2.1 eMT, iE, XE, mTV.

For cMT Series, actions can be classified into action groups, whichs are put in sequence. Actions within a group are executed at the same time, and after the actions in the current group have been all triggered, the actions in the next group will then be triggered. Please see Ch13.53 Action Trigger for more information on the notes about action groups.

Setting	Description
Down actions	Execute action when the button is pressed.
Up actions	Execute action after the button is pressed and released.
Delay	Delay action for the specified time (ms).
Set Bit	Sets the designated bit address ON or OFF. **Set ON** Sets ON the designated bit. **Set OFF** Sets OFF the designated bit.

	Toggle Alternates the bit state. **Momentary** Momentary is a two-step action where pressing the button (Down action) will set the bit ON and then releasing the button (Up action) will set the bit OFF. Corresponding Down and Up actions will be created.
Set Word	Changes the value in the designated word address. **Write constant value** Writes the constant value to the designated register. **Increment value (JOG+)** Increases value in register by a set amount in [Inc. value], up to the [Upper limit]. **Decrement Value (JOG-)** Decreases value in register by a set amount in [Dec. value], down to the [Bottom limit]. **Dynamic Limits (JOG+, JOG-)** Sets the [Upper limit](JOG+) and [Buttom limit](JOG-) by a designated register. **Write constant string** Writes the constant string to the designated register.
Change window	Switches to the designated window. **Change full-screen window:** Changes to another base window. **Change common window:** Changes common window. **Return to previous window:** Changes from current screen to the previous one displayed. For example, when window no. 10 is changed to window no. 20, this function can be used to return to window no. 10. This function is only available for base window. **Animation Setting:** The effects are: Fade, Fly, Float, Wipe, Split, Circle, Clock, Zoom, Turn, Push. Different effects may be used for Start (window appears) and End (window disappears). [Duration] specifies how many milliseconds (ms) a transition effect takes to complete. [Direction] The direction of the transition.
Execute Macro	Executes one of the Macros from the drop down list that has already been configured by users. In an action group, [Execute Macro] is seen as completed as soon as it is triggered. Ever if a

	particular macro takes longer to complete, the system will not wait for the macro to complete running before moving on to the next action group.
Popup Window	Opens a designated window.
Close Window	Closes currently opened window.
Keyboard Input	Configures the button as a keypad key, and the character it enters, via [Numeric] or [ASCII] objects. **Enter:** Same as the keyboard's "Enter" function. **Backspace:** Same as the keyboard's "Backspace" function. **Clear**: Clear the value in the word register. **Esc:** Same as the [Close window] function; it is used to close the keyboard window. **Delete:** Same as the keyboard's "Delete" function, deletes the number or character on the right side of the text cursor. **Left:** Same as the keyboard's "←" key moves the text cursor to the left side of the previous number or character. **Right:** Same as the keyboard's "→" key moves the text cursor to the left side of the next number or character. **Inc:** Increment by 1. **Dec:** Decrement by 1. **ASCII/UNICODE:** Specify the character to be entered by this key.
Screen Hardcopy	Saves current screen as a hard copy file into a SD card or USB disk.
Acknowledge all events (Alarms)	Acknowledges all events once.
Import Data	Imports e-mail contacts or user accounts, or is used as USB Security Key login. **Data Position:** [SD card] or [USB disk]. **Account import mode:** If [Overwrite] is selected, the existing accounts will be overwritten by new accounts. If [Append] is selected, accounts are appended. **Delete file after importing user accounts:** The system will delete the account data saved in the external device after importing; this can prevent the account data from

	leaking out.
Wait Until	The next group will be executed only when the condition set for a designated bit or word address is met.
Data Transfer (Global)	Sends the value in the designated address to another address.
File Transfer	Transfer files by FTP. HMI will connect to the FTP server in passive mode. **General Tab:** Download: Transfer file from FTP server to local HMI. Upload: Transfer file from local HMI to FTP server. **Server address:** [Direct]: Designate server address related settings in EBPro. [Dynamic]: Designate an address for changing the server address related settings dynamically on HMI. Suppose the dynamic address is set to LW-n, then: LW-n: Host usage (0: IP, 1 Domain name) LW-n+1: IP (4 words) LW-n+5: Port LW-n+6: Authentication (0: None, 1: Account) LW-n+7: Username (16 words) LW-n+23: Password (16 words) LW-n+39: Domain name (64 words) **Delete the source file after successful transfer** Following successful transfer, the source file will be deleted. When transferring in [Download (FTP -> HMI)] mode, please make sure that the account has the write permission on the FTP server. **File Tab:** Set file position and full path of FTP server / Local HMI. When a folder path is specified, all the files in that folder will be transferred, not including files in subfolders. When a file with identical file name already exists, it will be overwritten regardless. **Status Tab:** Designate an address for showing file transfer result and FTP server response. Please use the following link for more information on FTP server return codes. https://en.wikipedia.org/wiki/List_of_FTP_server_return_codes

Note

- One Combo Button can only do one of these window actions, and only once: [Change Window], [Popup Window], and [Close Window].
- At most 20 groups can be created in a Combo Button, and at most 20 actions can be added into a group.
- File Transfer Error Codes:

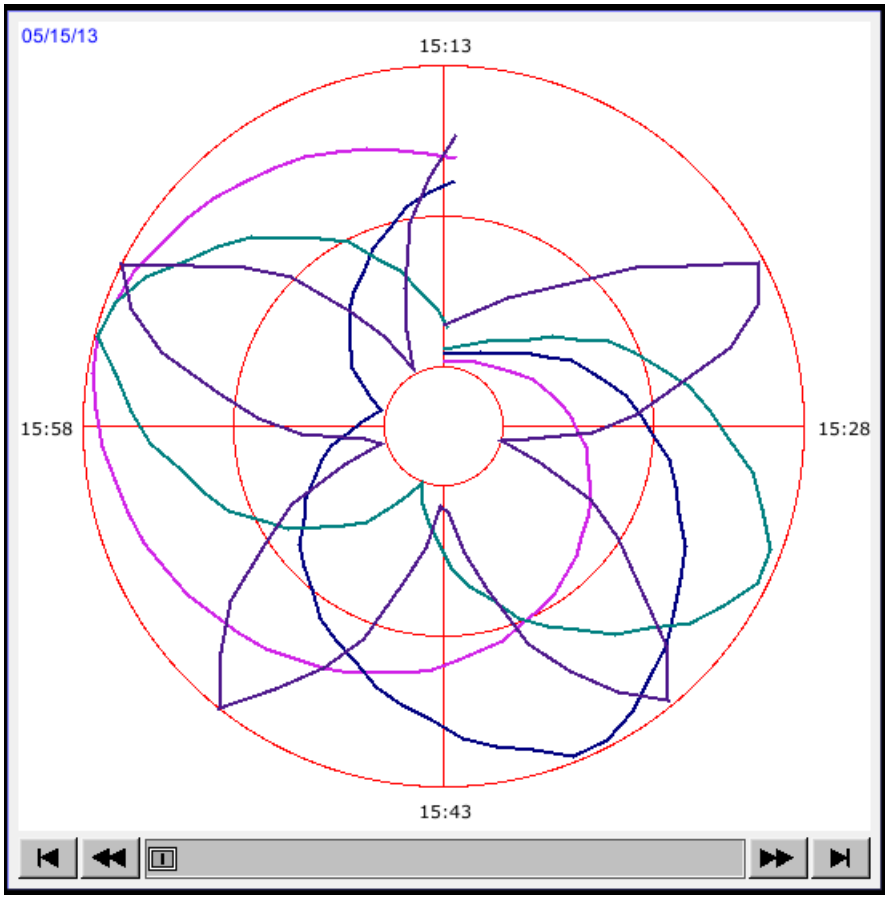
Error Code	Description
0	File transferred successfully.
1	The HMI directory for download does not exist.
3	USB disk or SD card is not found.
4	HMI or FTP directory is empty.
5	Uploaded file does not exist.
8	Operation rejected by FTP server.
9	USB disk or SD card is full.
10	Unknown error.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.37. Circular Trend Display

13.37.1. Overview

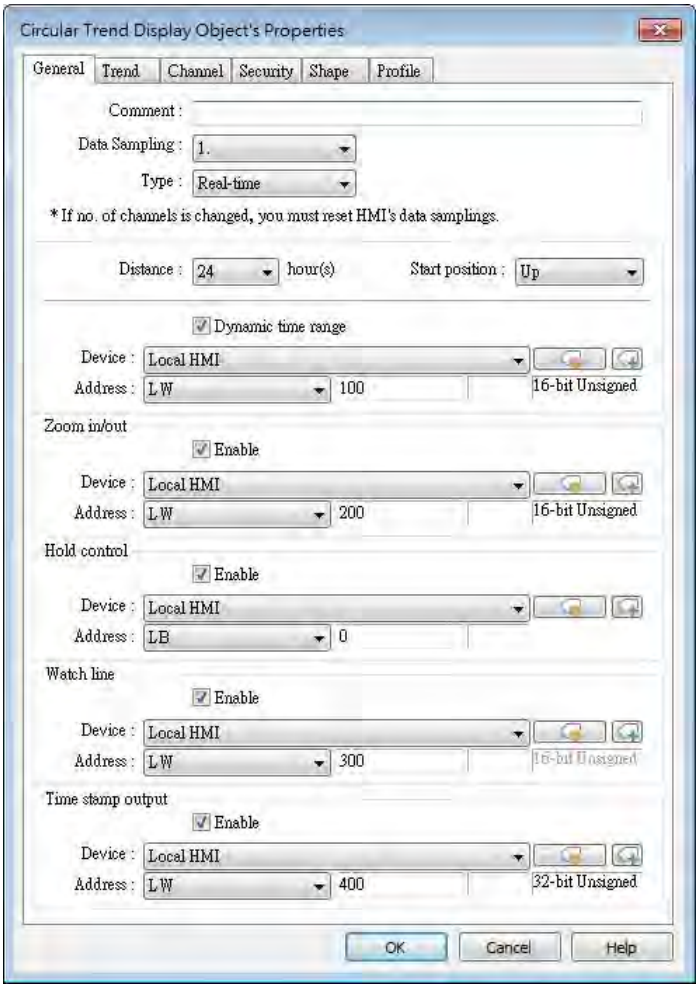
Circular Trend Display object draws the trend curve of Data Sampling in a polar coordinate system, where y-axis represents the radial coordinate and the x-axis represents the angular coordinate. The way to use this object is similar to using Trend Display object.



13.37.2. Configuration

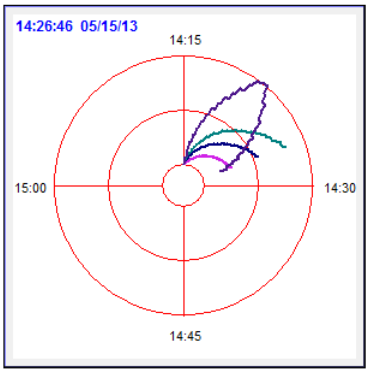
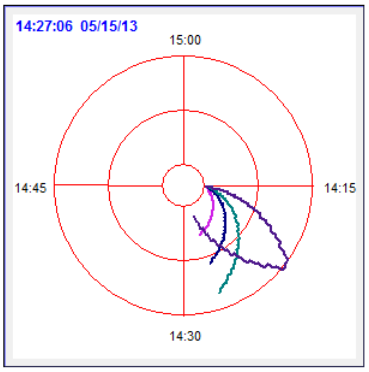
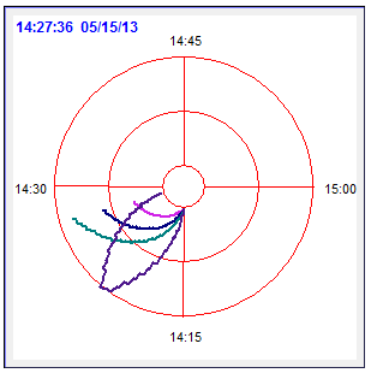
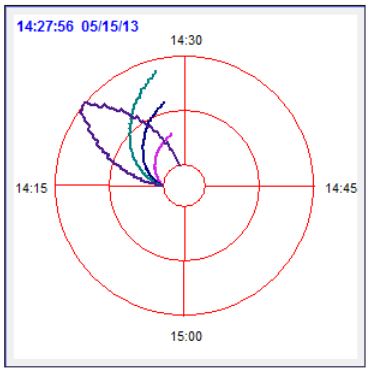
Click the Circular Trend Display icon on the toolbar to open the property dialog box. Set up the properties, press OK button, and a new Circular Trend Display object will be created.

General Tab



Setting	Description
Data Sampling	Selects the data source for drawing the trend curve.
Type	Selects the type of the trend from [Real-time] or [History]. **Real-time** In this mode, it displays a fixed number of sampling data from the moment HMI starts to present. The number of sampling data is determined by the [Max. data records (real-time mode)] setting of Data Sampling object. If the number of sampling data exceeds this number, the earlier data will not be displayed. To display earlier data or the data in other days, please select [History] mode. [Hold control] address can be used to pause refreshing the display. This only stops displaying new data in the Circular Trend Display object, and the data is still being sampled by Data Sampling object.

	History In this mode, it displays the sampled data sorted by date. Select the data source from [Data Sampling], and then use [History control] address to view the records of different dates. **Note** If [Show scroll control] check box in Trend Tab is not selected, the earlier data cannot be viewed when exceeding the specified [Distance]. For example: Set [Distance] to 1 (hour.), then sampling data earlier than one hour is not displayed.
Refresh data automatically	If enabled, the window in which the Circular Trend Display object (in history mode) is placed will be refreshed once per second. ● The scroll controls can be used to check the refresh status. If button is displayed, the Circular Trend Display will be automatically refreshed. If button is displayed, the Circular Trend Display will stop being refreshed. ● Scrolling backward and viewing earlier data will disable [Refresh data automatically]. The button displayed is at this moment. ● If [Refresh data automatically] is selected, the display is refreshed when change back to this window, regardless of the use of scroll controls. Example: If [Refresh data automatically] is selected, scrolling to the earlier display stops auto-refresh. At this moment change to another window and then change back, the Circular Trend Display is still refreshed. If [Refresh data automatically] is not enabled when building the project, to enable it directly on HMI, simply press . Please note that auto-refresh remains disabled after window change.

Distance	Defines the time length that the circle represents, the unit is hour, the range is 1 to 24 (hours).
Start position	The position from where to start drawing the trend curve. (Up)  (Right)  (Down)  (Left) 
Dynamic X-axis time range	If selected, a word register can be designated to control the time length of Circular Trend Display. The unit is hour. If no value is entered, the distance is set to default.
Zoom in / out	Enlarge or reduce the size of the object. The maximum size is 10 times larger. If 0 is entered in the designated register, the effect equals entering 1, the object remains the original size.
Hold control	When the register is set ON, suspend the update of Circular Trend Display. It does not stop the sampling process of Data Sampling object. This setting is available only in Real-time mode.
History control	History data is sorted by date. The system uses [History control] to select the history data that are created in different dates and then displays it by Circular Trend Display object. If the value of the designated register in [History control] is 0, the Circular Trend Display object displays the latest

record. If the value is 1, the second latest record is displayed and so on. This setting is available only in History mode.

If use with Option List object and select data source as [Dates of historical data], the history data will be sorted by date and displayed in Option List object, see “13.29 Option List”.

In the following example, when history control address is set to LW-n, and there are 4 sampling data: 20061120.dtl, 20061123.dtl, 0061127.dtl, 20061203.dtl. The corresponding data selected by the value in history control address is as the following list.

Value in LW-n	The sampling data displayed
0	20061203.dtl
1	20061127.dtl
2	20061123.dtl
3	20061120.dtl

Watch line

Displays a watch line when user touches the Circular Trend Display object, and the sampling data at the position of the watch line is output to the designated register. To display sampling data with multiple channels, the system consecutively writes the data of each channel to the designated word register and the following registers. If the data format of each channel is different, the channels are sorted by the data format of its corresponding register.

In the following example, when watch address is set to LW-n, and there are 4 sampling data, the format of each data is: 16-bit Unsigned, 32-bit Unsigned, 32bit Signed, and 16-bit Signed. The corresponding watch address is as the following list.

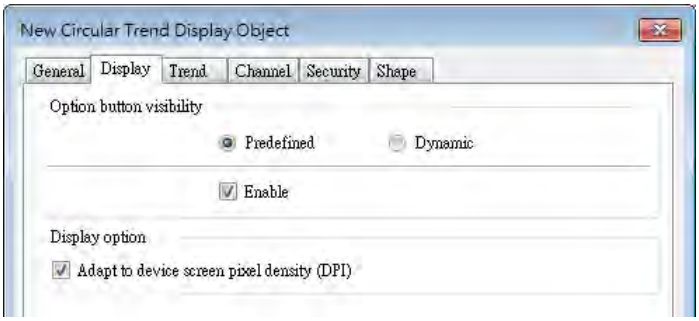
Channel	Data Format	Data Length	Watch Address
0	16-bit Unsigned	1 Word	LW-n
1	32-bit Unsigned	2 Words	LW-n+1
2	32-bit Signed	2 Words	LW-n+3
3	16-bit Signed	1 Word	LW-n+5

Time stamp output

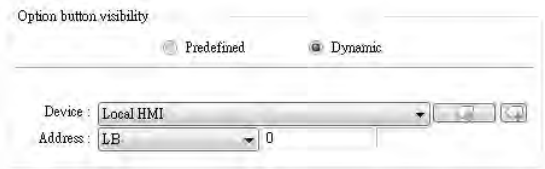
If selected, the system will start counting time from the first data sampled, and output the elapsed time counted of

	the latest data sampled to the register designated in [Time stamp output + 2]. When pressing a point on the trend curve, the relative time of the nearest data sample is then output to [Time stamp output address]. **Note** The format of the register designated in [Time stamp output] and [Time stamp output + 2] must be 32-bit. [Time stamp output + 2] is only available for Real-time mode while [Time stamp output] is available for Real time mode and History mode.

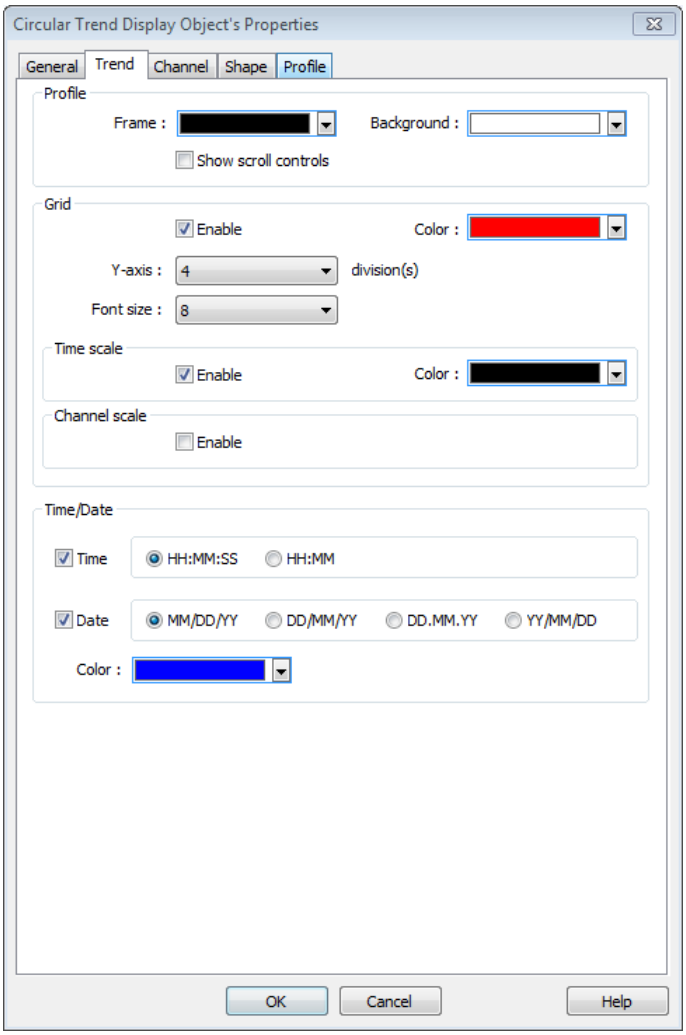
Display Tab

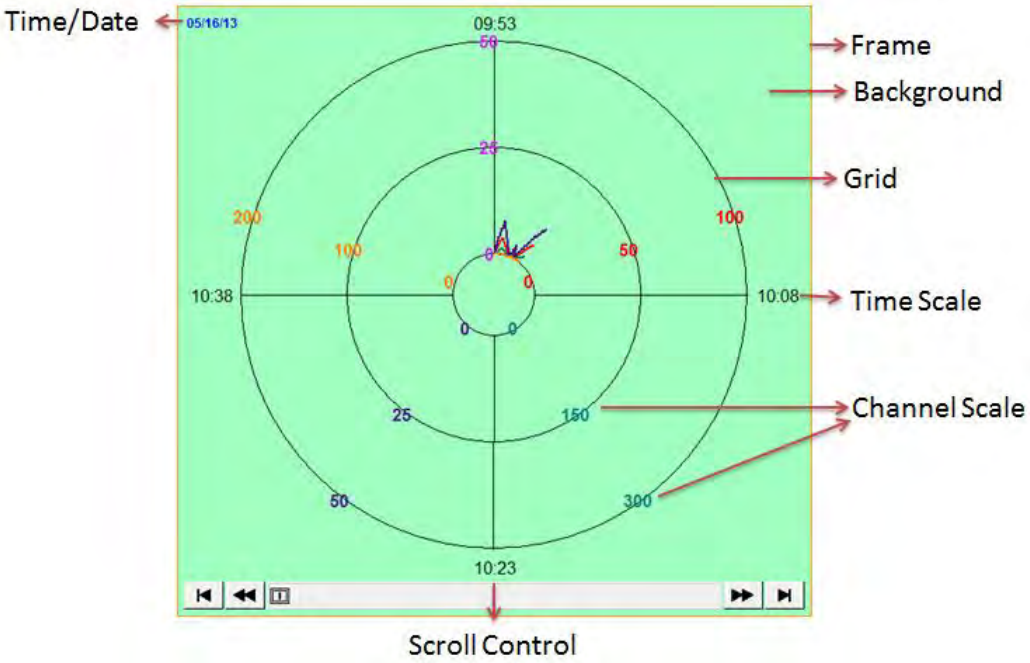


Display settings in this page are only supported on cMT / cMT X Series models.

Setting	Description
Option button visibility	Show or hide the option button . **Predefined** Select [Enable] to show the option button. **Dynamic**  Designate a bit address to control option button visibility.
Display option	**Adapt to device screen pixel density (DPI)** With this option selected, text size, position, and the spacing between Y axes will be adjusted according to the DPI of the screen of the connected device.

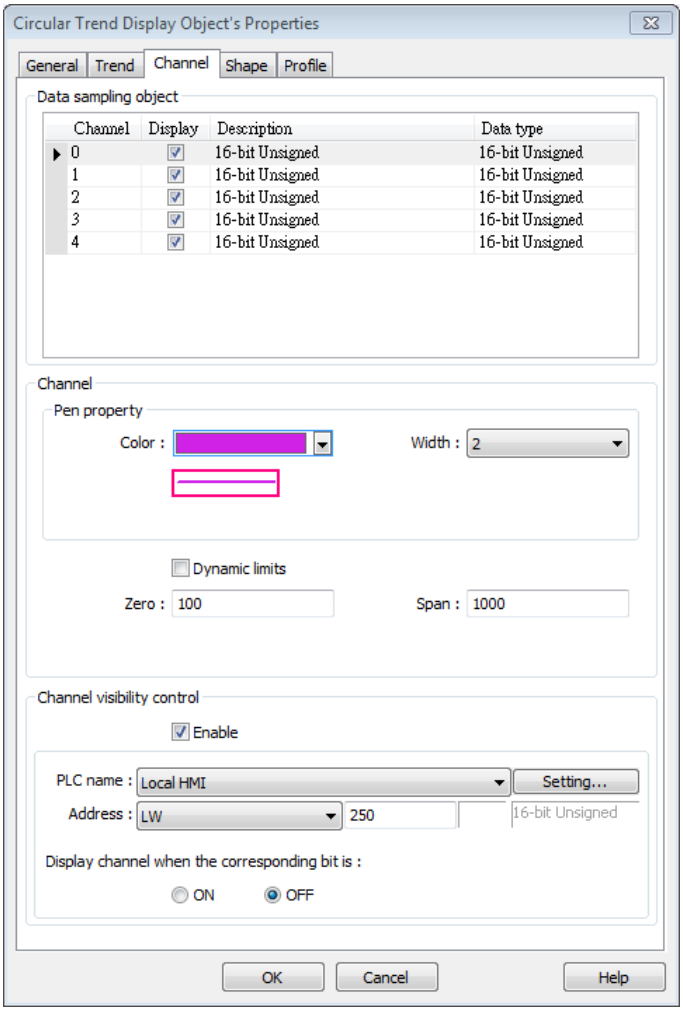
Trend Tab


Circular Trend Display Object's Properties
General
Trend
Channel
Shape
Profile
Profile
Frame :
Background :
Show scroll controls
Grid
Enable
Color :
Y-axis :
4
division(s)
Font size :
8
Time scale
Enable
Color :
Channel scale
Enable
Time/Date
Time
HH:MM:SS
HH:MM
Date
MM/DD/YY
DD/MM/YY
DD.MM.YY
YY/MM/DD
Color :
OK
Cancel
Help


Time/Date
05/16/13
09:53
Frame
Background
Grid
10:38
10:08
Time Scale
Channel Scale
10:23
Scroll Control

Setting	Description
Profile	**Frame** Sets the color of the frame of the object. **Background** Sets the color of the background of the object. **Show scroll controls** The scroll controls are displayed for scrolling to view the sampling data of other time range. The minimum scrolling unit is determined by the setting of [Distance] in General tab. If [Show scroll controls] check box is not selected, the earlier data cannot be viewed when exceeding the specified [Distance]. For example: Set [Distance] to 1 (hour.), then sampling data earlier than one hour is not displayed.
Grid	Sets the number and the color of the dividing lines. **Y-axis** Sets the number of divisions on Y axis. **Font size** Sets the size of the font which labels the time scale or channel scale. **Time scale** If enabled, displays the time scale. When the time length is longer than 1 hour, the scale division is 1 hour. When the time length is set to 1 hour, the scale division is 15 minutes. **Channel scale** If enabled, displays the channel scale. The color of the font which labels the channel scale corresponds to the setting of the trend curve of each channel.
Time / Date	**Time** Sets the display format of time. **Date** Sets the display format of date.

Channel Tab



Setting	Description
Channel	Sets the style and the color of the trend curve, and the upper and lower limit of data that can be drawn on the trend curve. Up to 8 channels are supported simultaneously. **Not selecting [Dynamic limits]** The upper and lower limits of the data are set by constants. **Selecting [Dynamic limits]** The upper and lower limits are set by the designated register. When the address is LW-n, the corresponding addresses are as the following list. Data format: 16-bit / 32-bit Lower limit: LW-n / LW-n

	Upper limit	LW-n+1	LW-n+2

Channel Visibility Control

If [Enable] is selected, the bits of the designated word register will be used to show/hide each channel. First bit (Bit-0) controls the first channel; second bit (Bit-1) controls the second channel, and so on.

Display channel when the corresponding bit is:

If [ON] is selected, when the corresponding bit is OFF, the channel is hidden. If [OFF] is selected, when the corresponding bit is ON, the channel is hidden.

In the following example, the control address of channel visibility is set to LW-0 and each channel shows when the corresponding bit is OFF. If there are 5 channels, the visibility of the channels is as the following list.

Channel	Control address	Bit state	Display
0	LW_bit-000	OFF	YES
1	LW_bit-001	ON	NO
2	LW_bit-002	ON	NO
3	LW_bit-003	OFF	YES
4	LW_bit-004	OFF	YES

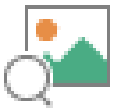
13.38. Picture View

13.38.1. Overview

Picture View object plays slideshow of picture files saved in an external device such as a USB drive or SD card.

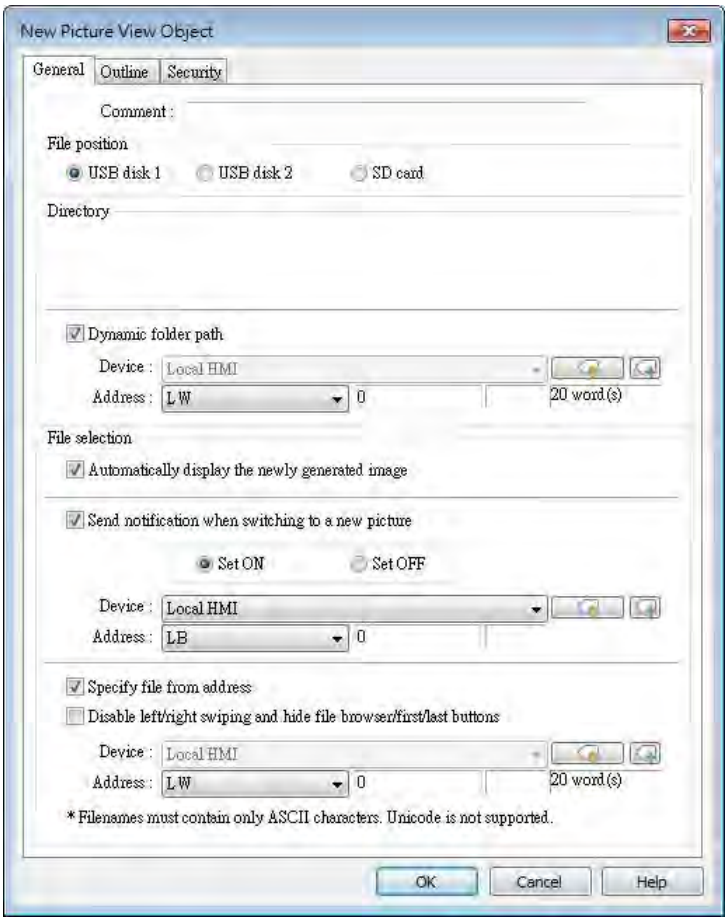
This object does not work remotely on cMT Viewer.

13.38.2. Configuration



Click the Picture View icon on the toolbar to open the property dialog box. Set up the properties, press OK button, and a new Picture View object will be created.

General Tab

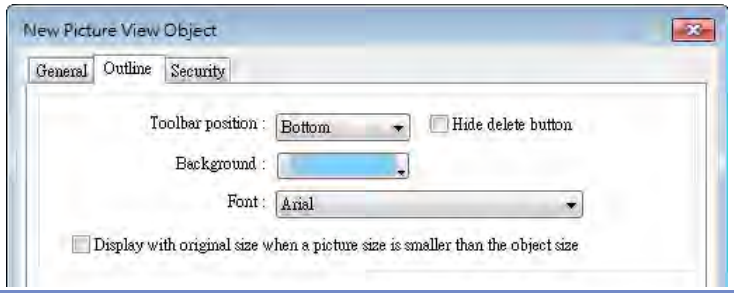


Setting	Description
File position	Select the file source of the picture files from [SD card] or [USB disk].
Directory	Specify the directory where the picture files are saved. The directory can be the one used for Video Input's capture feature or Function Key's screen hardcopy feature. The picture file saved in the external device can be immediately displayed. **Dynamic folder path** Designate folder path by a local address.
File selection	**Automatically display the newly generated image** When a new image is generated in the designated folder path, Picture View object will automatically display the new image. **Send notification when switching to a new picture** When [Automatically display the newly generated image] is selected, the state of the designated address changes to On/Off when the new picture is displayed on HMI. **Specify file from address (hide toolbar)** When enabled, the displayed picture is designated by a file name in a local address, and the toolbar will be hidden.

Note

- The file name must be all in ASCII characters, and the Unicode characters are not supported.
- The supported picture formats are: .jpg, .bmp, .gif, .png.
- The size of the picture that can be displayed on an iP/iE/eMT/XE/mTV model is calculated in the following formula:
  (image.width × image.height) < (screen.width × screen.height) × 4
  Pictures with sizes exceeding this range cannot be displayed on HMI.
- Two-finger zoom in / out gestures are supported on capacitive touchscreens.

Outline Tab



Setting	Description
Outline	Specify the position, background color, and font of Picture View object. (Supported only on iP/iE/eMT/XE/mTV models.) **Hide delete button** When selected, the delete button for deleting the viewed picture will be hidden in the Picture view object. **Display with original size when a picture size is smaller than the object size.** When selected, the picture will be displayed in its original size if it is smaller than the Picture View object. This can avoid distortion caused by enlarging the picture.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.39. File Browser

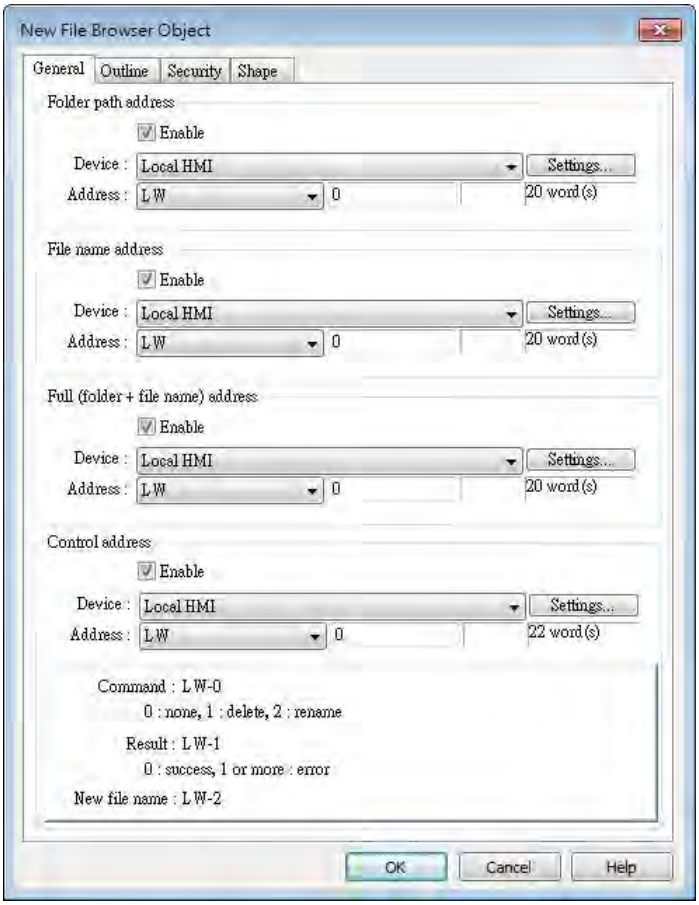
13.39.1. Overview

File Browser object can display filenames and their directories saved in the SD card or USB disk. Apart from browsing for the files in the external devices, the filenames and the directories selected in File Browser object can be written to the designated address.
This object does not work remotely on cMT Viewer.

13.39.2. Configuration

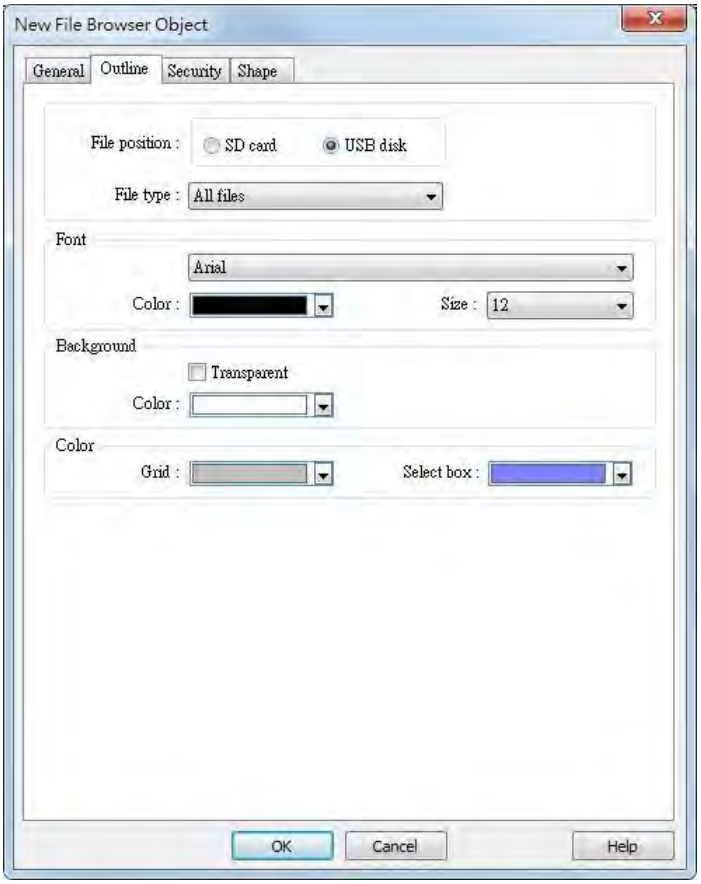
Click the File Browser icon on the toolbar, or select [Object] » [File Browser] to open a File Brower object property dialog box and set up the properties.

General Tab



Setting	Description
Folder path address	Current directory.
File name address	The file name of the currently selected file.
Full (folder + file name) address	The full directory and file name of the currently selected file.
Control address	Designate the control address used for deleting a file or changing file name in File Browser. **Command: Control Address** 0: None 1: Delete 2: Rename 3: Select (only supported on cMT / cMT X Series models) **Result: Control Address+1** 0: Success 1 or more: Error **New File Name: Control Address +2**

Outline Tab



Setting	Description
Folder position	Select the position of the file from SD card or USB disk.
File type	File types to be displayed. Options include all files, CSV files, or image files. For cMT / cMT X Series models, there is also a PDF file option for use in conjunction with PDF reader.
Font / Background / Color	Set the attributes and font of the object.

Note

- The file name and the directory of the selected file will be written to the designated address. To change file selection in File Browser to another file by changing the address, please enter the position of the file and then use the select command (=3).
- The system will read the folder path address and file name address when the HMI is restarted or when an external device is inserted to the unit. If valid data is can be read

from the designated address, the system will then automatically navigate to the appropriate directory and highlight the file according to the data read. If [Folder path address] is not enabled, the data at Full (folder + file name) address will be read.

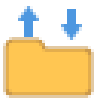
Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

13.40. Import/Export

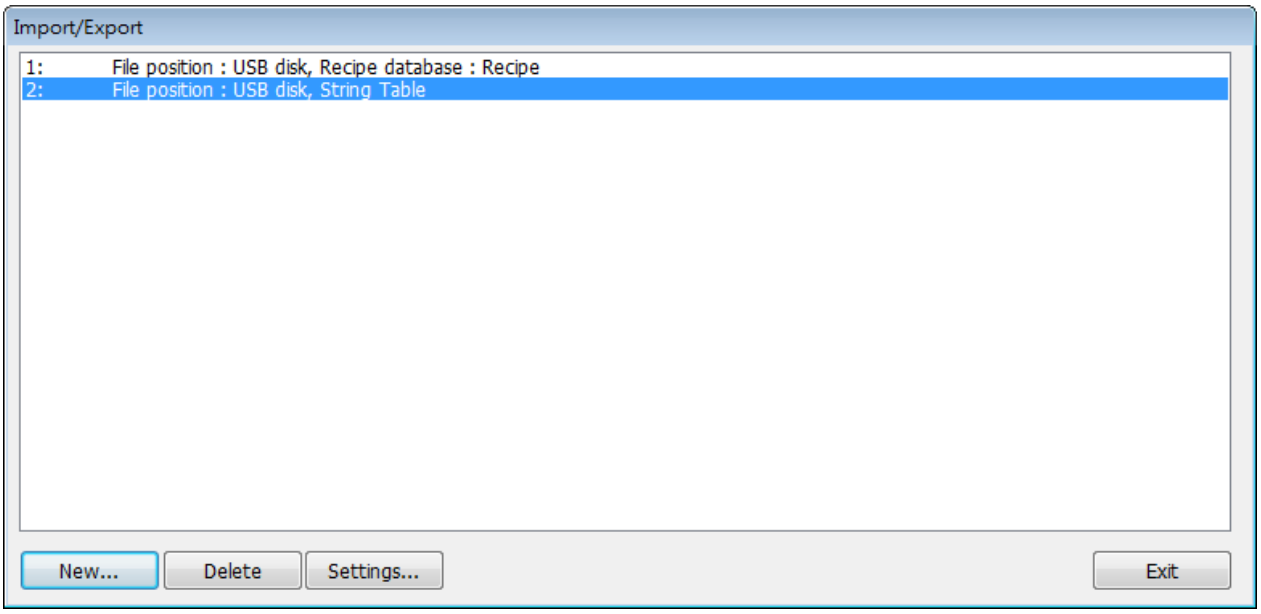
13.40.1. Overview

With Import/Export object, Recipe Database or String Table can be imported or exported.

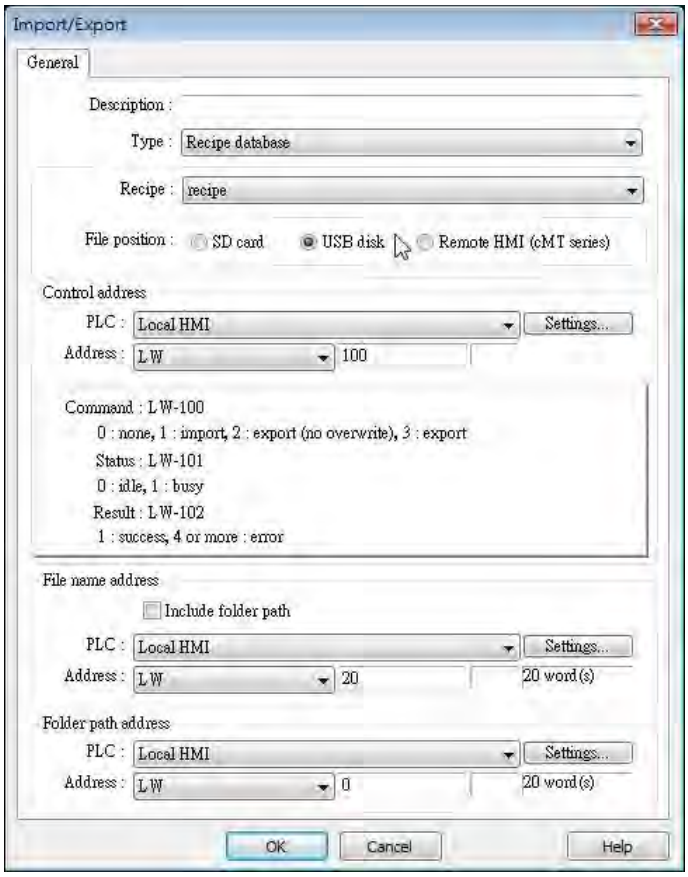
13.40.2. Configuration



Click the Import/Export icon on the toolbar to open the Import/Export Object management dialog box. To add an Import/Export object, click [New], set up the properties, press OK button and a new Import/Export object will be created.



General Tab



Setting	Description
Type	Select the file source from Recipe Database or String Table.
File position	Select the position of the file to be imported / exported from SD card, USB disk, or Remote HMI (cMT / cMT X Series). When Remote HMI is selected, please note that only files in cMT / cMT X Series models can be imported.
Recipe	Select the recipe. This option is hidden when select String Table.
Control address	Designate the control address used for performing import/export, or displaying the result. **Control: Control Address** Recipe Database: 0: None 1: Import 2: Export (no overwrite)

	3: Export String Table: 0: None 1: Delete 2: Import 4: Export (no overwrite) 5: Export **Status: Control Address+1** 0: Idle 1: Busy **Result: Control Address +2** 1: Success 4: The file already exists, no overwriting. Other: Error
File name address	The name of the imported/exported file. If [Include folder path] is selected, the full directory and file name will be included at this address.
Folder path address	The directory of the imported/exported file.
Remote HMI address	When the file position is [Remote HMI (cMT / cMT X Series)], please enter the remote HMI's IP address in this field.

Example 1

The following is an example on recipe export/import settings.

Field	Setting
File position	USB disk
Recipe	Recipe_A (or other recipe)
Control address	LW-100
File name address	LW-200
Folder path address	LW-250

1. Create two ASCII Input objects. Set address to LW-200 and LW-250 respectively.
2. Enter the file name in LW-200: 2015_recipe.csv.
3. Enter the folder path in LW-250: Setting.
4. Use a Set Word object to write value 3 to LW-100. Then, Recipe_A will be exported to the USB disk, in the "Setting/2015_recipe.csv" file.

Note

- When performing “Export (no overwrite)” command, if the target file already exists, the export operation will be canceled, and the result value will be set to “4”.
  The following lists the result values and the information.

Result (HEX)	Information
0x1	Success.
0x4	File already existed and will not overwrite.
0x10	Invalid command.
0x100	Data contains non-numeric data.
0x101	Path contains invalid string “..”.
0x102	Communication error while updating Recipe DB.
0x103	Error while reading Recipe DB information from project file.
0x200	General exception.
0x201	General status error.
0x202	Import to unknown database type.
0x203	Error while validating Recipe DB table definition.
0x204	Error while validating Recipe DB table data.
0x205	Error while writing Recipe DB table definition.
0x206	Error while writing Recipe DB table data.
0x300	File error: Unknown error.
0x301	File error: Empty file name.
0x302	File error: The external device does not exist.
0x303	File error: Invalid file name (directory or special files), or a folder with the same name already exits.
0x304	File error: Unable to remove file.
0x305	File error: Open file stream error.
0x306	File error: Unhandled BOM.
0x307	File error: Error while parsing CSV file (incorrect formats).
0x308	File error: Insufficient space on the external device.
0x309	File error: Unable to find the file.
0x30A	File error: The CSV file contains over 10000 data rows.
0x400	Database general exception.

0x401	Database error: Unable to open table.
0x402	Database error: Unable to get rows.
0x403	Number of columns in CSV file and in Recipe DB do not match.
0x501	Unable to connect remote HMI.
0x503	Import from remote HMI database is prohibited.
0x504	Import from remote HMI database is not supported.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.41. Pie Chart

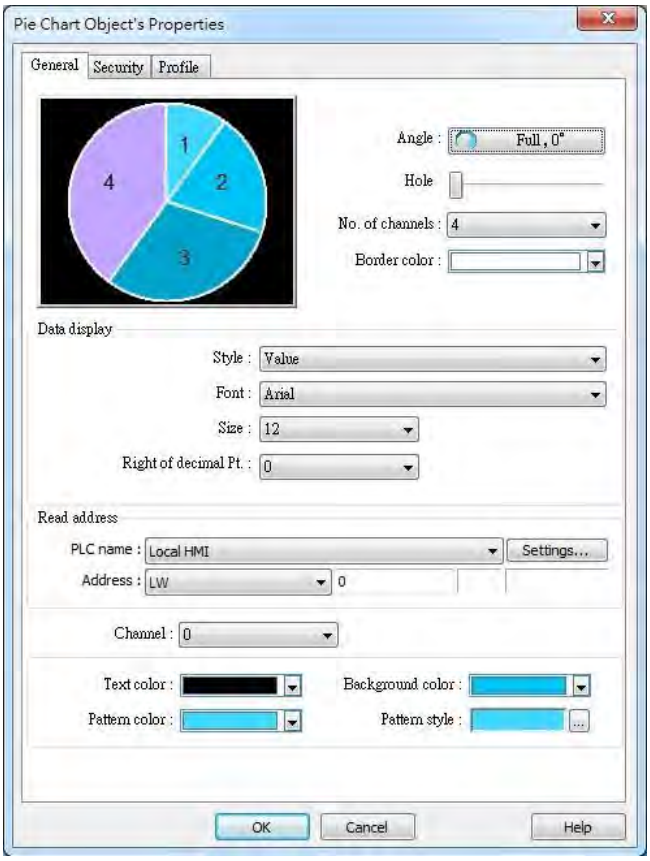
13.41.1. Overview

The Pie Chart object draws a pie chart that is divided into slices to illustrate numerical proportion, according to the value of the designated read address.

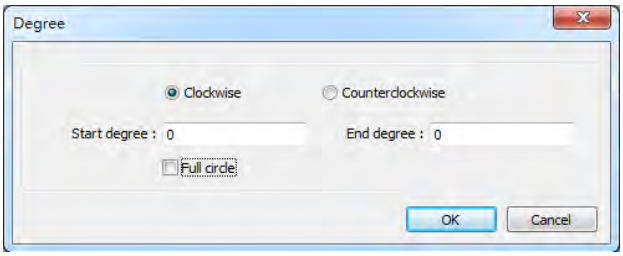
13.41.2. Configuration

Click the Pie Chart icon on the toolbar to open the property dialog box. Set up the properties, press OK button, and a new Pie Chart object will be created.

General Tab

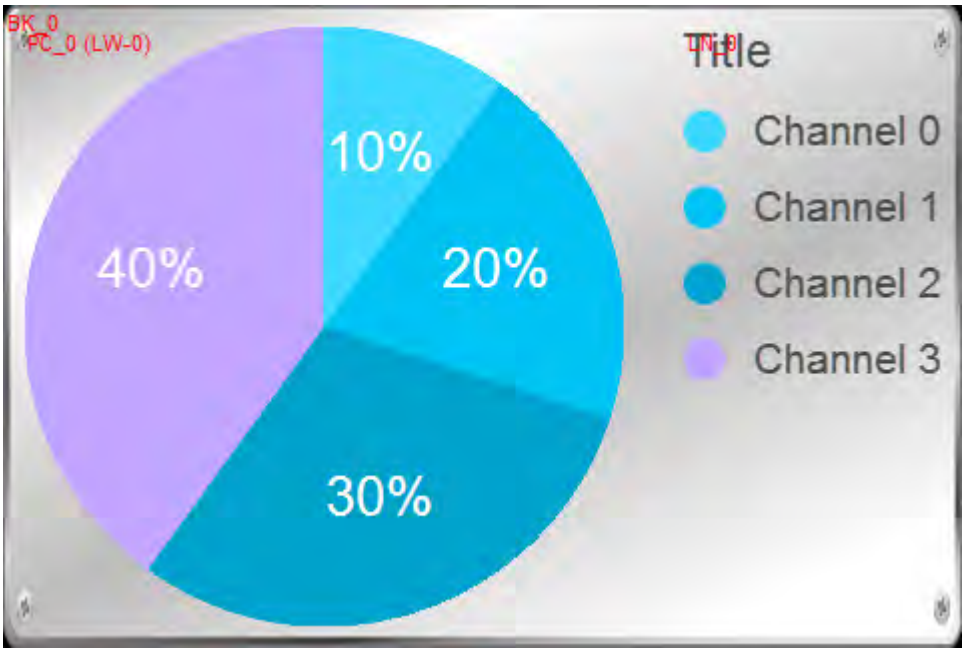


Setting	Description
Angle	Set the [Start degree] of the chart. Choose the Chart to be [Clockwise] or [Counter clockwise]. If [Full circle] isn't selected, then [End degree] must

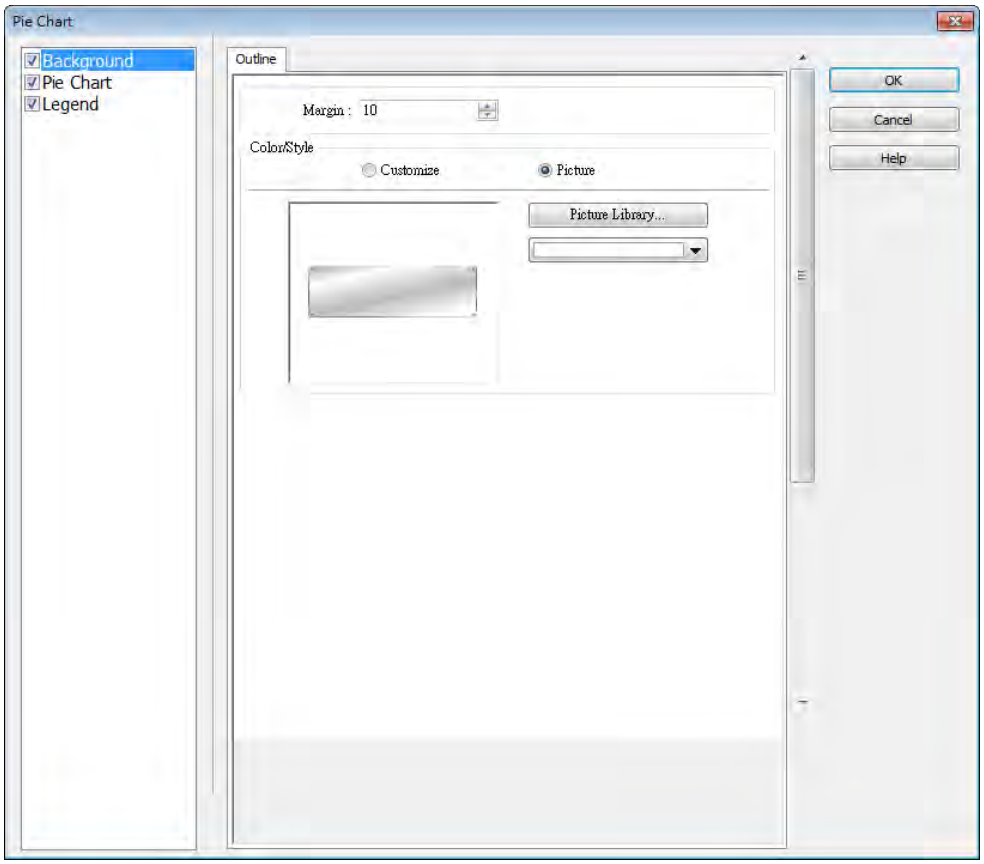
	be set. 
Hole	Set the size of the hollow circle in the center of Pie Chart.
No. of channels	Set the number of channels to be displayed in the chart. The range is from 2 to 16.
Border color	Set the color of the border.
Data display	The [Style] can be set as [None], [Value] and [Percentage]. The [Font] and [Size] are for the text on the chart. For Value Style, the [Right of decimal Pt.] can set the value on the chart to be displayed with the decimal point. The [Right of decimal Pt.] option is only available for [Value] style.
Read Address	The address is for channel 1. The following consecutive addresses are for the rest of the channels. For example, if the Read Address is LW-0, then the Read Address for channel 2 is LW-1; channel 3 is LW-2…and so on.
Channel	Set the [Text color], [Background color], [Pattern color], and [Pattern style] of the selected channel. The [Background color] is for the [Pattern style] that has background. If the [Pattern style] doesn't have a background, then the [Background color] doesn't need to be set.

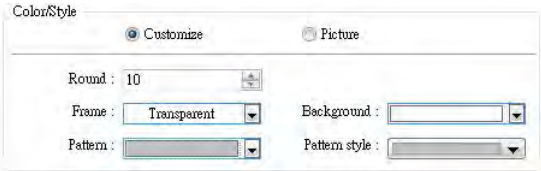
13.41.3. Combo Setting

cMT / cMT X Series HMI support combo setting for Pic Chart, which allows setting of multiple related objects at a time. Pic chart can be set with Background and Legend.

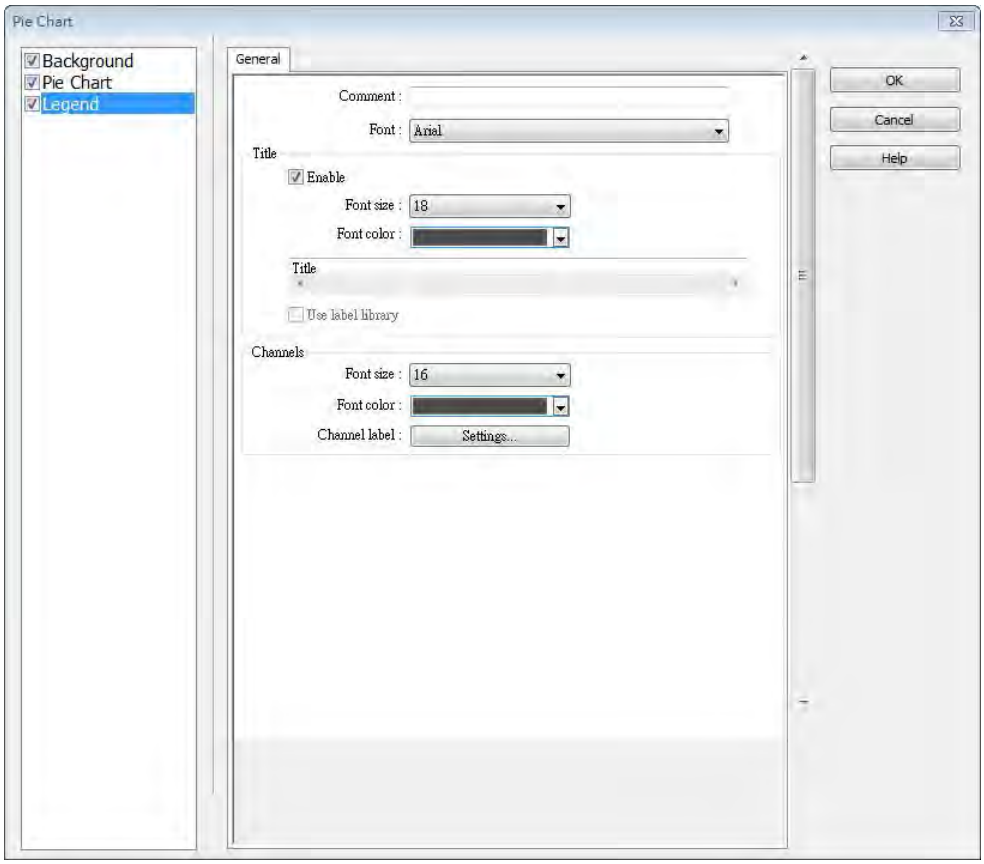


Background



Setting	Description
Margin	Specify the space between the background edge and the objects.
Color/Style	**Customize**  Select a suitable background pattern and color. **Picture** Use the default picture or choose a picture from Picture Library.

Legend



Setting	Description
Title	Set whether to use a title for Pic Chart, and set the font size / font color of the title. The title can be selected from Label Library.
Channel	Set the channel label. When using Label Library, the number of the channels should be the same as the number of states in the library.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.42. Barcode

13.42.1. 2D Barcode Display

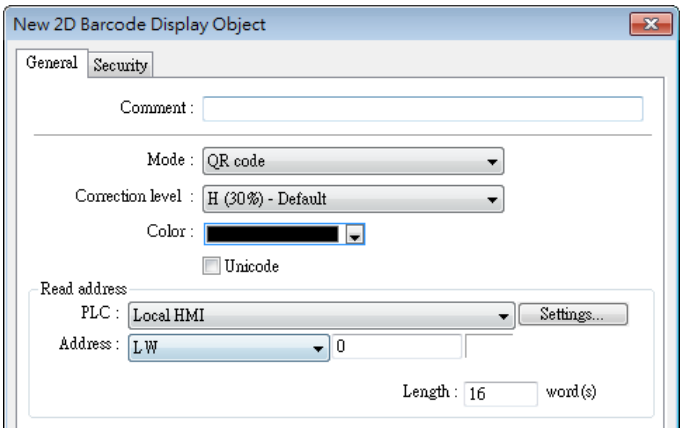
13.42.1.1. Overview

The 2D Barcode Display object transfers the information from the read address into QR Code or Aztec Code.

13.42.1.2. Configuration

Click the 2D Barcode Display icon on the toolbar to open the property dialog box. Set up the properties, press OK button, and a new 2D Barcode Display object will be created.

General Tab



Setting	Description
Mode	Supports QR code and Aztec code.
Correction level	2D barcodes have error correction capability to restore data if the barcode is dirty or damaged. **QR code** Four correction levels are available: L, M, Q, and H. The data restoration rate is listed below. (The data restoration rate for total codewords. Codeword is a unit that constructs the data area.) **Correction Level** L 7% M 15% Q 25% H 30% **Aztec code** Aztec code supports error correction levels from 5% to 95%.

	Specifying a higher correction value results in a larger printed symbol and increases accuracy.
Color	Set the 2D barcode color.
UNICODE	By default, the 2D barcode is generated via ASCII encoding. If this check box is selected, the 2D barcode is generated via UNICODE encoding. For characters that are not ASCII defined English alphabets or numbers, for example, Chinese or Korean characters, please select the Unicode check box.
Read address	The 2D Barcode Display object will display the 2D barcode generated from the information entered by the read address. The word length limit: 1 ~ 1024.

13.42.2. Push Notification Barcode Display

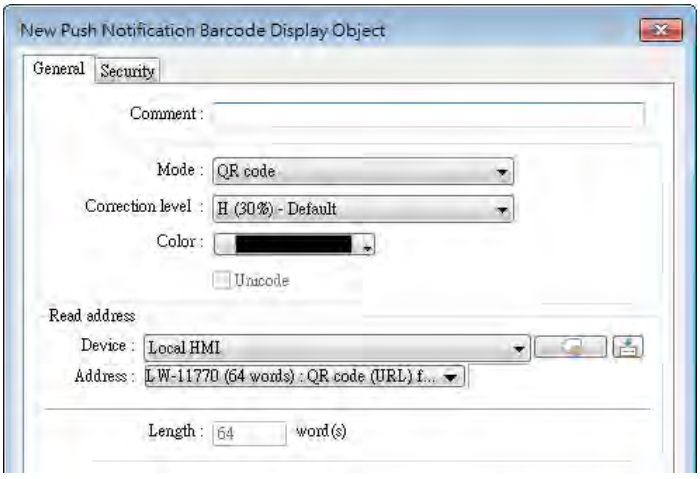
13.42.2.1. Overview

Push Notification Barcode Display object displays the QR code needed for setting up EasyAccess 2.0 push notification via messaging applications.

13.42.2.2. Configuration

Click the [Object] » [Barcodes] » [Push Notification Barcode Display] to open the property dialog box. Set up the properties, press OK button, and a new Push Notification Barcode Display object will be created.

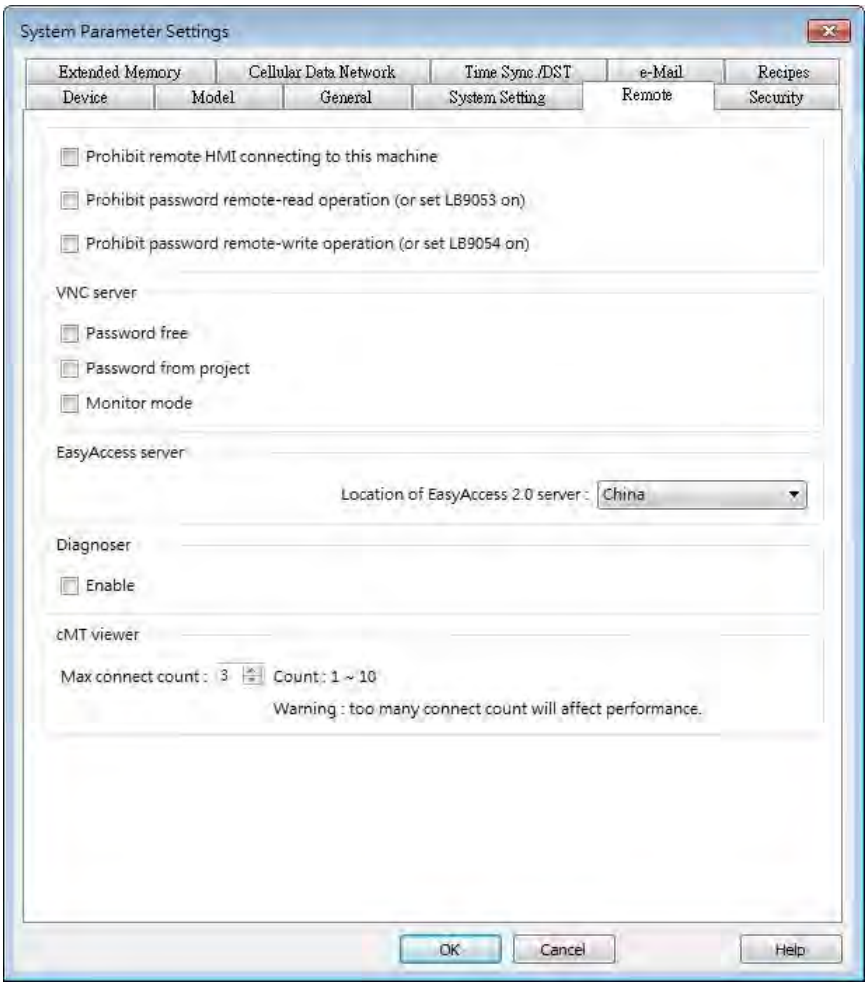
General Tab



Setting	Description
Mode	Supports QR code and Aztec code.

Correction level	2D barcodes have error correction capability to restore data if the barcode is dirty or damaged. **QR code** Four correction levels are available: L, M, Q, and H. The data restoration rate is listed below. (The data restoration rate for total codewords. Codeword is a unit that constructs the data area.) **Correction Level** L 7% M 15% Q 25% H 30% **Aztec code** Aztec code supports error correction levels from 5% to 95%. Specifying a higher correction value results in a larger printed symbol and increases accuracy.
Color	Set the 2D barcode color.
UNICODE	Cannot be changed by default.
Read address	Cannot be changed by default.

System Parameter Settings



Setting	Description
EasyAccess Server	Global: Supports push notification features of Wechat, Facebook, and Line. China: Supports push notification features of WeChat only.

Push Notification Settings

After downloading the project file to HMI, the QR code will be shown on HMI screen. For more information on setting up push notifications in WeChat, Facebook, and LINE, please refer to the link below (see version 2.8 and later).

https://support.ihmi.net/ea20/release-notes

13.42.3. Barcode Scanner (Android Camera)

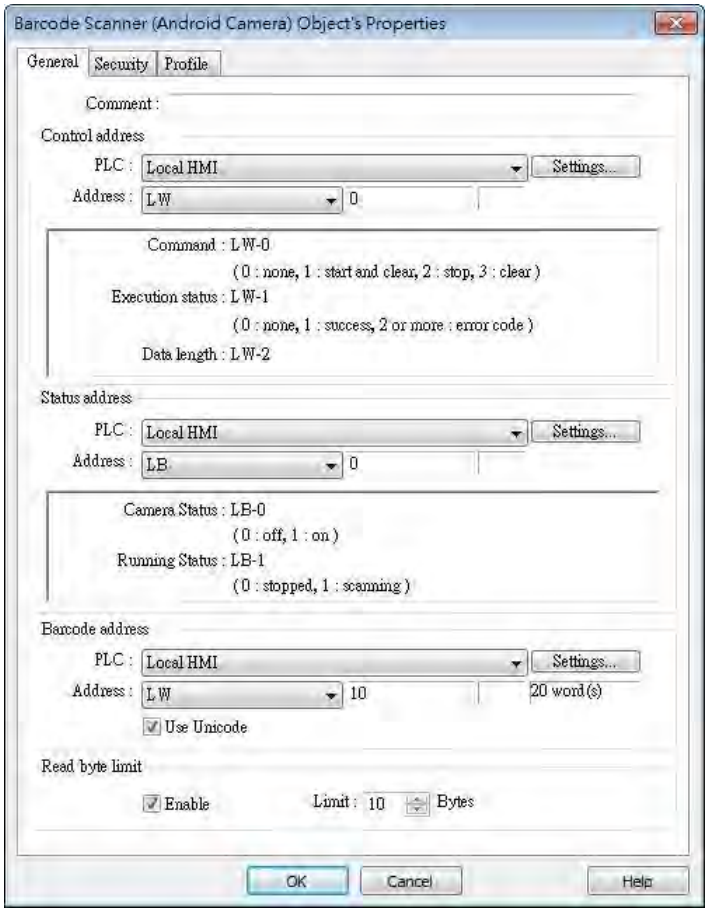
13.42.3.1. Overview

By connecting an Android device (smartphone/tablet) equipped with a camera to a cMT / cMT X HMI using cMT Viewer installed on the Android device, the camera can be used to scan 1D or 2D barcodes.

13.42.3.2. Configuration

Click the Barcode Scanner icon on the toolbar or click [Objects] » [Barcodes] and then select [Barcode Scanner]. Configure the parameters and click OK; a Barcode Scanner object will be created.

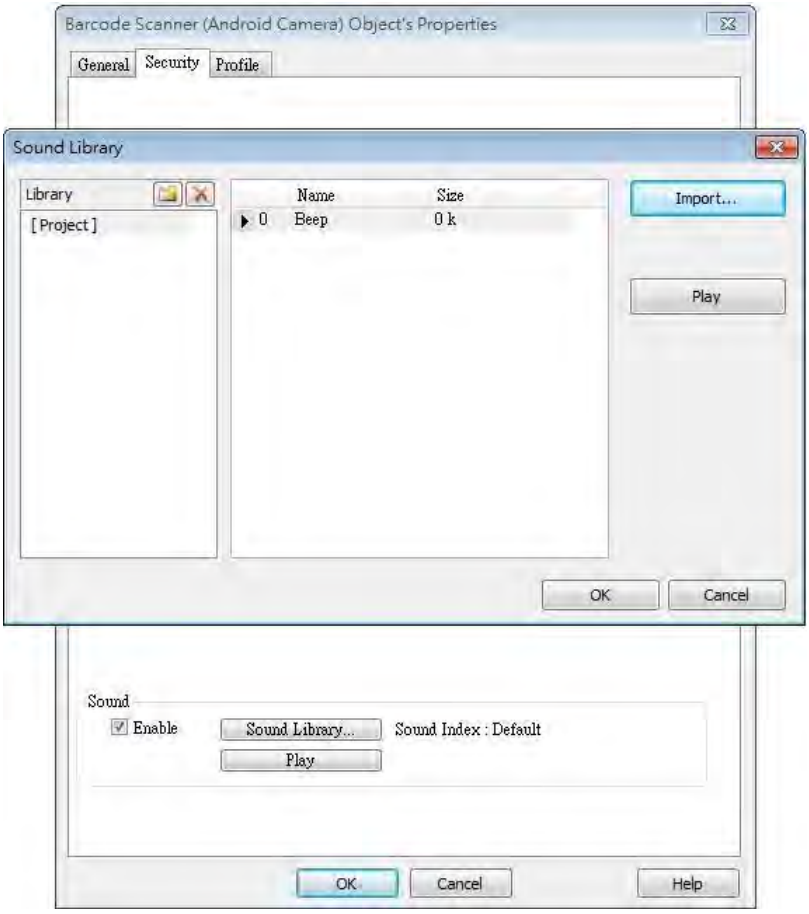
General Tab



Setting	Description
Control address	**Control address: Gives Command to Barcode**

	Scanner. 0: None 1: Start and Clear 2: Stop 3: Clear **Control address + 1: Shows Execution Status.** 0: None 1: Success 2 or more: Error Code **Control address + 2: Shows data length scanned.**
Status address	**Status address: Shows camera status is On / Off.** 0: Off 1: On **Status address + 1: Shows whether scanning is ready.** 0: Stopped 1: Ready for scanning
Barcode address	The address that stores the data read, UNICODE is allowed.
Read byte limit	If the data read exceeds this setting, the execution status turns to 2 (error code).

Security Tab



Setting	Description
Sound	If Enable is selected, when data is read, a sound is emitted. The supported sound file format is .wav.

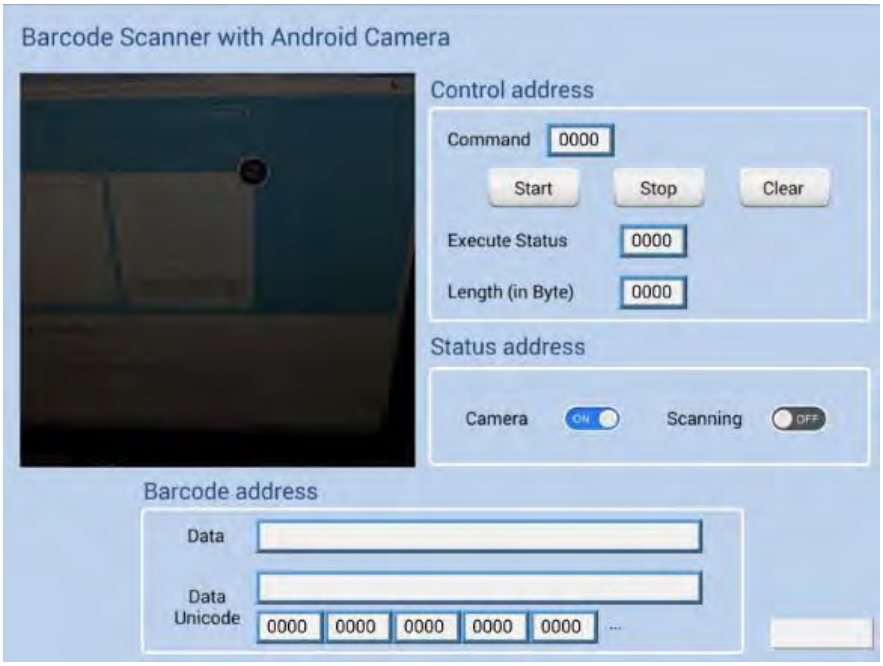
Note

- Barcode Scanner is currently supported on cMT-SVR and cMT3151 models. Barcode Scanner cannot be opened using simulation mode or cMT Viewer.
- Supports: EAN/UPC, Code 128, Code 39, Interleaved 2 of 5 and QR Code.
- On the device, if other applications are also using the camera, or the camera is locked, cMT Viewer may not operate properly.
- In the project, when multiple cMT Viewer devices are connected, since the same address is shared between the devices, the devices will simultaneously scan if they are displaying the same window with Barcode Scanner.

Example 1

The following demo project shows how to scan QR code using a tablet.

1. At the beginning, the display is dark.



2. Tap Start button, the status of Scanning turns ON, the display turns bright, and is ready for scanning.

3. When a QR code is read, the Execute Status turns 1, and the QR code is captured, its content will be displayed in Barcode Address group box. Unicode is also supported.

4. If the size of the data read exceeds the maximum allowable size set in Read Byte Limit (10 bytes in this project), the Execution Status turns to 2 (error code). The exceeding part will still be displayed in the ASCII objects in Barcode Address group box, since the data length displayed depends on the ASCII object settings (20 words in this project).

5. After changing to another page, Scanning turns OFF, the parameters are reserved. The parameters will be cleared when next time Start button is tapped, or Clear button is tapped.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

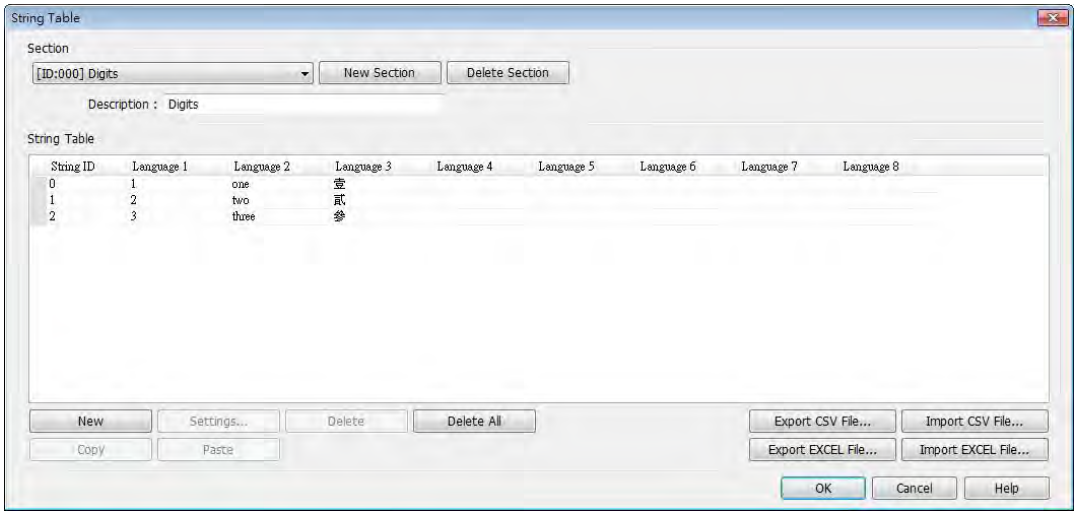
13.43. String Table

13.43.1. Overview

By pre-configuring the texts and their corresponding numbers in the String Table, the text can be changed dynamically on HMI. String Table can also be used in a multi-language environment.

13.43.2. Configuration

Click [Project] » [String] to open the property dialog box. Set up the properties, press OK button, and a new String Table object will be created.



Setting	Description
Section	A list of all the existing String Tables. [New Section] Add a new String Table. [Delete Section] Delete the selected String Table.
New	Add a new string in the table.
Settings	Set the content of the selected string.
Export CSV File…	Export all the existing String Tables as a *.csv file.
Import CSV File…	Import *.csv file into the String Table.
Export EXCEL File…	Export all the existing String Tables as a *.xls file.
Import EXCEL File…	Import *.xls file into the String Table.

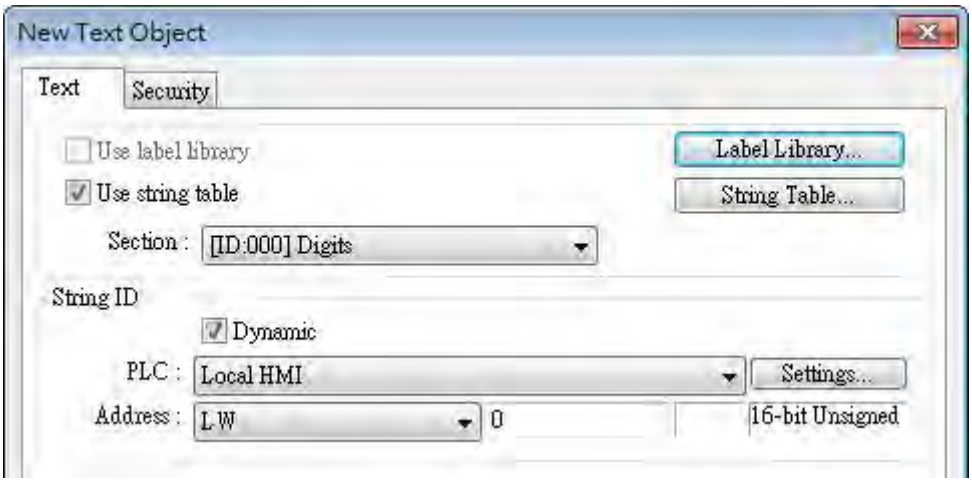
Note

- The font of each language in the String Table must be specified in Label Text Library.

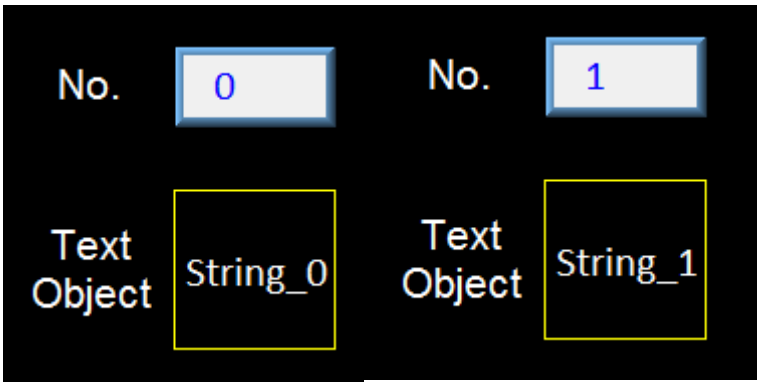
- Number of rows from all sections combined is limited to 10000.

Example 1

1. Create a String Table using the same settings as the preceding figure.
2. Create a Text object, select [Use string table] check box. In String ID group box, select [Dynamic] and set read address to LW-0.



3. Create a Numeric object, set address to LW-0.
4. When the value in LW-0 is 0, the No. 0 string is displayed; when the value in LW-0 is 1, the No. 1 string is displayed.



13.44. Database

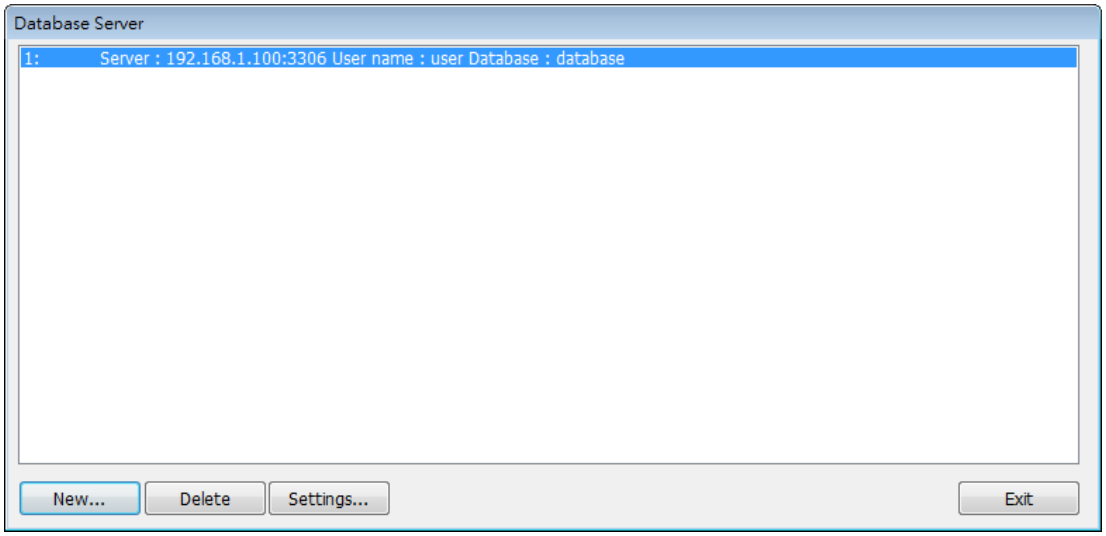
13.44.1. Database Server

13.44.1.1. Overview

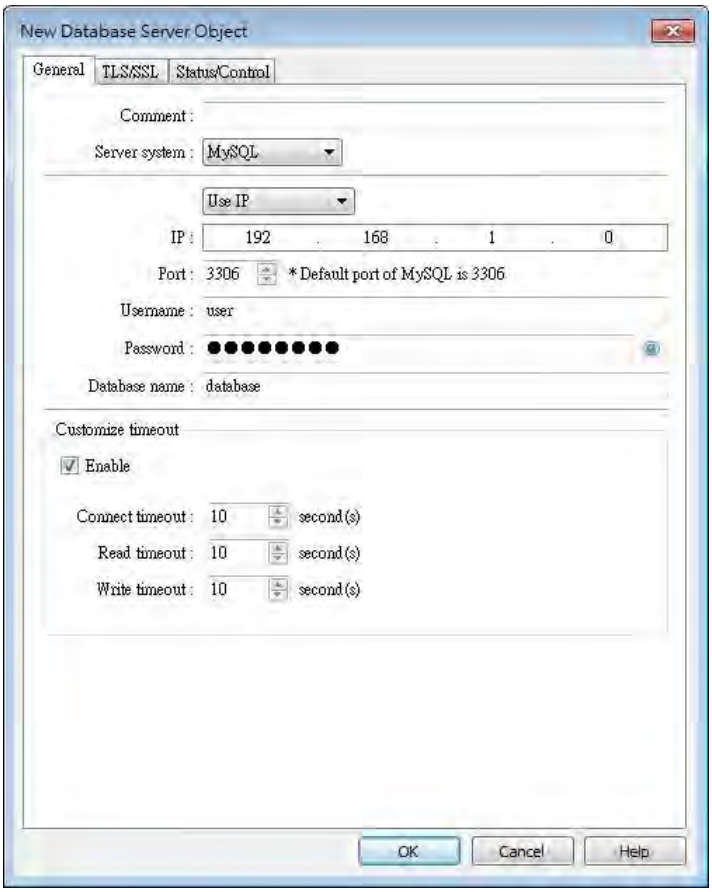
Database Server object enables connection with MySQL or MS SQL database server, allowing users to send data log or event log to the database, or use SQL Query to access data from the database.

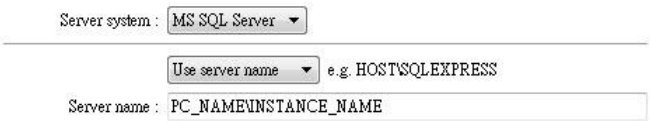
13.44.1.2. Configuration

Click the Database Server icon on the toolbar to create a Database Server object. Or, click [Data/History] » [Database Server] in the menu.



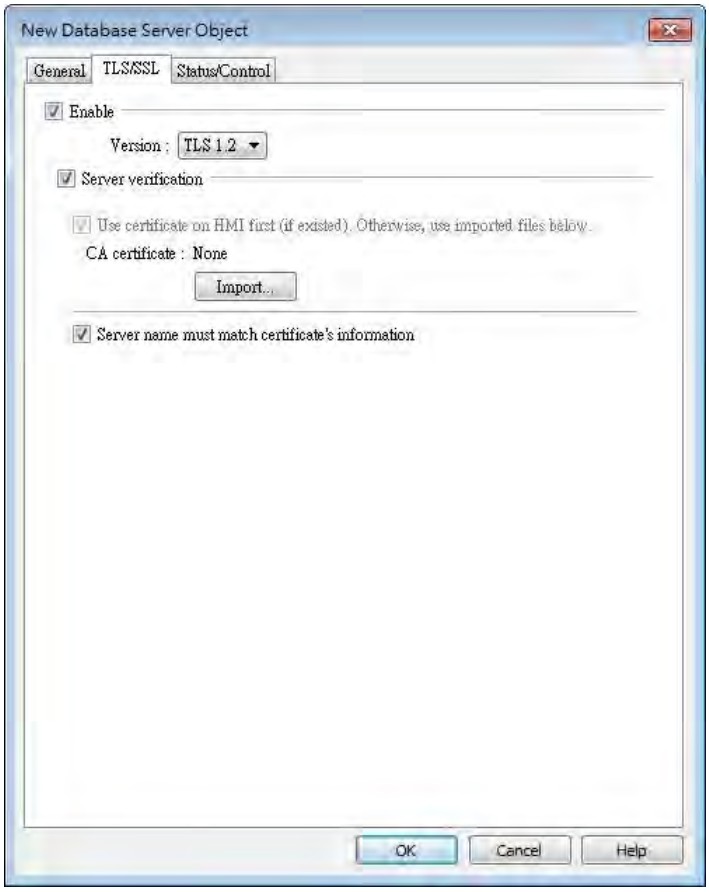
General Tab



Setting	Description
Server system	Supported server systems: MySQL, MS SQL Server
IP	Enter the IP address of the database.
Use domain name	 Supports designating a server using a domain name.
Use server name	 This option is only available when the selected server system is MS SQL Server. An instance stands for a communication port number, and the port number can be used to identify multiple database servers on the same computer. A server name can be in one of the following formats. ● <Computer name>\<Instance name>

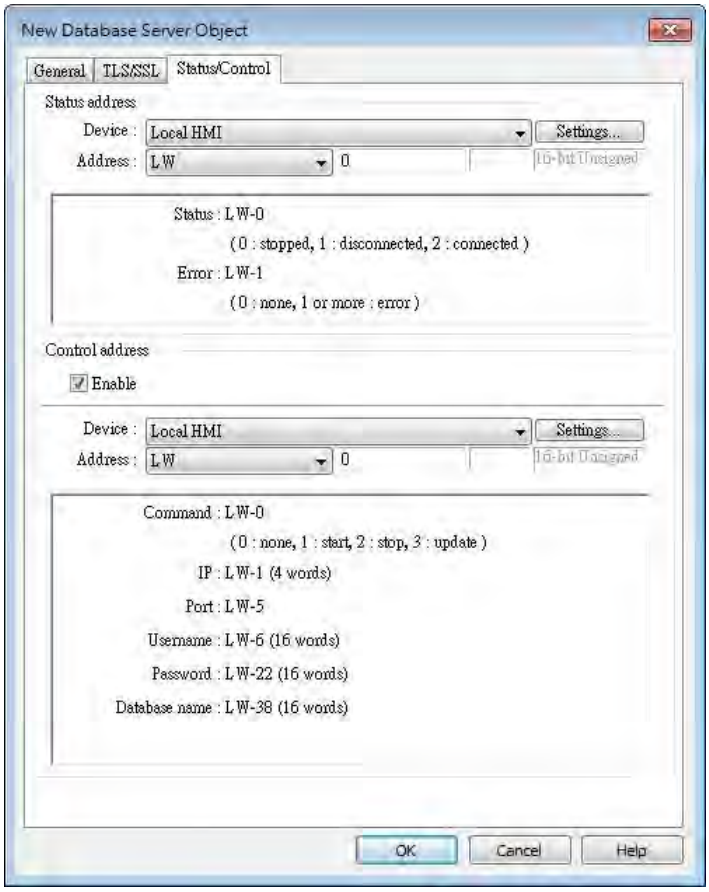
	• <Computer name> (Connect to Default Instance : MSSQLSERVER) • <IP address>\<Instance name> • <IP address> (Connect to Default Instance : MSSQLSERVER)
Port	Enter the port number of the database.
Username	Enter the username for connecting the database. The maximum is 32 words.
Password	Enter the password for connecting the database. The maximum is 32 words.
Database name	Enter the name of the database for collecting historical data.
Customize timeout	Timeouts for the following operations can be customized. **Connect timeout** The timeout for connection attempts. Connection attempts will be terminated when the timeout is reached. **Read timeout** The timeout for read operations. **Write timeout** The timeout for write operations.

TLS/SSL Tab



Setting	Description
Enable	Enable TLS/SSL security. TLS version can be selected from: TLS 1.0, TLS 1.1, and TLS 1.2.
Server verification	**Enable** Verify whether the server certificate is signed by CA (certificate authority) certificate. Server certificate is sent from server during connection. **Server name must match certificate's information** Verify whether the server's domain name or IP matches the records in the server certificate. Domain name and IP records are stored in Subject Alternative Name of the certificate.

Status/Control Tab



Setting	Description
Status address	LW-n: Displays the connection status of Database Server. Value / Description: 0 – Not attempting to connect to database. 1 – Failed to connect to database. 2 – Connection succeeded. LW-n+1: Error indicator. Value / Description: 0 – No error. 1 – Unknown error. 2 – Failed to connect to database. 3 – Database blocks the unauthorized connection. 4 – Incorrect database name. 5 – Invalid domain name.
Control address	LW-n: Controls the operation of Database Server. Value / Description: 0 – Ready 1 – Start 2 – Stop 3 – Update

	LW-n+1: Sets the IP address of the database. LW-n+5: Sets the port number of the database. LW-n+6: Sets the username for connecting database. LW-n+22: Sets the password for connecting database. LW-n+38: Sets the name of the database for collecting historical data.

Note

- If sampled data is successfully synchronized to the SQL database, three tables will be generated in the database, and the sampled data is saved in *_data table.

Table	Description
*<HMI NAME>_<DATALOG NAME>*_data	Saves data sampling
*<HMI NAME>_<DATALOG NAME>*_data_format	System folder
*<HMI NAME>_<DATALOG NAME>*_data_section	System folder

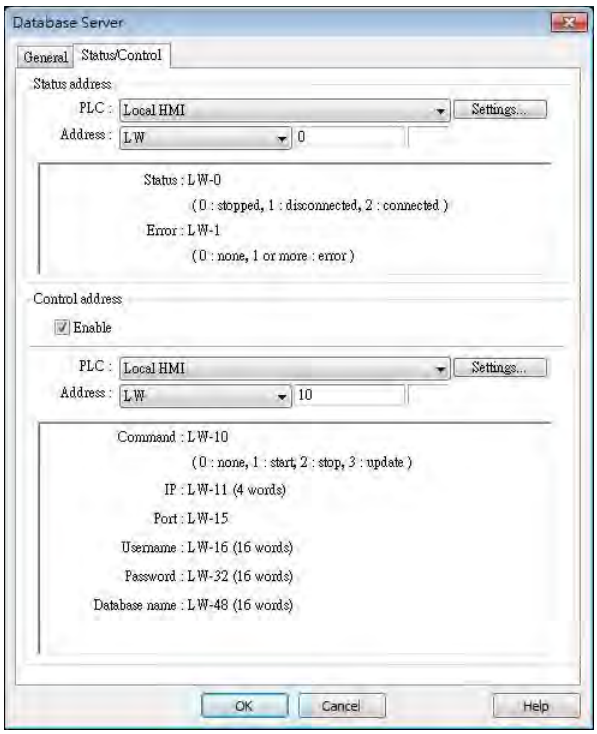
- When synchronizing event log, the three tables generated in the database are listed as the following table, and the event log is saved in *_event table.

Table	Description
*<HMI NAME>*_event	Saves event log
*<HMI NAME>*_event_log	System folder
*<HMI NAME>*_event_update_time	System folder

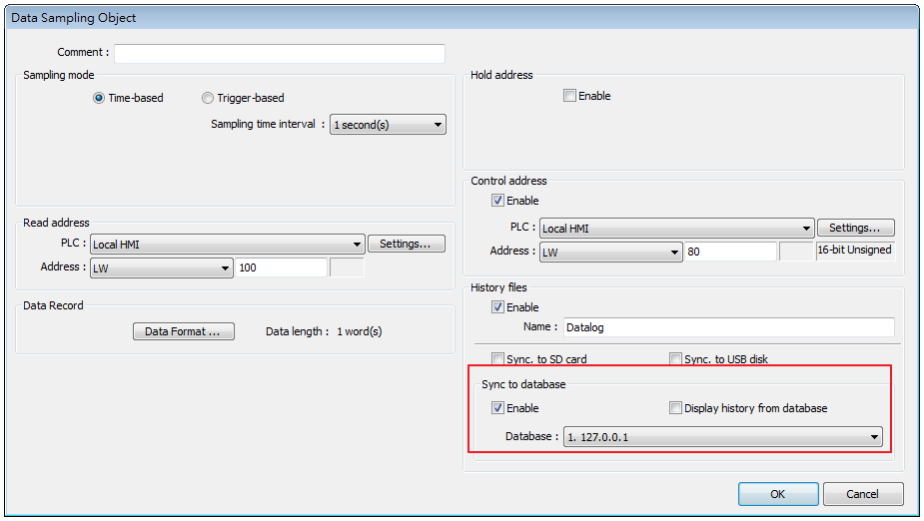
- If the content of data sampling / event log, such as data format or event message, is changed and downloaded to HMI, please delete the tables listed above first, and then the new content will thus be effective.
- Azure SQL Server is supported. To use Azure SQL Server, select MS SQL as server system and use domain name. Azure SQL Server has firewall settings, please add the client into whitelist, and see how to configure by opening the link below: https://docs.microsoft.com/en-us/azure/azure-sql/database/single-database-manage#manage-an-existing-server

Example 1

1. Create a Database Server object, set Status Address to LW-0, and Control Address to LW-10.



2. Create a Data Sampling object, in [Sync. to database] group box select [Enable], and set Control Address to LW-80, to update or clear HMI historical data.



3. If the database is successfully connected, the status indicator LW-0 displays 2 (connection succeeded), and the error indicator LW-10 displays 0 (no error).
4. Write 2 in LW-80 (sync. data). Open SQL database, the data can be found in table <*HMI NAME*>_<*DATALOG NAME*>_data.

	Table	Action	Records	Type	Collation	Size	Overhead
	hostname_datalog_data		6	MyISAM	utf8_unicode_ci	2.1 KiB	-
	hostname_datalog_data_format		1	MyISAM	utf8_unicode_ci	2.0 KiB	-
	hostname_datalog_data_section		0	MyISAM	utf8_unicode_ci	1.0 KiB	-
	3 table(s)	Sum	7	MyISAM	utf8_unicode_ci	5.2 KiB	0 B

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.44.2. SQL Query

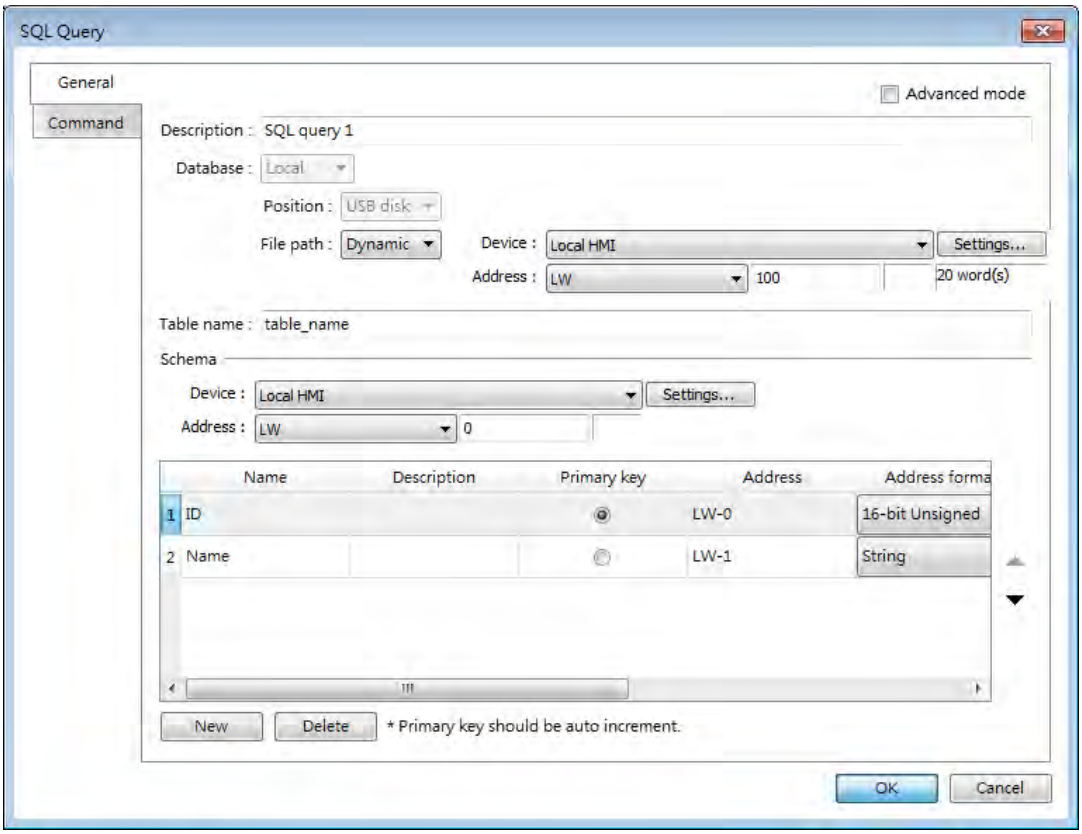
13.44.2.1. Overview

SQL Query can exchange data with SQL database. Before enabling SQL Query, please configure Database Server.

13.44.2.2. Configuration

Click [Data/History] » [SQL Query] to open the settings dialog box. Configure the parameters and click OK; a SQL Query will be created.

General Tab

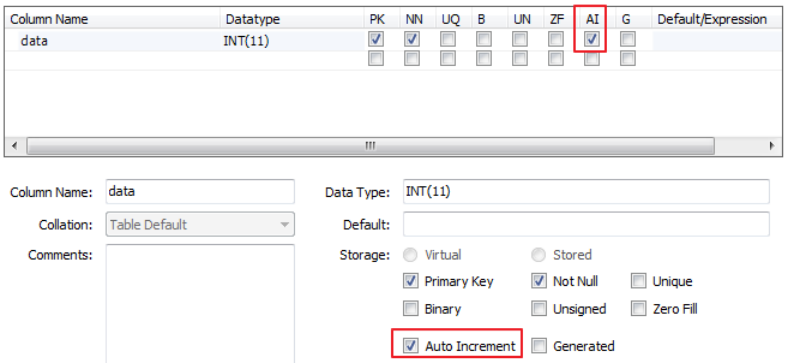


Setting	Description
Advanced Mode	**Without [Advanced Mode] selected:** Click [New] to add a new row or click [Import from server] to import an existing database. **With [Advanced Mode] selected:** Manually enter syntax in Command tab to control MySQL/MS SQL database. Please note that returning from Advanced Mode to General Mode is not

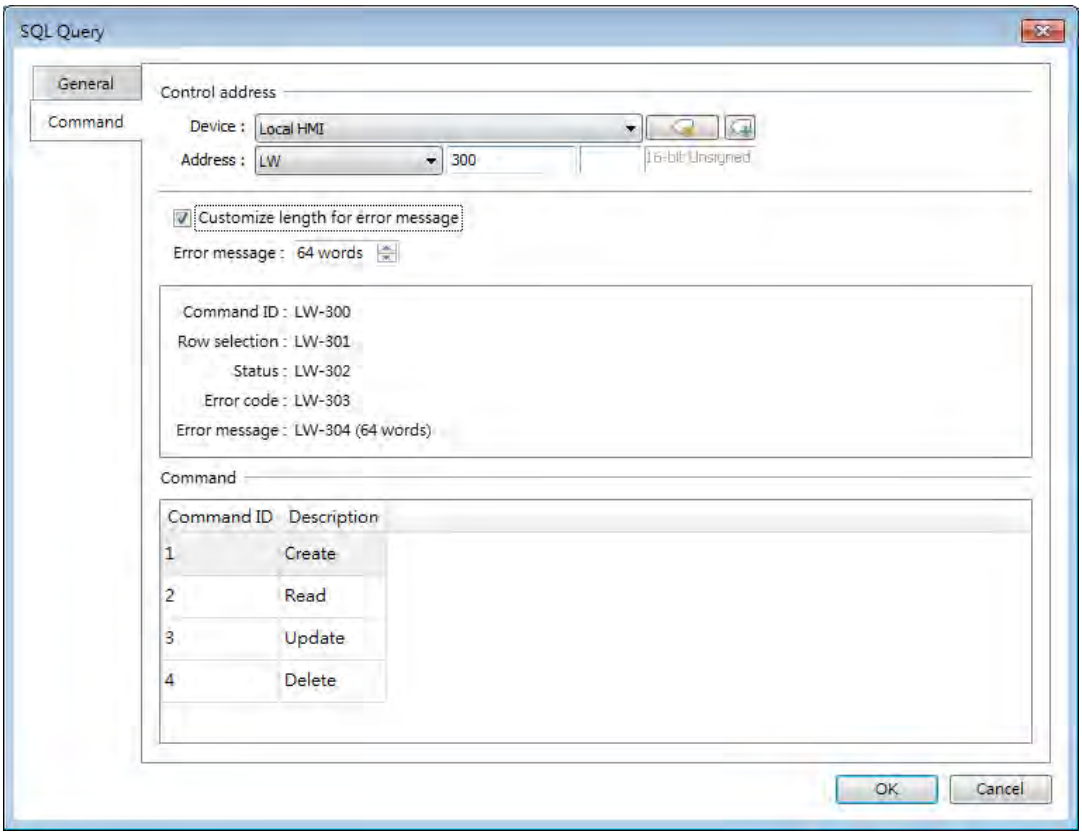
	possible.
Description	User's description about this query.
Database	Select the source database to read from. Local: The source database is the SQLite database stored in a USB disk or SD card attached to HMI. Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film. Remote: The source database is the designated Database Server.
File path	Use [Static] or [Dynamic] directory for a local database. When [Static] is selected, please enter the directory in this dialog box. When [Dynamic] is selected, please designate an address as the data source for the file path.
Table name	Enter the name of this query table.
Schema	The data read from database will be filled into the corresponding address specified in the schema. Please manually set Address Format after reading data.

Note

- A Primary Key should contain only numeric values.
- In MySQL, Auto Increment must be enabled for Primary Key.



Command Tab

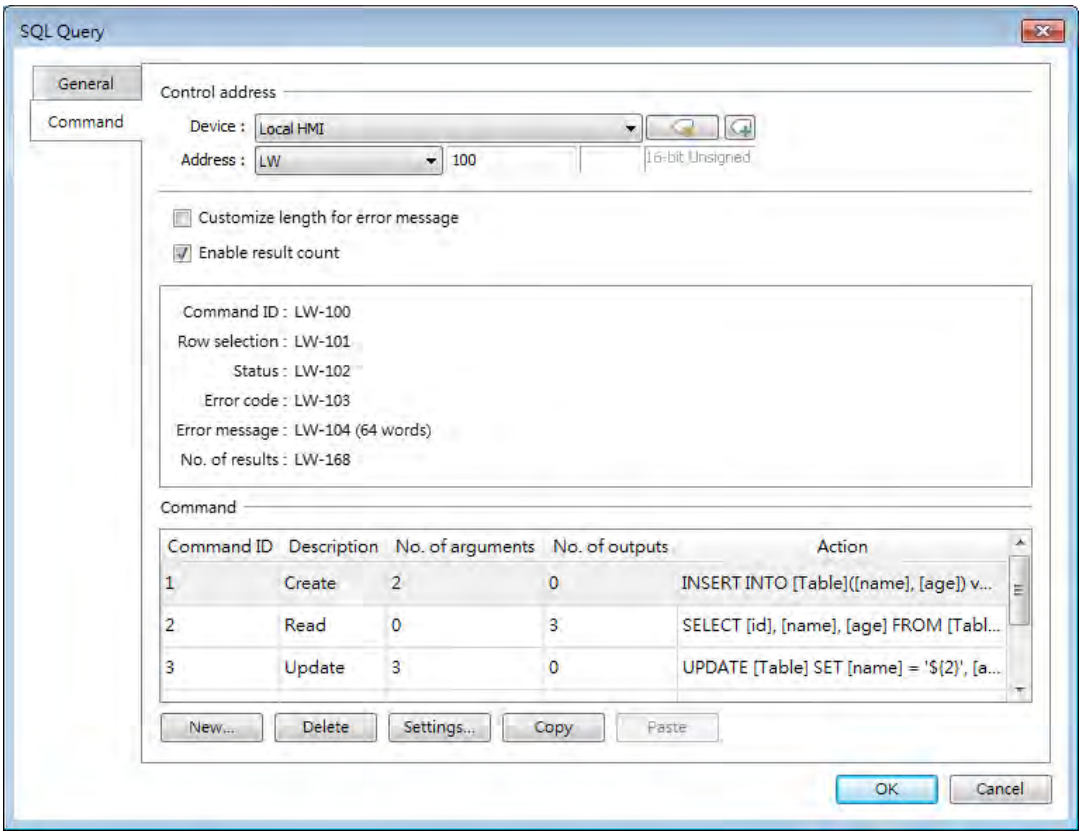


Setting	Description
Control address	Designate five consecutive registers to execute SQL commands and show results. Of the results, the length of error message can be customized.
Command	In standard mode (without checking Advanced mode box), the four basic commands of SQL (Create, Read, Update, and Delete) and their command ID are shown.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

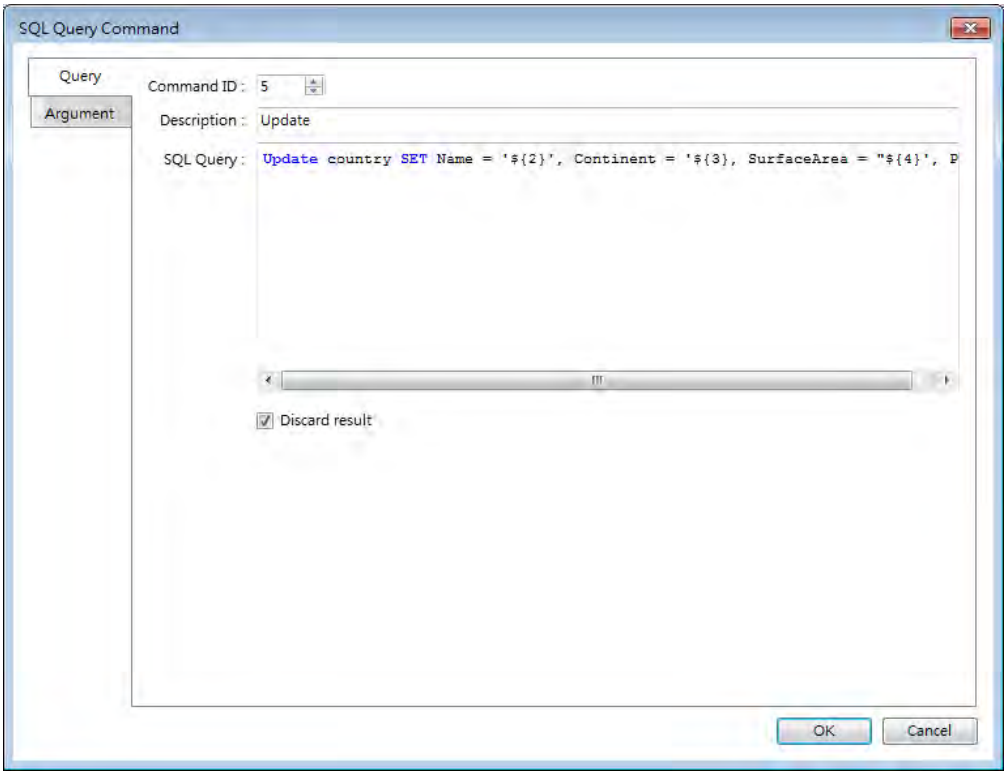
13.44.2.3. Advanced Mode

In the Command table in Advanced Mode, [No. of arguments], [No. of outputs], and [Action] columns can be found. Outputting multiple results to consecutive addresses at a time is possible.



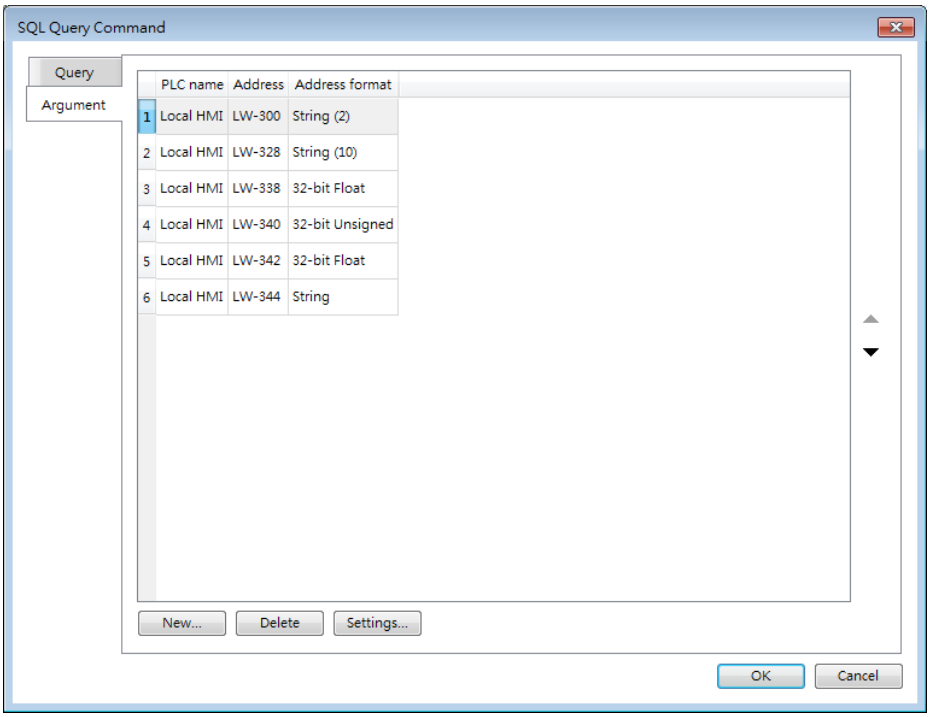
Setting	Description
Control address	Specify the ID number used to give this command.
Row selection	Specify a row.
Status	Displays the execution status. See Ch13.44.2.4.
Error code	Displays the execution result. See Ch13.44.2.5.
Error message	Displays the error message sent from database server. The default length of the error message is 64 words. Customizing the length of the error message is also possible, and the upper limit is 200 words.
No. of results	Displays the number of output results.
New / Delete / Settings / Copy	Add, delete, set, or copy a SQL Query command.

Query Tab



Setting	Description
Command ID	Specify the ID number used to give this command.
Description	Enter the description of this command.
SQL Query	Enter the syntax for this command. An argument should be enclosed in braces: ${*argument no.*}
Discard result	With this checkbox selected, the result of executing this command will not be shown in SQL Query Result Viewer object. This checkbox can be selected for commands that are done directly to the database without the need for returning a result, such as INSERT INTO, UPDATE, DELETE…etc.

Argument Tab

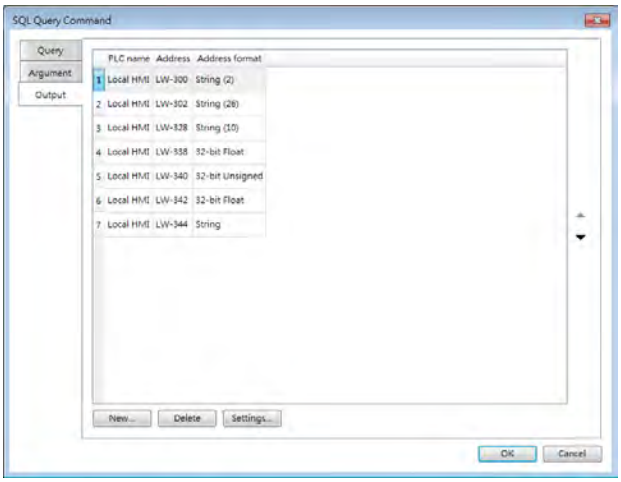


If argument is used in the syntax of a command in [Query] tab, the system will refer to the address specified in this tab according to the argument number enclosed in ${*argument no.*}.

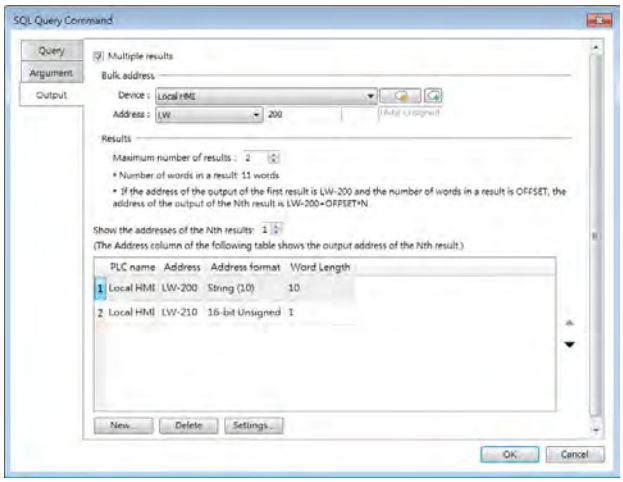
Output Tab

After reading database under Advanced Mode, the result will be stored in the addresses specified in this tab. Selecting [Multiple results] option can output multiple query results to consecutive addresses at a time.

[Multiple results] not selected



[Multiple results] selected



Setting	Description
Bulk address	Designate the start address for bulk output of results. The upper limit of data records is 10000

	(maximum number of results x number of columns).
Maximum number of results	The upper limit is 1000 results. Suppose maximum number of results is set to 100, and the returned number of query results is 300, then: When [Row selection] is set to 20, the 20th~119th results are output to bulk addresses. When [Row selection] is set to 260, the 260th~299th results are output to bulk addresses.
Show the addresses of the Nth results	Shows the output address of the Nth result in the Address column of the table below this setting. If the address of the output of the first result is LW-200 and the number of words in a result is OFFSET, the address of the output of the Nth result is LW-200+OFFSET*N.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.44.2.4. Status

Value	Meaning
0	Normal
1	Query result exceeds 1000 records. Using LIMIT clause can constrain the number of rows in one page.

13.44.2.5. Error Code

Error Code	Meaning
0	No mistakes
1	Unknown error
2	Invalid command
3	Database Server is not connected yet
4	Argument cannot be read
5	Cannot write and output
6	Incorrect number of arguments
7	Error in MySQL, please read error message
8	Unsupported datatype
9	Number of columns exceeds the limit
10	Number of rows exceeds the limit
11	Unable to read local database directory
12	Name of local database does not exist

13	Internal error

13.44.2.6. Converting Datatype

Converting datatype as shown in the following table will take place after reading MySQL database. If conversion cannot run properly, error code 5 will show. For example, when converting MySQL's INT into EasyBuilder Pro's 16-bit Unsigned, if the value exceeds the limit of 16-bit Unsigned, error code 5 will show.

MySQL data format	EasyBuilder Pro datatype
TINYINT SMALLINT MEDIUMINT INT BIGINT BIT	16/32-bit BCD 16/32-bit HEX 16/32-bit Binary 16/32-bit Signed 16/32-bit Unsigned
FLOAT DOUBLE DECIMAL	32-bit Float
DATETIME CHAR, BINARY VARCHAR, VARBINARY TINYBLOB, TINYTEXT BLOB, TEXT MEDIUMBLOB, MEDIUMTEXT LONGBLOB, LONGTEXT	String

13.44.3. SQL Query Result Viewer

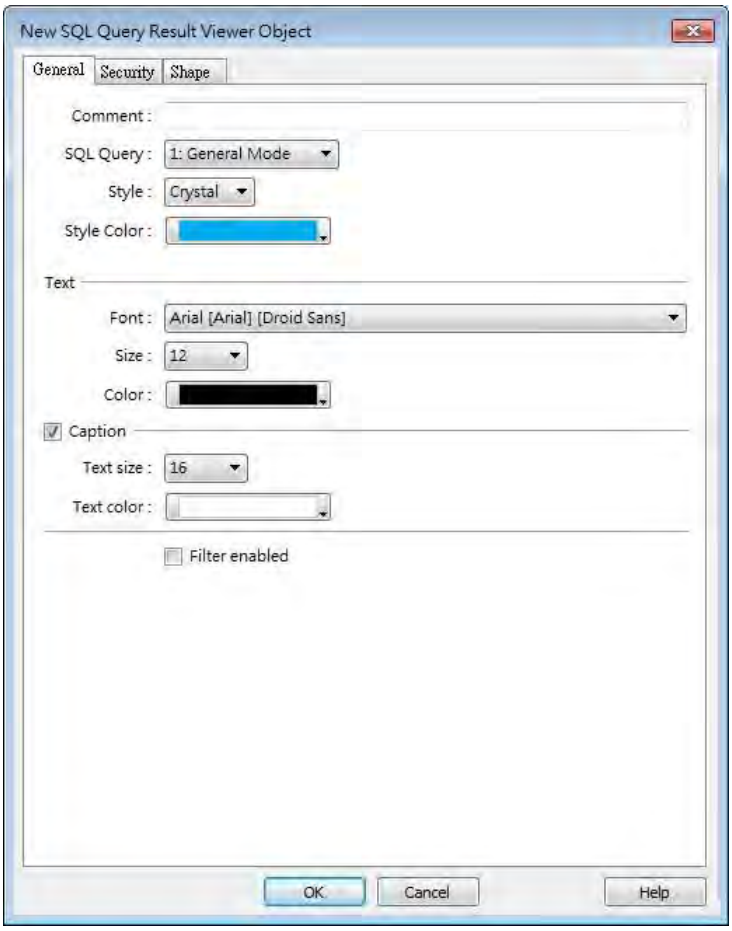
13.44.3.1. Overview

SQL Query Result Viewer shows the results obtained by running SQL Query.

13.44.3.2. Configuration

Click on [Data/History] » [SQL Query Result Viewer] to open the settings dialog box. Configure the parameters and click OK; a SQL Query Result Viewer will be created.

General Tab



Setting	Description
Comment	User's comment about this result viewer.
SQL Query	Select an existing SQL Query to show its result.
Style/ Style Color	Select a style and a color for this result viewer.
Text	Set the font, font size, and font color for the texts shown in this result viewer.

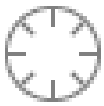
Caption	Set the font size and font color for the caption of this result viewer.
Table	This group box opens when selecting Default as style. The attributes of the result query table can be configured.
Filter enabled	With this checkbox selected, entering keywords in SQL Query Result Viewer to search for specific text is possible.

13.45. Dynamic Scale

13.45.1. Overview

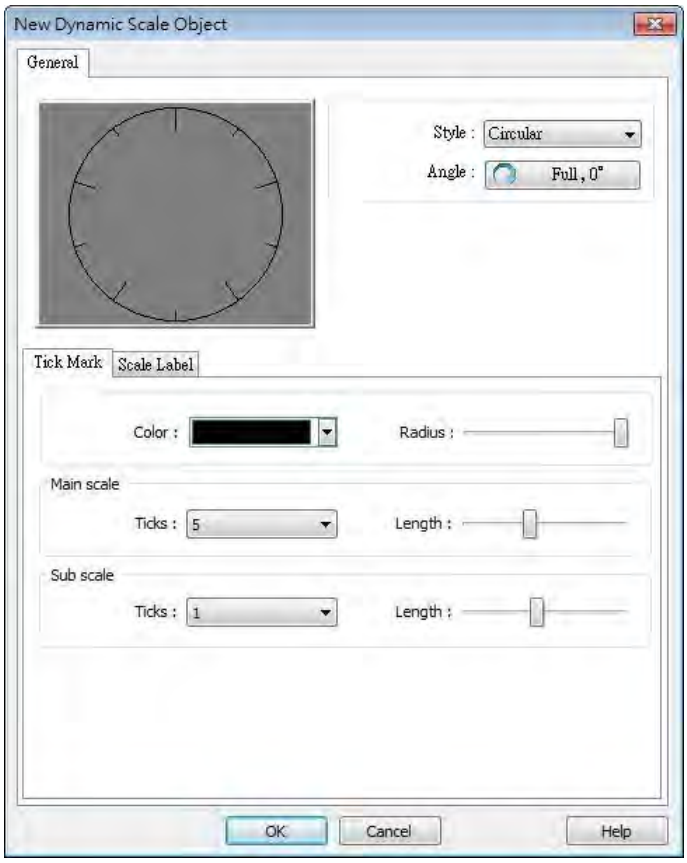
Dynamic Scale offers customizable tick marks and scale labels and can be used together with objects such as Trend Display, Bar Graph…, etc.

13.45.2. Configuration

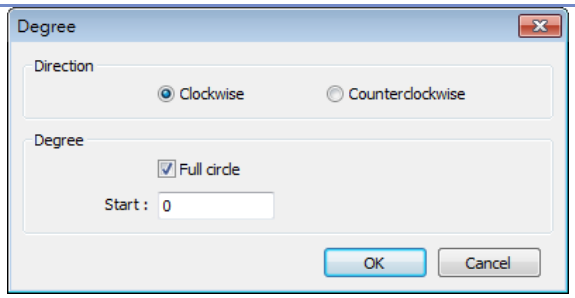


Click the Dynamic Scale icon on the toolbar to open a Dynamic Scale object property dialog box. Set up the properties, press OK button, and a new Dynamic Scale object will be created.

General Tab



Setting	Description
Style	Select the style from [Circular], [Horizontal], or [Vertical]. If select [Circular], set the [Direction] and [Degree].



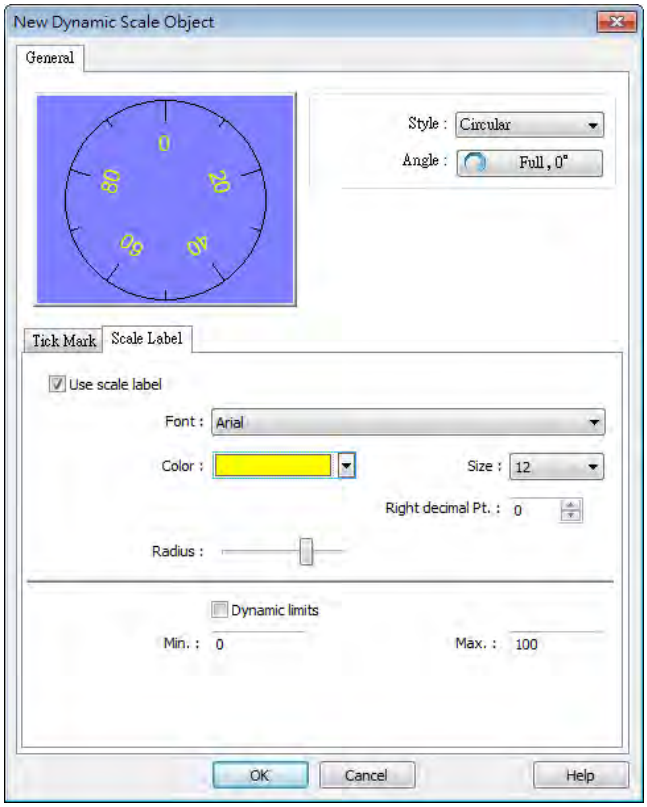
Direction

Select from [Clockwise] or [Counterclockwise].

Degree

If [Full circle] is selected, set the start degree.

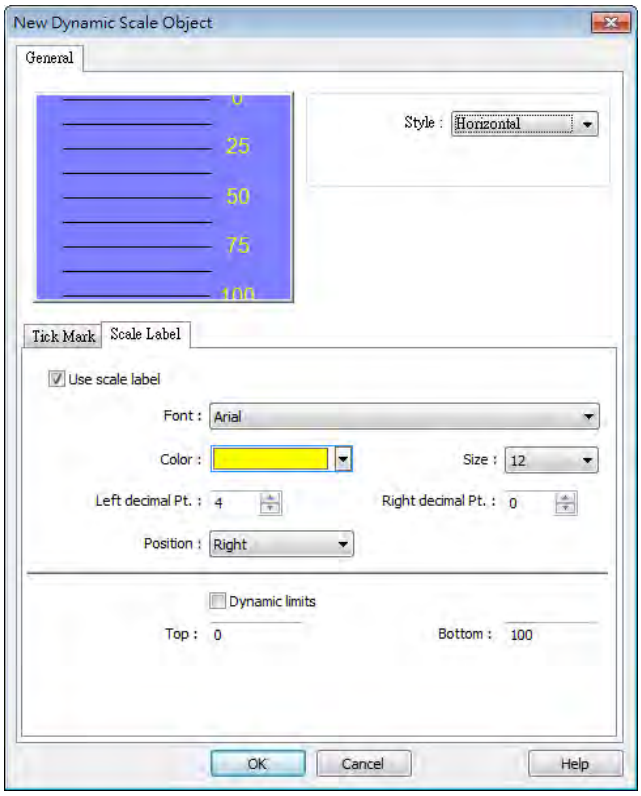
If [Full circle] is not selected, set the start and end degree.

Tick Mark	Select the color of the tick mark, and set the number of ticks for the main and sub scale (major and minor tick mark). For [Circular] style, the length of the major and minor tick mark, and the radius of the circular tick mark can be set.
Scale Label	Displays major tick labels. **Circular**  Set the font, font color, font size, and decimal point of the scale

label.

Set the radius start from the center of the object to the position to place the scale label.

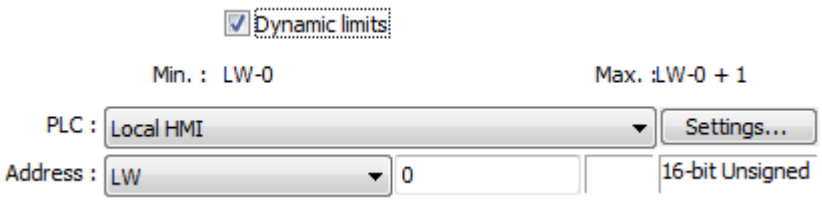
Virtical / Horizontal



Set the font, font color, font size, and decimal point of the scale label.

Set the position to display the scale label.

The Max. and Min. limits of the scale label can be specified. If [Dynamic limits] is selected, the Max. and Min. limits can be set by the designated word addresses.



Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.46. Dynamic Drawing

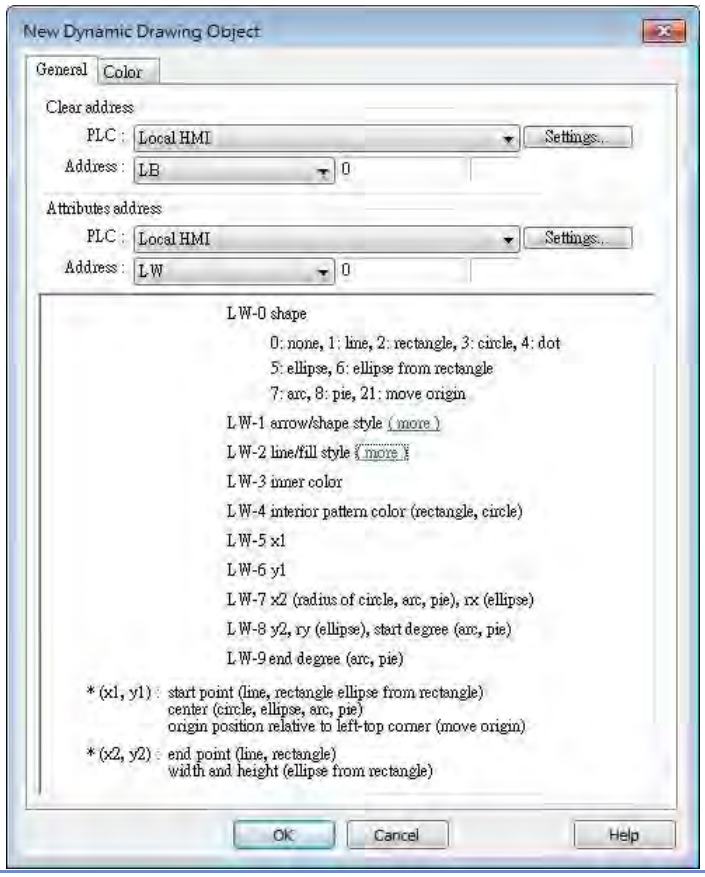
13.46.1. Overview

Dynamic Drawing object enables drawing a shape in a specified region on HMI screen at run time. The shape can be a line, a rectangle, a circle, or a dot. By setting the Attributes Addresses, the style and the color of the shape can be customized.

13.46.2. Configuration

Click the Dynamic Drawing icon on the toolbar to open a Dynamic Drawing object property dialog box. Set up the properties, press OK button, and a new Dynamic Drawing object will be created.

General Tab



Setting	Description
Clear Address	Clears the drawing.

Attributes Address Changes the attributes in the drawing.

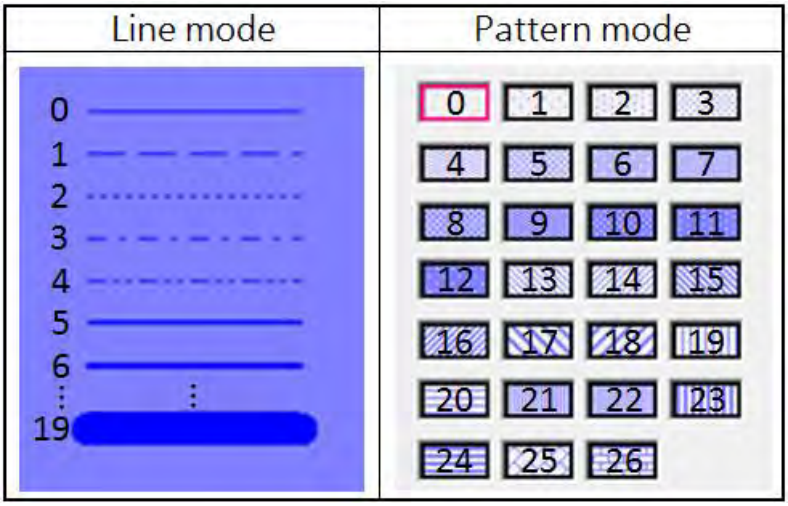
Entering different values in different Attributes Addresses brings different effects, as shown in the following table.

Attributes Address	Attributes Address+0	Attributes Address+1		Attributes Address+2	Attributes Address+3	Attributes Address+4
Default	0	Ones	Tens		Customizable	Customizable
Line	1	0: Non-arrow	0: Small	0: Solid line	Line color	
		1: Single-ended arrow (Hollow)	1: Large	1: Dashed line		
		2: Double-ended arrow (Hollow)		2: Dotted line		
		3: Single-ended arrow (Solid)		3: Dash Dot line		
		4: Double-ended arrow (Solid)		4: Dash Dot Dot line		
				5 and up: Solid line with thicknesses greater than 2		
Rectangle	2	0: Hollow		Drawn in Line mode	Rectangle color	Interior pattern color
		1: Solid		Drawn in Pattern mode		
Circle	3	0: Hollow		Drawn in Line mode	Circle color	Interior pattern color
		1: Solid		Drawn in Pattern mode		
Dot	4				Dot color	
Ellipse	5	0: Hollow		Drawn in Line mode	Ellipse color	Interior pattern color
		1: Solid		Drawn in Pattern mode		
Ellipse from Rectangle	6	0: Hollow		Drawn in Line mode	Ellipse color	Interior pattern color
		1: Solid		Drawn in Pattern mode		
Arc	7			Drawn in Line mode	Arc color	
Pie	8	0: Hollow		Drawn in Line mode	Pie color	Interior pattern color
		1: Solid		Drawn in Pattern mode		
Move Origin	21					

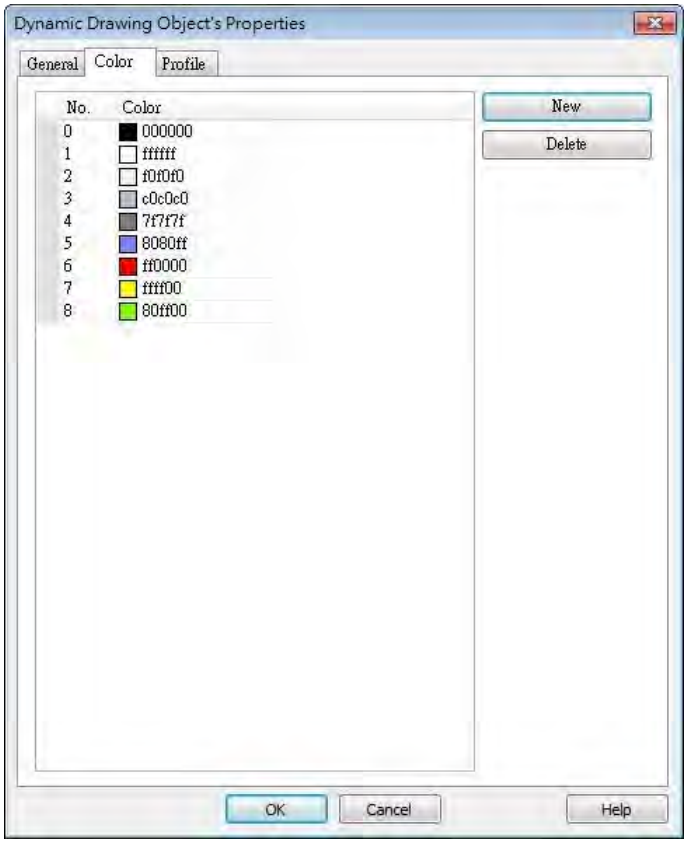
Attributes Address	Attributes Address+0	Attributes Address+5	Attributes Address+6	Attributes Address+7	Attributes Address+8	Attributes Address+9
Default	0					
Line	1	Start point X	Start point Y	End point X	End point Y	
Rectangle	2	Left-top point X	Left-top point Y	Right-bottom point X	Right-bottom point Y	
Circle	3	Center point X	Center point Y	Radius		
Dot	4	Dot X	Dot Y			
Ellipse	5	Center point X	Center point Y	Radius on the X	Radius on the Y	

				axis	axis	
Ellipse from Rectangle	6	Left-top point X	Left-top point Y	Width	Height	
Arc	7	Center point X	Center point Y	Radius	Start degree	End degree
Pie	8	Center point X	Center point Y	Radius	Start degree	End degree
Move Origin	21	New origin X	New origin Y			

The values in [Attributes Address+2] represent different Line or Pattern styles, as shown in the following table.



Color Tab

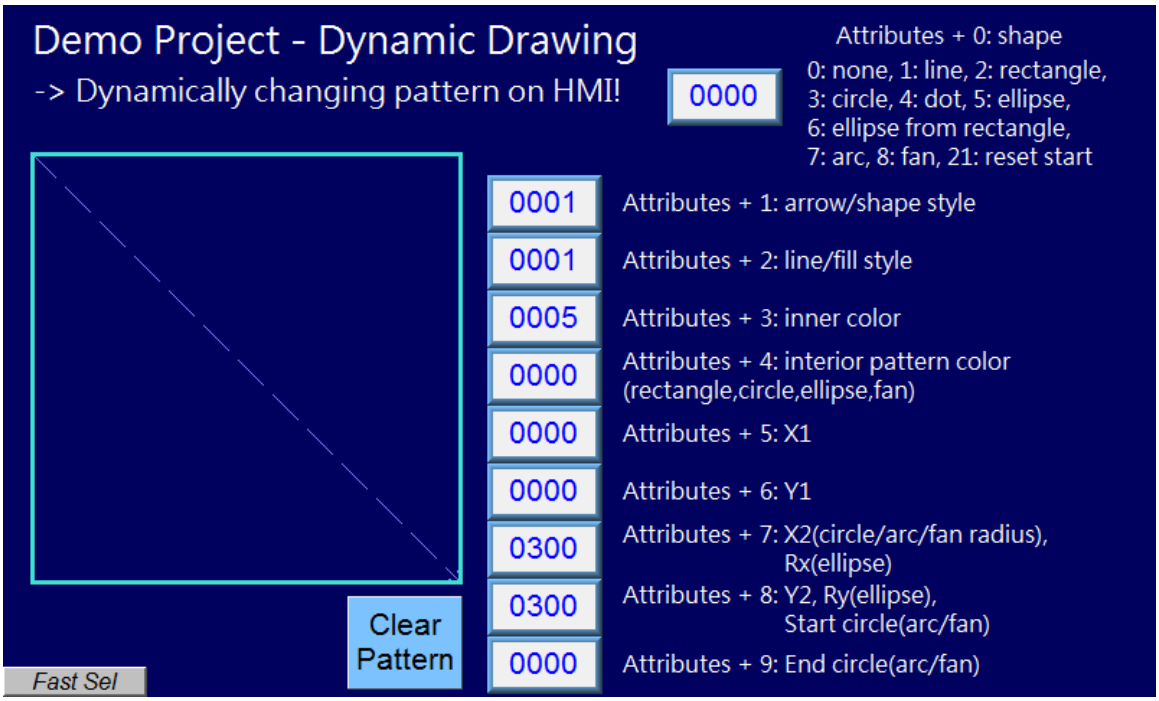


Setting	Description
New	Adds a color to be used in the drawing.
Delete	Deletes the selected color.

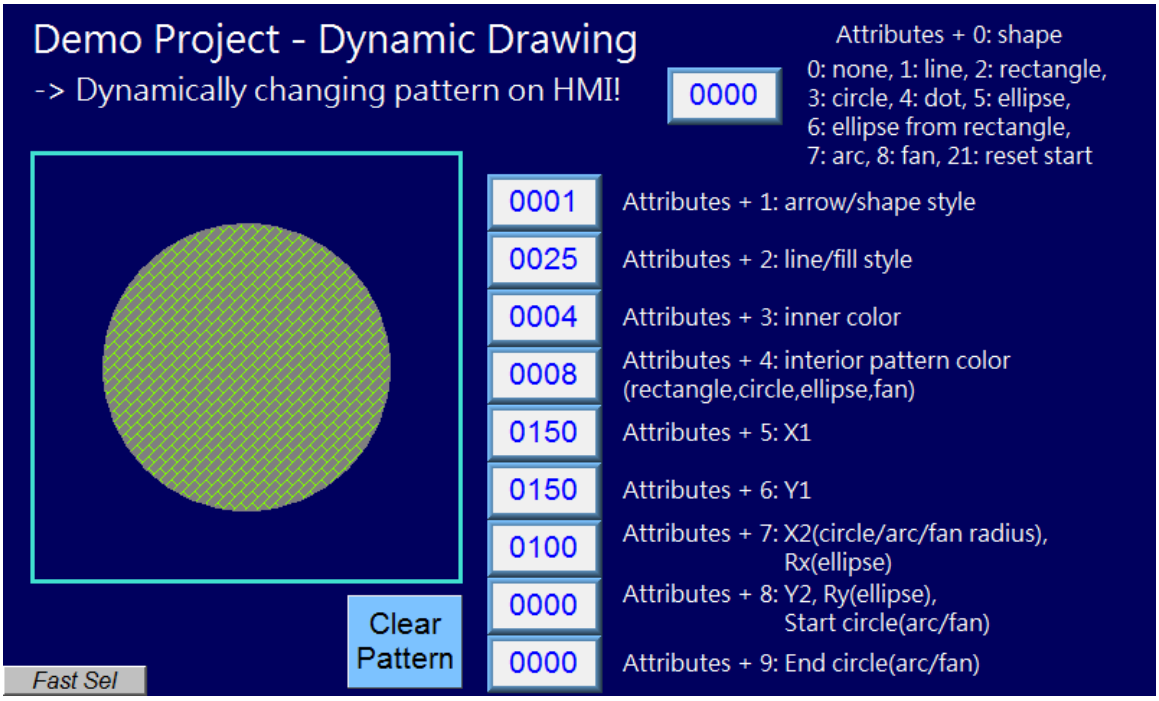
Example 1

The following demo project demonstrates how to dynamically draw an arrow or a circle on HMI screen. To build the project, follow these steps:

1. Create a Dynamic Drawing object, set Clear Address to LB-0, and Attributes Address to LW-0.
2. Create a Toggle Switch object, set address to LB-0, and select Toggle as switch style, for clearing the drawing.
3. Create 10 Numeric objects, set addresses to LW-0~LW-9, for specifying the attributes in the drawing.
4. Run simulation or download the project to HMI to see the result. By entering 1 in LW-0, a line is drawn, and entering different values in LW-1~9 can change the style, the color, and the position of the line.



5. Press Clear Pattern button, and enter 3 in LW-0, a circle is drawn, and entering different values in LW-1~9 can change the style, the color, and the position of the circle.



Note

- Before using Attributes Address, please define [Attributes Address + 1] ~ [Attributes Address + 9]. The system will reset the Attributes Address, after it is used.
- If the drawing is not cleared, the new drawing will overlap the previous one, and the maximum acceptable number of drawings in a Dynamic Drawing object is 1000.
- The maximum number of line styles is 19, which means the maximum thickness of a solid line is 16. The style numbers that exceed 19 will be displayed as 19.
- Find the color number in Color tab.
- The range of the start and end degree for Arc and Pie is 0 to 360 degrees.
- Origin position is relative to left-top corner (0,0). Giving “move origin” command will make (x1, y1) the new origin, and x1, y1 will keep on accumulating until being reset to (0,0) by clearing the drawing.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.47. PDF Reader

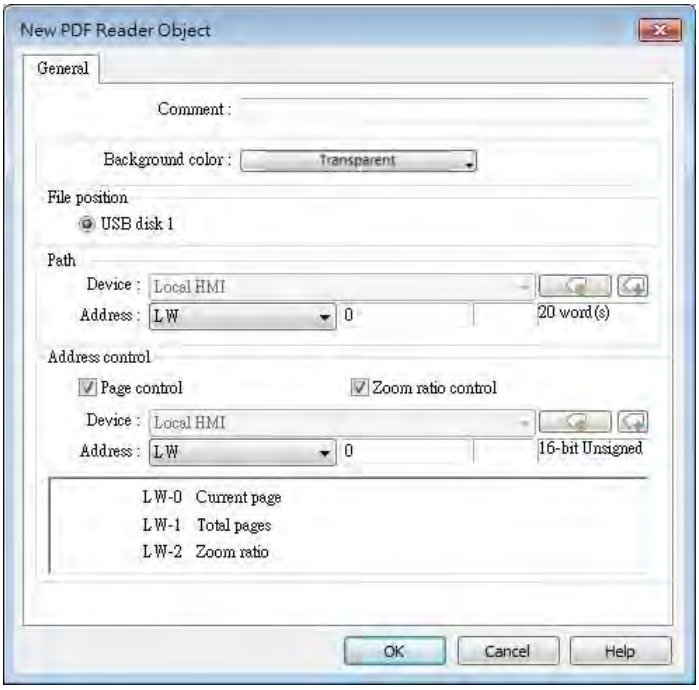
13.47.1. Overview

PDF Reader object enables viewing of PDF documents on HMI.

13.47.2. Configuration

Click the PDF Reader icon on the toolbar to create a PDF Reader object. Or, click [Object] » [Media] » [PDF Reader] in the menu.

General Tab



Setting	Description
File position	Select the position where the PDF file is stored.
Path	The directory of the PDF file stored in the external device.
Page Control	**Current page** The page number of the current page. Set a value to change the current page. **Total pages** Total number of pages of the PDF file.

Zoom ratio	Current zoom ratio. Set a value to change the zoom ratio. Unit: %; i.e., the number 100 indicates 100% display (1:1).

Note

- PDF Reader cannot be opened using simulation mode or cMT Viewer.
- The PDF files protected by passwords or restrictions cannot be read using PDF Reader.
- CPU loading may rise when multiple PDF Reader objects are opened simultaneously.
- When entering a page number in the Page Control register under multi-page view mode, the specified page will be opened in single-page view.
- Two-finger zoom in / out gestures are supported on capacitive touchscreens.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

13.48. Table

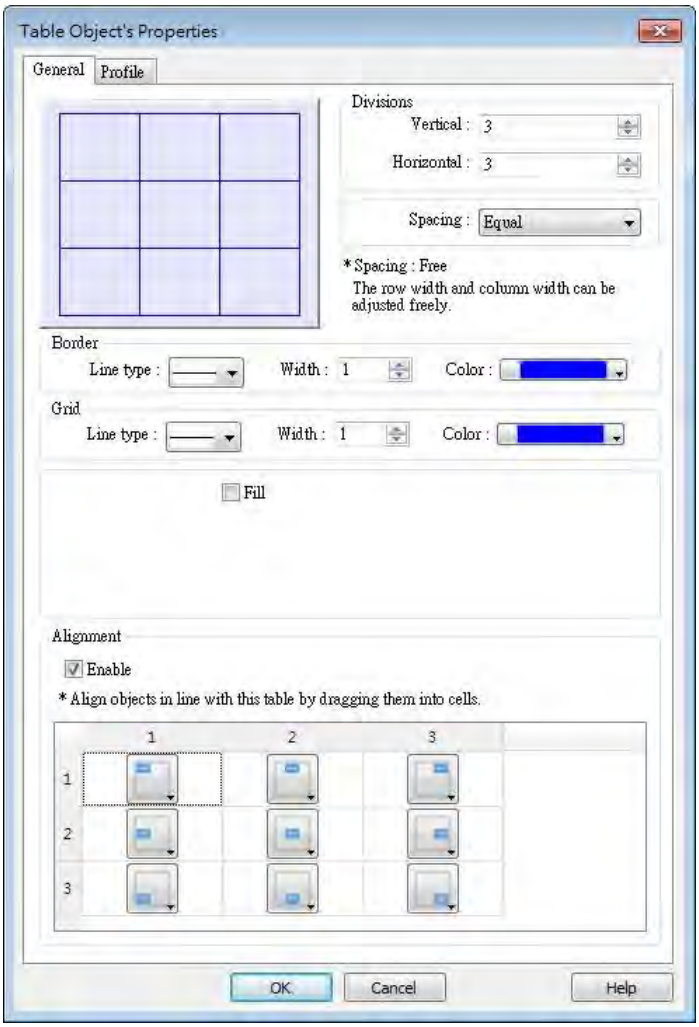
13.48.1. Overview

Table object allows users to draw a table in the editing window, and customize the border, grid, and pattern of the table.

13.48.2. Configuration

Click the Table icon on the toolbar to create a Table object. Or, click [Object] » [Table] in the menu.

General Tab



Setting	Description
Preview Window	Displays the settings result.
Vertical	Sets the number of columns in the table. Range: 1~255
Horizontal	Sets the number of rows in the table. Range: 1~255
Spacing	The available options are [Equal] and [Free]. When [Free] is selected, the user can manually adjust the column width or row height in the editing window.
Border	Sets the type, width, and color of the border. [Line width] setting is available only when Solid line type is selected. The range of width is 0~8. The line will become invisible when the width is set to 0.
Grid	Sets the type, width, and color of the grid. [Line width] setting is available only when Solid line type is selected. The range of width is 0~8. The line will become invisible when the width is set to 0.
Fill	Sets the pattern style and color.
Alignment	By dragging the objects into the cells in the Table, the objects can align to the preset position. Different alignments can be set for each cell in the Table, by default the objects are center-aligned, and nine different alignments can be selected as shown in the screenshot of the settings window.

13.49. VNC Viewer

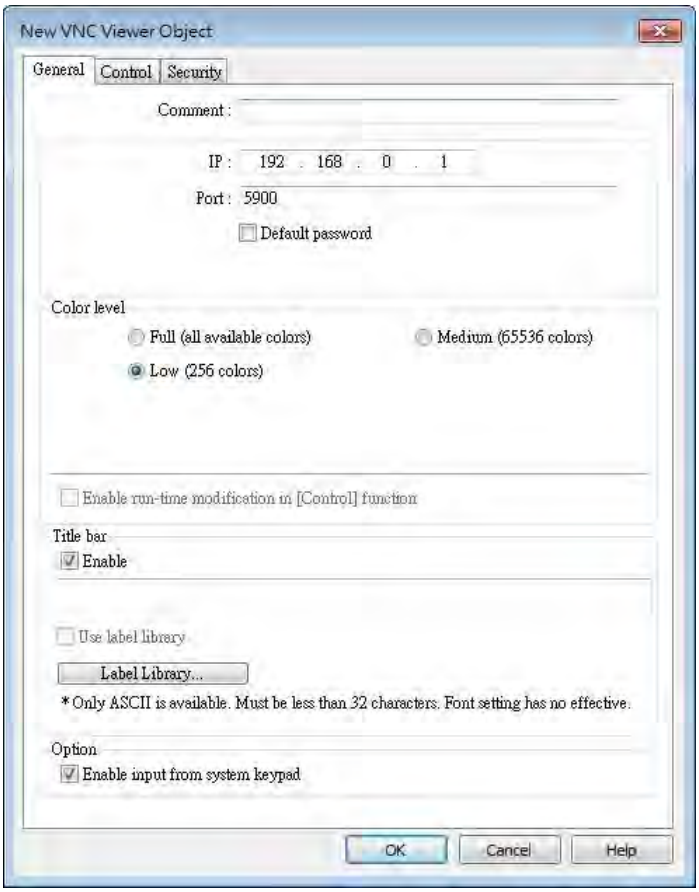
13.49.1. Overview

VNC Viewer can run on HMI to control a PC or a device remotely. VNC server must be installed on the remote device to be connected. On HMI the user can monitor and control the remote device.

13.49.2. Configuration

Click on the VNC Viewer icon on the toolbar or select [Objects] » [VNC Viewer] to open the settings dialog box. Configure the parameters and click OK; a VNC Viewer object will be created.

General Tab



Setting	Description
IP	Enter the IP address of the remote device to be

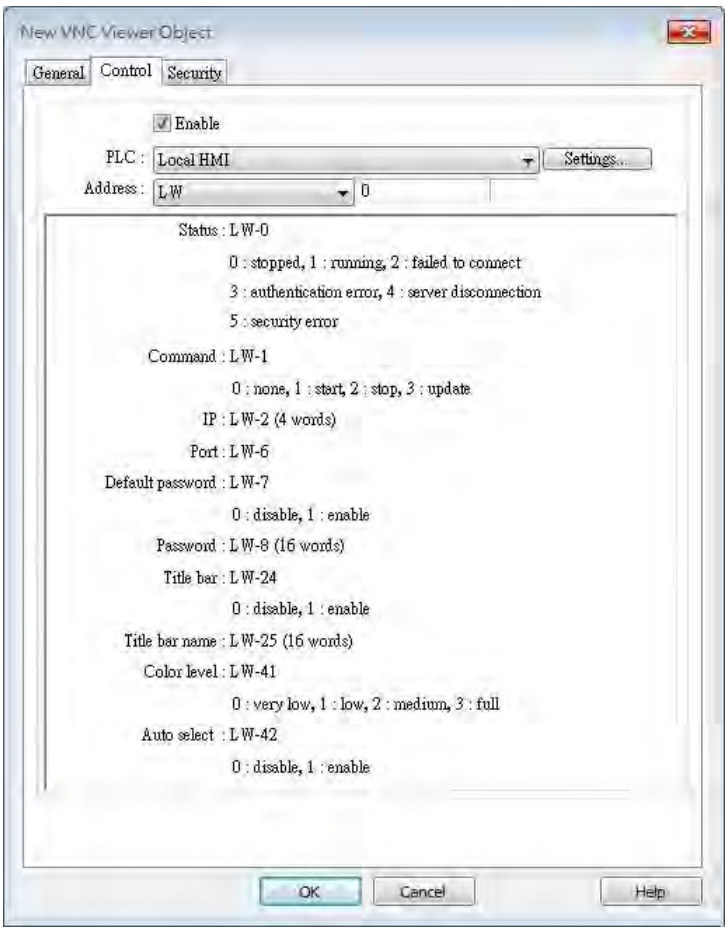
	connected.
Port	Enter the port number of the remote device to be connected.
Default Password	Enter the VNC password to log in the VNC server of the remote device. If Default Password is enabled, when connecting the remote device, VNC Viewer will automatically use the password specified here to log in, and the user doesn't need to enter the password.
Color level	Select from four color levels: Full (all available colors), Medium (256 colors), Low (64 colors), Very low (8 colors). **Auto select (not all servers support this function)** Allow VNC Viewer to automatically detect and select the color level supported by the server used. **Enable run-time modification in [Control] function** Select color level or decide whether to enable [Auto select] in HMI runtime using the control addresses that can be specified in the Control tab in the VNC Viewer settings dialog box.
Title bar	**Enable** When [Enable] is selected, a field shows for entering the caption in the title bar. The caption is limited to ASCII characters, and the font cannot be customized. The caption can be selected from Label Tag Library. Only when the title bar is enabled can the VNC Viewer window be moved or resized by dragging.
Option	**Enable input from system keypad** With this option selected, a keypad icon shows in the upper right corner of the cMT Viewer object, and the system keypad can be opened by clicking this icon.

Note

- eMT/iE/XE/mTV: VNC Viewer is supported for OS version 20160418 or later.
- cMT / cMT X Series: VNC Viewer is supported only on cMT series HMI screen accessed locally, and for OS version 20180928 or later.
- VNC Viewer cannot be simulated in on-line simulation mode.

- When [Default password] is selected, HMI's Virtual Keyboard can only be called out manually. If [Default password] is not selected, the Virtual Keyboard can pop up automatically.

Control Tab

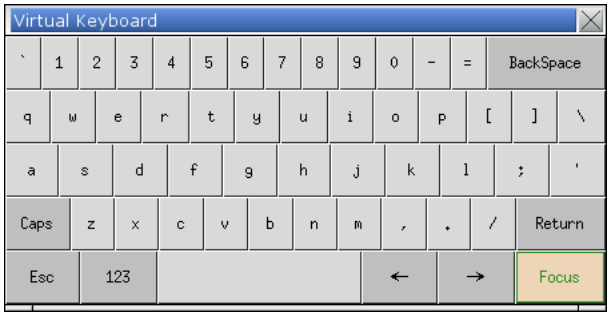


Setting	Description
IP	A set of word addresses can be specified to control VNC Viewer as well to display the connection status. **Control address: Shows the connection status** 0: Stopped 1: Running 2: Failed to connect 3: Authentication error 4: Server disconnection error 5: Security error **Control address + 1: Command** 0: None 1: Start

	2: Stop **Control address+2~+5: IP** **Control address+6: Port Number** **Control address+7: Default Password** 0: Disable 1: Enable **Control address+8: Password (16 words)** **Control address+24: Title bar** 0: Disable 1: Enable **Control address+25: Title bar name (16 words)** **Control address +41: Color level** 0: Very low 1: Low 2: Medium 3: Full **Control address +42: Auto select** 0: Disable 1: Enable

Note

- The allowable value range that can be entered in Control Address+6 is 0~99. The actual Port Number will be the value entered in Control Address+6 plus 5900. For example, if the user enters 1 in Control Address+6, the actual Port Number will be 5901. However, if the user enters 100 in Control Address+6, the value will not be added by 5900, and the actual Port Number will be 100.
- In VNC Viewer, to use the HMI built-in Virtual Keyboard, please tap the Focus button on the Virtual Keyboard first, and then tap VNC Viewer screen. This will change the input target.



Click the icon to download the demo project. Please confirm your internet connection

before downloading the demo project.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

13.50. Contacts Editor

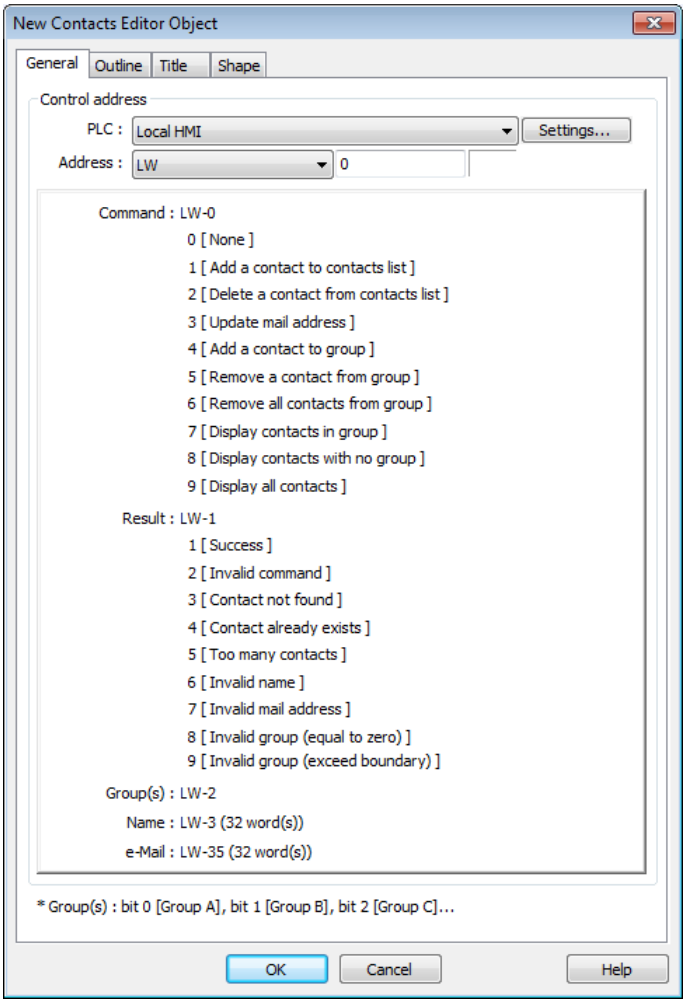
13.50.1. Overview

Contacts Editor enables users to dynamically add / modify / delete email contacts on HMI.

13.50.2. Configuration

Click the Contacts Editor icon on the toolbar to create a Contacts Editor object. Or, click [Objects] » [Contacts Editor] in the menu.

General Tab



Setting	Description
Control Address	A set of word addresses can be specified to change contact list or to show results.

Control address: Gives commands.

Value	Command
0	None
1	Add a contact to contacts list
2	Delete a contact from contacts list
3	Update mail address
4	Add a contact to group
5	Remove a contact from group
6	Remove all contacts from group
7	Display contacts in group
8	Display contacts with no group
9	Display all contacts

Control address + 1: Shows execution result.

Value	Result
1	Success
2	Invalid command
3	Contact not found
4	Contact already exists
5	Too many contacts
6	Invalid name
7	Invalid mail address
8	Invalid group (equal to zero)
9	Invalid group (exceed boundary)

Control address + 2: Group(s), uses bits to represent groups.

Value	Commands
0	Group A
1	Group B
2	Group C
3~15	Group D ~ Group P

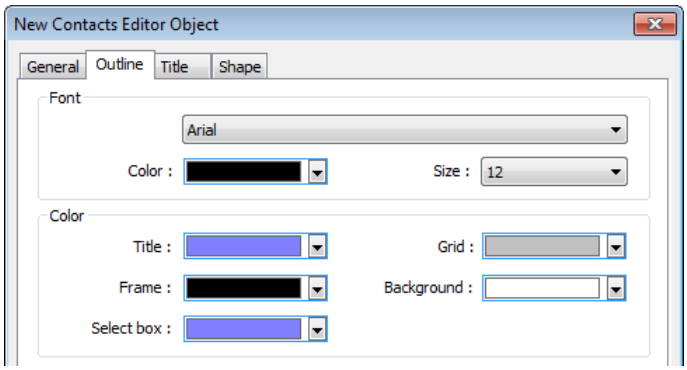
Control address + 3: Name (32 word(s)), contact name

Control address + 35: e-Mail (32 word(s)), e-mail address

Note

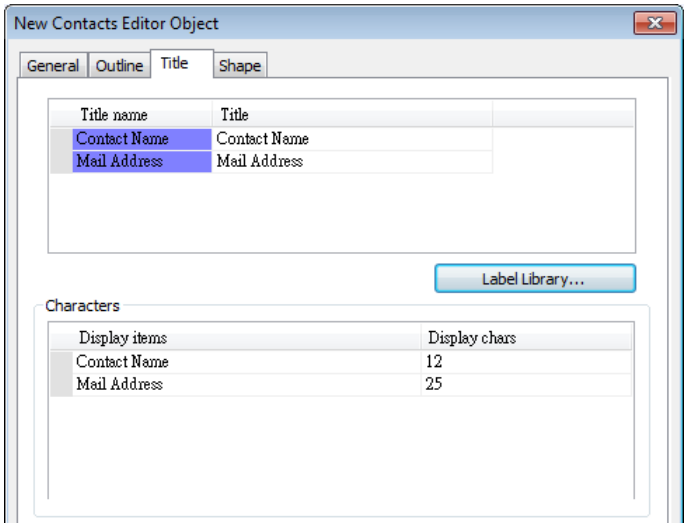
- General tab cannot be found when the model used is a cMT / cMT X model.
- Contact names do not support Unicode.
- The number of groups is specified in [System Parameter Settings] » [e-Mail] » [Recipients]. Please note that the number of groups cannot be dynamically changed on HMI.

Outline



Setting	Description
Font & Color	Select Contacts Editor object's font and color.

Title



Setting	Description
Title	The title shown in Contacts Editor.
Display chars	The displayable data length of each title in Contacts Editor. Range: 1~60

13.51. Event Bar Chart

13.51.1. Overview

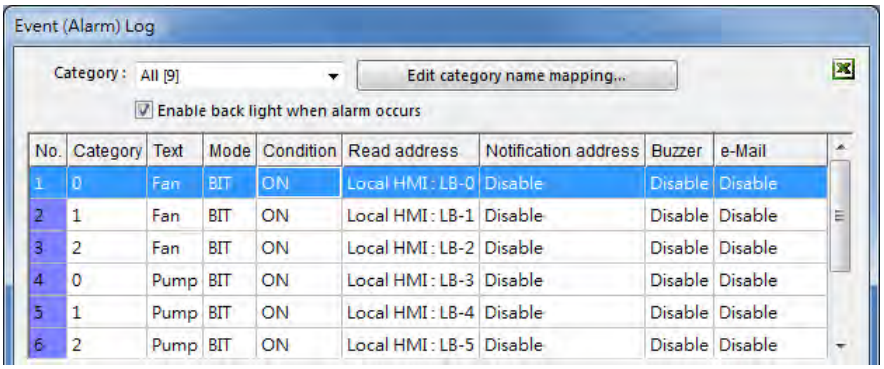
Event Bar Chart is a type of easy-to-use bar chart that can comprehensively illustrate project schedule. Using Event Bar Chart to illustrate HMI events or alarms can help users to clearly understand the time at which an event or alarm occurs, and its duration. Before drawing an Event Bar Chart, please configure Event Log object first.

This feature is only supported on cMT / cMT X Series models (excluding cMT-Gateway).

13.51.2. Configuration

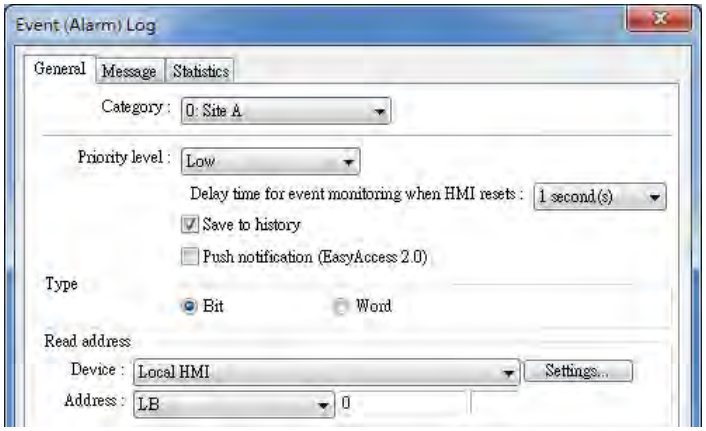
13.51.2.1. Event Log

Click [Data/History] » [Event Log] in the menu to configure several event logs. This section describes the Event Bar Chart related settings that can be found in the Event Log settings dialog box.



Setting	Description
Edit category name mapping	Event Bar Chart will show the name of each category.

13.51.2.2. General Tab



Setting	Description
Category	Event Bar Chart will illustrate the duration of all events in one category, please select correct category in this field.

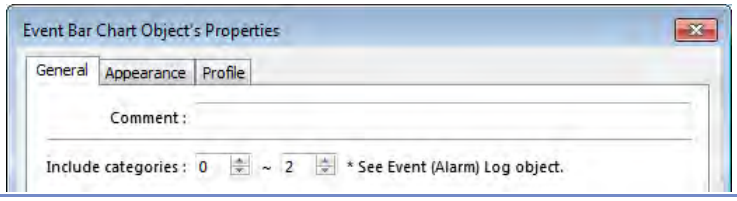
13.51.2.3. Message Tab

Setting	Description
Text	Enter the name of the event log.
Color	Select the color for the bar shown in Event Bar Chart that illustrates this event log.

13.51.2.4. Event Bar Chart

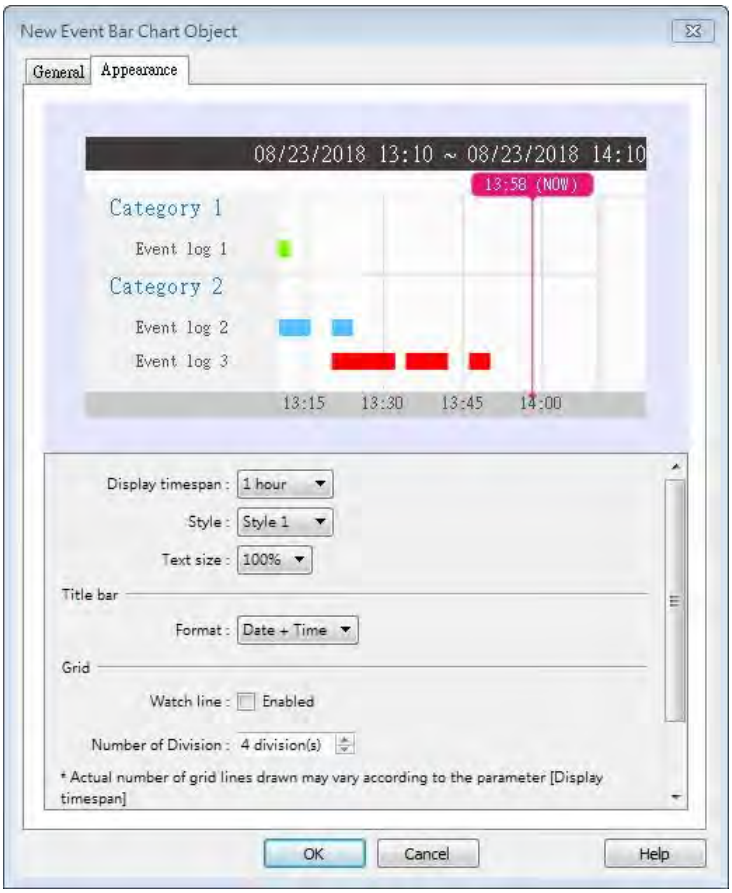
Click [Data/History] » [Event Bar Chart] in the menu to draw an Event Bar Chart.

General Tab

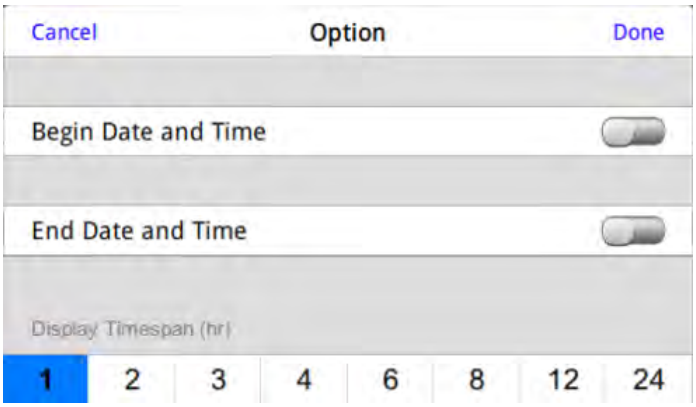


Setting	Description
Include categories	Select the categories that will be shown in Event BarChart.

Appearance Tab

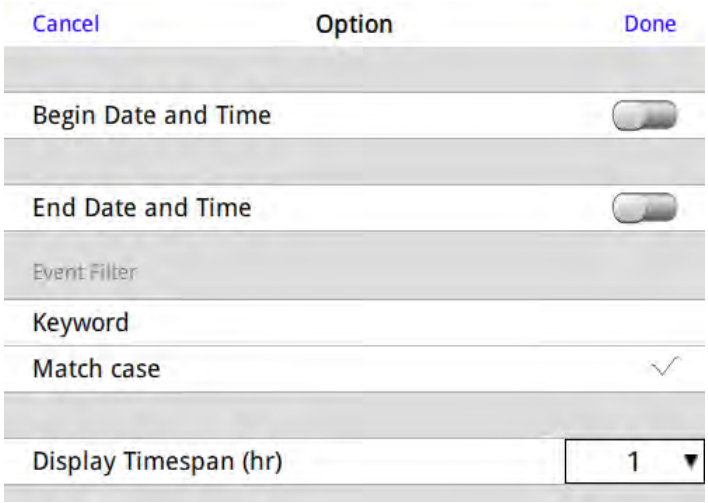


Setting	Description
Display timespan	Specify the time interval that is measured as a number of

	hours. Events occur during this time interval will be illustrated in Event Bar Chart. Tapping in the upper right corner of the HMI screen can also change this setting. 
Style	By default 3 styles are provided for users to choose from.
Text size	Text sizes range from 50%~200%.
Title bar Format	Date + Time: 08/16/2018 13:55 ~ 08/16/2018 14:55 Date only: 09/03/2018 ~ 09/04/2018 Time only: 13:57 ~ 14:57
Watch line	A watch line shows at the point in the Event Bar Chart that is touched. The time represented by the touched point will show at the top of the watch line.
Number of Division	The number of divisions on X axis.
Time label Format	The format in which the time label is displayed.

HMI Settings

On cMT HMI or cMT Viewer, tap the icon in the upper-right corner of the object to open the following settings window.



Setting	Description
Begin Date and Time	Specify the begin date and time at which Event Bar Chart displays data. By default this setting is disabled, and Event Bar Chart displays from the begin date and time of the stored data.
End Date and Time	Specify the end date and time at which Event Bar Chart displays data. By default this setting is disabled, and Event Bar Chart ends at the end date and time of the stored data.
Event Filter	**Keyword** Events can be filtered by entering a keyword. **Match case** Use case-sensitive search when searching for events in English.
Display Timespan (hr.)	Dynamically change the time range (1~96 hours.) shown in Event Bar Chart. Select all to display all existing events.

Note

- When HMI power is off, the HMI will not be able to obtain the alarm states; therefore, the alarm states during the power off period will not be illustrated in Event Bar Chart, not even after the power turns on.

13.52. Action Trigger

13.52.1. Overview

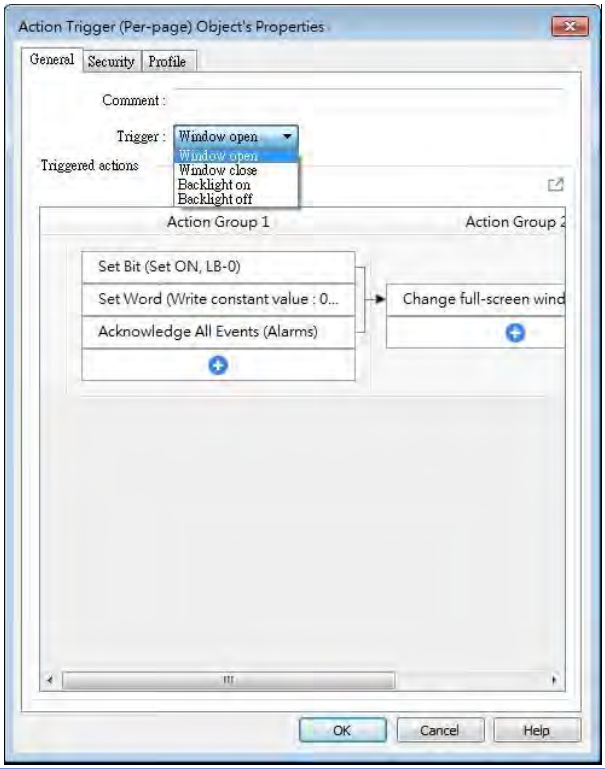
The Action Trigger object can trigger a sequence of action groups when the specified condition is met. Users can divide actions into action groups and the actions within the same group are executed at the same time when the specified condition is met. When all the actions within the same group are completed, the actions in the next group are then executed. This object is only available on cMT / cMT X Series models.
Action Trigger has two types, each with different modes for triggering action execution:
Action Trigger (per-page): [Window open/close] and [Backlight on/off] modes.
Action Trigger (Global): [Idle timeout], [Value changed], [Control token], [User secure access], and [Condition object] modes.

13.52.2. Configuration

Click [Object] » [Action-related] » [Action Trigger (per-page)] or [Action Trigger (Global)] icon on the toolbar to open an Action Trigger object property dialog box. Clicking [New] can add a new Action Trigger.

General Tab

Action Trigger (Per-page)



Action Trigger (Global)

Setting	Description
Mode (per-page)	**Window open:** Triggers actions when window opens. Settings in Security tab will not be available when using this mode. **Window close:** Triggers actions when window closes. **Backlight on:** Triggers actions when backlight turns on. **Backlight off:** Triggers actions when backlight turns off.
Mode: Idle Timeout (Global)	If the screen is left untouched for more than the specified time limit configured in [Idle Timeout], Action Trigger will start execution. Range: 1~43200 seconds
Mode: Value Changed (Global)	When the state of the designated register meets the preset condition, Action Trigger will start executing. **Check condition recursively when actions ended** When all actions are completed, the trigger condition will be checked again; and if the condition is met, the actions will be triggered again. **Dynamic condition value** This option is available when [Word] is selected as Type. Selecting this option opens the [Read/Condition use different addresses] option. When [Read/Condition use different addresses] is not selected, the condition value is read from the next contiguous

	register of [Value address].
Control Token (Global)	Triggers actions when the Control Token is acquired or unacquired.
User Secure Access (Global)	Triggers actions when access to the specified security class is obtained or lost. When the security class is [Any], the actions can be triggered, regardless of class, when the access is obtained or lost.
Condition Object (Global)	Triggers actions when the conditions set in Condition object are met. **Check condition recursively when actions ended** When all actions are completed, the trigger condition will be checked again; and if the condition is met, the actions will be triggered again.
Action Group	Actions are classified into groups. Actions in the same group will simultaneously be triggered, and only after the actions in the previous group have been all triggered, the actions in the next group will then be triggered.
Delay	Delays the action for the specified time period (milliseconds).
Set Bit	Sets the designated bit address ON or OFF. **Set ON** Sets ON the designated bit. **Set OFF** Sets OFF the designated bit. **Toggle** Alternates the bit state.
Set Word	Changes the value in the designated word address. **Write constant value** Writes the constant value to the designated register. **Increment value (JOG+)** Increases value in register by a set amount in [Inc. value], up to the [Upper limit]. **Decrement Value (JOG-)** Decreases value in register by a set amount in [Dec. value], down to the [Bottom limit]. **Dynamic Limits (JOG+, JOG-)** Sets the [Upper limit](JOG+) and [Bottom limit](JOG-) by a designated register. **Write constant string** Writes the constant string to the designated register. **Object Control Command (cMT / cMT X only)**

	Listed for selection are commands (with their corresponding command values) available for the control address of various functions. **System Tag Command (cMT / cMT X only)** When a system tag is set as write address; for example, LW-9134: Language Mode, the commands relating to the system tag can be selected.
Change window	Switches to the designated window. Change full-screen window can only be the last action in the last group in an Action Trigger. **Change full-screen window:** Changes to another base window. **Change common window:** Changes common window. **Return to previous window:** Changes from current screen to the previous one displayed. For example, when window no. 10 is changed to window no. 20, this function can be used to return to window no. 10. This function is only available for base window. **Animation Setting:** The effects are: Fade, Fly, Float, Wipe, Split, Circle, Clock, Zoom, Turn, Push. Different effects may be used for Start (window appears) and End (window disappears). [Duration] specifies how many milliseconds (ms) a transition effect takes to complete. [Direction] The direction of the transition.
Execute Macro	Executes one of the Macros from the drop-down list that has already been configured by users. Running a macro may take a while to complete; therefore, when [Execute Macro] is put into an action group, it is seen as completed when it is triggered. The system will not wait for the macro to complete running before moving on to the next action group.
Popup Window	Pops-up a designated window.
Popup Calculator Window	Opens a calculator with an unchangeable appearance, allowing operators to do simple calculation and enter the result to a destination object by pressing the Enter key.
Keyboard Input	Inputs characters/control signals, as often used with [Numeric] or [ASCII] objects. **Enter:** Same as the keyboard's "Enter" function. **Backspace:** Same as the keyboard's "Backspace" function. **Clear**: Clear the value in the word register.

	Esc: Same as the [Close window] function; it is used to close the keyboard window. **Delete:** Same as the keyboard's "Delete" function, deletes the number or character on the right side of the text cursor. **Left:** Same as the keyboard's "←" key moves the text cursor to the left side of the previous number or character. **Right:** Same as the keyboard's "→" key moves the text cursor to the left side of the next number or character. **Inc:** Add 1 to the current value. **Dec:** Minus 1 from the current value. **ASCII/UNICODE:** Specify the character to be entered by this key.
Screen Hardcopy	Saves current screen as a hard copy file into a SD card or USB disk.
Acknowledge all events (Alarms)	Acknowledges all events at once.
Import Data	Imports the e-mail contacts or user accounts, or logs in using USB Security Key. **Data Position:** The external device to read data from. Options: USB disk or SD card. **Account import mode:** If [Overwrite] is selected, there will be only imported accounts after import. If [Append] is selected, there will be imported accounts in addition to existing accounts after import. **Delete file after importing user accounts:** The system will delete the account data saved in the external device after importing; this can prevent the account data from leaking out.
Wait Until	The next group will be executed only when the condition set for a designated bit or word address is met. However, if the condition is not met even after the set wait time (timeout), it can either move on to the next action group or end prematurely without continuing.
Data Transfer (Global)	Sends the value in the designated address to another address.
File Transfer	Transfer files by FTP. HMI will connect to the FTP server in passive mode. **General Tab:** Download: Transfer file from FTP server to local HMI. Upload: Transfer file from local HMI to FTP server.

	Server address: [Direct]: Designate server address related settings in EBPro. [Dynamic]: Designate an address for changing the server address related settings dynamically on HMI. Suppose the dynamic address is set to LW-n, then: LW-n: Host usage (0: IP, 1 Domain name) LW-n+1: IP (4 words) LW-n+5: Port LW-n+6: Authentication (0: None, 1: Account) LW-n+7: Username (16 words) LW-n+23: Password (16 words) LW-n+39: Domain name (64 words) **Delete the source file after successful transfer** Following successful transfer, the source file will be deleted. When transferring in [Download (FTP -> HMI)] mode, please make sure that the account has the write permission on the FTP server. **File Tab:** Set file position and full path of FTP server / Local HMI. When a folder path is specified, all the files in that folder will be transferred, not including files in subfolders. When a file with identical file name already exists, it will be overwritten regardless. **Status Tab:** Designate an address for showing file transfer result and FTP server response. Please use the following link for more information on FTP server return codes. https://en.wikipedia.org/wiki/List_of_FTP_server_return_codes
Control Token	The action to acquire or unacquire a Control Token. **Control Token:** Select a Control Token. **Action:** Set this action to acquire or unacquire a Control Token. **Max. waiting time:** Continues or stops running the next action group if the Control Token is still not acquired even after the set [Max. waiting time].

Note

- At most 1000 Action Triggers can be created in a project.
- At most 20 groups can be created in an Action Trigger, and at most 20 actions can be added into a group.

- File Transfer Error Codes:

Error Code	Description
0	File transferred successfully.
1	The HMI directory for download does not exist.
3	USB disk or SD card is not found.
4	HMI or FTP directory is empty.
5	Uploaded file does not exist.
8	Operation rejected by FTP server.
9	USB disk or SD card is full.
10	Unknown error.

13.53. Calendar

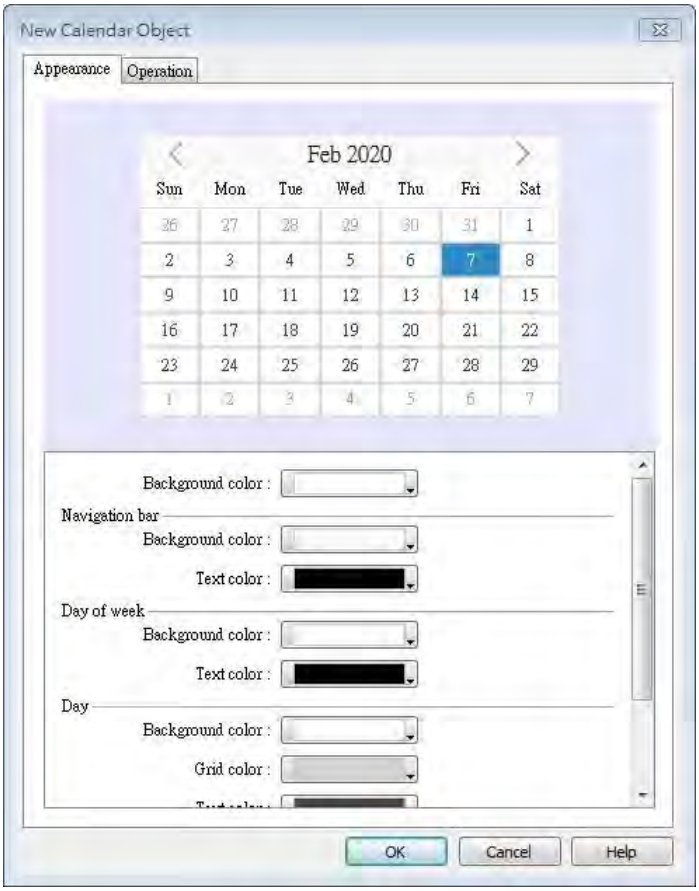
13.53.1. Overview

The Calendar object can show a calendar on cMT / cMT X Series HMI.

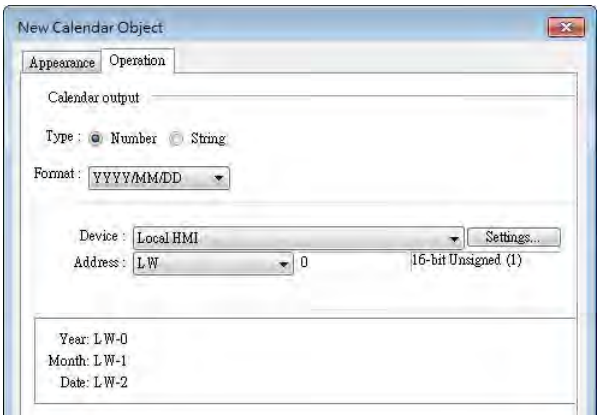
13.53.2. Configuration

Click [Object] » [Time-related] » [Calendar] icon on the toolbar to open a Calendar object property dialog box. Set up the properties, press OK button, and a new Calendar object will be created.

General Tab



Setting	Description
Appearance	**Navigation bar** Set the background color and text color of the navigation bar. **Day of week** Set the background color and text color of the days of

	week. **Day** Set the background color, grid color, and text color of the days. **Today** Set the background color and text color to show today.
Operation	Output the date information in a number or a string to the designated address. 

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

13.54. Touch Gesture

13.54.1. Overview

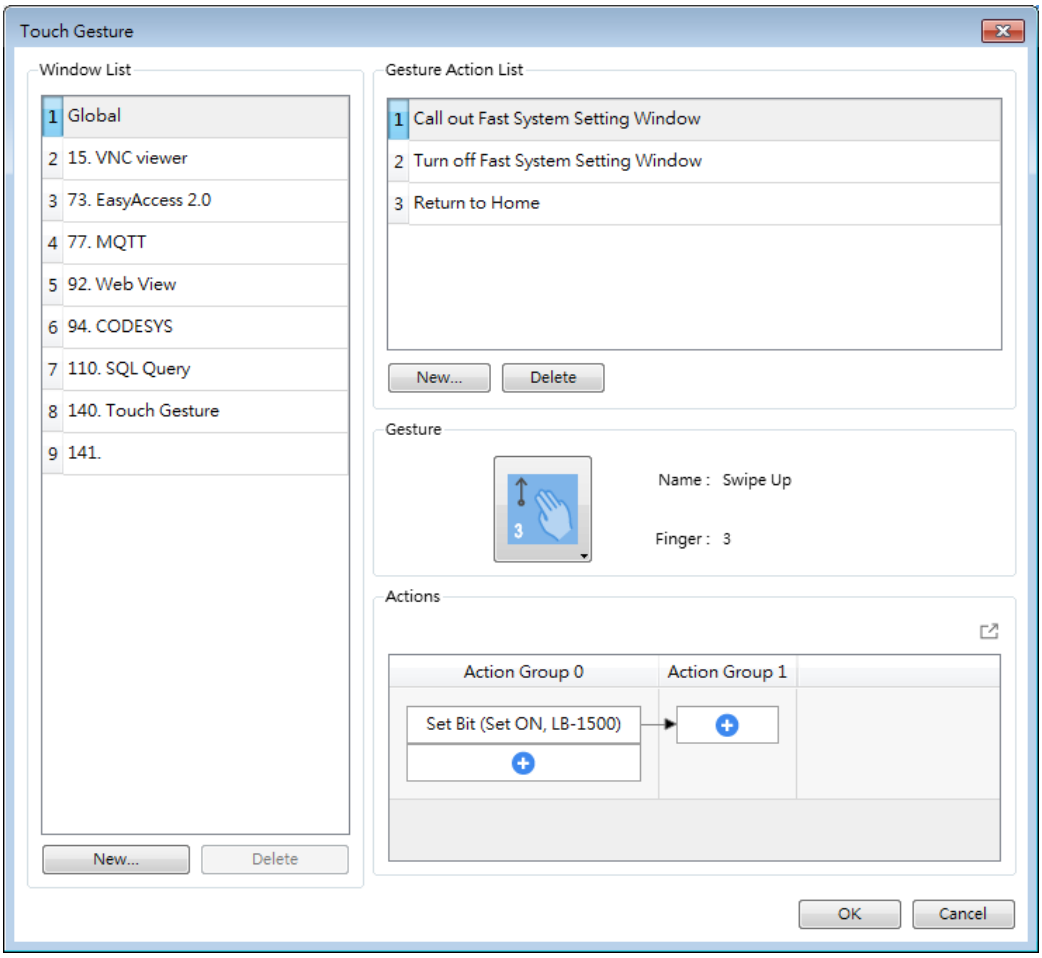
For some time, smartphones or tablets nowadays have been using gesture-based operation, e.g., pinch with five fingers to close an app. Now, with HMI having Touch Gesture object that supports up to 23 touch gestures, HMI can provide operating experience comparable to that of a smartphone or tablet.

The Touch Gesture finger limit is 5.

13.54.2. Configuration

Click [Object] » [Action Related] »[Touch Gesture] icon on the toolbar to open the Touch Gesture property dialog box. Select a window and then add its allowable touch gestures.

General Tab



Setting	Description
Window List	Allow the gesture actions to be used in a specific window or all windows. When Global is selected, one-finger gestures are not supported. **Add/Delete** Add or delete a window that allows this gesture action.
Gesture Action List	Shows gesture action list for the window. **Add/Delete** Add or delete a gesture action.
Actions	Set an action or group of actions triggered using this gesture. For more information on the actions, see Chapter 13.36 or 13.52 in this manual.

Note

- Up to 16 gestures can be used for each window setting (global/individual).
- When the same gesture is used in both the Global window and an individual window, the gesture is effective only for the individual window.
- When a user performs a gesture on the screen, objects are still triggered when they are touched.
- The touch gesture for an individual window is only effective when the window is a base window. If the said window is a pop-up window, or overlaid windows, its touch gesture will be ineffective and its gesture actions will not be executed.
- Detection for the next gesture will only start when the actions for the current gesture have been completed.

13.55. PLC Web Browser

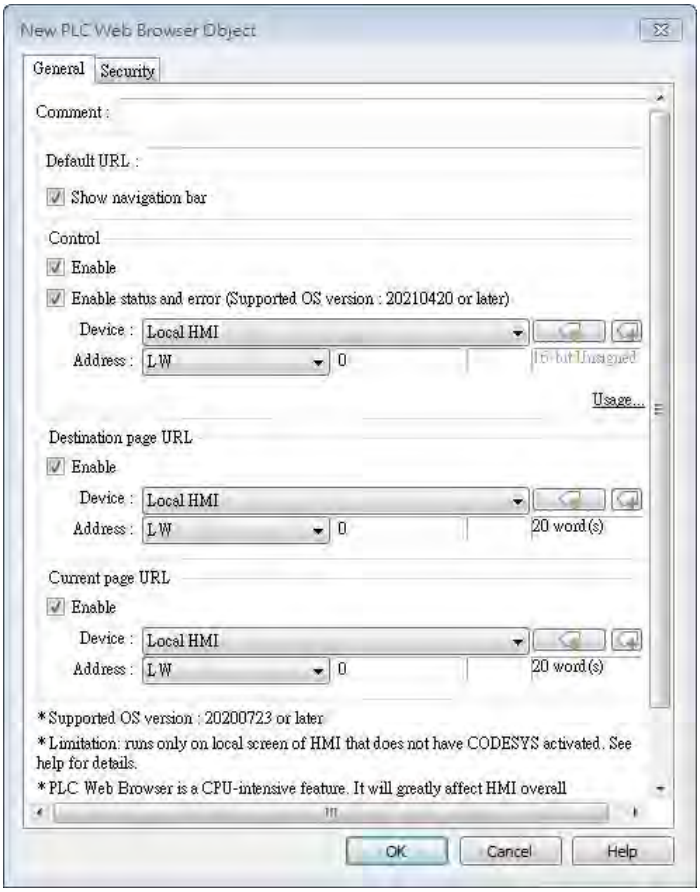
13.55.1. Overview

The PLC Web Browser object can be used for browsing PLC web pages. PLC Web Browser is supported only on cMT3072X, cMT3072XH, cMT3092X, cMT3152X, and cMT3162X.

13.55.2. Configuration

Click [Object] » [Media] » [PLC Web Browser] icon on the toolbar to open a PLC Web Browser object property dialog box. Set up the properties, press OK button, and a new PLC Web Browser object will be created.

General Tab



Setting	Description
Default URL	Enter the URL into the provided field.
	Show navigation bar

With this option selected, users can directly change URL, refresh page, or zoom in/out web page by clicking the icons in the navigation bar displayed on HMI.

Icon	Command
←	Go back to the previous page
→	Go forward to the next page
	Zoom out the page
	Zoom in the page
	Return to homepage
	Refresh the page

Control address

LW-n: Command ID

Value	Command
0	None
1	Go destination
2	Reload
3	Stop
4	Go back
5	Go forward
6	Clear cache
7	Clear cookie

LW-n+1 Status

Value	Command
0	Load started
1	Load stopped
2	Load succeeded
3	Load failed

LW-n+2 Error Code

Value	Command
0	No error
1	Internal error
2	Connection error
3	Certificate error
4	HTTP error
5	FTP error
6	DNS error

Destination page URL

The designated register is for entering a URL with length limit 255 words. After entering the URL, please enter value 1 in the Command

address	ID address to go to the destination page.
Current page URL address	Shows the URL of current page, the length limit is 255 words.

13.55.3. Notes on PLC Web Browser

1. This object runs on local screens of HMIs. Supported models and OS versions:
   cMT3072X, cMT3072XH: 20200723 or later
   cMT3092X, cMT3152X, cMT3162X: 20201230 or later
2. PLC Web Brower object is designed specifically for connection to PLC's built-in web server, allowing users to access PLC web page to configure PLC and view PLC data. Using PLC Web Browser to connect to other types of websites may lead to unexpected results or errors.
3. Displaying relatively complex web page using PLC Web Browser will occupy more memory and consume a large amount of CPU, which can significantly slow down HMI's operating speed. Users are advised to use change page function or a Direct/Indirect window to display PLC Web Browser object only when it is needed, in order to prevent reduction of system performance.
4. As HMI memory is limited, when the PLC Web Browser occupies too much memory, the system will automatically stop PLC Web Browser to ensure proper operation of HMI.
5. In consideration of performance, PLC Web Browser can only run on an HMI whose CODESYS is not activated.
6. Due to rapid change in web technologies, it is not guaranteed that all PLC web servers are supported by PLC Web Browser in EasyBuilder Pro, and certain PLC web pages may not be correctly displayed using PLC Web Browser.
7. When multiple PLC Web Browser objects are placed in a single window, only one of them will run while the rest remain ineffective.
8. PLC Web Browser object displays web page on top of everything else even if there are other objects placed above it.
9. When the PLC Web Browser is clicked or tapped in cMT Viewer running on a PC, smart phone or tablet device, the default browser for the device will be used to open the web page. Note that, in this case, to access the PLC within the same network as HMI, the device running cMT Viewer should also be in the same network as the PLC.
10. Browsing web pages using PLC Web Browser on cMT-iV5/cMT-iV6 is not possible.
11. Features not supported by PLC Web Browser:
    - Save file as…
    - Text selection
    - Right-click menu
    - Drag & Drop
    - Windows prompt for entering account name and password (e.g., FTP login)

- Playing videos, e.g. YouTube…etc.
- Uploading / Download files.

13.56. Condition

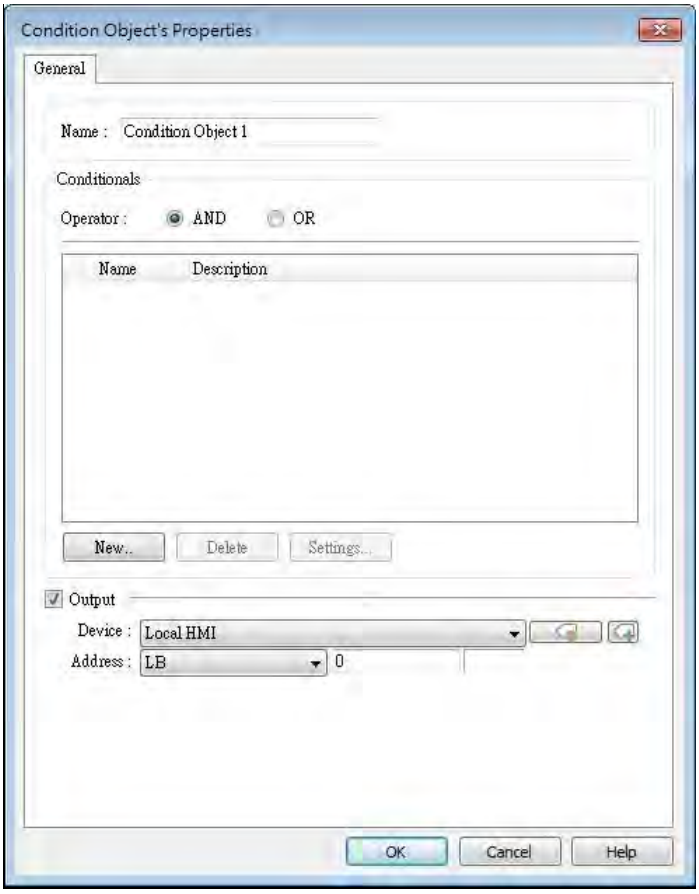
13.56.1. Overview

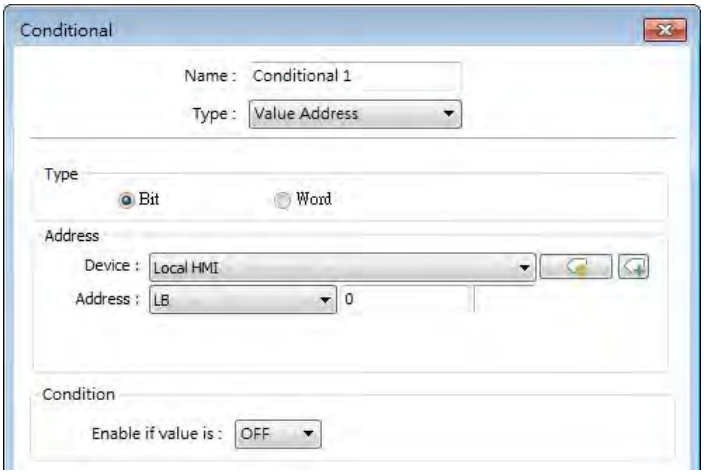
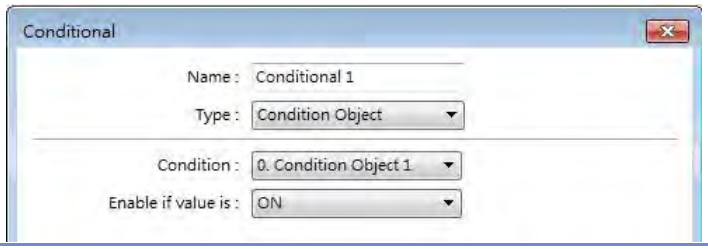
The Condition object can be used to replace the if-else statement in macro. When the specified conditions in the Condition object are met, the designated output bit address will be set ON. The output bit address can be used in objects' security settings, Event Log, Backup, and Data Transfer…etc. The Condition object can be selected in Action Trigger (Global) object's mode setting. Note that this object is available only on cMT / cMT X models.

13.56.2. Configuration

Click [Object] » [Condition] icon on the toolbar to open a Condition object property dialog box. Click [New] to add a new condition, and then click [OK] to create a Condition object.

General Tab



Setting	Description
Name	Enter the name of this object.
Conditionals	**Operator:** Select AND / OR operator. **AND:** The Condition object is effective when all the specified conditions are met. **OR:** The Condition object is effective when one of the specified conditions is met.
Type	**Value Address:** Select an address type from Bit / Word.  **Condition Object:** Select another Condition object. 
Output	With this option selected, when the conditions are met, the specified bit address will be set ON. When the conditions are not met, the specified bit address will be set OFF.

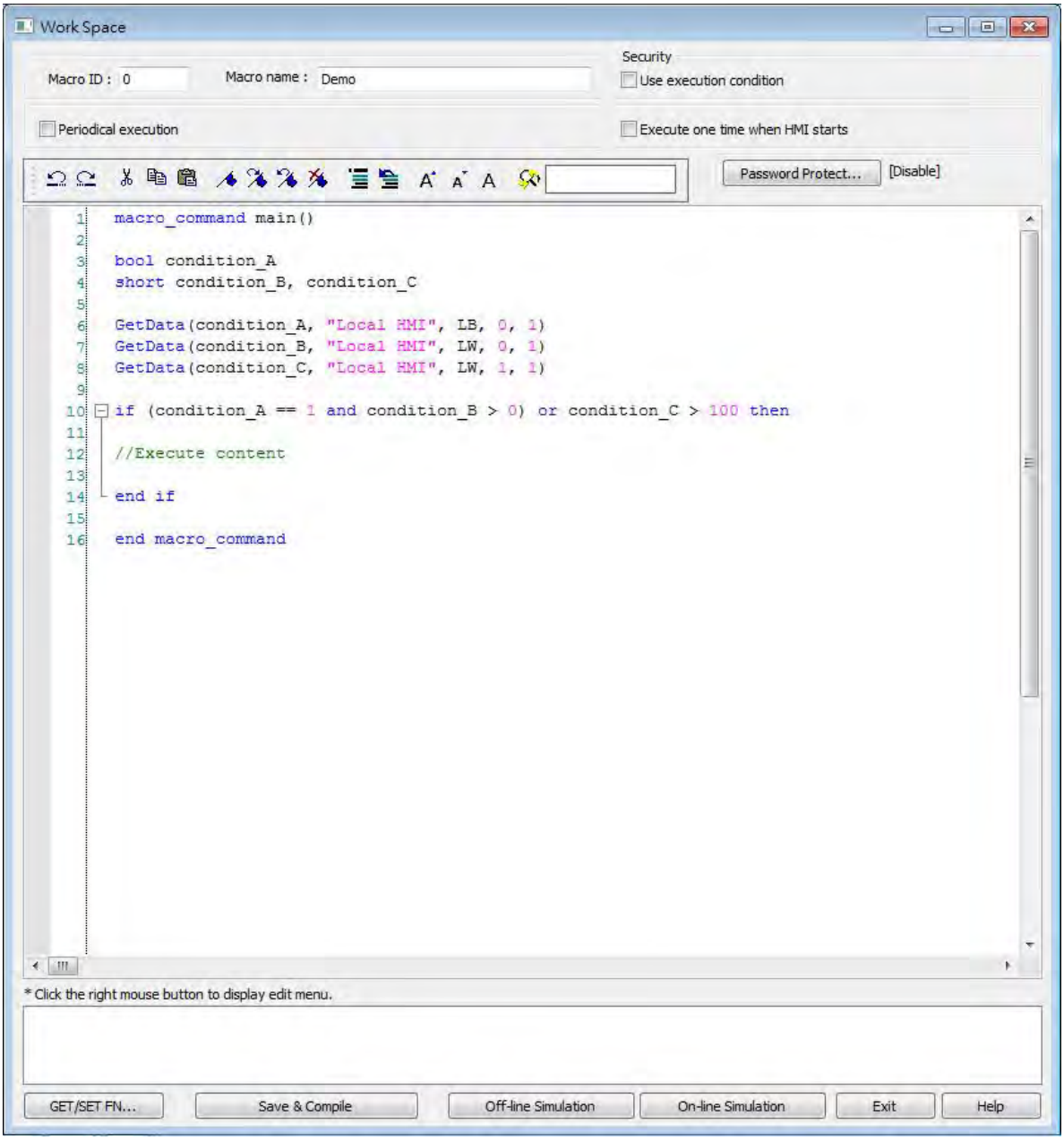
Note

- At most 150 Condition objects can be created in a project.
- At most 8 conditions can be created in a Condition object.

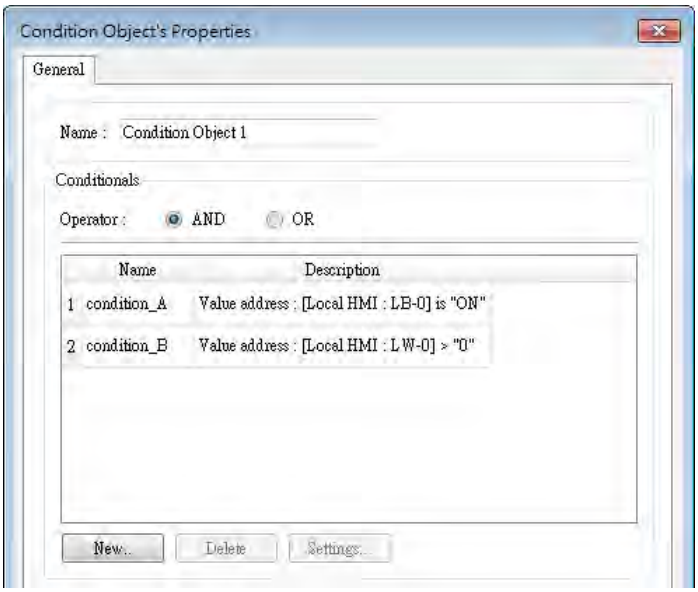
Example 1

Here is an example of using Condition object to replace the if-else statement in macro.

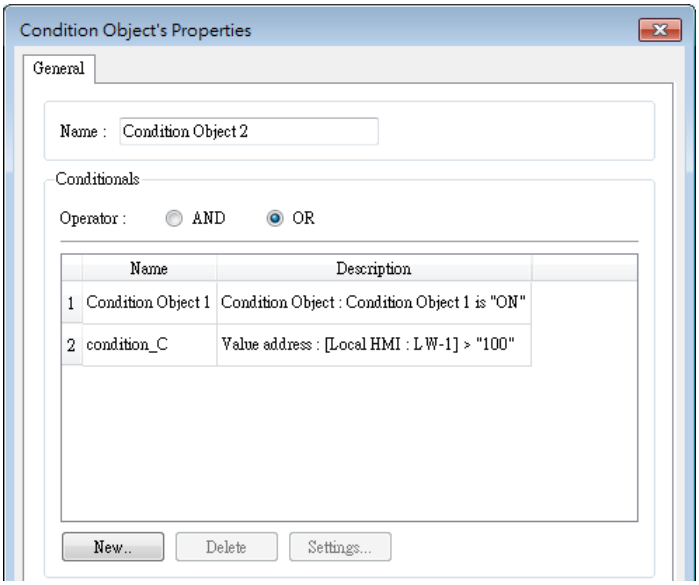
The macro below contains three conditions.



1. Create a Condition object, add condition_A and condition_B, and then select AND operator as shown below.



2. Create another Condition object, add Condition Object 1 created in the previous step, add condition_C, and then select OR operator as shown below.



3. Now the Condition objects can do the same thing as the macro, which triggers actions according to the conditions.

14. Shape Library and Picture Library

This chapter explains how to build Shape Library and Picture Library.

14.1. Overview 14-2
14.2. Building Shape Library 14-2
14.3. Building Picture Library 14-9
14.4. Immediate Modification and Preview 14-15

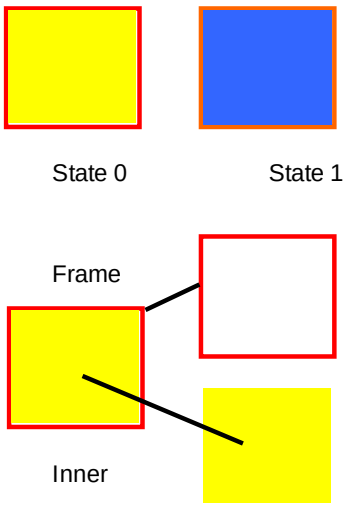
14.1. Overview

EasyBuilder Pro provides Shape Library and Picture Library for visual effects on objects.
Picture Manager provides two modes: [Project] and [Library]. Pictures in [Project] mode will be stored in .emtp project file. Pictures in [Library] mode will be stored in EasyBuilder Pro libraries, or the user-defined directory.
Each Shape or Picture includes up to 256 states. This chapter explains how to build Shape Library and Picture Library.

For more information about using libraries while creating an object, see “9 Object General Properties”.

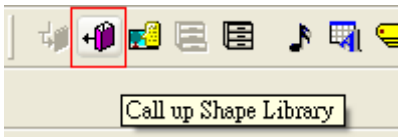
14.2. Building Shape Library

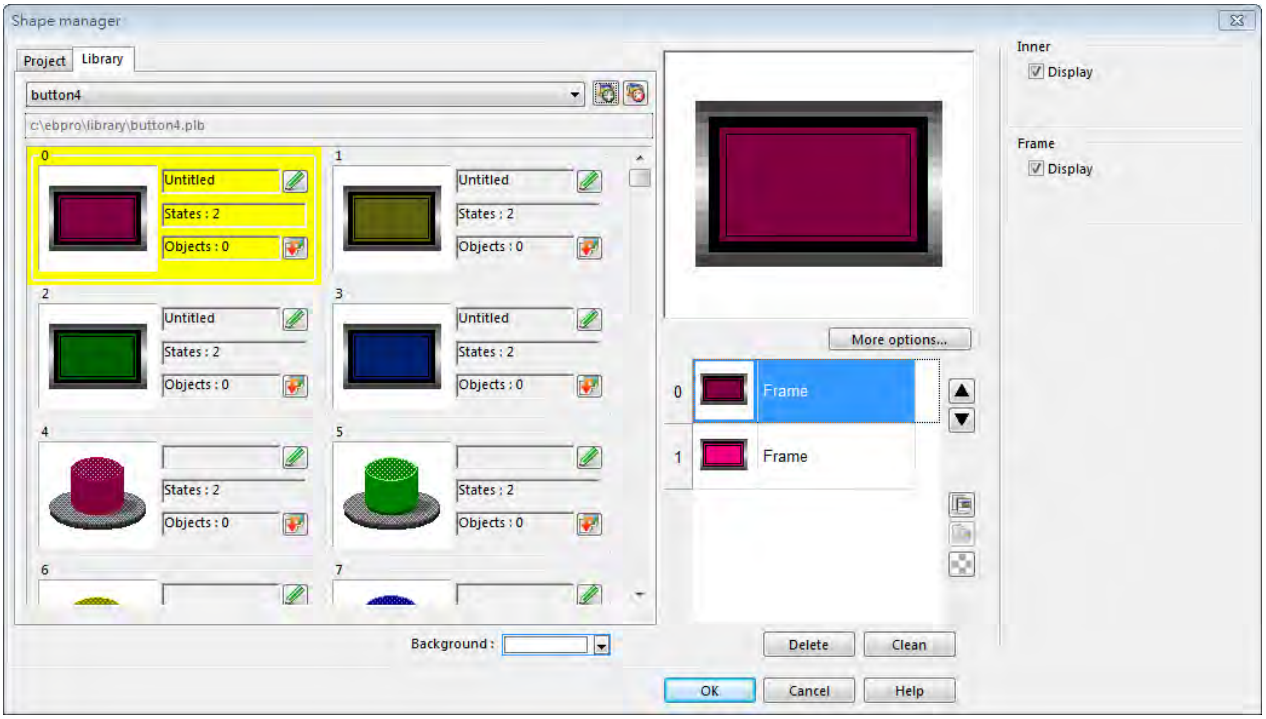
Shapes are vector graphics constructed by lines, curves or polygons. A Shape can have more than one state, and each state includes two parts: frame and inner, as shown in the following figure.



14.2.1. Shape manager

An object can use frame, inner or both. Click [Project] » [Shape], and the [Shape manager] dialog box appears.



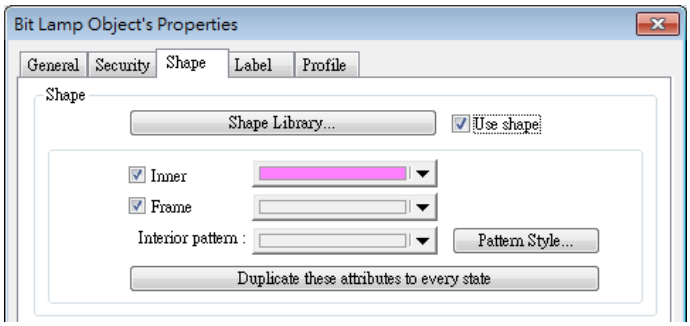


Setting	Description
Project	The Shape edited here will be saved in .emtp. Up to 1000 Shapes can be added.
Library	The Shape edited here will be saved to the library directory on PC and will not be saved to .emtp project file.
New library	Include existing .plb shape library files or create a new one. To create an empty library, enter a new file name and click [Open]. Up to 40 library files can be added.
Unattach library	Exclude currently selected library.
Copy to project	Copy the selected Shape to [Project]. Only the shapes that do not belong to the System Libraries can be copied. Shapes in System Frame/System Button/System Lamp/System Pipe cannot be copied.
Background	Select and preview the background color of the Shape. The color is only displayed in [Shape manager] dialog box, and is not displayed when placing the object in the screen.
More options	Set the color and style of [Inner], [Frame], and

	[Pattern].
	Move the Shape to the previous / next state.
Copy	Copy the selected Shape.
Paste	Paste the copied Shape.
Insert transparent state	Insert a blank state after the selected state.
Delete	Delete the selected state of the shape.
Clean	Delete all the states of the selected shape.
OK	Confirm to save the edited Shape.
Cancel	Cancel the editing event.
Help	Open help files.

Note

- The color of [Inner] and [Frame] can be selected in Shape Library. The selection of [Pattern Style] is only available in System Frame / System Button Library.



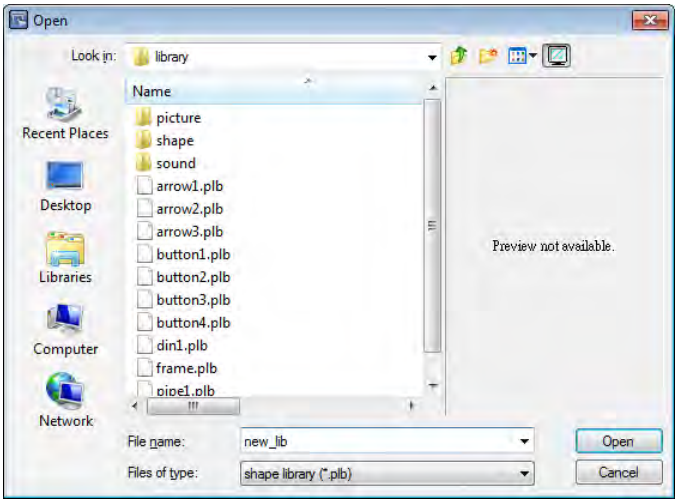
- cMT / cMT X Series supports using gradient patterns in [Pattern Style], as follow:

14.2.2. Steps to Build Shape Library

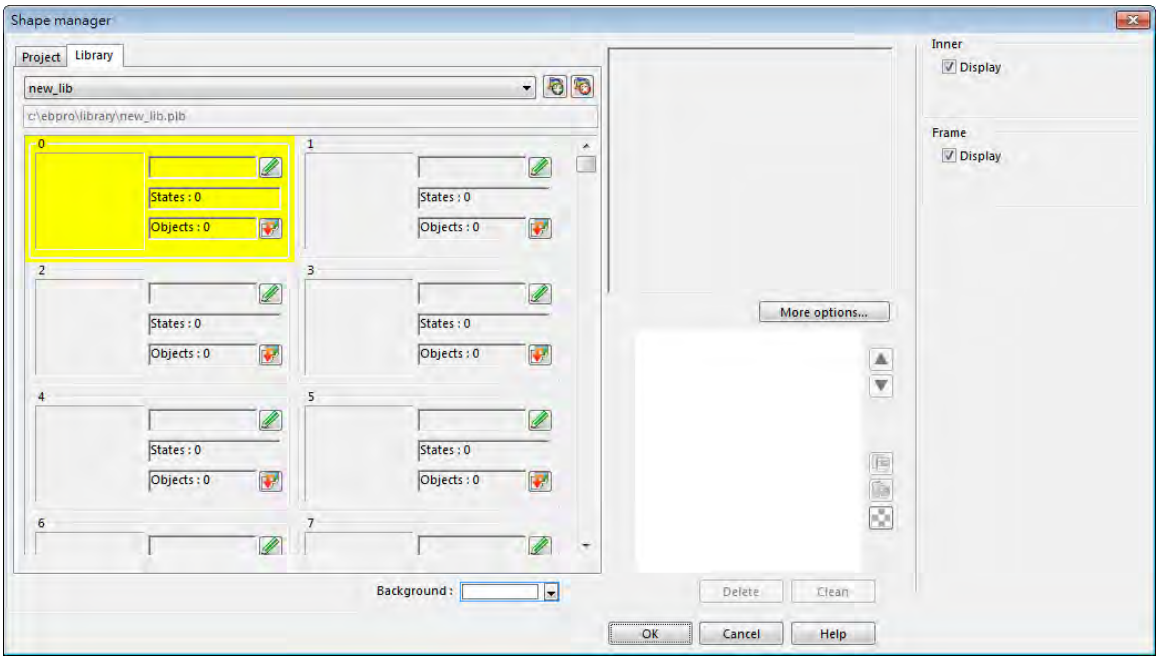
The following explains how to create a new Shape Library and add a Shape with two states into

the library.

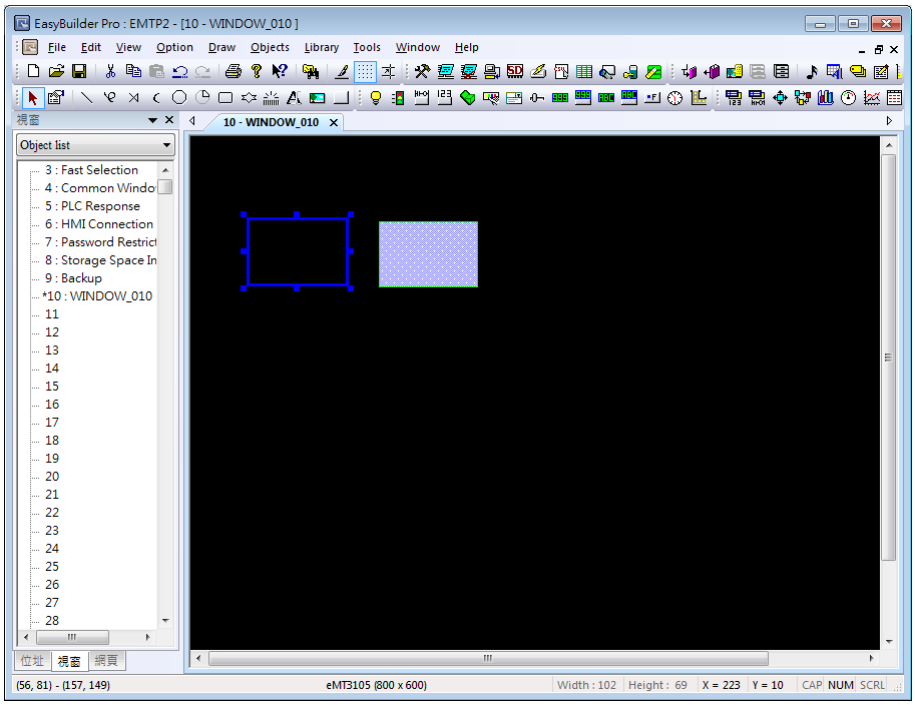
1. Click [New library] and enter the name of the new Shape Library, for example, “new_lib”.



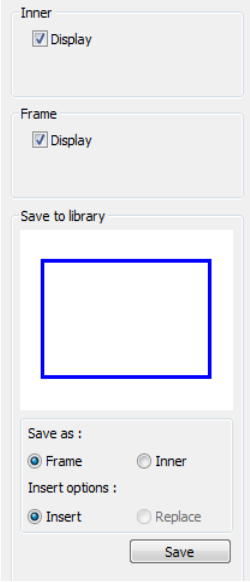
2. Click [Open], a popup dialog appears; click [Yes] to create the file.
3. A new Shape Library [new_lib] is added in [Shape manager], and its directory is shown below the name. This library is empty now as shown in the following figure.



4. Add a state to the selected Shape. First, use the drawing tools to draw a frame and inner in the window and select the frame to add to the Shape Library.



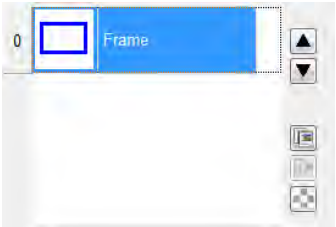
5. Click [Project] » [Shape] » [Save to shape] button in the toolbar, select [new_lib], and select a number in this library. The selected number is highlighted yellow.
6. Save the Shape as [Frame], select [Insert], and click [Save].



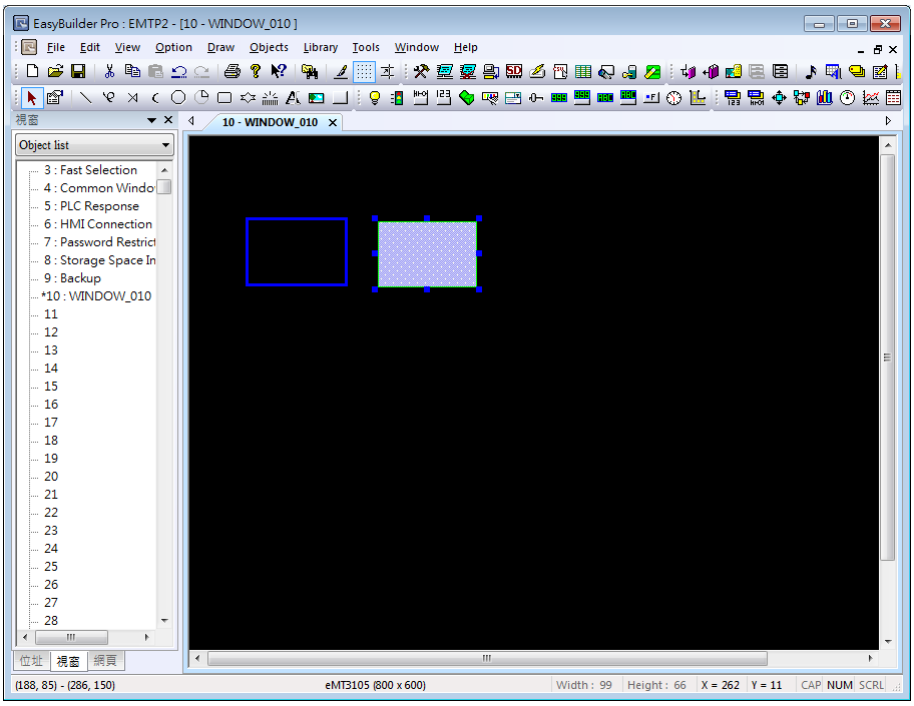
Setting	Description
Inner	Displays the inner of the Shape.
Frame	Displays the frame of the Shape.
Save to library	**Save as Frame** Saves the Shape as a frame. **Save as Inner** Saves the Shape as inner.

	Insert Inserts the Shape to be a new state. **Replace** Replaces a state with this Shape.
Save	Saves the settings above.

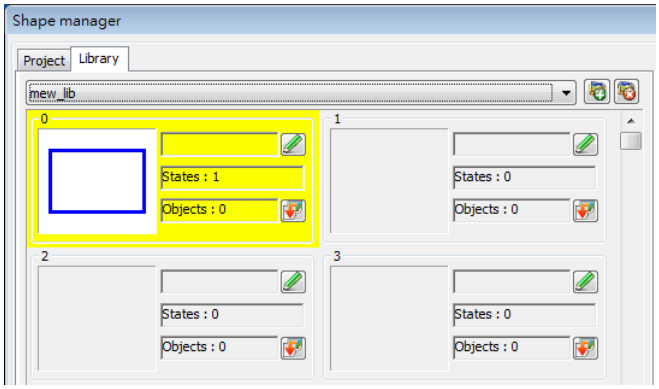
7. The following shows that a state of the Shape is added, and is defined as a frame.



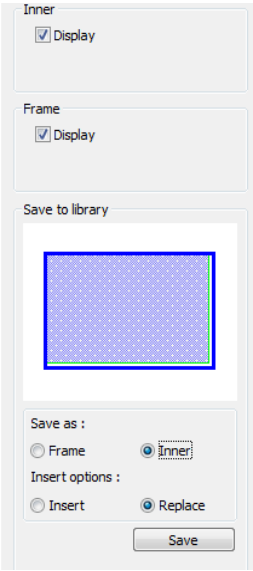
8. Create the shape to be saved as inner. Select the shape drawn in the window.



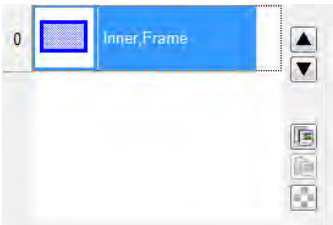
9. Click [Save to Shape Library] button in the toolbar, select [new_lib], and select the same number as in creating the frame in this library. The selected number is highlighted yellow.



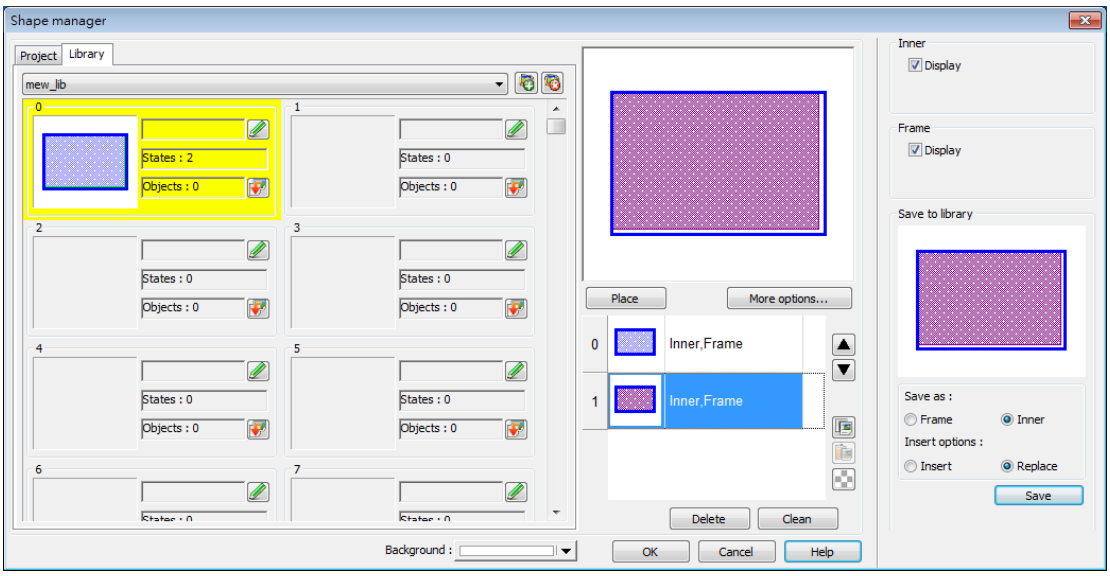
10. Save the Shape as [Inner], select [Replace], and click [Save].



11. A state of a Shape can include [Inner], [Frame], or both. The state 0 of the Shape shown in the following figure includes both frame and inner. Click [OK], the state 0 of the Shape is created.



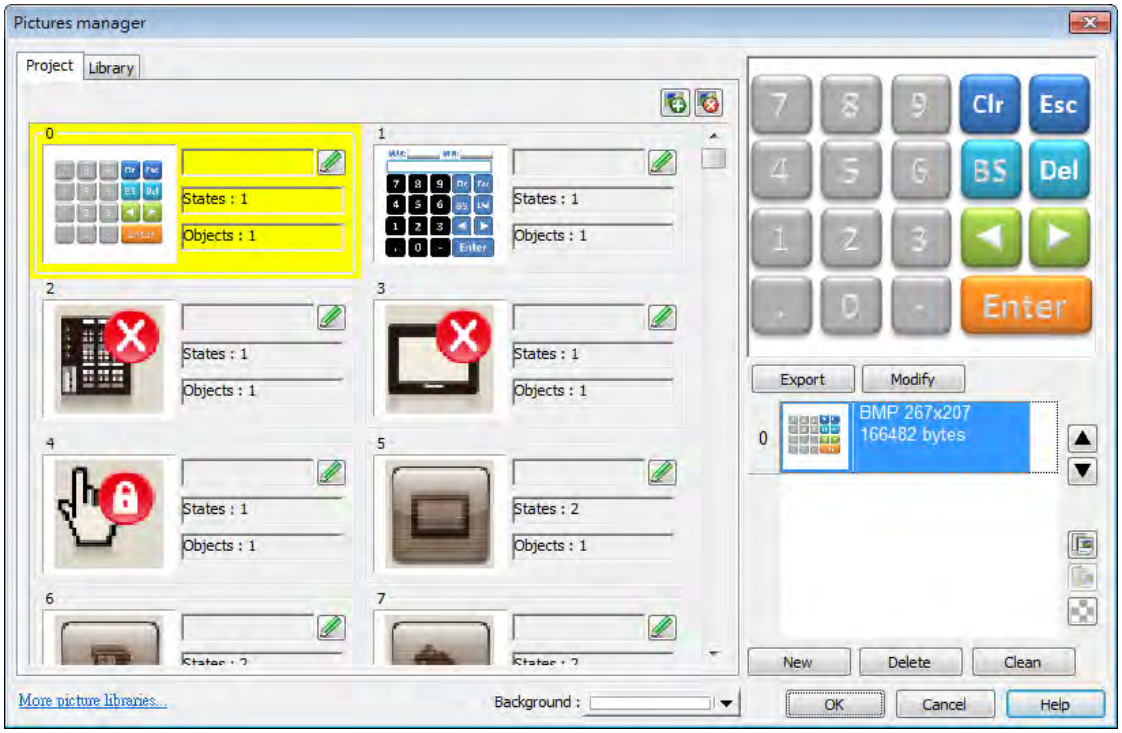
12. Follow the steps of creating state 0 and insert a new state set to state 1 as shown in the following figure. The Shape now has two states, click [OK] to finish setting.



14.3. Building Picture Library

14.3.1. Picture manager

Click [Project] » [Picture] button in the toolbar and the [Picture manager] dialog box appears.



Setting	Description
Project	The Picture edited here will be saved in .emtp. Up to 1000 Pictures can be added.
Library	The Picture edited here will be saved to the library directory on PC and will not be saved to .emtp project file.
New library	Add the existing .flbx and .flb picture library files. To add a new library that does not exist, enter a new file name and click [Open], an empty library file is created. Up to 40 library files can be added.
Unattach library	Delete the currently selected library.

Browse libraries	Search for path and view the pictures in the path.
Copy to project	Copy the Picture to [Project].
Background	Select the background color of the Picture. The color is only displayed in [Picture manager] dialog box, and is not displayed when placing the object in the screen.
More picture libraries	Log in to Weintek Official Website to download more libraries.
Export	Export the selected Picture.
Modify	Modify the settings of the selected Picture.
	Move the Picture to the previous / next state.
Copy	Copy the selected Picture.
Paste	Paste the copied Picture. The Picture copied to the clipboard can be imported to the library by pasting.
Insert transparent state	Insert a blank state after the selected state.
New	Add a new Picture.
Delete	Delete the selected Picture.
Clean	Delete all the Pictures listed here.
OK	Confirm to save the edited Shape.
Cancel	Cancel the editing event.
Help	Open help files.

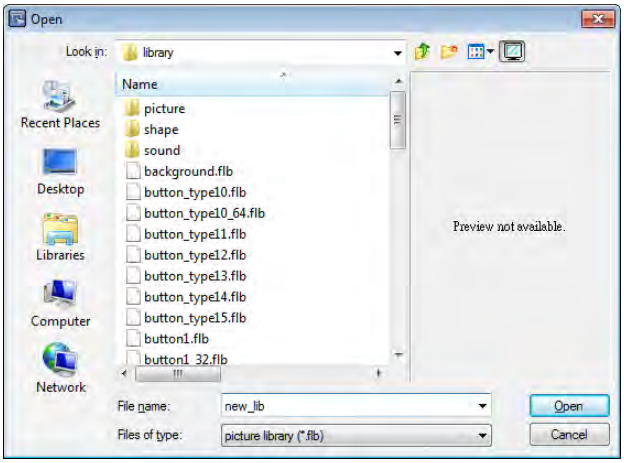
Note

- The supported picture formats are .bmp, .jpg, .gif, .dpd, .svg and .png. When adding a gif animation file in Picture Library, the loop times of this animated Picture can be set.

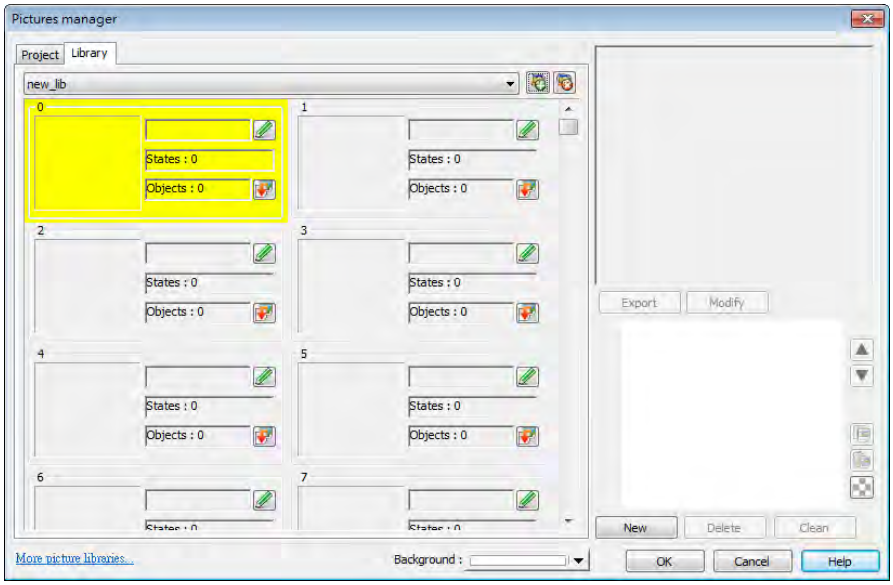
14.3.2. Steps to Build Picture Library

The following example explains how to create a new Picture Library and add a Picture with two states into the library.

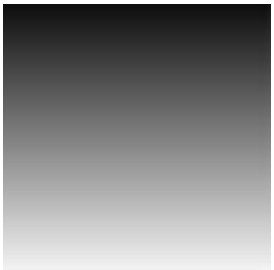
1. Click [New library] and enter the name of the new Picture Library.



2. Click [Open], a popup dialog appears; click [Yes] to create the file.
3. A new Picture Library [new_lib] is added in [Picture manager]. This library is empty as shown in the following figure.

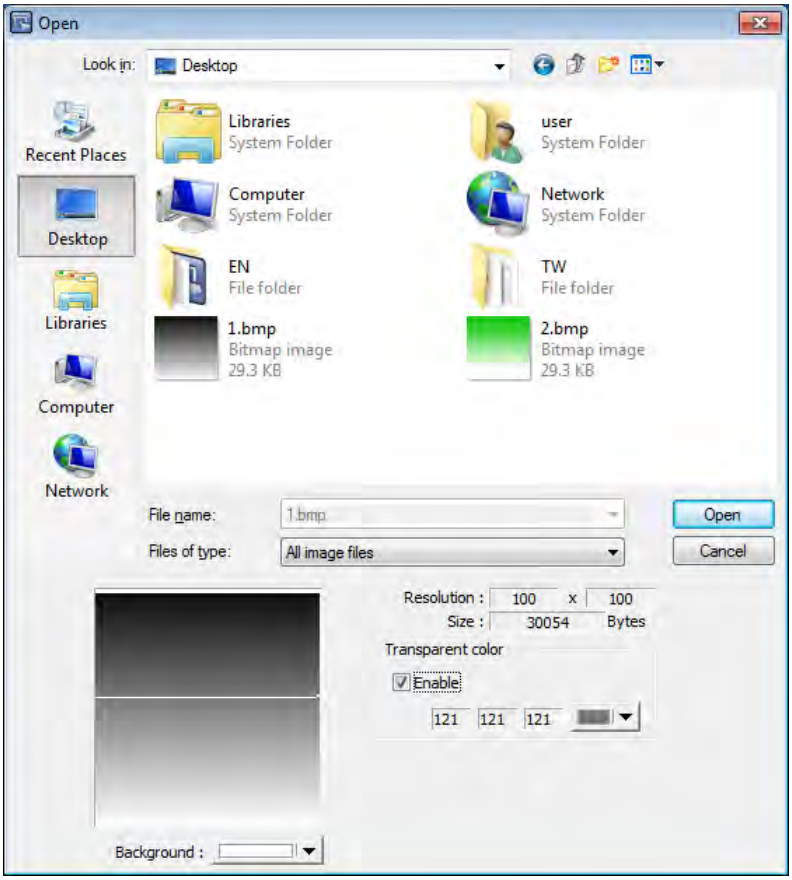


4. Draw the two pictures below to represent state 0 and state 1 respectively.

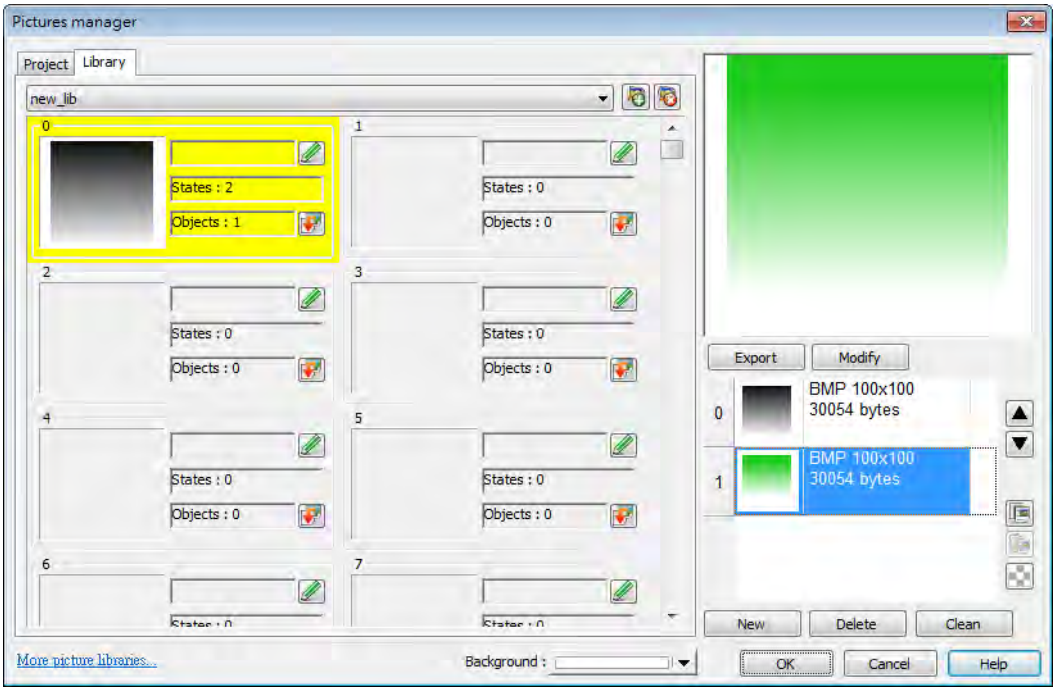


5. Select [new_lib], and select a number in this library. The selected number is highlighted yellow.

6. Click [New], and select the picture for state 0.
7. When the following dialog box is shown, select [Enable] check box to use transparent color. Set to RGB (121, 121, 121), the corresponding color in the picture below will be transparent. Or, click on a desired area with mouse to be the transparent area, the system will show the RGB of the clicked area automatically.



8. The Picture of state 0 is created. Follow the steps of creating state 0 to create state 1 by clicking [New] as shown in the following figure.



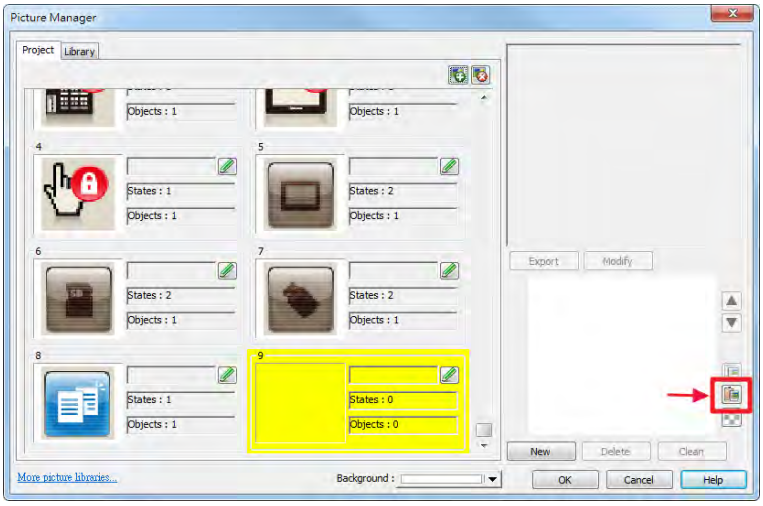
9. When finished, a complete Picture is created, click [OK]. In [Picture manager] dialog box it shows that the newly added Picture Number 0 is a bitmap picture with two states.

14.3.3. Steps to Import Picture by Pasting

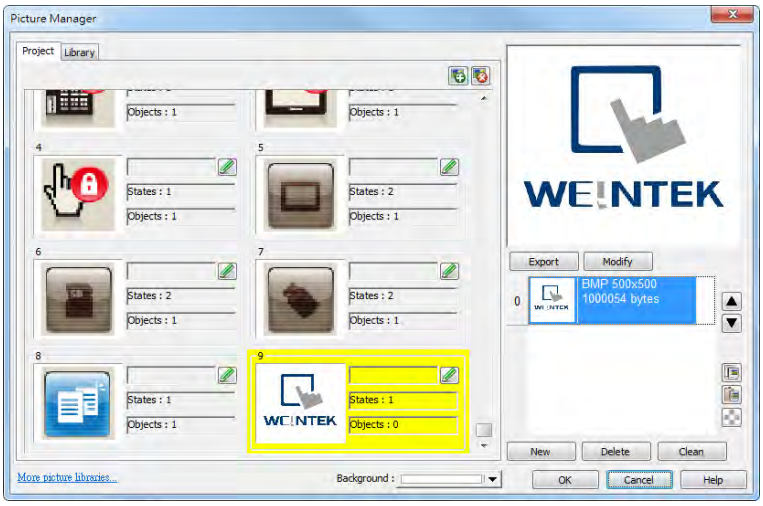
The following example explains how to import a Picture into the library by pasting the picture from the clipboard.

1. Copy the following picture to the clipboard.

2. Click the Paste icon on the right side.



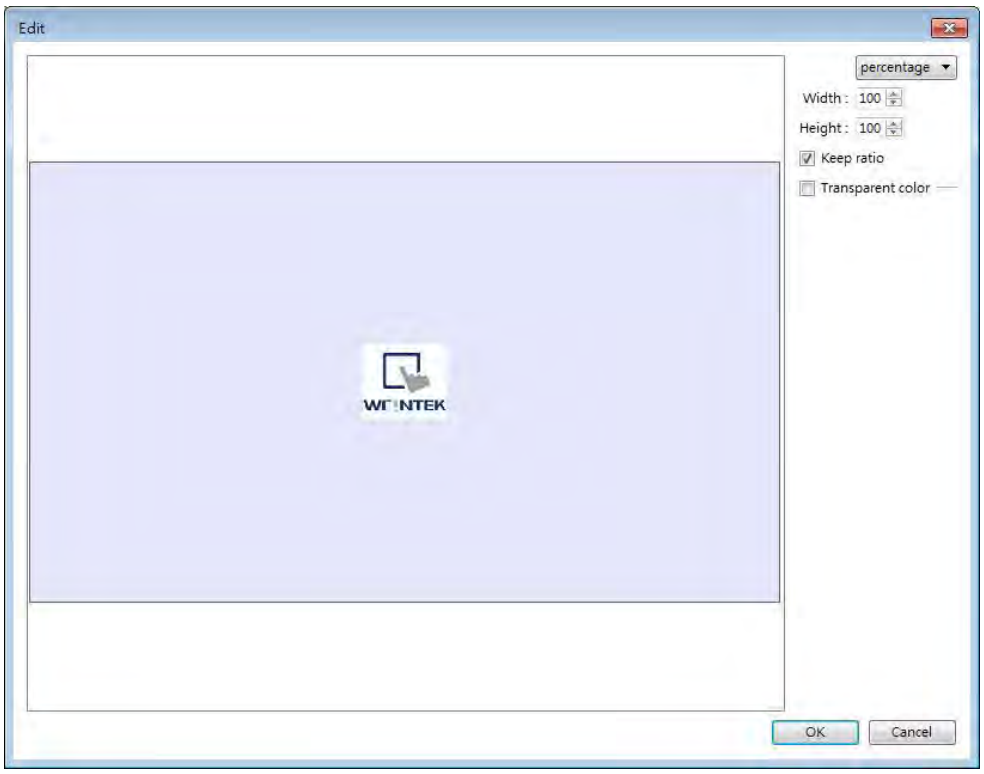
3. The Pictur can be easily imported to the library.



Pasting a picture directly from the clipboard into the project editing window is possible.

1. If the picture below is already in the clipboard.

2. Pressing Ctrl+V key in the editing window will open the following dialog box.



3. The picture will be pasted into the editing window after clicking OK.
4. Double clicking the picture and open Picture Library, the picture can be found in the

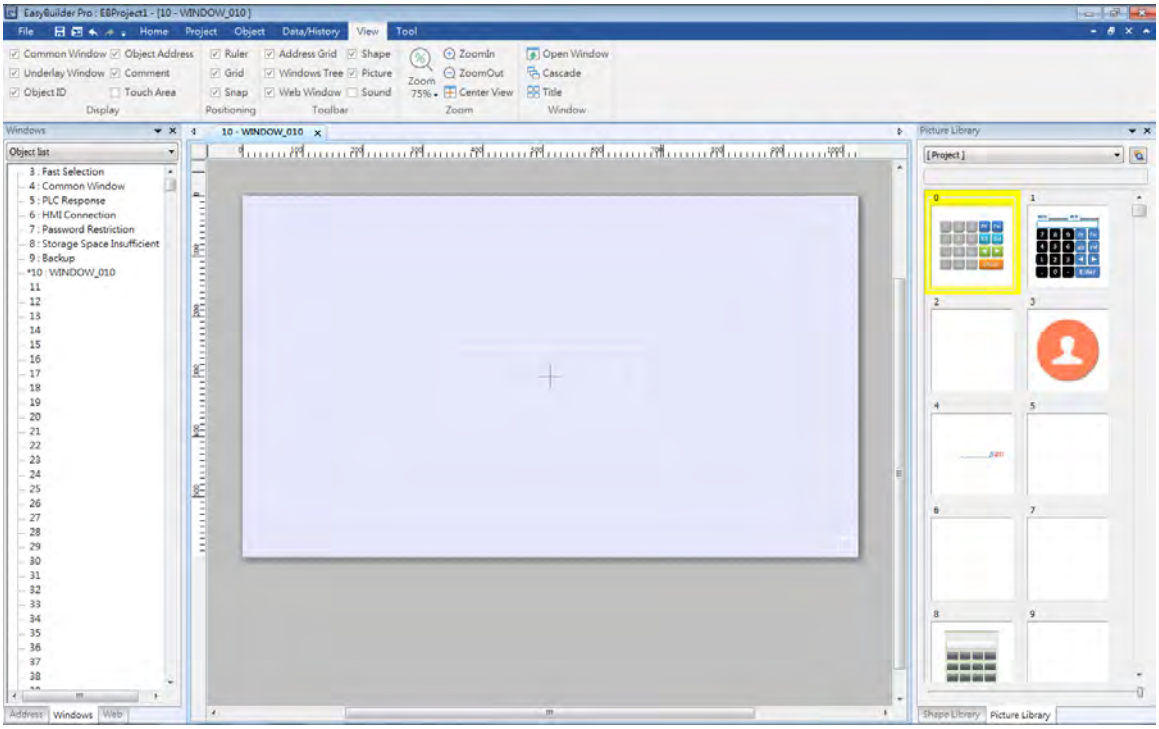
Project tab in Picture Manager.

Note

- Transparent color can only be set for .bmp, .dpd, and .jpg picture files.

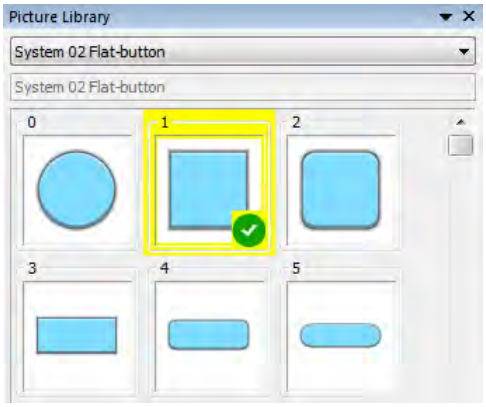
14.4. Immediate Modification and Preview

Shape Library / Picture Library settings windows are opened in the editing window. The user can immediately change the picture used by an object or multiple objects, and certain libraries support immediate color change.



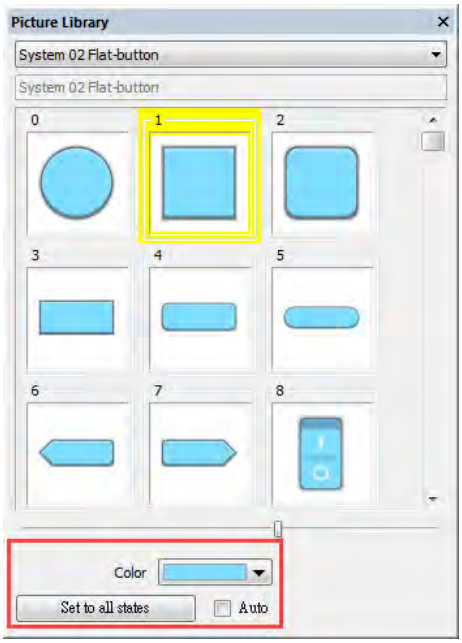
14.4.1. Replacing a Picture

1. Click the object or objects whose picture is to be replaced.
2. Locate a new picture in Picture Library window and click the lower-right corner of that picture.

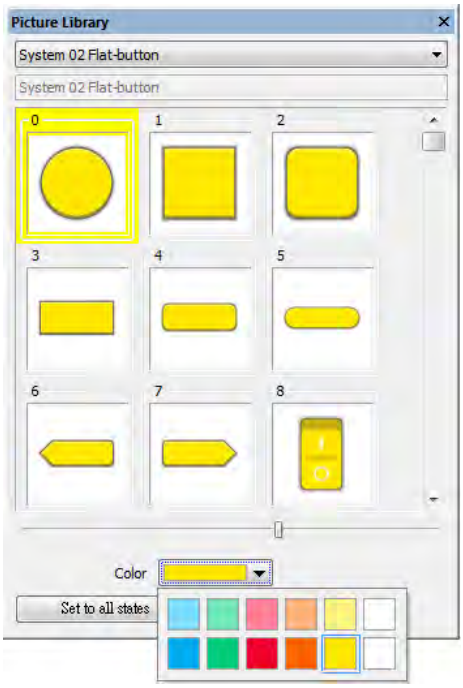


14.4.2. Replacing a Color

The [Color] drop-down list appears when the selected library supports immediate color change.



1. Select an object or objects whose color is to be replaced.
2. Select a color from the Color drop-down list.



3. Click the lower-right corner of the new picture to apply.

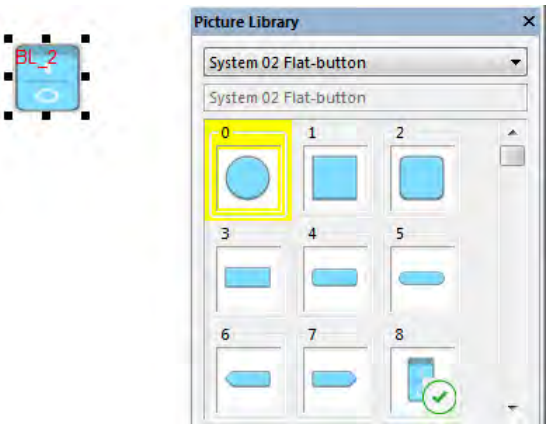
Setting	Description
Set to all states	Apply the selected color to all states of the object that use the same picture.
Auto	Automatically execute [Set to all states].

14.4.3. Extended Features

- In the settings window, dragging the slider leftward or rightward can make the thumbnail of the pictures smaller or bigger.

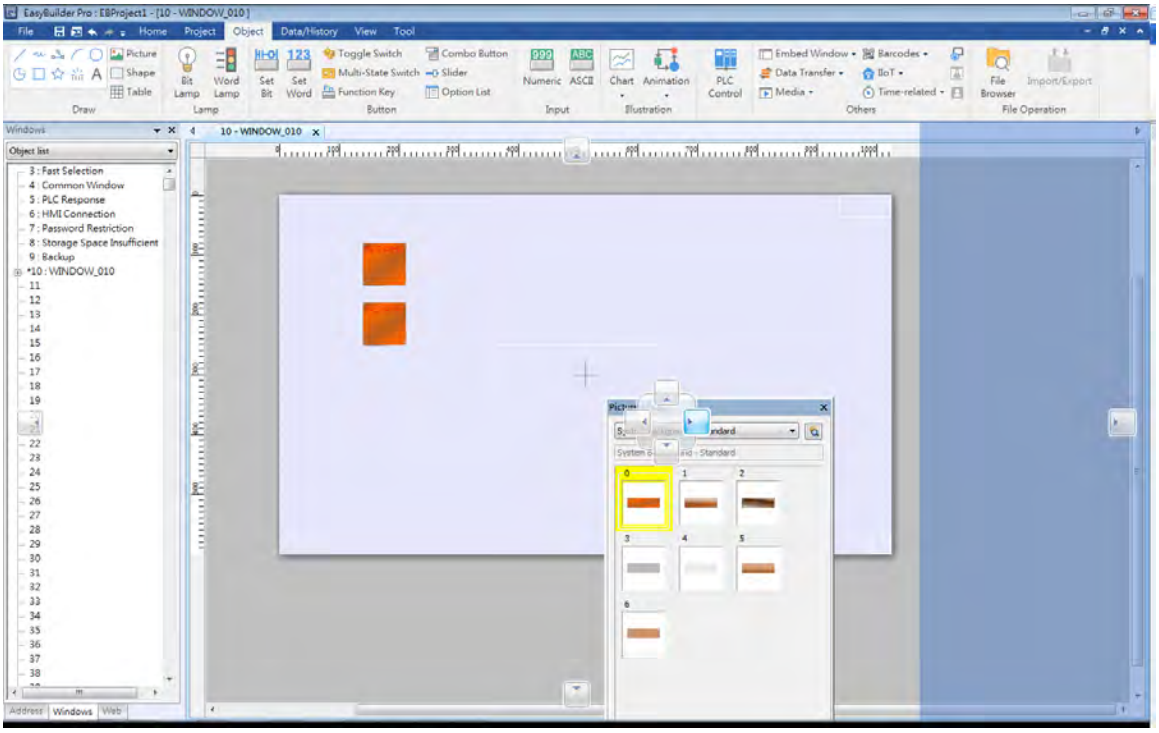
- Immediate preview can be achieved by:

1. Select an object.
2. Press and hold the Shift key on the keyboard.
3. Select a picture in the library to preview, and release the Shift key to return to current picture.



14.4.4. Docking a Window

Users can drag the Shape / Picture Library window to a desired destination position. When dragging the window, a semi-transparent outline of the window shows where the window will be docked when releasing the mouse cursor at that point.



15. Label Tag Library and Multi-Language

This chapter explains how to build and use Label Tag Library.

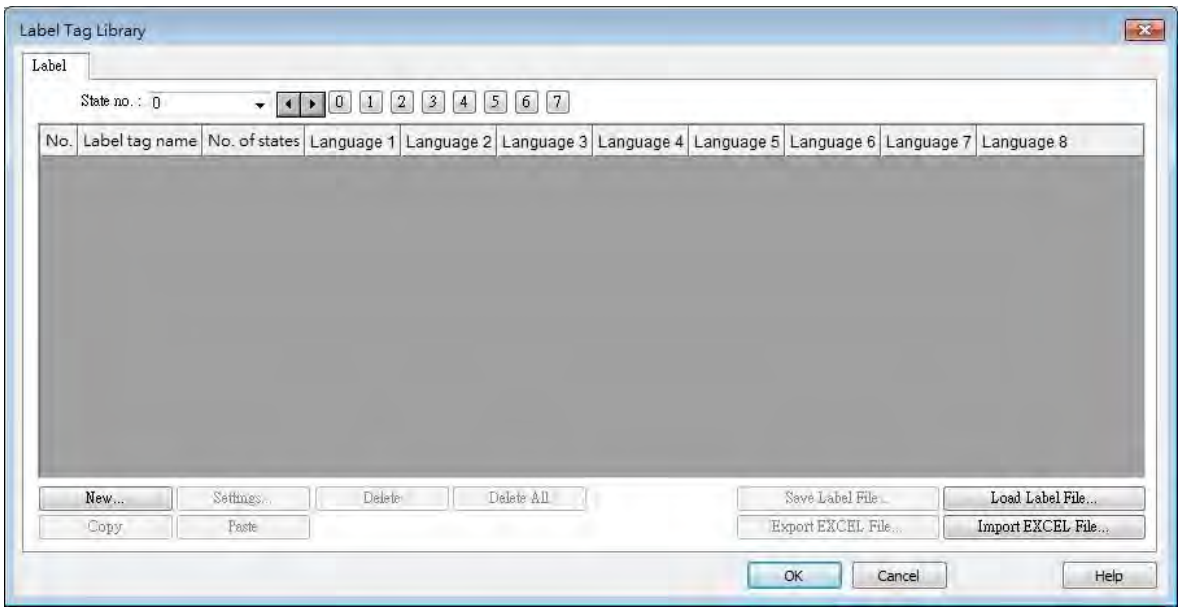
15.1. Overview 15-2
15.2. Label Tag Library Manager 15-2
15.3. Steps to create Label Tag Library 15-3
15.4. Using Label Tag Library 15-4
15.5. Selecting Language & Font 15-5
15.6. Settings of Multi-Language 15-9

15.1. Overview

The Label Tag Library feature enables a multi-language environment. When multiple languages are required, users can create the Label Tag Library and then select a suitable label in the project. The project will display the corresponding language in runtime based on the settings. EasyBuilder Pro supports up to 24 different languages simultaneously. This chapter will explain how to create and use the Label Tag Library.

15.2. Label Tag Library Manager

Click [Project] » [Label] on the toolbar and the [Label Tag Library] dialog box appears.



Setting	Description
State no.	Indicates the current state. Each Label has a maximum of 256 states (state no. 0 ~ 255). The state no. is determined by [Language no.]. If less than 3 languages are used, the maximum state no. is 256. If more than 4 languages are used, divide 768 by the language number to get the maximum state no.. For example, the number of languages is 24, then there are only 768/24 = 32 states.
New	Adds a new Label.
Settings	Sets the selected Label.
Save Label File	Saves all Labels in .lbl format.
Load Label File	Loads the existing .lbl file to the Label Library.
Export EXCEL File	Saves all Labels in .csv, .xls, or .xlsx format.

Import EXCEL file	Loads the existing .csv, .xls, or .xlsx file to the Label Library.

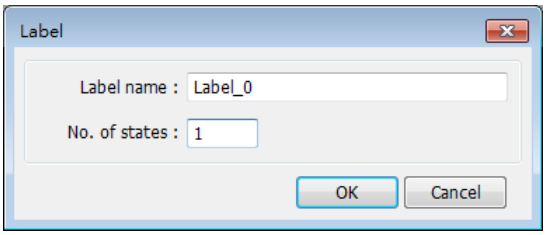
Note

- Unicode is not supported when importing and exporting an Excel file.

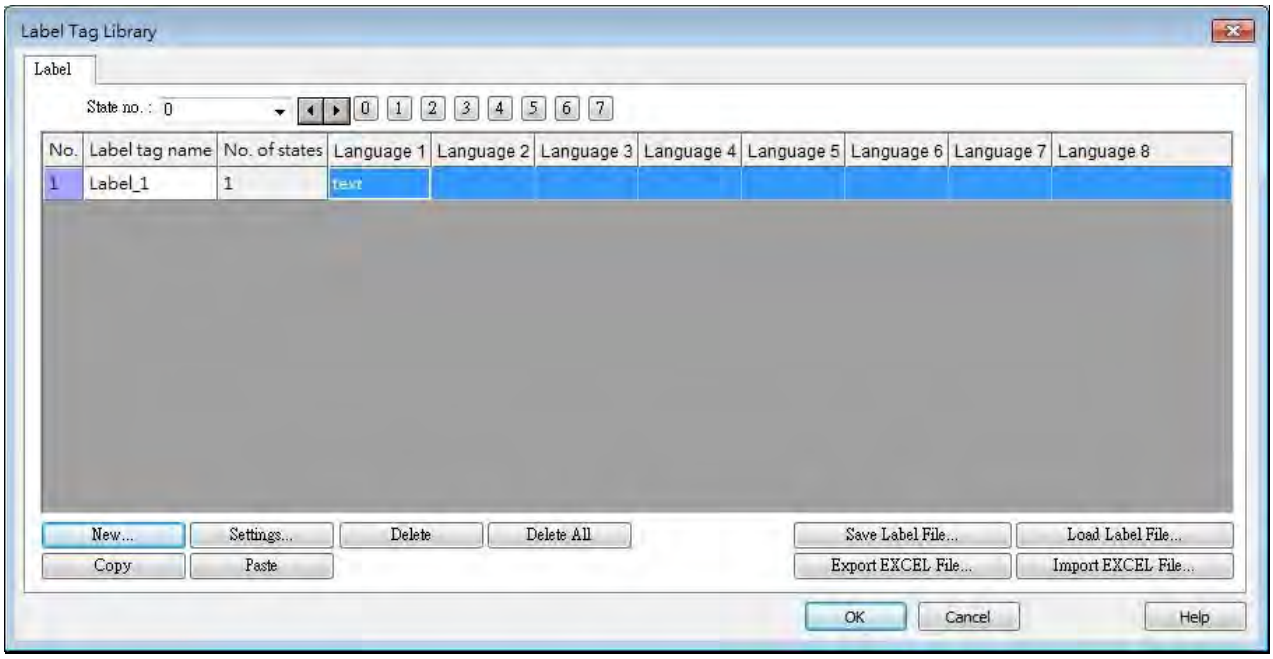
15.3. Steps to create Label Tag Library

Please follow the steps to create a Label Tag Library.

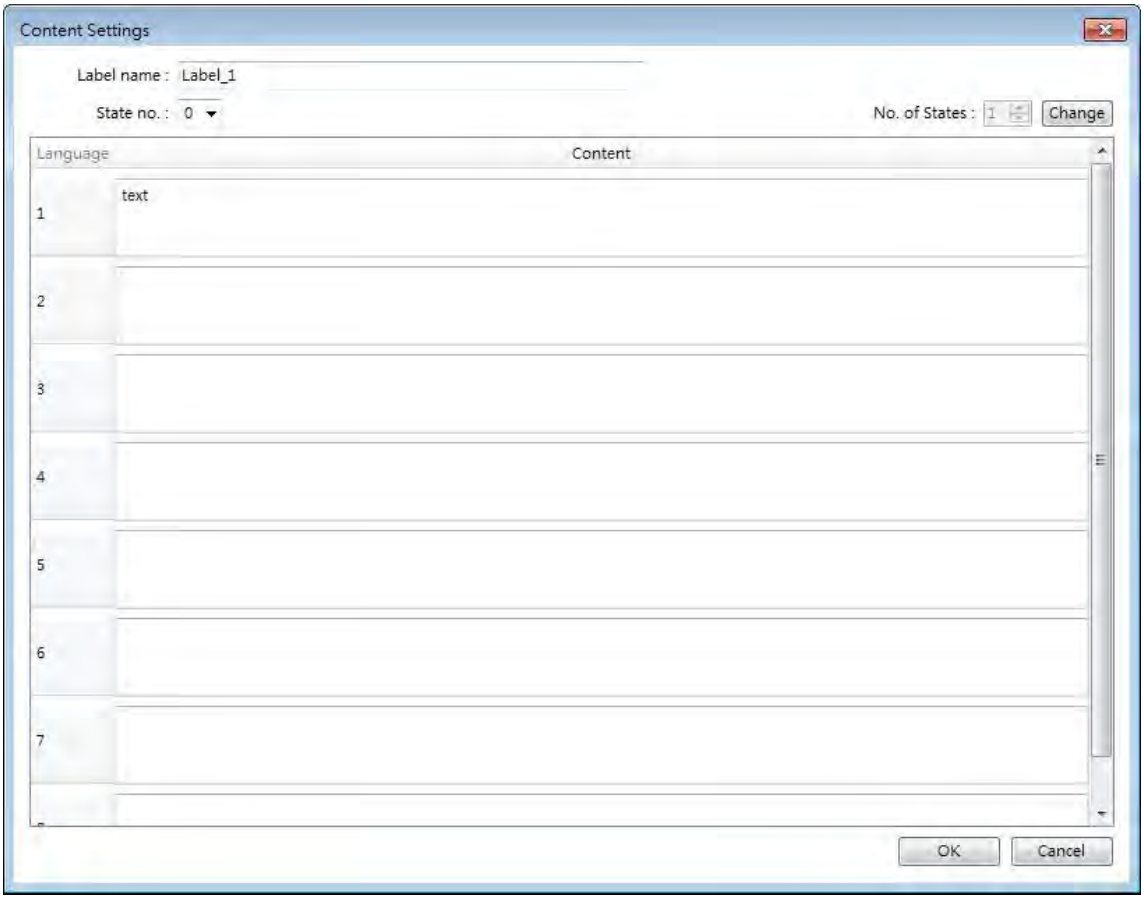
1. From the Library menu, click [Label]. The Label Tag Library dialog box appears. Click [New] to specify the name of the Label and the number of states to be displayed by this Label.



2. Click [OK] and a new label is added to the Label Tag Library. Select the label and click [Settings] to edit its content.

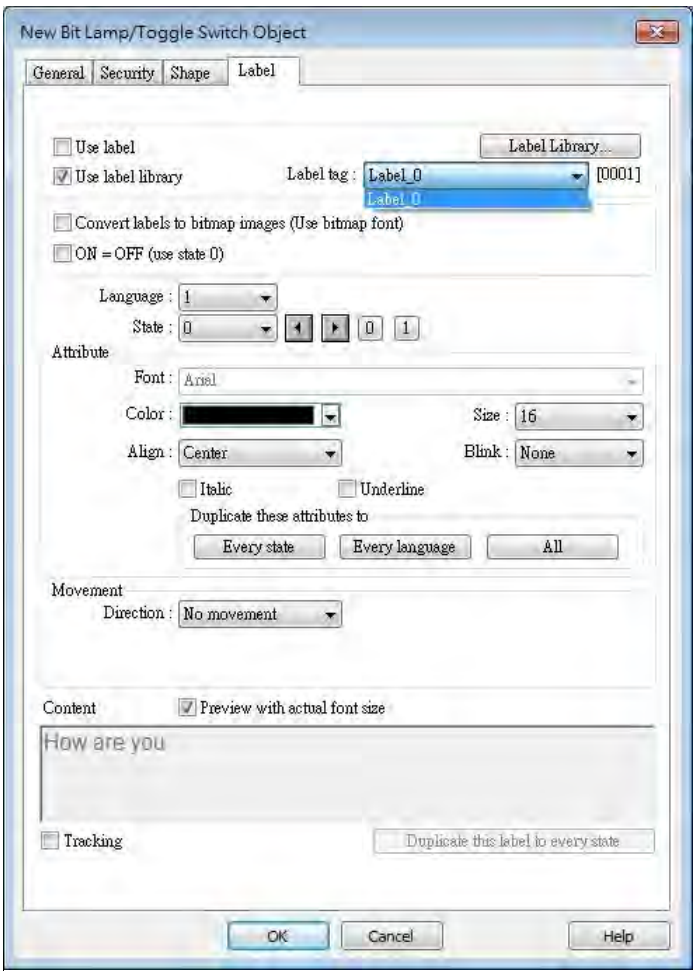


3. Edit the corresponding language content.



15.4. Using Label Tag Library

When there are defined labels in the Label Library, the labels can be found in the object's [Label] tab. Select [Use label library] check box, and select the label from the pull-down list [Label tag].

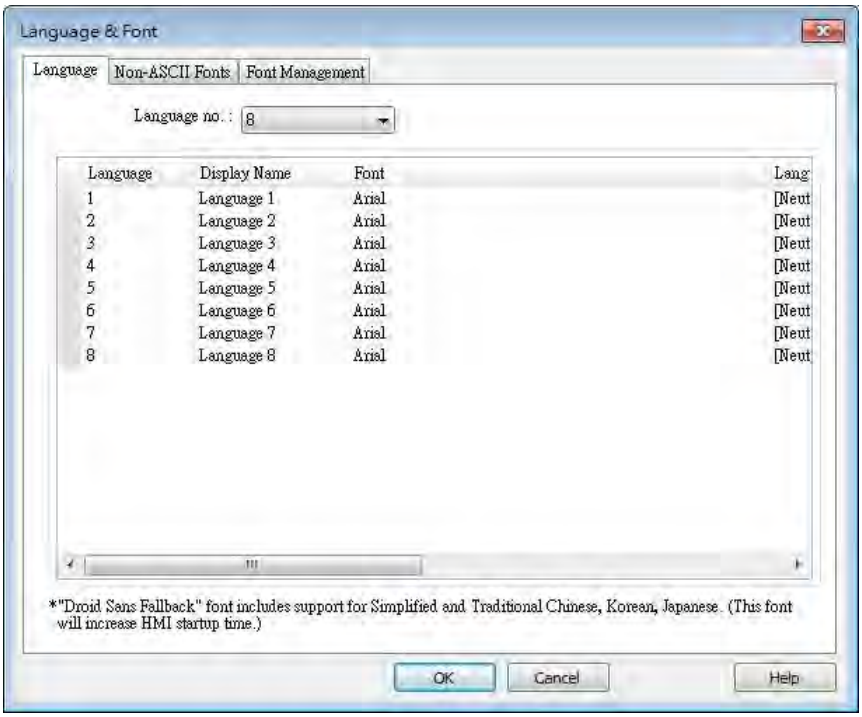


When a tag is selected, the content of the selected tag is shown in the [Content] field in its corresponding font style. Please note that from Language 2 to Language 24 can only be set the Font [Size], the others such as [Color], [Align], [Blink], etc. will follow the settings of Language 1.

15.5. Selecting Language & Font

15.5.1. Language

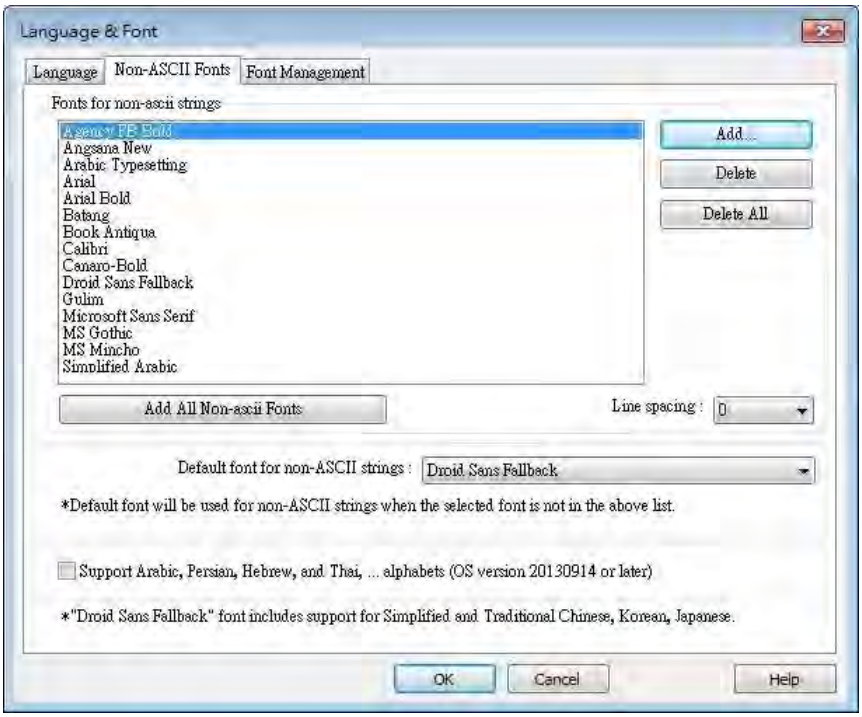
Click [Project] » [Language & Font] on the toolbar and the [Language & Font] dialog box appears.



Setting	Description
Language no.	Specifies the number of languages used in a project.
Font	Shows language names and the fonts used for each language . A font description can be entered. [Language Code] allows changing the language used for push notification about the events in EasyAccess 2.0, when [Push notification (EasyAccess 2.0)] is enabled in Event Log. Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

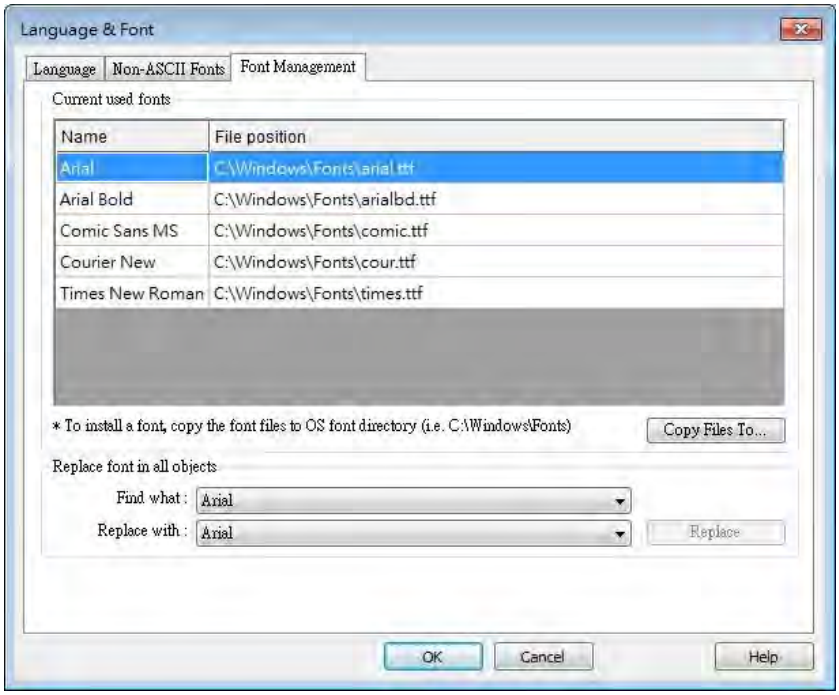
15.5.2. Non-ASCII Fonts

Parameters in [Non-ASCII Fonts] tab determine the non-ASCII fonts. The non-ASCII fonts used are listed here. When using non-ascii characters or double byte characters (including Simplified or Traditional Chinese, Japanese, or Korean) which are not listed in [Fonts for non-ascii strings] table, EasyBuilder Pro will select a font from the list to substitute for it automatically.



Setting	Description
Add All Non-ascii Fonts	The non-ascii fonts in Windows can be added to the [Fonts for non-ascii strings] table. **Line spacing** Set the space between multiple text lines.
Default font for non-ascii strings	EasyBuilder will use the selected font as default font for non-ascii strings.
Support Arabic, Persian, Hebrew, and Thai,… alphabets	Select this check box to correctly display these alphabets.

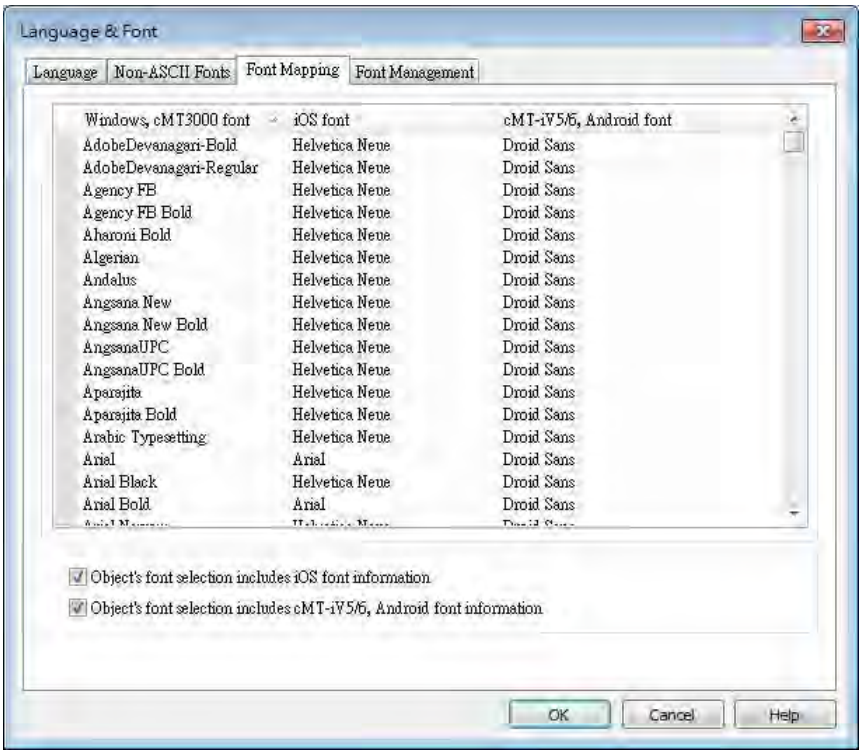
15.5.3. Font Management



Setting	Description
Current used fonts	This table shows all the fonts used in this project and the file position of its font file.
Copy Files To	Copy the font file to the specified position.
Replace font in all objects	Change font at a time for all the objects using that font.

15.5.4. Font Mapping

This tab lists the fonts used in Windows / cMT3000 and the mapping fonts in iOS / cMT-iV5 /cMT-iV6 / Android devices.



Setting	Description
Object's font selection includes iOS / cMT-iV5 / cMT-iV6 /Android font information	When the checkbox is selected, the fonts in iOS / cMT-iV5 / cMT-iV6 / Android that map to the select font in object property settings will be shown.

15.6. Settings of Multi-Language

When displaying the texts in multiple languages, the system register "[LW-9134]: language mode" should be used too.

The value of [LW-9134: language mode] is ranged from 0 to 23. Different values correspond to different languages.

If not all languages are selected to compile and download, [LW-9134] will work differently.

For example, user defines 5 different languages in the Label Tag Library:

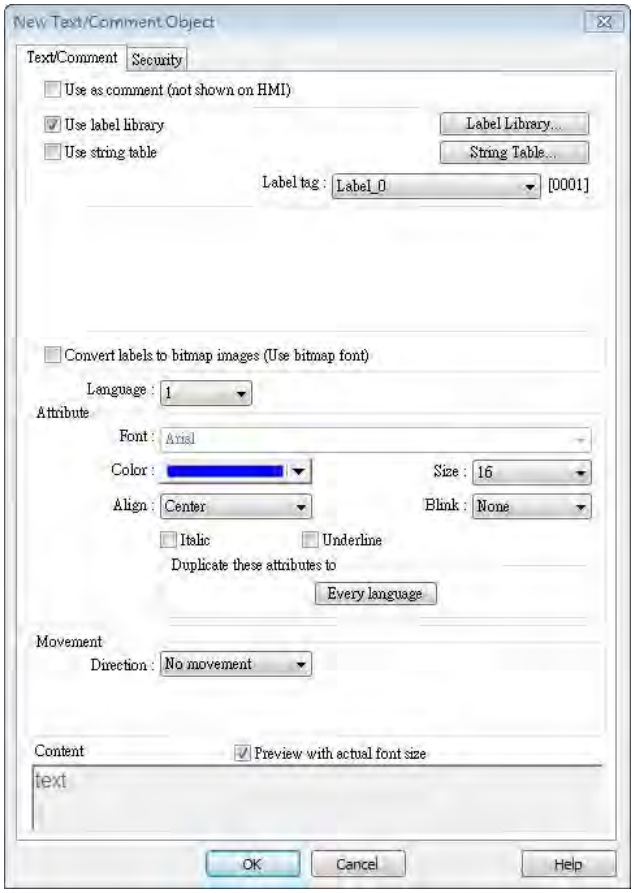
1: English, 2: Traditional Chinese, 3: Simplified Chinese, 4: French, 5: Korean

If only Language 1, Language 3, and Language 5 are selected to compile then the corresponding values of [LW-9134] are:

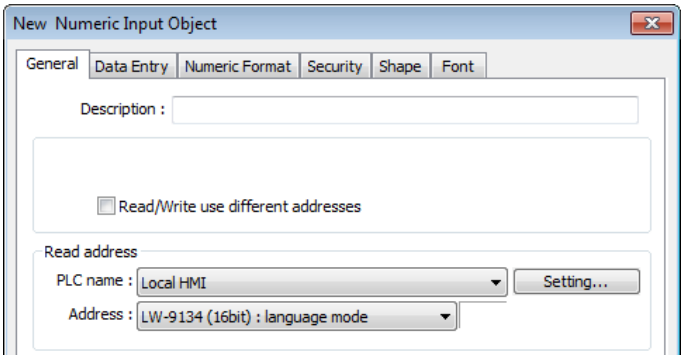
0: English, 1: Simplified Chinese, 2: Korean

Please follow the steps to use multiple languages.

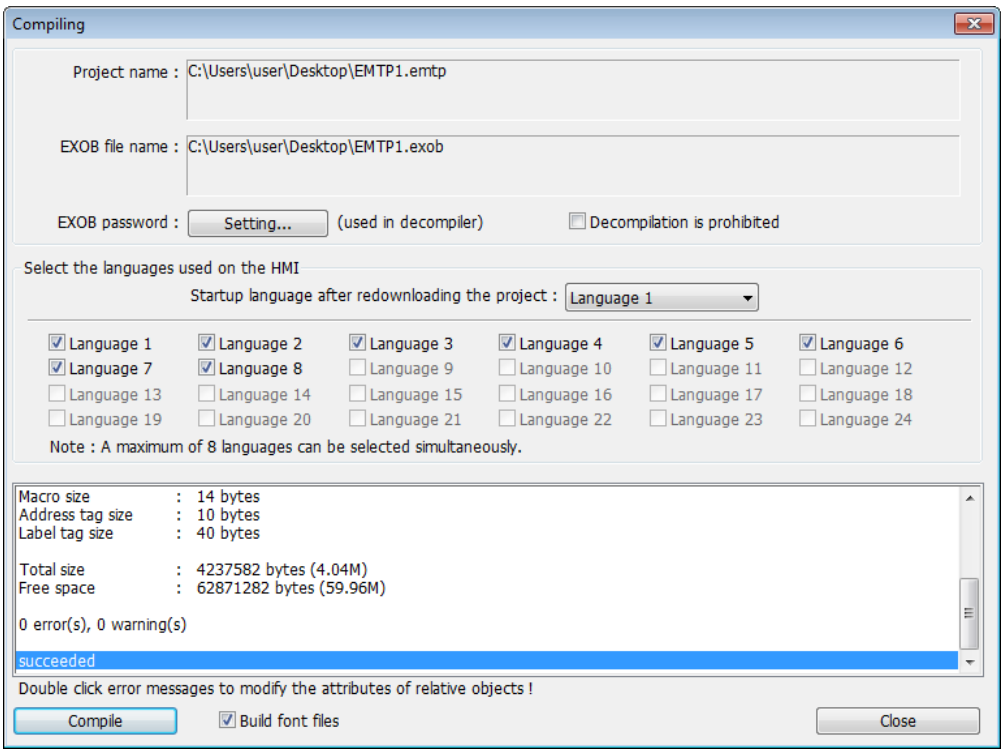
1. Create a Text/Comment object and select [Use label library] checkbox.



2. Create a Numeric Input Object and use the system register [LW-9134].



3. When compiling, select the defined languages.



4. The simulation is shown as followed: If the value of [LW-9134] is changed, the content of the Text object will be changed.

English

LW9134 : language mode 0

简体中文(SIMPLE)

LW9134 : language mode 2

한국어 웹(KOREAN)

LW9134 : language mode 4

Note

- When using a cMT / cMT X Series model, system register [LW-9134] is used to change the language mode in server, while [PLW-9134] is used to change the language mode on a portable tablet device.

Click the icon to download the demo project that illustrates how to use the Option List object to switch between multiple languages. Please confirm your internet connection before downloading the demo project.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

16. Address Tag Library

This chapter explains how to build and use Address Tag Library.

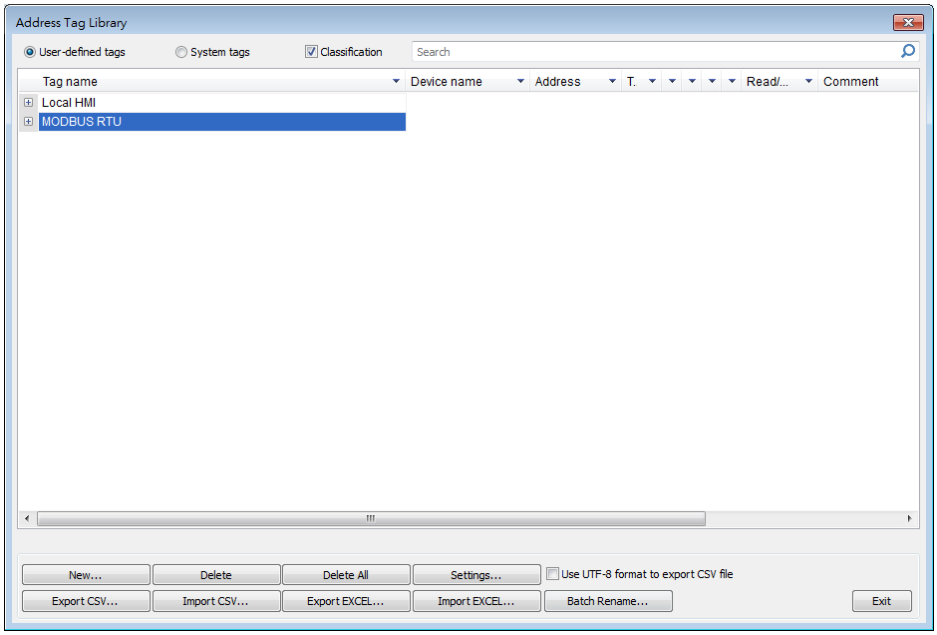
16.1. Overview 16-2
16.2. Building Address Tag Library 16-2
16.3. Using Address Tag Library 16-5

16.1. Overview

Generally it is recommended to define the commonly used addresses in Address Tag Library when starting to build a project. It not only avoids accidental reuse of addresses but also improves project readability.

16.2. Building Address Tag Library

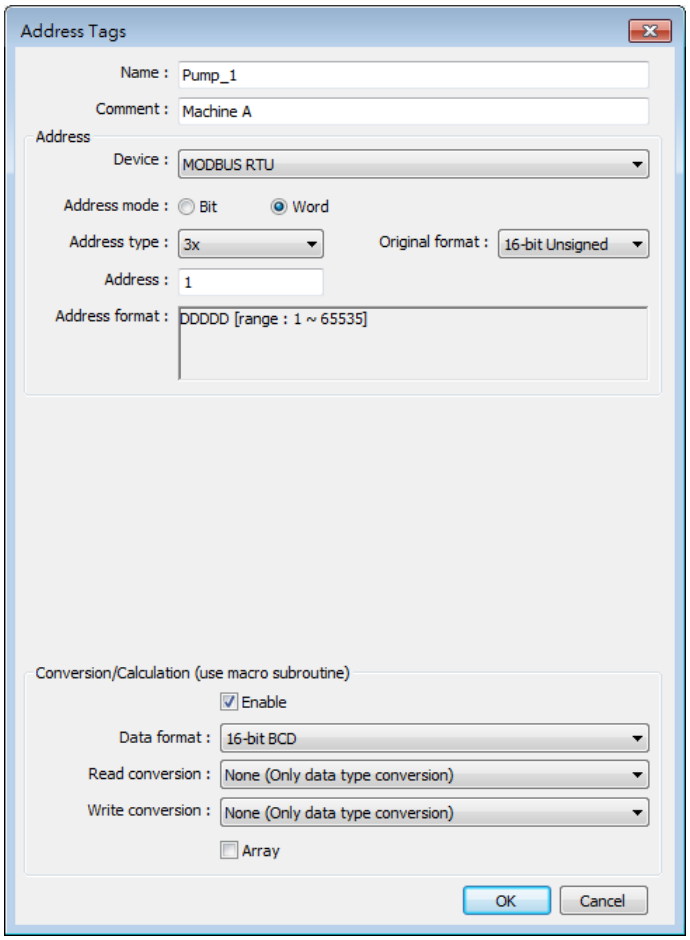
Click [Project] » [Address] on the toolbar and the [Address Tag Library] dialog box appears.



Setting	Description
User-defined tags	Displays user-defined address tags.
System tags	Displays system registers. The registers listed cannot be deleted or changed.
Classification	With this checkbox selected, the system tags can be classified into categories according to their function. Users may customize classification by editing the system_tag.xml file, found under EasyBuilder Pro installation directory.
Search	A key word may be entered in the provided field to search for a specific tag.
New	Adds a new address tag. Please see the steps next page.
Settings	Sets the selected address tag.

Delete All	Deletes all existing address tags.
Delete	Deletes the selected address tag.
Export CSV	Saves all current address tags as .csv file. Tag names of the tags used in the project file will be displayed in red in Excel.
Import CSV	Loads the existing .csv file of address tag to the current project.
Export EXCEL	Saves all current address tags as .xls file.
Import EXCEL	Loads the existing .xls file of address tag to the current project.
Use UTF-8 format to export CSV file	If selected, the .csv file will be exported in UTF-8 format. If not selected, in ANSI format.
Batch rename	Rename multiple address tags of a device at a time.

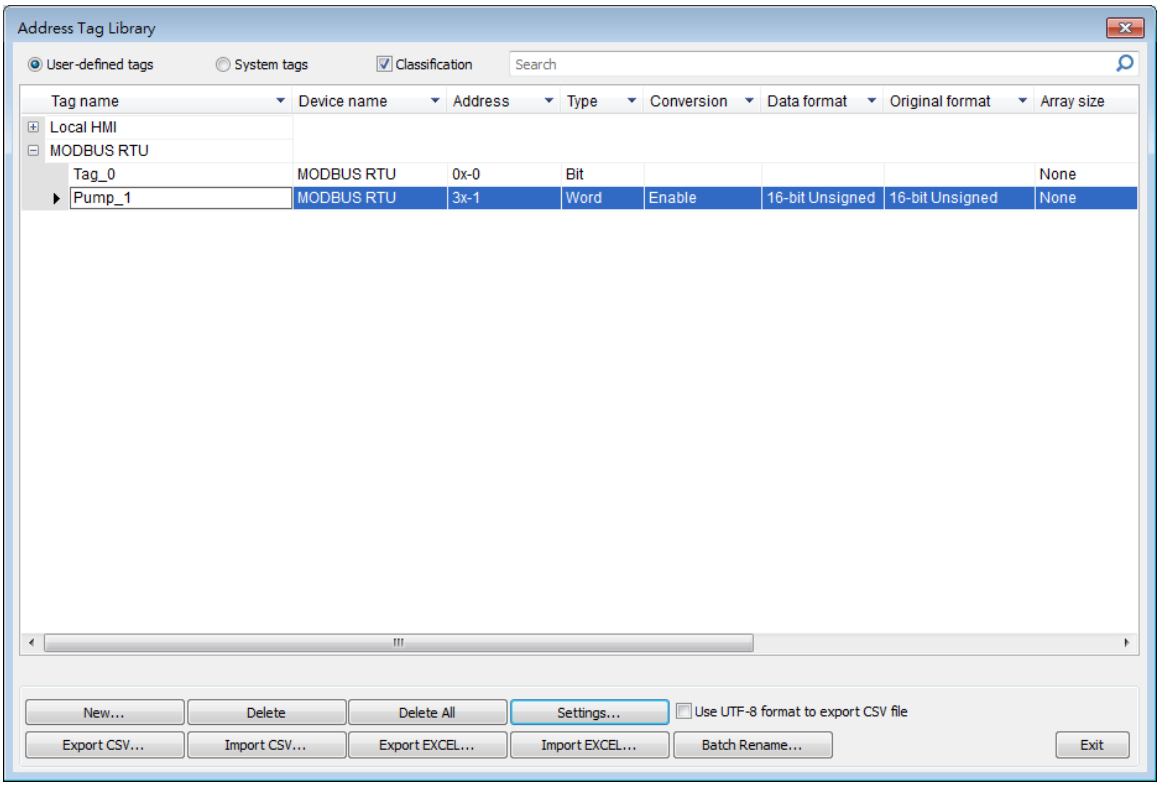
1. Click [New] and set the relevant properties.



Setting	Description
Name	The name of the address tag.
Comment	The information about the address tag.
Name	The name of the address tag.
Device	As defined in [System Parameter Settings] » [Device list].
Address mode	The tag address type; select [Bit] or [Word].
Address type	The available address types depend on [Device] and [Address mode].
Address	Address of the tag.
Original format	If select [Word] in [Address mode], the data format can be specified. When a tag with [Original format] specified is selected, the system will only use the specified format.
Conversion/Calculation (Use macro subroutine)	When enabled, the data format that the address tag will be converted into can be specified. Macro subroutines can be selected to do read/write conversion.
Read / Write	Select the macro subroutine to do read/write conversion.

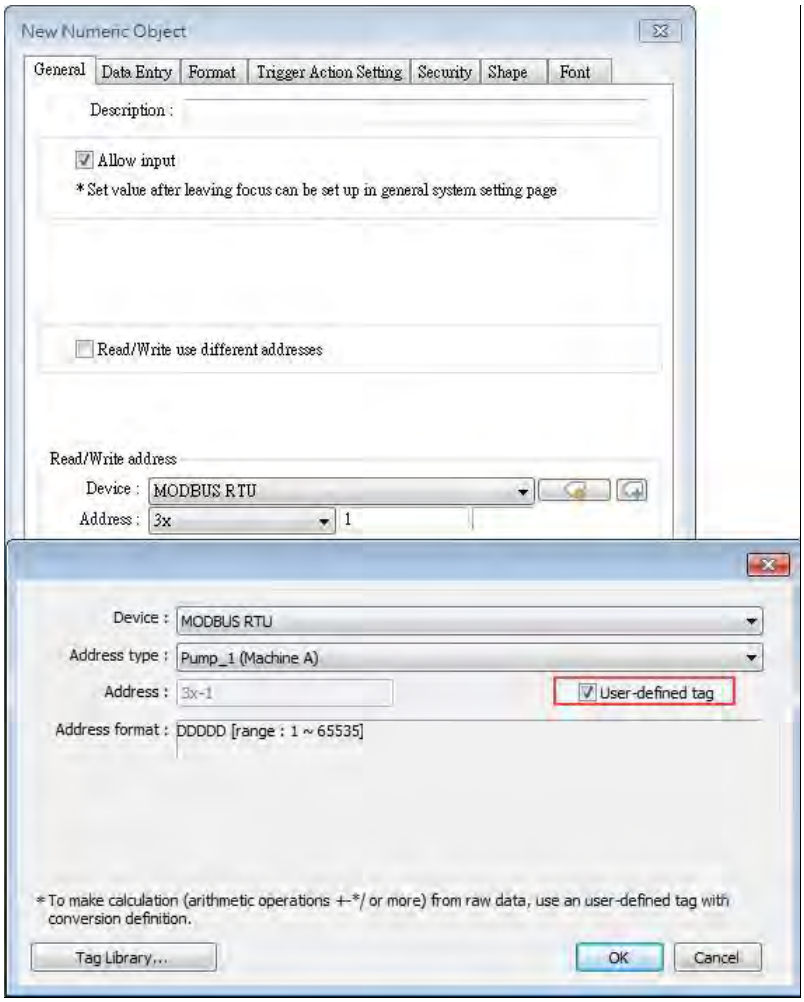
conversion	The macro subroutine can only be selected when the data format is identical to the one in the macro subroutine.

2. Click [OK], a newly added tag can be found in the [User-defined tags] library.

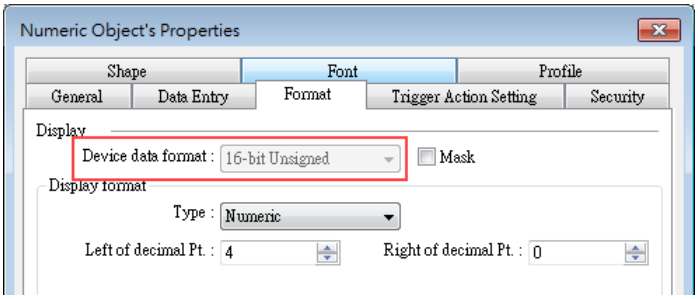


16.3. Using Address Tag Library

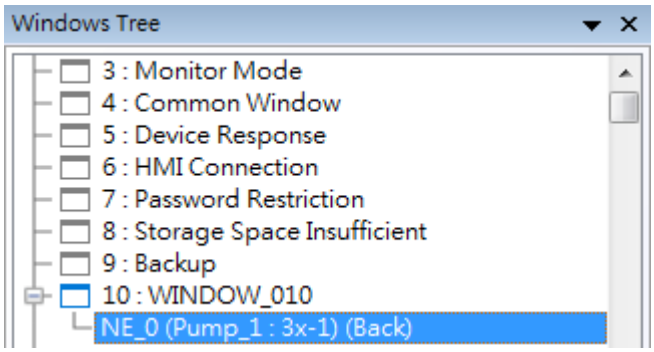
1. Create a tag in Address Tag Library.
2. Create an object, select [General] » [Device].
3. Click [Settings] to finish the settings.
4. Select [User-defined tag] check box.



5. In [Address type] select the defined tag.
6. If the data type is selected when creating the address tag, the system automatically restricts the data format to the one selected.

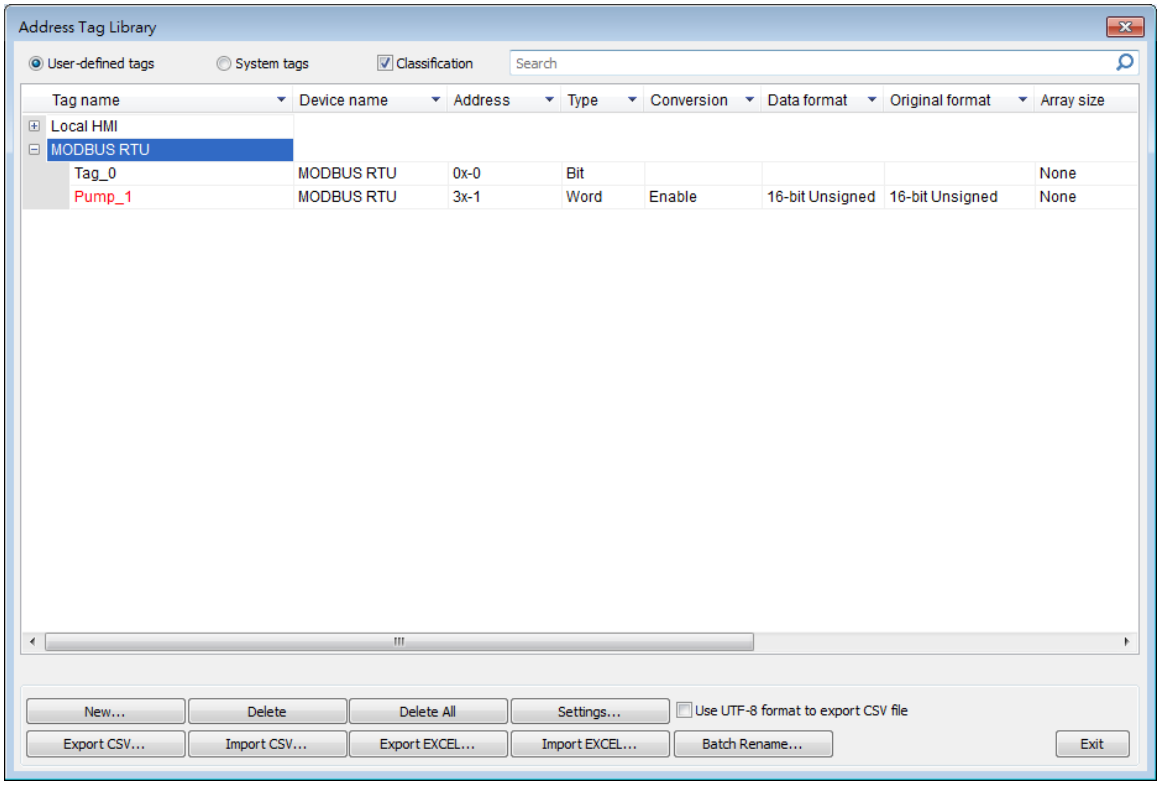


7. When finished, the window tree will show the address tag name used by the object.



Note

- The name of the used tags will be written in red font in Address Tag Library.



Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

17. Transferring Recipe Data

This chapter explains how to transfer recipe data.

17.1. Overview 17-2
17.2. Steps to Update Recipe Data with Ethernet or USB Cable 17-2
17.3. Steps to Update Recipe Data with SD Card or USB Disk 17-3
17.4. Transferring Recipe Data 17-4
17.5. Saving Recipe Data Automatically 17-5

17.1. Overview

Recipe Data refers to the data stored in RW and RW_A addresses. The way of reading and writing these addresses is the same as operating a word register. The difference is that recipe data is stored in flash memory, when restarting HMI, the latest data records in RW and RW_A are kept.

The size of recipe data a RW address can store is 512K words, and RW_A is 64K words. Users can update recipe data with SD card, USB disk, USB cable or Ethernet and use the data to update PLC data. Recipe Data can also be uploaded to PC; furthermore, PLC data can be saved in recipe data. The following explains the ways of transferring recipe data.

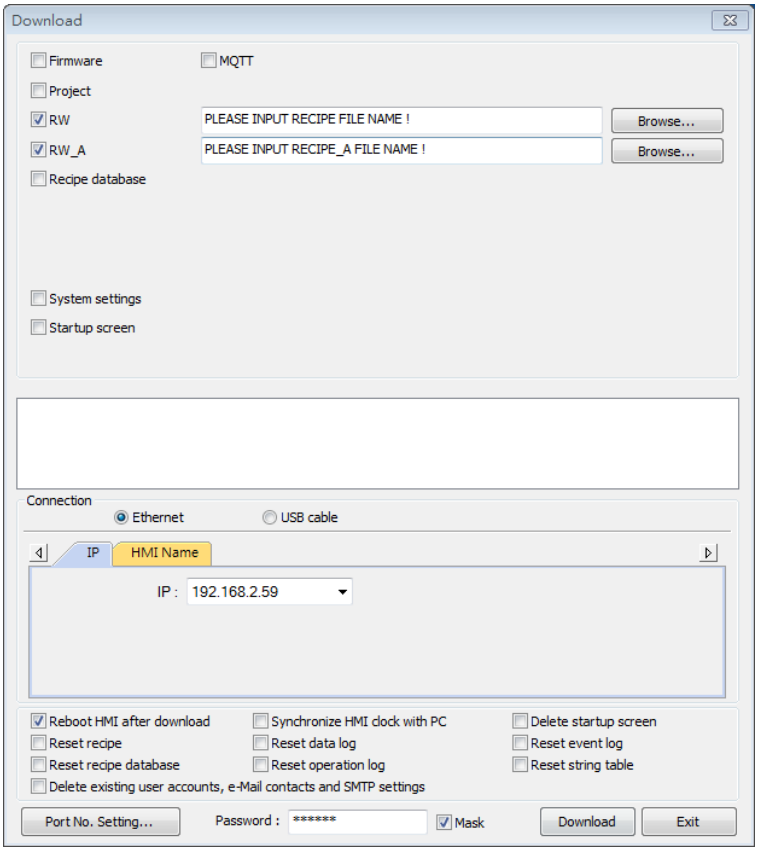
Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

17.2. Steps to Update Recipe Data with Ethernet or USB Cable

1. Open Utility Manager and click [Publish] » [Download].
2. Select [RW] and [RW_A] and [Browse] for the source file.
3. After downloading, restart HMI, RW and RW_A will be updated.

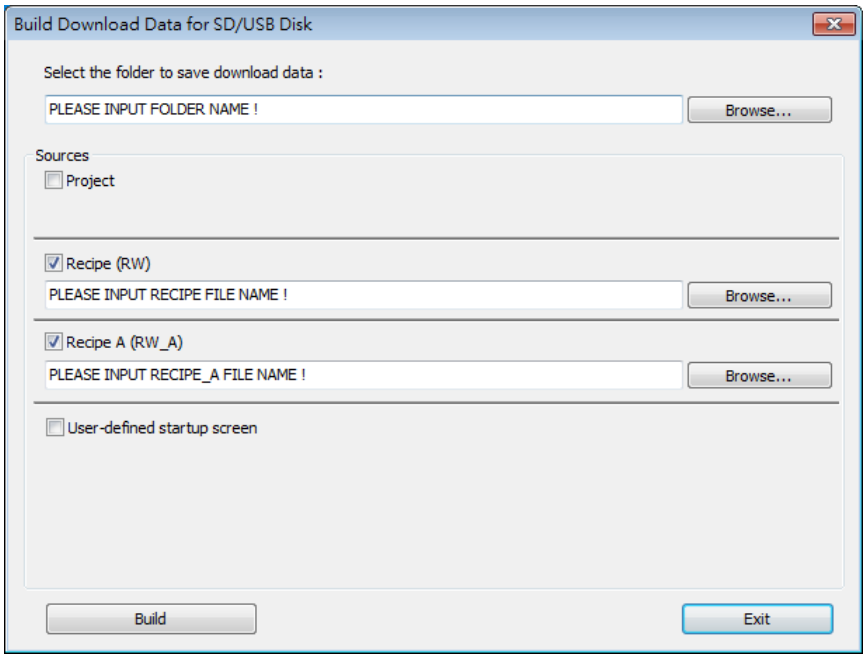
When [Reboot HMI after download] is selected, users don't have to manually reboot HMI.

When [Reset recipe] check box is selected, the system will clear all the data in [RW] and [RW_A] before downloading.



17.3. Steps to Update Recipe Data with SD Card or USB Disk

1. Open Utility Manager and click [Publish] » [Build Download Data for SD/USB Disk].
2. Insert a SD card or USB disk into PC.
3. Click [Browse] to designate the file path.
4. Click [Build], EasyBuilder Pro will save the data in SD card or USB disk.

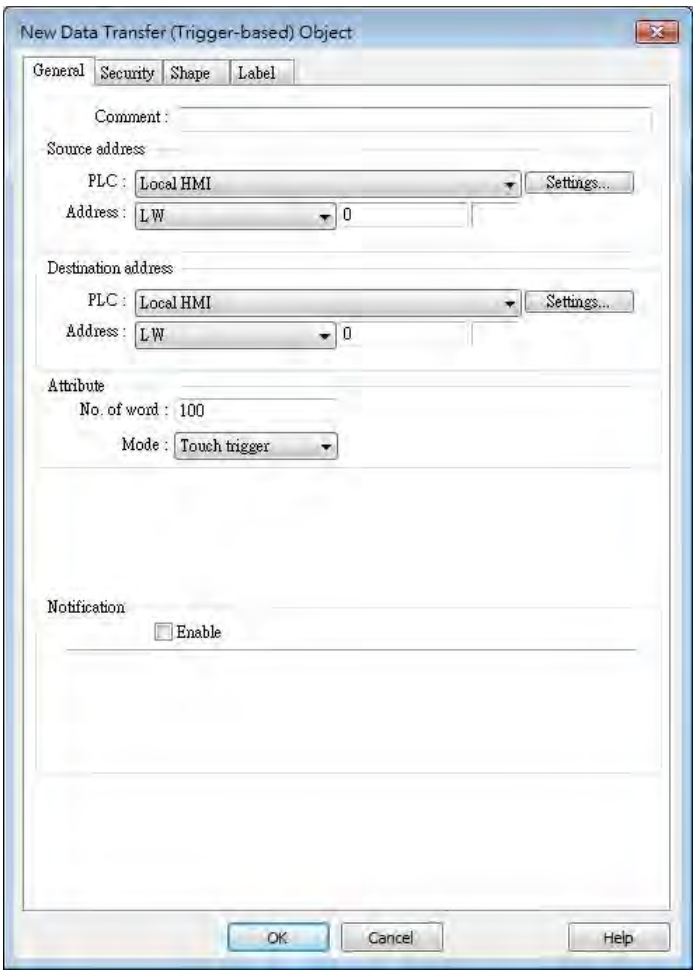


Note

- When download data is successfully built, two folders can be found. The folder named according to the model used *(e.g. emt3000, mt8000ie, mt8000xe, mtv, mtip…etc)* is for storing project file, and the folder named *history* is for storing recipe data and data sampling / event log records.
- Please manually reboot HMI to update recipe data.

17.4. Transferring Recipe Data

Use [Data Transfer (Trigger-based) Object] to transfer recipe data to a specific address, or save the data of this address in [RW] and [RW_A].



Setting	Description
Source address	Sets the source of the data.
Destination address	Sets the destination of the data to transfer to.
Attribute	Sets the number of words to transfer from source to destination.

17.5. Saving Recipe Data Automatically

In order to prolong the life span of HMI flash memory, the system will automatically save the recipe data to HMI every minute. To avoid losing data when turning HMI off during the interval of saving data, system register [LB-9029: Save all recipe data to machine (set ON)] is provided. Set ON LB-9029 will make the system save recipe data once. Set ON [LB-9028: Reset all recipe data (set ON)], the system will clear all recipe data.

18. Macro Reference

This chapter describes the syntax, programming methods and usage of macro commands.

18.1. Overview 18-2
18.2. Instructions to use the Macro Editor 18-2
18.3. Configuration 18-7
18.4. Syntax 18-8
18.5. Statement 18-14
18.6. Function Blocks 18-19
18.7. Built-In Function Block 18-22
18.8. How to Create and Execute a Macro 18-97
18.9. User Defined Macro Function 18-101
18.10. Some Notes about Using the Macro 18-114
18.11. Use the Free Protocol to Control a Device 18-114
18.12. Compiler Error Message 18-119
18.13. Sample Macro Code 18-125
18.14. Macro TRACE Function 18-130
18.15. Example of String Operation Functions 18-135
18.16. Macro Password Protection 18-142
18.17. Reading / Writing CAN bus Address Using Variable 18-144

18.1. Overview

Macros provide the additional functionality your application may need. Macros are automated sequences of commands that are executed at run-time. Macros allow you to perform tasks such as complex scaling operations, string handling, and user interactions with your projects. This chapter describes syntax, usage, and programming methods of macro commands.

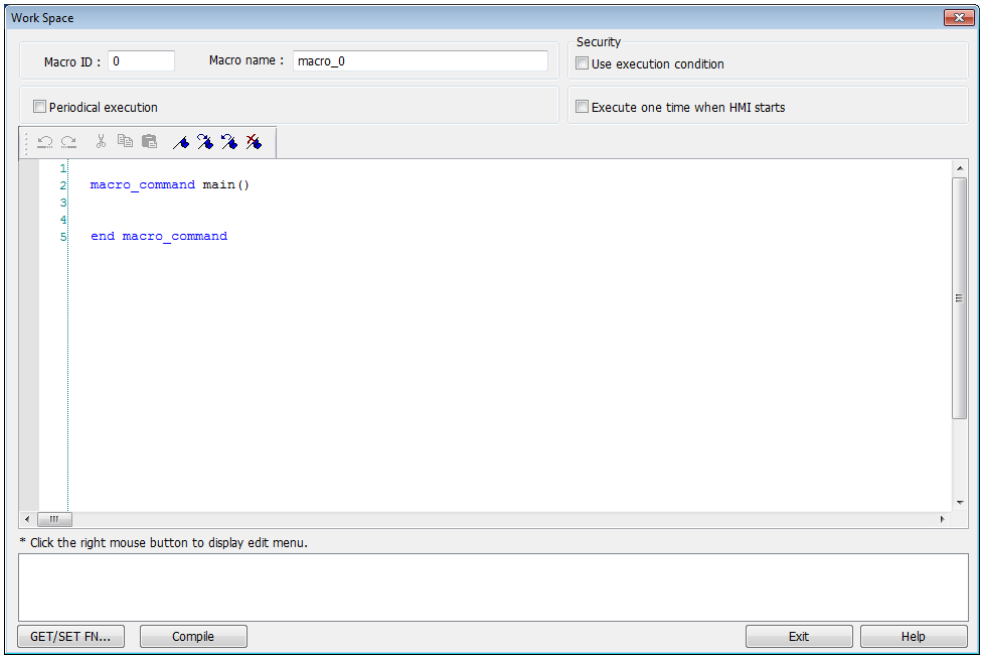
18.2. Instructions to use the Macro Editor

Macro editor provides the following functions:

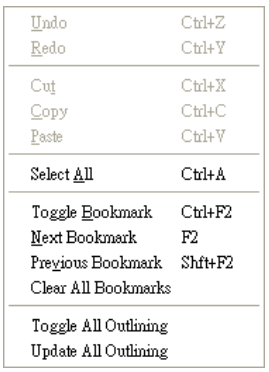
- Display line number
- Undo / Redo
- Cut / Copy / Paste
- Select All
- Toggle Bookmark / Previous Bookmark / Next Bookmark / Clear All Bookmarks
- Toggle All Outlining
- Security -> Use execution condition
- Periodical execution
- Execute one time when HMI starts

The instructions in the following part show you how to use these functions.

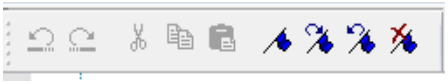
1. Open the macro editor; you'll see the line numbers displayed on the left-hand side of the edit area.



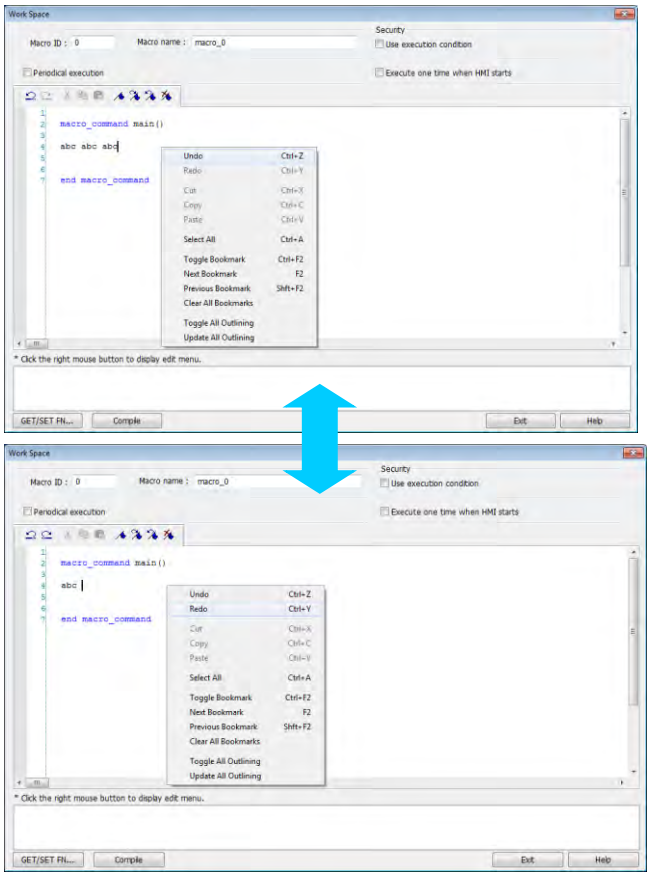
2. Right click on the edit area to open the pop-up menu as shown in the following figure. Disabled operations are colored grey, which indicates that it is not possible to use that function in the current status of the editor. For example, you should select some text to enable the copy function, otherwise it will be disabled. Keyboard shortcuts are also shown.



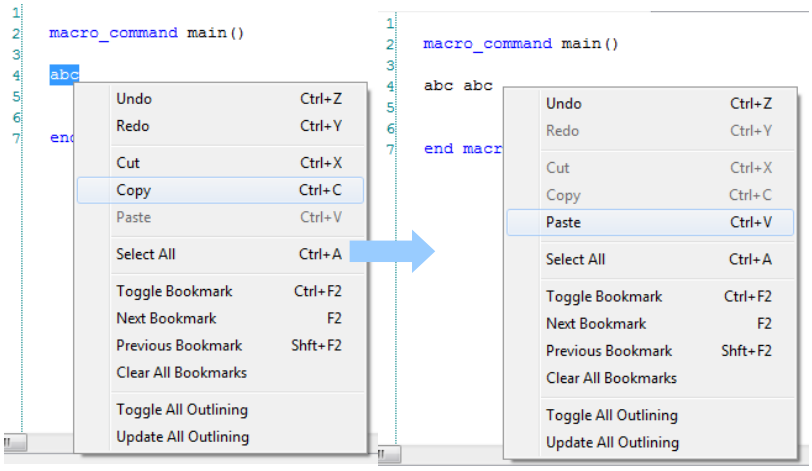
3. The toolbar provides [Undo], [Redo], [Cut], [Copy], [Paste], [Toggle Bookmark], [Next Bookmark], [Previous Bookmark] and [Clear All Bookmarks] buttons.



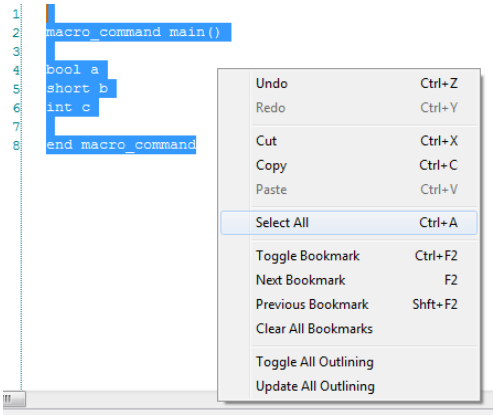
4. Any modification will enable the [Undo] function. [Redo] function will be enabled after the undo action is used. To perform the undo/redo, right click to select the item or use the keyboard shortcuts. (Undo: Ctrl+Z, Redo: Ctrl+Y).



5. Select a word in the editor to enable the [Cut] and [Copy] function. After [Cut] or [Copy] is performed, [Paste] function is enabled.

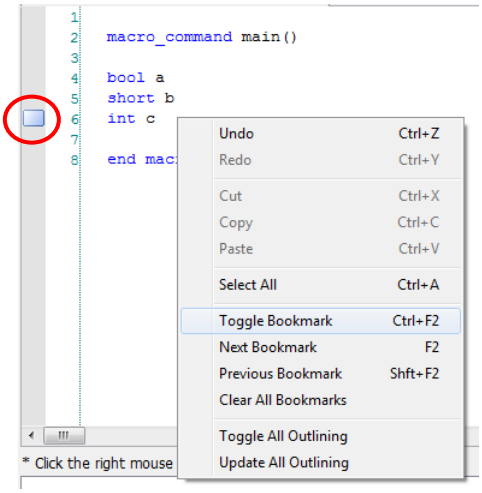


6. Use [Select All] to include all the content in the edit area.

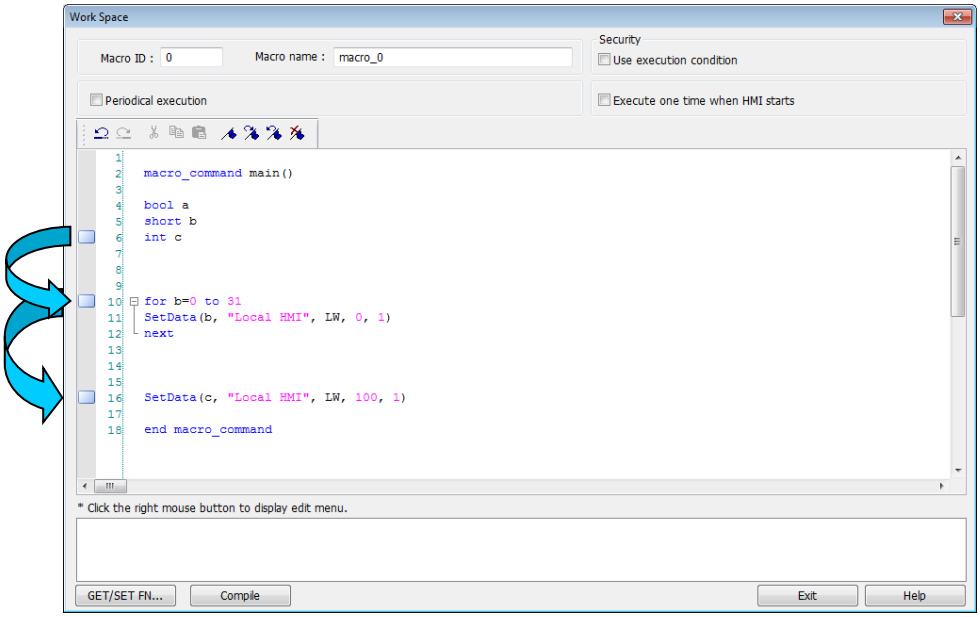


7. If the macro is too long, use bookmarks to manage and read the code with ease. The following illustration shows how it works.

- Move your cursor to the position in the edit area where to insert a bookmark. Right click, select [Toggle Bookmark]. There will be a blue little square that represents a bookmark on the left hand side of edit area.

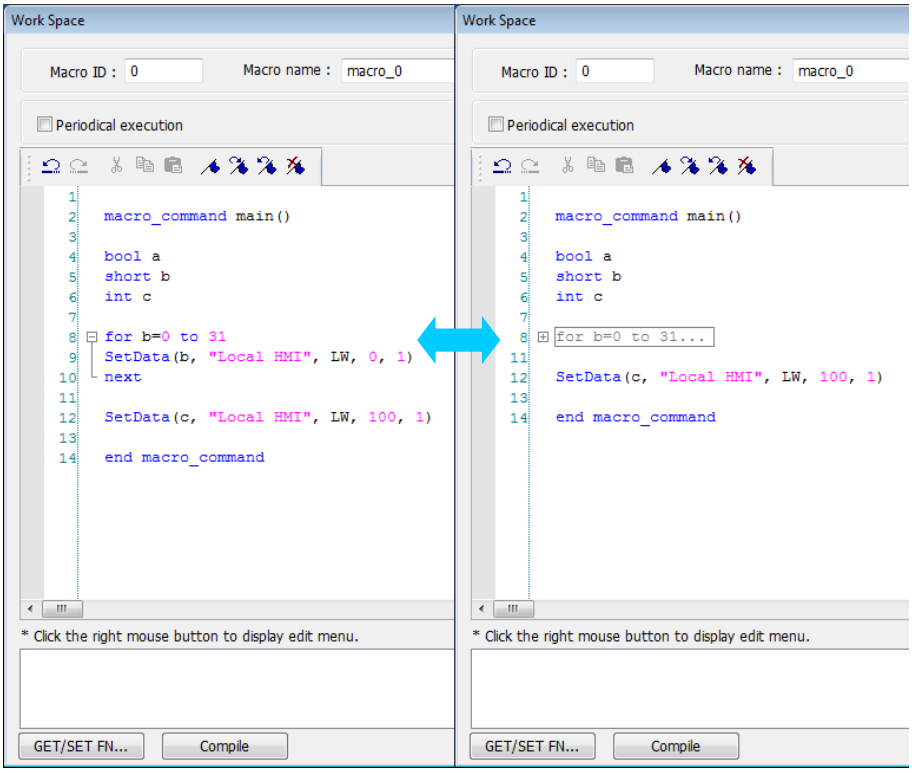


- If there is already a bookmark where the cursor is placed, select [Toggle Bookmark] to close it, otherwise to open it.
- Right click and select [Next Bookmark], the cursor will move to where the next bookmark locates. Selecting [Previous Bookmark] will move the cursor to the previous bookmark.

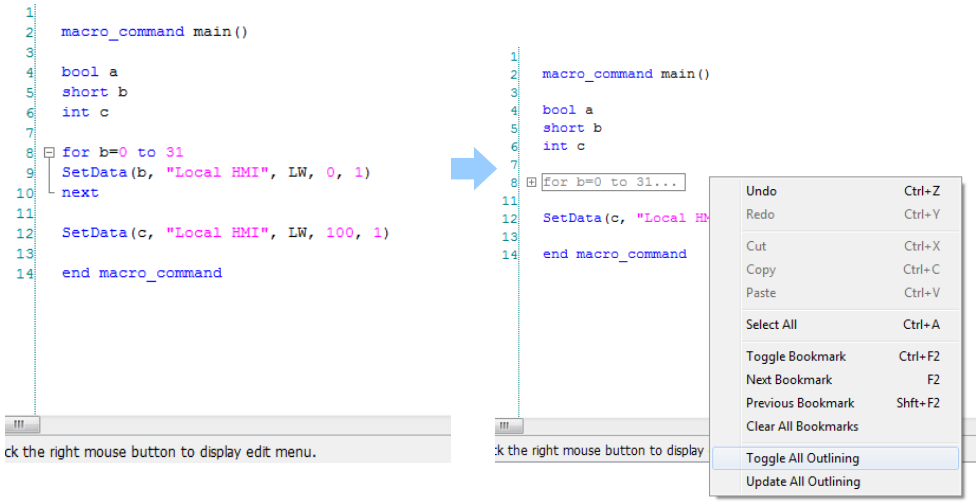


- Selecting [Clear All Bookmarks] will delete all bookmarks.

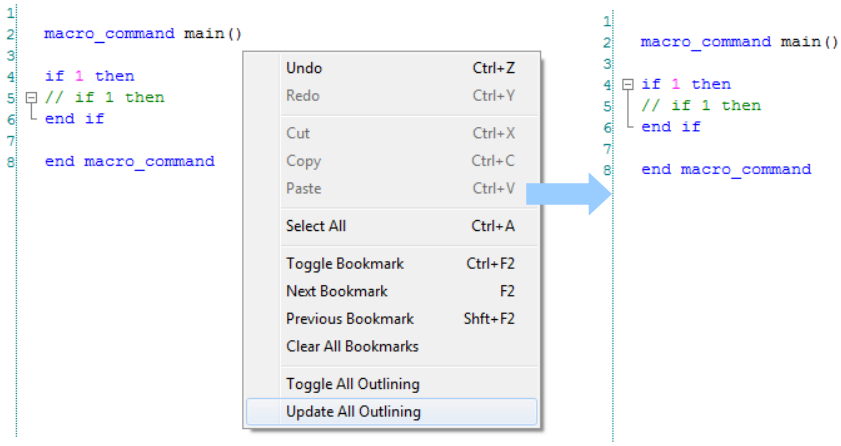
8. Macro editor provides outlining (or code-folding). Outlining will hide macro codes that belong to the same block, and display them as ⊞ . . . . There will be a tree diagram on the left hand side of edit area. Click ⊟ to hide the block or ⊞ to open, as shown in the following figure.



9. Right click to select [Toggle All Outlining] to open all folded macro code blocks.



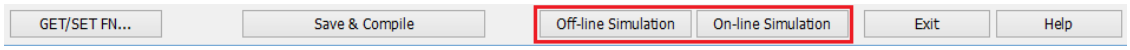
10. Sometimes the outlining might be incorrect since that the keywords are misjudged as shown in the following figure. To solve this problem, right click and select [Update All Outlining].



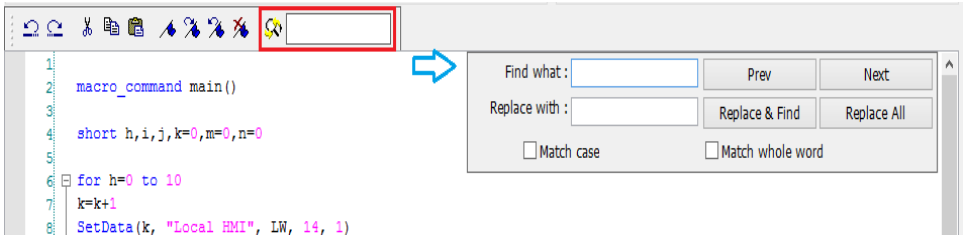
11. The statements enclosed in the following keywords are called a “block” of the macro code:

- Function block: sub – end sub
- Iterative statements:
  - i. for – next
  - ii. while – wend
- Logical statements:
  - i. if – end if
- Selective statements: select case – end select

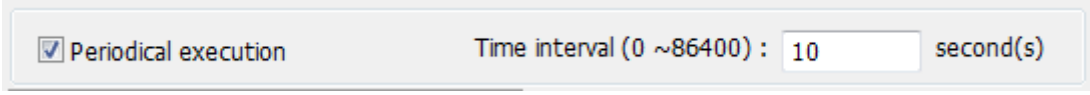
12. The macro editor is not a monopoly window. Returning to the main screen and editing the project with the Work Space window open is allowed.



13. The macro editor provides Find and Replace features.

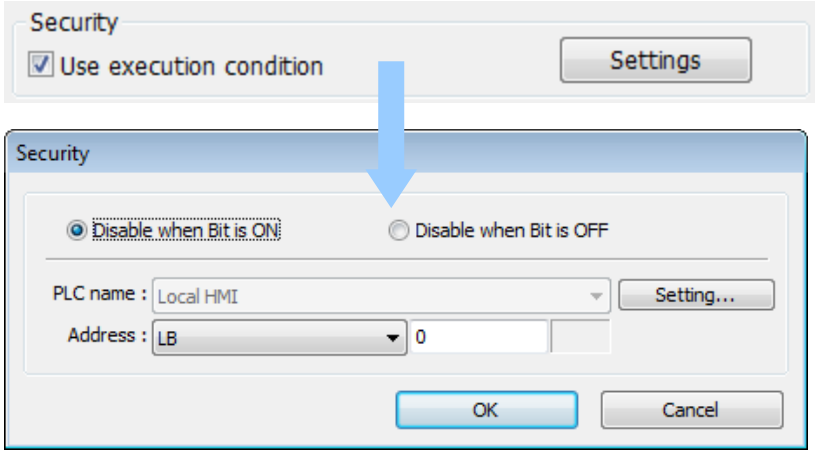


14. When [Periodical execution] is checked, this macro will be triggered periodically.



15. Select [Security] » [Use execution condition] » [Settings] to enable security settings:

- [Disable when Bit is ON]: When Bit is ON, this macro is disabled.
- [Disable when Bit is OFF]: When Bit is OFF, this macro is disabled.



16. Select [Execute one time when HMI starts], this macro will be executed once when HMI starts up.

18.3. Configuration

A macro contains statements. The statements contain constants, variables, and operations. The statements are put in a specific order to create the desired output.

A macro has the following structure:

```
Global Variable Declaration          ---------------------------------- Optional

  Sub Function Block Declarations    ---------------------------------- Optional
      Local Variable Declarations
  End Sub

  macro_command main()               ----------------------------------- Required
      Local Variable Declarations
      [Statements]
  end macro_command                  ----------------------------------- Required
```

Macro must have one and only one main function which is the execution start point of macro. The format is:

```
macro_command main()

end macro_command
```

Local variables are used within the main macro function or in a defined function block. Its value remains valid only within the specific block.
Global variables are declared before any function blocks and are valid for all functions in the macro. When local variables and global variables have the same declaration of name, only the local variables are valid.

The following example shows a simple macro which includes a variable declaration and a function call.

```
macro_command main()
  short pressure = 10                                  //  local variable declaration
  SetData(pressure, "Allen-Bradley DF1", N7, 0, 1)  //  function calling
end macro_command
```

18.4. Syntax

18.4.1. Constants and Variables

18.4.1.1. Constants

Constants are fixed values and can be directly written into statements. The formats are:

Constant Type	Note	Example
Decimal integer		345, -234, 0, 23456
Hexadecimal	Must begin with 0x	0x3b, 0xffff, 0x237
ASCII	Single character must be enclosed in single quotation marks and a string (group of characters) must be enclosed in double quotation marks. A backslash \ can be used to escape the quotation marks contained in a string. Therefore, to enclose a string containing double quotation marks, please use \" for the double quotation mark in the string.	‘a’, "data", "name"
Boolean		true, false

Here is an example using constants:

```
macro_command main()
short A, B        //  A and B are variables
A = 1234
B = 0x12          //  1234 and 0x12 are constants
end macro_command
```

18.4.1.2. Variables

Variables are names that represent information. The information can be changed as the variable is modified by statements.

Naming Rules for Variables

- A variable name must start with an alphabet.
- Variable names longer than 32 characters are not allowed.
- Reserved words cannot be used as variable names.

There are 8 different Variable types, 5 for signed data types and 3 for unsigned data types:

Variable Type	Description	Range
bool (boolean)	1 bit (discrete)	0, 1
char (character)	8 bits (byte)	+127 to -128
short (short integer)	16 bits (word)	+32767 to -32768
int (integer)	32 bits (double word)	+2147483647to -2147483648
float (floating point)	32 bits (double word)	
unsigned char	8 bits (byte)	0 to 255
unsigned short (short	16 bits (word)	0 to 65535

integer)		
unsigned int	32 bits (double word)	0 to 4,294,967,295
long (long integer)	64 bits (four words) (cMT / cMT X Series only)	+281474976710655 ~ -281474976710655
unsigned long (long integer)	64 bits (four words) (cMT / cMT X Series only)	0 ~ 281474976710655
double	64 bits (four words) (cMT / cMT X Series only)	

Declaring Variables

Variables must be declared before being used. To declare a variable, specify the type before the variable name.

Example:

```
int             a
short           b, switch
float           pressure
unsigned short  c
```

Declaring Arrays

Macros support one-dimensional arrays (zero-based index). To declare an array of variables, specify the type and the variable name followed by the number of variables in the array enclosed in brackets "[]". The length of an array could be 1 to 4096. (Macros only support at most 4096 variables per macro).

Example:

```
int     a[10]
short   b[20], switch[30]
float   pressure[15]
```

The minimum array index is 0 and the maximum is (array size – 1).

Example:

```
char data [100]     //  array size is 100
```

In this case, the minimum of array index is 0 and maximum of array index is 99 (=100-1)

Variable and Array Initialization

There are two ways variables can be initialized:

- By statement using the assignment operator (=)

Example:

```
int a
```

float b[3]

a = 10

b[0] = 1

- During declaration

char a = '5', b = 9

The declaration of arrays is a special case. The entire array can be initialized during declaration by enclosing comma separated values inside curly brackets "{}".

Example:

float data[4] = {11, 22, 33, 44} // now data[0] is 11, data[1] is 22....

18.4.2. Operators

Operators are used to designate how data is manipulated and calculated.

Operator	Description	Example
=	Assignment operator	pressure = 10

Arithmetic Operators	Description	Example
+	Addition	A = B + C
-	Subtraction	A = B – C
*	Multiplication	A = B * C
/	Division	A = B / C
% or mod	Modulo division (return remainder)	A = B % 5 or A = B mod 5

By default, integer numbers (1, 2,3..etc) are considered having integer data type; therefore, when division is carried out involving two integer numbers where the result should have decimal point, the decimal part will be removed. To avoid this, add .0 (1.0, 2.0, 3.0...etc) behind the dividend or the divisor to turn it into a floating point number calculation.

Examples:

A = 3 / 2 = 1 » 3 and 2 are both integers; therefore the result is an integer.

B = 3 / 2.0 = 1.5 » 3 is an integer whereas 2.0 is a floating point number, therefore the result is a floating point number.

C = 3.0 / 2 = 1.5 » 3.0 is a floating point number whereas 2 is an integer, therefore the result is a floating point number.

Comparison Operators	Description	Example
<	Less than	if A < 10 then B = 5
<=	Less than or equal to	if A <= 10 then B = 5
>	Greater than	if A > 10 then B = 5
>=	Greater than or equal to	if A >= 10 then B = 5
==	Equal to	if A == 10 then B = 5

<>	Not equal to	if A <> 10 then B = 5

Logic Operators	Description	Example
and	Logical AND	if A < 10 and B > 5 then C = 10
or	Logical OR	if A >= 10 or B > 5 then C = 10
xor	Logical Exclusive OR	if A xor 256 then B = 5
not	Logical NOT	if not A then B = 5

Shift and bitwise operators are used to manipulate bits of signed/unsigned character and integer variables. The priority of these operators is from left to right within the statement.

Shift Operators	Description	Example
<<	Shifts the bits in a bit set to the left a specified number of positions	A = B << 8
>>	Shifts the bits in a bit set to the right a specified number of positions	A = B >> 8

Bitwise Operators	Description	Example
&	Bitwise AND	A = B & 0xf
|	Bitwise OR	A = B | C
^	Bitwise XOR	A = B ^ C
~	One's complement	A = ~B

Priority of All Operators

The overall priority of all operations from highest to lowest is as follows:

1. Operations within parenthesis are carried out first
2. Arithmetic operations
3. Shift and Bitwise operations
4. Comparison operations
5. Logic operations
6. Assignment

Reserved Keywords

The following keywords are reserved for system. These keywords cannot be used as variable, array, or function names.

+, -, *, /, %, >=, >, <=, <, <>, ==, and, or, xor, not, <<, >>,=, &, |, ^, ~

exit, macro_command, for, to, down, step, next, return, bool, short, int, char, float, void, if, then, else, break, continue, set, sub, end, while, wend, true, false
SQRT, CUBERT, LOG, LOG10, SIN, COS, TAN, COT, SEC, CSC, ASIN, ACOS, ATAN, BIN2BCD, BCD2BIN, DATE2ASCII, DATE2DEC,DEC2ASCII, FLOAT2ASCII, HEX2ASCII, ASCII2DEC, ASCII2FLOAT, ASCII2HEX, FILL, RAND, DELAY, SWAPB, SWAPW, LOBYTE, HIBYTE, LOWORD, HIWORD, GETBIT, SETBITON, SETBITOFF, INVBIT, ADDSUM, XORSUM, CRC, CRC8, INPORT, OUTPORT, POW, GetCnvTagArrayIndex, GetError, GetData, GetDataEx, SetData, SetDataEx, SetRTS, GetCTS, Beep, SYNC_TRIG_MACRO, ASYNC_TRIG_MACRO, TRACE, FindDataSamplingDate, FindDataSamplingIndex, FindEventLogDate, FindEventLogIndex
StringGet, StringGetEx, StringSet, StringSetEx, StringCopy, StringMid, StringMD5, StringDecAsc2Bin, StringBin2DecAsc, StringDecAsc2Float, StringFloat2DecAsc, StringHexAsc2Bin, StringBin2HexAsc, StringLength, StringCat, StringCompare, StringCompareNoCase, StringFind, StringReverseFind, StringFindOneOf, StringIncluding, StringExcluding, StringToUpper, StringToLower, StringToReverse, StringTrimLeft, StringTrimRight, StringInsert, String2Unicode, Unicode2Utf8, UnicodeCat, UnicodeCompare, UnicodeCopy, UnicodeExcluding, Uft82Unicode

18.5. Statement

18.5.1. Definition Statement

This covers the declaration of variables and arrays. The formal construction is as follows:

```
type        name
```

This defines a variable with name as “name” and type as “type”.

Example:

int A // define a variable A as an integer

```
type        name[constant]
```

This defines an array variable called “name” with size as “constant” and type as “type”.

Example:

int B[10] // where define a variable B as a one-dimensional array of size 10

18.5.2. Assignment Statement

Assignment statements use the assignment operator to move data from the expression on the right side of the operator to the variable on the left side. An expression is the combination of variables, constants and operators to yield a value.

```
VariableName      Expression
```

Example

A = 2 where a variable A is assigned to 2

18.5.3. Logical Statements

Logical statements perform actions depending on the condition of a boolean expression.

The syntax is as follows:

Single-Line Format

```
If <Condition> then
   [Statements]
else
   [Statements]
end if
```

Example:

```
if a == 2 then
        b = 1
else
        b = 2
end if
```

Block Format

If <Condition> **then**
  [Statements]
else if <Condition-n> **then**
[Statements]
else
  [Statements]
end if

Example:

```
if a == 2 then
        b = 1
else if a == 3 then
        b = 2
else
        b = 3
end if
```

Syntax description

if	Must be used to begin the statement.
<Condition>	Required. This is the controlling statement. It is FALSE when the <Condition> evaluates to 0 and TRUE when it evaluates to non- zero.
then	Must precede the statements to execute if the <Condition> evaluates to TRUE.
[Statements]	It is optional in block format but necessary in single-line format without else. The statement will be executed when the <Condition> is TRUE.
else if	Optional. The else if statement will be executed when the relative <Condition-n> is TRUE.
<Condition-n>	Optional. see <Condition>
else	Optional. The else statement will be executed when <Condition> and <Condition-n> are both FALSE.
end if	Must be used to end an if-then statement.

18.5.4. Selective Statements

The select-case construction can be used like multiple if-else statements and perform selected actions depending on the value of the given variable. When the matched value is found, all the actions below will be executed until a break statement is met. The syntax is as follows:

Format without a Default Case

```
Select Case [variable]
Case [value]
    [Statements]
    break
end Select
```

Example:

```
Select Case A
    Case 1
        b=1
    break
end Select
```

Format with a Default Case (Case else)

```
Select Case [variable]
Case [value]
    [Statements]
    break
    Case else
    [Statements]
    break

end Select
```

Example:

```
Select Case A
    Case 1
        b=1
    break
    Case else
        b=0
    break
end Select
```

Multiple cases in the same block

```
Select Case [variable]
Case [value1]
    [Statements]
Case [value2]
    [Statements]
    break

end Select
```

Example:

```
Select Case A
    Case 1
    break
    Case 2
        b=2
    break
    Case 3
        b=3
    break
end Select
```

Syntax description

Select Case	Must be used to begin the statement.
[variable]	Required. The value of this variable will be compared to the value of each case.
Case else	Optional. It represents the default case. If none of the cases above are matched, the statements under default case will be executed. When a default case is absent, it will skip directly to the end of the select-case statements if there is no matched case.
break	Optional. The statements under the matched case will be executed until the break command is reached. If a break command is absent, it simply keeps on executing next statement until the end command is reached.
end Select	Indicates the end of the select-case statements.

18.5.5. Iterative Statements

Iterative statements control loops and repetitive tasks depending on condition. There are two types of iterative statements.

18.5.5.1. for-next Statements

The for-next statement runs for a fixed number of iterations. A variable is used as a counter to track the progress and test for ending conditions. Use this for fixed execution counts. The syntax is as follows:

```
for [Conunter] = <StartValue> to <EndValue> [step <StepValue>]
    [Statements]
next [Counter]
```

Or

```
for [Conunter] = <StartValue> to <EndValue> [step <StepValue>]
    [Statements]
next [Counter]
```

Example:

```
for a = 0 to 10 step 2
    b = a
next a
```

Syntax description

for	Must be used to begin the statement
[Counter]	Required. This is the controlling statement. The result of evaluating the variable is used as a test of comparison.
<StartValue>	Required. The initial value of [Counter]
to/down	Required. This determines if the <step> increments or decrements the <Counter>. “to” increments <Counter> by <StepValue>. “down” decrements <Counter> by <StepValue>.
<EndValue>	Required. The test point. If the <Counter> is greater than this value, the macro exits the loop.
step	Optional. Specifies that a <StepValue> other than one is to be used.
[StepValue]	Optional. The increment/decrement step of <Counter>. It can be omitted when the value is 1 If [step <StepValue>] are omitted the step value defaults to 1.
[Statements]	Optional. Statements to execute when the evaluation is TRUE. “for-next” loops may be nested.
next	Required.
[Counter]	Optional. This is used when nesting for-next loops.

18.5.5.2. while-wend Statements

The while-wend statement runs for an unknown number of iterations. A variable is used to test for ending conditions. When the condition is TRUE, the statements inside are executed repetitively until the condition becomes FALSE. The syntax is as follows.

```
while <Condition>
        [Statements]
wend
```

Example:

```
        while a < 10
            a = a + 10
        wend
```

Syntax description

while	Must be used to begin the statement.
continue	Required. This is the controlling statement. When it is TRUE, the loop begins execution. When it is FALSE, the loop terminates.
return [value]	Statements to execute when the evaluation is TRUE.
wend	Indicates the end of the while-end statements.

18.5.5.3. Other Control Commands

break	Used in for-next and while-wend. It skips immediately to the end of the iterative statement.
continue	Used in for-next and while-wend. It ends the current iteration of a loop and starts the next one.
return	The return command inside the main block can force the macro to stop anywhere. It skips immediately to the end of the main block.

18.6. Function Blocks

Function blocks are useful for reducing repetitive codes. It must be defined before use and supports any variable and statement type. A function block could be called by putting its name followed by parameters in parenthesis. After the function block is executed, it returns the value to the caller function where it is used as an assignment value or as a condition. A return type is not required in function definition, which means that a function block does not have to return a value. The parameters can also be ignored in function definition while the function has no need to take any parameters from the caller. The syntax is as follows:

Function definition with return type

```
sub type <name> [(parameters)]
        Local variable declarations
        [Statements]
        [return [value]]
end sub
```

Example:

```
   sub int Add(int x, int y)
                int result
                result = x +y
                return result
            end sub

          macro_command main()
                int a = 10, b = 20, sum
                sum = Add(a, b)
          end macro_command
```

or:

```
   sub int Add()
                int result, x=10, y=20
                result = x +y
                return result
            end sub
          macro_command main()
                int    sum
                sum = Add()
          end macro_command
```

Function definition without return type

```
sub <name> [(parameters)]
        Local variable declarations
        [Statements]
end sub
```

Example:

```
   sub Add(int x, int y)
                int result
                result = x +y
```

```
        end sub

    macro_command main()
        int    a = 10, b = 20
        Add(a, b)
    end macro_command
```

or:

```
  sub Add()
        int result, x=10, y=20
        result = x +y
      end sub

    macro_command main()
        Add()
    end macro_command
```

Syntax description

sub	Must be used to begin the function block
type	Optional. This is the data type of value that the function returns. A function block is not always necessary to return a value.
(parameters)	Optional. The parameters hold values that are passed to the function. The passed parameters must have their type declared in the parameter field and assigned a variable name. For example: sub int MyFunction(int x, int y). x and y would be integers passed to the function. This function is called by a statement that looks similar to this: ret = MyFunction(456, pressure) where “pressure” must be integer according to the definition of function. Notice that the calling statement can pass hard coded values or variables to the function. After this function is executed, an integer values is return to ‘ret’.
Local variable declaration	Variables that are used in the function block must be declared first. This is in addition to passed parameters. In the above example x and y are variables that the function can used. Global variables are also available for use in function block.
[Statements]	Statements to execute
[return [value]]	Optional. Used to return a value to the calling statement. The value can be a constant or a variable. Return also ends function block execution. A function block is not always necessary to return a value, but, when the return type is defined in the beginning of the definition of function, the return command is needed.
end sub	Must be used to end a function block.

18.7. Built-In Function Block

EasyBuilder Pro has many built-in functions for retrieving and transferring data to the devices, data management and mathematical functions.

18.7.1. Table of Functions

Please click on one of the function names in the table to see its details.

Function Name	Description
Device Functions	
GetData	Receives data from the device.
GetDataEx	Receives data from the device and continues executing next command even if there's no response from the device.
GetError	Gets an error code.
SetData	Sends data to the device.
SetDataEx	Sends data to the device and continues executing next command even if there's no response from the device.
Free Protocol Functions	
GetCTS	Gets CTS signal of RS-232.
INPORT	Reads data from a COM port or Ethernet port.
INPORT2	Reads data from a COM port or Ethernet port and then wait for a the designated period of time.
INPORT3	Reads data from a COM port or Ethernet port according to the specified data size.
INPORT4	Reads data from a COM port or Ethernet port and then stops reading data when the ending character is reached.
OUTPORT	Sends out the specified data to a device or controller via a COM port or Ethernet port.
PURGE	Clears the input and output buffers associated with the COM port.
SetRTS	Raises or lowers the RTS signal of RS-232.
Process Control Functions	
ASYNC_TRIG_MACRO	Triggers the execution of a macro asynchronously in a running macro.
SYNC_TRIG_MACRO	Triggers the execution of a macro synchronously in a running macro. The current macro will pause until the end of execution of this called macro.

DELAY	Suspends the execution of the current macro for at least the specified interval (*time*).
Data Operation Functions	
FILL	Sets array elements to the specified value.
SWAPB	Exchanges the high-byte and low-byte data of a 16-bit (Word).
SWAPW	Exchanges the high-word and low-word data of a 32-bit (DINT).
LOBYTE	Retrieves the low byte of a 16-bit source.
HIBYTE	Retrieves the high byte of a 16-bit source.
LOWORD	Retrieves the low word of a 32-bit source.
HIWORD	Retrieves the high word of a 32-bit source.
INVBIT	Inverts the state of designated bit position of a data source.
SETBITON	Changes the state of designated bit position of a data source to 1.
SETBITOFF	Changes the state of designated bit position of a data source to 0.
GETBIT	Gets the state of designated bit position of a data source.
Data Type Conversion Functions	
ASCII2DEC	Converts an ASCII string to a decimal value.
ASCII2FLOAT	Converts an ASCII string to a float value.
ASCII2HEX	Converts an ASCII string to a hexadecimal value.
BIN2BCD	Converts a binary-type value to a BCD-type value.
BCD2BIN	Converts a BCD-type value to a binary-type value.
DATE2ASCII	Converts current date to an ASCII string.
DATE2DEC	Converts current date to a decimal value.
DEC2ASCII	Converts a decimal value to an ASCII string.
FLOAT2ASCII	Converts a floating value to an ASCII string.
HEX2ASCII	Converts a hexadecimal value to an ASCII string.
StringDecAsc2Bin	Converts a decimal string to an integer.
StringBin2DecAsc	Converts an integer to a decimal string.
StringDecAsc2Float	Converts a decimal string to floats.
StringFloat2DecAsc	Converts a float to a decimal string.
StringHexAsc2Bin	Converts a hexadecimal string to binary data.
StringBin2HexAsc	Converts binary data to a hexadecimal string.
String Operation Functions	
String2Unicode	Converts all the characters in the source string to Unicode.
StringCat	Appends source string to destination string.
StringCompare	Performs a case-sensitive comparison of two strings.

StringCompareNoCase	Performs a case-insensitive comparison of two strings.
StringCopy	Copies one string to another.
StringExcluding	Retrieves a substring of the source string that contains characters that are not in the set string.
StringFind	Returns the zero-based index of the first character of substring in the source string that matches the target string.
StringFindOneOf	Returns the zero-based index of the first character in the source string that is also in the target string.
StringGet	Receives data from the device.
StringGetEx	Receives data from the device and continues executing next command even if there's no response from the device.
StringIncluding	Retrieves a substring of the source string that contains characters in the set string, beginning with the first character in the source string and ending when a character is found in the source string that is not in the target string.
StringInsert	Inserts a string in a specific location within the destination string content.
StringLength	Obtains the length of a string.
StringMD5	Generates a string using MD5 message-digest algorithm.
StringMid	Retrieves a character sequence from the specified offset of the source string.
StringReverseFind	Returns the position of the last occurrence of target string in the source string.
StringSet	Sends data to the device.
StringSetEx	Sends data to the device and continues executing next command even if there's no response from the device.
StringToUpper	Converts all the characters in the source string to uppercase characters.
StringToLower	Converts all the characters in the source string to lowercase characters.
StringToReverse	Reverses the characters in the source string
StringTrimLeft	Trims the leading specified characters in the set buffer from the source string.
StringTrimRight	Trims the trailing specified characters in the set buffer from the source string.
Unicode2Utf8	Converts a Unicode string to a UTF8 string.
UnicodeCat	Concatenates two Unicode Strings

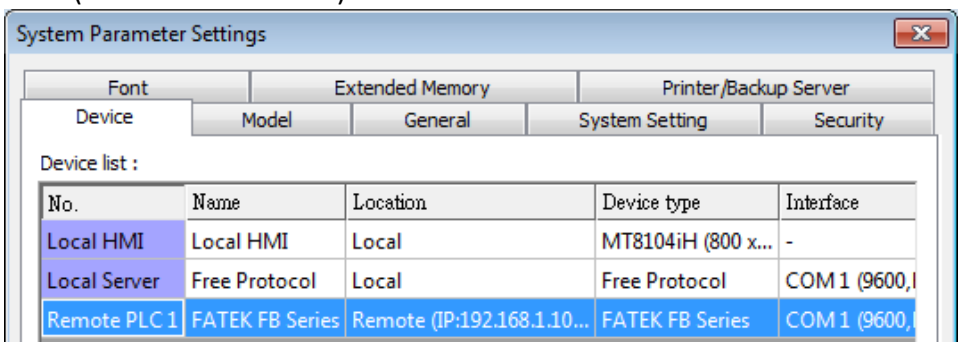
UnicodeCompare	Performs case-sensitive comparison between two Unicode strings.
UnicodeCopy	Copies a Unicode string.
UnicodeExcluding	Retrieves a substring of the source string that contains characters that are not in the set string.
UnicodeLength	Obtains the length of a Unicode string.
Utf82Unicode	Converts a UTF8 string to a Unicode string.
Mathematics Functions	
SQRT	Calculates the square root of source.
CUBERT	Calculates the cube root of source.
POW	Calculates the exponential of source.
SIN	Calculates the sine of source.
COS	Calculates the cosine of source.
TAN	Calculates the tangent of source.
COT	Calculates the cotangent of source.
SEC	Calculates the secant of source
CSC	Calculates the cosecant of source.
ASIN	Calculates the arc sine of source.
ACOS	Calculates the arc cosine of source.
ATAN	Calculates the arc tangent of source.
LOG	Calculates the natural logarithm of a number.
LOG10	Calculates the base-10 logarithm of a number.
RAND	Calculates a random integer.
CEIL	Get the smallest integral value that is not less than input.
FLOOR	Get the largest integral value that is not greater than input.
ROUND	Get the integral value that is nearest the input.
Statistics Functions	
AVERAGE	Gets the average value from array.
HARMEAN	Gets the harmonic mean value from array.
MAX	Gets the maximum value from array.
MEDIAN	Gets the median value from array.
MIN	Gets the minimum value from array.
STDEVP	Gets the standard deviation value from array.
STDEVS	Gets the sample standard deviation value from array.
Recipe Database Functions	
RecipeGetData	Gets recipe Data.
RecipeQuery	Queries recipe data.

RecipeQueryGetData	Gets the data in the query result obtained by RecipeQuery.
RecipeQueryGetRecordID	Gets the record ID numbers of those records gained by RecipeQuery.
RecipeSetData	Writes data to recipe database.
Data / Event Log Functions	
FindDataSamplingDate	Finds the date of the specified data sampling file.
FindDataSamplingIndex	Finds the file index of the specified data sampling file.
FindEventLogDate	Finds the date of the specified event log file.
FindEventLogIndex	Finds the file index of the specified event log file.
Checksum Functions	
ADDSUM	Adds up the elements of an array to generate a checksum.
XORSUM	Uses XOR to calculate the checksum.
BCC	Same as XORSUM.
CRC	Calculates 16-bit CRC of the variables to generate a checksum.
CRC8	Calculates 8-bit CRC of the variables to generate a checksum.
Miscellaneous Functions	
Beep	Plays beep sound.
Buzzer	Turns ON / OFF the buzzer.
TRACE	Prints out the current value of variables during run-time of macro for debugging.
GetCnvTagArrayIndex	When an user-defined conversion tag uses array, the [Read conversion] subroutine can get the relative array index before doing conversion.

18.7.2. Device

Name	GetData
Syntax	GetData(*read_data*[*start*], *device_name, device_type, address_offset, data_count*) or GetData(*read_data, device_name, device_type, address_offset, 1*)
Description	Receives data from the device. Data is stored into *read_data*[*start*]~ *read_data*[*start + data_count - 1*]. *data_count* is the amount of received data. In general, *read_data* is an array, but if *data_count* is 1, *read_data* can be an array or an ordinary variable. Below are two methods to read one word data from the device. macro_command main() short read_data_1[2], read_data_2 GetData(read_data_1[0], "FATEK KB Series", RT, 5, 1) GetData(read_data_2, "FATEK KB Series", RT, 5, 1) end macro_command

Device_name is the device name enclosed in the double quotation marks (“) and this name has been defined in the device list of system parameters as follows (see FATEK KB Series):



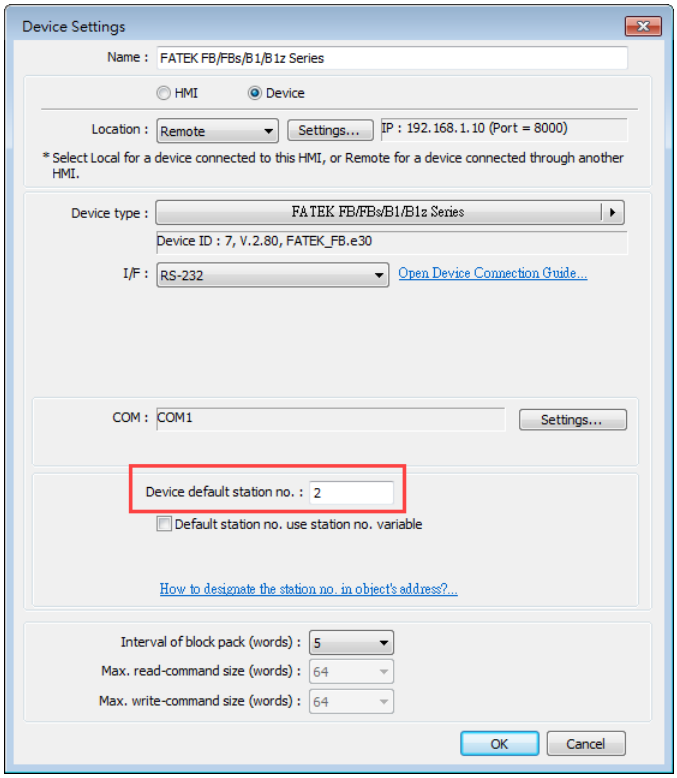
Device_type is the device type and encoding method (binary or BCD) of the device data. For example, if *device_type* is LW_BIN, it means the register is LW and the encoding method is binary. If use BIN encoding method, "_BIN" can be ignored.

If *device_type* is LW_BCD, it means the register is LW and the encoding method is BCD.

Address_offset is the address offset in the device.

For example, GetData(read_data_1[0], "FATEK KB Series", RT, 5, 1) represents that the address offset is 5.

If *address_offset* uses the format –"N#AAAAA", N indicates that device’s station number is N. AAAAA represents the address offset. This format is used while multiple devices or controllers are connected to a single serial port. For example, GetData(read_data_1[0], "FATEK KB Series", RT, 2#5, 1) represents that the device’s station number is 2. If GetData() uses the default station number defined in the device list as follows, it is not necessary to define station number in *address_offset*.



The number of registers actually read from depends on both the type of the *read_data* variable and the value of the number of *data_count*.

type of read_data	data_count	actual number of 16-bit register read
char (8-bit)	1	1
char (8-bit)	2	1
bool (8-bit)	1	1
bool (8-bit)	2	1
short (16-bit)	1	1
short (16-bit)	2	2
int (32-bit)	1	2
int (32-bit)	2	4
float (32-bit)	1	2
float (32-bit)	2	4

When a GetData() is executed using a 32-bit data type (int or float), the function will automatically convert the data. For example,

```
macro_command main()
float f
GetData(f, "MODBUS", 6x, 2, 1)      // f will contain a floating point value
end macro_command
```

Example

```
macro_command main()
bool a
bool b_array[30]
char c
```

	char c_array[20]
	short s
	short s_array[50]
	int i
	int i_array[10]
	float f
	float f_array[15]double g[10]
	
	// get the state of LB2 to the variable a
	GetData(a, "Local HMI", LB, 2, 1)
	
	// get 30 states of LB0 ~ LB29 to the variables b_array[0] ~ b_array[29]
	GetData(b_array[0], "Local HMI", LB, 0, 30)
	
	// get lower byte of LW-0 to the variable c
	// note that char is 1 byte, and a LW address occupies 2 bytes (1 word). Reading the first byte in a word register will get the lower byte of the word.
	// Ex: when the value in LW-0 is 0x0201, then variable c will read 0x01
	GetData(c, "Local HMI", LW, 0, 1)
	
	// get data of LW1 ~ LW10 to the c_array[0] ~ c_array[19]
	GetData(c_array[0], "Local HMI", LB, 0, 20)
	
	// get one word from LW-2 to the variable s
	GetData(s, "Local HMI", LW, 2, 1)
	
	// get 50 words from LW-0 ~ LW-49 to the variables s_array[0] ~ s_array[49]
	GetData(s_array[0], "Local HMI", LW, 0, 50)
	
	// get 2 words from LW-6 ~ LW-7 to the variable e
	// Ex: When value in LW-6 is 0x0002, in LW-7 is 0x0001, then i will read 0x00010002(65538)
	// note that int occupies 2 words (32-bit)
	GetData(i, "Local HMI", LW, 6, 1)
	
	// get 20 words (10 integer values) from LW-0 ~ LW-19 to variables i_array[0] ~ i_array[9], note that type of i_array[10] is int.
	GetData(i_array[0], "Local HMI", LW, 0, 10)
	
	// get data from LW-10 ~ LW-11 to the variable f
	// note that type of variable f is float.
	GetData(f, "Local HMI", LW, 10, 1)
	
	// get 30 words (15 float variables) from LW-0 ~ LW-29 to variables f_array[0] ~ f_array[14], note that type of f_array[15] is float.
	// note that float occupies 2 words (32-bit)

	GetData(f_array[0], "Local HMI", LW, 0, 15) end macro_command

Name	GetDataEx
Syntax	GetDataEx(*read_data*[*start*], *device_name, device_type, address_offset, data_count*) or GetDataEx(*read_data, device_name, device_type, address_offset, 1*)
Description	Receives data from the device and continues executing next command even when the read operation fails. Descriptions of *read_data, device_name, device_type, address_offset* and *data_count* are the same as GetData.

Example	macro_command main()
	bool a
	bool b
	bool b_array[30]
	char c
	char c_array[20]
	short s
	short s_array[50]
	int i
	int i_array[10]
	float f
	float f_array[15]
	
	// get the state of LB2 to the variable a
	GetDataEX(a, "Local HMI", LB, 2, 1)
	
	// get 30 states of LB0 ~ LB29 to the variables b_array[0] ~ b_array[29]
	GetDataEX(b_array[0], "Local HMI", LB, 0, 30)
	
	// get lower byte of LW-0 to the variable c
	// note that char is 1 byte, and a LW address occupies 2 bytes (1 word). Reading the first byte in a word register will get the lower byte of the word.
	// Ex: when the value in LW-0 is 0x0201, then variable c will read 0x01
	GetDataEX(c, "Local HMI", LW, 0, 1)
	
	// get data of LW1 ~ LW10 to the c_array[0] ~ c_array[19]
	GetDataEX(c_array[0], "Local HMI", LB, 0, 20)
	
	// get one word from LW-2 to the variable s
	GetDataEX(s, "Local HMI", LW, 2, 1)
	
	// get 50 words from LW-0 ~ LW-49 to the variables s_array[0] ~ s_array[49]
	GetDataEX(s_array[0], "Local HMI", LW, 0, 50)

	// get 2 words from LW-6 ~ LW-7 to the variable e
	// Ex: When value in LW-6 is 0x0002, in LW-7 is 0x0001, then i will read 0x00010002(65538)
	// note that int occupies 2 words (32-bit)
	GetDataEX(i, "Local HMI", LW, 6, 1)
	
	// get 20 words (10 integer values) from LW-0 ~ LW-19 to variables i_array[0] ~ i_array[9], note that type of i_array[10] is int.
	GetDataEX(i_array[0], "Local HMI", LW, 0, 10)
	
	// get data from LW-10 ~ LW-11 to the variable f
	// note that type of variable f is float.
	GetDataEX(f, "Local HMI", LW, 10, 1)
	
	// get 30 words (15 float variables) from LW-0 ~ LW-29 to variables f_array[0] ~ f_array[14], note that type of f_array[15] is float.
	// note that float occupies 2 words (32-bit)
	GetDataEX(f_array[0], "Local HMI", LW, 0, 15)
	
	end macro_command

Name	GetError
Syntax	GetError (*err*)
Description	Gets an error code.

Example	macro_command main()
	short err
	char byData[10]
	
	GetDataEx(byData[0], "MODBUS RTU", 4x, 1, 10)// read 10 bytes
	
	// if err is equal to 0, it is successful to execute GetDataEx()
	GetErr(err)// save an error code to err
	
	end macro_command
	
	Error code:
	0: Normal
	1: GetDataEx error
	2: SetDataEx error

Name	SetData
Syntax	SetData(*send_data*[*start*], *device_name, device_type, address_offset, data_count*)

	or SetData(*send_data, device_name, device_type, address_offset, 1*)
Description	Sends data to the device. Data is defined in *send_data*[*start*]~ *send_data*[*start + data_count - 1*]. *data_count* is the amount of sent data. In general, *send_data* is an array, but if *data_count* is 1, *send_data* can be an array or an ordinary variable. Below are two methods to send one word data. macro_command main() short send_data_1[2] = { 5, 6}, send_data_2 = 5 SetData(send_data_1[0], "FATEK KB Series", RT, 5, 1) SetData(send_data_2, "FATEK KB Series", RT, 5, 1) end macro_command *device_name* is the device name enclosed in the double quotation marks (") and this name has been defined in the device list of system parameters. *device_type* is the device type and encoding method (binary or BCD) of the device data. For example, if *device_type* is LW_BIN, it means the register is LW and the encoding method is binary. If use BIN encoding method, "_BIN" can be ignored. If *device_type* is LW_BCD, it means the register is LW and the encoding method is BCD. *address_offset* is the address offset in the device. For example, SetData(read_data_1[0], "FATEK KB Series", RT, 5, 1) represents that the address offset is 5. If *address_offset* uses the format –"N#AAAAA", N indicates that device's station number is N. AAAAA represents the address offset. This format is used while multiple devices or controllers are connected to a single serial port. For example, SetData(read_data_1[0], "FATEK KB Series", RT, 2#5, 1) represents that the device's station number is 2. If SetData () uses the default station number defined in the device list, it is not necessary to define station number in *address_offset*. The number of registers actually sends to depends on both the type of the *send_data* variable and the value of the number of *data_count.*

type of *read_data*	*data_count*	actual number of 16-bit register send
char (8-bit)	1	1
char (8-bit)	2	1
bool (8-bit)	1	1
bool (8-bit)	2	1
short (16-bit)	1	1
short (16-bit)	2	2
int (32-bit)	1	2
int (32-bit)	2	4

float (32-bit)	1	2
float (32-bit)	2	4

When a SetData() is executed using a 32-bit data type (int or float), the function will automatically send int-format or float-format data to the device. For example,

```
macro_command main()
float f = 2.6
SetData(f, "MODBUS", 6x, 2, 1)      // will send a floating point value to the device
end macro_command
```

Example

```
macro_command main()
int i
bool a = true
bool b[30]
short c = false
short d[50]
int e = 5
int f[10]

for i = 0 to 29
b[i] = true
next i

for i = 0 to 49
d[i] = i * 2
next i

for i = 0 to 9
f [i] = i * 3
next i

//  set the state of LB2
SetData(a, "Local HMI", LB, 2, 1)

//    set the states of LB0 ~ LB29
SetData(b[0], "Local HMI", LB, 0, 30)

//    set the value of LW-2
SetData(c, "Local HMI", LW, 2, 1)

//    set the values of LW-0 ~ LW-49
SetData(d[0], "Local HMI", LW, 0, 50)

//    set the values of LW-6 ~ LW-7,    note that the type of e is int
SetData(e, "Local HMI", LW, 6, 1)
```

	// set the values of LW-0 ~ LW-19 // 10 integers equal to 20 words, since each integer value occupies 2 words. SetData(f[0], "Local HMI", LW, 0, 10) end macro_command

Name	SetDataEx
Syntax	SetDataEx (*send_data*[*start*], *device_name, device_type, address_offset, data_count*) or SetDataEx (*send_data, device_name, device_type, address_offset, 1*)
Description	Sends data to the device and continues executing next command even when the write operation fails. Descriptions of *send_data, device_name, device_type, address_offset* and *data_count* are the same as SetData.

Example	macro_command main()
	int i
	bool a = true
	bool b[30]
	short c = false
	short d[50]
	int e = 5
	int f[10]
	
	for i = 0 to 29
	b[i] = true
	next i
	
	for i = 0 to 49
	d[i] = i * 2
	next i
	
	for i = 0 to 9
	f [i] = i * 3
	next i
	
	// set the state of LB2
	SetDataEx (a, "Local HMI", LB, 2, 1)
	
	// set the states of LB0 ~ LB29
	SetDataEx (b[0], "Local HMI", LB, 0, 30)
	
	// set the value of LW-2
	SetDataEx (c, "Local HMI", LW, 2, 1)

	// set the values of LW-0 ~ LW-49
	SetDataEx (d[0], "Local HMI", LW, 0, 50)
	
	// set the values of LW-6 ~ LW-7, note that the type of e is int
	SetDataEx (e, "Local HMI", LW, 6, 1)
	
	// set the values of LW-0 ~ LW-19
	// 10 integers equal to 20 words, since each integer value occupies 2 words.
	SetDataEx (f[0], "Local HMI", LW, 0, 10)
	
	end macro_command

18.7.3. Free Protocol

Name	GetCTS
Syntax	GetCTS(*com_port, result*)
Description	Gets CTS state for RS232. *com_port* refers to the COM port number. It can be either a variable or a constant. *result* is used for receiving the CTS signal. It must be a variable. This command receives CTS signal and stores the received data in the *result* variable. When the CTS signal is pulled high, it writes 1 to *result*, otherwise, it writes 0.

Example	macro_command main()
	char com_port=1
	char result
	
	GetCTS(com_port, result) // get CTS signal of COM1
	
	GetCTS (1, result) // get CTS signal of COM1
	
	end macro_command

Name	INPORT
Syntax	INPORT(*read_data*[*start*], *device_name, read_count, return_value*)
Description	Reads data from a COM port or the Ethernet port. The data is stored to *read_data*[*start*]~ *read_data*[*start + read_count - 1*]. *device_name* is the name of a device defined in the device table and the device must be a “Free Protocol”-type device. *read_count* is the required amount of reading and can be a constant or a variable. If the function is used successfully to get sufficient data, *return_value* is 1, otherwise is 0.

Example	Below is an example of executing an action of reading holding registers of a MODBUS device.
	
	// Read Holding Registers
	macro_command main()
	char command[32], response[32]
	short address, checksum
	short read_no, return_value, read_data[2]
	
	FILL(command[0], 0, 32)// command initialization
	FILL(response[0], 0, 32)
	
	command[0] = 0x1// station no
	command[1] = 0x3// function code : Read Holding Registers

	address = 0
	HIBYTE(address, command[2])
	LOBYTE(address, command[3])
	
	read_no = 2// read 2 words (4x_1 and 4x_2)
	HIBYTE(read_no, command[4])
	LOBYTE(read_no, command[5])
	
	CRC(command[0], checksum, 6)
	
	LOBYTE(checksum, command[6])
	HIBYTE(checksum, command[7])
	
	// send out a 'Read Holding Registers" command
	OUTPORT(command[0], "MODBUS RTU Device", 8)
	
	// read responses for a 'Read Holding Registers" command
	INPORT(response[0], "MODBUS RTU Device", 9, return_value)
	
	if return_value > 0 then
	read_data[0] = response[4] + (response[3] << 8)// data in 4x_1
	read_data[1] = response[6] + (response[5] << 8)// data in 4x_2
	
	SetData(read_data[0], "Local HMI", LW, 100, 2)
	end if
	
	end macro_command

Name	INPORT2
Syntax	INPORT2(*response*[*start*], *device_name, receive_len, wait_time*)
Description	Reads data from a COM port or the Ethernet port. The data read will be saved in the response array. *device_name* is the name of a device defined in the device table and the device must be a "Free Protocol"-type device. *receive_len* stores the length of the data received. It must be a variable. *receive_len* can't exceed the size of response array. *wait_time* (in millisecond) can be a constant or variable. After the data is read, if there's no upcoming data during the designated time interval, the function returns.
Example	macro_command main() short wResponse[6], receive_len, wait_time=20 INPORT2(wResponse[0], "Free Protocol", receive_len, wait_time) // wait_time unit : millisecond

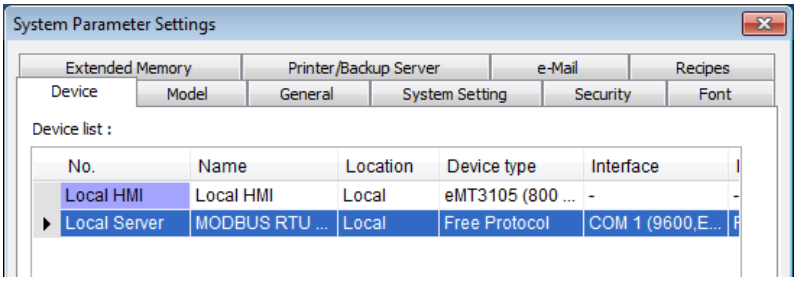
	if receive_len > 0 then
	SetData(wResponse[0], "Local HMI", LW, 0, 6)
	// set responses to LW0
	end if
	
	end macro_command

Name	INPORT3
Syntax	INPORT3(*response*[*start*], *device_name, read_count, receive_len*)
Description	Reads data from a communication port (COM Port or Ethernet Port). The data read will be saved in the response array. The amount of data to be read can be specified. The data that is not read yet will be stored in HMI buffer memory for the next read operation, in order to prevent losing data. *device_name* is the name of a device defined in the device table and the device must be a “Free Protocol”-type device. *read_count* stores the length of the data read each time. *receive_len* stores the length of the data received. It must be a variable. *receive_len* can’t exceed the size of response array.

Example	macro_command main()
	
	short wResponse[6], receive_len
	
	INPORT3(wResponse[0], "Free Protocol", 6, receive_len) // read 6 words
	
	if receive_len >= 6 then
	
	SetData(wResponse[0], "Local HMI", LW, 0, 6) // set responses to LW0
	
	end if
	
	end macro_command

Name	INPORT4
Syntax	INPORT4(*response*[*start*], *device_name, receive_len, tail_ascii*)
Description	Reads data from a communication port (COM Port or Ethernet Port). The data read will be saved in the response array. tail_ascii specifies the ending character. Data reading will stop when the ending character is reached. *device_name* is the name of a device defined in the device table and the device must be a “Free Protocol”-type device. *receive_len* stores the length of the data received. It must be a variable. *receive_len* can’t exceed the size of response array.
Example	macro_command main()

```
char tail_ascii = 0x03// == ETX

short wResponse[1024], receive_len

INPORT4(wResponse[0], "Free Protocol", receive_len, 0x0d)//    0x0d == CR

INPORT4(wResponse[0], "Free Protocol", receive_len, tail_ascii)

if receive_len >= 6 then

SetData(wResponse[0], "Local HMI", LW, 0, 6)//    set responses to LW0

end if
end macro_command
```

Name	OUTPORT
Syntax	OUTPORT(*source*[*start*], *device_name, data_count*)
Description	Sends out the specified data from source[*start*] to source[*start + data_count -1*] to the device via a COM port or an Ethernet port. *device_name* is the name of a device defined in the device table and the device must be a “Free Protocol”-type device. *data_count* is the amount of sent data and can be a constant or a variable.
Example	To use an OUTPORT function, a “Free Protocol” device must be created first as follows:  The device is named ”MODBUS RTU Device”. The port attribute depends on the setting of this device. (the current setting is “19200,E, 8, 1”) Below is an example of executing an action of writing single coil (SET ON) to a MODBUS device. macro_command main() char command[32] short address, checksum FILL(command[0], 0, 32)// command initialization command[0] = 0x1// station no command[1] = 0x5// function code : Write Single Coil

	address = 0 HIBYTE(address, command[2]) LOBYTE(address, command[3]) command[4] = 0xff// force bit on command[5] = 0 CRC(command[0], checksum, 6) LOBYTE(checksum, command[6]) HIBYTE(checksum, command[7]) // send out a “Write Single Coil” command OUTPORT(command[0], "MODBUS RTU Device", 8) end macro_command

Name	PURGE
Syntax	PURGE (*com_port*)
Description	*com_port* refers to the COM port number which ranges from 1 to 3. It can be either a variable or a constant. This function is used to clear the input and output buffers associated with the COM port.
Example	macro_command main() int com_port=3 PURGE (com_port) PURGE (1) end macro_command

Name	SetRTS
Syntax	SetRTS(*com_port, source*)
Description	Sets RTS state for RS232. *com_port* refers to the COM port number. It can be either a variable or a constant. *source* can be either a variable or a constant. This command raise RTS signal while the value of source is greater than 0 and lower RTS signal while the value of *source* equals to 0.

Example	macro_command main()
	char com_port=1
	char value=1
	
	SetRTS(com_port, value) // raise RTS signal of COM1 while value>0
	
	SetRTS(1, 0) // lower RTS signal of COM1
	
	end macro_command

18.7.4. Process Control

Name	ASYNC_TRIG_MACRO
Syntax	ASYNC_TRIG_MACRO (*macro_id or name*)
Description	Triggers the execution of a macro asynchronously (use *macro_id or macro name* to designate this macro) in a running macro. The current macro will continue executing the following instructions after triggering the designated macro; in other words, the two macros will be active simultaneously. *macro_id* can be a constant or a variable.

Example	macro_command main()
	char ON = 1, OFF = 0
	
	SetData(ON, "Local HMI", LB, 0, 1)
	
	ASYNC_TRIG_MACRO(5)// call a macro (its ID is 5)
	
	ASYNC_TRIG_MACRO("macro_1") // call a macro (its name is macro_1)
	
	SetData(OFF, "Local HMI", LB, 0, 1)
	
	end macro_command

Name	DELAY
Syntax	DELAY(*time*)
Description	Suspends the execution of the current macro for at least the specified interval (*time*). The unit of *time* is millisecond. *time* can be a constant or a variable.

Example	macro_command main()
	int time == 500
	
	DELAY(100)// delay 100 ms
	DELAY(time)// delay 500 ms
	
	end macro_command

Name	SYNC_TRIG_MACRO
Syntax	SYNC_TRIG_MACRO(*macro_id or name*)
Description	Triggers the execution of a macro synchronously (use *macro_id* or macro name to designate this macro) in a running macro. The current macro will pause until the end of execution of this called macro. *macro_id* can be a constant or a variable.
Example	macro_command main() char ON = 1, OFF = 0

	SetData(ON, "Local HMI", LB, 0, 1)
	
	SYNC_TRIG_MACRO(5) // call a macro (its ID is 5)
	
	SYNC_TRIG_MACRO("macro_1") // call a macro (its name is macro_1)
	
	SetData(OFF, "Local HMI", LB, 0, 1)
	
	end macro_command

18.7.5. Data Operation

Name	FILL
Syntax	FILL(*source[start], preset, count*)
Description	Sets array elements from '`source[start]`' to '`source[start + count – 1]`' to the specified value (`preset`). *source* and *start* must be a variable, and *preset* can be a constant or variable.
Example	macro_command main() char result[4] char preset FILL(result[0], 0x30, 4) // result[0] is 0x30, result[1] is 0x30, , result[2] is 0x30, , result[3] is 0x30 preset = 0x31 FILL(result[0], preset, 2) // result[0] is 0x31, result[1] is 0x31 end macro_command

Name	SWAPB
Syntax	SWAPB(*source, result*)
Description	Exchanges the high-byte and low-byte data of a 16-bit *source* into *result*. *source* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	short source, result
	
	SWAPB(0x5678, result)// result is 0x7856
	
	source = 0x123
	SWAPB(source, result)// result is 0x2301
	
	end macro_command

Name	SWAPW
Syntax	SWAPW(*source, result*)
Description	Exchanges the high-word and low-word data of a 32-bit *source* into *result*. *source* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	int source, result
	
	SWAPW (0x12345678, result)// result is 0x56781234
	
	source = 0x12345
	SWAPW (source, result)// result is 0x23450001
	
	end macro_command

Name	LOBYTE
Syntax	LOBYTE(*source, result*)
Description	Retrieves the low byte of a 16-bit *source* into *result*. *source* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	short source, result
	
	LOBYTE(0x1234, result)// result is 0x34
	
	source = 0x123
	LOBYTE(source, result)// result is 0x23
	
	end macro_command

Name	HIBYTE
Syntax	HIBYTE(*source, result*)
Description	Retrieves the high byte of a 16-bit *source* into *result*. *source* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	short source, result
	
	HIBYTE(0x1234, result)// result is 0x12
	
	source = 0x123
	HIBYTE(source, result)// result is 0x01
	
	end macro_command

Name	LOWORD
Syntax	LOWORD(*source, result*)
Description	Retrieves the low word of a 32-bit source into result. *source* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	int source, result
	
	LOWORD(0x12345678, result)// result is 0x5678
	
	source = 0x12345
	LOWORD(source, result)// result is 0x2345
	
	end macro_command

Name	HIWORD
Syntax	HIWORD(*source, result*)
Description	Retrieves the high word of a 32-bit source into result. *source* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	int source, result
	
	HIWORD(0x12345678, result)// result is 0x1234
	
	source = 0x12345
	HIWORD(source, result)// result is 0x0001
	
	end macro_command

Name	INVBIT
Syntax	INVBIT(*source, result, bit_pos*)
Description	Inverts the state of designated bit position of a data (*source*), and puts changed data into *result*. *source* and *bit_pos* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	int source, result
	short bit_pos
	
	INVBIT(4, result, 1)// result = 6
	
	source = 6
	bit_pos = 1
	INVBIT(source, result, bit_pos)// result = 4

	end macro_command

Name	SETBITON
Syntax	SETBITON(*source, result, bit_pos*)
Description	Changes the state of designated bit position of a data (*source*) to 1, and puts changed data into *result*. *source* and *bit_pos* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	int source, result
	short bit_pos
	
	SETBITON(1, result, 3)// result is 9
	
	source = 0
	bit_pos = 2
	SETBITON (source, result, bit_pos)// result is 4
	
	end macro_command

Name	SETBITOFF
Syntax	SETBITOFF(*source, result, bit_pos*)
Description	Changes the state of designated bit position of a data (*source*) to 0, and puts changed data into *result*. *source* and *bit_pos* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	int source, result
	short bit_pos
	
	SETBITOFF(9, result, 3)// result is 1
	
	source = 4
	bit_pos = 2
	SETBITOFF(source, result, bit_pos)// result is 0
	
	end macro_command

Name	GETBIT
Syntax	GETBIT(*source, result, bit_pos*)
Description	Gets the state of designated bit position of a data (*source*) into *result*. *result* value will be 0 or 1. *source* and *bit_pos* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	int source, result
	short bit_pos
	
	GETBIT(9, result, 3)// result is 1
	
	source = 4
	bit_pos = 2
	GETBIT(source, result, bit_pos)// result is 1
	
	end macro_command

18.7.6. Data Type Conversion

Name	ASCII2DEC
Syntax	ASCII2DEC(*source[start], result, len*)
Description	Transforms a string (*source*) into a decimal value saved to a variable (*result*). The length of the string is *len*. The first character of the string is *source[start]*. *source* and *len* can be a constant or a variable. *result* must be a variable. *start* must be a constant.
Example	macro_command main() char source[4] short result source[0] = '5' source[1] = '6' source[2] = '7' source[3] = '8' ASCII2DEC(source[0], result, 4) // result is 5678 end macro_command

Name	ASCII2FLOAT
Syntax	ASCII2FLOAT(*source[start], result, len*)
Description	Transforms a string (*source*) into a float value saved to a variable (*result*). The length of the string is *len*. The first character of the string is *source[start]*. *source* and *len* can be a constant or a variable. *result* must be a variable. *start* must be a constant.
Example	macro_command main() char source[4] float result source[0] = '5' source[1] = '6' source[2] = '.' source[3] = '8' ASCII2FLOAT (source[0], result, 4) // result is 56.8 end macro_command

Name	ASCII2HEX

Syntax	ASCII2HEX (*source[start], result, len*)
Description	Transforms a string (*source*) into a hexadecimal value saved to a variable (*result*). The length of the string is *len*. The first character of the string is *source[start]*. *source* and *len* can be a constant or a variable. *result* must be a variable. *start* must be a constant.
Example	macro_command main() char source[4] short result source[0] = '5' source[1] = '6' source[2] = '7' source[3] = '8' ASCII2HEX (source[0], result, 4) // result is 0x5678 end macro_command

Name	BIN2BCD
Syntax	BIN2BCD(*source, result*)
Description	Transforms a binary-type value (*source*) into a BCD-type value (*result*). *source* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	
	short source, result
	
	BIN2BCD(1234, result)// result is 0x1234
	
	source = 5678
	BIN2BCD(source, result)// result is 0x5678
	
	end macro_command

Name	BCD2BIN
Syntax	BCD2BIN(*source, result*)
Description	Transforms a BCD-type value (*source*) into a binary-type value (*result*). source can be a constant or a variable. result must be a variable.
Example	macro_command main() short source, result BCD2BIN(0x1234, result)// result is 1234

	source = 0x5678 BCD2BIN(source, result)// result is 5678 end macro_command

Name	DATE2ASCII
Syntax	DATE2ASCII(*day_offset, date[start], count, [separator]*)
Description	Transforms a date with *day_offset* added into an ASCII string, and saves it to an array (*date*). *count* represents the length of the string and the unit of length depends on result's type. *separator* separates year, month, and day. By default, the separator is "/". *day_offset* and *count* can be a constant or a variable. *start* and *separator* must be a constant.
Example	macro_command main() char result[10] DATE2ASCII(5, result[0], 10) // result[0]~[9] == "2019/02/16"// today is 2019/02/11 DATE2ASCII(5, result[0], 10,2019/02/16"// today is 2019/02/11-16"// today is 2019/02/11 end macro_command

Name	DATE2DEC
Syntax	DATE2DEC(*day_offset, date*)
Description	Transforms a date with *day_offset* added into a decimal value saved to a variable (*date*). *day_offset* can be a constant or a variable. *date* must be a variable.

Example	macro_command main()
	int day_offset = 5, date
	
	DATE2DEC(0, date) // date == 20190211 (Today is 2019/02/11)
	DATE2DEC(day_offset, date) // date == 20190216 (20190211 + 5)
	
	end macro_command

Name	DEC2ASCII
Syntax	DEC2ASCII(*source, result[start], len*)
Description	Transforms a decimal value (*source*) into an ASCII string and saves it to an

	array (*result*). *len* represents the length of the string and the unit of length depends on result's type., i.e. if result's type is "char" (the size is byte), the length of the string is (byte * *len*). If result's type is "short" (the size is word), the length of the string is (word * *len*), and so on. The first character is put into *result[start]*, the second character is put into *result[start + 1]*, and the last character is put into *result[start + (len -1)]*. *source* and *len* can be a constant or a variable. *result* must be a variable. *start* must be a constant.

Example	macro_command main()
	short source
	char result1[4]
	short result2[4]
	char result3[6]
	source = 5678
	
	DEC2ASCII(source, result1[0], 4)
	// result1[0] is '5', result1[1] is '6', result1[2] is '7', result1[3] is '8'
	// the length of the string (result1) is 4 bytes(= 1 * 4)
	DEC2ASCII(source, result2[0], 4)
	// result2[0] is '5', result2[1] is '6', result2[2] is '7', result2[3] is '8'
	// the length of the string (result2) is 8 bytes(= 2 * 4)
	
	source=-123
	DEC2ASCII(source, result3[0], 6)
	// result1[0] is '-', result1[1] is '0', result1[2] is '0', result1[3] is '1'
	// result1[4] is '2', result1[5] is '3'
	// the length of the string (result1) is 6 bytes(= 1 * 6)
	
	end macro_command

Name	FLOAT2ASCII
Syntax	FLOAT2ASCII(*source, result[start], len*)
Description	Transforms a floating value (*source*) into ASCII string saved to an array (*result*). *len* represents the length of the string and the unit of length depends on result's type., i.e. if result's type is "char" (the size is byte), the length of the string is (byte * *len*). If result's type is "short" (the size is word), the length of the string is (word * *len*), and so on. *source* and len can be a constant or a variable. *result* must be a variable. start must be a constant.
Example	macro_command main() float source char result[4]

	source = 56.8 FLOAT2ASCII (source, result[0], 4) // result[0] is '5', result[1] is '6', result[2] is '.', result[3] is '8' end macro_command

Name	HEX2ASCII
Syntax	HEX2ASCII(*source, result[start], len*)
Description	Transforms a hexadecimal value (*source*) into ASCII string saved to an array (*result*). *len* represents the length of the string and the unit of length depends on result's type., i.e. if result's type is "char" (the size is byte), the length of the string is (byte * *len*). If result's type is "short" (the size is word), the length of the string is (word * *len*), and so on. *source* and *len* can be a constant or a variable. *result* must be a variable. *start* must be a constant.
Example	macro_command main() short source char result[4] source = 0x5678 HEX2ASCII (source, result[0], 4) // result[0] is '5', result[1] is '6', result[2] is '7', result[3] is '8' end macro_command

Name	StringDecAsc2Bin
Syntax	success = StringDecAsc2Bin(source[start], destination) or success = StringDecAsc2Bin("source", destination)
Description	This function converts a decimal string to an integer. It converts the decimal string in source parameter into an integer, and stores it in the destination variable. The source string parameter accepts both static string (in the form: "source") and char array (in the form: source[start]). Destination must be a variable, to store the result of conversion. This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. The string can only contain these characters: +, -, and 0 to 9. If the string contains other characters, it returns false. The success field is optional.
Example	macro_command main() char src1[5]="12345" int result1 bool success1

	success1 = StringDecAsc2Bin(src1[0], result1)
	// success1=true, result1 is 12345
	
	char src2[5] = "-6789"
	short result2
	bool success2
	success2 = StringDecAsc2Bin(src2[0], result2)
	// success2 = true，result2 is -6789
	
	char result3
	bool success3
	success3 = StringDecAsc2Bin("32768", result3)
	// success3=true, but the result exceeds the data range of result3
	
	char src4[2]="4b"
	char result4
	bool success4
	success4 = StringDecAsc2Bin (src4[0], result4)
	// success4=false, because src4 contains characters other than '+' or '-' and '0' to '9'
	
	end macro_command

Name	StringBin2DecAsc
Syntax	success = StringBin2DecAsc (source, destination[start])
Description	This function converts an integer to a decimal string. It converts the integer in source parameter into a decimal string, and stores it in the destination buffer. Source can be either a constant or a variable. Destination must be an one-dimensional char array, to store the result of conversion. This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of decimal string after conversion exceeds the size of destination buffer, it returns false. The success field is optional.
Example	macro_command main() int src1 = 2147483647 char dest1[20] bool success1 success1 = StringBin2DecAsc(src1, dest1[0]) // success1=true, dest1="2147483647" short src2 = 0x3c char dest2[20] bool success2

	success2 = StringBin2DecAsc(src2, dest2[0])
	// success2=true, dest2="60"
	
	int src3 = 2147483647
	char dest3[5]
	bool success3
	success3 = StringBin2DecAsc(src3, dest3[0])
	// success3=false, dest3 remains the same.
	
	end macro_command

Name	StringDecAsc2Float
Syntax	success = StringDecAsc2Float (source[start], destination) or success = StringDecAsc2Float ("source", destination)
Description	This function converts a decimal string to floats. It converts the decimal string in source parameter into float, and stores it in the destination variable. The source string parameter accepts both static string (in the form: "source") and char array (in the form: source[start]). Destination must be a variable, to store the result of conversion. This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the source string contains characters other than '0' to '9' or '.', it returns false. The success field is optional.

Example	macro_command main()
	char src1[10]="12.345"
	float result1
	bool success1
	success1 = StringDecAsc2Float(src1[0], result1)
	// success1=true, result1 is 12.345
	
	float result2
	bool success2
	success2 = StringDecAsc2Float("1.234567890", result2)
	// success2=true, but the result exceeds the data range of result2, which
	// might result in loss of precision
	
	char src3[2]="4b"
	float result3
	bool success3
	success3 = StringDecAsc2Float(src3[0], result3)
	// success3=false, because src3 contains characters other than '0' to '9' or
	// '.'
	end macro_command

Name	StringFloat2DecAsc
Syntax	success = StringFloat2DecAsc(source, destination[start])
Description	This function converts a float to a decimal string. It converts the float in source parameter into a decimal string, and stores it in the destination buffer. Source can be either a constant or a variable. Destination must be an one-dimensional char array, to store the result of conversion. This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of decimal string after conversion exceeds the size of destination buffer, it returns false. The success field is optional.

Example	macro_command main()
	float src1 = 1.2345
	char dest1[20]
	bool success1
	success1 = StringFloat2DecAsc(src1, dest1[0])
	// success1=true, dest1="1.2345"
	
	float src2 = 1.23456789
	char dest2 [20]
	bool success2
	success2 = StringFloat2DecAsc(src2, dest2 [0])
	// success2=true, but it might lose precision
	
	float src3 = 1.2345
	char dest3[5]
	bool success3
	success3 = StringFloat2DecAsc(src3, dest3 [0])
	// success3=false, dest3 remains the same.
	
	end macro_command

Name	StringHexAsc2Bin
Syntax	success = StringHexAsc2Bin (source[start], destination) or success = StringHexAsc2Bin ("source", destination)
Description	This function converts a hexadecimal string to binary data. It converts the hexadecimal string in source parameter into binary data, and stores it in the destination variable. The source string parameter accepts both static string (in the form: "source") and char array (in the form: source[start]). Destination must be a variable, to store the result of conversion. This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the

	source string contains characters other than ‘0’ to ‘9’, ‘a’ to ‘f’ or ‘A’ to ‘F’, it returns false. The success field is optional.

Example	macro_command main()
	char src1[5]="0x3c"
	int result1
	bool success1
	success1 = StringHexAsc2Bin(src1[0], result1)
	// success1=true, result1 is 3c
	
	short result2
	bool success2
	success2 = StringDecAsc2Bin("1a2b3c4d", result2)
	// success2=true, result2=3c4d.The result exceeds the data range of
	// result2
	
	char src3[2]="4g"
	char result3
	bool success3
	success3 = StringDecAsc2Bin (src3[0], result3)
	// success3=false, because src3 contains characters other than ‘0’ to ‘9’
	// , ‘a’ to ‘f’ or ‘A’ to ‘F’
	
	end macro_command

Name	StringBin2HexAsc
Syntax	success = StringBin2HexAsc (source, destination[start])
Description	This function converts binary data to a hexadecimal string. It converts the binary data in source parameter into a hexadecimal string, and stores it in the destination buffer. Source can be either a constant or a variable. Destination must be an one-dimensional char array, to store the result of conversion. This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of hexadecimal string after conversion exceeds the size of destination buffer, it returns false. The success field is optional. Please note that this function cannot convert negative values.
Example	macro_command main() int src1 = 20 char dest1[20] bool success1 success1 = StringBin2HexAsc(src1, dest1[0]) // success1=true, dest1="14"

	short src2 = 0x3c
	char dest2[20]
	bool success2
	success2 = StringBin2HexAsc(src2, dest2[0])
	// success2=true, dest2="3c"
	
	int src3 = 0x1a2b3c4d
	char dest3[6]
	bool success3
	success3 = StringBin2HexAsc(src3, dest3[0])
	// success3=false, dest3 remains the same.
	
	end macro_command

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

18.7.7. String Operation

Name	String2Unicode
Syntax	result = String2Unicode("source", destination[start])
Description	Converts all the characters in the source string to Unicode and stores the result in the destination buffer. The length of result string after conversion will be stored to result. Source must be a constant but not a variable.

Example	macro_command main()
	
	char dest[20]
	int result
	result = String2Unicode("abcde", dest[0]) // "result" will be set to 10.
	result = String2Unicode("abcdefghijklmno", dest[0]) // "result" will be set to 20.
	// "result" will be the length of converted Unicode string
	
	end macro_command

Name	StringCat
Syntax	success = StringCat (source[start], destination[start]) or success = StringCat ("source", destination[start])
Description	This function appends source string to destination string. It adds the contents of source string to the last of the contents of destination string. The source string parameter accepts both static string (in the form: "source") and char array (in the form: source[start]). Destination must be an one-dimensional char array.

	This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of result string after concatenation exceeds the max. size of destination buffer, it returns false. The success field is optional.

Example	macro_command main()
	char src1[20]="abcdefghij"
	char dest1[20]="1234567890"
	bool success1
	success1= StringCat(src1[0], dest1[0])
	// success1=true, dest1="123456790abcdefghij"
	
	char dest2 [10]="1234567890"
	bool success2
	success2= StringCat("abcde", dest2 [0])
	// success2=false, dest2 remains the same.
	
	char src3[20]="abcdefghij"
	char dest3[20]
	bool success3
	success3= StringCat(src3[0], dest3[15])
	// success3=false, dest3 remains the same.
	
	end macro_command

Name	StringCompare
Syntax	ret = StringCompare (str1[start], str2[start]) ret = StringCompare ("string1", str2[start]) ret = StringCompare (str1[start], "string2") ret = StringCompare ("string1", "string2")
Description	Performs a case-sensitive comparison of two strings. The two string parameters accept both static string (in the form: "string1") and char array (in the form: str1[start]). This function returns a Boolean indicating the result of comparison. If two strings are identical, it returns true. Otherwise it returns false. The ret field is optional.
Example	macro_command main() char a1[20]="abcde" char b1[20]="ABCDE" bool ret1 ret1= StringCompare(a1[0], b1[0]) // ret1=false char a2[20]="abcde" char b2[20]="abcde"

	bool ret2
	ret2= StringCompare(a2[0], b2[0])
	// ret2=true
	
	char a3 [20]="abcde"
	char b3[20]="abcdefg"
	bool ret3
	ret3= StringCompare(a3[0], b3[0])
	// ret3=false
	
	end macro_command

Name	StringCompareNoCase
Syntax	ret = StringCompareNoCase(str1[start], str2[start]) ret = StringCompareNoCase("string1", str2[start]) ret = StringCompareNoCase(str1[start], "string2") ret = StringCompareNoCase("string1", "string2")
Description	Performs a case-insensitive comparison of two strings. The two string parameters accept both static string (in the form: "string1") and char array (in the form: str1[start]). This function returns a Boolean indicating the result of comparison. If two strings are identical, it returns true. Otherwise it returns false. The ret field is optional.

Example	macro_command main()
	char a1[20]="abcde"
	char b1[20]="ABCDE"
	bool ret1
	ret1= StringCompareNoCase(a1[0], b1[0])
	// ret1=true
	
	char a2[20]="abcde"
	char b2[20]="abcde"
	bool ret2
	ret2= StringCompareNoCase(a2[0], b2[0])
	// ret2=true
	
	char a3 [20]="abcde"
	char b3[20]="abcdefg"
	bool ret3
	ret3= StringCompareNoCase(a3[0], b3[0])
	// ret3=false
	
	end macro_command

Name	StringCopy

Syntax	success = StringCopy ("source", destination[start]) or success = StringCopy (source[start], destination[start])
Description	Copies one string to another. This function copies a static string (which is enclosed in quotes) or a string that is stored in an array to the destination buffer. The source string parameter accepts both static string (in the form: "source") and char array (in the form: source[start]). destination[start] must be an one-dimensional char array. This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of source string exceeds the max. size of destination buffer, it returns false and the content of destination remains the same. The success field is optional.

Example	macro_command main()
	char src1[5]="abcde"
	char dest1[5]
	bool success1
	success1 = StringCopy(src1[0], dest1[0])
	// success1=true, dest1="abcde"
	
	char dest2[5]
	bool success2
	success2 = StringCopy("12345", dest2[0])
	// success2=true, dest2="12345"
	
	char src3[10]="abcdefghij"
	char dest3[5]
	bool success3
	success3 = StringCopy(src3[0], dest3[0])
	// success3=false, dest3 remains the same.
	
	char src4[10]="abcdefghij"
	char dest4[5]
	bool success4
	success4 = StringCopy(src4[5], dest4[0])
	// success4=true, dest4="fghij"
	
	end macro_command

Name	StringExcluding
Syntax	success = StringExcluding (source[start], set[start], destination[start]) success = StringExcluding ("source", set[start], destination[start]) success = StringExcluding (source[start], "set", destination[start]) success = StringExcluding ("source", "set", destination[start])
Description	Retrieves a substring of the source string that contains characters that are not in the set string, beginning with the first character in the source string and ending when a character is found in the source string that is also in the target string. The source string and set string parameters accept both static string (in the form: "source") and char array (in the form: source[start]). This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of retrieved substring exceeds the size of destination buffer, it returns false.

Example	macro_command main()
	char src1[20]="cabbageabc"
	char set1[20]="ge"
	char dest1[20]
	bool success1
	success1 = StringExcluding(src1[0], set1[0], dest1[0])
	// success1=true, dest1="cabba"
	
	char src2[20]="cabbage"
	char dest2[20]
	bool success2
	success2 = StringExcluding(src2[0], "abc", dest2[0])
	// success2=true, dest2=""
	
	char set3[20]="ge"
	char dest3[4]
	bool success3
	success3 = StringExcluding("cabbage", set3[0], dest3[0])
	// success3=false, dest3 remains the same.
	
	end macro_command

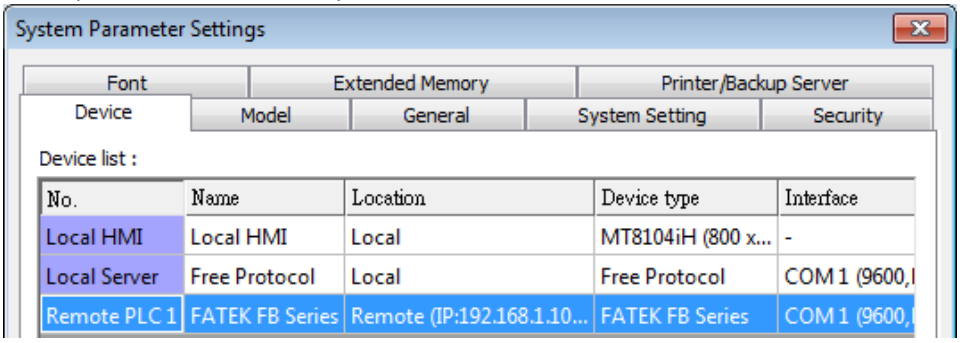
Name	StringFind
Syntax	position = StringFind (source[start], target[start]) position = StringFind ("source", target[start]) position = StringFind (source[start], "target") position = StringFind ("source", "target")
Description	Returns the position of the first occurrence of target string in the source string. The two string parameters accept both static string (in the form: “source”) and

	char array (in the form: source[start]). This function returns the zero-based index of the first character of substring in the source string that matches the target string. Notice that the entire sequence of characters to find must be matched. If there is no matched substring, it returns -1.

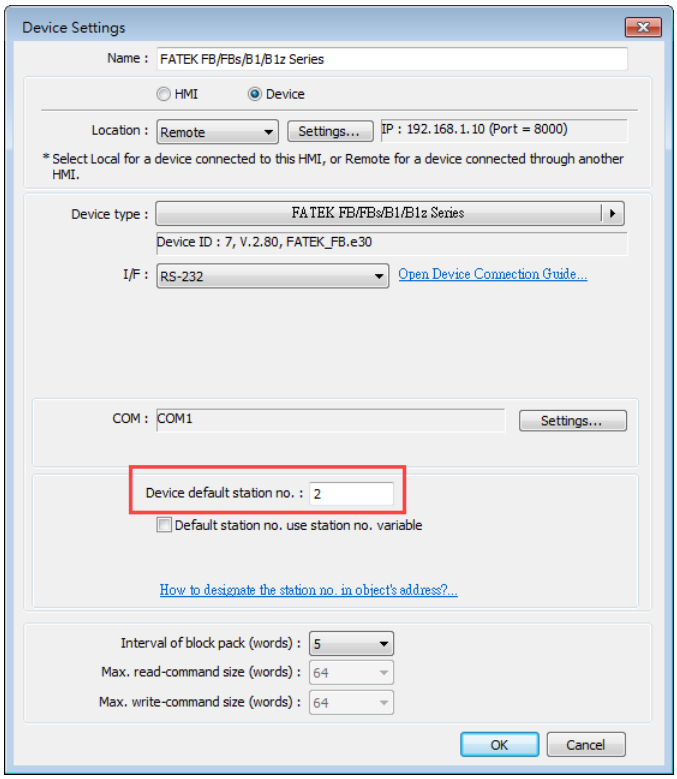
Example	macro_command main()
	char src1[20]="abcde"
	char target1[20]="cd"
	short pos1
	pos1= StringFind(src1[0], target1[0])
	// pos1=2
	
	char target2[20]="ce"
	short pos2
	pos2= StringFind("abcde", target2[0])
	// pos2=-1
	
	char src3[20]="abcde"
	short pos3
	pos3= StringFind(src3[3], "cd")
	// pos3=-1
	
	end macro_command

Name	StringFindOneOf
Syntax	position = StringFindOneOf (source[start], target[start]) position = StringFindOneOf ("source", target[start]) position = StringFindOneOf (source[start], "target") position = StringFindOneOf ("source", "target")
Description	Returns the position of the first character in the source string that matches any character contained in the target string. The two string parameters accept both static string (in the form: "source") and char array (in the form: source[start]). This function returns the zero-based index of the first character in the source string that is also in the target string. If there is no match, it returns -1.

Example	macro_command main()
	char src1[20]="abcdeabcde"
	char target1[20]="sdf"
	short pos1
	pos1= StringFindOneOf(src1[0], target1[0])
	// pos1=3
	
	char src2[20]="abcdeabcde"
	short pos2
	pos2= StringFindOneOf(src2[1], "agi")
	// pos2=4
	
	char target3 [20]="bus"
	short pos3
	pos3= StringFindOneOf("abcdeabcde", target3[1])
	// pos3=-1
	
	end macro_command

Name	StringGet
Syntax	StringGet(*read_data*[*start*], *device_name, device_type, address_offset, data_count)*
Description	Receives data from the device. The String data is stored into *read_data*[*start*]~ *read_data*[*start + data_count - 1*]. *read_data* must be a one-dimensional char array. *Data_count* is the number of received characters, it can be either a constant or a variable. *Device_name* is the device name enclosed in the double quotation marks (") and this name has been defined in the device list of system parameters as follows (see FATEK KB Series):  Device_type is the device type and encoding method (binary or BCD) of the device data. For example, if device_type is LW_BIN, it means the register is LW and the encoding method is binary. If use BIN encoding method, "_BIN" can be ignored. If device_type is LW_BCD, it means the register is LW and the encoding method is BCD. Address_offset is the address offset in the device.

For example, StringGet(read_data_1[0], "FATEK KB Series", RT, 5, 1) represents that the address offset is 5.
If address_offset uses the format –"N#AAAAA", N indicates that device's station number is N. AAAAA represents the address offset. This format is used while multiple devices or controllers are connected to a single serial port. For example, StringGet(read_data_1[0], "FATEK KB Series", RT, 2#5, 1) represents that the device's station number is 2. If StringGet() uses the default station number defined in the device list as follows, it is not necessary to define station number in address_offset.



The number of registers actually read from depends on the value of the number of *data_count* since that the *read_data* is restricted to char array.

type of read_data	data_count	actual number of 16-bit register read
char (8-bit)	1	1
char (8-bit)	2	1

1 WORD register(16-bit) equals to the size of 2 ASCII characters. According to the above table, reading 2 ASCII characters is actually reading the content of one 16-bit register.

Example

```
macro_command main()
char str1[20]

//    read 10 words from LW-0~LW-9 to the variables str1[0] to str1[19]
//    since that 1 word can store 2 ASCII characters, reading 20 ASCII
//    characters is actually reading 10 words of register
StringGet(str1[0], "Local HMI", LW, 0, 20)
```

	end macro_command

Name	StringGetEx
Syntax	StringGetEx (*read_data*[*start*], *device_name, device_type, address_offset, data_count*)
Description	Receives data from the device and continues executing next command even if there's no response from this device. Descriptions of *read_data, device_name, device_type, address_offset* and *data_count* are the same as GetData.

Example	macro_command main()
	char str1[20]
	short test=0
	
	// macro will continue executing test = 1 even if the MODBUS device is
	// not responding
	StringGetEx(str1[0], "MODBUS RTU", 4x, 0, 20)
	test = 1
	
	// macro will not continue executing test = 2 until MODBUS device responds
	StringGet(str1[0], "MODBUS RTU", 4x, 0, 20)
	test = 2
	
	end macro_command

Name	StringIncluding
Syntax	success = StringIncluding (source[start], set[start], destination[start]) success = StringIncluding ("source", set[start], destination[start]) success = StringIncluding (source[start], "set", destination[start]) success = StringIncluding ("source", "set", destination[start])
Description	Retrieves a substring of the source string that contains characters in the set string, beginning with the first character in the source string and ending when a character is found in the source string that is not in the target string. The source string and set string parameters accept both static string (in the form: "source") and char array (in the form: source[start]). This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of retrieved substring exceeds the size of destination buffer, it returns false.

Example	macro_command main()
	char src1[20]="cabbageabc"
	char set1[20]="abc"
	char dest1[20]
	bool success1
	success1 = StringIncluding(src1[0], set1[0], dest1[0])
	// success1=true, dest1="cabba"
	
	char src2[20]="gecabba"
	char dest2[20]
	bool success2
	success2 = StringIncluding(src2[0], "abc", dest2[0])
	// success2=true, dest2=""
	
	char set3[20]="abc"
	char dest3[4]
	bool success3
	success3 = StringIncluding("cabbage", set3[0], dest3[0])
	// success3=false, dest3 remains the same.
	
	end macro_command

Name	StringInsert
Syntax	success = StringInsert (pos, insert[start], destination[start]) success = StringInsert (pos, "insert", destination[start]) success = StringInsert (pos, insert[start], length, destination[start]) success = StringInsert (pos, "insert", length, destination[start])
Description	Inserts a string in a specific location within the destination string content. The insert location is specified by the pos parameter. The insert string parameter accepts both static string (in the form: “source”) and char array (in the form: source[start]). The number of characters to insert can be specified by the length parameter. This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of string after insertion exceeds the size of destination buffer, it returns false.

Example	macro_command main()
	
	char str1[20]="but the question is"
	char str2[10]=", that is"
	char dest[40]="to be or not to be"
	bool success
	
	success = StringInsert(18, str1[3], 13, dest[0])
	// success=true, dest="to be or not to be the question"
	
	success = StringInsert(18, str2[0], dest[0])
	// success=true, dest="to be or not to be, that is the question"
	
	success = StringInsert(0, "Hamlet:", dest[0])
	// success=false, dest remains the same.
	
	end macro_command

Name	StringLength
Syntax	length = StringLength (source[start]) or length = StringLength ("source")
Description	Obtains the length of a string. It returns the length of source string and stores it in the length field on the left-hand side of '=' operator. The source string parameter accepts both static string (in the form: "source") and char array (in the form: source[start]). The return value of this function indicates the length of the source string.

Example	macro_command main()
	char src1[20]="abcde"
	int length1
	length1= StringLength(src1[0])
	// length1=5
	
	char src2[20]={'a', 'b', 'c', 'd', 'e'}
	int length2
	length2= StringLength(src2[0])
	// length2=5
	
	char src3[20]="abcdefghij"
	int length3
	length3= StringLength(src3 [2])
	// length3=8
	
	end macro_command

Name	StringMD5
Syntax	result = StringMD5(source[start], destination[start]) result = StringMD5("source", destination[start])
Description	Retrieves a string using MD5 Message-Digest algorithm. The source string parameter accepts both static string (in the form: "source") and char array (in the form: source[start]). For source[start], the start offset of the substring is specified by the index value. *destination[start]* must be a one-dimensional char array, to store the retrieved substring. This function returns the length of MD5 string stored in *result.*

Example	macro_command main()
	char source[32] = "password", dest[32]
	int result
	
	result = StringMD5(source[0], dest[0])
	result = StringMD5("password", dest[0]) // "result" will be set to 32.
	// "result" will be the length of MD5 string.
	// dest[] = 5f4dcc3b5aa765d61d8327deb882cf99
	
	end macro_command

Name	StringMid
Syntax	success = StringMid (source[start], count, destination[start]) or success = StringMid ("string", start, count, destination[start])
Description	Retrieves a character sequence from the specified offset of the source string and stores it in the destination buffer. The source string parameter accepts both static string (in the form: "source") and char array (in the form: source[start]). For source[start], the start offset of the substring is specified by the index value. For static source string(“source”), the second parameter(start) specifies the start offset of the substring. The count parameter specifies the length of substring being retrieved. Destination must be an one-dimensional char array, to store the retrieved substring. This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of retrieved substring exceeds the size of destination buffer, it returns false. The success field is optional.
Example	macro_command main() char src1[20]="abcdefghijklmnopqrst" char dest1[20] bool success1 success1 = StringMid(src1[5], 6, dest1[0])

	// success1=true, dest1="fghijk"
	
	char src2[20]="abcdefghijklmnopqrst"
	char dest2[5]
	bool success2
	success2 = StringMid(src2[5], 6, dest2[0])
	// success2=false, dest2 remains the same.
	
	char dest3[20]="12345678901234567890"
	bool success3
	success3 = StringMid("abcdefghijklmnopqrst", 5, 5, dest3[15])
	// success3= true, dest3="123456789012345fghij"
	
	end macro_command

Name	StringReverseFind
Syntax	position = StringReverseFind (source[start], target[start]) position = StringReverseFind ("source", target[start]) position = StringReverseFind (source[start], "target") position = StringReverseFind ("source", "target")
Description	Returns the position of the last occurrence of target string in the source string. The two string parameters accept both static string (in the form: "source") and char array (in the form: source[start]). This function returns the zero-based index of the first character of substring in the source string that matches the target string. Notice that the entire sequence of characters to find must be matched. If there exists multiple substrings that matches the target string, function will return the position of the last matched substring. If there is no matched substring, it returns -1.

Example	macro_command main()
	char src1[20]="abcdeabcde"
	char target1[20]="cd"
	short pos1
	pos1= StringReverseFind(src1[0], target1[0])
	// pos1=7
	
	char target2[20]="ce"
	short pos2
	pos2= StringReverseFind("abcdeabcde", target2[0])
	// pos2=-1
	
	char src3[20]="abcdeabcde"
	short pos3
	pos3= StringReverseFind(src3[6], "ab")
	// pos3=-1
	
	end macro_command

Name	StringSet
Syntax	StringSet(*send_data*[*start*], *device_name, device_type, address_offset, data_count*)
Description	Sends data to the device. Data is defined in *send_data*[*start*]~ *send_data*[*start + data_count - 1*]. send_data must be a one-dimensional char array. *data_count* is the number of sent characters, it can be either a constant or a variable. *device_name* is the device name enclosed in the double quotation marks (") and this name has been defined in the device list of system parameters. *device_type* is the device type and encoding method (binary or BCD) of the device data. For example, if *device_type* is LW_BIN, it means the register is LW and the encoding method is binary. If use BIN encoding method, "_BIN" can be ignored. If *device_type* is LW_BCD, it means the register is LW and the encoding method is BCD. *address_offset* is the address offset in the device. For example, StringSet(read_data_1[0], "FATEK KB Series", RT, 5, 1) represents that the address offset is 5. If *address_offset* uses the format –"N#AAAAA", N indicates that device's station number is N. AAAAA represents the address offset. This format is used while multiple devices or controllers are connected to a single serial port. For example, StringSet(read_data_1[0], "FATEK KB Series", RT, 2#5, 1) represents that the device's station number is 2. If SetData () uses the default station number defined in the device list, it is not necessary to define station number in *address_offset*.

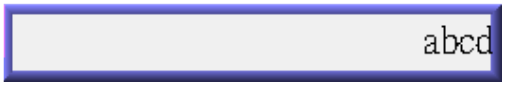
The number of registers actually sends to depends on the value of the number of *data_count*, since that *send_data* is restricted to char array.

type of read_data	data_count	actual number of 16-bit register send
char (8-bit)	1	1
char (8-bit)	2	1

1 WORD register(16-bit) equals to the size of 2 ASCII characters. According to the above table, sending 2 ASCII characters is actually writing to one 16-bit register. The ASCII characters are stored into the WORD register from low byte to high byte. While using the ASCII Display object to display the string data stored in the registers, *data_count* must be a multiple of 2 in order to display full string content. For example:

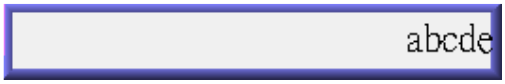
```
macro_command main()
char src1[10]="abcde"
StringSet(src1[0], "Local HMI", LW, 0, 5)
end macro_command
```

The ASCII Display object shows:



If *data_count* is an even number that is greater than or equal to the length of the string, the content of string can be completely shown:

```
macro_command main()
char src1[10]="abcde"
StringSet(src1[0], "Local HMI", LW, 0, 6)
end macro_command
```



Example

```
macro_command main()

char str1[10]="abcde"

//    Send 3 words to LW-0~LW-2
//    Data are being sent until the end of string is reached.
//    Even though the value of data_count is larger than the length of string
//    , the function will automatically stop.
StringSet(str1[0], "Local HMI", LW, 0, 10)
```

	end macro_command

Name	StringSetEx
Syntax	StringSetEx (*send_data*[*start*], *device_name, device_type, address_offset, data_count*)
Description	Sends data to the device and continues executing next command even if there's no response from this device. Descriptions of *send_data, device_name, device_type, address_offset* and *data_count* are the same as StringSet.

Example	macro_command main()
	char str1[20]="abcde"
	short test=0
	
	// macro will continue executing test = 1 even if the MODBUS device is
	// not responding
	StringSetEx(str1[0], "MODBUS RTU", 4x, 0, 20)
	test = 1
	
	// macro will not continue executing test = 2 until MODBUS device responds
	StringSet(str1[0], "MODBUS RTU", 4x, 0, 20)
	test = 2
	
	end macro_command

Name	StringToUpper
Syntax	success = StringToUpper (source[start], destination[start]) success = StringToUpper ("source", destination[start])
Description	Converts all the characters in the source string to uppercase characters and stores the result in the destination buffer. The source string parameter accepts both static string (in the form: "source") and char array (in the form: source[start]). This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of result string after conversion exceeds the size of destination buffer, it returns false.

Example	macro_command main()
	char src1[20]="aBcDe"
	char dest1[20]
	bool success1
	success1 = StringToUpper(src1[0], dest1[0])
	// success1=true, dest1="ABCDE"
	
	char dest2[4]
	bool success2
	success2 = StringToUpper("aBcDe", dest2[0])
	// success2=false, dest2 remains the same.
	
	end macro_command

Name	StringToLower
Syntax	success = StringToLower (source[start], destination[start]) success = StringToLower ("source", destination[start])
Description	Converts all the characters in the source string to lowercase characters and stores the result in the destination buffer. The source string parameter accepts both static string (in the form: "source") and char array (in the form: source[start]). This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of result string after conversion exceeds the size of destination buffer, it returns false.

Example	macro_command main()
	char src1[20]="aBcDe"
	char dest1[20]
	bool success1
	success1 = StringToLower(src1[0], dest1[0])
	// success1=true, dest1="abcde"
	
	char dest2[4]
	bool success2
	success2 = StringToLower("aBcDe", dest2[0])
	// success2=false, dest2 remains the same.
	
	end macro_command

Name	StringToReverse
Syntax	success = StringToReverse (source[start], destination[start]) success = StringToReverse ("source", destination[start])
Description	Reverses the characters in the source string and stores it in the destination buffer.

	The source string parameter accepts both static string (in the form: "source") and char array (in the form: source[start]). This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of reversed string exceeds the size of destination buffer, it returns false.

Example	macro_command main()
	char src1[20]="abcde"
	char dest1[20]
	bool success1
	success1 = StringToReverse(src1[0], dest1[0])
	// success1=true, dest1="edcba"
	
	char dest2[4]
	bool success2
	success2 = StringToReverse("abcde", dest2[0])
	// success2=false, dest2 remains the same.
	
	end macro_command

Name	StringTrimLeft
Syntax	success = StringTrimLeft (source[start], set[start], destination[start]) success = StringTrimLeft ("source", set[start], destination[start]) success = StringTrimLeft (source[start], "set", destination[start]) success = StringTrimLeft ("source", "set", destination[start])
Description	Trims the leading specified characters in the set buffer from the source string. The source string and set string parameters accept both static string (in the form: "source") and char array (in the form: source[start]). This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of trimmed string exceeds the size of destination buffer, it returns false.

Example	macro_command main()
	char src1[20]= "# *a*#bc"
	char set1[20]="# *"
	char dest1[20]
	bool success1
	success1 = StringTrimLeft (src1[0], set1[0], dest1[0])
	// success1=true, dest1="a*#bc"
	
	char set2[20]={'#', ' ', '*'}
	char dest2[4]
	bool success2
	success2 = StringTrimLeft ("# *a*#bc", set2[0], dest2[0])
	// success2=false, dest2 remains the same.

	char src3[20]="abc *#" char dest3[20] bool success3 success3 = StringTrimLeft (src3[0], "# *", dest3[0]) // success3=true, dest3="abc *#" end macro_command

Name	StringTrimRight
Syntax	success = StringTrimRight (source[start], set[start], destination[start]) success = StringTrimRight ("source", set[start], destination[start]) success = StringTrimRight (source[start], "set", destination[start]) success = StringTrimRight ("source", "set", destination[start])
Description	Trims the trailing specified characters in the set buffer from the source string. The source string and set string parameters accept both static string (in the form: "source") and char array (in the form: source[start]). This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of trimmed string exceeds the size of destination buffer, it returns false.

Example	macro_command main()
	char src1[20]= "# *a*#bc# * "
	char set1[20]="# *"
	char dest1[20]
	bool success1
	success1 = StringTrimRight(src1[0], set1[0], dest1[0])
	// success1=true, dest1="# *a*#bc"
	
	char set2[20]={'#', ' ', '*'}
	char dest2[20]
	bool success2
	success2 = StringTrimRight("# *a*#bc", set2[0], dest2[0])
	// success2=true, dest2="# *a*#bc"
	
	char src3[20]="ab**c *#"
	
	char dest3[4]
	bool success3
	success3 = StringTrimRight(src3[0], "# *", dest3[0])

	// success3=false, dest3 remains the same. end macro_command

Name	Unicode2Utf8
Syntax	result = Unicode2Utf8(source[start], destination[start])
Description	Converts the source Unicode string to UTF8 string and stores the result in the destination buffer. This function returns a Boolean indicating whether the process is successfully done or not. If successful, it returns true,; otherwise it returns false.
Example	macro_command main() char unicode_str[20] char utf8_str[20] String2Unicode("ABC", unicode_str[0]) bool result result = Unicode2Utf8(unicode_str[0], utf8_str[0]) // "result" will be set to true. "utf8_str" will equal "ABC" encoded in UTF8 StringCat("DEF", utf8_str[0]) // "utf8_str" will equal "ABCDEF" encoded in UTF8 char dst[20] bool result2 result2 = Utf82Unicode(utf8_str[0], dst[0]) // "result" will be set to true. "dst" will equal "ABCDEF" encoded in Unicode. end macro_command

Name	UnicodeCat
Syntax	result = UnicodeCat(source[start], destination[start]) or result = UnicodeCat("source", destination[start])
Description	This function concatenate strings. It appends the source string to the destination string. The source string parameter accepts both static string (e.g. "source") and char array (e.g. source[start]). *destination[start]* must be an one-dimensional char array. This function returns a Boolean indicating whether the process has been successfully completed. If successful, it returns true; otherwise it returns false. If the length of the result string after concatenation exceeds the max. size of destination buffer, it returns false, and the destination string remains unchanged.
Example	macro_command main()

	char strSrc[12]="αθβγθδ"
	char strDest[28]="ζηθλ1234"
	bool result
	
	result = UnicodeCat(strSrc[0], strDest[0]) // "result" will be set to true
	//"strDest" will be set to "ζηθλ1234αθβγθδ"
	
	result = UnicodeCat("ζηθλ", strDest[0]) // the function fails.
	// "result" will be set to false due to insufficient destination buffer size.
	// In this case, the content of "strDest" remains the same.
	
	end macro_command

Name	UnicodeCompare
Syntax	result = UnicodeCompare(string1[start], string2[start]) result = UnicodeCompare("string1", string2[start]) result = UnicodeCompare(string1[start], "string2") result = UnicodeCompare("string1", "string2")
Description	Performs case-sensitive comparison of two strings. The two string parameters accept both static string (e.g. "string") and char array (e.g. string[start]). This function returns a Boolean indicating the result of comparison. If two strings are identical, it returns true. Otherwise it returns false.

Example	macro_command main()
	char str1[10]=" θαβθγ"
	char str2[8]="αβγδ"
	bool result
	
	result = UnicodeCompare(str1[0], str2[0]) // "result" will be set to false.
	result = UnicodeCompare(str1[0], "θαβθγ") // "result" will be set to true.
	
	end macro_command

Name	UnicodeCopy
Syntax	result = UnicodeCopy("source", destination[start]) or result = UnicodeCopy(source[start], destination[start])
Description	Copies a string. This function copies a static string (which is enclosed in quotes)

	or a string that is stored in an array to the destination buffer. The source string parameter accepts both static string (e.g. "source") and char array (in the form: source[start]). destination[start] must be an one-dimensional char array. This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of source string exceeds the max. size of destination buffer, it returns false and the content of destination remains unchanged. The result field is optional.

Example	macro_command main()
	char strSrc[14]="αβθγδθε" //αβθγδθε
	char strDest[14]
	bool result
	
	result = UnicodeCopy(strSrc[0], strDest[0]) // "result" will be set to true.
	result = UnicodeCopy("αβθγδθε", strDest[0])
	// "result" will be set to true, strDest = αβθγδθε"
	result = UnicodeCopy("αβγδεζαβγδεζ", strDest[0])
	// "result" will be set to false.
	// The size of source string exceeds the size of destination string.
	
	end macro_command

Name	UnicodeExcluding
Syntax	result = UnicodeExcluding(source[start], set[start], destination[start]) result = UnicodeExcluding("source", set[start], destination[start]) result = UnicodeExcluding(source[start], "set", destination[start]) result = UnicodeExcluding("source", "set", destination[start])
Description	Retrieves a substring of the source string that contains characters that are not in the set string. The result string is the part of the source string beginning with the first character and ending before any character in the target string is found in the source string. The source string and set string parameters accept both static string (in the form: "source") and char array (in the form: source[start]). This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false. If the length of retrieved substring exceeds the size of destination buffer, it returns false.

Example	macro_command main()
	char source[14]="γδξκθλθ, dest[8]
	char set[4]="λθ"
	bool result
	
	result = UnicodeExcluding(source[0], set[0], dest[0]) // the function succeeds.
	// "result" will be set to true and "dest" will be set to "γδξκ".
	
	result = UnicodeExcluding(source[0], set[0], dest[4]) // the function fails.
	// "result" will be set to false due to insufficient destination buffer size.
	
	end macro_command

Name	UnicodeLength
Syntax	result = UnicodeLength(source[start]) or result = UnicodeLength("source")
Description	Obtains the length of a Unicode string. The source string parameter accepts both static string (e.g. "source") and char array (in the form: source[start]). The returned value is the length of the source string.

Example	macro_command main()
	
	char strSrc[6]="ÅÈÑ"
	int result1, result2
	
	result1 = UnicodeLength(strSrc[0]) // "result1" is equal to 3
	result2 = UnicodeLength("trSrc[0]) // "re2" is equal to 3
	
	end macro_command

Name	Utf82Unicode
Syntax	result = Utf82Unicode(source[start], destination[start])
Description	Converts the source UTF8 string to a Unicode string and stores the result in the destination buffer. This function returns a Boolean indicating whether the process has been successfully completed. If so, it returns true; otherwise it returns false.
Example	macro_command main()

	char unicode_str[20] char utf8_str[20] String2Unicode("ABC", unicode_str[0]) bool result result = Unicode2Utf8(unicode_str[0], utf8_str[0]) // "result" will be set to true. "utf8_str" will equal "ABC" encoded in UTF8 StringCat("DEF", utf8_str[0]) // "utf8_str" will equal "ABCDEF" encoded in UTF8 char dst[20] bool result2 result2 = Utf82Unicode(utf8_str[0], dst[0]) // "result" will be set to true. "dst" will equal "ABCDEF" encoded in Unicode. end macro_command

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

18.7.8. Mathematics

Name	SQRT
Syntax	SQRT(*source, result*)
Description	Calculates the square root of *source* and stores the result into *result*. *source* can be a constant or a variable. *result* must be a variable. *source* must be a nonnegative value.
Example	macro_command main() float source, result SQRT(15, result) source = 9.0 SQRT(source, result)// result is 3.0 end macro_command

Name	CUBERT
Syntax	CUBERT(*source, result*)
Description	Calculates the cube root of source and stores the result into *result*. *source* can be a constant or a variable. *result* must be a variable. *source* must be a nonnegative value.

Example	macro_command main()
	float source, result
	
	CUBERT (27, result) // result is 3.0
	
	source = 27.0
	CUBERT(source, result)// result is 3.0
	
	end macro_command

Name	POW
Syntax	POW(*source1, source2, result*)
Description	Calculates *source1* to the power of *source2*. *source1* and *source2* can be a constant or a variable. *result* must be a variable. *source1* and *source2* must be a nonnegative value.
Example	macro_command main() float y, result y = 0.5 POW (25, y, result) // result = 5 end macro_command

Name	SIN
Syntax	SIN(*source, result*)
Description	Calculates the sine of *source* (degree) into *result*. *source* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	float source, result
	
	SIN(90, result)// result is 1
	
	source = 30
	SIN(source, result)// result is 0.5
	
	end macro_command

Name	COS
Syntax	COS(*source, result*)
Description	Calculates the cosine of *source* (degree) into *result*.

	source can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	float source, result
	
	COS(90, result)// result is 0
	
	source = 60
	COS(source, result)// result is 0.5
	
	end macro_command

Name	TAN
Syntax	TAN(*source, result*)
Description	Calculates the tangent of *source* (degree) into *result*. *source* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	float source, result
	
	TAN(45, result)// result is 1
	
	source = 60
	TAN(source, result)// result is 1.732
	
	end macro_command

Name	COT
Syntax	COT(*source, result*)
Description	Calculates the cotangent of *source* (degree) into *result*. *source* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	float source, result
	
	COT(45, result)// result is 1
	
	source = 60
	COT(source, result)// result is 0.5774
	
	end macro_command

Name	SEC
Syntax	SEC(*source, result*)
Description	Calculates the secant of *source* (degree) into *result*. *source* can be a constant or a variable. *result* must be a variable.
Example	macro_command main()

	float source, result
	
	SEC(45, result)// result is 1.414
	
	source = 60
	SEC(source, result)// if source is 60, result is 2
	
	end macro_command

Name	CSC
Syntax	CSC(*source, result*)
Description	Calculates the cosecant of *source* (degree) into *result*. *source* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	float source, result
	
	CSC(45, result)// result is 1.414
	
	source = 30
	CSC(source, result)// result is 2
	
	end macro_command

Name	ASIN
Syntax	ASIN(*source, result*)
Description	Calculates the arc sine of *source* into *result* (degree). *source* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	float source, result
	
	ASIN(0.8660, result)// result is 60
	
	source = 0.5
	ASIN(source, result)// result is 30
	
	end macro_command

Name	ACOS
Syntax	ACOS(*source, result*)
Description	Calculates the arc cosine of *source* into *result*. *source* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	float source, result
	
	ACOS(0.8660, result)// result is 30
	
	source = 0.5
	ACOS(source, result)// result is 60
	
	end macro_command

Name	ATAN
Syntax	ATAN(*source, result*)
Description	Calculates the arc tangent of *source* into *result*. *source* can be a constant or a variable. *result* must be a variable.

Example	macro_command main()
	float source, result
	
	ATAN(1, result)// result is 45
	
	source = 1.732
	ATAN(source, result)// result is 60
	
	end macro_command

Name	LOG
Syntax	LOG (*source, result*)
Description	Calculates the natural logarithm of a number and saves into *result*. *source* can be either a variable or a constant. *result* must be a variable.
Example	macro_command main() float source = 100, result LOG (source, result)// result is approximately 4.6052 end macro_command

Name	LOG10
Syntax	LOG10(*source, result*)
Description	Calculates the base-10 logarithm of a number and saves into *result*. *source* can be either a variable or a constant. *result* must be a variable.
Example	macro_command main() float source = 100, result LOG10 (source, result) // result is 2 end macro_command

Name	RAND
Syntax	RAND(*result*)
Description	Calculates a random integer and saves into *result*. (Range: 0 ~ 32766) *result* must be a variable.
Example	macro_command main() short result RAND (result) //result is not a fixed value when executes macro every time end macro_command

Name	CEIL
Syntax	result=CEIL(source)
Description	Get the smallest integral value that is not less than input.
Example	macro_command main() float x = 3.8 int result result = CEIL(x)// result = 4 end macro_command

Name	FLOOR
Syntax	result=FLOOR(source)
Description	Get the largest integral value that is not greater than input.
Example	macro_command main() float x = 3.8 int result result = FLOOR(x) // result = 3 end macro_command

Name	ROUND
Syntax	result=ROUND(source)
Description	Get the integral value that is nearest the input.
Example	macro_command main() float x = 5.55 int result result = ROUND(x) // result = 6 end macro_command

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

18.7.9. Statistics

Name	AVERAGE
Syntax	AVERAGE(*source[start], result, count*)
Description	Gets the average value from array.

Example	int data[5] = {1, 2, 3, 4, 5}
	float result
	
	AVERAGE(data[0], result, 5) // result is equal to 3
	AVERAGE(data[2], result, 3) // result is equal to 4

Name	HARMEAN
Syntax	HARMEAN(*source[start], result, count*)
Description	Gets the harmonic mean value from array.
Example	int data[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

	float result HARMEAN(data[0], result, 10) // result is equal to 3.414

Name	MAX
Syntax	MAX(*source[start], result, count*)
Description	Gets the maximum value from array.

Example	int data[5] = {1, 2, 3, 4, 5}
	int result
	
	MAX(data[0], result, 5) // result is equal to 5
	MAX(data[1], result, 3) // result is equal to 4

Name	MEDIAN
Syntax	MEDIAN(*source[start], result, count*)
Description	Gets the median value from array.
Example	int data[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} float result MEDIAN(data[0], result, 10) // result is equal to 5.5

Name	MIN
Syntax	MIN(*source[start], result, count*)
Description	Gets the minimum value from array.

Example	int data[5] = {1, 2, 3, 4, 5}
	int result
	
	MIN(data[0], result, 5) // result is equal to 1
	MIN(data[1], result, 3) // result is equal to 2

Name	STDEVP
Syntax	STDEVP(*source[start], result, count*)
Description	Gets the standard deviation value from array.
Example	int data[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} float result STDEVP(data[0], result, 10) // result is equal to 2.872

Name	STDEVS
Syntax	STDEVS(*source[start], result, count*)
Description	Gets the sample standard deviation value from array.
Example	int data[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} float result

	STDEVS(data[0], result, 10) // result is equal to 3.027

18.7.10. Recipe Database

Name	RecipeGetData
Syntax	RecipeGetData(*destination, recipe_address, record_ID*)
Description	Gets Recipe Data. The gained data will be stored in *destination*, and must be a variable. *recipe_address* consists of recipe name and item name: "recipe_name.item_name". record_ID specifies the ID number of the record in recipe being gained.

Example	macro_command main()
	int data=0
	char str[20]
	int recordID
	bool result
	
	recordID = 0
	result = RecipeGetData(data, "TypeA.item_weight", recordID)
	// From recipe "TypeA" get the data of the item "item_weight" in record 0.
	
	recordID = 1
	result = RecipeGetData(str[0], "TypeB.item_name", recordID)
	  // From recipe "TypeB" get the data of the item "item_name" in record 1.
	
	end macro_command

Name	RecipeQuery
Syntax	RecipeQuery (*SQL_command, destination*)
Description	Uses SQL statement to query recipe data. The number of records of query result will be stored in the *destination*. This must be a variable. SQL command can be static string or char array. Example: RecipeQuery("SELECT * FROM TypeA", destination) or RecipeQuery(sql[0], destination) SQL statement must start with "SELECT * FROM" followed by recipe name and query condition.
Example	macro_command main() int total_row=0 char sql[100]="SELECT * FROM TypeB" short var bool result result = RecipeQuery("SELECT * FROM TypeA", total_row) // Query Recipe "TypeA". Store the number of records of query result in total_row.

	result = RecipeQuery(sql[0], total_row)
	// Query Recipe "TypeB". Store the number of records of query result in total_row.
	
	result = RecipeQuery("SELECT * FROM Recipe WHERE Item >%(var)", total_row)
	// Query "Recipe", where "Item" is larger than var. Store the number of records of query result in total_row.
	
	end macro_command

Name	RecipeQueryGetData
Syntax	RecipeQueryGetData (*destination, recipe_address, result_row_no*)
Description	Gets the data in the query result obtained by RecipeQuery. This function must be called after calling RecipeQuery, and specify the same recipe name in *recipe_address* as RecipeQuery. *result_row_no* specifies the sequence row number in query result

Example	macro_command main()
	
	int data=0
	int total_row=0
	int row_number=0
	bool result_query
	bool result_data
	
	result_query = RecipeQuery("SELECT * FROM TypeA", total_row)
	// Query Recipe "TypeA". Store the number of records of query result in total_row.
	if (result_query) then
	for row_number=0 to total_row-1
	result_data = RecipeQueryGetData(data, "TypeA.item_weight", row_number)
	next row_number
	end if
	
	end macro_command

Name	RecipeQueryGetRecordID
Syntax	RecipeQueryGetRecordID (*destination, result_row_no*)
Description	Gets the record ID numbers of those records gained by RecipeQuery. This function must be called after calling RecipeQuery. *result_row_no* specifies the sequence row number in query result, and write the obtained record ID to destination.
Example	macro_command main() int recordID=0 int total_row=0 int row_number=0 bool result_query

	bool result_id
	
	result_query = RecipeQuery("SELECT * FROM TypeA", total_row)
	// Query Recipe "TypeA". Store the number of records of query result in total_row.
	if (result_query) then
	for row_number=0 to total_row-1
	result_id = RecipeQueryGetRecordID(recordID, row_number)
	next row_number
	end if
	
	end macro_command

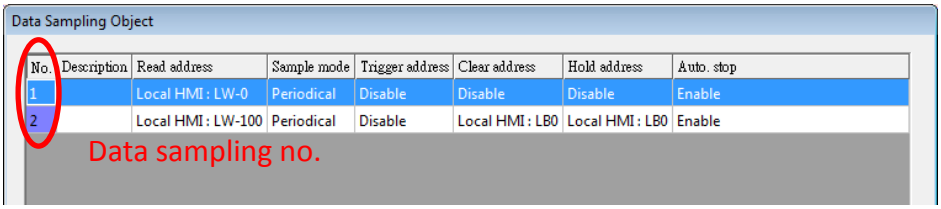
Name	RecipeSetData
Syntax	RecipeSetData(*source*, *recipe address*, *record_ID*)
Description	Writes data to recipe. If success, returns true, else, returns false. *recipe_address* consists of recipe name and item name: "recipe_name.item_name". *record_ID* specifies the ID number of the record in recipe being modified.

Example	macro_command main()
	
	int data=99
	char str[20]="abc"
	int recordID
	bool result
	
	recordID = 0
	result = RecipeSetData(data, "TypeA.item_weight", recordID)
	// set data to recipe "TypeA", where item name is "item_weight" and the record ID is 0.
	
	recordID = 1
	result = RecipeSetData(str[0], "TypeB.item_name", recordID)
	// set data to recipe "TypeB", where item name is "item_name" and the record ID is 1.
	
	end macro_command

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

18.7.11. Data/Event Log

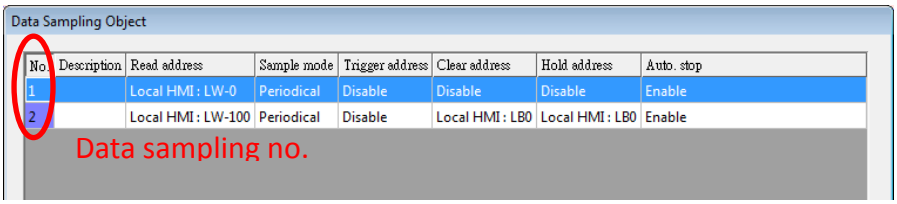
Name	FindDataSamplingDate
Syntax	return_value = FindDataSamplingDate *(data_log_number, index, year, month, day*) or FindDataSamplingDate (*data_log_number, index, year, month, day*)

Description	A query function for finding the date of specified data sampling file according to the data sampling no. and the file index. The date is stored into year, month and day respectively in the format of YYYY, MM and DD.  The directory of saved data: [Storage location]\[filename]\yyyymmdd.dtl. The data sampling files under the same directory are sorted according to the file name and are indexed starting from 0. The most recently saved file has the smallest file index number. For example, if there are four data sampling files as follows: 20101210.dtl 20101230.dtl 20110110.dtl 20110111.dtl The file index are: 20101210.dtl -> index is 3 20101230.dtl -> index is 2 20110110.dtl -> index is 1 20110111.dtl -> index is 0 *return_value* equals to 1 if referred data sampling file is successfully found, otherwise it equals to 0. *data_log_number* and *index* can be constant or variable. *year, month, day* and *return_value* must be variable. *return_value* is optional.

Example	macro_command main()
	short data_log_number = 1, index = 2, year, month, day
	short success
	
	// if there exists a data sampling file named 20101230.dtl, with data sampling // number 1 and file index 2.
	// the result after execution: success == 1, year == 2010, month == 12 and //day == 30
	success = FindDataSamplingDate(data_log_number, index, year, month, day)
	
	end macro_command

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Name	FindDataSamplingIndex
Syntax	return_value = FindDataSamplingIndex (*data_log_number, year, month, day, index*) or FindDataSamplingIndex (*data_log_number, year, month, day, index*)
Description	A query function for finding the file index of specified data sampling file according to the data sampling no. and the date. The file index is stored into

	index. year, month and day are in the format of YYYY, MM and DD respectively.  The directory of saved data: [Storage location]\[filename]\yyyymmdd.dtl. The data sampling files under the same directory are sorted according to the file name and are indexed starting from 0. The most recently saved file has the smallest file index number. For example, if there are four data sampling files as follows: 20101210.dtl 20101230.dtl 20110110.dtl 20110111.dtl The file index are: 20101210.dtl -> index is 3 20101230.dtl -> index is 2 20110110.dtl -> index is 1 20110111.dtl -> index is 0 *return_value* equals to 1 if referred data sampling file is successfully found, otherwise it equals to 0. *data_log_number, year, month* and *day* can be constant or variable. *index* and *return_value* must be variable. *return_value* is optional.

Example	macro_command main()
	short data_log_number = 1, year = 2010, month = 12, day = 10, index
	short success
	
	// if there exists a data sampling file named 20101210.dtl, with data sampling // number 1 and file index 2.
	// the result after execution: success == 1 and index == 2
	success = FindDataSamplingIndex (data_log_number, year, month, day, index)
	
	end macro_command

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Name	FindEventLogDate
Syntax	return_value = FindEventLogDate (*index, year, month, day*) or FindEventLogDate (*index, year, month, day*)
Description	A query function for finding the date of specified event log file according to file index. The date is stored into year, month and day respectively in the format of YYYY, MM and DD. The event log files stored in the designated position (such as HMI memory storage or external memory device) are sorted according to the file name and are indexed starting from 0. The most recently saved file has the smallest file

	index number. For example, if there are four event log files as follows: EL_20101210.evt EL_20101230.evt EL_20110110.evt EL_20110111.evt The file index are: EL_20101210.evt -> index is 3 EL_20101230.evt -> index is 2 EL_20110110.evt -> index is 1 EL_20110111.evt -> index is 0 *return_value* equals to 1 if referred data sampling file is successfully found, otherwise it equals to 0. *index* can be constant or variable. *year, month, day* and *return_value* must be variable. *return_value* is optional.

Example	macro_command main()
	short index = 1, year, month, day
	short success
	
	// if there exists an event log file named EL_20101230.evt，with index 1
	// the result after execution: success == 1, year == 2010, month == 12, day //== 30
	success = FindEventLogDate (index, year, month, day)
	
	end macro_command

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Name	FindEventLogIndex
Syntax	return_value = FindEventLogIndex (*year, month, day, index*) or FindEventLogIndex (*year, month, day, index*)
Description	A query function for finding the file index of specified event log file according to date. The file index is stored into index. year, month and day are in the format of YYYY, MM and DD respectively. The event log files stored in the designated position (such as HMI memory storage or external memory device) are sorted according to the file name and are indexed starting from 0. The most recently saved file has the smallest file index number. For example, if there are four event log files as follows: EL_20101210.evt EL_20101230.evt EL_20110110.evt EL_20110111.evt The file index are: EL_20101210.evt -> index is 3 EL_20101230.evt -> index is 2 EL_20110110.evt -> index is 1 EL_20110111.evt -> index is 0 *return_value* equals to 1 if referred data sampling file is successfully found, otherwise it equals to 0.

	index can be constant or variable. *year, month, day* and *return_value* must be variable. *return_value* is optional.

Example	macro_command main()
	short year = 2010, month = 12, day = 10, index
	short success
	
	// if there exists an event log file named EL_20101210.evt, with index 2
	// the result after execution: success == 1, index == 2
	success = FindEventLogIndex (year, month, day, index)
	
	end macro_command

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

18.7.12. Checksum

Name	ADDSUM
Syntax	ADDSUM(*source*[*start*], *result, data_count*)
Description	Adds up the elements of an array (*source*) from *source[start]* to *source*[*start + data_count - 1*] to generate a checksum. Puts in the checksum into *result*. *result* must be a variable. *data_count* is the amount of the accumulated elements and can be a constant or a variable.
Example	macro_command main() char data[5] short checksum data[0] = 0x1 data[1] = 0x2 data[2] = 0x3 data[3] = 0x4 data[4] = 0x5 ADDSUM(data[0], checksum, 5)// checksum is 0xf end macro_command

Name	XORSUM
Syntax	XORSUM(*source*[*start*], *result, data_count*)
Description	Uses XOR to calculate the checksum from *source*[*start*] to *source*[*start + data_count - 1*]. Puts the checksum into *result*. *result* must be a variable. *data_count* is the amount of the calculated elements of the array and can be a constant or a variable.
Example	macro_command main() char data[5] = {0x1, 0x2, 0x3, 0x4, 0x5} short checksum

	XORSUM(data[0], checksum, 5)// checksum is 0x1 end macro_command

Name	BCC
Syntax	BCC(*source[start], result, data_count*)
Description	Same as XORSUM.
Example	macro_command main() char data[5] = {0x1, 0x2, 0x3, 0x4, 0x5} char checksum BCC(data[0], checksum, 5) // checksum is 0x1 end macro_command

Name	CRC
Syntax	CRC(*source*[*start*], *result, data_count*)
Description	Calculates 16-bit CRC of the variables from *source*[*start*] to source[*start + data_count - 1*]. Puts in the 16-bit CRC into *result*. *result* must be a variable. *data_count* is the amount of the calculated elements of the array and can be a constant or a variable.
Example	macro_command main() char data[5] = {0x1, 0x2, 0x3, 0x4, 0x5} short checksum CRC(data[0], checksum, 5) // checksum is 0xbb2a, 16-bit CRC end macro_command

Name	CRC8
Syntax	CRC8(*source*[*start*], *result, data_count*)
Description	Calculates 8-bit CRC of the variables from *source*[*start*] to source[*start + data_count - 1*]. Puts in the 8-bit CRC into *result*. *result* must be a variable. *data_count* is the amount of the calculated elements of the array and can be a constant or a variable.
Example	macro_command main() char source[5] = {1, 2, 3, 4, 5} short CRC8_result CRC8(source[0], CRC8_result, 5) // CRC8_result = 188 end macro_command

18.7.13. Miscellaneous

Name	Beep
Syntax	Beep ()
Description	Plays beep sound. This command plays a beep sound with frequency of 800 hertz and duration of 30 milliseconds.
Example	macro_command main() Beep() end macro_command

Name	Buzzer
Syntax	Buzzer ()
Description	Turns ON / OFF the buzzer.

Example	macro_command main()
	
	char on = 1, off = 0
	Buzzer(on) // turn on the buzzer
	
	DELAY(1000) // delay 1 second
	
	Buzzer(off) // turn off the buzzer
	
	DELAY(500) // delay 500ms
	
	Buzzer(1) // turn on the buzzer
	
	DELAY(1000) // delay 1 second
	
	Buzzer(0) // turn off the buzzer
	
	end macro_command

Name	TRACE
Syntax	TRACE(*format, argument*)
Description	Use this function to send specified string to the EasyDiagnoser / cMT Diagnoser. Users can print out the current value of variables during run-time of macro for debugging. When TRACE encounters the first format specification (if any), it converts the value of the first argument after format and outputs it accordingly. format refers to the format control of output string. A format specification, which consists of optional (in [ ]) and required fields (in red font), has the following form: %[flags] [width] [.precision] type Each field of the format specification is described as below:

	flags (optional): - : Aligns left. When the value has fewer characters than the specified width, it will be padded with spaces on the left. + : Precedes the result with a plus or minus sign (+ or -) *width* (optional): A nonnegative decimal integer controlling the minimum number of characters printed. *precision* (optional): A nonnegative decimal integer which specifies the precision and the number of characters to be printed. *type*: C or c : specifies a single-byte character d : signed decimal integer i : signed decimal integer o : unsigned octal integer u : unsigned decimal integer X or x : unsigned hexadecimal integer lld : signed long integer (64-bit) (cMT / cMT X Series only) llu : unsigned long integer (64-bit) (cMT / cMT X Series only) f : signed floating-point value llf : double-precision floating-point value E or e : Scientific notation in the form "[–]d.dddd **e** [sign]ddd" , where d is a single decimal digit, dddd is one or more decimal digits, ddd is exactly three decimal digits, and sign is + or –. The length of output string is limited to 256 characters. Extra characters will be ignored. The *argument* part is optional. One format specification converts exactly one argument.

Example	macro_command main()
	char c1 = 'a'
	short s1 = 32767
	float f1 = 1.234567
	
	TRACE("The results are") // output: The results are
	TRACE("c1 = %c, s1 = %d, f1 = %f", c1, s1, f1)
	// output: c1 = a, s1 = 32767, f1 = 1.234567
	
	end macro_command

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

Name	GetCnvTagArrayIndex
Syntax	GetCnvTagArrayIndex(*array_index*)
Description	When a user-defined conversion tag uses array, the GetCnvTagArrayIndex() function of [Read conversion] subroutine can get the relative array index before doing conversion.

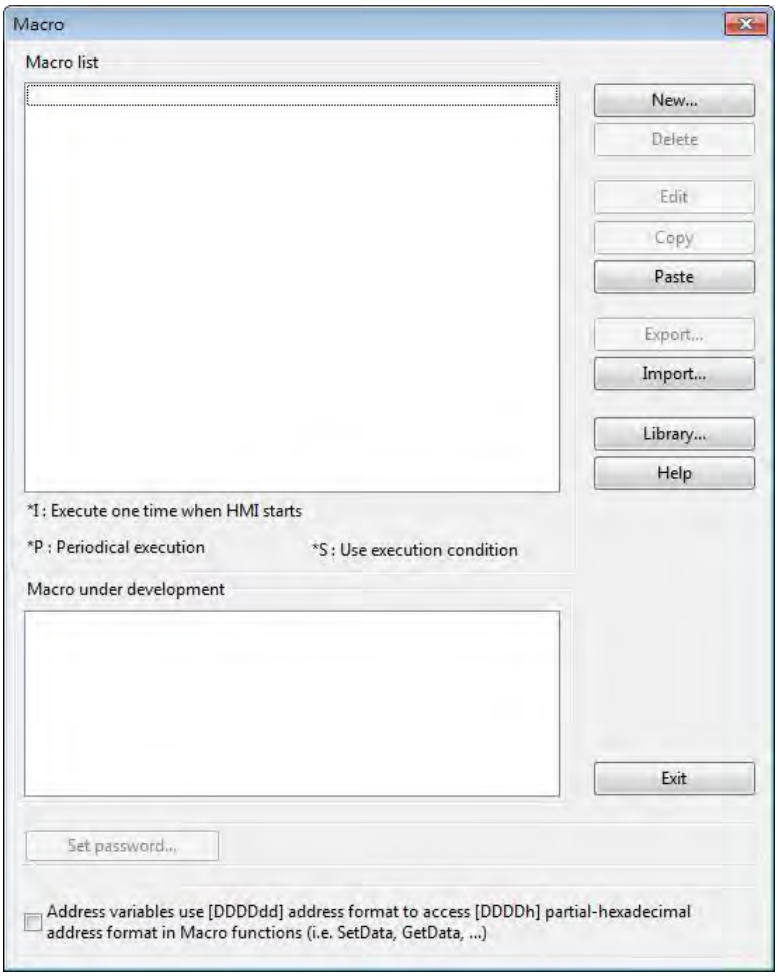
Example	Sub short newfun(short param) Int index GetCnvTagArrayIndex(index) If index is 2, the third data record in the array will be converted. return param end sub

18.8. How to Create and Execute a Macro

18.8.1. How to Create a Macro

Please follow the steps below to create a macro.

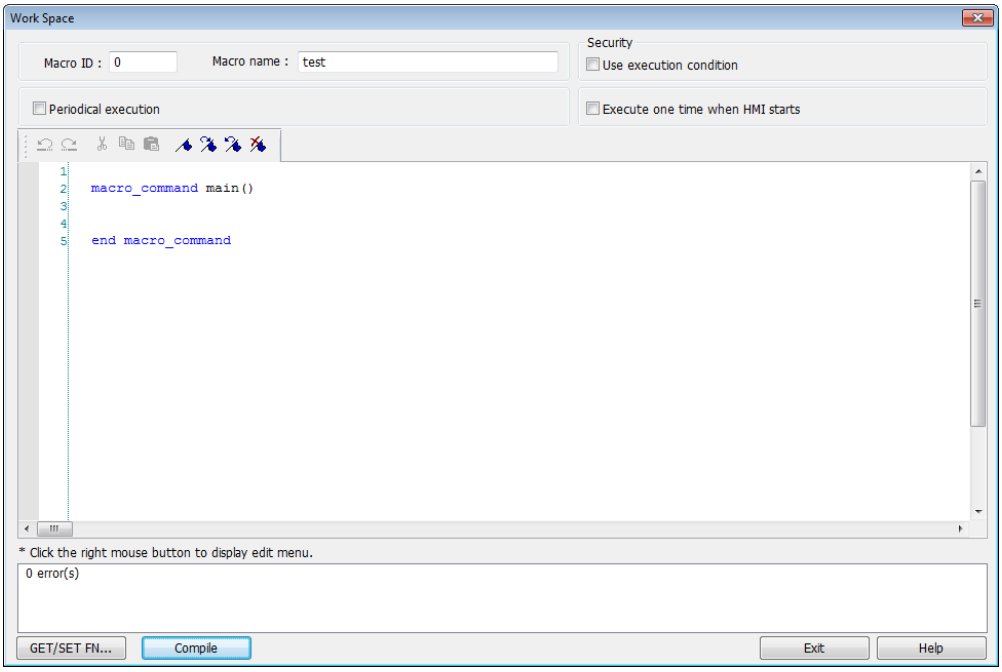
1. Click [Project] » [Macro] to open Macro Manager dialog box.



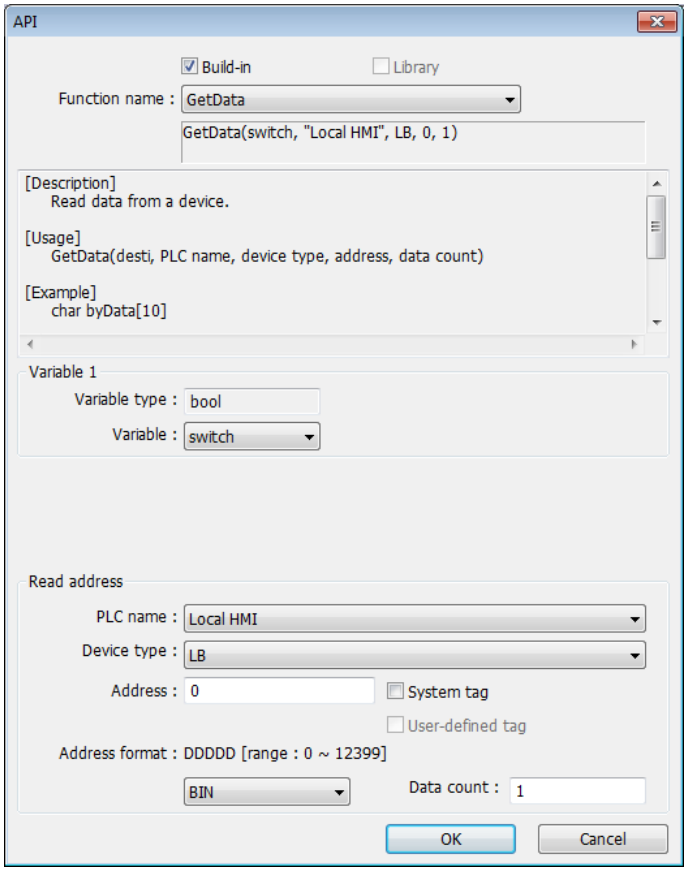
In Macro Manager, all macros compiled successfully are displayed in “Macro list”, and all macros under development or cannot be compiled are displayed in “Macro under development”. The following is a description of the various buttons.

Setting	Description
New	Opens a blank “WorkSpace” editor for creating a new macro.
Delete	Deletes the selected macro.
Edit	Opens the “WorkSpace” editor, and loads the selected macro.
Copy	Copies the selected macro into the clipboard.
Paste	Pastes the macro in the clipboard into the list, and creates a new name for the macro.
Export	Save the selected macro as *.edm file.
Import	Import an *.edm file to the project.
Library	Open Macro Function Library managing dialog.

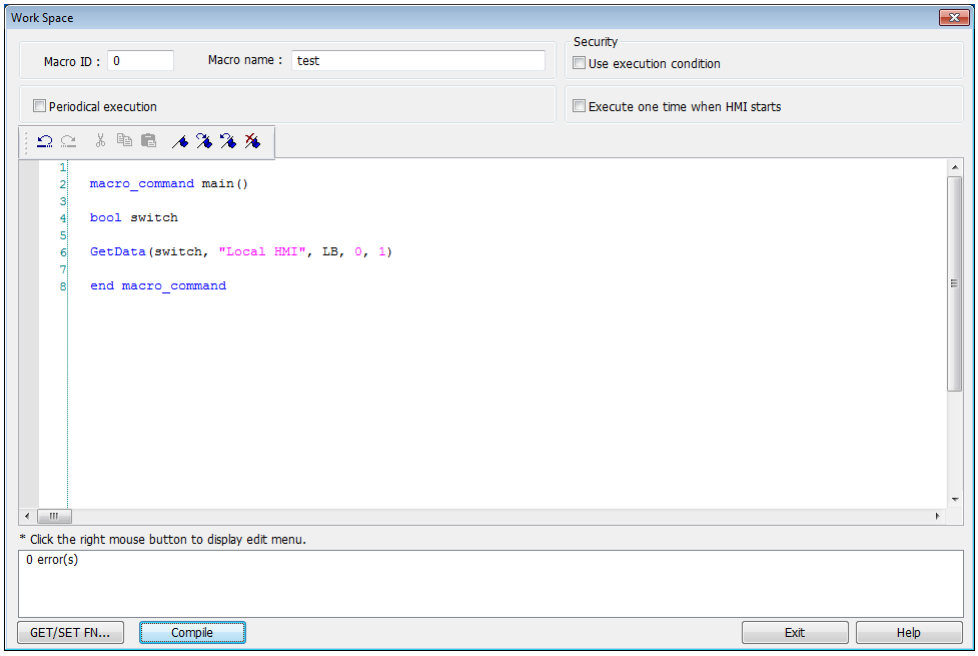
2. Press the [New] button to create an empty macro and open the macro editor. Every macro has a unique number defined at [Macro ID], and must have a macro name, otherwise an error will appear while compiling.



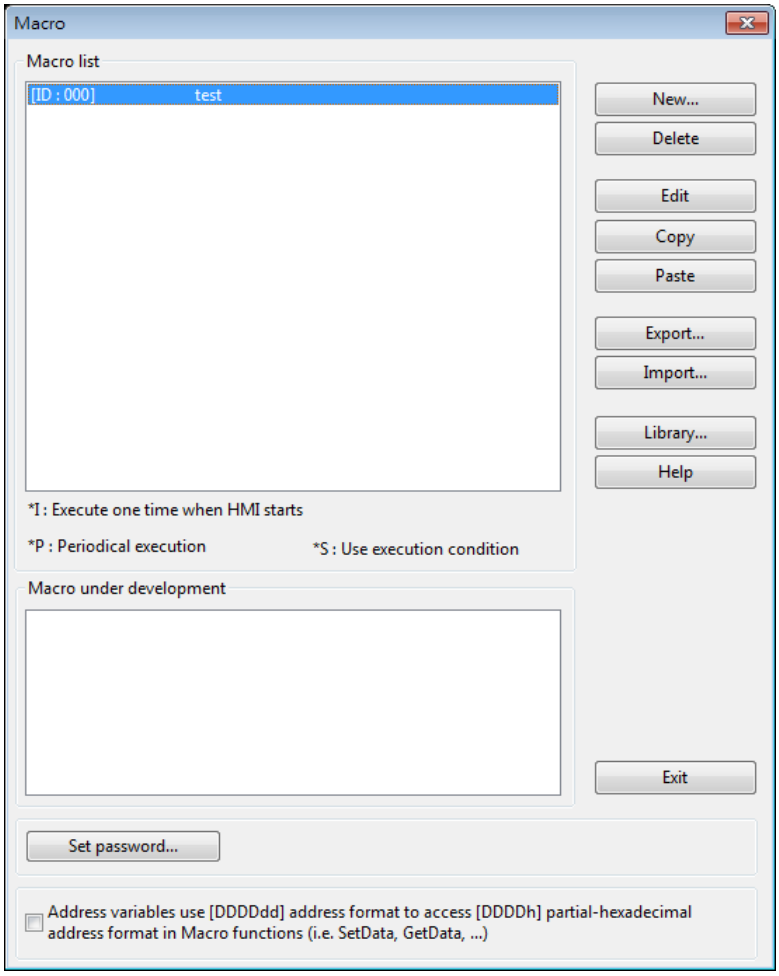
3. Design your macro. To use built-in functions (like SetData() or GetData()), press [Get/Set FN…] button to open API dialog box and select the function and set essential parameters.



4. After the completion of a new macro, press [Compile] button to compile the macro.



5. If there is no error, press [Exit] button and a new macro “macro_test” will be in “Macro list”.


Macro
Macro list
[ID : 000] test
New...
Delete
Edit
Copy
Paste
Export...
Import...
Library...
Help
*I : Execute one time when HMI starts
*P : Periodical execution
*S : Use execution condition
Macro under development
Exit
Set password...
Address variables use [DDDDdd] address format to access [DDDDh] partial-hexadecimal address format in Macro functions (i.e. SetData, GetData, ...)

18.8.2. Execute a Macro

There are several ways to execute a macro.

- Use a PLC Control object

1. Open [PLC Control] and add one PLC Control object with the [Type of control] as [Execute macro program].
2. Select the macro in [Macro name]. Choose a bit and select a trigger condition to trigger the macro. In order to guarantee that the macro will run only once, consider latching the trigger bit, and then resetting the trigger condition within the macro.
3. Use a [Set Bit] or Toggle Switch object to change the bit to activate the macro.

- Use a [Set Bit] or Toggle Switch object

1. On the [General] tab of the [Set Bit] or [Toggle Switch] dialog box, select the [Execute Macro] option.
2. Select the macro to execute. The macro will be executed one time when the button is activated.

- Use a Function Key object

1. On the [General] tab of the [Function Key] dialog, select the [Execute Macro] option.
2. Select the macro to execute. The macro will execute one time when the button is activated.

- In macro editor, use

1. [Periodical Execution]: Macro will be triggered periodically.
2. [Execute one time when HMI starts]: Macro will be executed once HMI starts.

- In Window Settings, Macro group box

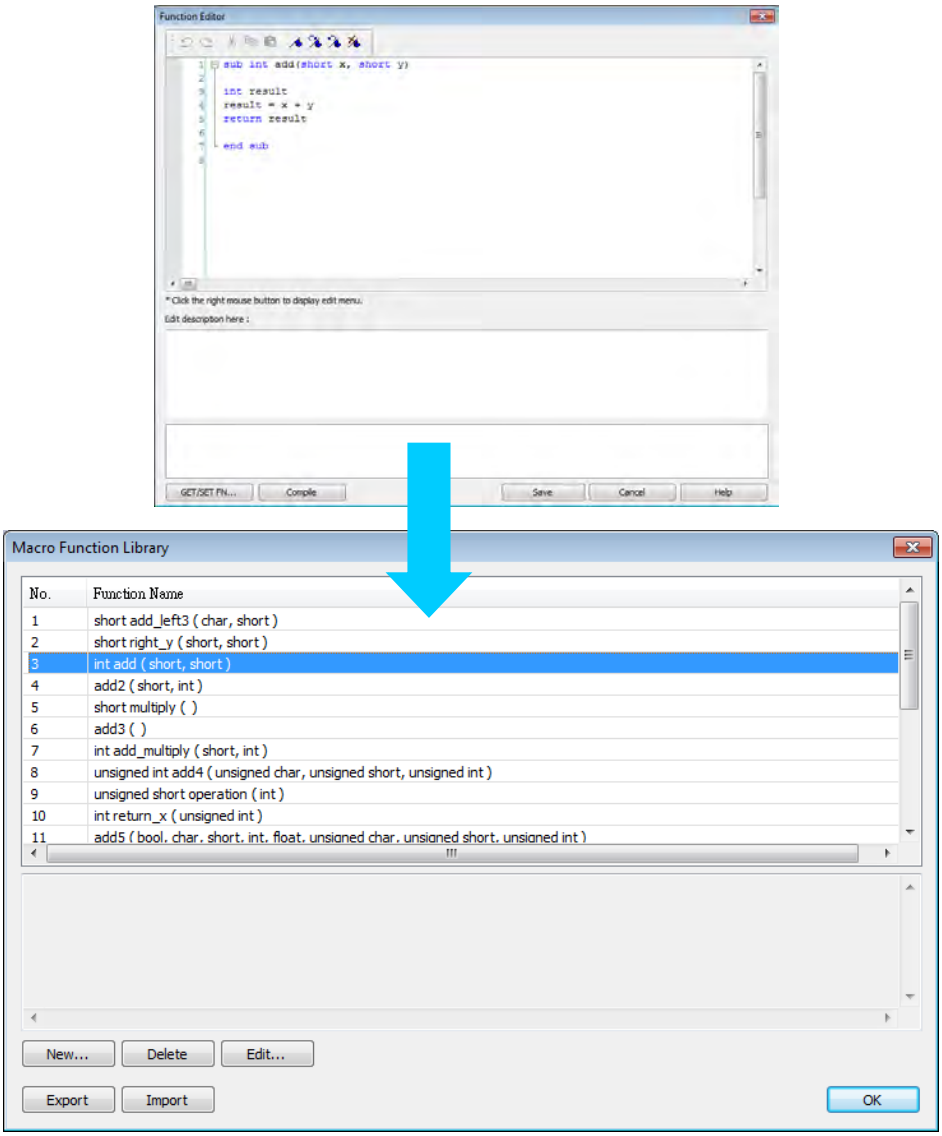
1. [Open]: When the window opens, run the selected macro once.
2. [Cycle]: When the window opens, run the selected macro every 0.5 second.
3. [Close]: When the window closes, run the selected macro once.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

18.9. User Defined Macro Function

When editing Macro, to save time of defining functions, user may search for the needed from built-in Macro Function Library. However, certain functions, though frequently used, may not be found there. In this case, user may define the needed function and save it for future use. Next time when the same function is required, the saved functions can be called from [Macro

Function Library] for easier editing. Additionally, [Macro Function Library] greatly enhances the portability of user-defined functions. Before building a function please check the built-in functions or online function library to see if it exists.



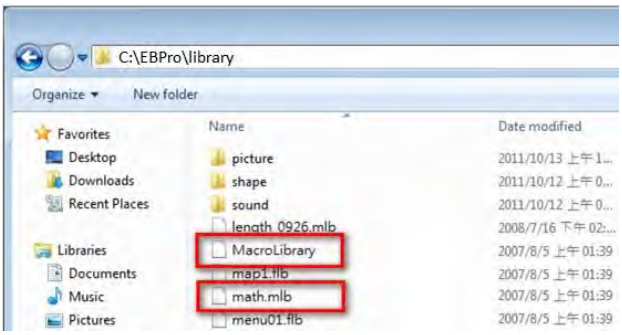
18.9.1. Import Function Library File

Open a project in HMI programming software, the default Function Library File will be read automatically and the function information will be loaded in. At this moment if a user-defined function is called, the relevant .mlb file must be imported first.

1. Default Function Library File Name: MacroLibrary (without filename extension)
2. Function Library Directory: HMI programming software installation directory\library (folder)
3. \library (folder) contains two types of function library files:
   Without filename extension: MacroLibrary, the Default Function Library for HMI programming software to read at the beginning.

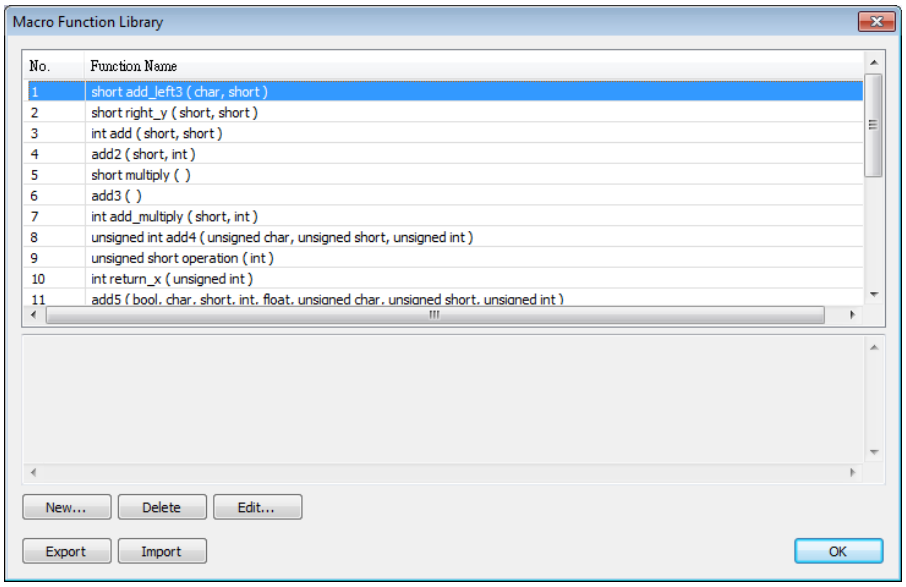
With filename extension (.mlb): Such as ”math.mlb”. The files to be read / written when users import / export. These files are portable and can be called from the folder when needed.

4. When opening HMI programming software, only the functions in Default Function Library will be loaded in, to use functions in .mlb files, please import them first.

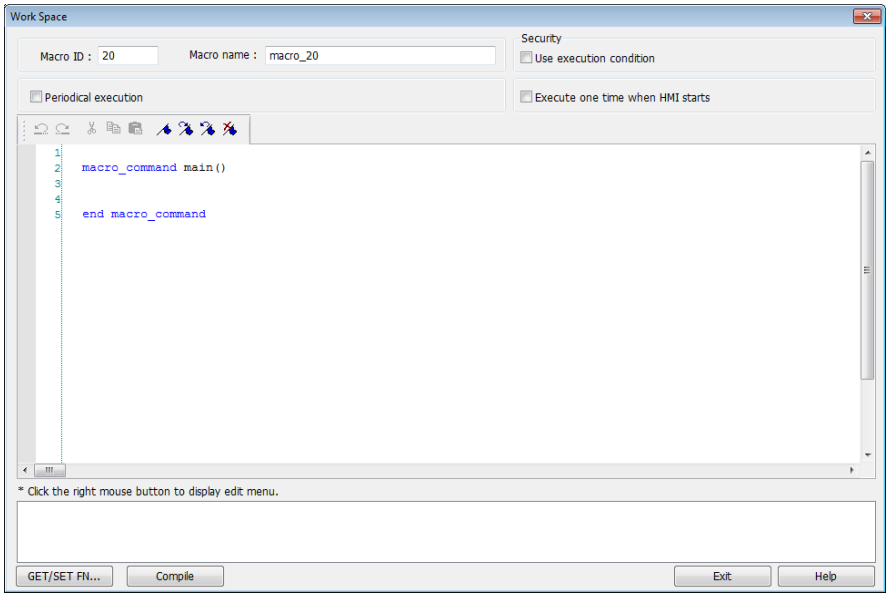


18.9.2. How to Use Macro Function Library

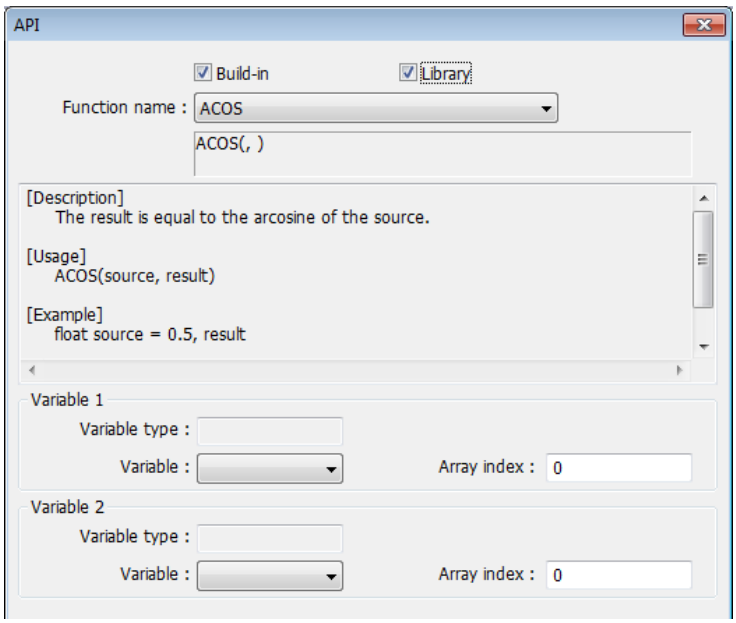
1. Select the function directly from Macro Function Library.



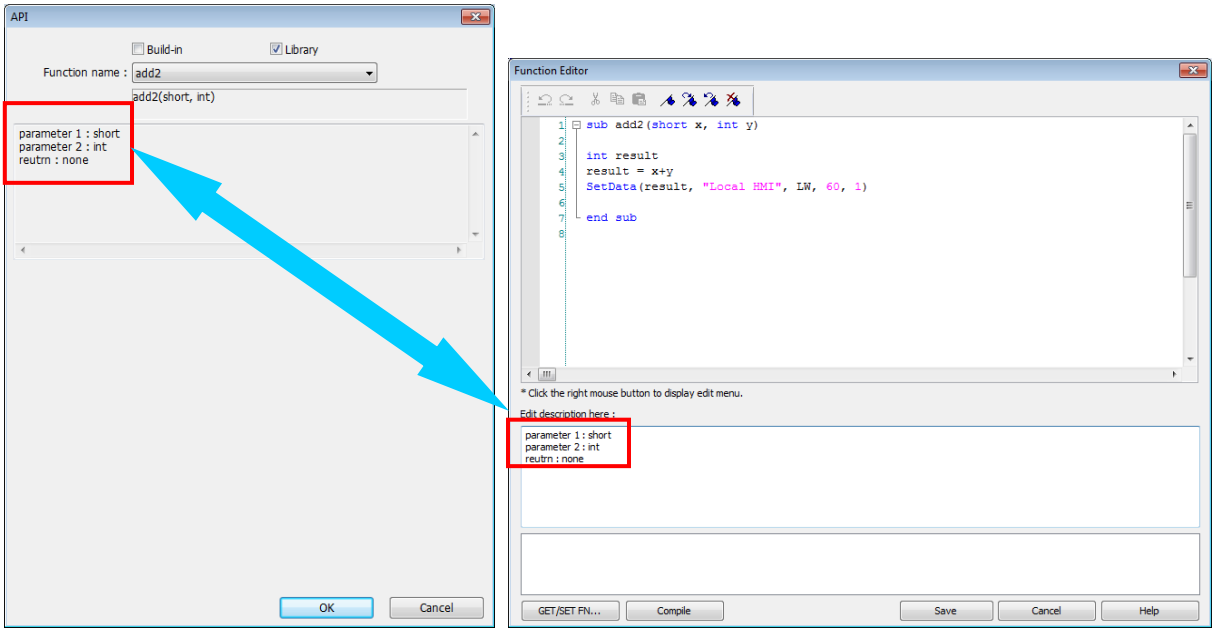
2. In WorkSpace click [GET/SET FN...] to open API dialog box.



3. At least check one from [Library] or [Build-in] and select the function to be used.



4. The description displayed in API dialog box is the same as written in Function Editor.



5. Select the function to be used, fill in the corresponding variables according to the data type.

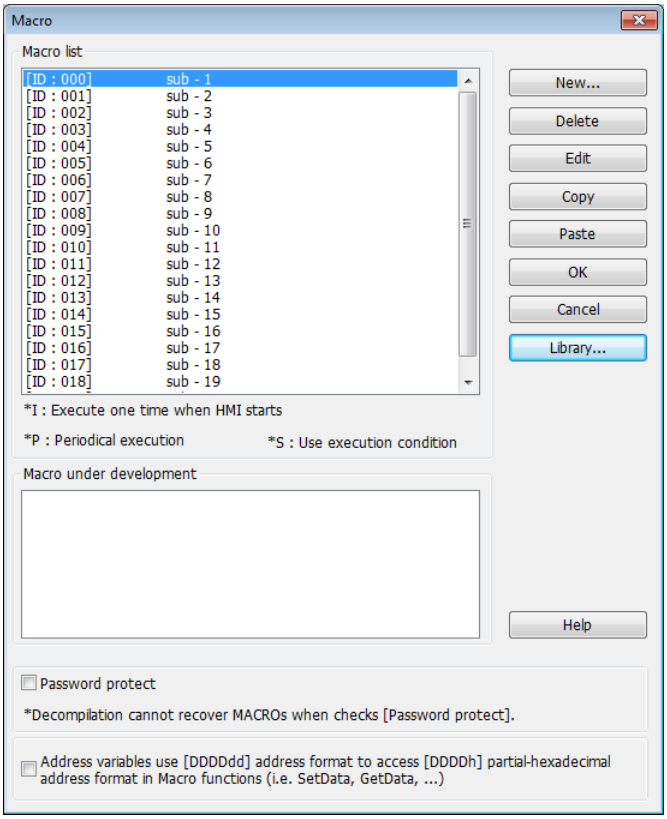
```
1
2  macro_command main()
3
4  short a
5  int b,result
6
7  add2(short, int)
8
9  end macro_command
```

```
1
2  macro_command main()
3
4  short a
5  int b,result
6
7  result = add2(a, b)
8
9  end macro_command
```

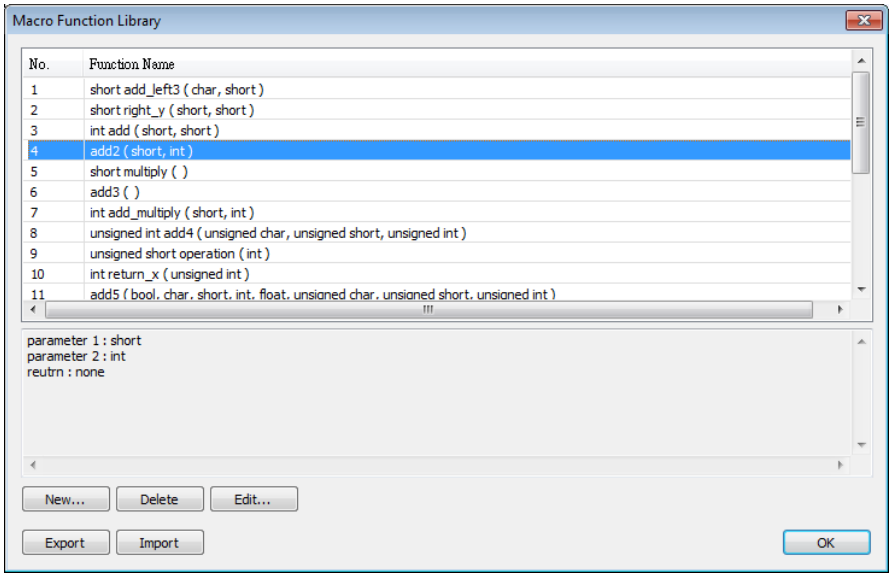
6. Upon completion of the steps above, user-defined functions can be used freely without defining the same functions repeatedly.

18.9.3. Function Library Management Interface

1. Open macro management dialog, click [Library] to open [Macro Function Library] dialog box.



2. A list of functions is shown. When the project is opened, the software will load all the functions in the Macro Function Library.



3. Each listed function has the following format:

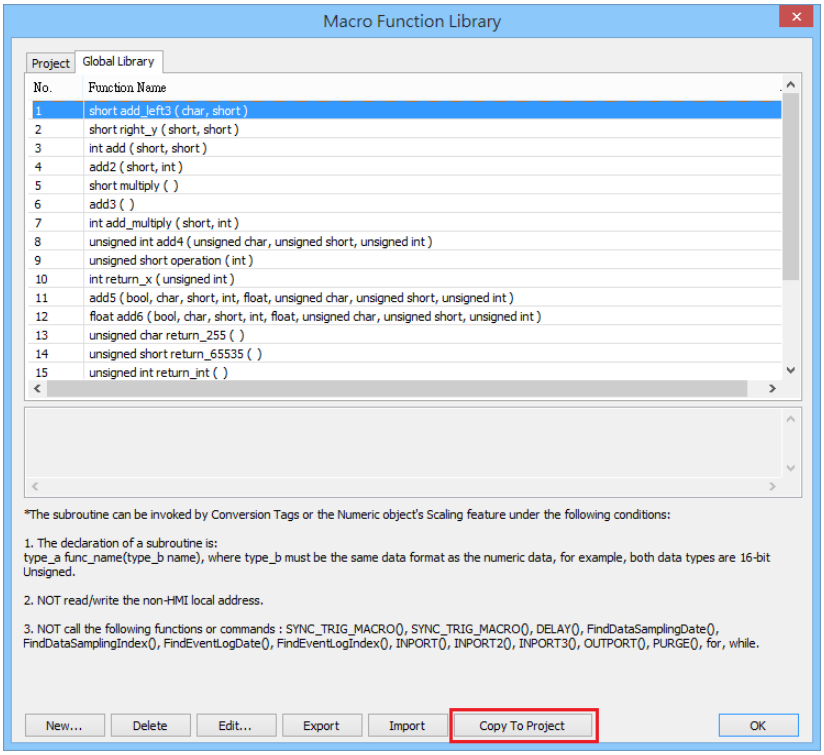
return_type function_name (*parameter_type1*, …, *parameter_typeN*)

return_type indicates the type of the return value. If this value does not exist, this column will be omitted. function_name indicates the name of the function. “N” in *parameter_typeN* stands for the number of parameter types. If this function does not

need any parameter, this column will be omitted.

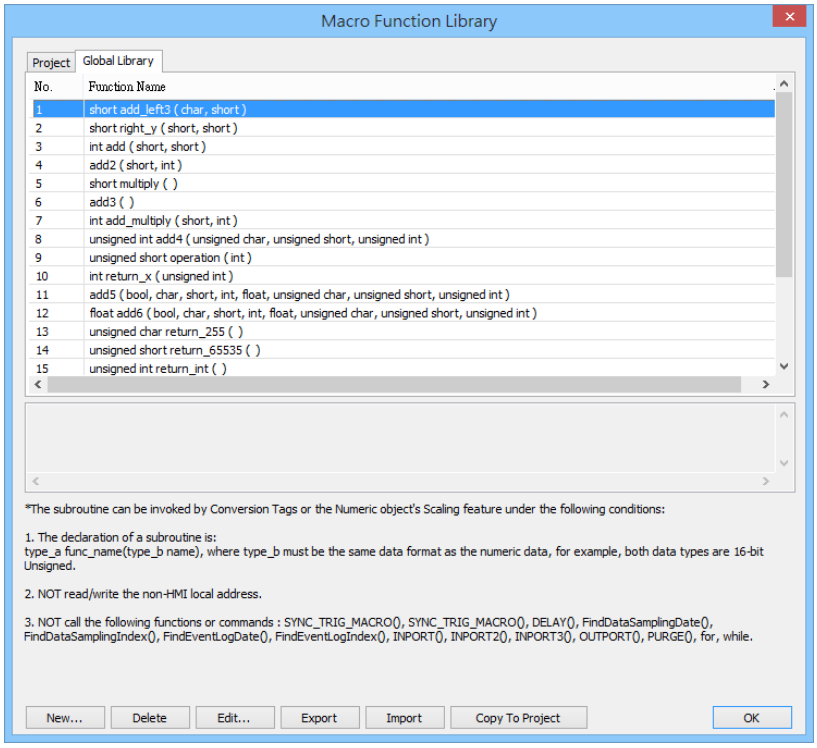
```
1  sub int ADD(int a, int b)
2    int ret
3    ret = a+b
4    return ret
5  end sub
6
```

4. Macro function can be embedded in the project file. Select the function and then click [Copy To Project], then you can find this function in [Project] tab. When opening the project on another computer, this function can still be used. When compiling the project, the .exob file will included the functions that are used. Please note that decompiling the project will only produce the macro commands that are used.

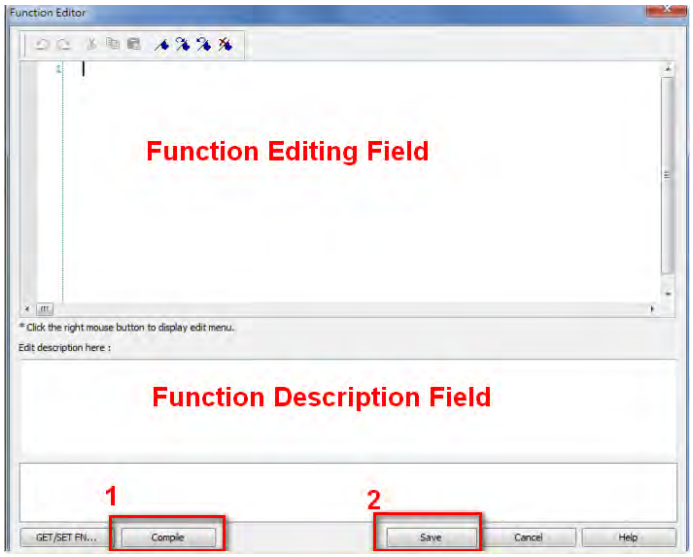


18.9.3.1. Create a Function

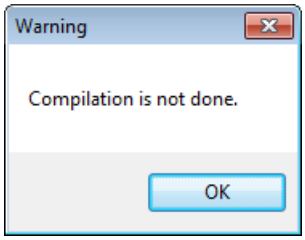
1. Click [New] to enter Function Editor.



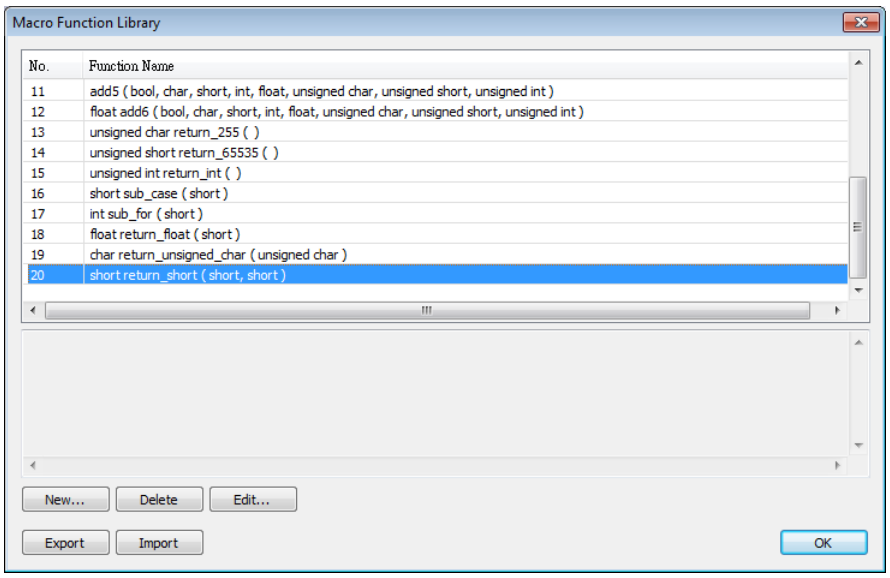
2. Edit function in Function Editor.



3. Edit the function description to describe what the specification is, how to use … etc.
4. After editing, click [Compile] and [Save] to save this function to the Library. Otherwise, a warning is shown.



5. Successfully add a function into Macro Function Library.

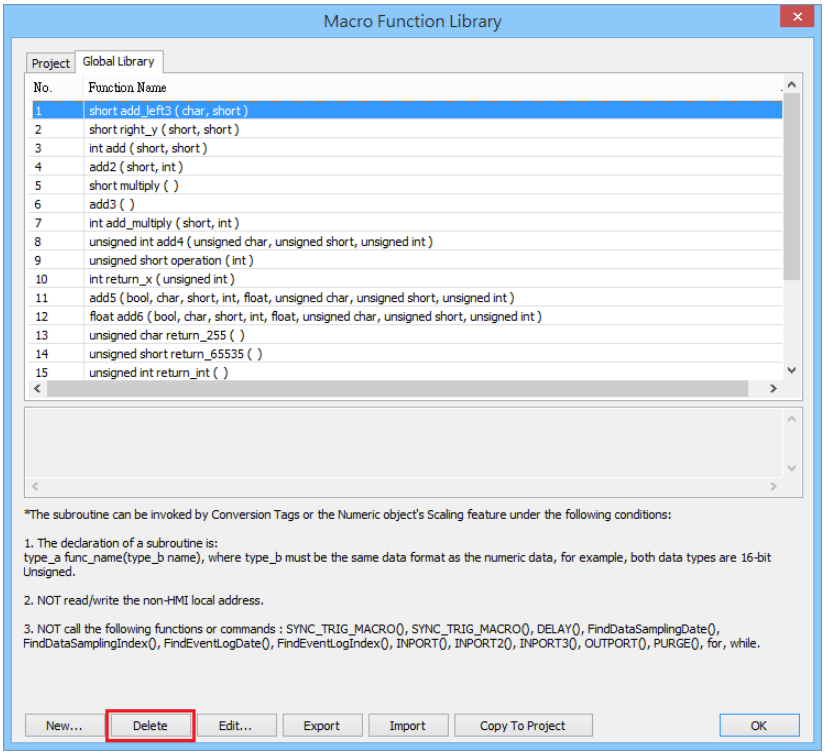


Note

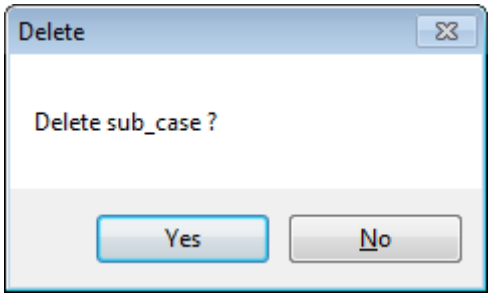
- The total size of data type can be declared in a function is 4096 bytes.
- Function name must only contain alphanumeric characters, and cannot start with a number.

18.9.3.2. Delete a Function

1. In function list select the function to be deleted and click [Delete].

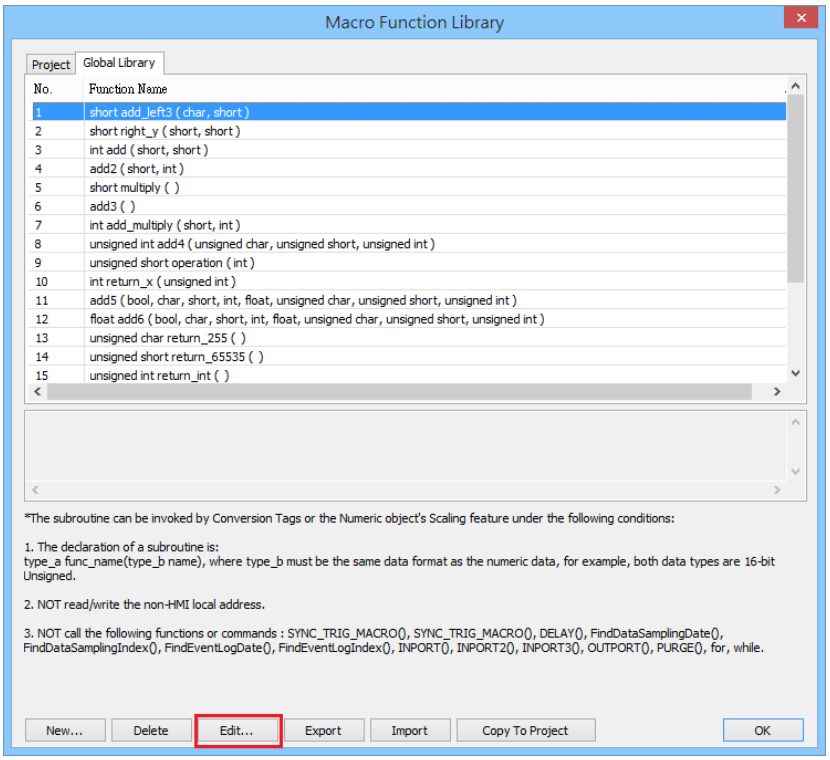


2. Click [Yes] to confirm, [No] to cancel the deletion. Click [Yes] to delete MAX_SHORT function.

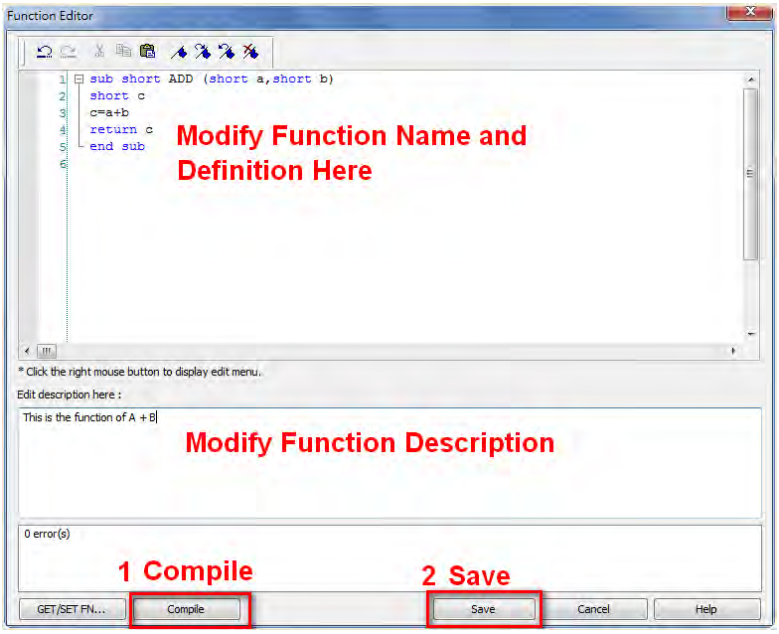


18.9.3.3. Modify a Function

1. Users can modify the functions exist in the Library.
2. Select a function to modify by clicking [Edit] to enter Function Editor.



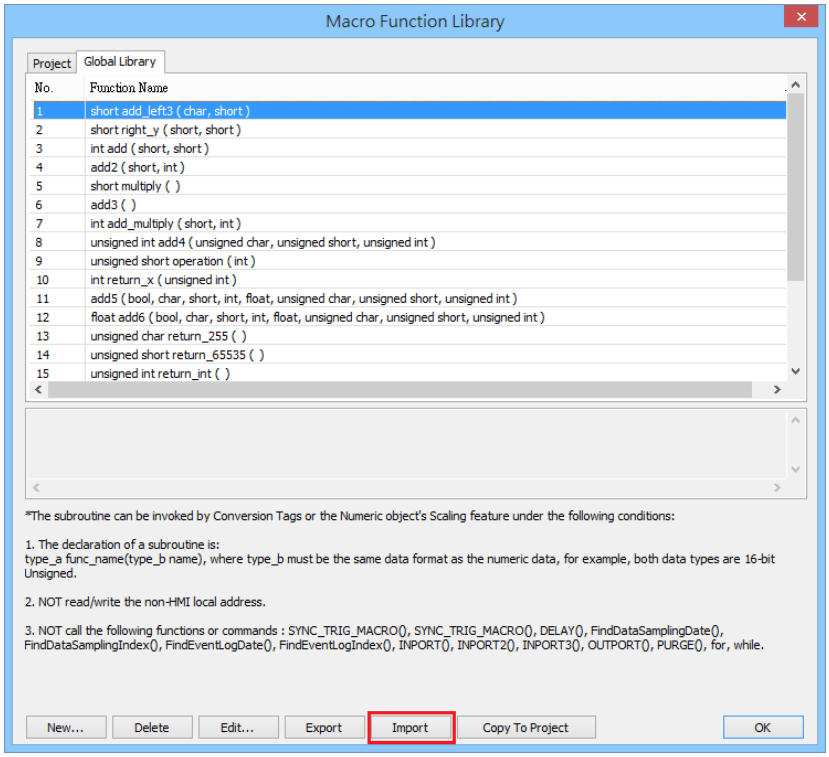
3. Double click the function to be modified can also enter Function Editor.



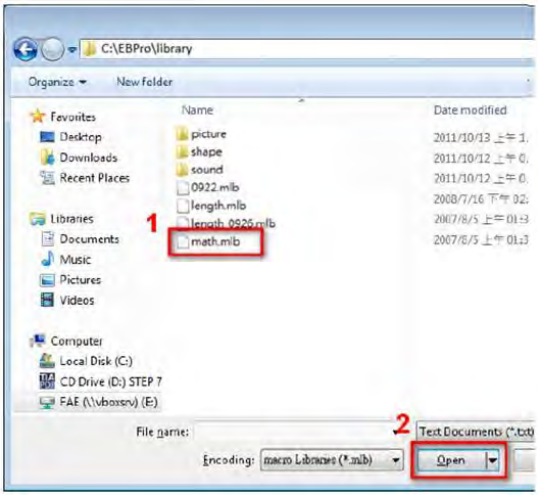
4. After modifying, [Compile] then [Save] before leaving.

18.9.3.4. Import a Function

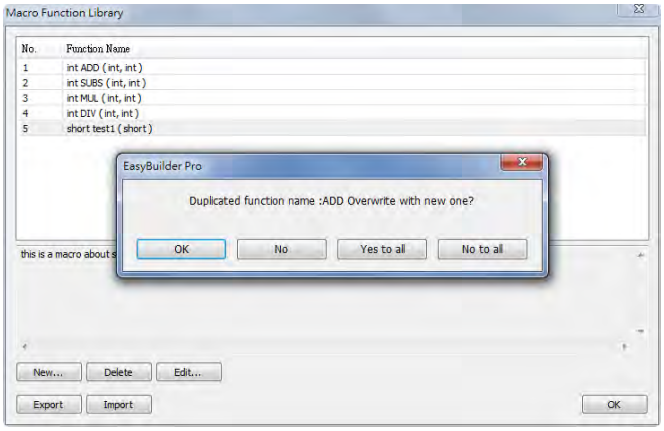
1. Functions can be imported using an external .mlb file.



2. For example, import a function library “math.mlb” which contains a function “test1”. Click [Open].



3. When importing a function which already exists in the Library, a confirmation pop-up will be shown. The buttons are:



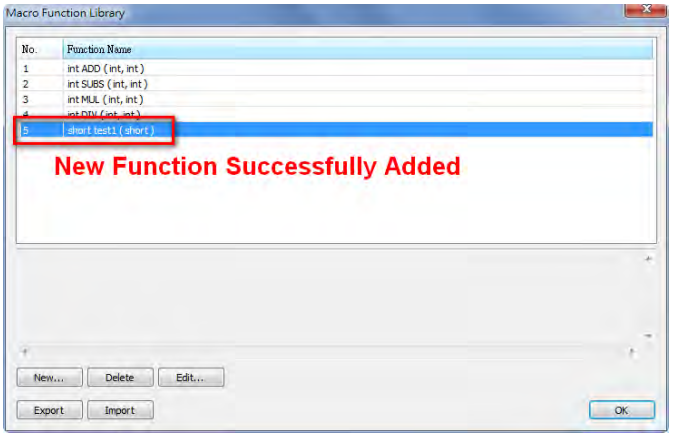
[OK]: Overwrite the existing function with the imported one.

[NO]: Cancel the importing of the function with the same name.

[Yes to all]: Overwrite using all the imported functions with the same name.

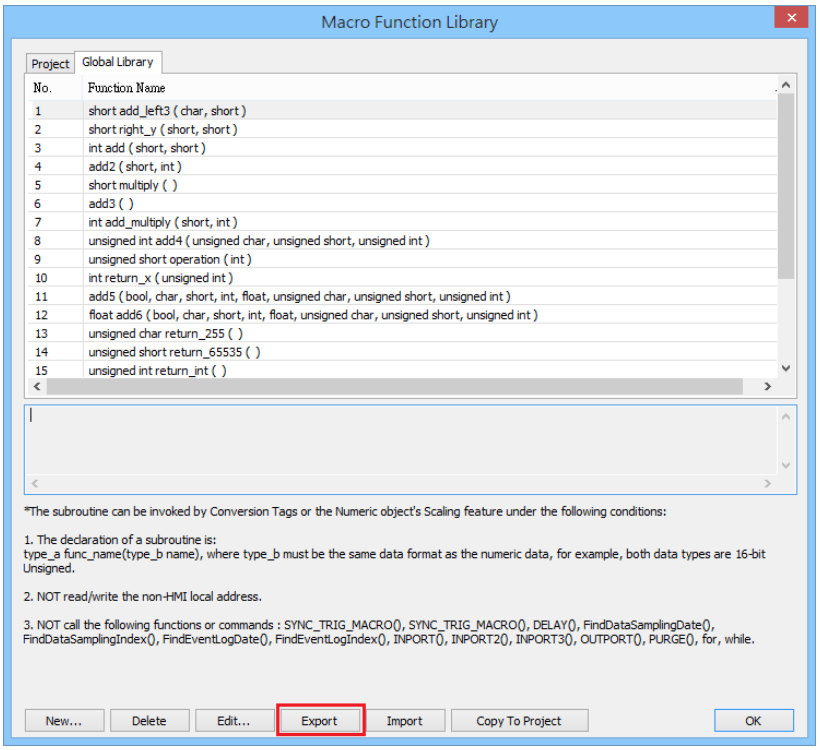
[No to all]: Cancel the importing of all the functions with the same name.

4. The imported functions will be saved in Default Function Library, so if “math.mlb” file is deleted, “test1” will still exist in the Library, even restarting EasyBuilder Pro.

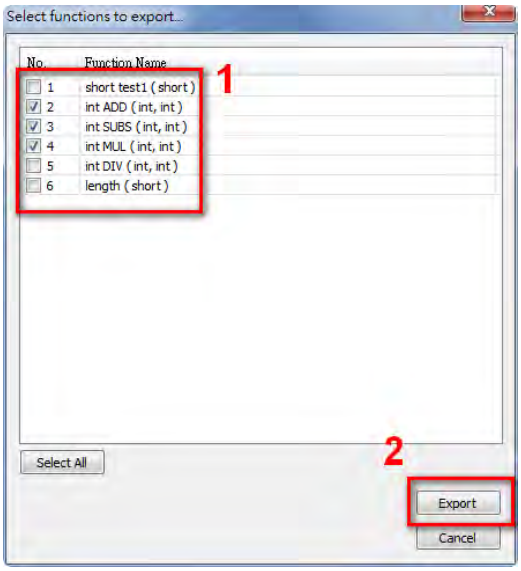


18.9.3.5. Export a Function

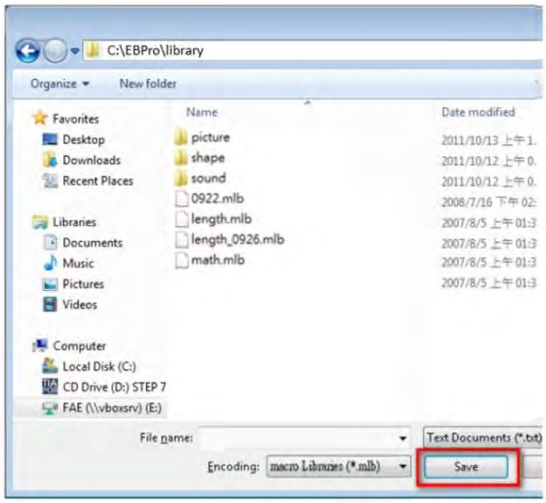
1. Export the function from Function Library and save as .mlb file. Click [Export].



2. Select the function to be exported, and click [Export].



3. A "math.mlb" file can be found under export directory. This file contains 4 functions: ADD, SUBS, MUL, and DIV.
4. The exported .mlb file can be imported on another PC. Open HMI programming software, import, then the functions in this file can be used.



18.10. Some Notes about Using the Macro

1. The maximum storage space of local variables in a macro is 4K bytes. So the maximum array size of different variable types are as follows:

char	a[4096]
bool	b[4096]
short	c[2048]
int	d[1024]
float	e[1024]
long	f[512]
double	g[512]

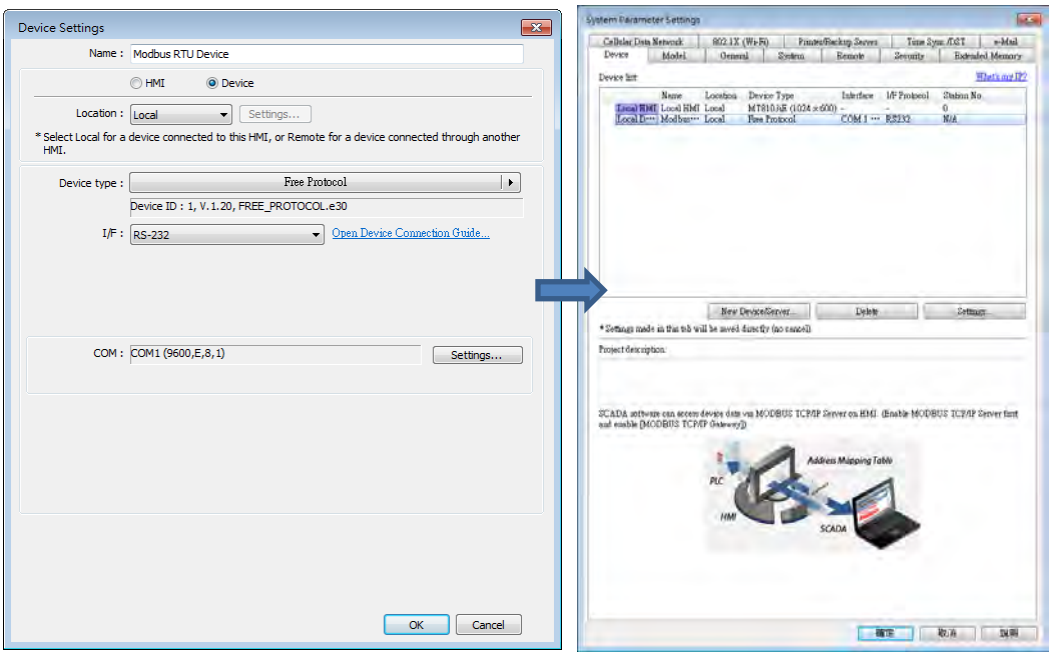
2. A maximum of 255 macros are allowed in an EasyBuilder Pro project. However, for cMT X Series projects, that number is increased to 500.
3. A macro may cause the HMI to be unresponsive. Possible reasons may include:
   - It contains an undesired infinite loop.
   - Array size exceeds the available variable storage space in a macro.
4. The device communication speed may affects execution speed of the macro . Similarly, having too many macros may slow down the communication between an HMI and a device.

18.11. Use the Free Protocol to Control a Device

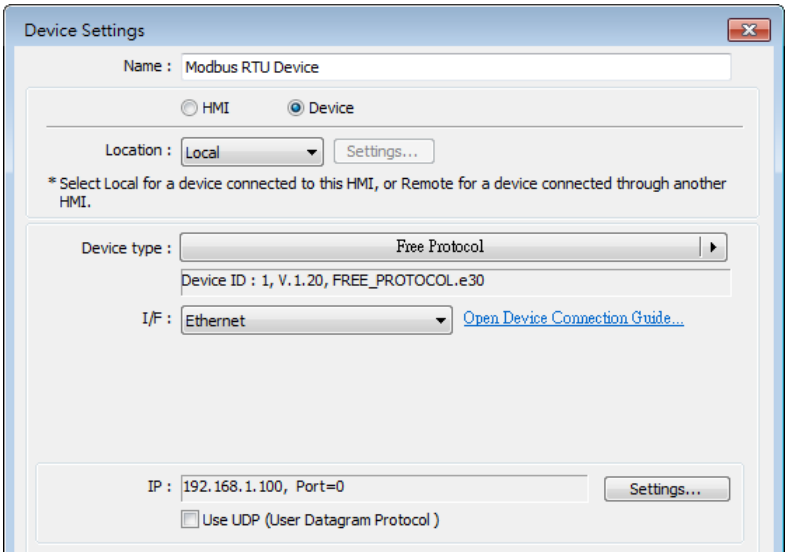
If EasyBuilder Pro does not provide a driver for a specific device, users can use OUTPORT and INPORT built-in functions to control the device. The data sent by OUTPORT and INPORT must follow the communication protocol of the device. The following example explains how to use

these two functions to control a MODBUS RTU device.

1. First, create a new device in the device table. The device type of the new device is set to "Free Protocol" and named with "MODBUS RTU device" as follows:



2. The interface of the device (I/F) uses [RS-232]. If a MODBUS TCP/IP device is connected, the interface should be [Ethernet] with correct IP and port number as follows:



Suppose that the HMI will read the data of 4x_1 and 4x_2 on the device. First, utilize OUTPORT to send out a read request to the device. The format of OUTPORT is:

OUTPORT(command[start], device_name, cmd_count)

Since "MODBUS RTU device" is a MODBUS RTU device, the read request must follow MODBUS RTU protocol. The request uses "Reading Holding Registers (0x03)" command to read data. The following picture displays the content of the command. (The items of the station number (byte 0) and the last two bytes (CRC) are ignored).

Request

Function code	1 Byte	0x03
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	1 to 125 (0x7D)

Response

Function code	1 Byte	0x03
Byte count	1 Byte	2 x N*
Register value	N* x 2 Bytes	

*N = Quantity of Registers

Error

Error code	1 Byte	0x83
Exception code	1 Byte	01 or 02 or 03 or 04

Depending on the protocol, the content of a read command as follows (The total is 8 bytes):

command[0]: station number (BYTE 0)
command[1]: function code (BYTE 1)
command[2]: high byte of starting address (BYTE 2)
command[3]: low byte of starting address (BYTE 3)
command[4]: high byte of quantity of registers (BYTE 4)
command[5]: low byte of quantity of registers (BYTE 5)
command[6]: low byte of 16-bit CRC (BYTE 6)
command[7]: high byte of 16-bit CRC (BYTE 7)

So a read request is designed as follows:

```
char command[32]
short address, checksum

FILL(command[0], 0, 32)    //    initialize command[0]~command[31] to 0

command[0] = 0x1    //    station number
command[1] = 0x3    //    read holding registers (function code    is 0x3)

address =    //    starting address (4x_1) is 0
HIBYTE(address, command[2])
LOBYTE(address, command[3])

read_no = 2    //    the total words of reading is 2 words
HIBYTE(read_no, command[4])
LOBYTE(read_no, command[5])

CRC(command[0], checksum, 6)    //    calculate 16-bit CRC

LOBYTE(checksum, command[6])
HIBYTE(checksum, command[7])
```

Lastly, use OUPORT to send out this read request to the device.

```
OUTPORT(command[0], “MODBUS RTU Device”, 8)    //    send read request
```

After sending out the request, use INPORT to get the response from the device. Depending on the protocol, the content of the response is as follows (the total byte is 9):

response[0]: station number (BYTE 0)
response[1]: function code (BYTE 1)
response[2]: byte count (BYTE 2)
response[3]: high byte of 4x_1 (BYTE 3)
response[4]: low byte of 4x_1 (BYTE 4)
response[5]: high byte of 4x_2 (BYTE 5)
response[6]: high byte of 4x_2 (BYTE 6)
response[7]: low byte of 16-bit CRC (BYTE 7)
response[8]: high byte of 16-bit CRC (BYTE 8)

The format of INPORT is:

```
INPORT(response[0], “MODBUS RTU Device”, 9, return_value)    //    read response
```

Where the real read count is restored to the variable return_value (unit is byte). If return_value is 0, it means reading fails in executing INPORT.

According to the MODBUS RTU protocol specification, the correct response[1] must be equal to 0x03. After getting correct response, calculate the data of 4x_1 and 4x_2 and put in the data into LW-100 and LW-101 of HMI.

```
If (return_value) >0 and response[1] == 0x3) then
    read_data[0] = response[4] + (response[3] << 8) // 4x_1
    read_data[1] = response[6] + (response[5] << 8) // 4x_2

    SetData(read_data[0], “Local HMI”, LW, 100, 2)
endif
```

The complete macro is as follows:

```
// Read Holding Registers
macro_command main()

    char command[32], response[32]
    short address, checksum
    short read_no, return_value, read_data[2], i

    FILL(command[0], 0, 32)//    initialize command[0]~command[31] to 0
    FILL(response[0], 0, 32)

    command[0] = 0x1//    station number
    command[1] = 0x3//    read holding registers (function code is 0x3)

    address = 0
    address =    0//    starting address (4x_1) is 0
    HIBYTE(address, command[2])
    LOBYTE(address, command[3])

    read_no = 2/    the total words of reading is 2 words
    HIBYTE(read_no, command[4])
    LOBYTE(read_no, command[5])

    CRC(command[0], checksum, 6)//    calculate 16-bit CRC

    LOBYTE(checksum, command[6])
    HIBYTE(checksum, command[7])

    OUTPORT(command[0], "MODBUS RTU Device", 8 )//    send request
    INPORT(response[0], "MODBUS RTU Device", 9, return_value)//    read response

    if (return_value > 0 and response[1] == 0x3) then
        read_data[0] = response[4] + (response[3] << 8)//    4x_1
        read_data[1] = response[6] + (response[5] << 8)//    4x_2

        SetData(read_data[0], "Local HMI", LW, 100, 2)
    end if

    end macro_command
```

The following example explains how to design a request to set the status of 0x_1. The request uses ”Write Single Coil(0x5)” command.

Request

Function code	1 Byte	**0x05**
Output Address	2 Bytes	0x0000 to 0xFFFF
Output Value	2 Bytes	0x0000 or 0xFF00

Response

Function code	1 Byte	**0x05**
Output Address	2 Bytes	0x0000 to 0xFFFF
Output Value	2 Bytes	0x0000 or 0xFF00

Error

Error code	1 Byte	**0x85**
Exception code	1 Byte	01 or 02 or 03 or 04

The complete macro is as follows:

```
//    Write Single Coil (ON)
macro_command main()

char command[32], response[32]
short address, checksum
short i, return_value

FILL(command[0], 0, 32)//    initialize command[0]~ command[31] to 0
FILL(response[0], 0, 32)

command[0] = 0x1//    station number
command[1] = 0x5//    function code : write single coil

address = 0
HIBYTE(address, command[2])
LOBYTE(address, command[3])

command[4] = 0xff//    force 0x_1 on
command[5] = 0

CRC(command[0], checksum, 6)

LOBYTE(checksum, command[6])
HIBYTE(checksum, command[7])

OUTPORT(command[0], "MODBUS RTU Device", 8)//    send request
INPORT(response[0], "MODBUS RTU Device", 8, return_value)//    read response

end macro_command
```

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

18.12. Compiler Error Message

- Error Message Format

error C# : error description
(# is the error message number)
Example: error C37 : undeclared identifier : i
When there are compile errors, the description of the error can be found by the compiler error message number.

- Error Description

(C1) syntax error：'identifier'
There are many possibilities to cause compiler error.

For example:
```
macro_command main()
char i, 123xyz    //    this is an unsupported variable name
end macro_command
```

(C2) 'identifier' used without having been initialized
Macro must define the size of an array during declaration.

For example:
```
macro_command main()
char i
int g[i]    //    i must be a numeric constant
end macro_command
```

(C3) redefinition error : 'identifier'
The name of variable and function within its scope must be unique.

For example:
```
macro_command main()
int g[10]，g    //    error
end macro_command
```

(C4) function name error : 'identifier'
Reserved keywords and constant cannot be the name of a function

For example：
```
sub int if()    // error
```

(C5) parentheses have not come in pairs
Statement missing “(“ or “)”
For example：
macro_command main) // missing “(“

(C6) illegal expression without matching ‘if’
Missing expression in “if” statement

(C7) illegal expression (no ‘then’) without matching ‘if’
Missing “then” in “if” statement

(C8) illegal expression (no ‘end if’)
Missing “end if”

(C9) illegal ‘end if’ without matching ‘if’
Unfinished “If’ statement before “End If”

(C10) illegal ‘else’
The format of “if” statement is :
if [logic expression] then
[else [if [logic expression] then]]

end if

Any format other than this format will cause a compile error.

(C17) illegal expression (no 'for') without matching ‘next’
“for” statement error : missing “for” before “next”

(C18) illegal variable type (not integer or char)
Should be integer or char variable

(C19) variable type error
Missing assign statement

(C20) must be keyword ‘to’ or ‘down’
Missing keyword “to” or “down”

(C21) illegal expression (no 'next')
The format of “for” statement is:
for [variable] = [initial value] to [end value] [step]

next [variable]

Any format other than this format will cause a compile error.

(C22) ‘wend’ statement contains no ‘while’
“While” statement error : missing “while” before “Wend”

(C23) illegal expression without matching ‘wend’
The format of “While” statement is :

while [logic expression]

wend

Any format other than this format will cause a compile error.

(C24) syntax error : ‘break’
“break” statement can only be used in “for”, “while” statement.

(C25) syntax error : ‘continue’
“continue” statement can only be used in “for” statement, or “while” statement.

(C26) syntax error
Error in expression.

(C27) syntax error
The mismatch of an operation object in expression can cause a compile error.

For example :

```
macro_command main( )
int a, b
for a = 0 to 2
b = 4 + xyz //    illegal : xyz is undefined
next a
```

end macro_command

(C28) must be ‘macro_command’
There must be ’macro_command’

(C29) must be key word ‘sub’
The format of function declaration is:

```
sub [data type] function_name(…)
    ……..
end sub
```

For example::

```
sub int pow(int exp)
…….
end sub
```

format other than this format will cause a compile error.

(C30) number of parameters is incorrect
Mismatch of the number of parameters

(C31) parameter type is incorrect
Mismatch of data type of parameter. When a function is called, the data type and the number of parameters should match the declaration of function, otherwise it will cause a compile error.

(C32) variable is incorrect
The parameters of a function must be equivalent to the arguments passing to a function to avoid compile error.

(C33) function name : undeclared function

(C34) expected constant expression
Illegal array index format.

(C35) invalid array declaration

(C36) array index error

(C37) undeclared identifier : i ‘identifier’
Any variable or function should be declared before use.

(C38) un-supported device data address
The parameter of GetData(…) , SetData(…) should be legal device address. If the address is illegal, this error message will be shown.

(C39) ‘identifier’ must be integer, char or constant
The format of array is:
Declaration: array_name[constant] (constant is the size of the array)
Usage: array_name[integer, character or constant]
Any format other than this format will cause a compile error.

(C40) execution syntax should not exist before variable declaration or constant definition

For example :

```
macro_command main(    )
int a, b
for a = 0 To 2
    b = 4 + a
int h , k    //    illegal – definitions must occur before any statements or expressions
             //    for example, b = 4 + a
next a
end macro_command
```

(C41) float variables cannot be contained in shift calculation

(C42) function must return a value

(C43) function should not return a value

(C44) float variables cannot be contained in calculation

(C45) device address error

(C46) array size overflow (max. 4k)

(C47) macro command entry function is not only one

(C48) macro command entry function must be only one
The only one main entrance of macro is :
macro_command function_name( )
end macro_command

(C49) an extended addressee's station number must be between 0 and 255

For example :
SetData(bits[0] , "PLC 1", LB , 300#123, 100)
// illegal : 300#123 means the station number is 300, but the maximum is 255

(C50) an invalid device name
Device name is not defined in the device list of system parameters.

(C51) macro command do not control a remote device
A macro can only control a local machine.

For example :
SetData(bits[0] , "PLC 1", LB , 300#123, 100)
"PLC 1" is connected with the remote HMI ,so it cannot work.

18.13. Sample Macro Code

- "for" statement and other expressions (arithmetic, bitwise shift, logic and comparison)

```
macro_command main()
int a[10], b[10], i

b[0] = (400 + 400 << 2) / 401
b[1] = 22 *2 - 30 % 7
b[2] = 111 >> 2
b[3] = 403 > 9 + 3 >= 9 + 3 < 4 + 3 <= 8 + 8 == 8
b[4] = not 8 + 1 and 2 + 1 or 0 + 1 xor 2
b[5] = 405 and 3 and not 0
b[6] = 8 & 4 + 4 & 4 + 8 | 4 + 8 ^ 4
b[7] = 6 – (~4)
b[8] = 0x11
```

```
b[9] = 409

for i = 0 to 4 step 1
  if (a[0] == 400) then
        GetData(a[0], "Device 1", 4x, 0,9)
        GetData(b[0],”Device 1", 4x, 11,10)
  end If
  next i
  end macro_command
```

- “while”, “if” and “break” statements

```
macro_command main()
int b[10], i
i = 5
while i == 5 - 20 % 3
      GetData(b[1], "Device 1", 4x, 11, 1)

      if b[1] == 100 then
            break
      end if
wend
end macro_command
```

- Global variables and function call

```
char g
sub int fun(int j, int k)
      int y

      SetData(j, "Local HMI", LB, 14, 1)
      GetData(y, "Local HMI", LB, 15, 1)
      g = y

      return y
end Sub

macro_command main()
      int a, b, i
```

```
a = 2
b = 3
    i = fun(a, b)
    SetData(i, "Local HMI", LB, 16, 1)
end macro_command
```

- "if" statement

```
macro_command main()
    int k[10], j

    for j = 0 to 10
        k[j] = j
    next j

    if k[0] == 0 then
    SetData(k[1], "Device 1", 4x, 0, 1)
    end if

if k[0] == 0 then
        SetData(k[1], "Device 1", 4x, 0, 1)
    else
    SetData(k[2], "Device 1", 4x, 0, 1)
end if

    if k[0] == 0 then
        SetData(k[1], "Device 1", 4x, 1, 1)
    else if k[2] == 1 then
    SetData(k[3], "Device 1", 4x, 2, 1)
end If

    if k[0] == 0 then
    SetData(k[1], "Device 1", 4x, 3, 1)
else if k[2] == 2 then
    SetData(k[3], "Device 1", 4x, 4, 1)
else
    SetData(k[4], "Device 1", 4x, 5, 1)
end If
end macro_command
```

- "while" and "wend" statements

```
macro_command main()
char i = 0
int a[13], b[14], c = 4848

b[0] = 13

while b[0]
        a[i] = 20 + i * 10

    if a[i] == 120 then
        c =200
            break
    end if

    i = i + 1
wend

SetData(c, "Device 1", 4x, 2, 1)
end macro_command
```

- "break" and "continue" statements

```
macro_command main()
char i = 0
int a[13], b[14], c = 4848

b[0] = 13

while b[0]
        a[i] = 20 + i * 10

        if a[i] == 120 then
        c =200
        i = i + 1
            continue
        end if
```

```
        i = i + 1

        if c == 200 then
            SetData(c, "Device 1", 4x, 2, 1)
        break
            end if
wend
end macro_command
```

- Array

```
macro_command main()
int a[25], b[25], i

b[0] = 13

for i = 0 to b[0] step 1
        a[i] = 20 + i * 10
next i

SetData(a[0], "Device 1", 4x, 0, 13)
end macro_command
```

- Syntax for placing quotation marks in a string applies to variable declaration and function's argument.

```
macro_command main()
char data[40]= "\"Note\" "

StringCopy("This is a \"test\" for weintek", data[7])
//The string contains "Note" This is a "test" for weintek
end macro_command
```

18.14. Macro TRACE Function

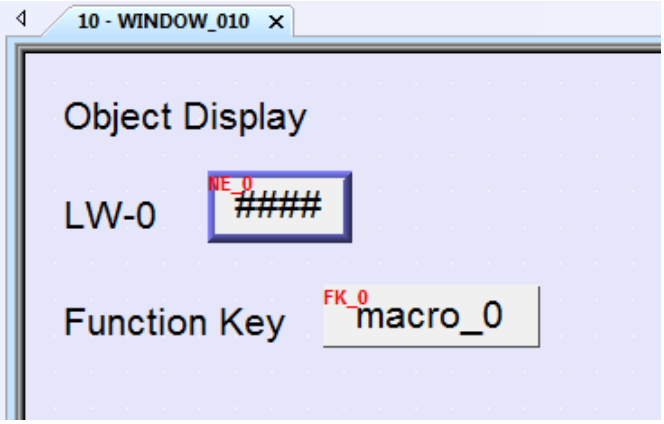
TRACE function can be used with EasyDiagnoser / cMT Diagnoser to show the current content of the variables. For users of cMT /cMT X series, a better, more straightforward way would be to use the Macro Debugger in the cMT Diagnoser for debugging. The use of TRACE function is not required.

The following example explains how to use TRACE function in macro and then use EasyDiagnoser for monitoring.

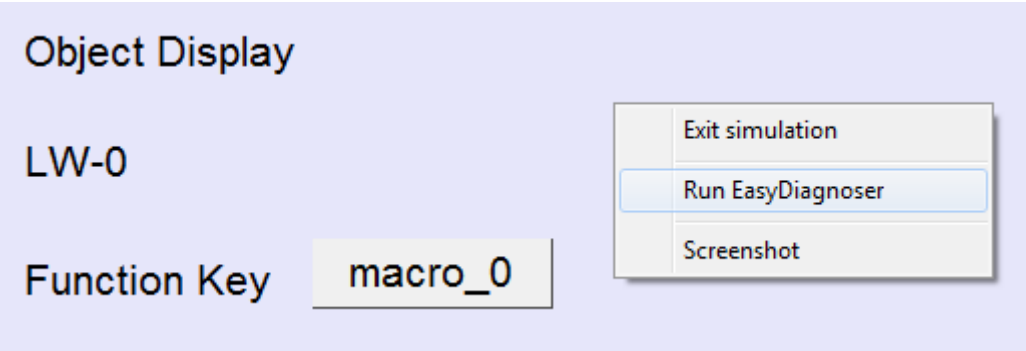
1. First of all, add a new macro “macro_0” in the project, and in “macro_0” add TRACE (“LW = %d”, a). “%d” indicates display current value of LW in decimal format. The content of “macro_0” is as follows:

```
1
2   macro_command main()
3
4   short a
5   GetData(a, "Local HMI", LW, 0, 1)
6   a=a+1
7   SetData(a, "Local HMI", LW, 0, 1)
8   TRACE ("LW0 = %d" , a)
9
10  end macro_command
```

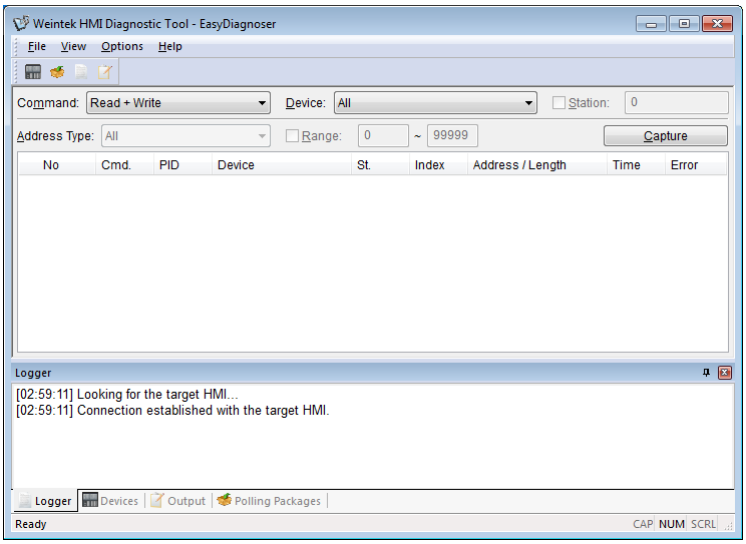
2. Secondly, add a Numeric Display object and a Function Key object in window no. 10 of the project. The Function Key object is used to execute macro_0.



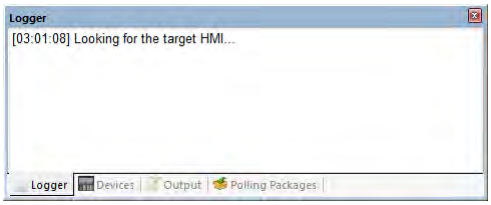
3. Lastly, compile the project and execute [Off-line simulation] or [On-line simulation].
4. When processing simulation on PC, right click and select “Run EasyDiagnoser” in the pop-up menu.



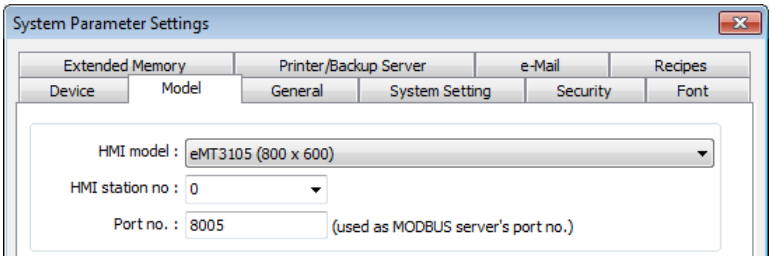
5. Afterwards, EasyDiagnoser will be started. [Logger] window displays whether EasyDiagnoser is able to connect with the HMI to be watched or not. [Output] window displays the output of the TRACE function. The illustration below shows that EasyDiagnoser succeeds in connecting with HMI.



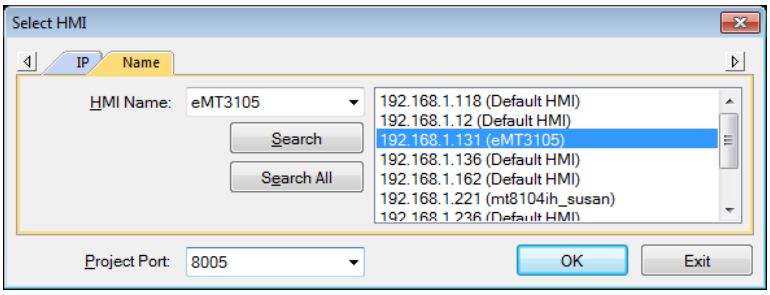
When EasyDiagnoser is not able to connect with HMI, [Logger] window displays content as shown in the following figure:



6. The possible reason of not being able to get connection with HMI can be failure in executing simulation on PC. Another reason is that the Port No. used in project for simulation on PC is incorrect (or occupied by system). Please change Port No. as shown, compile project then do simulation again.



7. In EasyDiagnoser, the Port No. should be set the same as the Port No. in the project.



The three consecutive ports of the project port no. are preserved for HMI communication. In the setting above as an example, Port No. is set as 8005. Port 8005, 8006 and 8007 should be reserved. In this case when executing simulation on PC, please make sure that these ports are not occupied by other programs.

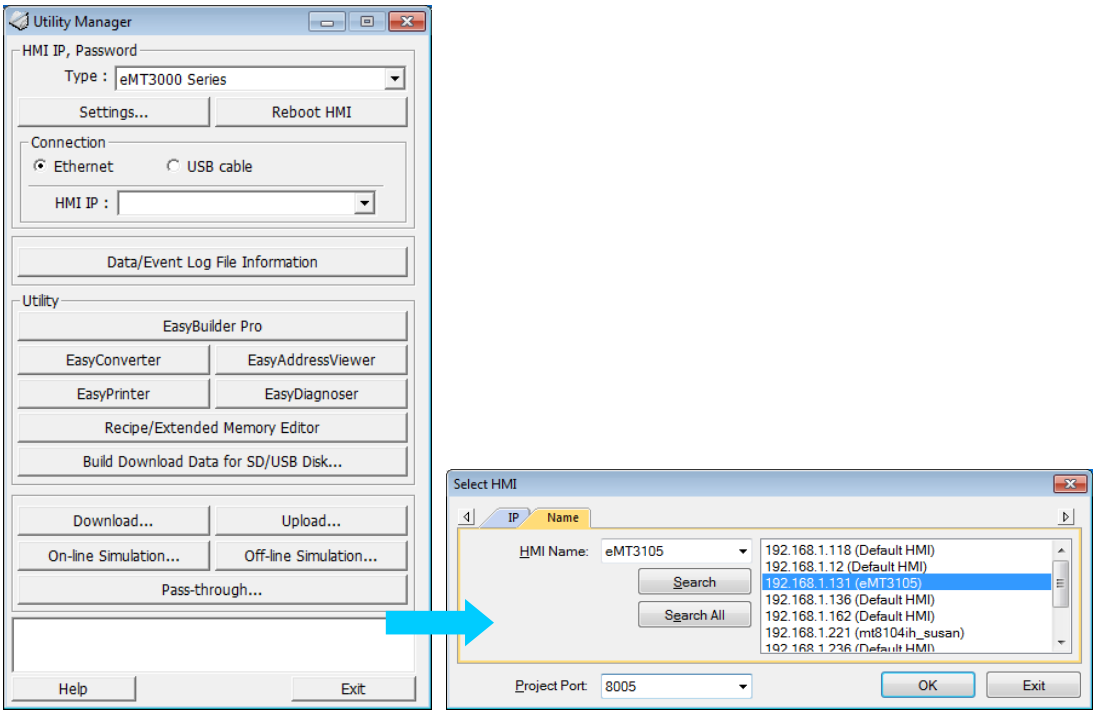
TRACE Syntax List

Name	TRACE
Syntax	TRACE(format, argument)
Description	Use this function to send specified string to the EasyDiagnoser / cMT Diagnoser. Users can print out the current value of variables during run-time of macro for debugging. When TRACE encounters the first format specification (if any), it converts the value of the first argument after format and outputs it accordingly. format refers to the format control of output string. A format specification, which consists of optional (in [ ]) and required fields (in red font), has the following form: %[flags] [width] [.precision] type Each field of the format specification is described as below: *flags* (optional): - : Aligns left. When the value has fewer characters than the specified width, it will be padded with spaces on the left. + : Precedes the result with a plus or minus sign (+ or -) *width* (optional): A nonnegative decimal integer controlling the minimum number of characters printed. *precision* (optional): A nonnegative decimal integer which specifies the precision and the number of characters to be printed. *type*: C or c : specifies a single-byte character d : signed decimal integer

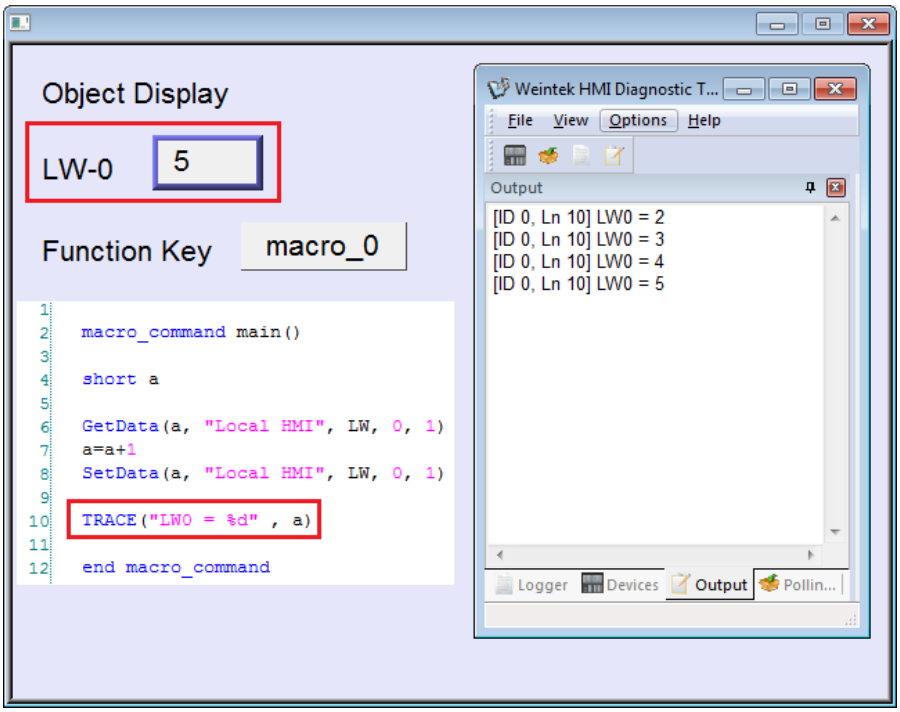
	i : signed decimal integer o : unsigned octal integer u : unsigned decimal integer X or x : unsigned hexadecimal integer lld : signed long integer (64-bit) (cMT / cMT X Series only) llu : unsigned long integer (64-bit) (cMT / cMT X Series only) f : signed floating-point value llf : double-precision floating-point value E or e : Scientific notation in the form "[–]d.dddd **e** [sign]ddd" , where d is a single decimal digit, dddd is one or more decimal digits, ddd is exactly three decimal digits, and sign is + or –. The length of output string is limited to 256 characters. Extra characters will be ignored. The *argument* part is optional. One format specification converts exactly one argument.

Example	macro_command main()
	char c1 = ' a'
	short s1 = 32767
	float f1 = 1.234567
	
	TRACE("The results are") // output: The results are
	TRACE("c1 = %c, s1 = %d, f1 = %f", c1, s1, f1)
	// output: c1 = a, s1 = 32767, f1 = 1.234567
	
	end macro_command

8. Use LB-9059 to disable MACRO TRACE function (when ON). When set ON, the output message of TRACE won't be sent to EasyDiagnoser.
9. Users can directly execute EasyDiagnoser.exe from Utility Manager. In Utility Manager, current HMI on line will be listed; users can simply select the HMI to be watched. Please note that Project Port should be the same as Port No. used in project file.

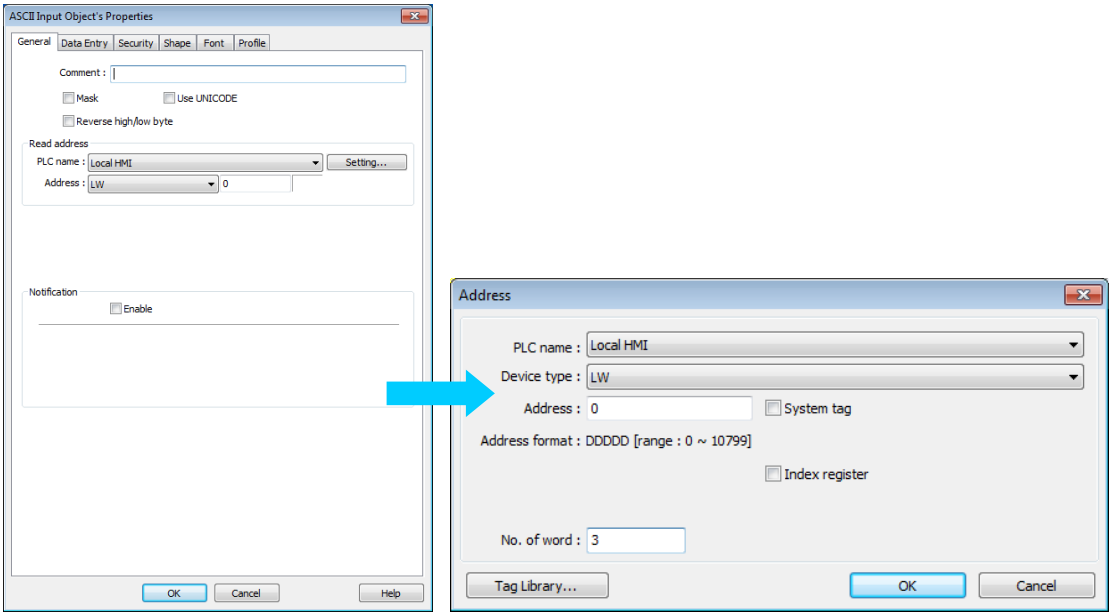


10. Download the project to HMI and start the project. If EasyDiagnoser is unable to get connection with the HMI to be watched, it is possible that HMI power is not ON, or Port No. is incorrect. This may cause EasyDiagnoser to connect then disconnect with HMI continuously. Please check the Port No. in EasyDiagnoser settings.
11. When EasyDiagnoser succeeds in connecting with HMI, simply execute macro_0, [Output] window will then display the output of the TRACE function.



18.15. Example of String Operation Functions

String operation functions are added to macro to provide a convenient way to operate strings. The term “string” means a sequence of ASCII characters, and each of them occupies 1 byte. The sequence of characters can be stored into 16-bit registers with least significant byte first. For example, create an ASCII Input object and setup as follows:



Run simulation and input “abcdef”:

The string “abcdef” is stored in LW-0~LW-2 as follows (LB represents low byte and HB represents high byte):

	HB	LB
LW0	'B'	'A'
LW1	'D'	'C'
LW2	'F'	'E'
LW3		
LW4		
LW5		

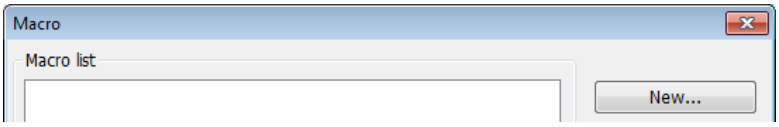
The ASCII Input object reads 1 word (2 bytes) at a time as described in the previous chapter. Suppose an ASCII Input object is set to read 3 words as shown in the above example, it can actually read at most 6 ASCII characters since that one ASCII character occupies 1 byte.

In order to demonstrate the powerful usage of string operation functions, the following examples will show you step by step how to create executable project files using the new

functions; starts from creating a macro, ends in executing simulation.

1. To read (or write) a string from a device:

   Create a new macro:



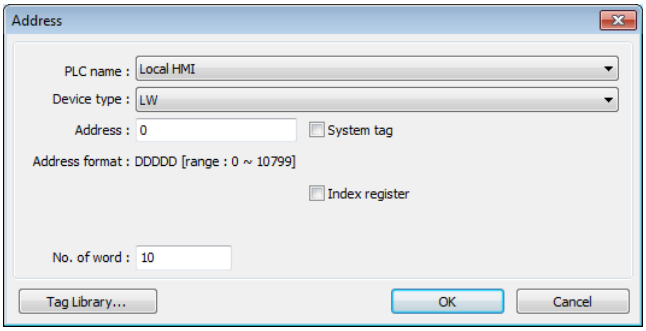
   Edit the content:

```
1
2  macro_command main()
3
4  char str[20]
5
6  StringGet(str[0], "Local HMI", LW, 0, 20)
7  StringSet(str[0], "Local HMI", LW, 50, 20)
8
9  end macro_command
```

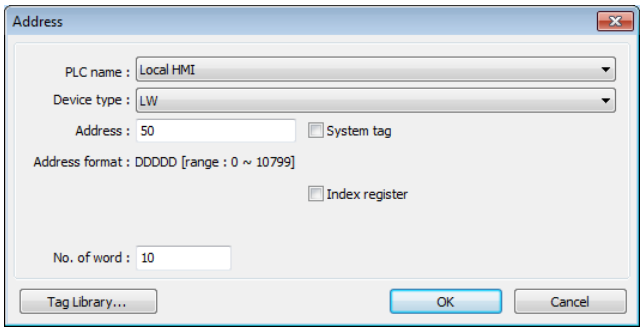
The first function “StringGet” is used to read a string from LW-0~LW-19, and store it into the str array. The second function “StringSet” is used to output the content of str array.

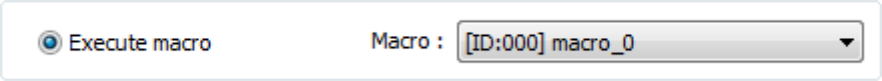
Add one ASCII Input object and one Function Key object in window 10 of the project. The settings of these objects are shown as below. Function Key object is used to execute macro_0.

ASCII Input object:



Function Key object:



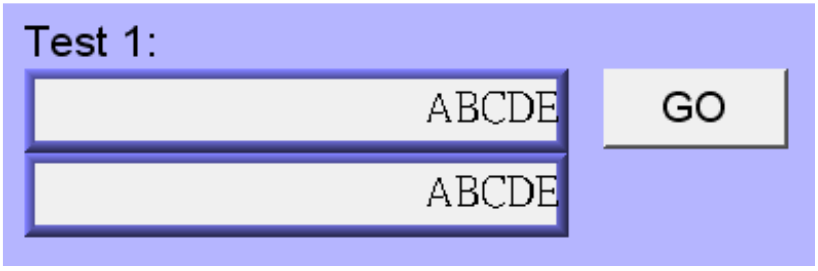


Lastly, use [Compile] to compile the project and execute [Off-line simulation] or [On-line simulation]. Follow the steps below to operate the executing project:

Step 1. Input string.

Step 2. Press “GO” button.

Step 3. Output string.

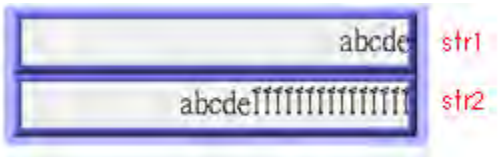


2. Initialization of a string.

Create a new macro and edit the content:

```
1
2   macro_command main()
3
4   char str1[20]="abcde"
5   char str2[20]={'a','b','c','d','e'}
6
7   StringSet(str1[0], "Local HMI", LW, 0, 20)
8   StringSet(str2[0], "Local HMI", LW, 50, 20)
9
10  end macro_command
```

The data enclosed in double quotation mark (" ") is viewed as a string. str1 is initialized as a string while str2 is initialized as a char array. The following snapshot of simulation shows the difference between str1 and str2 using two ASCII Input objects.



Macro compiler will add a terminating null character ('\0') at the end of a string. The function "StringSet" will send each character of str1 to registers until a null character is reached. The extra characters following the null character will be ignored even if the data count is set to a larger value than the length of string.

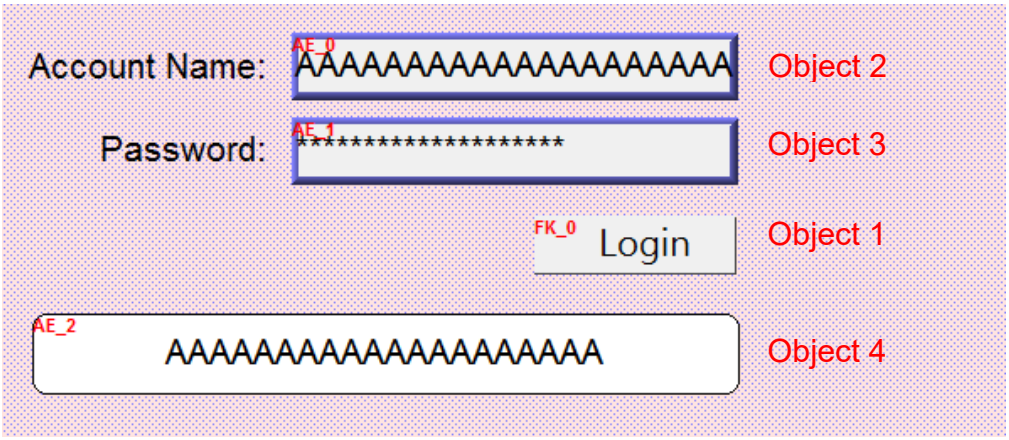
On the contrary, macro compiler will not add a terminating null character ('\0') at the end of a char array. The actual number of characters of str2 being sent to registers depends on the value of data count that is passed to the "StringSet" function.

3. A simple login page.

Create a new macro and edit the content, for example, Macro [ID:001] macro_1.

```
1
2   macro_command main()
3
4   char name[20]="admin"
5   char password[20]="123456"
6   char name_input[20], password_input[20]
7   char message_success[40]="Success! Access Accepted."
8   char message_fail[40]="Fail! Access Denied."
9   char message_clear[40]
10  bool name_match=false, password_match=false
11
12  StringGet(name_input[0], "Local HMI", LW, 0, 20)
13  StringGet(password_input[0], "Local HMI", LW, 50, 20)
14
15  name_match = StringCompare(name_input[0], name[0])
16  password_match = StringCompare(password_input[0], password[0])
17
18  FILL(message_clear[0], 0x20, 40) //FILL with white space
19  StringSet(message_clear[0], "Local HMI", LW, 100, 40)
20
21  if (name_match==true and password_match==true) then
22      StringSet(message_success[0], "Local HMI", LW, 100, 40)
23  else
24      StringSet(message_fail[0], "Local HMI", LW, 100, 40)
25  end if
26
27  end macro_command
```

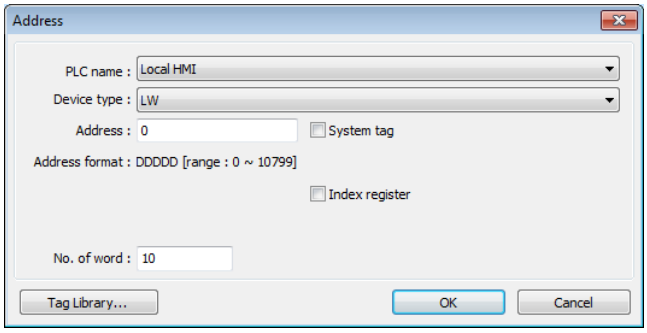
The first two “StringGet” functions will read the strings input by users and store them into arrays named name_input and password_input separately. Use the function “StringCompare” to check if the input account name and password are matched. If the account name is matched, name_match is set true; if the password is matched, password_match is set true. If both name_match and password_match are true, output the string “Success! Access Accepted.”. Otherwise, output the string “Fail! Access Denied.”. Add ASCII Input and Function Key objects in window 10 of the project. The settings of these objects are shown as below. Function Key object is used to execute macro_1.



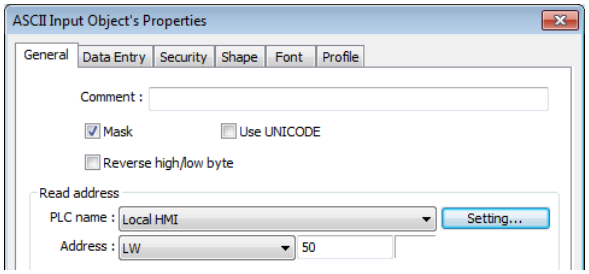
Object 1: Function Key

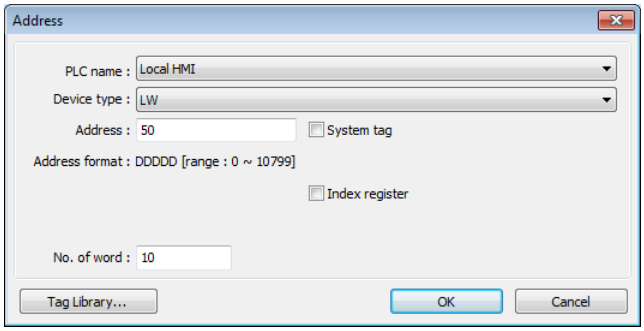
Select [Execute macro] and Macro: [ID:000] macro_1.

Object 2: ASCII Input

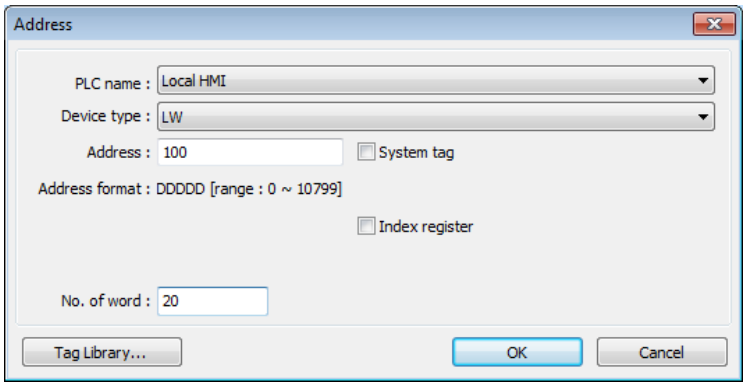


Object 3: ASCII Input



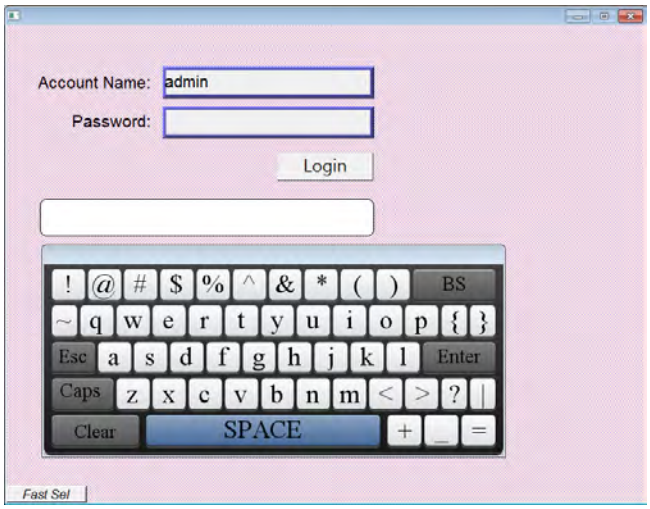


Object 4: ASCII Display

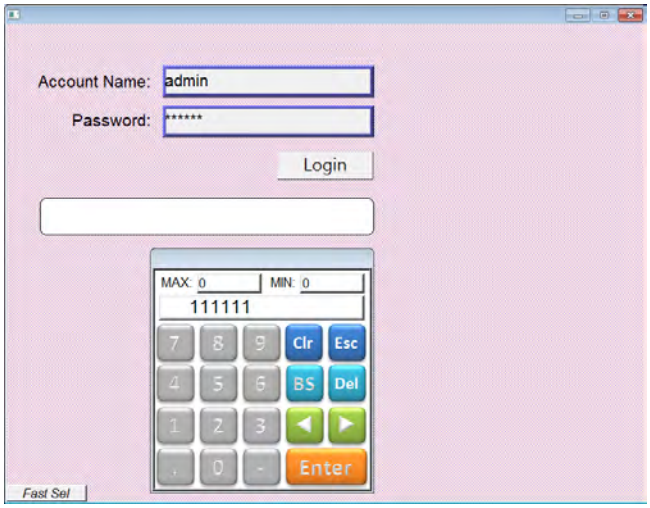


Lastly, use [Compile] to compile the project and execute [Off-line simulation] or [On-line simulation]. Follow the steps below to operate the executing project:

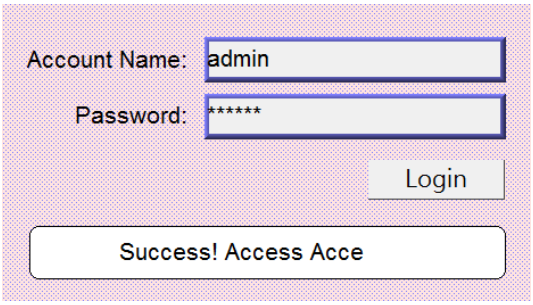
Step 1. Enter account name.

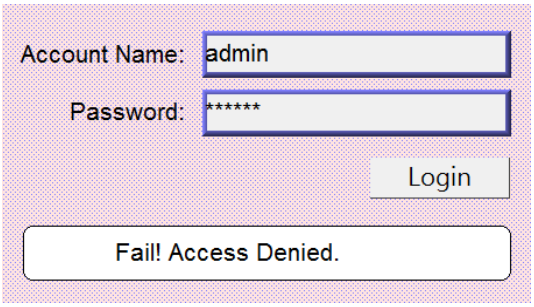


Step 2. Enter password and press [Login] button.



Step 3. Login succeeded or failed.

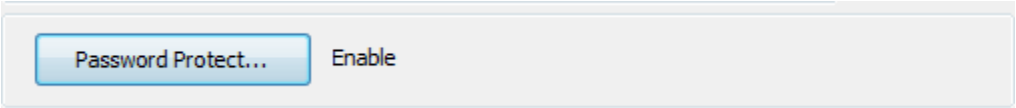




18.16. Macro Password Protection

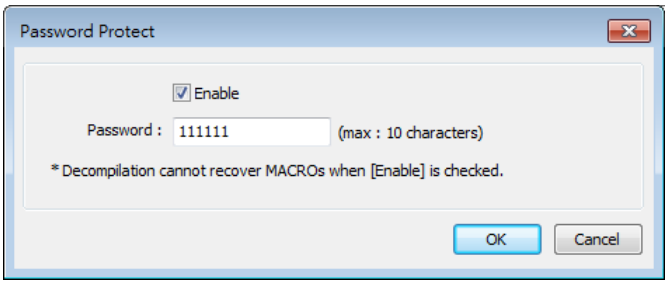
A password can be set to protect all the macros in the list, or an individual macro.

Protecting all macros:

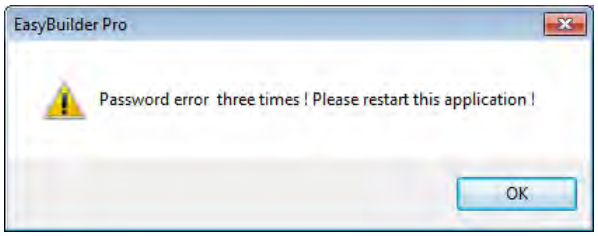


In Macro Manager window there's the [Password Protect…] button, click it and then click [Enable] to set a password less than or equals to 10 characters (support ASCII character only, e.g. "a$#*hFds").

After setting the password, users will have to enter correct password when opening Macro Manager.

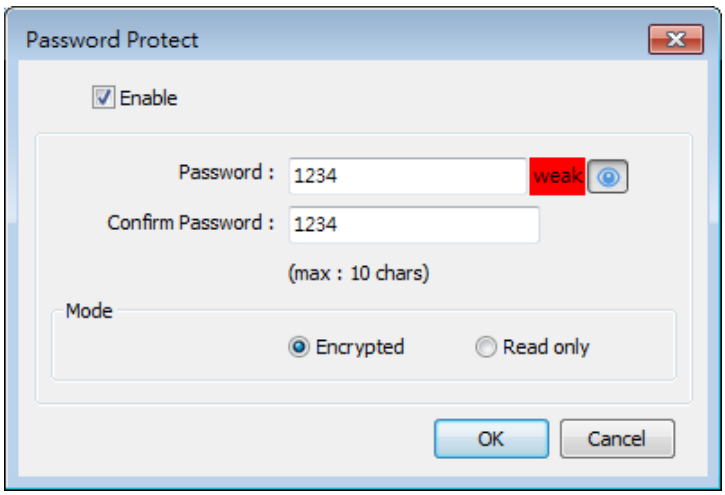


EasyBuilder Pro should be rebooted for typing the password again after 3 incorrect attempts.



Protecting individual macro:

In the Work Space for editing an individual macro, click the [Password Protect…] button and then click [Enable] to set a password less than or equals to 10 characters (support ASCII character only, e.g. "a$#*hFds"). [Encrypted] and [Read only] modes work as follows.



[Encrypted]

Encrypt the macro content. Entering macro editing window will require password.

EasyBuilder Pro should be rebooted for typing the password again after 3 incorrect attempts opening the same macro.

(The number of allowable incorrect attempts may vary between macros.)

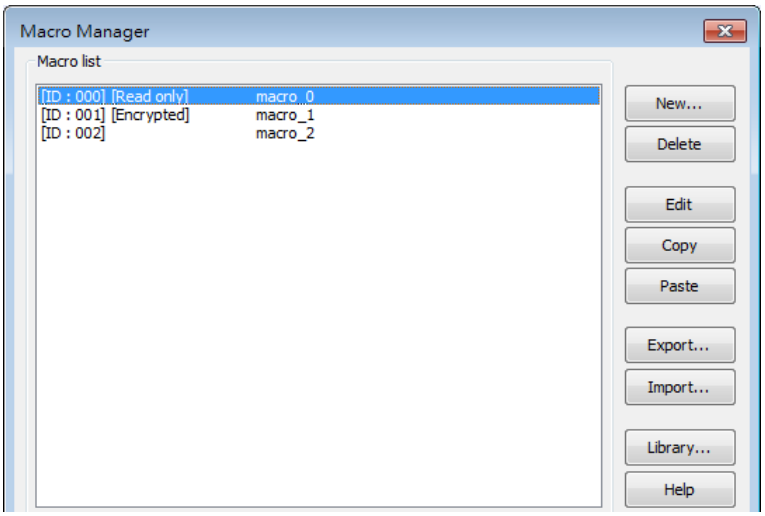
[Read-only]

The user can only view the content of the macro and will not be able to edit it.

With this mode selected, macro editing window can be opened directly from Macro Manager; however, a password is required after clicking [Password Protect…] button.

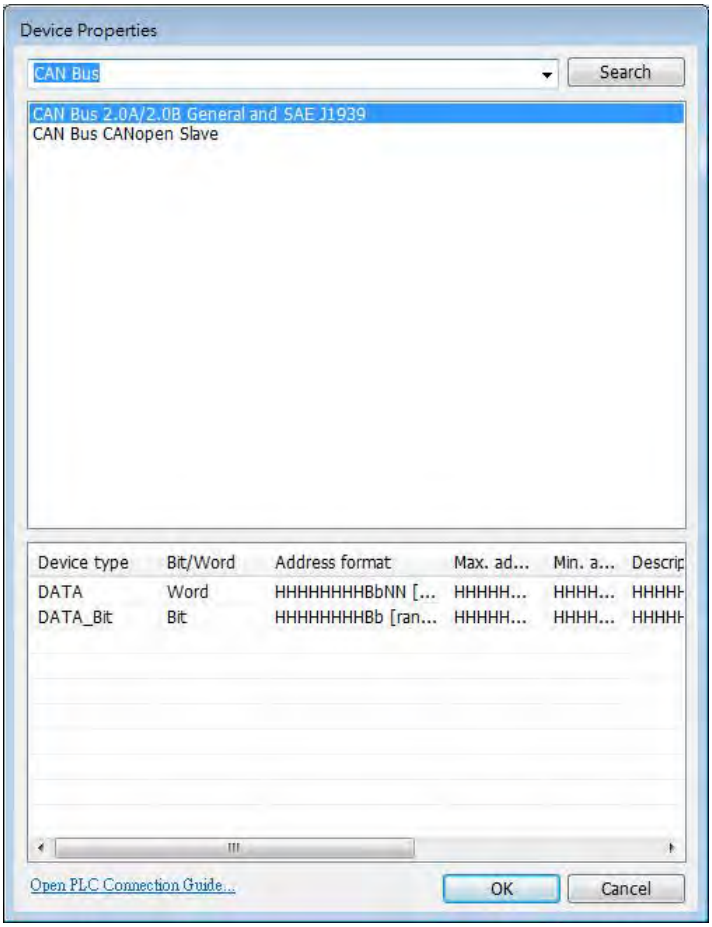
EasyBuilder Pro should be rebooted for typing the password again after 3 incorrect attempts.

In the macro list, the selected mode for each macro is shown.



18.17. Reading / Writing CAN bus Address Using Variable

In “CAN Bus 2.0A/2.0B General and SAE J1939” driver, two device types can be found: DATA and DATA_Bit, and the formats of these device types are shown in the following window.



Device Type & Address Format	Description
DATA HHHHHHHHBbNN	H: ID B: Byte position(1~8) b: Bit position (1~8) NN: Bit number(1~64)
DATA_Bit HHHHHHHHBb	H: ID B: Byte position(1~8) b: Bit position(1~8)

The ID is represented in hexadecimal while the position and number are represented in decimal, please see the usage below.

Examples:

Variable is **not** used:

short f

GetData(f, “CAN Device”, DATA, 4e55108, 1)

GetData(f, “CAN Device”, DATA, 4e65108, 1

Variable is used:

short f

unsigned int address = **0x**4e55108

GetData(f, “CAN Device”, DATA, address, 1)

address = address + 0x10000// == 0x4e65108

GetData(f, “CAN Device”, DATA, address, 1)

Please note that:

1. Declare variable as “Unsigned int” and use hexadecimal to represent address.
   Since the size of Unsigned int is 4 bytes and Bb, NN take 1 byte respectively, when using a variable for address parameter to read/write DATA_Bit device type, the format will change to HHHHHHBb (Max. ID: 0xffffff), and when using a variable for address parameter to read/write DATA device type, the format will change to HHHHBbNN (Max. ID: 0xffff).

19. Configure HMI as a MODBUS Server

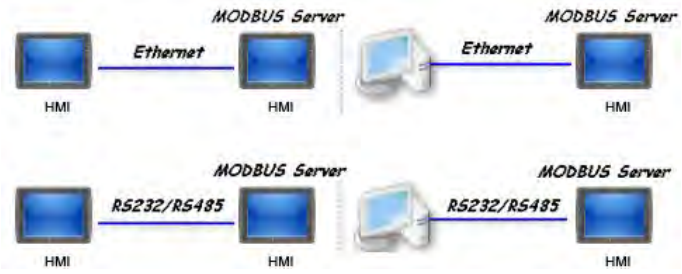
This chapter explains how to configure HMI as a MODBUS Server.

19.1. Overview 19-2
19.2. Steps to Create a MODBUS Server 19-2
19.3. Steps to Access a MODBUS Server 19-4
19.4. Changing MODBUS Server Station Number Online 19-7
19.5. MODBUS Address Type 19-7

19.1. Overview

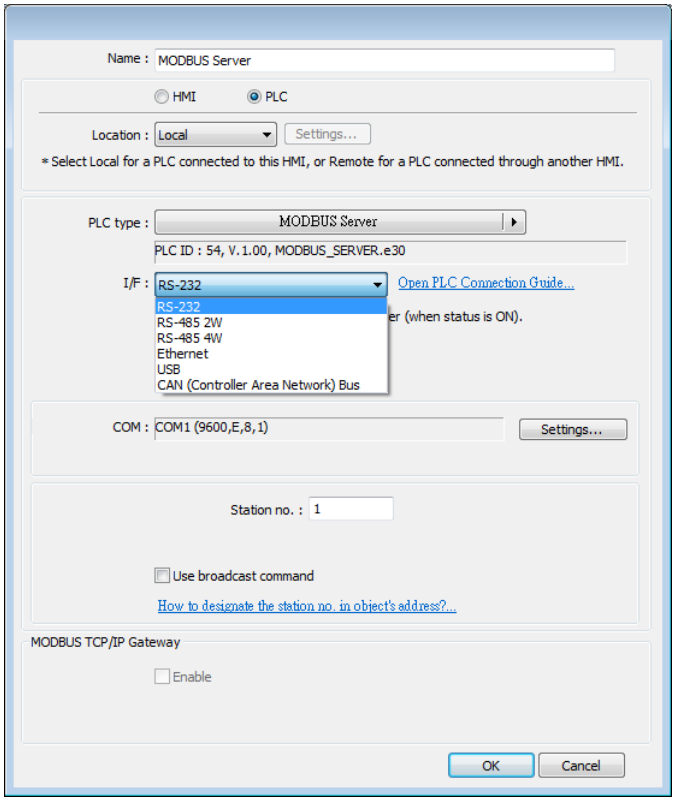
Once the HMI is configured as a MODBUS device, the data of HMI can be read or written via MODBUS protocol.

As shown in the following figure, the HMI is configured as a MODBUS device (also called MODBUS Server). The HMI, PC or other devices can use MODBUS protocol to read or write HMI data via Ethernet or RS-232 / RS-485 interface.

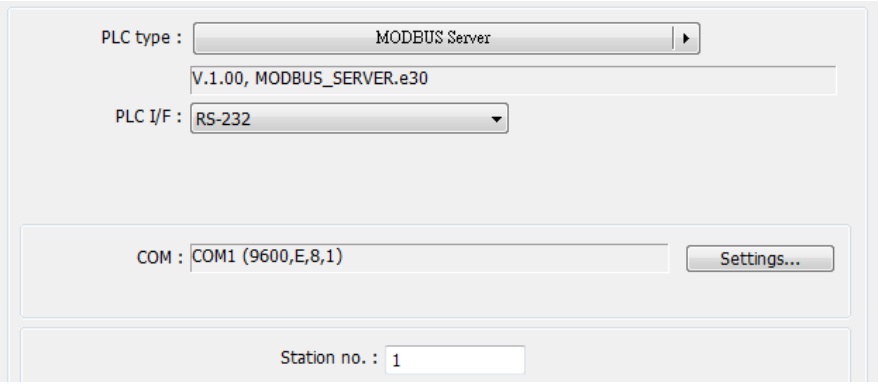


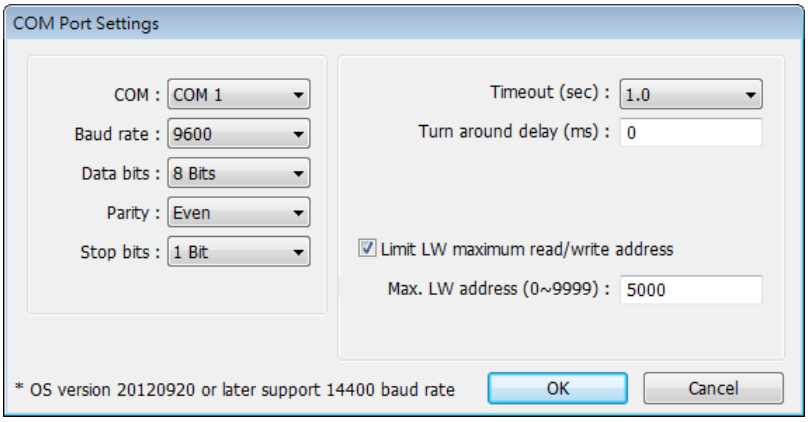
19.2. Steps to Create a MODBUS Server

1. To configure the HMI as a MODBUS device, add a new device to the device list in [System Parameters] first. Click [PLC type] drop-down box and select “MODBUS Server” driver. [PLC I/F] includes: RS-232 / RS-485 2W / RS-485 4W / Ethernet / USB / CAN BUS. Choose the PLC interface appropriate for your PLC model.

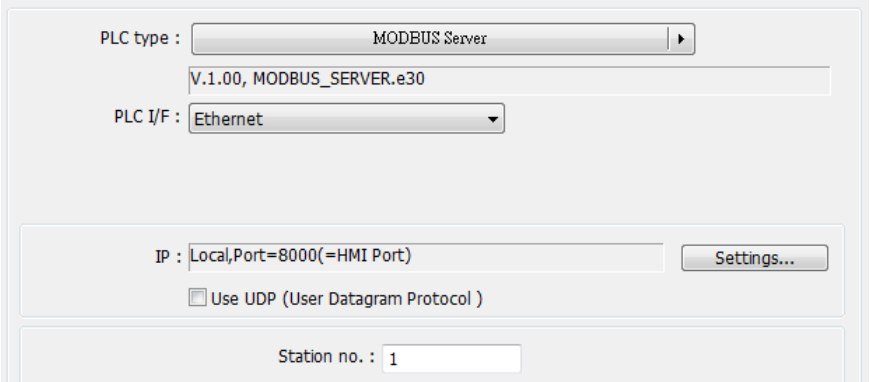


2. If [PLC I/F] is set to [RS-232] or [RS-485], please select [COM] (COM 1 ~ COM 3) and set correct communication parameters as shown in the following figure. MODBUS Server [Station no.] is set to 1. Click [Settings], the maximum LW address range read / written by Modbus Client can be set. When the object in the project uses a LW register, the Modbus Client will not be able to read or write an address that is not within the specified range.

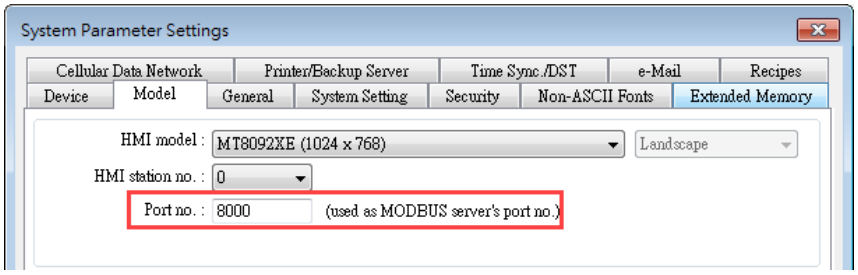




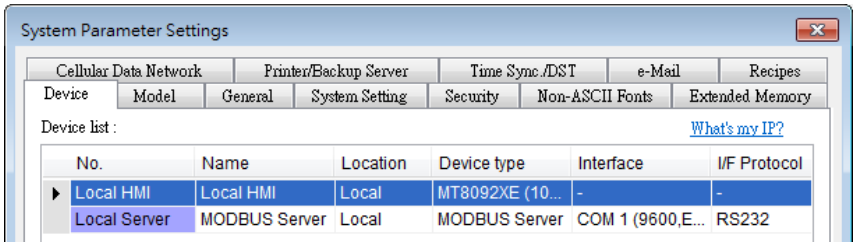
If [PLC I/F] is set to [Ethernet], please set [Port no.].



The [Port no.] of MODBUS Server and HMI must be the same. To change the port number, please set in the [System Parameters] » [Model] tab.

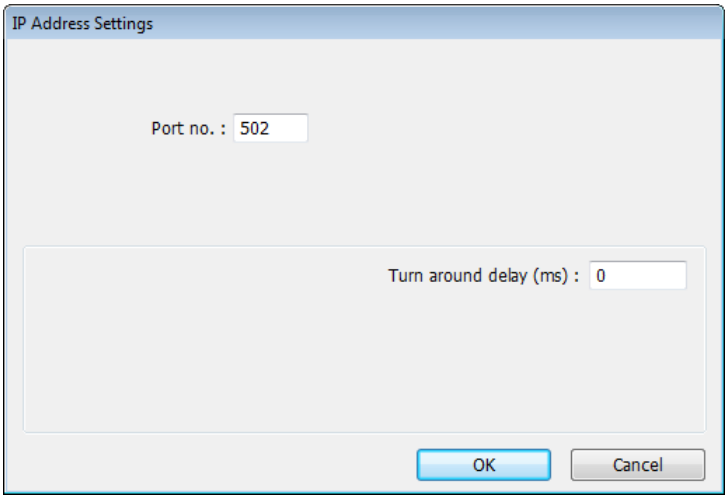


3. When finished, MODBUS Server is listed in [Device] tab. The configuration of MODBUS device is completed. Compile the .emtp file and download the compiled .exob file to the HMI, then HMI data can be read or written by using MODBUS protocol.



Note

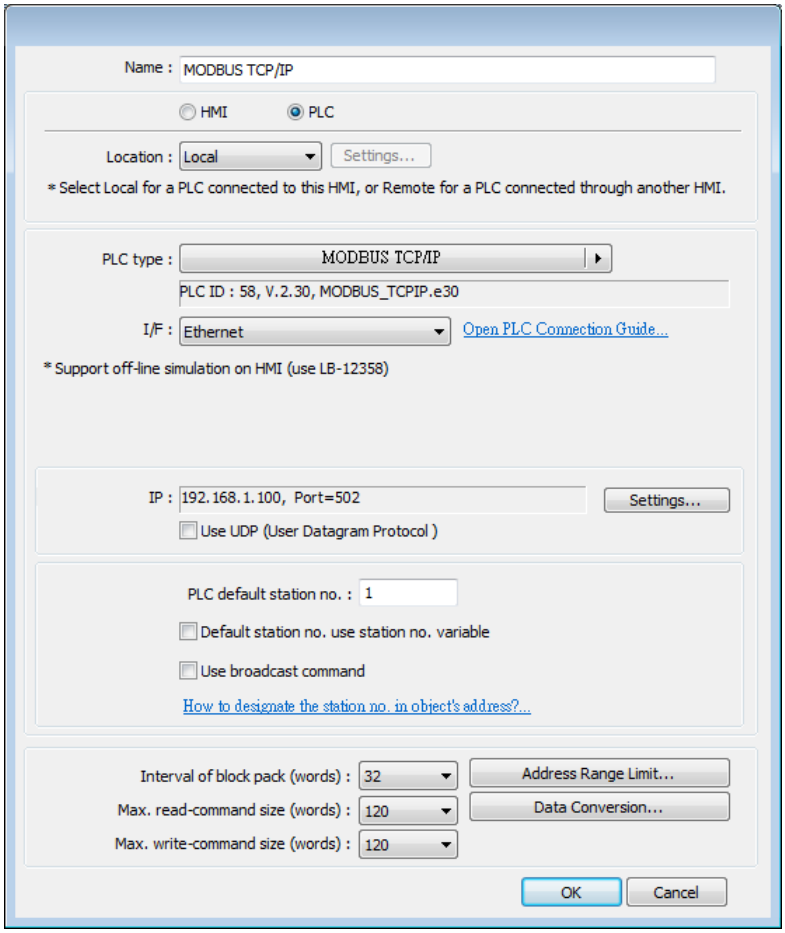
- For cMT / cMT X Series models, if [Ethernet] PLC interface is chosen, port number can be specifed.



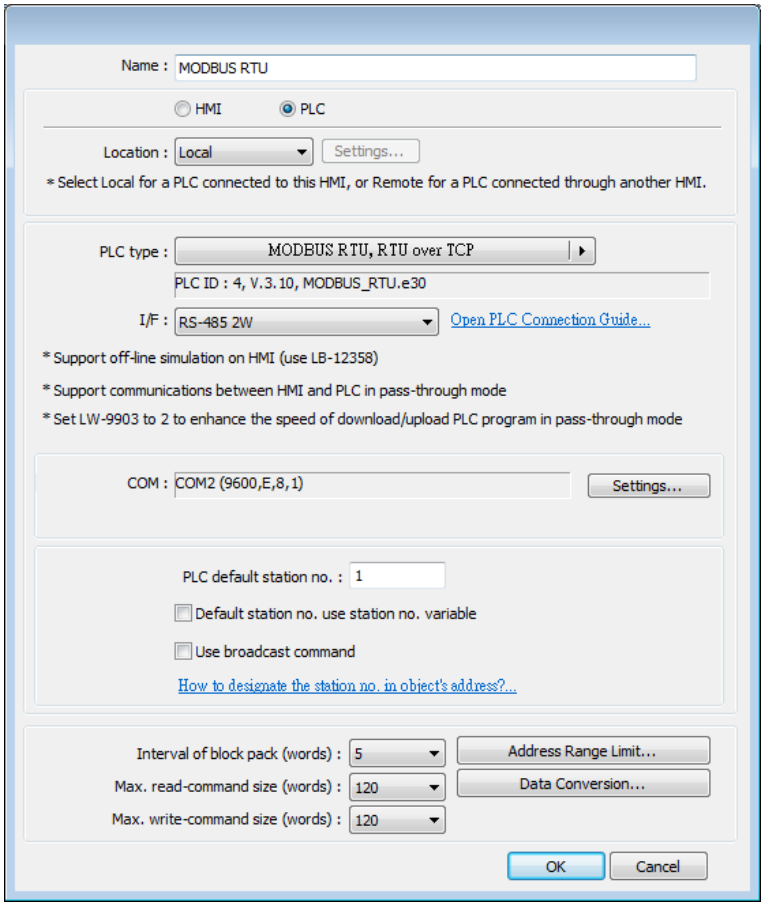
19.3. Steps to Access a MODBUS Server

Two HMIs can be configured as one MODBUS client and one MODBUS server to communicate and exchange data.

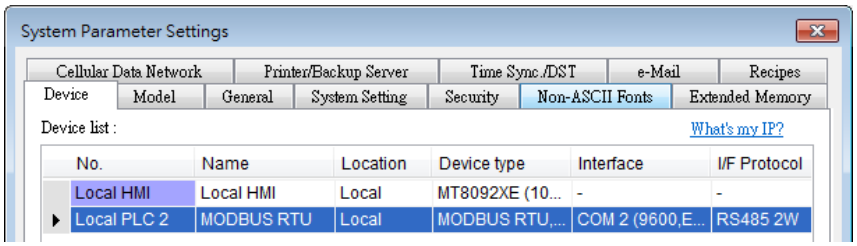
1. Add a new device in client's device list. If the client chooses [Ethernet] PLC interface, set [PLC type] to "MODBUS TCP/IP" and fill in the correct [IP address] (the IP of MODBUS Server), [Port no.], and [Station no.].



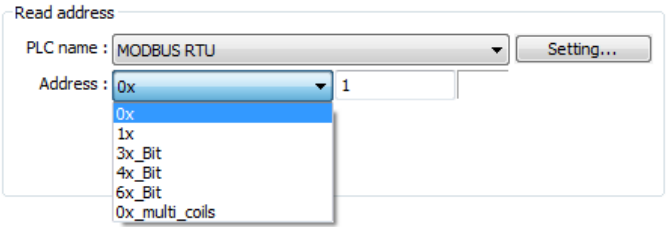
If the client chooses [RS-232] or [RS-485] PLC interface, the [PLC type] must be set to "MODBUS RTU", and its communication parameters also must be configured correctly.



2. When finished, click [OK], then a new device "MODBUS RTU" is listed in the [Device] tab.



3. In the setting page of each object, select "MODBUS RTU" in [PLC name], and set the address of MODBUS RTU.



Since the server is an HMI, the corresponding read and write addresses are listed below：

0x/1x (1 ~ 12800)	LB (0 ~ 12799)
3x/4x/5x (1 ~ 9999)	LW (0 ~ 9998)
3x/4x/5x (10000 ~ 65535)	RW (0 ~ 55535)

19.4. Changing MODBUS Server Station Number Online

EasyBuilder Pro provides the following system registers to change MODBUS Server station number online.

LW-9541	MODBUS/ASCII server station no. (COM 1)
LW-9542	MODBUS/ASCII server station no. (COM 2)
LW-9543	MODBUS/ASCII server station no. (COM 3)
LW-9544	MODBUS/ASCII server station no. (Ethernet)

19.5. MODBUS Address Type

In the EasyBuilder Pro, the address types of MODBUS protocol are 0x, 1x, 3x, 4x, 5x, 6x, 3x_bit and 4x_bit. MODBUS RTU function codes are listed below:

0x: Coils	A read and write device type. When reading a bit with this device type, the function code is 01H. When writing a bit, the function code is 05H. When writing multiple bits, the function code is 0fH.
1x: Discrete Inputs	A read only device type. When reading a bit the function code is 02H.
3x: Input Registers	A read only device type. When reading data, the function code is 04H.
4x: Holding Register	A read and write device type. When reading data, the function code is 03H. When writing data, the function code is 10H.
5x	The function code is the same as 4x. The difference is that 5x makes double word swap when the format is 32-bit unsigned. If the data read by 4x is 0x1234, the data read by 5x is 0x3412.
6x	A read and write device type. When reading data, the function code is 03H. The difference from 4x is that when writing data, the function code is 06H, meaning to write a single register.
3x_bit	The function code is the same as 3x. The difference is that 3x_bit reads a single bit in the data.
4x_bit	The function code is the same as 4x. The difference is that 4x_bit reads a single bit in the data.
6x_bit	The function code is the same as 6x. The difference is that 6x_bit reads a single bit in the data.

For more information, see “37 MODBUS TCP/IP Gateway”.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

20. How to Connect a Barcode Reader

This chapter explains how to connect a Barcode reader and the relevant settings.

20.1. Overview 20-2
20.2. Steps to Connect a Barcode Reader 20-2

20.1. Overview

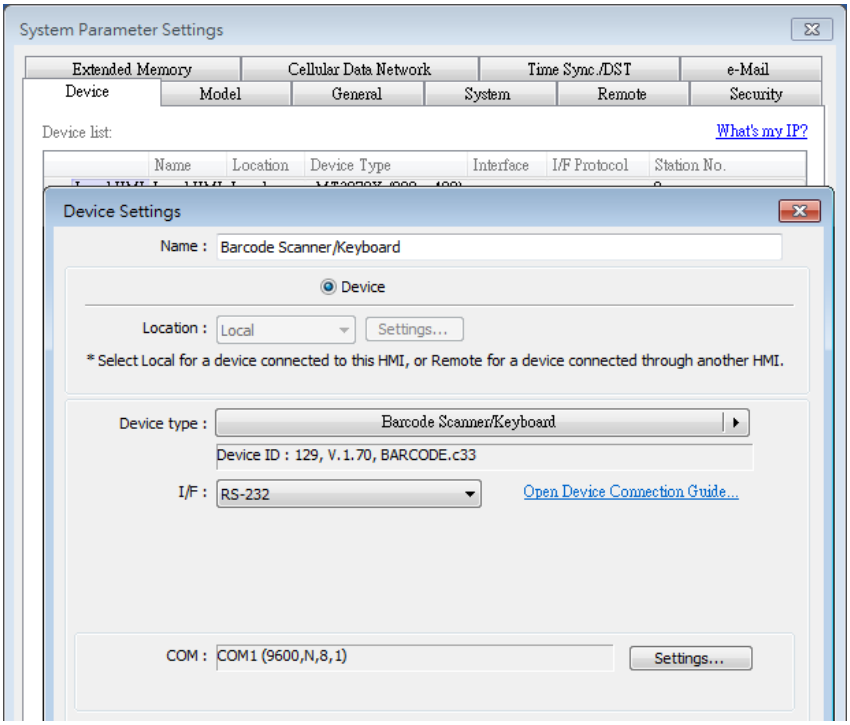
HMI can connect with barcode reader via the following interfaces:

- USB
- COM port

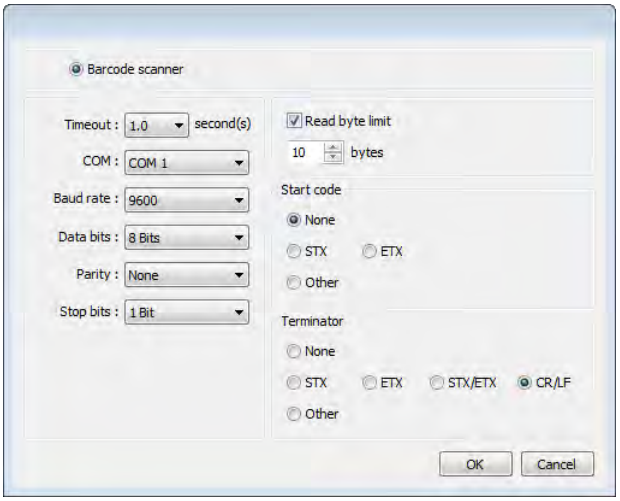
To connect a barcode reader, please add a new device by the following steps.

20.2. Steps to Connect a Barcode Reader

1. In EasyBuilder Pro click [System Parameters] » [Device list] and add a new device.



2. Click [Settings] and finish the settings.



Setting	Description
Timeout	When select [Barcode device], if the device reads slowly, a longer timeout is suggested to read data completely. When select [Keyboard], a time range can be set for keyboard entries. The system starts counting time from the first entry.
COM **Baud rate** **Data bits** **Parity** **Stop bits**	When using COM port, please set the communication parameters correctly. When using USB, there is no need to set the parameters.
Read byte limit	If this check box is selected, the number of bytes a barcode reader reads is restricted in order to prevent overloading. The range is 10 to 1024. Please note that the data cannot be read if it exceeds the limit.
Start code	The data is only valid when the first data is identical to the start code, otherwise the data will be ignored. The start code will not be stored in the address of barcode reader. **None** When no start code is used, HMI will save all the data to the designated address of barcode reader. **STX** Use 0x02 as start code. **ETX** Use 0x03 as start code. **Other** User user-defined start code. Example: If the start code is 255 (0xff), and the data read is: 0xff 0x34 0x39 0x31 0x32 0x30 0x30 0x34 0x37 The data saved in the designated barcode reader address will be: 0x34 0x39 0x31 0x32 0x30 0x30 0x34 0x37
Terminator	Terminator means the end of data. When a terminator is detected, it stands for the end of data stream. **None**

	When no terminator is used, HMI will save all the data to the designated address of barcode reader. **STX** Use 0x02 as terminator. **ETX** Use 0x03 as terminator. **STX/ETX** Use 0x02 or 0x03 as terminator. **CR/LF** Use 0x0a or 0x0d as terminator. **Other** Use user-defined terminator. Example: If the terminator is 55 (0x37), and the data read is: 0x34 0x39 0x31 0x32 0x30 0x30 0x34 0x37 The data saved in the designated barcode reader address will be: 0x34 0x39 0x31 0x32 0x30 0x30 0x34

When finish setting, a new device is added to the [Device list].
Now the barcode reader can be selected in [PLC type] when creating an object. The address types are listed in the following table.

Address Type	Address Name	Description
Bit	FLAG	**FLAG 0** indicates the status of data reading. When reading data, the status of FLAG 0 is set OFF and will return ON after reading successfully.
	RESET	**RESET 0** clears the data of BARCODE and RESULT when set ON.
	CONNECT_STATUS	**CONNECT_STATUS 0** indicates whether the barcode reader (USB interface) is connected. When the status is ON, the barcode reader is connected.
Word	BARCODE	**BARCODE 0:** Number of bytes currently read. **BARCODE 1 ~ n:** Stores the data read.
	RESULT	**RESULT 0** indicates the result of data reading. The following codes indicate: **0x00** Waiting to read BARCODE. **0x01** BARCODE successfully read. **0x02** Invalid BARCODE format. **0x03** The number of bytes specified in [Read byte limit] exceeded. **0x04** The Start Code of the data read does not match the setting. **0x05** The Terminator of the data read does not match the setting.

Example 1

The following is a setting example, the barcode is 9421007480830. BARCODE 0 is the address of Numeric Object (BYTES) and BARCODE 1 ~ n is the address of ASCII object (BARCODE).

In the example the data stored in the barcode reader address is listed in the following table:

Barcode Reader Address	Data
BARCODE 0	13 bytes (decimal) However, the data saved is 14 bytes = 7 words. It is because when the number of bytes is an odd number, the system adds a byte (0x00) to make it an even number.
BARCODE 1	3439 (HEX)
BARCODE 2	3132 (HEX)
BARCODE 3	3030 (HEX)
BARCODE 4	3437 (HEX)
BARCODE 5	3038 (HEX)
BARCODE 6	3338 (HEX)
BARCODE 7	0030 (HEX)

Note

- HMI can only connect with one USB barcode reader. When the device list in the project includes this kind of device, the system register LB-9064: [enable USB barcode device (disable keyboard) (when ON)] is set ON. To enable USB keyboard again and stop using USB barcode reader, please set LB-9064 OFF.

Click the icon to download the demo project. Please confirm your internet connection.

21. Ethernet Communication and Multi-HMI Connection

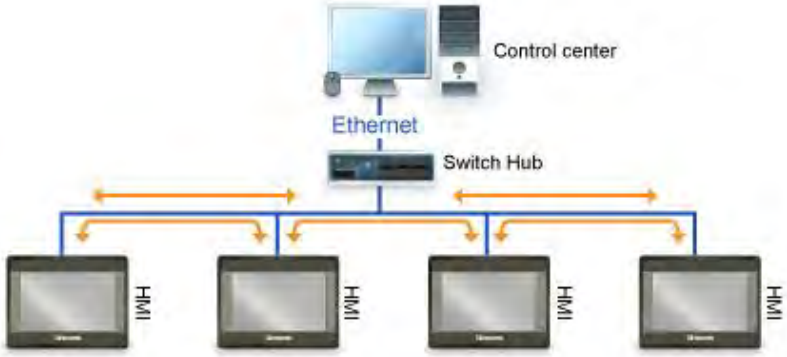
This chapter explains how to connect multiple devices via Ethernet.

21.1. Overview 21-2
21.2. HMI to HMI Communication 21-2
21.3. PC to HMI Communication 21-3
21.4. Operate the PLC Connected with Other HMI 21-4

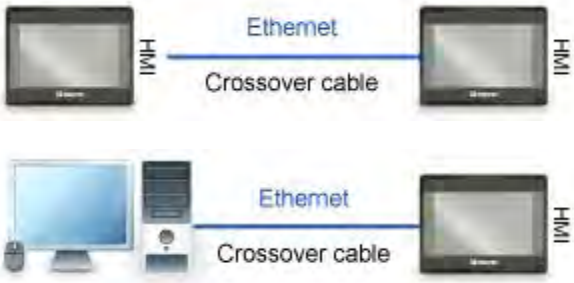
21.1. Overview

There are two ways of Ethernet communication:

- Use RJ45 straight through cable and hub.



- Use RJ45 crossover cable and without hub, but this is limited to point-to-point connection (HMI to HMI or PC to HMI).

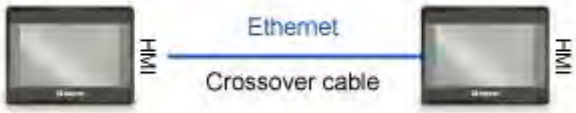


Through Ethernet network, the system provides the following methods for data transmission:

- HMI to HMI communication.
- PC to HMI communication.
- Operating the PLC connected to another HMI.

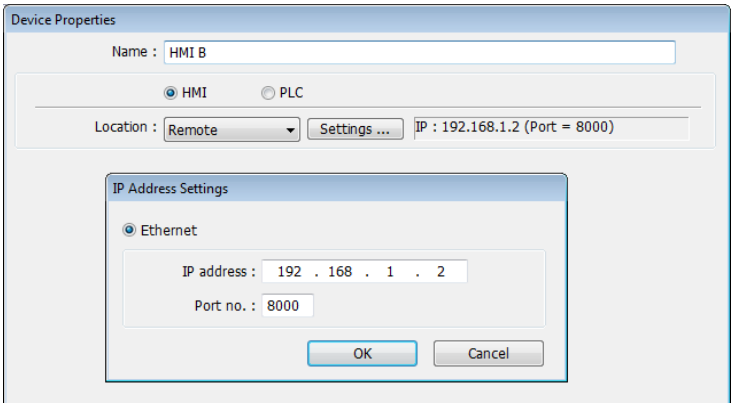
21.2. HMI to HMI Communication

To exchange data between one HMI and another HMI, add a new remote HMI device in [System Parameter Settings]. If there are 2 HMIs (HMI A and HMI B), in order to use a Set Bit object on HMI A to control [LB-0] on HMI B, the setting of the project of HMI A is explained in the following part.

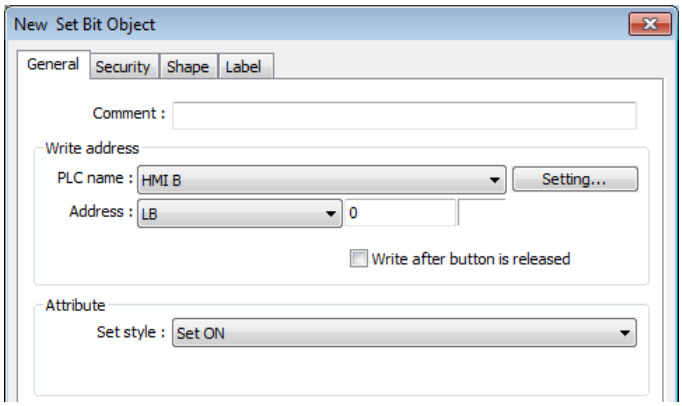


1. Set the IP address of the two HMIs, for example, HMI A: 192.168.1.1, HMI B: 192.168.1.2.

2. In [System Parameters] » [Device list], add a remote HMI B (IP: 192.168.1.2).



3. Create a Set Bit Object, select "HMI B" in [PLC name] to control the address of the remote HMI.

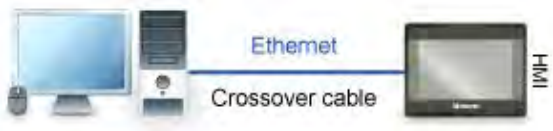


Note

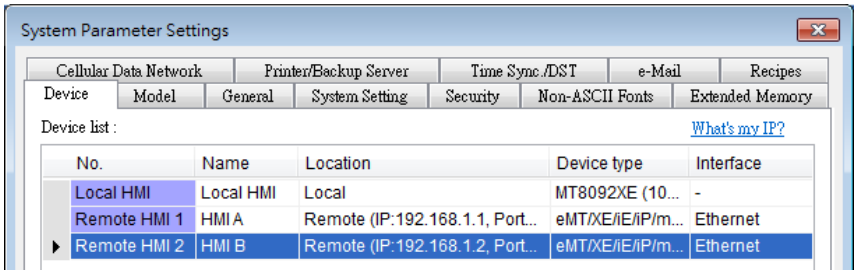
- One HMI can handle requests from a maximum of 64 HMIs simultaneously.
- One cMT / cMT X Series model can handle requests from a maximum of 32 HMIs simultaneously.

21.3. PC to HMI Communication

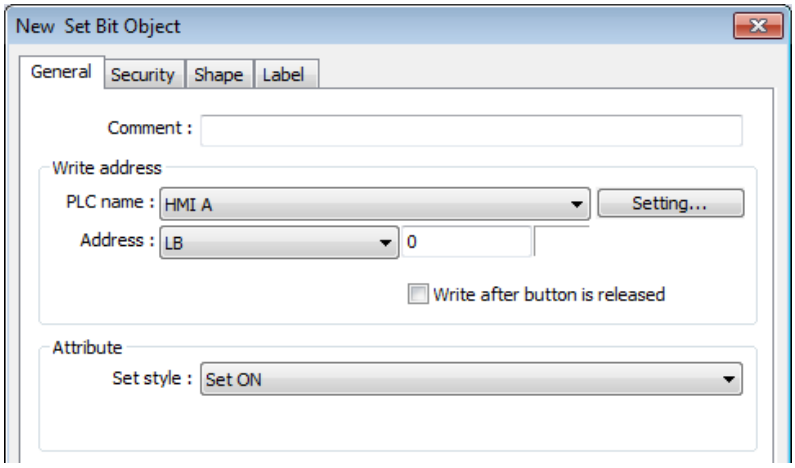
With On-line Simulation, PC can collect data from HMI through Ethernet network and save the data files to PC. To connect PC with two HMIs (HMI A and HMI B), the setting of the project on PC is explained in the following part.



1. Set the IP address of the two HMIs, for example, HMI A: 192.168.1.1, HMI B: 192.168.1.2.
2. In [System Parameter Settings] » [Device list], add a remote HMI A (IP: 192.168.1.1) & HMI B (IP: 192.168.1.2).



3. Create a Set Bit Object, select "HMI A" in [PLC name] to control the address of the remote HMI A. Same for the HMI B.

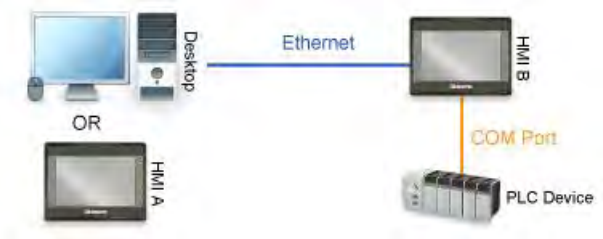


Note

- A PC can control at most 64 HMIs simultaneously.
- As shown above, HMI can also control PC. PC can be seen as another HMI, that is, adding a remote HMI in the project of HMI A / HMI B, and the IP of the remote HMI is set to the IP of PC.

21.4. Operating the PLC Connected with Other HMI

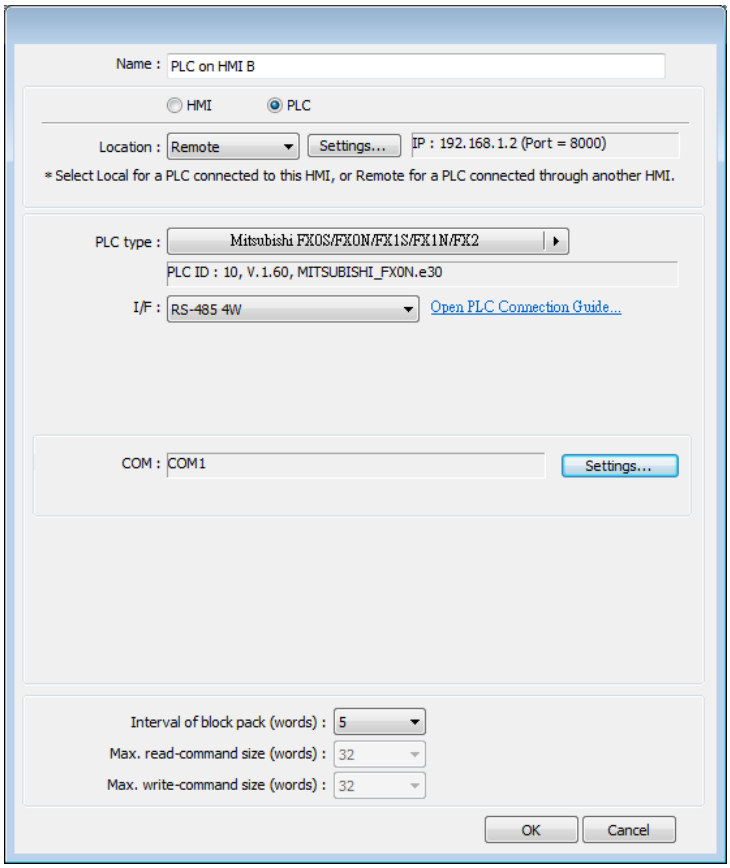
Through Ethernet network, PC or HMI can operate the PLC that is connected to another HMI. If PLC is connected to COM 1of HMI B, when using PC or HMI A to read PLC data, the setting of the project of PC or HMI A is explained in the following part.



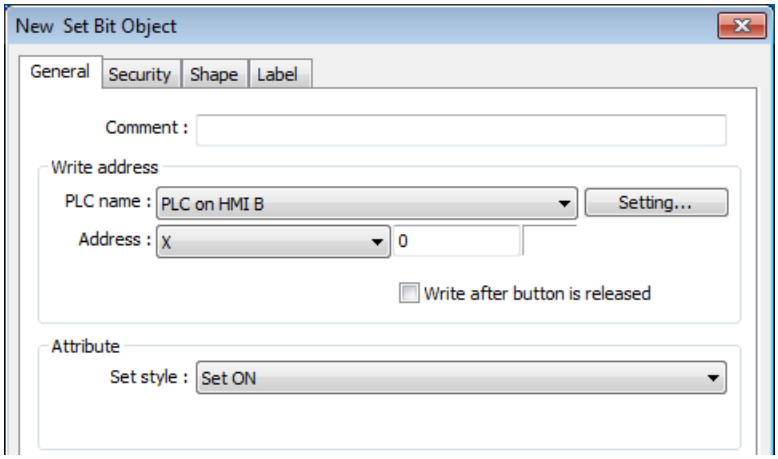
21.4.1. Settings of eMT / iE / XE / mTV / iP Series

1. Set the IP address of HMI B, for example, 192.168.1.2.

2. In [System Parameter Settings] » [Device list], add a remote PLC, and set [Name] to "PLC on HMI B". Set correct parameters. Since this PLC is connected to remote HMI B, set the IP address to HMI B (IP: 192.168.1.2).

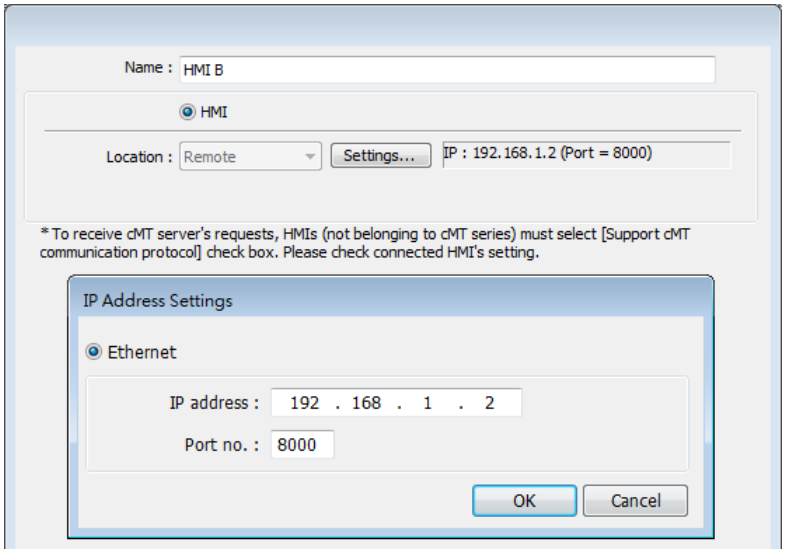


3. Create a Set Bit Object, select "PLC on HMI B" in [PLC name] to control the PLC connected with the remote HMI B.

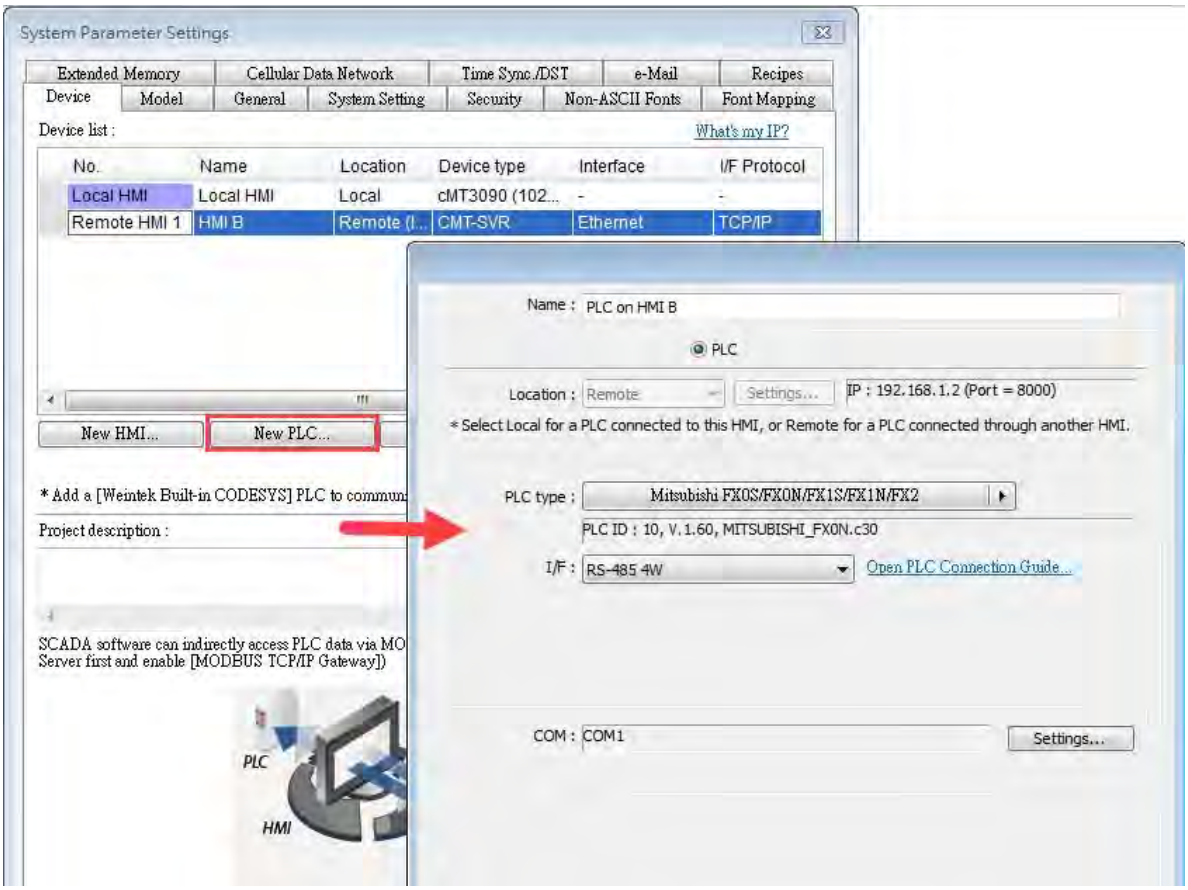


21.4.2. Settings of cMT / cMT X Series models

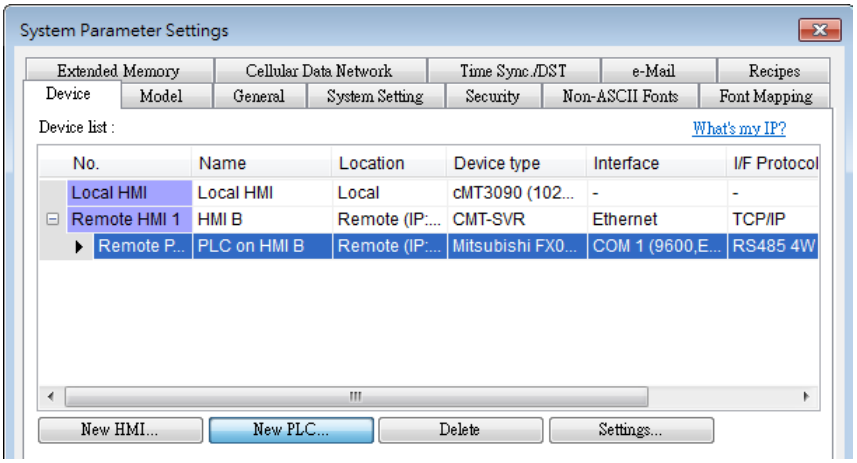
1. Set the IP address of HMI B, for example, 192.168.1.2.
2. In [System Parameters] » [Device list], click [New HMI]. Set the IP address to HMI B (IP: 192.168.1.2).



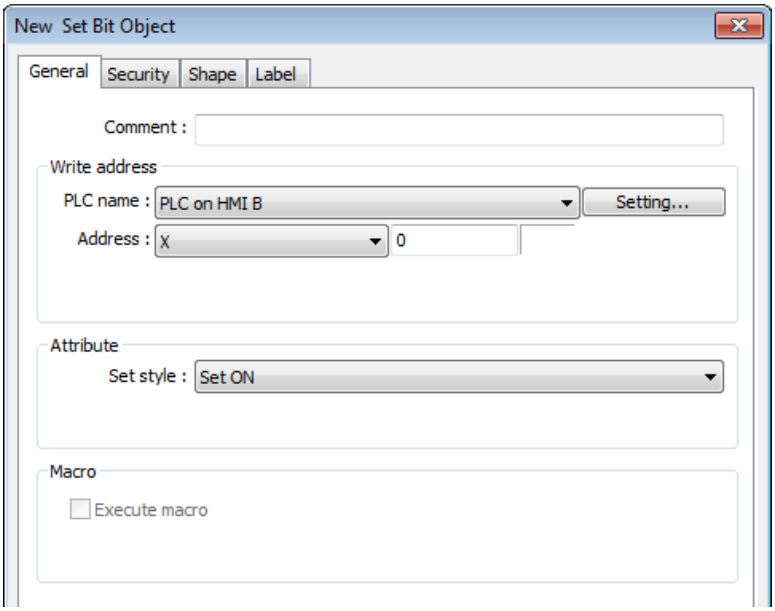
3. In the project of HMI B, go to [System Parameter Settings] » [Device list], click [New PLC], set [Name] to "PLC on HMI B". Set correct parameters.



4. When finished, a remote PLC can be found under Remote HMI 1. Local HMI 1 stands for HMI A, Remote HMI 1 stands for HMI B, and Remote PLC 1 is connected with HMI B.



5. Create a Set Bit Object, select "PLC on HMI B" in [PLC name] to control the PLC connected with the remote HMI B.



Note

- When the remote HMI in a cMT/cMT X Series project is an eMT/iE/XE/mTV model, please select [Support iE/XE/eMT/mTV HMI communication protocol and EasyWatch] check box in the [Model] tab in [System Parameters]. Similarly, when the remote HMI in an eMT/iE/XE/mTV project is a cMT / cMT X Series model, please select [Support cMT communication protocol] to establish communication between cMT/cMT X and eMT/iE/XE/mTV models.

22. System Registers

This chapter introduces different types of registers.

22.1. Overview 22-3
22.2. The Address Ranges of Local HMI 22-4
22.2.1. Bits 22-4
22.2.2. Words 22-4
22.3. System Registers 22-4
22.3.1. HMI Time 22-5
22.3.2. HMI Operation 22-7
22.3.3. Touch Position 22-8
22.3.4. Local HMI Network Information 22-9
22.3.5. Project File Information 22-11
22.3.6. Storage Space Management 22-12
22.3.7. Recipe and Extended Memory 22-13
22.3.8. Data Sampling 22-16
22.3.9. Event Log 22-18
22.3.10. Station Number Variables 22-20
22.3.11. Index Registers 22-22
22.3.12. MODBUS Server Communication 22-24
22.3.13. Communication Parameter Settings 22-25
22.3.14. Communication Status and Control with Device (COM) 22-29
22.3.15. Communication Status and Control with Device (Ethernet) 22-31
22.3.16. Communication Status and Control with Device (USB) 22-41
22.3.17. Communication Status and Control with Device (CAN Bus) 22-42
22.3.18. Communication Status and Control with Remote HMI 22-43
22.3.19. Communication Status and Control with Remote Device 22-58
22.3.20. Local/Remote Operation Restrictions 22-60
22.3.21. Communication Error Codes 22-61
22.3.22. Driver ID 22-63
22.3.23. DLT645 Controller 22-63
22.3.24. [Device No Response] Window Control 22-63
22.3.25. [Fast Selection] Window Control 22-64
22.3.26. EasyAccess 22-64
22.3.27. EasyAccess 2.0 22-64

22.3.28. Remote Print/Backup Server 22-65
22.3.29. Pass-Through Settings 22-66
22.3.30. VNC Control 22-68
22.3.31. Project Key and HMI Key 22-68
22.3.32. USB Security Key 22-69
22.3.33. User Name and Password 22-69
22.3.34. Macro 22-70
22.3.35. Input Object Function 22-73
22.3.36. Time Sync./Daylight Saving Time 22-73
22.3.37. Cellular Data Network 22-75
22.3.38. WiFi Setting 22-76
22.3.39. OPC UA Server 22-78
22.3.40. e-Mail 22-79
22.3.41. cMT Viewer and Diagnoser 22-80
22.3.42. CODESYS Information 22-83
22.3.43. Miscellaneous 22-84

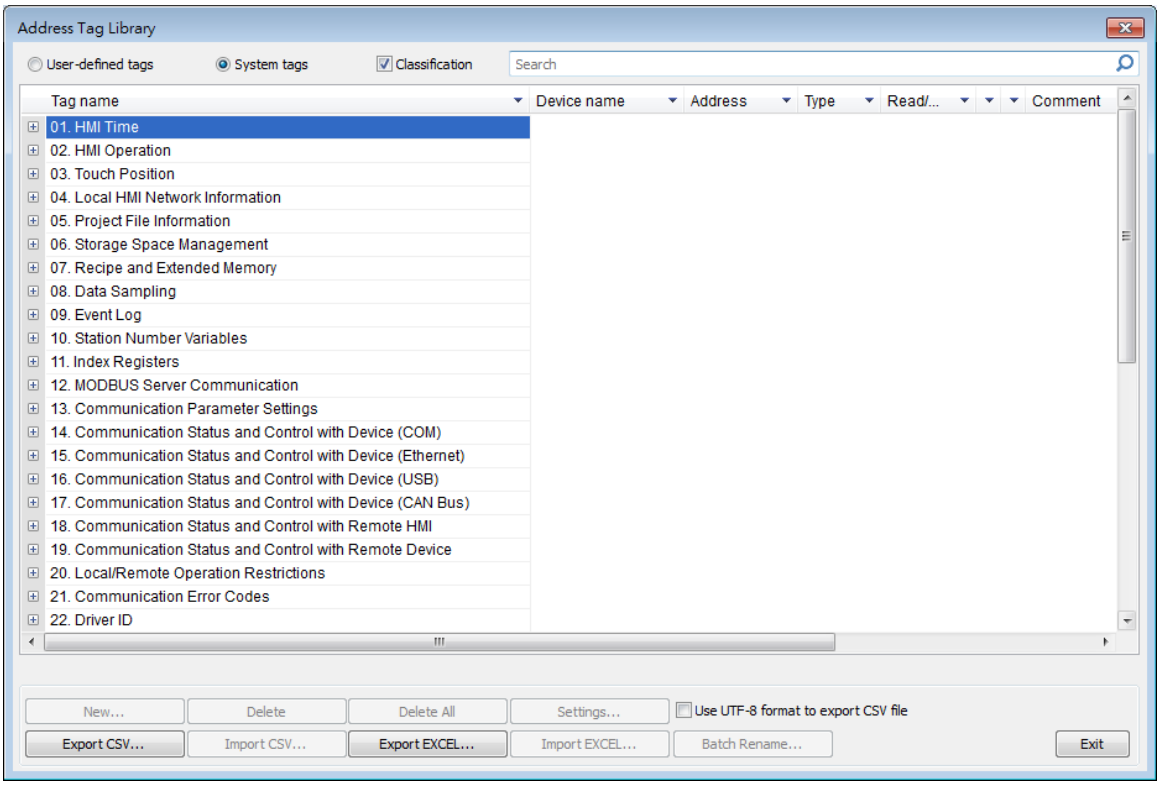
22.1. Overview

Some Word and Bit addresses are reserved in EasyBuilder Pro. These registers are reserved for different functions. This chapter introduces different types of registers.

The “C” letter in the register tables stands for “Control”, which means that this register not only allows write operation, but also can be controlled by Macro or a remote HMI.

When using a cMT/cMT X Series model, the PLW and PLB registers can be selected. LW/LB are local registers whereas PLW/PLB are client registers. The client device can be: cMT-iV5, cMT-iV6, iOS, Android device…etc. When connecting a cMT/cMT X Series model with one or multiple client devices, the PLW/PLB registers can be set on the client devices.

Please note that the supported system tags may vary between models and can be checked in the Address Tag Library, where the unsupported ones will be highlighted in gray.



22.2. The Address Ranges of Local HMI

22.2.1. Bits

Register	Device Type	Range	Format
Local Bits (for users)	LB	0 ~ 8999	DDDDD
Local Bits (for system registers)	LB	9000 ~ 12895	DDDDD
Client Bits	PLB	0 ~ 12399	DDDDD
Local Word Bits	LW_Bit	0 ~ 1290015	DDDDDdd DDDDD: address dd: bit no. (00 ~ 15)
Client Word Bits	PLW_Bit	0 ~ 1230015	DDDDDdd DDDDD: address dd: bit no. (00 ~ 15)
Retentive Bit Index	RBI	0 ~ 65535f	DDDDDh DDDDD: address h: bit no. (0 ~ f) Use LW-9000 as Index Register, and correspond to RW_Bit
Retentive Word Bits	RW_Bit	0 ~ 524287f	DDDDDh DDDDD: address h: bit no. (0 ~ f)
Retentive A Word Bits	RW_A_Bit	0 ~ 65535f	DDDDDh DDDDD: address h: bit no. (0 ~ f)
Client Retentive Word Bits	PRW_Bit	0 ~ 52428715	DDDDDdd DDDDD: address dd: bit no. (00 ~ 15)

22.2.2. Words

Register	Device Type	Range	Format
Local Words (for users)	LW	0 ~ 8999	DDDDD
Local Words (for system registers)	LW	9000 ~ 12900	DDDDD
Client Words	PLW	0 ~ 12300	DDDDD

Retentive Words	RW	0 ~ 524287	DDDDDD
Retentive A Words	RW_A	0 ~ 65535	DDDDD
Retentive Word Index	RWI	0 ~ 65535	DDDDD Use LW-9000 as Index Register, and correspond to RW
Client Retentive Words	PRW	0 ~ 524287	DDDDDD
Extended Memory Words	EM0 ~ EM15	0 ~ 1073741823	DDDDDDDDDD

22.3. System Registers

22.3.1. HMI Time

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-11958	time setting error (when ON) *Note 3	R	R	R	Y
LW-9010	(16bit-BCD) : local second	R/W	R/C	R/C	Y
LW-9011	(16bit-BCD) : local minute	R/W	R/C	R/C	Y
LW-9012	(16bit-BCD) : local hour	R/W	R/C	R/C	Y
LW-9013	(16bit-BCD) : local day	R/W	R/C	R/C	Y
LW-9014	(16bit-BCD) : local month	R/W	R/C	R/C	Y
LW-9015	(16bit-BCD) : local year	R/W	R/C	R/C	Y
LW-9016	(16bit-BCD) : local week	R	R	R	Y
LW-9017	(16bit) : local second	R/W	R/C	R/C	Y
LW-9018	(16bit) : local minute	R/W	R/C	R/C	Y
LW-9019	(16bit) : local hour	R/W	R/C	R/C	Y
LW-9020	(16bit) : local day	R/W	R/C	R/C	Y
LW-9021	(16bit) : local month	R/W	R/C	R/C	Y
LW-9022	(16bit) : local year *Note 1	R/W	R/C	R/C	Y
LW-9023	(16bit) : local week *Note 2	R	R	R	Y
LW-9030	(32bit) : system time (unit : 0.1 second)	R	R	R	Y

LW-9048	(16bit) : time (0 : AM, 1 : PM)	R/W	R/C	R/C	Y
LW-9049	(16bit) : local hour (12-hour format)	R/W	R/C	R/C	Y

Note

1. Value range: 2000 ~ 2035.
2. Value range: 0 ~ 6, stand for Sunday ~ Saturday.
3. When using LW-9010 to LW-9023 to update RTC time and an invalid value is set, the time before the update will be restored and the system register [LB-11958: time setting error] will be set ON. Updating time on PC during simulation by using LW-9010 to LW-9023 is ineffective.

22.3.2. HMI Operation

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9018	disable(set ON)/enable (set OFF) mouse cursor	R/W	R/C	R/C	Y
LB-9019	disable(set ON)/enable (set OFF) sound output	R/W	R/C	R/C	Y
LB-9020	show (set ON)/ hide (set OFF) system setting bar	R/W	R/C	R/C	Y
LB-9033	disable(when on)/enable (when off) HMI upload function *Note 1	R/W	R/C	R	Y
LB-9040	backlight up (set ON) *Note 2	W	C	C	Y
LB-9041	backlight down (set ON) *Note 2	W	C	C	Y
LB-9047	reboot HMI (set ON when LB-9048 is on)	W	C	C	Y
LB-9048	reboot-HMI protection	R/W	R/C	R/C	Y
LB-9062	open hardware setting dialog (set ON)	W	C	C	Y
LB-9063	disable(set ON)/enable(set OFF) popping information dialog while finding an USB disk	R/W	R/C	R/C	Y
LB-9064	enable USB barcode device (disable keyboard) (when ON) *Note 5	R/W	R/C	R	Y
LB-11959	LED indicator control *Note 4	R/W	R/C	R/C	N
LB-12042	open/close [System information] dialog (set ON/set OFF)	R/W	R/C	R/C	N
LB-12051	buzzer control (active when ON)	R/W	R/C	R/C	Y
LB-12360	CPU loading alarm (> 95%) *Note 6	R	R	R	Y
LB-12364	show (set ON)/hide (set OFF) [Reset HMI to default] button in calibration mode	R/W	R/C	R/C	N
LB-12660	status of reset button (when the button is pressed, status is ON)	R	R	R	Y
LB-12665	open USB/SD card download/upload dialog	R/W	R/C	R/C	Y
LB-12752	disable(set OFF)/enable(set ON) vibration. (Supported OS version : 20200618 or later) *Note 7	R/W	R/C	R/C	Y
LW-9007	(16bit) : hardware index	R	R	R	N
LW-9008	(32bit-float) : battery voltage *Note 3	R	R	R	N
LW-9025	(16bit) : CPU loading (x 100%)	R	R	R	Y
LW-9026	(16bit) : OS version (year)	R	R	R	Y
LW-9027	(16bit) : OS version (month)	R	R	R	Y

LW-9028	(16bit) : OS version (day)	R	R	R	Y
LW-9040	(16bit) : backlight index *Note 2	R/W	R/C	R/C	Y
LW-9051	(16bit) : audio volume (0 ~100)	R/W	R/C	R/C	Y
LW-9054	(32bit) : HMI model ID	R	R	R	Y
LW-9080	(16bit) : backlight saver time (unit : minute)	R/W	R/C	R/C	Y
LW-9081	(16bit) : screen saver time (unit : minute)	R/W	R/C	R/C	Y
LW-9141	(16bit) : HMI station no.	R/W	R/C	R/C	Y
LW-9199	(16bit) : external keyboard layout : 0: (QWERTY), 1: (AZERTY), 2: (Cyrillic)	R/W	R/C	R/C	Y
LW-9350	(16bit) : pending command no. in local HMI	R	R	R	Y
LW-10884	(16 words) : HMI name	R/W	R/C	R/C	Y
LW-11155	(32bit) : the total size of HMI memory (K bytes)	R	R	R	Y
LW-11157	(32bit) : the free size of HMI memory (K bytes)	R	R	R	Y
LW-11159	(16bit) : memory loading (x 100%)	R	R	R	Y
LW-11382	(16bit) : DIP switch status (bit 0 : DIP 1, bit 1 : DIP 2, bit 2 : DIP 3)	R	R	R	Y

Note

1. After changing the settings, please reboot HMI for the updates to take effect.
2. Use LW-9040 together with LB-9040 ~ LB-9041 to adjust the backlight brightness, range: 0 ~ 31.
3. Only supported by eMT Series. When the battery voltage level, indicated by LW-9008, drops below 2.8V, battery replacement is recommended.
4. When multiple mTV or cMT-SVR/cMT-HDM/cMT-FHD devices are used, this register can be triggered to make the LED indicator blink for identifying the device.
5. LB-9064: Enable USB barcode device (disable keyboard).

Click the icon to download the demo project. Please confirm your internet connection.

6. This register will be set to ON if CPU loading remains over 95% for 30 seconds.
7. Only supported on cMT3162X.

22.3.3. Touch Position

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-9041	(16bit) : touch status word(bit 0 on = user is touching the screen)	R	R	R	Y
LW-9042	(16bit) : touch x position	R	R	R	Y

LW-9043	(16bit) : touch y position	R	R	R	Y
LW-9044	(16bit) : leave x position	R	R	R	Y
LW-9045	(16bit) : leave y position	R	R	R	Y

Click the icon to download the demo project that explains how to how to trigger relevant registers to change page with finger slide. Please confirm your internet connection.

22.3.4. Local HMI Network Information

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-12041	refresh HMI ethernet information (DHCP, gateway, netmask, IP) (set ON)	R/W	R/C	R/C	Y
LB-12094	update ethernet 1 setting (IP, netmask, gateway) (set ON)	R/W	R/C	R/C	Y
LB-12095	update ethernet 2 setting (IP, netmask) (set ON)	R/W	R/C	R/C	Y
LW-9125	(16bit) : HMI ethernet 1 gateway 0 (for HMI use only)	R/W	R/C	R/C	Y
LW-9126	(16bit) : HMI ethernet 1 gateway 1 (for HMI use only)	R/W	R/C	R/C	Y
LW-9127	(16bit) : HMI ethernet 1 gateway 2 (for HMI use only)	R/W	R/C	R/C	Y
LW-9128	(16bit) : HMI ethernet 1 gateway 3 (for HMI use only)	R/W	R/C	R/C	Y
LW-9129	(16bit) : HMI ethernet 1 IP 0 (for HMI use only)	R/W	R/C	R/C	Y
LW-9130	(16bit) : HMI ethernet 1 IP 1 (for HMI use only)	R/W	R/C	R/C	Y
LW-9131	(16bit) : HMI ethernet 1 IP 2 (for HMI use only)	R/W	R/C	R/C	Y
LW-9132	(16bit) : HMI ethernet 1 IP 3 (for HMI use only)	R/W	R/C	R/C	Y
LW-9133	(16bit) : ethernet port no.	R	R	R	Y
LW-9135	(16bit) : HMI media access control (MAC) address 0	R	R	R	Y
LW-9136	(16bit) : HMI media access control (MAC) address 1	R	R	R	Y
LW-9137	(16bit) : HMI media access control (MAC) address 2	R	R	R	Y
LW-9138	(16bit) : HMI media access control (MAC) address 3	R	R	R	Y

LW-9139	(16bit) : HMI media access control (MAC) address 4	R	R	R	Y
LW-9140	(16bit) : HMI media access control (MAC) address 5	R	R	R	Y
LW-10750	(16bit) : HMI ethernet 1 Mask 0 (for HMI use only)	R/W	R/C	R/C	Y
LW-10751	(16bit) : HMI ethernet 1 Mask 1 (for HMI use only)	R/W	R/C	R/C	Y
LW-10752	(16bit) : HMI ethernet 1 Mask 2 (for HMI use only)	R/W	R/C	R/C	Y
LW-10753	(16bit) : HMI ethernet 1 Mask 3 (for HMI use only)	R/W	R/C	R/C	Y
LW-10786	(16bit) : HMI ethernet 2 IP 0 (for HMI use only)	R/W	R/C	R/C	Y
LW-10787	(16bit) : HMI ethernet 2 IP 1 (for HMI use only)	R/W	R/C	R/C	Y
LW-10788	(16bit) : HMI ethernet 2 IP 2 (for HMI use only)	R/W	R/C	R/C	Y
LW-10789	(16bit) : HMI ethernet 2 IP 3 (for HMI use only)	R/W	R/C	R/C	Y
LW-10790	(16bit) : HMI ethernet 2 netmask 0 (for HMI use only)	R/W	R/C	R/C	Y
LW-10791	(16bit) : HMI ethernet 2 netmask 1 (for HMI use only)	R/W	R/C	R/C	Y
LW-10792	(16bit) : HMI ethernet 2 netmask 2 (for HMI use only)	R/W	R/C	R/C	Y
LW-10793	(16bit) : HMI ethernet 2 netmask 3 (for HMI use only)	R/W	R/C	R/C	Y
LW-10794	(16bit) : HMI ethernet 2 gateway 0 (for HMI use only)	R/W	R/C	R/C	Y
LW-10795	(16bit) : HMI ethernet 2 gateway 1 (for HMI use only)	R/W	R/C	R/C	Y
LW-10796	(16bit) : HMI ethernet 2 gateway 2 (for HMI use only)	R/W	R/C	R/C	Y
LW-10797	(16bit) : HMI ethernet 2 gateway 3 (for HMI use only)	R/W	R/C	R/C	Y
LW-10798	(16bit) : ethernet 2 media access control (MAC) address 0	R	R	R	Y
LW-10799	(16bit) : ethernet 2 media access control (MAC) address 1	R	R	R	Y
LW-10800	(16bit) : ethernet 2 media access control	R	R	R	Y

	(MAC) address 2				
LW-10801	(16bit) : ethernet 2 media access control (MAC) address 3	R	R	R	Y
LW-10802	(16bit) : ethernet 2 media access control (MAC) address 4	R	R	R	Y
LW-10803	(16bit) : ethernet 2 media access control (MAC) address 5	R	R	R	Y
LW-10804	(16bit) : HMI ethernet 1 domain name system (DNS) server IP0	R/W	R/C	R/C	Y
LW-10805	(16bit) : HMI Ethernet 1 domain name system (DNS) server IP1	R/W	R/C	R/C	Y
LW-10806	(16bit) : HMI Ethernet 1 domain name system (DNS) server IP2	R/W	R/C	R/C	Y
LW-10807	(16bit) : HMI Ethernet 1 domain name system (DNS) server IP3	R/W	R/C	R/C	Y
LW-10808	(16bit) : HMI ethernet 2 domain name system (DNS) server IP0	R/W	R/C	R/C	Y
LW-10809	(16bit) : HMI ethernet 2 domain name system (DNS) server IP1	R/W	R/C	R/C	Y
LW-10810	(16bit) : HMI ethernet 2 domain name system (DNS) server IP2	R/W	R/C	R/C	Y
LW-10811	(16bit) : HMI ethernet 2 domain name system (DNS) server IP3	R/W	R/C	R/C	Y
LW-10812	(16bit) : obtain an IP address automatically (DHCP => 0 : off, 1 : on)	R/W	R/C	R/C	Y
LW-10813	(16bit) : obtain an ethernet 2 IP address automatically (DHCP => 0 : off, 1 : on)	R/W	R/C	R/C	Y
LW-10815	(16bit) : link speed of ethernet 1 (0:failure, 10 (10M), 100 (100M), 1000 (1G))	R	R	R	Y
LW-10816	(16bit) : link speed of ethernet 2 (0:failure, 10 (10M), 100 (100M), 1000 (1G))	R	R	R	Y

22.3.5. Project File Information

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-9100	(16bit) : project name (16 words)	R	R	R	Y

LW-9116	(32bit) : project size in bytes	R	R	R	Y
LW-9118	(32bit) : project size in K bytes	R	R	R	Y
LW-9120	(32bit) : EasyBuilder Pro version	R	R	R	Y
LW-9122	(16bit) : project compiled date [year]	R	R	R	Y
LW-9123	(16bit) : project compiled date [month]	R	R	R	Y
LW-9124	(16bit) : project compiled date [day]	R	R	R	Y
LW-11440	(16bit) : project compiled time [hour] (24-hour format)	R	R	R	Y
LW-11441	(16bit) : project compiled time [minute]	R	R	R	Y
LW-11442	(16bit) : project compiled time [second]	R	R	R	Y
LW-12752	(16 words) : project checksum	R	R	R	Y

22.3.6. Storage Space Management

System registers listed in this section are available exclusively for models equipped with an SD card slot or two USB ports only.

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9035	HMI free space insufficiency alarm (when ON)	R	R	R	Y
LB-9036	SD card free space insufficiency alarm (when ON)	R	R	R	Y
LB-9037	USB disk 1 free space insufficiency alarm (when ON)	R	R	R	Y
LB-9038	USB disk 2 free space insufficiency alarm (when ON) *Note 1	R	R	R	Y
LB-12048	USB disk 1 status (exists when ON)	R	R	R	Y
LB-12049	USB disk 2 status (exists when ON) *Note 1	R	R	R	Y
LB-12050	SD card status (exists when ON)	R	R	R	Y
LW-9070	(16bit) : free space insufficiency warning (Mega bytes)	R	R	R	N
LW-9071	(16bit) : reserved free space size (Mega bytes)	R	R	R	N
LW-9072	(32bit) : HMI current free space (K bytes)	R	R	R	Y
LW-9074	(32bit) : SD current free space (K bytes)	R	R	R	Y
LW-9076	(32bit) : USB disk 1 current free space (K bytes)	R	R	R	Y
LW-9078	(32bit) : USB disk 2 current free space (K bytes) *Note 1	R	R	R	N

LW-11458	(32bit) : HMI total space for history data (K bytes) *Note 2	R	R	R	Y
LW-11460	(32bit) : HMI current free space for history data (K bytes) *Note 2	R	R	R	Y
LW-12492	(4 words) : last detected USB partition ID *Note 3	R	R	R	Y
LW-12496	(32-bit float) : last detected USB partition size (GB) *Note 3	R	R	R	Y
LW-12498	(4 words) : set the partition ID to be used as USB1 *Note 3	R/W	R/C	R/C	Y
LW-12502	(4 words) : set the partition ID to be used as USB2 *Note 3	R/W	R/C	R/C	Y

Note

1. The registers relating to USB disk 2 are only supported on iP / iE and cMT X Series models.
2. Only available on cMT Series model.
3. When a USB's partition ID is manually set in LW-12498/LW-12502, only this USB disk will be identified as USB1/USB2. To remove the partition ID association, please set an empty value in LW-12498/LW-12502. When the same partition ID is used as both USB1 ID (LW-12498) and USB2 ID (LW-12502), the USB disk associated with this USB ID will be identified as USB1.

Click the icon to download the demo project that explains how to use LW-9072 ~ LW-9076 with Backup Object. Please confirm your internet connection.

22.3.7. Recipe and Extended Memory

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9028	reset all recipe data (set ON)	W	C	C	Y
LB-9029	save all recipe data to machine (set ON)	W	C	C	Y
LB-9460	EM0's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-9461	EM1's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-9462	EM2's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-9463	EM3's storage device (SD card) does not exist	R	R	R	Y

	(when ON)				
LB-9464	EM4's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-9465	EM5's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-9466	EM6's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-9467	EM7's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-9468	EM8's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-9469	EM9's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-9470	EM0's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-9471	EM1's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-9472	EM2's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-9473	EM3's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-9474	EM4's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-9475	EM5's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-9476	EM6's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-9477	EM7's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-9478	EM8's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-9479	EM9's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-9480	EM0's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y
LB-9481	EM1's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y

LB-9482	EM2's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y
LB-9483	EM3's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y
LB-9484	EM4's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y
LB-9485	EM5's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y
LB-9486	EM6's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y
LB-9487	EM7's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y
LB-9488	EM8's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y
LB-9489	EM9's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y
LB-12363	Prohibit recipe database update from remote HMI (set ON)	R/W	R/C	R/C	Y
LB-12801	EM10's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-12802	EM11's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-12803	EM12's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-12804	EM13's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-12805	EM14's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-12806	EM15's storage device (SD card) does not exist (when ON)	R	R	R	Y
LB-12807	EM10's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-12808	EM11's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-12809	EM12's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-12810	EM13's storage device (USB disk) does not	R	R	R	Y

	exist (when ON)				
LB-12811	EM14's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-12812	EM15's storage device (USB disk) does not exist (when ON)	R	R	R	Y
LB-12813	EM10's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y
LB-12814	EM11's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y
LB-12815	EM12's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y
LB-12816	EM13's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y
LB-12817	EM14's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y
LB-12818	EM15's storage device (USB disk 2) does not exist (when ON)	R	R	R	Y

22.3.8. Data Sampling

System registers listed in this section are available exclusively for models equipped with an SD card slot or two USB ports only.

Using system registers to delete files, view or update Data Sampling related file information is not supported on cMT Series models. Files can be deleted using the control address designated in Data Sampling settings window, please see Chapter 8 in this manual for more information.

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9025	delete the earliest data sampling file on HMI memory (set ON)	W	C	C	N
LB-9026	delete all data sampling files on HMI memory (set ON)	W	C	C	Y
LB-9027	refresh data sampling information on HMI memory (set ON)	W	C	C	N
LB-9034	save event/data sampling to HMI, USB disk, SD card (set ON) *Note 1	W	C	C	Y
LB-11949	delete the earliest data sampling file on SD card (set ON)	W	C	C	N

LB-11950	delete all data sampling files on SD card (set ON)	W	C	C	Y
LB-11951	refresh data sampling information on SD card (set ON)	W	C	C	N
LB-11952	delete the earliest data sampling file on USB disk 1 (set ON)	W	C	C	N
LB-11953	delete all data sampling files on USB disk 1 (set ON)	W	C	C	Y
LB-11954	refresh data sampling information on USB disk 1 (set ON)	W	C	C	N
LB-11955	delete the earliest data sampling file on USB disk 2 (set ON) *Note 3	W	C	C	N
LB-11956	delete all data sampling files on USB disk 2 (set ON) *Note 3	W	C	C	Y
LB-11957	refresh data sampling information on USB disk 2 (set ON) *Note 3	W	C	C	N
LW-9063	(16bit) : no. of data sampling files on HMI memory	R	R	R	N
LW-9064	(32bit) : size of data sampling files on HMI memory (bytes)	R	R	R	N
LW-10489	(16bit) : no. of data sampling files on SD card	R	R	R	N
LW-10490	(32bit) : size of data sampling files on SD card (bytes)	R	R	R	N
LW-10492	(16bit) : no. of data sampling files on USB disk 1	R	R	R	N
LW-10493	(32bit) : size of data sampling files on USB disk 1 (bytes)	R	R	R	N
LW-10495	(16bit) : no. of data sampling files on USB disk 2 *Note 3	R	R	R	N
LW-10496	(32bit) : size of data sampling files on USB disk 2 (bytes) *Note 3	R	R	R	N

Note

1. The shortest interval between two successful executions is 2 seconds.
2. The registers for deleting or updating data samplings do not work during simulation on PC.
3. The registers relating to USB disk 2 are only supported on iP / iE Series models.

22.3.9. Event Log

System registers listed in this section are available exclusively for models equipped with an SD card slot or two USB ports only.
Using system registers to delete files, view or update Event Log related file information is not supported on cMT Series models. Files can be deleted using the control address designated in Event Log settings window, please see Chapter 7 in this manual for more information.

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9021	reset current event log (OFF->ON)	W	C	C	Y
LB-9022	delete the earliest event log file on HMI memory (set ON)	W	C	C	N
LB-9023	delete all event log files on HMI memory (set ON)	W	C	C	Y
LB-9024	refresh event log information on HMI memory (set ON)	W	C	C	N
LB-9034	save event/data sampling to HMI, USB disk, SD card (set ON) *Note 2	W	C	C	Y
LB-9042	acknowledge all alarm events (set ON)	W	C	C	Y
LB-9043	unacknowledged events exist (when ON)	R	R	R	Y
LB-11940	delete the earliest event log file on SD card (set ON)	W	C	C	N
LB-11941	delete all event log files on SD card (set ON)	W	C	C	Y
LB-11942	refresh event log information on SD card (set ON)	W	C	C	N
LB-11943	delete the earliest event log file on USB disk 1 (set ON)	W	C	C	N
LB-11944	delete all event log files on USB disk 1 (set ON)	W	C	C	Y
LB-11945	refresh event log information on USB disk 1 (set ON)	W	C	C	N
LB-11946	delete the earliest event log file on USB disk 2 (set ON) *Note 4	W	C	C	N
LB-11947	delete all event log files on USB disk 2 (set ON) *Note 4	W	C	C	Y
LB-11948	refresh event log information on USB disk 2 (set ON) *Note 4	W	C	C	N

LB-12024	disable alarm buzzer (set ON)	W	C	C	Y
LB-12399	status is on when alarm exists in any category	R	R	R	Y
LB-12400	status is on when alarm exists in category 0	R	R	R	Y
LB-12401	status is on when alarm exists in category 1	R	R	R	Y
LB-12402	status is on when alarm exists in category 2	R	R	R	Y
LB-12403	status is on when alarm exists in category 3	R	R	R	Y
LB-12404	status is on when alarm exists in category 4	R	R	R	Y
LB-12405	status is on when alarm exists in category 5	R	R	R	Y
LB-12406	status is on when alarm exists in category 6	R	R	R	Y
LB-12407	status is on when alarm exists in category 7	R	R	R	Y
LB-12655	status is on when alarm exists in category 255	R	R	R	Y
LW-9060	(16bit) : no. of event log files on HMI memory	R	R	R	N
LW-9061	(32bit) : size of event log files on HMI memory (bytes)	R	R	R	N
LW-9450	(16bit) : time tag of event log – second *Note 1	R/W	R/C	R/C	Y
LW-9451	(16bit) : time tag of event log – minute *Note 1	R/W	R/C	R/C	Y
LW-9452	(16bit) : time tag of event log – hour *Note 1	R/W	R/C	R/C	Y
LW-9453	(16bit) : time tag of event log – day *Note 1	R/W	R/C	R/C	Y
LW-9454	(16bit) : time tag of event log – month *Note 1	R/W	R/C	R/C	Y
LW-9455	(16bit) : time tag of event log – year *Note 1	R/W	R/C	R/C	Y
LW-10480	(16bit) : no. of event log files on SD card	R	R	R	N
LW-10481	(32bit) : size of event log files on SD card (bytes)	R	R	R	N
LW-10483	(16bit) : no. of event log files on USB disk 1	R	R	R	N
LW-10484	(32bit) : size of event log files on USB disk 1 (bytes)	R	R	R	N
LW-10486	(16bit) : no. of event log files on USB disk 2 *Note 4	R	R	R	N
LW-10487	(32bit) : size of event log files on USB disk 2 (bytes) *Note 4	R	R	R	N
LW-11443	(16bit) : push notification alarm status (0 :	R	R	R	Y

	none; 1 : green; 2 : yellow; 3 : red)				
LW-11499	total no. of alarms	R	R	R	Y
LW-11500	no. of alarms in category 0	R	R	R	Y
LW-11501	no. of alarms in category 1	R	R	R	Y
LW-11502	no. of alarms in category 2	R	R	R	Y
LW-11503	no. of alarms in category 3	R	R	R	Y
LW-11504	no. of alarms in category 4	R	R	R	Y
LW-11505	no. of alarms in category 5	R	R	R	Y
LW-11506	no. of alarms in category 6	R	R	R	Y
LW-11507	no. of alarms in category 7	R	R	R	Y
LW-11755	no. of alarms in category 255	R	R	R	Y

Note

1. If LW-9450 ~ LW-9455 are used to get Event Log time, please enable in [system parameters] » [General].

   Click the icon to download the demo project that explains how to use the system registers LW-9450 to LW-9455 to be the time tag of event log. Please confirm your internet connection.
2. The shortest interval between two successful executions is 2 seconds.
3. The registers for deleting or updating event logs do not work during simulation on PC.
4. The registers relating to USB disk 2 are only supported on iP / iE Series models.

22.3.10. Station Number Variables

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-10000	(16bit) : var0 - station no variable (usage : var0#address)	R/W	R/C	R/C	Y
LW-10001	(16bit) : var1 - station no variable (usage : var1#address)	R/W	R/C	R/C	Y
LW-10002	(16bit) : var2 - station no variable (usage : var2#address)	R/W	R/C	R/C	Y
LW-10003	(16bit) : var3 - station no variable (usage : var3#address)	R/W	R/C	R/C	Y
LW-10004	(16bit) : var4 - station no variable (usage : var4#address)	R/W	R/C	R/C	Y
LW-10005	(16bit) : var5 - station no variable (usage :	R/W	R/C	R/C	Y

	var5#address)				
LW-10006	(16bit) : var6 - station no variable (usage : var6#address)	R/W	R/C	R/C	Y
LW-10007	(16bit) : var7 - station no variable (usage : var7#address)	R/W	R/C	R/C	Y
LW-10008	(16bit) : var8 - station no variable (usage : var8#address)	R/W	R/C	R/C	Y
LW-10009	(16bit) : var9 - station no variable (usage : var9#address)	R/W	R/C	R/C	Y
LW-10010	(16bit) : var10 - station no variable (usage : var10#address)	R/W	R/C	R/C	Y
LW-10011	(16bit) : var11 - station no variable (usage : var11#address)	R/W	R/C	R/C	Y
LW-10012	(16bit) : var12 - station no variable (usage : var12#address)	R/W	R/C	R/C	Y
LW-10013	(16bit) : var13 - station no variable (usage : var13#address)	R/W	R/C	R/C	Y
LW-10014	(16bit) : var14 - station no variable (usage : var14#address)	R/W	R/C	R/C	Y
LW-10015	(16bit) : var15 - station no variable (usage : var15#address)	R/W	R/C	R/C	Y

Click the icon to download the demo project. Please confirm your internet connection.

22.3.11. Index Registers

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-9000	(16bit) : index register of RWI	R/W	R/C	R/C	Y
LW-9200	(16bit) : address index 0	R/W	R/C	R/C	Y
LW-9201	(16bit) : address index 1	R/W	R/C	R/C	Y
LW-9202	(16bit) : address index 2	R/W	R/C	R/C	Y
LW-9203	(16bit) : address index 3	R/W	R/C	R/C	Y
LW-9204	(16bit) : address index 4	R/W	R/C	R/C	Y
LW-9205	(16bit) : address index 5	R/W	R/C	R/C	Y
LW-9206	(16bit) : address index 6	R/W	R/C	R/C	Y
LW-9207	(16bit) : address index 7	R/W	R/C	R/C	Y
LW-9208	(16bit) : address index 8	R/W	R/C	R/C	Y
LW-9209	(16bit) : address index 9	R/W	R/C	R/C	Y
LW-9210	(16bit) : address index 10	R/W	R/C	R/C	Y
LW-9211	(16bit) : address index 11	R/W	R/C	R/C	Y
LW-9212	(16bit) : address index 12	R/W	R/C	R/C	Y
LW-9213	(16bit) : address index 13	R/W	R/C	R/C	Y
LW-9214	(16bit) : address index 14	R/W	R/C	R/C	Y
LW-9215	(16bit) : address index 15	R/W	R/C	R/C	Y
LW-9230	(32bit) : address index 16	R/W	R/C	R/C	Y
LW-9232	(32bit) : address index 17	R/W	R/C	R/C	Y
LW-9234	(32bit) : address index 18	R/W	R/C	R/C	Y
LW-9236	(32bit) : address index 19	R/W	R/C	R/C	Y
LW-9238	(32bit) : address index 20	R/W	R/C	R/C	Y
LW-9240	(32bit) : address index 21	R/W	R/C	R/C	Y
LW-9242	(32bit) : address index 22	R/W	R/C	R/C	Y
LW-9244	(32bit) : address index 23	R/W	R/C	R/C	Y
LW-9246	(32bit) : address index 24	R/W	R/C	R/C	Y
LW-9248	(32bit) : address index 25	R/W	R/C	R/C	Y
LW-9250	(32bit) : address index 26	R/W	R/C	R/C	Y
LW-9252	(32bit) : address index 27	R/W	R/C	R/C	Y
LW-9254	(32bit) : address index 28	R/W	R/C	R/C	Y
LW-9256	(32bit) : address index 29	R/W	R/C	R/C	Y
LW-9258	(32bit) : address index 30	R/W	R/C	R/C	Y
LW-9260	(32bit) : address index 31	R/W	R/C	R/C	Y

PLW-9200	(16bit) : private address index 0	R/W	N/A	N/A	Y
PLW-9201	(16bit) : private address index 1	R/W	N/A	N/A	Y
PLW-9202	(16bit) : private address index 2	R/W	N/A	N/A	Y
PLW-9203	(16bit) : private address index 3	R/W	N/A	N/A	Y
PLW-9204	(16bit) : private address index 4	R/W	N/A	N/A	Y
PLW-9205	(16bit) : private address index 5	R/W	N/A	N/A	Y
PLW-9206	(16bit) : private address index 6	R/W	N/A	N/A	Y
PLW-9207	(16bit) : private address index 7	R/W	N/A	N/A	Y
PLW-9208	(16bit) : private address index 8	R/W	N/A	N/A	Y
PLW-9209	(16bit) : private address index 9	R/W	N/A	N/A	Y
PLW-9210	(16bit) : private address index 10	R/W	N/A	N/A	Y
PLW-9211	(16bit) : private address index 11	R/W	N/A	N/A	Y
PLW-9212	(16bit) : private address index 12	R/W	N/A	N/A	Y
PLW-9213	(16bit) : private address index 13	R/W	N/A	N/A	Y
PLW-9214	(16bit) : private address index 14	R/W	N/A	N/A	Y
PLW-9215	(16bit) : private address index 15	R/W	N/A	N/A	Y
PLW-9230	(32bit) : private address index 16	R/W	N/A	N/A	Y
PLW-9232	(32bit) : private address index 17	R/W	N/A	N/A	Y
PLW-9234	(32bit) : private address index 18	R/W	N/A	N/A	Y
PLW-9236	(32bit) : private address index 19	R/W	N/A	N/A	Y
PLW-9238	(32bit) : private address index 20	R/W	N/A	N/A	Y
PLW-9240	(32bit) : private address index 21	R/W	N/A	N/A	Y
PLW-9242	(32bit) : private address index 22	R/W	N/A	N/A	Y
PLW-9244	(32bit) : private address index 23	R/W	N/A	N/A	Y
PLW-9246	(32bit) : private address index 24	R/W	N/A	N/A	Y
PLW-9248	(32bit) : private address index 25	R/W	N/A	N/A	Y
PLW-9250	(32bit) : private address index 26	R/W	N/A	N/A	Y
PLW-9252	(32bit) : private address index 27	R/W	N/A	N/A	Y
PLW-9254	(32bit) : private address index 28	R/W	N/A	N/A	Y
PLW-9256	(32bit) : private address index 29	R/W	N/A	N/A	Y
PLW-9258	(32bit) : private address index 30	R/W	N/A	N/A	Y
PLW-9260	(32bit) : private address index 31	R/W	N/A	N/A	Y

Click the icon to download the demo project. Please confirm your internet connection.

22.3.12. MODBUS Server Communication

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9055	MODBUS server (COM 1) receives a request (when ON)	R	R	R	Y
LB-9056	MODBUS server (COM 2) receives a request (when ON)	R	R	R	Y
LB-9057	MODBUS server (COM 3) receives a request (when ON)	R	R	R	Y
LB-9058	MODBUS server (ethernet) receives a request (when ON)	R	R	R	Y
LB-12052	MODBUS server status (disabled when ON)	R/W	R/C	R/C	Y
LW-9270	(16bit) : request's function code - MODBUS server (COM 1)	R	R	R	Y
LW-9271	(16bit) : request's starting address - MODBUS server (COM 1)	R	R	R	Y
LW-9272	(16bit) : request's quantity of registers - MODBUS server (COM 1)	R	R	R	Y
LW-9275	(16bit) : request's function code - MODBUS server (COM 2)	R	R	R	Y
LW-9276	(16bit) : request's starting address - MODBUS server (COM 2)	R	R	R	Y
LW-9277	(16bit) : request's quantity of registers - MODBUS server (COM 2)	R	R	R	Y
LW-9280	(16bit) : request's function code - MODBUS server (COM 3)	R	R	R	Y
LW-9281	(16bit) : request's starting address - MODBUS server (COM 3)	R	R	R	Y
LW-9282	(16bit) : request's quantity of registers - MODBUS server (COM 3)	R	R	R	Y
LW-9285	(16bit) : request's function code - MODBUS server (ethernet)	R	R	R	Y
LW-9286	(16bit) : request's starting address - MODBUS server (ethernet)	R	R	R	Y
LW-9287	(16bit) : request's quantity of registers - MODBUS server (ethernet)	R	R	R	Y
LW-9288	(16bit) : last error code - MODBUS server	R	R	R	Y

	(ethernet)				
LW-9289	(16bit) : last error code - MODBUS server (COM 1)	R	R	R	Y
LW-9290	(16bit) : last error code - MODBUS server (COM 2)	R	R	R	Y
LW-9291	(16bit) : last error code - MODBUS server (COM 3)	R	R	R	Y
LW-9541	(16bit) : MODBUS/ASCII server station no. (COM 1)	R/W	R/C	R/C	Y
LW-9542	(16bit) : MODBUS/ASCII server station no. (COM 2)	R/W	R/C	R/C	Y
LW-9543	(16bit) : MODBUS/ASCII server station no. (COM 3)	R/W	R/C	R/C	Y
LW-9544	(16bit) : MODBUS/ASCII server station no. (ethernet)	R/W	R/C	R/C	Y
LW-9570	(32bit) : received data count (bytes) (COM 1 MODBUS server)	R	R	R	Y
LW-9572	(32bit) : received data count (bytes) (COM 2 MODBUS server)	R	R	R	Y
LW-9574	(32bit) : received data count (bytes) (COM 3 MODBUS server)	R	R	R	Y
LW-9576	(32bit) : received data count (bytes) (Ethernet MODBUS server)	R	R	R	Y

22.3.13. Communication Parameter Settings

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9030	activate COM 1 new communication settings (LW-9550~9554) (set ON)	R/W	R/C	R/C	Y
LB-9031	activate COM 2 new communication settings (LW-9555~9559) (set ON)	R/W	R/C	R/C	Y
LB-9032	activate COM 3 new communication settings (LW-9560~9564) (set ON)	R/W	R/C	R/C	Y
LB-9065	disable/enable COM 1 broadcast station no.	R/W	R/C	R/C	Y
LB-9066	disable/enable COM 2 broadcast station no.	R/W	R/C	R/C	Y
LB-9067	disable/enable COM 3 broadcast station no.	R/W	R/C	R/C	Y

LW-9550	(16bit) : COM 1 mode (0:RS232,1:RS485 2W,2:RS485 4W) (use LB-9030 to activate all communication settings)	R/W	R/C	R/C	Y
LW-9551	(16bit) : COM 1 baud rate (7:1200,8:2400,0:4800,1:9600,10:14400, 2:19200,11:28800,3:38400,4:57600,..) *Note 1	R/W	R/C	R/C	Y
LW-9552	(16bit) : COM 1 databits (7 : 7 bits, 8 : 8 bits)	R/W	R/C	R/C	Y
LW-9553	(16bit) : COM 1 parity (0:none, 1:even, 2:odd, 3:mark, 4:space)	R/W	R/C	R/C	Y
LW-9554	(16bit) : COM 1 stop bits (1 : 1 bit, 2 : 2 bits)	R/W	R/C	R/C	Y
LW-9555	(16bit) : COM 2 mode (0:RS232,1:RS485 2W,2:RS485 4W) (use LB-9031 to activate all communication settings)	R/W	R/C	R/C	Y
LW-9556	(16bit) : COM 2 baud rate (7:1200,8:2400,0:4800,1:9600,10:14400, 2:19200,11:28800,3:38400,4:57600,..) *Note 1	R/W	R/C	R/C	Y
LW-9557	(16bit) : COM 2 databits (7 : 7 bits, 8 : 8 bits)	R/W	R/C	R/C	Y
LW-9558	(16bit) : COM 2 parity (0:none, 1:even, 2:odd, 3:mark, 4:space)	R/W	R/C	R/C	Y
LW-9559	(16bit) : COM 2 stop bits (1 : 1 bit, 2 : 2 bits)	R/W	R/C	R/C	Y
LW-9560	(16bit) : COM 3 mode (0:RS232,1:RS485 2W) (use LB-9032 to activate all communication settings)	R/W	R/C	R/C	Y
LW-9561	(16bit) : COM 3 baud rate (7:1200,8:2400,0:4800,1:9600,10:14400, 2:19200,11:28800,3:38400,4:57600,..) *Note 1	R/W	R/C	R/C	Y
LW-9562	(16bit) : COM 3 databits (7 : 7 bits, 8 : 8 bits)	R/W	R/C	R/C	Y
LW-9563	(16bit) : COM 3 parity (0:none, 1:even, 2:odd, 3:mark, 4:space)	R/W	R/C	R/C	Y
LW-9564	(16bit) : COM 3 stop bits (1 : 1 bit, 2 : 2 bits)	R/W	R/C	R/C	Y
LW-9565	(16bit) : COM 1 broadcast station no.	R/W	R/C	R/C	Y
LW-9566	(16bit) : COM 2 broadcast station no.	R/W	R/C	R/C	Y
LW-9567	(16bit) : COM 3 broadcast station no.	R/W	R/C	R/C	Y
LW-10500	(16bit) : device 1 timeout (unit : 100ms, 0 :	R/W	R/C	R/C	Y

	50ms)				
LW-10501	(16bit) : device 1 turn around delay (unit : ms)	R/W	R/C	R/C	Y
LW-10502	(16bit) : device 1 send ACK delay (unit : ms)	R/W	R/C	R/C	Y
LW-10503	(16bit) : device 1 parameter 1	R/W	R/C	R/C	Y
LW-10504	(16bit) : device 1 parameter 2	R/W	R/C	R/C	Y
LW-10505	(16bit) : device 2 timeout (unit : 100ms, 0 : 50ms)	R/W	R/C	R/C	Y
LW-10506	(16bit) : device 2 turn around delay (unit : ms)	R/W	R/C	R/C	Y
LW-10507	(16bit) : device 2 send ACK delay (unit : ms)	R/W	R/C	R/C	Y
LW-10508	(16bit) : device 2 parameter 1	R/W	R/C	R/C	Y
LW-10509	(16bit) : device 2 parameter 2	R/W	R/C	R/C	Y
LW-10510	(16bit) : device 3 timeout (unit : 100ms, 0 : 50ms)	R/W	R/C	R/C	Y
LW-10511	(16bit) : device 3 turn around delay (unit : ms)	R/W	R/C	R/C	Y
LW-10512	(16bit) : device 3 send ACK delay (unit : ms)	R/W	R/C	R/C	Y
LW-10513	(16bit) : device 3 parameter 1	R/W	R/C	R/C	Y
LW-10514	(16bit) : device 3 parameter 2	R/W	R/C	R/C	Y
LW-10515	(16bit) : device 4 timeout (unit : 100ms)	R/W	R/C	R/C	Y
LW-10516	(16bit) : device 4 turn around delay (unit : ms)	R/W	R/C	R/C	Y
LW-10517	(16bit) : device 4 send ACK delay (unit : ms) (SIEMENS S7/400 Link type)	R/W	R/C	R/C	Y
LW-10518	(16bit) : device 4 parameter 1 (SIEMENS S7/400 rack)	R/W	R/C	R/C	Y
LW-10519	(16bit) : device 4 parameter 2 (SIEMENS S7/400 CPU slot)	R/W	R/C	R/C	Y
LW-10520	(16bit) : device 5 timeout (unit : 100ms)	R/W	R/C	R/C	Y
LW-10521	(16bit) : device 5 turn around delay (unit : ms)	R/W	R/C	R/C	Y
LW-10522	(16bit) : device 5 send ACK delay (unit : ms) (SIEMENS S7/400 Link type)	R/W	R/C	R/C	Y
LW-10523	(16bit) : device 5 parameter 1 (SIEMENS S7/400 rack)	R/W	R/C	R/C	Y
LW-10524	(16bit) : device 5 parameter 2 (SIEMENS S7/400 CPU slot)	R/W	R/C	R/C	Y
LW-10525	(16bit) : device 6 timeout (unit : 100ms)	R/W	R/C	R/C	Y
LW-10526	(16bit) : device 6 turn around delay (unit : ms)	R/W	R/C	R/C	Y
LW-10527	(16bit) : device 6 send ACK delay (unit : ms) (SIEMENS S7/400 Link type)	R/W	R/C	R/C	Y

LW-10528	(16bit) : device 6 parameter 1 (SIEMENS S7/400 rack)	R/W	R/C	R/C	Y
LW-10529	(16bit) : device 6 parameter 2 (SIEMENS S7/400 CPU slot)	R/W	R/C	R/C	Y
LW-10530	(16bit) : device 7 timeout (unit : 100ms)	R/W	R/C	R/C	Y
LW-10531	(16bit) : device 7 turn around delay (unit : ms)	R/W	R/C	R/C	Y
LW-10532	(16bit) : device 7 send ACK delay (unit : ms) (SIEMENS S7/400 Link type)	R/W	R/C	R/C	Y
LW-10533	(16bit) : device 7 parameter 1 (SIEMENS S7/400 rack)	R/W	R/C	R/C	Y
LW-10534	(16bit) : device 7 parameter 2 (SIEMENS S7/400 CPU slot)	R/W	R/C	R/C	Y
LW-10535	(16bit) : device 8 timeout (unit : 100ms)	R/W	R/C	R/C	Y
LW-10536	(16bit) : device 8 turn around delay (unit : ms)	R/W	R/C	R/C	Y
LW-10537	(16bit) : device 8 send ACK delay (unit : ms) (SIEMENS S7/400 Link type)	R/W	R/C	R/C	Y
LW-10538	(16bit) : device 8 parameter 1 (SIEMENS S7/400 rack)	R/W	R/C	R/C	Y
LW-10539	(16bit) : device 8 parameter 2 (SIEMENS S7/400 CPU slot)	R/W	R/C	R/C	Y
LW-10655	(16bit) : device 32 timeout (unit : 100ms)	R/W	R/C	R/C	Y
LW-10656	(16bit) : device 32 turn around delay (unit : ms)	R/W	R/C	R/C	Y
LW-10657	(16bit) : device 32 send ACK delay (unit : ms)	R/W	R/C	R/C	Y
LW-10658	(16bit) : device 32 parameter 1	R/W	R/C	R/C	Y
LW-10659	(16bit) : device 32 parameter 2	R/W	R/C	R/C	Y

Note

1. The Baud Rates are: 0:4800, 1:9600, 2:19200, 3:38400, 4:57600, 5:115200, 6:187.5K, 7:1200, 8:2400, 10:14400, 11:28800, 12:76800

22.3.14. Communication Status and Control with Device (COM)

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9150	auto. connection for device 1 (COM 1) (when ON)	R/W	R/C	R/C	Y
LB-9151	auto. connection for device 2 (COM 2) (when ON)	R/W	R/C	R/C	Y
LB-9152	auto. connection for device 3 (COM 3) (when ON)	R/W	R/C	R/C	Y
LB-9200	device 1 status (SN0, COM 1), set on to retry connection.	R/W	R/C	R/C	Y
LB-9201	device 1 status (SN1, COM 1), set on to retry connection	R/W	R/C	R/C	Y
LB-9202	device 1 status (SN2, COM 1), set on to retry connection	R/W	R/C	R/C	Y
LB-9203	device 1 status (SN3, COM 1), set on to retry connection	R/W	R/C	R/C	Y
LB-9204	device 1 status (SN4, COM 1), set on to retry connection	R/W	R/C	R/C	Y
LB-9205	device 1 status (SN5, COM 1), set on to retry connection	R/W	R/C	R/C	Y
LB-9206	device 1 status (SN6, COM 1), set on to retry connection	R/W	R/C	R/C	Y
LB-9207	device 1 status (SN7, COM 1), set on to retry connection	R/W	R/C	R/C	Y
LB-9455	device 1 status (SN255, COM 1), set on to retry connection	R/W	R/C	R/C	Y
LB-9500	device 2 status (SN0, COM 2), set on to retry connection.	R/W	R/C	R/C	Y
LB-9501	device 2 status (SN1, COM 2), set on to retry connection	R/W	R/C	R/C	Y
LB-9502	device 2 status (SN2, COM 2), set on to retry connection	R/W	R/C	R/C	Y
LB-9503	device 2 status (SN3, COM 2), set on to retry connection	R/W	R/C	R/C	Y
LB-9504	device 2 status (SN4, COM 2), set on to retry connection	R/W	R/C	R/C	Y

LB-9505	device 2 status (SN5, COM 2), set on to retry connection	R/W	R/C	R/C	Y
LB-9506	device 2 status (SN6, COM 2), set on to retry connection	R/W	R/C	R/C	Y
LB-9507	device 2 status (SN7, COM 2), set on to retry connection	R/W	R/C	R/C	Y
LB-9755	device 2 status (SN255, COM 2), set on to retry connection	R/W	R/C	R/C	Y
LB-9800	device 3 status (SN0, COM 3), set on to retry connection	R/W	R/C	R/C	Y
LB-9801	device 3 status (SN1, COM 3), set on to retry connection	R/W	R/C	R/C	Y
LB-9802	device 3 status (SN2, COM 3), set on to retry connection	R/W	R/C	R/C	Y
LB-9803	device 3 status (SN3, COM 3), set on to retry connection	R/W	R/C	R/C	Y
LB-9804	device 3 status (SN4, COM 3), set on to retry connection	R/W	R/C	R/C	Y
LB-9805	device 3 status (SN5, COM 3), set on to retry connection	R/W	R/C	R/C	Y
LB-9806	device 3 status (SN6, COM 3), set on to retry connection	R/W	R/C	R/C	Y
LB-9807	device 3 status (SN7, COM 3), set on to retry connection	R/W	R/C	R/C	Y
LB-10055	device 3 status (SN255, COM 3), set on to retry connection	R/W	R/C	R/C	Y
LB-12030	COM 1 status (OFF : normal, ON : open failed) *Note 1	R	R	R	Y
LB-12031	COM 2 status (OFF : normal, ON : open failed)	R	R	R	Y
LB-12032	COM 3 status (OFF : normal, ON : open failed)	R	R	R	Y
LB-12033	COM 4 status (OFF : normal, ON : open failed)	R	R	R	Y
LB-12034	COM 5 status (OFF : normal, ON : open failed)	R	R	R	Y
LB-12035	COM 6 status (OFF : normal, ON : open failed)	R	R	R	Y
LB-12036	COM 7 status (OFF : normal, ON : open failed)	R	R	R	Y
LB-12037	COM 8 status (OFF : normal, ON : open failed)	R	R	R	Y
LB-12038	COM 9 status (OFF : normal, ON : open failed)	R	R	R	Y
LW-9351	(16bit) : pending command no. in device 1	R	R	R	Y

	(COM 1)				
LW-9352	(16bit) : pending command no. in device 2 (COM 2)	R	R	R	Y
LW-9353	(16bit) : pending command no. in device 3 (COM 3)	R	R	R	Y

Note

1. The ON state of COM is for checking if COM is occupied by other program during simulation on PC.

22.3.15. Communication Status and Control with Device (Ethernet)

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9153	auto. connection for device 4 (ethernet) (when ON)	R/W	R/C	R/C	Y
LB-9154	auto. connection for device 5 (ethernet) (when ON)	R/W	R/C	R/C	Y
LB-9155	auto. connection for device 6 (ethernet) (when ON)	R/W	R/C	R/C	Y
LB-9156	auto. connection for device 7 (ethernet) (when ON)	R/W	R/C	R/C	Y
LB-9157	auto. connection for device 8 (ethernet) (when ON)	R/W	R/C	R/C	Y
LB-9158	auto. connection for device 9 (ethernet) (when ON)	R/W	R/C	R/C	Y
LB-9189	auto. connection for device 40 (ethernet) (when ON)	R/W	R/C	R/C	Y
LB-10070	forced to reconnect device 4 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/C	R/C	Y
LB-10071	forced to reconnect device 5 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/C	R/C	Y
LB-10072	forced to reconnect device 6 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/C	R/C	Y
LB-10073	forced to reconnect device 7 (ethernet) when	R/W	R/C	R/C	Y

	IP or system parameters changed on-line (set ON)				
LB-10074	forced to reconnect device 8 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/C	R/C	Y
LB-10075	forced to reconnect device 9 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/C	R/C	Y
LB-10099	forced to reconnect device 33 (ethernet) when IP or system parameters changed on-line (set ON)	R/W	R/C	R/C	Y
LB-10100	device 4 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10101	device 4 status (SN0, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10102	device 4 status (SN1, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10103	device 4 status (SN2, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10104	device 4 status (SN3, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10105	device 4 status (SN4, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10106	device 4 status (SN5, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10107	device 4 status (SN6, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10108	device 4 status (SN7, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10356	device 4 status (SN255, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10400	device 5 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10401	device 5 status (SN0, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10402	device 5 status (SN1, ethernet), set on to retry connection	R/W	R/C	R/C	Y

LB-10403	device 5 status (SN2, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10404	device 5 status (SN3, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10405	device 5 status (SN4, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10406	device 5 status (SN5, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10407	device 5 status (SN6, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10408	device 5 status (SN7, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10656	device 5 status (SN255, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10700	device 6 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10701	device 6 status (SN0, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10702	device 6 status (SN1, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10703	device 6 status (SN2, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10704	device 6 status (SN3, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10705	device 6 status (SN4, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10706	device 6 status (SN5, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10707	device 6 status (SN6, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10708	device 6 status (SN7, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-10956	device 6 status (SN255, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11000	device 7 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11001	device 7 status (SN0, ethernet), set on to	R/W	R/C	R/C	Y

	retry connection				
LB-11002	device 7 status (SN1, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11003	device 7 status (SN2, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11004	device 7 status (SN3, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11005	device 7 status (SN4, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11006	device 7 status (SN5, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11007	device 7 status (SN6, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11008	device 7 status (SN7, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11256	device 7 status (SN255, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11300	device 8 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11301	device 8 status (SN0, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11302	device 8 status (SN1, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11303	device 8 status (SN2, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11304	device 8 status (SN3, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11305	device 8 status (SN4, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11306	device 8 status (SN5, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11307	device 8 status (SN6, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11308	device 8 status (SN7, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11556	device 8 status (SN255, ethernet), set on to retry connection	R/W	R/C	R/C	Y

LB-11600	device 9 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11601	device 9 status (SN0, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11602	device 9 status (SN1, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11603	device 9 status (SN2, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11604	device 9 status (SN3, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11605	device 9 status (SN4, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11606	device 9 status (SN5, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11607	device 9 status (SN6, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11608	device 9 status (SN7, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11856	device 9 status (SN255, ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11900	device 10 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11901	device 11 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11902	device 12 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11903	device 13 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11904	device 14 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11905	device 15 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11906	device 16 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-11939	device 49 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-12670	auto. connection for device 41 (ethernet)	R/W	R/C	R/C	Y

	(when ON)				
LB-12671	auto. connection for device 42 (ethernet) (when ON)	R/W	R/C	R/C	Y
LB-12672	auto. connection for device 43 (ethernet) (when ON)	R/W	R/C	R/C	Y
LB-12673	auto. connection for device 44 (ethernet) (when ON)	R/W	R/C	R/C	Y
LB-12693	auto. connection for device 64 (ethernet) (when ON)	R/W	R/C	R/C	Y
LB-12700	device 50 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-12701	device 51 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-12702	device 52 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-12703	device 53 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-12714	device 64 status (ethernet), set on to retry connection	R/W	R/C	R/C	Y
LB-12720	forced to reconnect device 34 (ethernet) when IP or system	R/W	R/C	R/C	Y
LB-12721	forced to reconnect device 35 (ethernet) when IP or system	R/W	R/C	R/C	Y
LB-12722	forced to reconnect device 36 (ethernet) when IP or system	R/W	R/C	R/C	Y
LB-12723	forced to reconnect device 37 (ethernet) when IP or system	R/W	R/C	R/C	Y
LB-12750	forced to reconnect device 64 (ethernet) when IP or system	R/W	R/C	R/C	Y
LW-9354	(16bit) : pending command no. in device 4 (ethernet)	R	R	R	Y
LW-9355	(16bit) : pending command no. in device 5 (ethernet)	R	R	R	Y
LW-9356	(16bit) : pending command no. in device 6 (ethernet)	R	R	R	Y
LW-9357	(16bit) : pending command no. in device 7 (ethernet)	R	R	R	Y

LW-9389	(16bit) : pending command no. in device 39 (ethernet)	R	R	R	Y
LW-9600	(16bit) : device 4's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9601	(16bit) : device 4's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9602	(16bit) : device 4's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9603	(16bit) : device 4's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9604	(16bit) : device 4's port no.	R/W	R/C	R/C	Y
LW-9605	(16bit) : device 5's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9606	(16bit) : device 5's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9607	(16bit) : device 5's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9608	(16bit) : device 5's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9609	(16bit) : device 5's port no.	R/W	R/C	R/C	Y
LW-9610	(16bit) : device 6's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9611	(16bit) : device 6's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9612	(16bit) : device 6's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9613	(16bit) : device 6's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9614	(16bit) : device 6's port no.	R/W	R/C	R/C	Y
LW-9615	(16bit) : device 7's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9616	(16bit) : device 7's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9617	(16bit) : device 7's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9618	(16bit) : device 7's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y

LW-9619	(16bit) : device 7's port no.	R/W	R/C	R/C	Y
LW-9620	(16bit) : device 8's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9621	(16bit) : device 8's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9622	(16bit) : device 8's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9623	(16bit) : device 8's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9624	(16bit) : device 8's port no.	R/W	R/C	R/C	Y
LW-9625	(16bit) : device 9's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9626	(16bit) : device 9's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9627	(16bit) : device 9's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9628	(16bit) : device 9's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9629	(16bit) : device 9's port no.	R/W	R/C	R/C	Y
LW-9765	(16bit) : device 37's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9766	(16bit) : device 37's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9767	(16bit) : device 37's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9768	(16bit) : device 37's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9769	(16bit) : device 37's port no.	R/W	R/C	R/C	Y
LW-11472	(16bit) : device 4's ID0 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11473	(16bit) : device 4's ID1 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11474	(16bit) : device 4's ID2 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11475	(16bit) : device 4's ID3 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11476	(16bit) : device 4's ID4 (Beckhoff AMS NetId =	R/W	R/C	R/C	Y

	ID0:ID1:ID2:ID3:ID4:ID5)				
LW-11477	(16bit) : device 4's ID5 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11478	(16bit) : device 5's ID0 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11479	(16bit) : device 5's ID1 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11480	(16bit) : device 5's ID2 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11481	(16bit) : device 5's ID3 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11482	(16bit) : device 5's ID4 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11483	(16bit) : device 5's ID5 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11484	(16bit) : device 6's ID0 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11485	(16bit) : device 6's ID1 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11486	(16bit) : device 6's ID2 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11487	(16bit) : device 6's ID3 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11488	(16bit) : device 6's ID4 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11489	(16bit) : device 6's ID5 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11490	(16bit) : device 7's ID0 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11491	(16bit) : device 7's ID1 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11492	(16bit) : device 7's ID2 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11493	(16bit) : device 7's ID3 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-11494	(16bit) : device 7's ID4 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y

LW-11495	(16bit) : device 7's ID5 (Beckhoff AMS NetId = ID0:ID1:ID2:ID3:ID4:ID5)	R/W	R/C	R/C	Y
LW-12110	(16bit) : device 38's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12111	(16bit) : device 38's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12112	(16bit) : device 38's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12113	(16bit) : device 38's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12114	(16bit) : device 38's port no.	R/W	R/C	R/C	Y
LW-12115	(16bit) : device 39's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12116	(16bit) : device 39's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12117	(16bit) : device 39's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12118	(16bit) : device 39's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12119	(16bit) : device 39's port no.	R/W	R/C	R/C	Y
LW-12120	(16bit) : device 40's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12121	(16bit) : device 40's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12122	(16bit) : device 40's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12123	(16bit) : device 40's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12124	(16bit) : device 40's port no.	R/W	R/C	R/C	Y
LW-12125	(16bit) : device 41's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12126	(16bit) : device 41's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12127	(16bit) : device 41's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12128	(16bit) : device 41's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y

LW-12129	(16bit) : device 41's port no.	R/W	R/C	R/C	Y
LW-12240	(16bit) : device 64's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12241	(16bit) : device 64's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12242	(16bit) : device 64's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12243	(16bit) : device 64's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12244	(16bit) : device 64's port no.	R/W	R/C	R/C	Y

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

22.3.16. Communication Status and Control with Device (USB)

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9190	auto. connection for device (USB) (when ON)	R/W	R/C	R/C	Y
LB-9191	device status (USB), set on to retry connection	R/W	R/C	R/C	Y
LW-9390	(16bit) : pending command no. in device (USB)	R	R	R	Y

22.3.17. Communication Status and Control with Device (CAN Bus)

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-12080	auto. connection for device (CAN Bus) (when ON)	R/W	R/C	R/C	Y
LB-12081	device status (CAN Bus) set on to retry connection	R/W	R/C	R/C	Y
LB-12100	pause CAN Bus device 1 communication (when ON)	R/W	R/C	R/C	Y
LB-12101	pause CAN Bus device 2 communication (when ON)	R/W	R/C	R/C	Y
LB-12102	pause CAN Bus device 3 communication (when ON)	R/W	R/C	R/C	Y
LB-12103	pause CAN Bus device 4 communication (when ON)	R/W	R/C	R/C	Y
LB-12104	pause CAN Bus device 5 communication (when ON)	R/W	R/C	R/C	Y
LB-12105	pause CAN Bus device 6 communication (when ON)	R/W	R/C	R/C	Y
LB-12106	pause CAN Bus device 7 communication (when ON)	R/W	R/C	R/C	Y
LB-12107	pause CAN Bus device 8 communication (when ON)	R/W	R/C	R/C	Y
LB-12108	pause CAN Bus device 9 communication (when ON)	R/W	R/C	R/C	Y
LB-12109	pause CAN Bus device 10 communication (when ON)	R/W	R/C	R/C	Y
LB-12354	pause CAN Bus device 255 communication (when ON)	R/W	R/C	R/C	Y
LW-9392	(16bit) : pending command no. in device (CAN Bus)	R	R	R	Y

22.3.18. Communication Status and Control with Remote HMI

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9068	auto. connection for remote HMI 1 (when ON)	R/W	R/C	R/C	Y
LB-9069	auto. connection for remote HMI 2 (when ON)	R/W	R/C	R/C	Y
LB-9070	auto. connection for remote HMI 3 (when ON)	R/W	R/C	R/C	Y
LB-9071	auto. connection for remote HMI 4 (when ON)	R/W	R/C	R/C	Y
LB-9072	auto. connection for remote HMI 5 (when ON)	R/W	R/C	R/C	Y
LB-9073	auto. connection for remote HMI 6 (when ON)	R/W	R/C	R/C	Y
LB-9074	auto. connection for remote HMI 7 (when ON)	R/W	R/C	R/C	Y
LB-9075	auto. connection for remote HMI 8 (when ON)	R/W	R/C	R/C	Y
LB-9099	auto. connection for remote HMI 32 (when ON)	R/W	R/C	R/C	Y
LB-9100	remote HMI 1 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-9101	remote HMI 2 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-9102	remote HMI 3 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-9103	remote HMI 4 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-9104	remote HMI 5 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-9105	remote HMI 6 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-9106	remote HMI 7 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-9107	remote HMI 8 status (set on to retry connection)	R/W	R/C	R/C	Y

LB-9148	remote HMI 49 status (set on to retry connection)	R/W	R/C	R/C	N/A
LB-9149	forced to reconnect remote HMI when IP changed on-line (set ON)	R/W	R/C	R/C	Y
LB-12754	auto. connection for remote HMI 33 (when ON)	R/W	R/C	R/C	Y
LB-12755	auto. connection for remote HMI 34 (when ON)	R/W	R/C	R/C	Y
LB-12756	auto. connection for remote HMI 35 (when ON)	R/W	R/C	R/C	Y
LB-12757	auto. connection for remote HMI 36 (when ON)	R/W	R/C	R/C	Y
LB-12758	auto. connection for remote HMI 37 (when ON)	R/W	R/C	R/C	Y
LB-12759	auto. connection for remote HMI 38 (when ON)	R/W	R/C	R/C	Y
LB-12760	auto. connection for remote HMI 39 (when ON)	R/W	R/C	R/C	Y
LB-12761	auto. connection for remote HMI 40 (when ON)	R/W	R/C	R/C	Y
LB-12762	auto. connection for remote HMI 41 (when ON)	R/W	R/C	R/C	Y
LB-12763	auto. connection for remote HMI 42 (when ON	R/W	R/C	R/C	Y
LB-12764	auto. connection for remote HMI 43 (when ON)	R/W	R/C	R/C	Y
LB-12765	auto. connection for remote HMI 44 (when ON)	R/W	R/C	R/C	Y
LB-12766	auto. connection for remote HMI 45 (when ON)	R/W	R/C	R/C	Y
LB-12767	auto. connection for remote HMI 46 (when ON)	R/W	R/C	R/C	Y
LB-12768	auto. connection for remote HMI 47 (when ON)	R/W	R/C	R/C	Y
LB-12769	auto. connection for remote HMI 48 (when ON)	R/W	R/C	R/C	Y
LB-12770	auto. connection for remote HMI 49 (when	R/W	R/C	R/C	Y

	ON)				
LB-12771	auto. connection for remote HMI 50 (when ON)	R/W	R/C	R/C	Y
LB-12772	auto. connection for remote HMI 51 (when ON)	R/W	R/C	R/C	Y
LB-12773	auto. connection for remote HMI 52 (when ON)	R/W	R/C	R/C	Y
LB-12774	auto. connection for remote HMI 53 (when ON)	R/W	R/C	R/C	Y
LB-12775	auto. connection for remote HMI 54 (when ON)	R/W	R/C	R/C	Y
LB-12776	auto. connection for remote HMI 55 (when ON)	R/W	R/C	R/C	Y
LB-12777	auto. connection for remote HMI 56 (when ON)	R/W	R/C	R/C	Y
LB-12778	auto. connection for remote HMI 57 (when ON)	R/W	R/C	R/C	Y
LB-12779	auto. connection for remote HMI 58 (when ON)	R/W	R/C	R/C	Y
LB-12780	auto. connection for remote HMI 59 (when ON)	R/W	R/C	R/C	Y
LB-12781	auto. connection for remote HMI 60 (when ON)	R/W	R/C	R/C	Y
LB-12782	auto. connection for remote HMI 61 (when ON)	R/W	R/C	R/C	Y
LB-12783	auto. connection for remote HMI 62 (when ON)	R/W	R/C	R/C	Y
LB-12784	auto. connection for remote HMI 63 (when ON)	R/W	R/C	R/C	Y
LB-12785	auto. connection for remote HMI 64 (when ON)	R/W	R/C	R/C	Y
LB-12786	remote HMI 50 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-12787	remote HMI 51 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-12788	remote HMI 52 status (set on to retry connection)	R/W	R/C	R/C	Y

LB-12789	remote HMI 53 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-12790	remote HMI 54 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-12791	remote HMI 55 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-12792	remote HMI 56 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-12793	remote HMI 57 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-12794	remote HMI 58 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-12795	remote HMI 59 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-12796	remote HMI 60 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-12797	remote HMI 61 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-12798	remote HMI 62 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-12799	remote HMI 63 status (set on to retry connection)	R/W	R/C	R/C	Y
LB-12800	remote HMI 64 status (set on to retry connection)	R/W	R/C	R/C	Y
LW-9800	(16bit) : remote HMI 1's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9801	(16bit) : remote HMI 1's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9802	(16bit) : remote HMI 1's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9803	(16bit) : remote HMI 1's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9804	(16bit) : remote HMI 1's port no.	R/W	R/C	R/C	Y
LW-9805	(16bit) : remote HMI 2's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9806	(16bit) : remote HMI 2's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y

LW-9807	(16bit) : remote HMI 2's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9808	(16bit) : remote HMI 2's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9809	(16bit) : remote HMI 2's port no.	R/W	R/C	R/C	Y
LW-9810	(16bit) : remote HMI 3's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9811	(16bit) : remote HMI 3's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9812	(16bit) : remote HMI 3's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9813	(16bit) : remote HMI 3's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9814	(16bit) : remote HMI 3's port no.	R/W	R/C	R/C	Y
LW-9815	(16bit) : remote HMI 4's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9816	(16bit) : remote HMI 4's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9817	(16bit) : remote HMI 4's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9818	(16bit) : remote HMI 4's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9819	(16bit) : remote HMI 4's port no.	R/W	R/C	R/C	Y
LW-9820	(16bit) : remote HMI 5's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9821	(16bit) : remote HMI 5's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9822	(16bit) : remote HMI 5's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9823	(16bit) : remote HMI 5's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9824	(16bit) : remote HMI 5's port no.	R/W	R/C	R/C	Y
LW-9825	(16bit) : remote HMI 6's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9826	(16bit) : remote HMI 6's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9827	(16bit) : remote HMI 6's IP2 (IP address =	R/W	R/C	R/C	Y

	IP0:IP1:IP2:IP3)				
LW-9828	(16bit) : remote HMI 6's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9829	(16bit) : remote HMI 6's port no.	R/W	R/C	R/C	Y
LW-9830	(16bit) : remote HMI 7's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9831	(16bit) : remote HMI 7's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9832	(16bit) : remote HMI 7's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9833	(16bit) : remote HMI 7's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9834	(16bit) : remote HMI 7's port no.	R/W	R/C	R/C	Y
LW-9835	(16bit) : remote HMI 8's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9836	(16bit) : remote HMI 8's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9837	(16bit) : remote HMI 8's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9838	(16bit) : remote HMI 8's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9839	(16bit) : remote HMI 8's port no.	R/W	R/C	R/C	Y
LW-9895	(16bit) : remote HMI 20's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9896	(16bit) : remote HMI 20's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9897	(16bit) : remote HMI 20's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9898	(16bit) : remote HMI 20's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9899	(16bit) : remote HMI 20's port no.	R/W	R/C	R/C	Y
LW-9905	(16bit) : remote HMI 21’s IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9906	(16bit) : remote HMI 21’s IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9907	(16bit) : remote HMI 21’s IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y

LW-9908	(16bit) : remote HMI 21's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9909	(16bit) : remote HMI 21's port no.	R/W	R/C	R/C	Y
LW-9910	(16bit) : remote HMI 22's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9911	(16bit) : remote HMI 22's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9912	(16bit) : remote HMI 22's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9913	(16bit) : remote HMI 22's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9914	(16bit) : remote HMI 22's port no.	R/W	R/C	R/C	Y
LW-9915	(16bit) : remote HMI 23's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9916	(16bit) : remote HMI 23's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9917	(16bit) : remote HMI 23's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9918	(16bit) : remote HMI 23's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9919	(16bit) : remote HMI 23's port no.	R/W	R/C	R/C	Y
LW-9920	(16bit) : remote HMI 24's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9921	(16bit) : remote HMI 24's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9922	(16bit) : remote HMI 24's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9923	(16bit) : remote HMI 24's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9924	(16bit) : remote HMI 24's port no.	R/W	R/C	R/C	Y
LW-9925	(16bit) : remote HMI 25's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9926	(16bit) : remote HMI 25's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9927	(16bit) : remote HMI 25's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9928	(16bit) : remote HMI 25's IP3 (IP address =	R/W	R/C	R/C	Y

	IP0:IP1:IP2:IP3)				
LW-9929	(16bit) : remote HMI 25's port no.	R/W	R/C	R/C	Y
LW-9930	(16bit) : remote HMI 26's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9931	(16bit) : remote HMI 26's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9932	(16bit) : remote HMI 26's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9933	(16bit) : remote HMI 26's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9934	(16bit) : remote HMI 26's port no.	R/W	R/C	R/C	Y
LW-9935	(16bit) : remote HMI 27's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9936	(16bit) : remote HMI 27's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9937	(16bit) : remote HMI 27's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9938	(16bit) : remote HMI 27's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9939	(16bit) : remote HMI 27's port no.	R/W	R/C	R/C	Y
LW-9940	(16bit) : remote HMI 28's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9941	(16bit) : remote HMI 28's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9942	(16bit) : remote HMI 28's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9943	(16bit) : remote HMI 28's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9944	(16bit) : remote HMI 28's port no.	R/W	R/C	R/C	Y
LW-9945	(16bit) : remote HMI 29's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9946	(16bit) : remote HMI 29's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9947	(16bit) : remote HMI 29's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9948	(16bit) : remote HMI 29's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y

LW-9949	(16bit) : remote HMI 29's port no.	R/W	R/C	R/C	Y
LW-9950	(16bit) : remote HMI 30's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9951	(16bit) : remote HMI 30's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9952	(16bit) : remote HMI 30's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9953	(16bit) : remote HMI 30's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9954	(16bit) : remote HMI 30's port no.	R/W	R/C	R/C	Y
LW-9955	(16bit) : remote HMI 31's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9956	(16bit) : remote HMI 31's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9957	(16bit) : remote HMI 31's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9958	(16bit) : remote HMI 31's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9959	(16bit) : remote HMI 31's port no.	R/W	R/C	R/C	Y
LW-9960	(16bit) : remote HMI 32's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9961	(16bit) : remote HMI 32's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9962	(16bit) : remote HMI 32's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9963	(16bit) : remote HMI 32's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9964	(16bit) : remote HMI 32's port no.	R/W	R/C	R/C	Y
LW-9995	(16bit) : remote HMI 39's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9996	(16bit) : remote HMI 39's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9997	(16bit) : remote HMI 39's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9998	(16bit) : remote HMI 39's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-9999	(16bit) : remote HMI 39's port no.	R/W	R/C	R/C	Y

LW-12506	(16bit) : remote HMI 40's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12507	(16bit) : remote HMI 40's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12508	(16bit) : remote HMI 40's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12509	(16bit) : remote HMI 40's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12510	(16bit) : remote HMI 40's port no.	R/W	R/C	R/C	Y
LW-12511	(16bit) : remote HMI 41's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12512	(16bit) : remote HMI 41's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12513	(16bit) : remote HMI 41's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12514	(16bit) : remote HMI 41's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12515	(16bit) : remote HMI 41's port no.	R/W	R/C	R/C	Y
LW-12516	(16bit) : remote HMI 42's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12517	(16bit) : remote HMI 42's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12518	(16bit) : remote HMI 42's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12519	(16bit) : remote HMI 42's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12520	(16bit) : remote HMI 42's port no.	R/W	R/C	R/C	Y
LW-12521	(16bit) : remote HMI 43's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12522	(16bit) : remote HMI 43's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12523	(16bit) : remote HMI 43's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12524	(16bit) : remote HMI 43's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12525	(16bit) : remote HMI 43's port no.	R/W	R/C	R/C	Y
LW-12526	(16bit) : remote HMI 44's IP0 (IP address =	R/W	R/C	R/C	Y

	IP0:IP1:IP2:IP3)				
LW-12527	(16bit) : remote HMI 44's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12528	(16bit) : remote HMI 44's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12529	(16bit) : remote HMI 44's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12530	(16bit) : remote HMI 44's port no.	R/W	R/C	R/C	Y
LW-12531	(16bit) : remote HMI 45's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12532	(16bit) : remote HMI 45's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12533	(16bit) : remote HMI 45's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12534	(16bit) : remote HMI 45's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12535	(16bit) : remote HMI 45's port no.	R/W	R/C	R/C	Y
LW-12536	(16bit) : remote HMI 46's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12537	(16bit) : remote HMI 46's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12538	(16bit) : remote HMI 46's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12539	(16bit) : remote HMI 46's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12540	(16bit) : remote HMI 46's port no.	R/W	R/C	R/C	Y
LW-12541	(16bit) : remote HMI 47's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12542	(16bit) : remote HMI 47's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12543	(16bit) : remote HMI 47's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12544	(16bit) : remote HMI 47's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12545	(16bit) : remote HMI 47's port no.	R/W	R/C	R/C	Y
LW-12546	(16bit) : remote HMI 48's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y

LW-12547	(16bit) : remote HMI 48's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12548	(16bit) : remote HMI 48's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12549	(16bit) : remote HMI 48's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12550	(16bit) : remote HMI 48's port no.	R/W	R/C	R/C	Y
LW-12551	(16bit) : remote HMI 49's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12552	(16bit) : remote HMI 49's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12553	(16bit) : remote HMI 49's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12554	(16bit) : remote HMI 49's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12555	(16bit) : remote HMI 49's port no.	R/W	R/C	R/C	Y
LW-12556	(16bit) : remote HMI 50's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12557	(16bit) : remote HMI 50's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12558	(16bit) : remote HMI 50's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12559	(16bit) : remote HMI 50's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12560	(16bit) : remote HMI 50's port no.	R/W	R/C	R/C	Y
LW-12561	(16bit) : remote HMI 51's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12562	(16bit) : remote HMI 51's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12563	(16bit) : remote HMI 51's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12564	(16bit) : remote HMI 51's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12565	(16bit) : remote HMI 51's port no.	R/W	R/C	R/C	Y
LW-12566	(16bit) : remote HMI 52's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12567	(16bit) : remote HMI 52's IP1 (IP address =	R/W	R/C	R/C	Y

	IP0:IP1:IP2:IP3)				
LW-12568	(16bit) : remote HMI 52's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12569	(16bit) : remote HMI 52's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12570	(16bit) : remote HMI 52's port no.	R/W	R/C	R/C	Y
LW-12571	(16bit) : remote HMI 53's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12572	(16bit) : remote HMI 53's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12573	(16bit) : remote HMI 53's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12574	(16bit) : remote HMI 53's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12575	(16bit) : remote HMI 53's port no.	R/W	R/C	R/C	Y
LW-12576	(16bit) : remote HMI 54's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12577	(16bit) : remote HMI 54's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12578	(16bit) : remote HMI 54's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12579	(16bit) : remote HMI 54's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12580	(16bit) : remote HMI 54's port no.	R/W	R/C	R/C	Y
LW-12581	(16bit) : remote HMI 55's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12582	(16bit) : remote HMI 55's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12583	(16bit) : remote HMI 55's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12584	(16bit) : remote HMI 55's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12585	(16bit) : remote HMI 55's port no.	R/W	R/C	R/C	Y
LW-12586	(16bit) : remote HMI 56's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12587	(16bit) : remote HMI 56's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y

LW-12588	(16bit) : remote HMI 56's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12589	(16bit) : remote HMI 56's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12590	(16bit) : remote HMI 56's port no.	R/W	R/C	R/C	Y
LW-12591	(16bit) : remote HMI 57's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12592	(16bit) : remote HMI 57's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12593	(16bit) : remote HMI 57's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12594	(16bit) : remote HMI 57's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12595	(16bit) : remote HMI 57's port no.	R/W	R/C	R/C	Y
LW-12596	(16bit) : remote HMI 58's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12597	(16bit) : remote HMI 58's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12598	(16bit) : remote HMI 58's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12599	(16bit) : remote HMI 58's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12600	(16bit) : remote HMI 58's port no.	R/W	R/C	R/C	Y
LW-12601	(16bit) : remote HMI 59's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12602	(16bit) : remote HMI 59's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12603	(16bit) : remote HMI 59's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12604	(16bit) : remote HMI 59's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12605	(16bit) : remote HMI 59's port no.	R/W	R/C	R/C	Y
LW-12606	(16bit) : remote HMI 60's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12607	(16bit) : remote HMI 60's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12608	(16bit) : remote HMI 60's IP2 (IP address =	R/W	R/C	R/C	Y

	IP0:IP1:IP2:IP3)				
LW-12609	(16bit) : remote HMI 60's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12610	(16bit) : remote HMI 60's port no.	R/W	R/C	R/C	Y
LW-12611	(16bit) : remote HMI 61's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12612	(16bit) : remote HMI 61's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12613	(16bit) : remote HMI 61's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12614	(16bit) : remote HMI 61's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12615	(16bit) : remote HMI 61's port no.	R/W	R/C	R/C	Y
LW-12616	(16bit) : remote HMI 62's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12617	(16bit) : remote HMI 62's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12618	(16bit) : remote HMI 62's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12619	(16bit) : remote HMI 62's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12620	(16bit) : remote HMI 62's port no.	R/W	R/C	R/C	Y
LW-12621	(16bit) : remote HMI 63's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12622	(16bit) : remote HMI 63's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12623	(16bit) : remote HMI 63's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12624	(16bit) : remote HMI 63's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12625	(16bit) : remote HMI 63's port no.	R/W	R/C	R/C	Y
LW-12626	(16bit) : remote HMI 64's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12627	(16bit) : remote HMI 64's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12628	(16bit) : remote HMI 64's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y

LW-12629	(16bit) : remote HMI 64's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-12630	(16bit) : remote HMI 64's port no.	R/W	R/C	R/C	Y

22.3.19. Communication Status and Control with Remote Device

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-10050	(16bit) : IP0 of the HMI connecting to remote device 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10051	(16bit) : IP1 of the HMI connecting to remote device 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10052	(16bit) : IP2 of the HMI connecting to remote device 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10053	(16bit) : IP3 of the HMI connecting to remote device 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10054	(16bit) : port no. of the HMI connecting to remote device 1	R/W	R/C	R/C	N
LW-10055	(16bit) : IP0 of the HMI connecting to remote device 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10056	(16bit) : IP1 of the HMI connecting to remote device 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10057	(16bit) : IP2 of the HMI connecting to remote device 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10058	(16bit) : IP3 of the HMI connecting to remote device 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10059	(16bit) : port no. of the HMI connecting to remote device 2	R/W	R/C	R/C	N
LW-10060	(16bit) : IP0 of the HMI connecting to remote device 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10061	(16bit) : IP1 of the HMI connecting to remote device 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10062	(16bit) : IP2 of the HMI connecting to remote device 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10063	(16bit) : IP3 of the HMI connecting to remote device 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10064	(16bit) : port no. of the HMI connecting to	R/W	R/C	R/C	N

	remote device 3				
LW-10065	(16bit) : IP0 of the HMI connecting to remote device 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10066	(16bit) : IP1 of the HMI connecting to remote device 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10067	(16bit) : IP2 of the HMI connecting to remote device 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10068	(16bit) : IP3 of the HMI connecting to remote device 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10069	(16bit) : port no. of the HMI connecting to remote device 4	R/W	R/C	R/C	N
LW-10205	(16bit) : IP0 of the HMI connecting to remote device 32 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10206	(16bit) : IP1 of the HMI connecting to remote device 32 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10207	(16bit) : IP2 of the HMI connecting to remote device 32 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10208	(16bit) : IP3 of the HMI connecting to remote device 32 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10209	(16bit) : port no. of the HMI connecting to remote device 32	R/W	R/C	R/C	N
LW-10300	(16bit) : remote device 1's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10301	(16bit) : remote device 1's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10302	(16bit) : remote device 1's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10303	(16bit) : remote device 1's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10304	(16bit) : remote device 1's port no.	R/W	R/C	R/C	N
LW-10305	(16bit) : remote device 2's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10306	(16bit) : remote device 2's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10307	(16bit) : remote device 2's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10308	(16bit) : remote device 2's IP3 (IP address =	R/W	R/C	R/C	N

	IP0:IP1:IP2:IP3)				
LW-10309	(16bit) : remote device 2's port no.	R/W	R/C	R/C	N
LW-10310	(16bit) : remote device 3's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10311	(16bit) : remote device 3's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10312	(16bit) : remote device 3's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10313	(16bit) : remote device 3's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10314	(16bit) : remote device 3's port no.	R/W	R/C	R/C	N
LW-10315	(16bit) : remote device 4's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10316	(16bit) : remote device 4's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10317	(16bit) : remote device 4's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10318	(16bit) : remote device 4's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10319	(16bit) : remote device 4's port no.	R/W	R/C	R/C	N
LW-10455	(16bit) : remote device 32's IP0 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10456	(16bit) : remote device 32's IP1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10457	(16bit) : remote device 32's IP2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10458	(16bit) : remote device 32's IP3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-10459	(16bit) : remote device 32's port no.	R/W	R/C	R/C	N

22.3.20. Local/Remote Operation Restrictions

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9044	disable remote control (when ON)	R/W	R/C	R/C	Y
LB-9053	prohibit password remote-read operation (when ON)	R/W	R/C	R/C	Y

LB-9054	prohibit password remote-write operation (when ON)	R/W	R/C	R/C	Y
LB-9196	local HMI supports monitor function only (when ON)	R/W	R/C	R/C	Y
LB-9197	support monitor function only for remote HMIs (when ON)	R/W	R/C	R/C	Y
LB-9198	disable local HMI to trigger a MACRO (when ON)	R/W	R/C	R/C	Y
LB-9199	disable remote HMI to trigger a MACRO (when ON)	R/W	R/C	R/C	Y

22.3.21. Communication Error Codes

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-9400	(16bit) : error code for device 1	R	R	R	Y
LW-9401	(16bit) : error code for device 2	R	R	R	Y
LW-9402	(16bit) : error code for device 3	R	R	R	Y
LW-9403	(16bit) : error code for device 4	R	R	R	Y
LW-9404	(16bit) : error code for device 5	R	R	R	Y
LW-9405	(16bit) : error code for device 6	R	R	R	Y
LW-9406	(16bit) : error code for device 7	R	R	R	Y
LW-9407	(16bit) : error code for device 8	R	R	R	Y
LW-9449	(16bit) : error code for device 50	R	R	R	Y
LW-9490	(16bit) : error code for USB device	R	R	R	Y
LW-9491	(16bit) : error code for CAN-Bus device	R	R	R	Y

Note

1. A list of the explanation of device communication error codes:

Error Code	Cause of Communication Error
0	Normal
1	The device is busy and not yet ready to process a command.
2	Communication error due to unexpected reason.
3	The device does not exist.
4	The device using the specified station number does not exist.
5	Incorrect address format.
6	Read/Write unsupported address.
7	The driver of the device does not exist.
8	The COM port does not exist.
9	Incorrect IP address or unable to connect the device.
10	Checksum error.
11	Unidentified command.
12	Ignore
20	The USB device is improperly connected.
21	The CAN Bus device is improperly connected.
22	No reply from the device.
23	Insufficient data read from the device before timeout.
24	The Conversion Tag used by the object does not exist or the content is incorrect.
25	HMI is not accepting any commands from a remote HMI.
251	Read/Write exceeding number of words from/to the register of the MODBUS device.
252	MODBUS device replies incorrect data format.
253	MODBUS device checksum error.

22.3.22. Driver ID

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-9300	(16bit) : driver ID of local device 1	R	R	R	Y
LW-9301	(16bit) : driver ID of local device 2	R	R	R	Y
LW-9302	(16bit) : driver ID of local device 3	R	R	R	Y
LW-9303	(16bit) : driver ID of local device 4	R	R	R	Y
LW-9331	(16bit) : driver ID of local device 32	R	R	R	Y

22.3.23. DLT645 Controller

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-10700	(4 words) : DLT_645 operator (COM 1)	R/W	R/C	R/C	Y
LW-10704	(4 words) : DLT_645 password (COM 1)	R/W	R/C	R/C	Y
LW-10708	(6 words) : DLT_645 address (COM 1)	R/W	R/C	R/C	Y
LW-10715	(4 words) : DLT_645 operator (COM 2)	R/W	R/C	R/C	Y
LW-10719	(4 words) : DLT_645 password (COM 2)	R/W	R/C	R/C	Y
LW-10723	(6 words) : DLT_645 address (COM 2)	R/W	R/C	R/C	Y
LW-10730	(4 words) : DLT_645 operator (COM 3)	R/W	R/C	R/C	Y
LW-10734	(4 words) : DLT_645 password (COM 3)	R/W	R/C	R/C	Y
LW-10738	(6 words) : DLT_645 address (COM 3)	R/W	R/C	R/C	Y

22.3.24. [Device No Response] Window Control

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9192	disable USB device's "Device No Response" dialog (when ON)	R/W	R/C	R/C	Y
LB-11960	disable device 1's "Device No Response" dialog (when ON)	R/W	R/C	R/C	Y
LB-11961	disable device 2's "Device No Response" dialog (when ON)	R/W	R/C	R/C	Y
LB-11962	disable device 3's "Device No Response" dialog (when ON)	R/W	R/C	R/C	Y
LB-11963	disable device 4's "Device No Response" dialog (when ON)	R/W	R/C	R/C	Y
LB-11964	disable device 5's "Device No Response" dialog (when ON)	R/W	R/C	R/C	Y
LB-11965	disable device 6's "Device No Response"	R/W	R/C	R/C	Y

	dialog (when ON)				
LB-11966	disable device 7's "Device No Response" dialog (when ON)	R/W	R/C	R/C	Y
LB-11967	disable device 8's "Device No Response" dialog (when ON)	R/W	R/C	R/C	Y
LB-12023	disable device 64's "Device No Response" dialog (when ON)	R/W	R/C	R/C	Y
LB-12082	disable CAN Bus device's "Device No Response" dialog (when ON)	R/W	R/C	R/C	Y

22.3.25. [Fast Selection] Window Control

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9013	hide (set ON)/show (set OFF) FS window	R/W	R/C	R/C	N
LB-9014	hide (set ON)/show (set OFF) FS button	R/W	R/C	R/C	N
LB-9015	hide (set ON)/show (set OFF) FS window/button	R/W	R/C	R/C	N

22.3.26. EasyAccess

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9051	disconnect (set OFF)/connect (set ON) EasyAccess server	R/W	R/C	R/C	N
LB-9052	status of connecting to EasyAccess server	R	R	R	N

For more information about EasyAccess, please visit http://www.ihmi.net/.

Click the icon to download the demo project. Please confirm your internet connection.

22.3.27. EasyAccess 2.0

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-10820	(16bit) : disable (set 0)/enable (set 1) (EasyAccess 2.0)	R/W	R/C	R/C	Y
LW-10821	(5 words) : session ID (EasyAccess 2.0)	R	R	R	Y
LW-10826	(2 words) : password (EasyAccess 2.0)	R	R	R	Y
LW-10828	(16bit) : execution status (EasyAccess 2.0)	R	R	R	Y
LW-10829	(16bit) : the last error code (EasyAccess	R	R	R	Y

	2.0)				
LW-11170	(16bit) : Proxy Disable/Enable (0:disable, 1:enable) (EasyAccess 2.0)	R/W	R/C	R/C	Y
LW-11171	(16bit) : Proxy Type (0:HTTP, 1:SOCKSv4, 2:SOCKSv5) (EasyAccess 2.0)	R/W	R/C	R/C	Y
LW-11172	(16bit) : Proxy Server IP0 (EasyAccess 2.0)	R/W	R/C	R/C	Y
LW-11173	(16bit) : Proxy Server IP1 (EasyAccess 2.0)	R/W	R/C	R/C	Y
LW-11174	(16bit) : Proxy Server IP2 (EasyAccess 2.0)	R/W	R/C	R/C	Y
LW-11175	(16bit) : Proxy Server IP3 (EasyAccess 2.0)	R/W	R/C	R/C	Y
LW-11176	(16bit) : Proxy Server Port (EasyAccess 2.0)	R/W	R/C	R/C	Y
LW-11177	(16bit) : Proxy authentication (0:disable, 1:enable) (EasyAccess 2.0)	R/W	R/C	R/C	Y
LW-11178	(16 words) : Proxy username (EasyAccess 2.0)	R/W	R/C	R/C	Y
LW-11194	(16 words) : Proxy password (EasyAccess 2.0)	R/W	R/C	R/C	Y
LW-11210	(20 words) : Hardware key (EasyAccess 2.0)	R	R	R	Y
LW-11296	(16bit) : Location of EasyAccess 2.0 server (0 : Global, 1 : China)	R	R	R	Y
LW-11770	(64 words) : QR code (URL) for WeChat push notification (EasyAccess 2.0)	R	R	R	Y
LW-11982	(16 words) : HMI domain (EasyAccess 2.0)	R	R	R	Y
LW-12773	(16 words) : Number of connected user (EasyAccess 2.0)	R	R	R	Y
LW-12774	(16 words) : Connected user 1 (EasyAccess 2.0)	R	R	R	Y
LW-12790	(16 words) : Connected user 2 (EasyAccess 2.0)	R	R	R	Y
LW-12806	(16 words) : Connected user 3 (EasyAccess 2.0)	R	R	R	Y

22.3.28. Remote Print/Backup Server

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-10069	forced to reconnect remote printer/backup server when IP changed on-line (set ON)	R/W	R/C	R/C	N
LB-12040	remote printer/backup server disconnection alarm (when ON)	R	R	R	N
LW-9770	(16bit) : remote printer/backup server IP0 (IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N

LW-9771	(16bit) : remote printer/backup server IP1 (IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-9772	(16bit) : remote printer/backup server IP2 (IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-9773	(16bit) : remote printer/backup server IP3 (IP0:IP1:IP2:IP3)	R/W	R/C	R/C	N
LW-9774	(6 words) : remote printer/backup server user name *Note 1	R/W	R/C	R/C	N
LW-9780	(6 words) : remote printer/backup server password *Note 1	R/W	R/C	R/C	N

Note

1. When change settings using LW-9774 and LW-9780, please reboot HMI for the new settings to take effect.

Click the icon to download the demo project. Please confirm your internet connection.

22.3.29. Pass-Through Settings

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-9901	(16bit) : pass-through source COM port (1~3 : COM 1~COM 3)	R/W	R/C	R/C	Y
LW-9902	(16bit) : pass-through destination COM port (1~3 : COM 1~COM 3)	R/W	R/C	R/C	Y
LW-9903	(16bit) : pass-through control (0 : normal, 1 : pause, 2 : stop communications between HMI and device when executing pass-through)	R/W	R/C	R/C	Y
LW-9904	(16bit) : pass-through server port no. (2000~2100)	R/W	R/C	R/C	Y
LW-10850	(16bit) : disable/enable (0 : disable, 1 : normal, 2 : IP limited) (siemens pass-through)	R/W	R/C	R/C	Y
LW-10851	(16bit) : destination COM port (siemens pass-through)	R/W	R/C	R/C	Y

LW-10852	(16bit) : destination PLC station no. (siemens pass-through)	R/W	R/C	R/C	Y
LW-10853	(16bit) : communication protocol (0 : invalid, 1 : PPI, 2 : MPI) (siemens pass-through)	R/W	R/C	R/C	Y
LW-10854	(16bit) : IP0 of connecting client (IP address = IP0:IP1:IP2:IP3) (siemens pass-through)	R/W	R/C	R/C	Y
LW-10855	(16bit) : IP1 of connecting client (IP address = IP0:IP1:IP2:IP3) (siemens pass-through)	R/W	R/C	R/C	Y
LW-10856	(16bit) : IP2 of connecting client (IP address = IP0:IP1:IP2:IP3) (siemens pass-through)	R/W	R/C	R/C	Y
LW-10857	(16bit) : IP3 of connecting client (IP address = IP0:IP1:IP2:IP3) (siemens pass-through)	R/W	R/C	R/C	Y
LW-10858	(16bit) : IP0 of designated client (IP address = IP0:IP1:IP2:IP3) (siemens pass-through)	R/W	R/C	R/C	Y
LW-10859	(16bit) : IP1 of designated client (IP address = IP0:IP1:IP2:IP3) (siemens pass-through)	R/W	R/C	R/C	Y
LW-10860	(16bit) : IP2 of designated client (IP address = IP0:IP1:IP2:IP3) (siemens pass-through)	R/W	R/C	R/C	Y
LW-10861	(16bit) : IP3 of designated client (IP address = IP0:IP1:IP2:IP3) (siemens pass-through)	R/W	R/C	R/C	Y
LW-10862	(16bit) : connection status (0 : ready, 1 : client connecting) (siemens pass-through)	R	R	R	Y
LW-10863	(16bit) : execution status (0 : normal, 1 : error) (siemens pass-through)	R	R	R	Y
LW-10864	(16bit) : the last error (siemens pass-through)	R	R	R	Y

For more information about Siemens pass-through feature, see “29 Pass-through”.

Click the icon to download the demo project. Please confirm your internet connection.

22.3.30. VNC Control

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-12088	enable VNC monitor mode (when ON) *Note 1	R/W	R/C	R/C	Y
LB-12089	VNC pass word free (when ON) *Note 1	R/W	R/C	R/C	Y
LB-12090	a VNC client connecting to HMI (when ON)(OS version 20120621 or later supports only)	R	R	R	Y
LB-12091	disable auto-logout function when a VNC client connecting to HMI (when ON)(OS version 20120621 or later supports only)	R/W	R/C	R/C	Y
LB-12092	enable VNC (set ON), disable VNC (set OFF)	R/W	R/C	R/C	Y
LB-12093	VNC connection mode (OFF: single connection, ON: multi connection) (OS version 2013.05.09 or later support)*Note1	R/W	R/C	R/C	Y
LW-9530	(4 words) : VNC server password	R/W	R/C	R/C	Y

Note

1. On eMT, iE, XE, mTV, iP models, to change VNC mode, use LB-12092 to stop and then restart VNC to update the setting.

22.3.31. Project Key and HMI Key

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9046	project key is different from HMI key (when ON)	R	R	R	Y
LW-9046	(32bit) : HMI key *Note 1	R/W	R/C	R	Y

Note

1. When change HMI Key using LW-9046, please reboot HMI for the new settings to take effect.

Click the icon to download the demo project. Please confirm your internet connection.

22.3.32. USB Security Key

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-11160	(16bit) : start time of USB security key - year	R	R	R	Y
LW-11161	(16bit) : start time of USB security key - month	R	R	R	Y
LW-11162	(16bit) : start time of USB security key - day	R	R	R	Y
LW-11163	(16bit) : start time of USB security key - hour	R	R	R	Y
LW-11164	(16bit) : start time of USB security key - minute	R	R	R	Y
LW-11165	(16bit) : expiration time of USB security key - year	R	R	R	Y
LW-11166	(16bit) : expiration time of USB security key - month	R	R	R	Y
LW-11167	(16bit) : expiration time of USB security key - day	R	R	R	Y
LW-11168	(16bit) : expiration time of USB security key - hour	R	R	R	Y
LW-11169	(16bit) : expiration time of USB security key - minute	R	R	R	Y

22.3.33. User Name and Password

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9050	user logout *Note 2	W	C	C	N
LB-9060	password error *Note 2	R	R	R	N
LB-9061	update password (set ON) *Note 2	W	C	C	N
LB-12056	the user touches an unauthorized object (when ON)	R	R	R	Y
PLB-12056	the user touches an unauthorized object (when ON) (on tablet)	R	N/A	N/A	Y
LW-9082	(16bit) : auto logout time (unit : minute, 0 : disable the function)	R/W	R/C	R/C	Y
LW-9219	(16bit) : user no. (1~12) *Note 2	R/W	R/C	R/C	N
LW-9220	(32bit) : password *Note 2	R/W	R/C	R/C	N
LW-9222	(16bit) : object classes can be operated for	R	R	R	Y

	current user (bit 0:A, bit 1:B,bit 2:C, ...)				
PLW-9222	(16bit) : object classes (on tablet) can be operated for current user (bit 0:A, bit 1:B,bit 2:C, ...)	R	N/A	N/A	Y
LW-9500	(32bit) : user 1's password *Note 2	R/W	R/C	R/C	N
LW-9502	(32bit) : user 2's password *Note 2	R/W	R/C	R/C	N
LW-9504	(32bit) : user 3's password *Note 2	R/W	R/C	R/C	N
LW-9506	(32bit) : user 4's password *Note 2	R/W	R/C	R/C	N
LW-9508	(32bit) : user 5's password *Note 2	R/W	R/C	R/C	N
LW-9510	(32bit) : user 6's password *Note 2	R/W	R/C	R/C	N
LW-9512	(32bit) : user 7's password *Note 2	R/W	R/C	R/C	N
LW-9514	(32bit) : user 8's password *Note 2	R/W	R/C	R/C	N
LW-9516	(32bit) : user 9's password *Note 2	R/W	R/C	R/C	N
LW-9518	(32bit) : user 10's password *Note 2	R/W	R/C	R/C	N
LW-9520	(32bit) : user 11's password *Note 2	R/W	R/C	R/C	N
LW-9522	(32bit) : user 12's password *Note 2	R/W	R/C	R/C	N
LW-10754	(8 words) : current user name *Note 1	R	R	R	Y
PLW-10754	(8 words) : current user name (on tablet) *Note 1	R	N/A	N/A	Y

Note

1. Only for [Security] » [Enhanced security mode].
2. Only for [Security] » [General mode].

Click the icon to download the demo project. Please confirm your internet connection.

22.3.34. Macro

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9059	disable macro TRACE function (when ON) *Note1	R/W	R/C	R/C	Y
LW-10900	(16bit) : macro 0 status (0:ready, 3:executing, 5:waiting response, 9:waiting sync, 17:delay, 32:abnormal end (exceed array size))	R	R	R	Y
LW-10901	(16bit) : macro 1 status (0:ready, 3:executing, 5:waiting response, 9:waiting sync, 17:delay, 32:abnormal end (exceed	R	R	R	Y

	array size))				
LW-10902	(16bit) : macro 2 status (0:ready, 3:executing, 5:waiting response, 9:waiting sync, 17:delay, 32:abnormal end (exceed array size))	R	R	R	Y
LW-10903	(16bit) : macro 3 status (0:ready, 3:executing, 5:waiting response, 9:waiting sync, 17:delay, 32:abnormal end (exceed array size))	R	R	R	Y
LW-10904	(16bit) : macro 4 status (0:ready, 3:executing, 5:waiting response, 9:waiting sync, 17:delay, 32:abnormal end (exceed array size))	R	R	R	Y
LW-10905	(16bit) : macro 5 status (0:ready, 3:executing, 5:waiting response, 9:waiting sync, 17:delay, 32:abnormal end (exceed array size))	R	R	R	Y
LW-10906	(16bit) : macro 6 status (0:ready, 3:executing, 5:waiting response, 9:waiting sync, 17:delay, 32:abnormal end (exceed array size))	R	R	R	Y
LW-10907	(16bit) : macro 7 status (0:ready, 3:executing, 5:waiting response, 9:waiting sync, 17:delay, 32:abnormal end (exceed array size))	R	R	R	Y
LW-10908	(16bit) : macro 8 status (0:ready, 3:executing, 5:waiting response, 9:waiting sync, 17:delay, 32:abnormal end (exceed array size))	R	R	R	Y
LW-10909	(16bit) : macro 9 status (0:ready, 3:executing, 5:waiting response, 9:waiting sync, 17:delay, 32:abnormal end (exceed array size))	R	R	R	Y
LW-11154	(16bit) : macro 254 status (0:ready, 3:executing, 5:waiting response, 9:waiting sync, 17:delay, 32:abnormal end (exceed array size))	R	R	R	Y
LW-12247	(16bit): macro 255 status (0: ready, 3: executing, 5: waiting response, 9: waiting sync, 17: delay, 32: abnormal end (exceed array size))	R	R	R	Y
LW-12248	(16bit): macro 256 status (0: ready, 3: executing, 5: waiting response, 9: waiting sync, 17: delay, 32: abnormal end (exceed array size))	R	R	R	Y

LW-12249	(16bit): macro 257 status (0: ready, 3: executing, 5: waiting response, 9: waiting sync, 17: delay, 32: abnormal end (exceed array size))	R	R	R	Y
LW-12250	(16bit): macro 258 status (0: ready, 3: executing, 5: waiting response, 9: waiting sync, 17: delay, 32: abnormal end (exceed array size))	R	R	R	Y
LW-12251	(16bit): macro 259 status (0: ready, 3: executing, 5: waiting response, 9: waiting sync, 17: delay, 32: abnormal end (exceed array size))	R	R	R	Y
LW-12252	(16bit): macro 260 status (0: ready, 3: executing, 5: waiting response, 9: waiting sync, 17: delay, 32: abnormal end (exceed array size))	R	R	R	Y
LW-12253	(16bit): macro 261 status (0: ready, 3: executing, 5: waiting response, 9: waiting sync, 17: delay, 32: abnormal end (exceed array size))	R	R	R	Y
LW-12254	(16bit): macro 262 status (0: ready, 3: executing, 5: waiting response, 9: waiting sync, 17: delay, 32: abnormal end (exceed array size))	R	R	R	Y
LW-12255	(16bit): macro 263 status (0: ready, 3: executing, 5: waiting response, 9: waiting sync, 17: delay, 32: abnormal end (exceed array size))	R	R	R	Y
LW-12256	(16bit): macro 264 status (0: ready, 3: executing, 5: waiting response, 9: waiting sync, 17: delay, 32: abnormal end (exceed array size))	R	R	R	Y
LW-12491	(16bit): macro 499 status (0: ready, 3: executing, 5: waiting response, 9: waiting sync, 17: delay, 32: abnormal end (exceed array size))	R	R	R	Y

Note

1. LB-9059: Disable macro trace function.

Click the icon to download the demo project. Please confirm your internet connection.

22.3.35. Input Object Function

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-9002	(32bit-float) : input high limit	R	R	R	Y
LW-9004	(32bit-float) : input low limit	R	R	R	Y
LW-9052	(32bit-float) : the previous input value of the numeric input object	R	R	R	Y
PLW-9052	(32bit-float) : the previous input value of the numeric input object	R	N/A	N/A	Y
LW-9150	(32 words) : keyboard's input data (ASCII)	R	R	R	Y
LW-9540	(16bit) : reserved for caps lock	R/W	R/C	R/C	Y

22.3.36. Time Sync./Daylight Saving Time

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-12055	failed to execute time synchronization (when ON)	R	R	R	Y
LB-12355	daylight saving time period (when ON)	R	R	R	Y
LW-11260	(16bit) : enable/disable the daylight saving time (DST) (0:disable, 1:enable)	R/W	R/C	R/C	Y
LW-11261	(16bit) : hour of the DST bias	R/W	R/C	R/C	Y
LW-11262	(16bit) : minute of the DST bias	R/W	R/C	R/C	Y
LW-11263	(16bit) : month of the year when DST starts	R/W	R/C	R/C	Y
LW-11264	(16bit) : week of the month when DST starts (1~5)	R/W	R/C	R/C	Y
LW-11265	(16bit) : day of the week when DST starts (0~6)	R/W	R/C	R/C	Y
LW-11266	(16bit) : hour of local time when DST starts	R/W	R/C	R/C	Y
LW-11267	(16bit) : minute of local time when DST starts	R/W	R/C	R/C	Y
LW-11268	(16bit) : month of the year when DST ends	R/W	R/C	R/C	Y
LW-11269	(16bit) : week of the month when DST ends (1~5)	R/W	R/C	R/C	Y
LW-11270	(16bit) : day of the week when DST ends (0~6)	R/W	R/C	R/C	Y
LW-11271	(16bit) : hour of local time when DST ends	R/W	R/C	R/C	Y
LW-11272	(16bit) : minute of local time when DST ends	R/W	R/C	R/C	Y

LW-11273	(16bit) : enable/disable time synchronization via NTP (Network Time Protocol) server (0:disable, 1:enable)	R/W	R/C	R/C	Y
LW-11274	(16bit) : execute time synchronization when HMI starts (0:disable, 1:enable)	R/W	R/C	R/C	Y
LW-11275	(16bit) : server response time has been adjusted in accordance with DST (0:disable, 1:enable)	R/W	R/C	R/C	Y
LW-11276	(16bit) : HMI time zone (unit : minute)	R/W	R/C	R/C	Y
LW-11277	(16bit) : server response time (server time zone) (unit : minute)	R/W	R/C	R/C	Y
LW-11278	(16bit) : IP 0 of network time server 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11279	(16bit) : IP 1 of network time server 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11280	(16bit) : IP 2 of network time server 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11281	(16bit) : IP 3 of network time server 1 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11282	(16bit) : IP 0 of network time server 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11283	(16bit) : IP 1 of network time server 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11284	(16bit) : IP 2 of network time server 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11285	(16bit) : IP 3 of network time server 2 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11286	(16bit) : IP 0 of network time server 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11287	(16bit) : IP 1 of network time server 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11288	(16bit) : IP 2 of network time server 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11289	(16bit) : IP 3 of network time server 3 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11290	(16bit) : IP 0 of network time server 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y

LW-11291	(16bit) : IP 1 of network time server 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11292	(16bit) : IP 2 of network time server 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11293	(16bit) : IP 3 of network time server 4 (IP address = IP0:IP1:IP2:IP3)	R/W	R/C	R/C	Y
LW-11294	(32bit) : update interval (time synchronization interval) (10 ~ 86400, unit : second)	R/W	R/C	R/C	Y

22.3.37. Cellular Data Network

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-11297	(16 words) : PIN code of SIM card (cellular network)	R/W	R/C	R/C	Y
LW-11313	(16 words) : Access Point Name (cellular network)	R/W	R/C	R/C	Y
LW-11329	(16 words) : username (cellular network)	R/W	R/C	R/C	Y
LW-11345	(16 words) : password (cellular network)	R/W	R/C	R/C	Y
LW-11361	(16 words) : dial number (cellular network)	R/W	R/C	R/C	Y
LW-11377	(16bit) : stop (set 0)/start (set 1) connection (cellular network)	R/W	R/C	R/C	Y
LW-11378	(16bit) : last error code (0:success, 1:incorrect PIN code, 2:no SIM, 3:no device, 4:puk locked, 5:other) (cellular network)	R	R	R	Y
LW-11379	(16bit) : connection status (0:no device, 1:disconnect, 2:connecting, 3:connected) (cellular network)	R	R	R	Y
LW-11380	(16bit) : stop (set 0)/start (set 1) connection (USB tethering)	R/W	R/C	R/C	Y
LW-11381	(16bit) : connection status (0:no device, 1:disconnect, 2:connected, 3:fail, 4:OS not support, 5:HMI not support) (USB tethering)	R	R	R	Y
LW-12631	(16bit) : signal strength (dBm) (0: failed, others: signal strength) (cellular network)	R	R	R	Y
LW-12632	(8 words) : IMEI (cellular network)	R	R	R	Y
LW-12640	(10 words) : ICCID code of SIM card (cellular	R	R	R	Y

	network)				

22.3.38. WiFi Setting

System registers listed in this section are available exclusively for HMI models that support WiFi only, e.g. MT8103iE, cMT3103, cMT-SVR-200…etc.

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-12365	update wifi setting (IP, netmask, gateway, DNS) (set ON)	R/W	R/C	R/C	Y
LB-12666	disable(set OFF)/enable(set ON) WiFi Hotspot	R/W	R/C	R/C	Y
LB-12667	refresh WiFi Hotspot setting (SSID, password) (set ON)	R/W	R/C	R/C	Y
LB-12820	M02 status (0: unplugged, 1: plugged in)	R	R	R	Y
LW-11383	(16bit) : WiFi control (1: disconnect, 2: connect, 3: popup setting dialog, 4: connect by SSID)	R/W	R/C	R/C	Y
LW-11384	(16bit) : WiFi error code (0: no error, 1: no such device, 2. radio is off, 3: parameter is invalid)	R	R	R	Y
LW-11385	(16bit) : WiFi status (0: stopped; 1: connecting; 2: connected)	R	R	R	Y
LW-11386	(16 words) : WiFi connected SSID	R	R	R	Y
LW-11402	(16bit) : WiFi signal level (0: none, 1: weak, 2: fair, 3: good, 4: excellent) *Note2	R	R	R	Y
LW-11403	(16bit) : WiFi country code *Note1	R/W	R/C	R/C	Y
LW-11404	(16bit) : WiFi radio (0: Off, 1: On)	R/W	R/C	R/C	Y
LW-11405	(16bit) : WiFi signal strength (dBm) (0, 1, 2: failed, others: signal strength)	R	R	R	Y
LW-11410	(16bit) : HMI WiFi IP 0 (for HMI use only)	R/W	R/C	R/C	Y
LW-11411	(16bit) : HMI WiFi IP 1 (for HMI use only)	R/W	R/C	R/C	Y
LW-11412	(16bit) : HMI WiFi IP 2 (for HMI use only)	R/W	R/C	R/C	Y
LW-11413	(16bit) : HMI WiFi IP 3 (for HMI use only)	R/W	R/C	R/C	Y
LW-11414	(16bit) : HMI WiFi netmask 0 (for HMI use only)	R/W	R/C	R/C	Y
LW-11415	(16bit) : HMI WiFi netmask 1 (for HMI use only)	R/W	R/C	R/C	Y

LW-11416	(16bit) : HMI WiFi netmask 2 (for HMI use only)	R/W	R/C	R/C	Y
LW-11417	(16bit) : HMI WiFi netmask 3 (for HMI use only)	R/W	R/C	R/C	Y
LW-11418	(16bit) : HMI WiFi gateway 0 (for HMI use only)	R/W	R/C	R/C	Y
LW-11419	(16bit) : HMI WiFi gateway 1 (for HMI use only)	R/W	R/C	R/C	Y
LW-11420	(16bit) : HMI WiFi gateway 2 (for HMI use only)	R/W	R/C	R/C	Y
LW-11421	(16bit) : HMI WiFi gateway 3 (for HMI use only)	R/W	R/C	R/C	Y
LW-11422	(16bit) : HMI WiFi media access control (MAC) address 0	R	R	R	Y
LW-11423	(16bit) : HMI WiFi media access control (MAC) address 1	R	R	R	Y
LW-11424	(16bit) : HMI WiFi media access control (MAC) address 2	R	R	R	Y
LW-11425	(16bit) : HMI WiFi media access control (MAC) address 3	R	R	R	Y
LW-11426	(16bit) : HMI WiFi media access control (MAC) address 4	R	R	R	Y
LW-11427	(16bit) : HMI WiFi media access control (MAC) address 5	R	R	R	Y
LW-11428	(16bit) : HMI WiFi domain name system (DNS) server IP 0	R/W	R/C	R/C	Y
LW-11429	(16bit) : HMI WiFi domain name system (DNS) server IP 1	R/W	R/C	R/C	Y
LW-11430	(16bit) : HMI WiFi domain name system (DNS) server IP 2	R/W	R/C	R/C	Y
LW-11431	(16bit) : HMI WiFi domain name system (DNS) server IP 3	R/W	R/C	R/C	Y
LW-11432	(16bit) : obtain an WiFi IP address automatically (DHCP => 0 : off, 1 : on)	R/W	R/C	R/C	Y
LW-12006	(16 words) : WiFi Hotspot SSID (enable to apply)	R/W	R/C	R/C	Y
LW-12022	(32 words) : WiFi Hotspot password (enable	R/W	R/C	R/C	Y

	to apply)				
LW-12703	(16 words : WiFi SSID to connect	R/W	R/C	R/C	Y
LW-12719	(32 words) : WiFi password to use	R/W	R/C	R/C	Y
LW-12751	(16bit) : WiFi security mode to use (0: WPA/WPA2, 1: WEP, 2: NONE)	R/W	R/C	R/C	Y
LW-12768	(16bit) : WiFi Hotspot security mode (0: NONE, 1: WPA-MIX-PSK)	R/W	R/C	R/C	Y
LW-12769	(16bit) : WiFi Hotspot server address IP0	R	R	R	Y
LW-12770	(16bit) : WiFi Hotspot server address IP1	R	R	R	Y
LW-12771	(16bit) : WiFi Hotspot server address IP2	R/W	R/C	R/C	Y
LW-12772	(16bit) : WiFi Hotspot server address IP3	R	R	R	Y

Note

1. Please enter WiFi country code in ASCII for uppercase letters, the setting will take effect after rebooting HMI. Wireless regulations vary from country to country. The country code selection affects the list of channels of the wireless radio.
2. The signal strength is classified into four levels: 1. Weak (<-70 dBm), 2. Fair (-60 ~ -70 dBm), 3. Good (-50 ~ -60 dBm), 4. Excellent (>-50 dBm)

22.3.39. OPC UA Server

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-11435	(16bit) : OPC UA Server status (0: Stopped, 1: Started)	R	R	R	Y
LW-11436	(16bit) : OPC UA Server error code (0: Success, 1 or more: Error) *Note 1	R	R	R	Y
LW-11437	(16bit) : OPC UA Server control command (0: None, 1: Start, 2: Stop)	R/W	R/C	R/C	Y
LB-12753	clear OPC UA HAD data	R/W	R/C	R/C	Y

Note

1. Error codes of failed steps include:

Code	Cause of Error
0	No error
1	Unknown error
2	Invalid certificate

22.3.40. e-Mail

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-12053	failed to send an [Event Log] e-Mail (when ON)	R	R	R	Y
LB-12054	failed to send an [Backup Object] e-Mail (when ON)	R	R	R	Y
LW-9216	(16bit) : the result of importing email data *Note 1	R	R	R	Y
LW-11444	(16bit) : failed step (e-Mail) *Note 2	R	R	R	Y
LW-11445	(16bit) : error code (e-Mail) *Note 3	R	R	R	Y

Note

1. 1: import succeeded, 2: import failed (file doesn't exist) , 4: incorrect file format or file cannot be decrypted.
2. Error codes of failed steps include:

Code	Cause of Error
0	CSMTP_NO_ERROR
100	WSA_STARTUP = Unable to initialize winsock2
101	WSA_VER = Wrong version of the winsock2
102	WSA_SEND = Function send() failed
103	WSA_RECV = Function recv() failed
104	WSA_CONNECT = Function connect failed
105	WSA_GETHOSTBY_NAME_ADDR = Unable to determine remote server
106	WSA_INVALID_SOCKET = Invalid winsock2 socket
107	WSA_HOSTNAME = Function hostname() failed
108	WSA_IOCTLSOCKET = Function ioctlsocket() failed
109	WSA_SELECT
110	BAD_IPV4_ADDR = Improper IPv4 address
200	UNDEF_MSG_HEADER = Undefined message header
201	UNDEF_MAIL_FROM = Undefined mail sender
202	UNDEF_SUBJECT = Undefined message subject
203	UNDEF_RECIPIENTS = Undefined at least one recipient
204	UNDEF_RECIPIENT_MAIL = Undefined recipient mail
205	UNDEF_LOGIN = Undefined user login
206	UNDEF_PASSWORD = Undefined user password
207	BAD_LOGIN_PASSWORD = Invalid user login or password

208	BAD_DIGEST_RESPONSE = Server returned a bad digest MD5 response
209	BAD_SERVER_NAME = Unable to determine server name for digest MD5 response
300	COMMAND_MAIL_FROM = Server returned error after sending MAIL FROM
301	COMMAND_EHLO = Server returned error after sending EHLO
302	COMMAND_AUTH_PLAIN = Server returned error after sending AUTH PLAIN
303	COMMAND_AUTH_LOGIN = Server returned error after sending AUTH LOGIN
304	COMMAND_AUTH_CRAMMD5 = Server returned error after sending AUTH CRAM-MD5
305	COMMAND_AUTH_DIGESTMD5 = Server returned error after sending AUTH DIGEST-MD5
306	COMMAND_DIGESTMD5 = Server returned error after sending MD5 DIGEST
307	COMMAND_DATA = Server returned error after sending DATA
308	COMMAND_QUIT = Server returned error after sending QUIT
309	COMMAND_RCPT_TO = Server returned error after sending RCPT TO
310	MSG_BODY_ERROR = Error in message body
400	CONNECTION_CLOSED = Server has closed the connection
401	SERVER_NOT_READY = Server is not ready
402	SERVER_NOT_RESPONDING = Server not responding
403	SELECT_TIMEOUT =
404	FILE_NOT_EXIST = File not exist
405	MSG_TOO_BIG = Message is too big
406	BAD_LOGIN_PASS = Bad login or password
407	UNDEF_XYZ_RESPONSE = Undefined xyz SMTP response
408	LACK_OF_MEMORY = Lack of memory
409	TIME_ERROR = time() error
410	RECVBUF_IS_EMPTY = RecvBuf is empty
411	SENDBUF_IS_EMPTY = SendBuf is empty
412	OUT_OF_MSG_RANGE = Specified line number is out of message size
413	COMMAND_EHLO_STARTTLS = Server returned error after sending STARTTLS
414	SSL_PROBLEM = SSL problem
415	COMMAND_DATABLOCK = Failed to send data block
416	STARTTLS_NOT_SUPPORTED = The STARTTLS command is not supported by the server
417	LOGIN_NOT_SUPPORTED = AUTH LOGIN is not supported by the server

3. Error messages sent from mail server can be shown by designating a word address (length adjustable) in System Parameter Settings » e-Mail tab » [Error message].

22.3.41. cMT Viewer and Diagnoser

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	

LB-12656	enable Diagnoser (set ON), disable (set OFF)	R/W	R/C	R/C	Y
LB-12657	Diagnoser password free (when ON)	R/W	R/C	R/C	Y
LW-11756	(4 words) : Diagnoser password	R/W	R/C	R/C	Y
LW-11839	(16bit) : number of connected clients (cMT Viewer)	R	R	R	Y
LW-11840	(8 words) : client 1 user name	R	R	R	Y
LW-11848	(8 words) : client 2 user name	R	R	R	Y
LW-11856	(8 words) : client 3 user name	R	R	R	Y
LW-11864	(8 words) : client 4 user name	R	R	R	Y
LW-11872	(8 words) : client 5 user name	R	R	R	Y
LW-11880	(8 words) : client 6 user name	R	R	R	Y
LW-11888	(8 words) : client 7 user name	R	R	R	Y
LW-11896	(8 words) : client 8 user name	R	R	R	Y
LW-11904	(8 words) : client 9 user name	R	R	R	Y
LW-11912	(8 words) : client 10 user name	R	R	R	Y
LW-11940	(16bit) : client 1 IP0 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11941	(16bit) : client 1 IP1 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11942	(16bit) : client 1 IP2 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11943	(16bit) : client 1 IP3 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11944	(16bit) : client 2 IP0 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11945	(16bit) : client 2 IP1 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11946	(16bit) : client 2 IP2 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11947	(16bit) : client 2 IP3 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11948	(16bit) : client 3 IP0 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11949	(16bit) : client 3 IP1 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11950	(16bit) : client 3 IP2 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y

LW-11951	(16bit) : client 3 IP3 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11952	(16bit) : client 4 IP0 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11953	(16bit) : client 4 IP1 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11954	(16bit) : client 4 IP2 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11955	(16bit) : client 4 IP3 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11956	(16bit) : client 5 IP0 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11957	(16bit) : client 5 IP1 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11958	(16bit) : client 5 IP2 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11959	(16bit) : client 5 IP3 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11960	(16bit) : client 6 IP0 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11961	(16bit) : client 6 IP1 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11962	(16bit) : client 6 IP2 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11963	(16bit) : client 6 IP3 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11964	(16bit) : client 7 IP0 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11965	(16bit) : client 7 IP1 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11966	(16bit) : client 7 IP2 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11967	(16bit) : client 7 IP3 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11968	(16bit) : client 8 IP0 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11969	(16bit) : client 8 IP1 (IP address =	R	R	R	Y

	IP0:IP1:IP2:IP3)				
LW-11970	(16bit) : client 8 IP2 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11971	(16bit) : client 8 IP3 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11972	(16bit) : client 9 IP0 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11973	(16bit) : client 9 IP1 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11974	(16bit) : client 9 IP2 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11975	(16bit) : client 9 IP3 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11976	(16bit) : client 10 IP0 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11977	(16bit) : client 10 IP1 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11978	(16bit) : client 10 IP2 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
LW-11979	(16bit) : client 10 IP3 (IP address = IP0:IP1:IP2:IP3)	R	R	R	Y
PLW-11998	(8 words) : Control Token release countdown	R	N/A	N/A	Y

22.3.42. CODESYS Information

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LW-11760	(16bit) : CODESYS firmware status (0:error, 1:start, 2:stop)	R	R	R	Y
LW-11761	(16bit) : CODESYS application status (0:error, 1:start, 2:stop)	R	R	R	Y
LW-11762	(16bit) : CODESYS login status (0:error, 1:login, 2:logout)	R	R	R	Y
LW-12059	(16bit) : CODESYS firmware version (year)	R	R	R	Y
LW-12060	(16bit) : CODESYS firmware version (month)	R	R	R	Y
LW-12061	(16bit) : CODESYS firmware version (day)	R	R	R	Y

22.3.43. Miscellaneous

Address	Description	Read(R)/Write(W)/Control(C)			Supported on cMT
		Local HMI	Macro	Remote HMI	
LB-9000~ LB-9009	initialized as ON	R/W	R/C	R/C	Y
LB-9010	data-transfer write indicator	R	R	R	Y
LB-9011	data-transfer read indicator	R	R	R	Y
LB-9012	data-transfer execution indicator	R	R	R	Y
LB-9016	status is on when a client connects to this HMI	R	R	R	Y
LB-9017	disable write-back in PLC control's [change window]	R/W	R/C	R/C	Y
LB-9039	status of file backup activity (backup in process if ON)	R	R	R	Y
LB-9045	memory-map communication fails (when ON)	R	R	R	Y
LB-9049	disable/enable watch dog (use LW-11456 set watch dog timeout) *Note 1	R/W	R/C	R/C	Y
LB-12356	enable(set on)/disable(set off) web streaming	R/W	R/C	R/C	Y
LB-12357	web streaming status (on: enabled / off: disabled)	R	R	R	Y
LB-12358	enable (when ON) / disable (when OFF) off-line simulation on HMI *Note 5	R/W	R/C	R/C	Y
LB-12361	status of operation log function (OFF : disabled, ON : enabled)	R	R	R	Y
LB-12664	Cache Hit (cMT-G03)	R	R	R	Y
LB-12668	SECS license status (0: not activated, 1: activated)	R	R	R	Y
LB-12819	RUN/STOP switch status (0: stop, 1: run) *Note 6	R	R	R	Y
LB-12822	disable(set OFF)/enable(set ON) ethernet bridge mode	R/W	R/C	R/C	Y
LW-9006	(16bit) : connected client no.	R	R	R	Y
LW-9024	(16bit) : memory link system register	R/W	R/C	R/C	Y
LW-9032	(8 words) : folder name of backup history files to SD, USB memory *Note 3	R/W	R/C	R/C	Y
LW-9050	(16bit) : current base window ID	R	R	R	Y
PLW-9050	(16bit) : current base window ID (on tablet)	R	N/A	N/A	Y
LW-9134	(16bit) : language mode *Note 2	R/W	R/C	R/C	Y
PLW-9134	(16bit) : language mode (on tablet) *Note 2	R/W	N/A	N/A	Y
LW-9900	(16bit) : HMI run mode (0 : normal mode,	R/W	R/C	R/C	Y

	1-3 : test mode (COM 1-COM 3)				
LW-10762	(8 words) : slot1 user name	R/W	R/C	R/C	Y
LW-10770	(8 words) : slot2 user name	R/W	R/C	R/C	Y
LW-10778	(8 words) : slot3 user name	R/W	R/C	R/C	Y
LW-10814	(16bit) : connecting to a Weintek HMI (0:none, 1:connecting) *Note 4	R	R	R	Y
LW-11456	(16bit) : watch dog timeout (3 ~ 10), unit : second *Note 1	R/W	R/C	R/C	Y
LW-12245	(32bit) : BACnet/IP server device id	R/W	R/C	R/C	Y

Note

1. When LB-9049 watch dog function is enabled, watch dog automatically reboots the system after the HMI stops functioning for a specified period of time.
2. To display texts on objects in multiple languages, except for using Label Library, the system reserved register [LW-9134: language mode] is needed. The value range in LW-9134 is 0 ~ 23. The values in LW-9134 relates to the languages downloaded to HMI. LW-9134 value and language correspondence vary according to the languages selected during project compilation and download.
   For example: If 5 languages are defined by user in Label Library as Language 1 (Traditional Chinese), Language 2 (Simplified Chinese), Language 3 (English), Language 4 (French), and Language 5 (Japanese). If only Language 1, 3, 5 are downloaded, the corresponding language of the value in LW-9134 will be 0 → Language 1 (Traditional Chinese), 1 → Language 3 (English), 2 → Language 5 (Japanese). The following demo project explains how to switch languages using Option List Object and LW-9134.

Click the icon to download the demo project. Please confirm your internet connection.

3. Click the icon to download the demo project. Please confirm your internet connection.
4. When the USB Host is connected to a Weintek HMI, the address is set to 1, for testing if the USB Client of another Weintek HMI works normally.
5. This register allows switching to off-line mode on HMI. In off-line mode, “Device No Response” message will not show even when HMI is not properly connected. In this case, device related objects can still operate, however, the values are not read by / written to the device.
6. Supported on cMT-CTRL01.

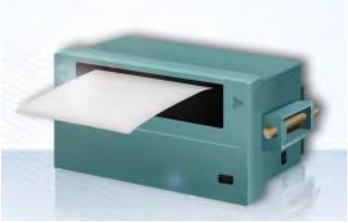
23. HMI Supported Printers

This chapter describes the printers supported by HMI and the setup steps.

23.1. The Supported Printer Types 23-2
23.2. Steps to Add a New Printer and Start Printing 23-5
23.3. Installing Network Printer on cMT / cMT X HMI 23-7
23.4. Installing PPD File on cMT / cMT X HMI 23-10

23.1. The Supported Printer Types

HMI supported printer drivers include the following types:

Printer type	Description
● **SP-M, D, E, F** 	Serial printers, please configure communication parameters to match the printer. [Pixels of width] must be correctly set and can't exceed printer default setting: 100 pixels for 1610 series printers. 220 pixels for 2407, 4004 series printers. The driver uses EPSON ESC Protocol for Serial Micro Printer.
● **EPSON ESC/P2 Series**	Serial printers, please configure communication parameters to match the printer. The ESPON ESC/P2 printer protocol is used. Impact Printer: LQ-300, LQ-300+, LQ-300K+ (RS-232), LQ-300+II (RS-232) Inkjet Printer: Stylus Photo 750 Laser Printer: EPL-5800
● **HP PCL Series (USB)**	HP compatible USB printers that support HP PCL5 protocol or PostScript3 Printer Control Language. The printers that support PCL5 or later versions will support PCL5 protocol due to the downward compatibility of PCL.

Axiohm A630	Micro printer from France connects via serial port; please configure communication parameters to match the printer.
SPRT	Serial printers, please configure communication parameters to match the printer. [Pixels of width] must be correctly set and can't exceed printer default setting "100".
EPSON TM-L90	Serial printers, please configure communication parameters to match the printer. [Pixels of width] must be correctly set and can't exceed printer default setting "576".
EPSON TM-T70	Serial printers, please configure communication parameters to match the printer. [Pixels of width] must be correctly set and can't exceed printer default setting "576". The paper cutting mode can be selected: [No cut] / [Partial cut].
BRIGHTEK WH-A19	Supported models: A92R10-00E72A 72 in model number represents hexadecimal printer, and A represents wide voltage 5~9V. This is the same as the A6 16 impact printer.

- **BRIGHTEK WH-E19**

Serial printers, please configure the same communication parameters as the printer.

- **BRIGHTEK WH-E22**

Supported models:
E22R10-00E725: Same as A7 16 impact printer.
A7 represents A72R90-31E72A.
E221R90-00E11740GA: Serial printer, connects through RS-485 port, please use a RS232-to-RS485 converter.

- **BRIGHTEK WH-C1/C2**

Serial printers, please configure communication parameters to match the printer. The paper cutting mode can be selected: [No cut] / [Half cut] / Full cut].

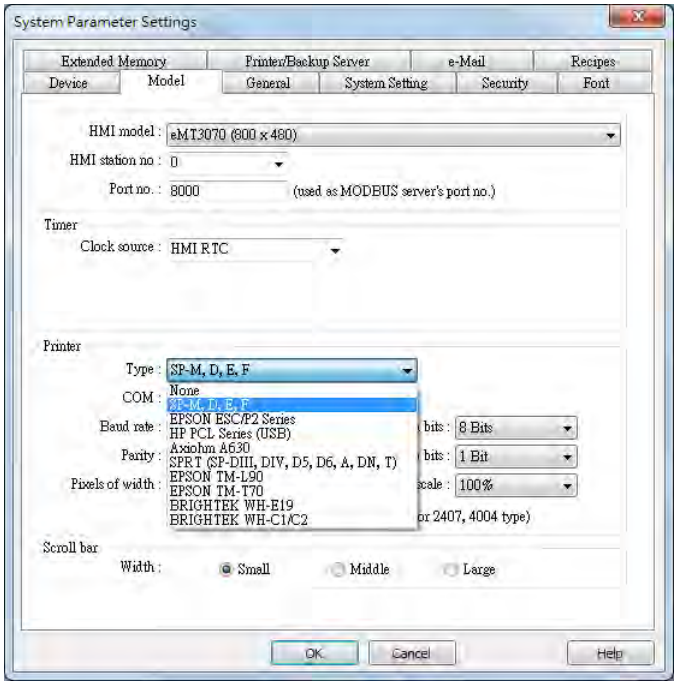
- **Remote Printer Server**

Use EasyPrinter to start printing by the printers connected with PC via Ethernet. This works under MS Windows so most printers on the market are supported.

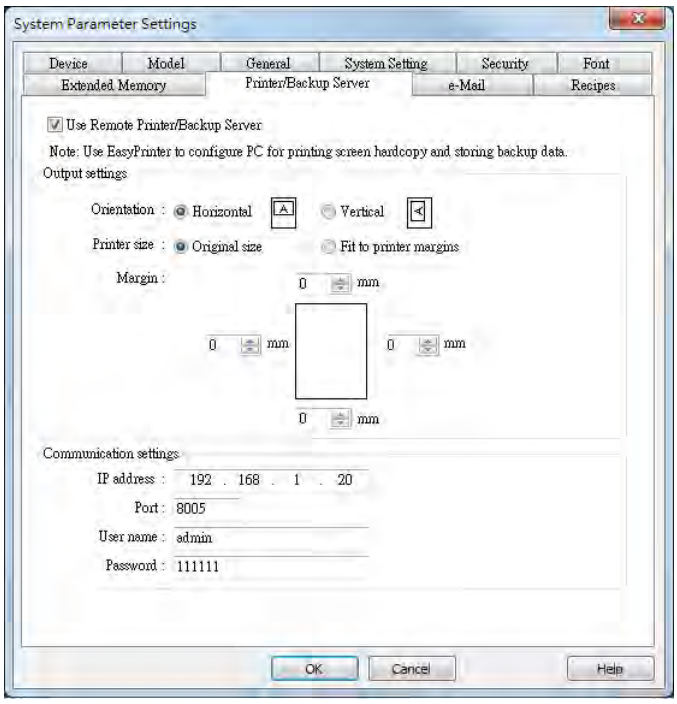
23.2. Steps to Add a New Printer and Start Printing

1. Add printer type.

- In [System Parameter Settings] » [Model] select the printer type and set the relevant parameters.

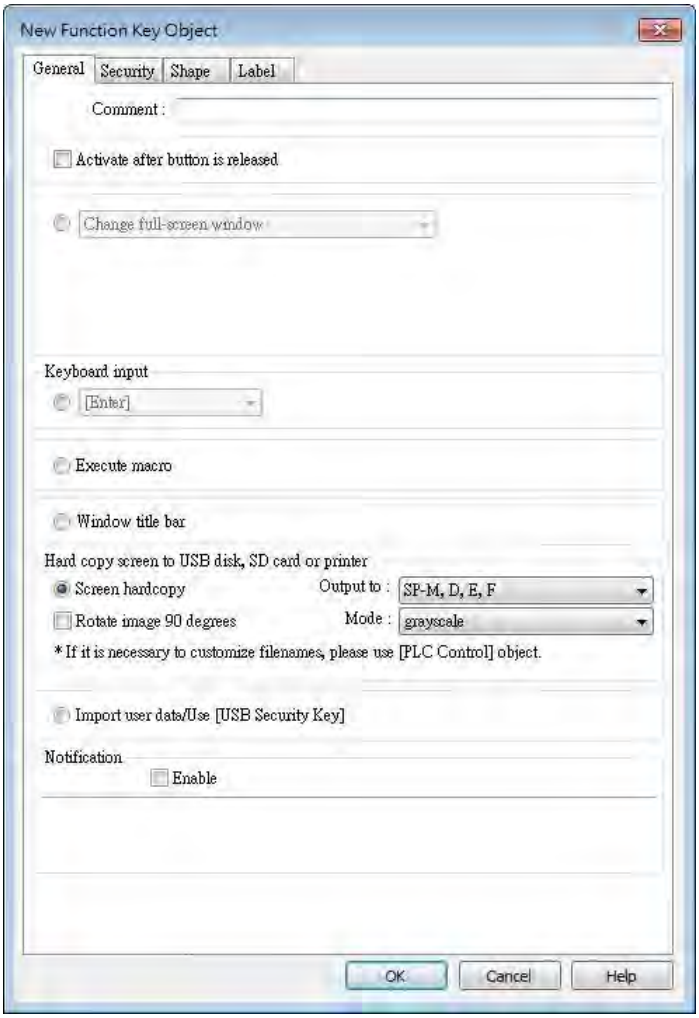


- To connect Remote Printer Server, set the parameters in [System Parameter Settings] » [Printer/Backup Server].

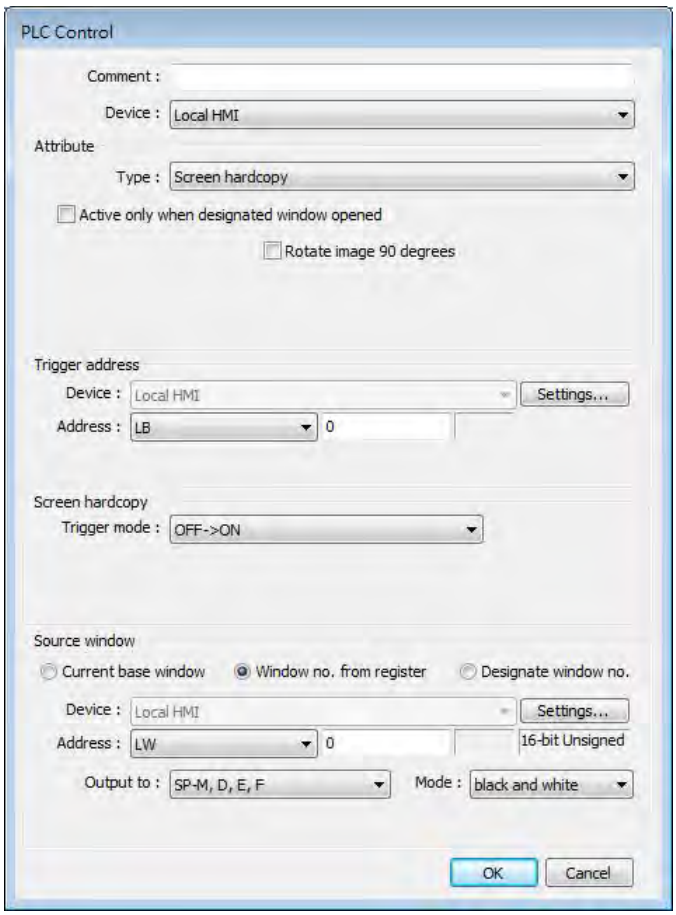


2. Start printing.

- Start printing with Function Key.



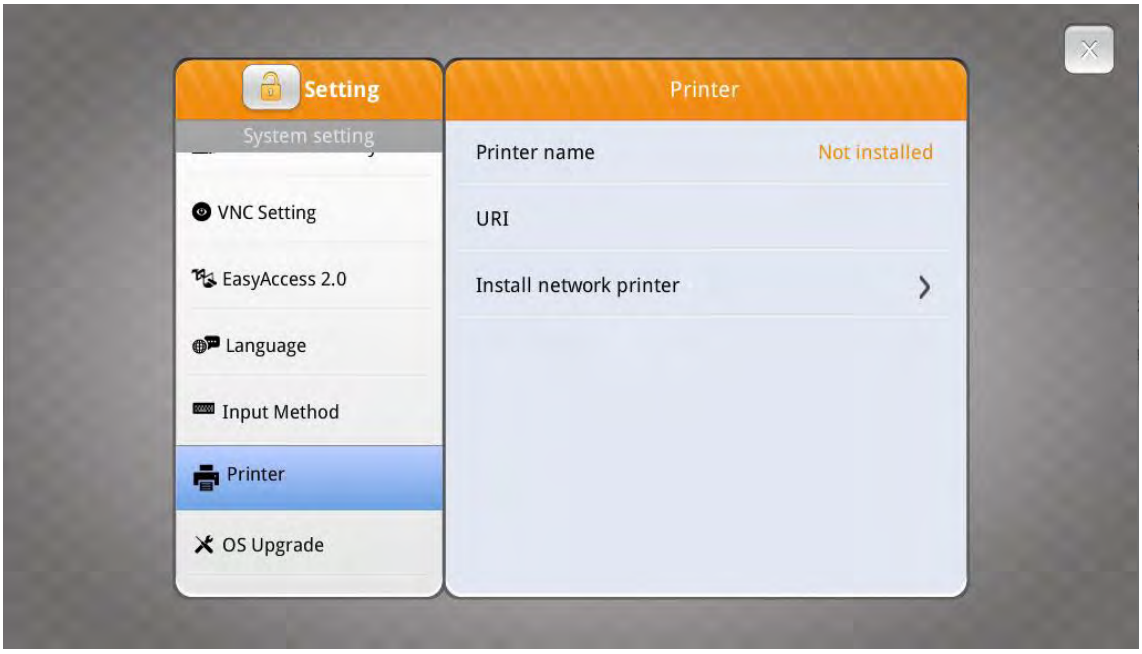
- Or, use PLC Control [Screen hardcopy] to start printing with a designated bit address.



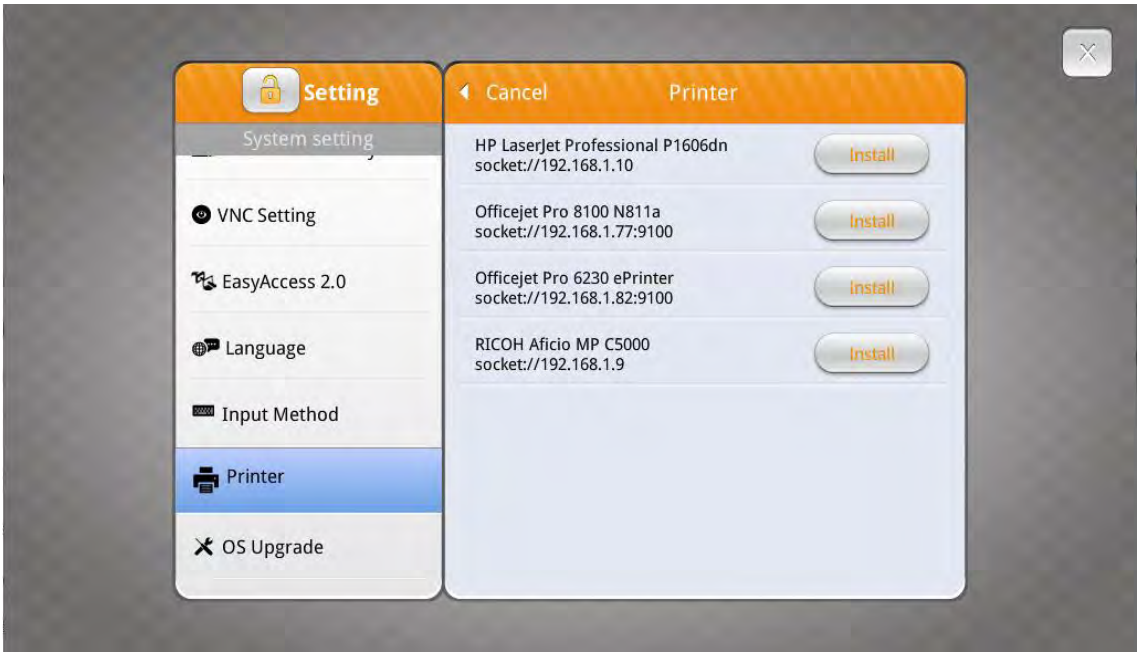
23.3. Installing Network Printer on cMT / cMT X HMI

Opening system setting on a cMT / cMT X Series model and select [Install network printer] can install more printers apart from the ones listed in the preceding chapters.

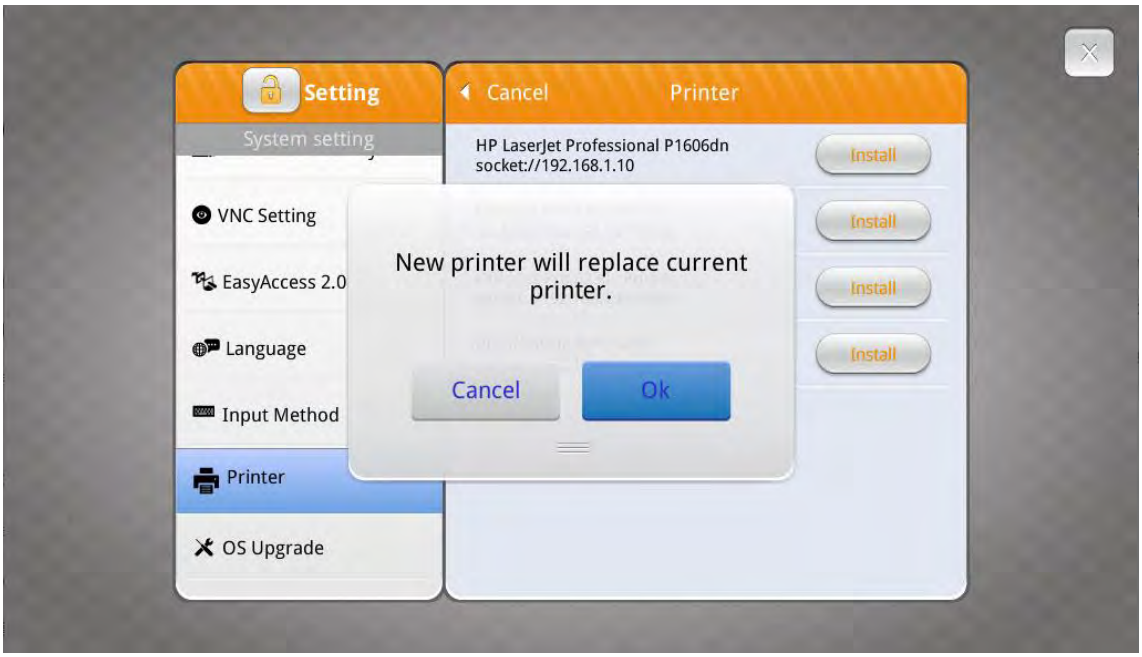
1. Tap to open system setting and log in, find [Printer] » [Install network printer].



2. HMI needs a few seconds to search for the network printers on the same network.



3. Select a network printer and tap [Install] to install its driver. The new printer driver will replace the former one since only one printer can be installed at a time.

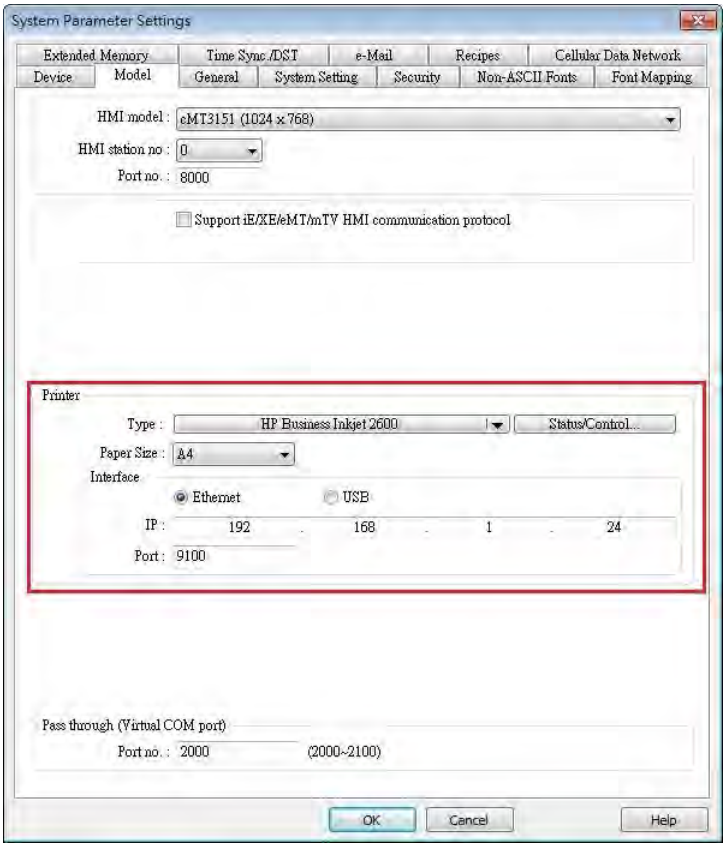


4. The network printer is ready for use after installation is completed.

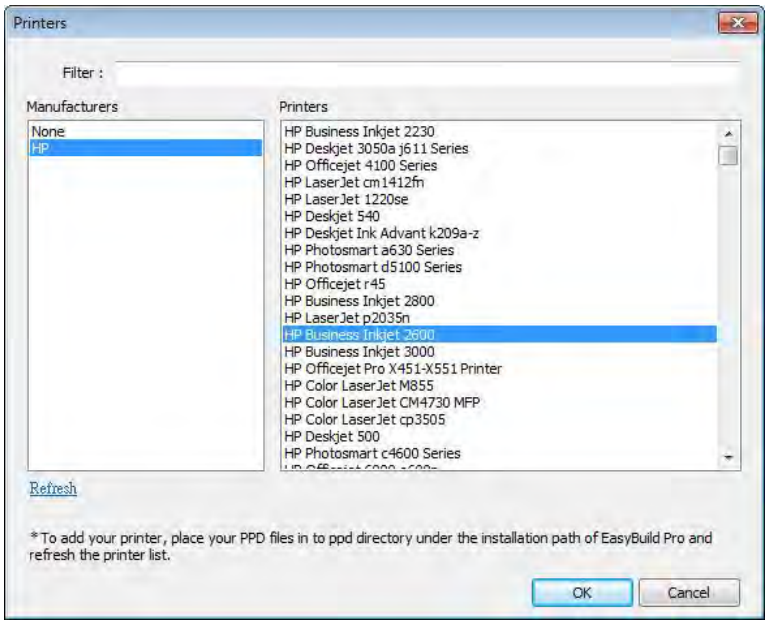
23.4. Installing PPD File on cMT / cMT X HMI

Installing the PPD file from a printer driver on a cMT / cMT X Series model is also possible.

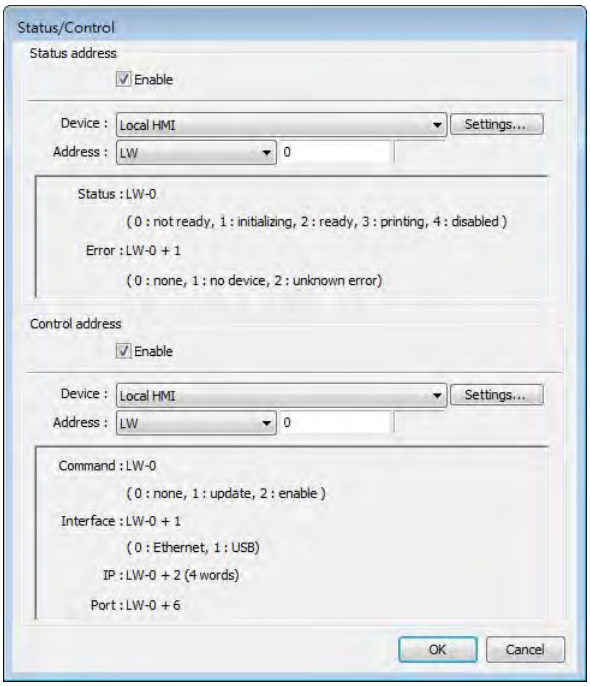
1. Go to [System Parameter Settings] » [Model] tab.



2. Select a printer type. To add a new printer, place the printer's PPD file in the ppd folder under EasyBuilder Pro installation directory, and then click [Refresh].

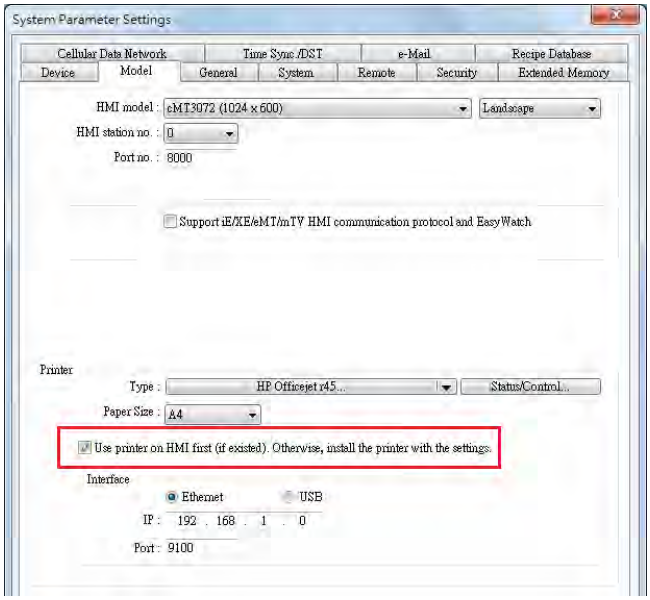


3. Use Status Address to monitor printer status, and use Control Address to dynamically update connection parameters.



Note

- Installing PPD file is not supported on cMT-SVR and cMT Gateway models.
- HP printers are recommended since they may have better compatibility with cMT / cMT X models. Printers should be tested beforehand.
- Please install a printer on a cMT / cMT X Series model by using [Install network printer] in the system setting first; if this doesn't work, installing PPD file is the second option.
- To use the printer selected in the project file, please do not select [Use printer on HMI first (if existed)] option if a printer driver is already installed on HMI. By giving command 1: update in the control address, the printer selected in the project file will be used.


System Parameter Settings
Cellular Data Network
Time Sync /DST
e-Mail
Recipe Database
Device
Model
General
System
Remote
Security
Extended Memory
HMI model :
eMT3072 (1024 x 600)
Landscape
HMI station no. :
0
Port no. :
8000
Support iE/XE/eMT/mTV HMI communication protocol and EasyWatch
Printer
Type :
HP Officejet r45...
Status/Control...
Paper Size :
A4
Use printer on HMI first (if existed). Otherwise, install the printer with the settings.
Interface
Ethernet
USB
IP :
192 . 168 . 1 . 0
Port :
9100

24. Recipe Editor

This chapter explains how to use Recipe Editor.

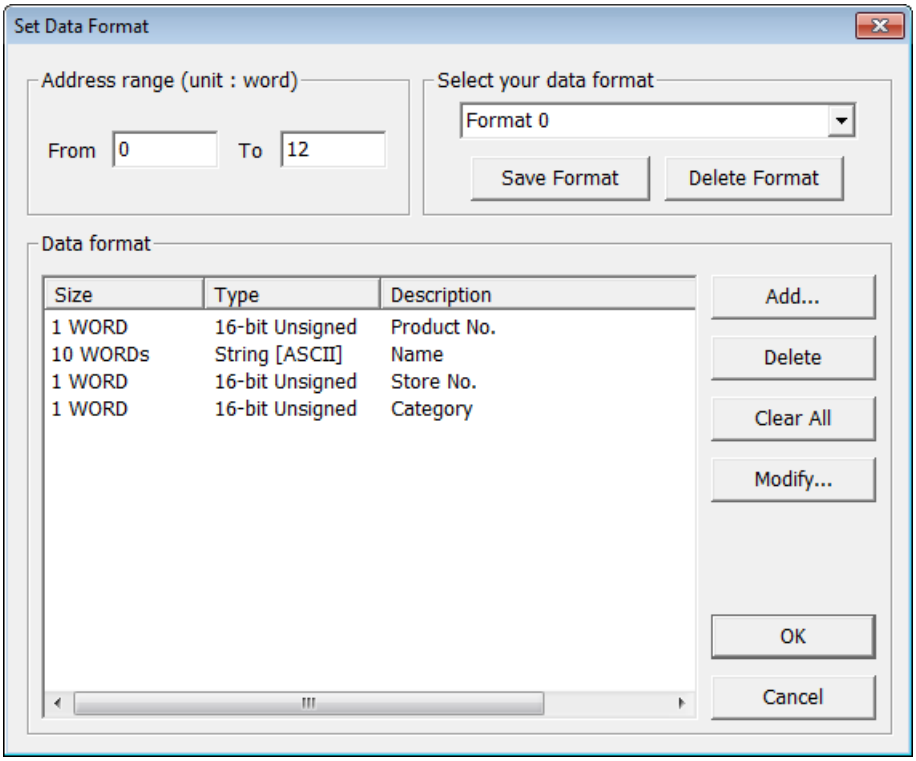
24.1. Overview 24-2
24.2. Recipe / Extended Memory Editor Setting 24-2
24.3. Recipe Database Settings 24-4

24.1. Overview

Recipe Editor is used to create, view, and edit recipe data.
EasyBuilder Pro also provides another tool for editing recipe: Recipe Records. To use this tool; first define a recipe in EasyBuilder Pro [Data/History] » [Recipe Database] tab, and then use [Recipe View Object] to display the content. The following introduces the usage of these two editing tools.

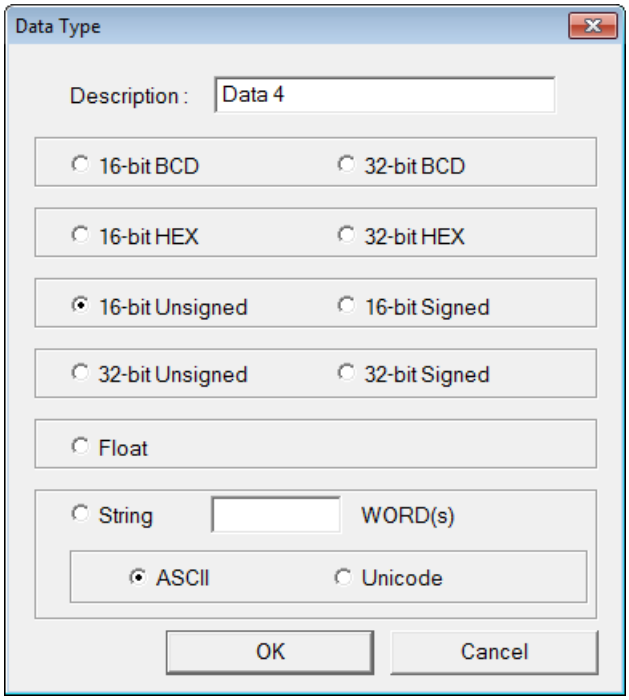
24.2. Recipe / Extended Memory Editor Setting

1. Open Utility Manager and click [Recipe/Extended Memory Editor].
2. To add new .rcp or .emi files, click [File] » [New].
3. Set address range and select data format.

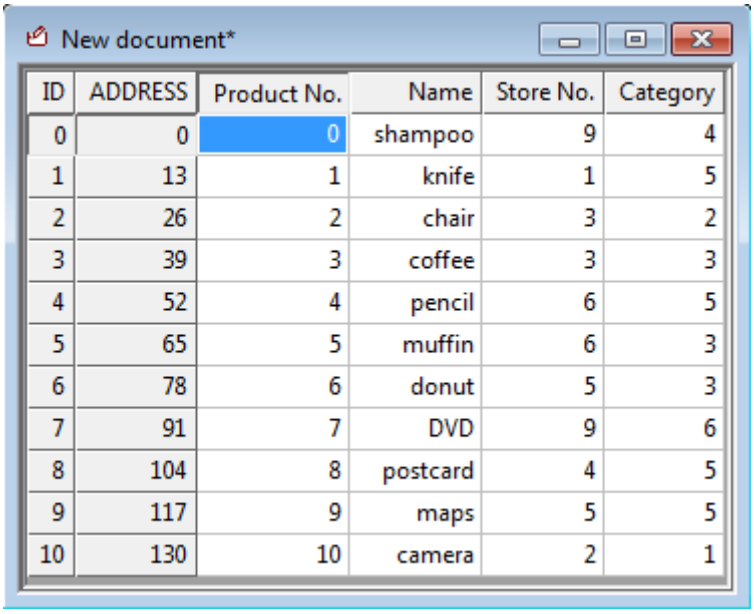


Setting	Description
Address range	Fill in address range, the unit of which is word.
Select your data format	Save the specified data format for loading next time. The saved file name is “dataEX.fmt” under EasyBuilder Pro’s installation directory.
Data format	Edit new data format in this field.

4. Click [Add] to enter a description of the data type, and select data format. When selecting [String], please enter the length (words) and select [ASCII] or [Unicode].



5. After setup, click [OK] to start editing recipe data.



ID	ADDRESS	Product No.	Name	Store No.	Category
0	0	0	shampoo	9	4
1	13	1	knife	1	5
2	26	2	chair	3	2
3	39	3	coffee	3	3
4	52	4	pencil	6	5
5	65	5	muffin	6	3
6	78	6	donut	5	3
7	91	7	DVD	9	6
8	104	8	postcard	4	5
9	117	9	maps	5	5
10	130	10	camera	2	1

In this example, the total length of data format is 13 words. Each 13 words will be one set of recipe data.

The first set: “product no.” = address 0, “Name” = address 1 ~ 10, “Store No.”= address 11, “Category” = address 12;

The second set: “product no.” = address 13, “Name” = address 14 ~ 23, “Store No.”= address 24, “Category” = address 25;…and so on.

Note

- After editing recipe data, it can be saved as .rcp, .emi, or .csv files. The .rcp files can be downloaded to HMI using Utility Manager or external devices (USB drive or SD card). The .emi files can be saved directly to the external device which is inserted to HMI as extended memory (EM).

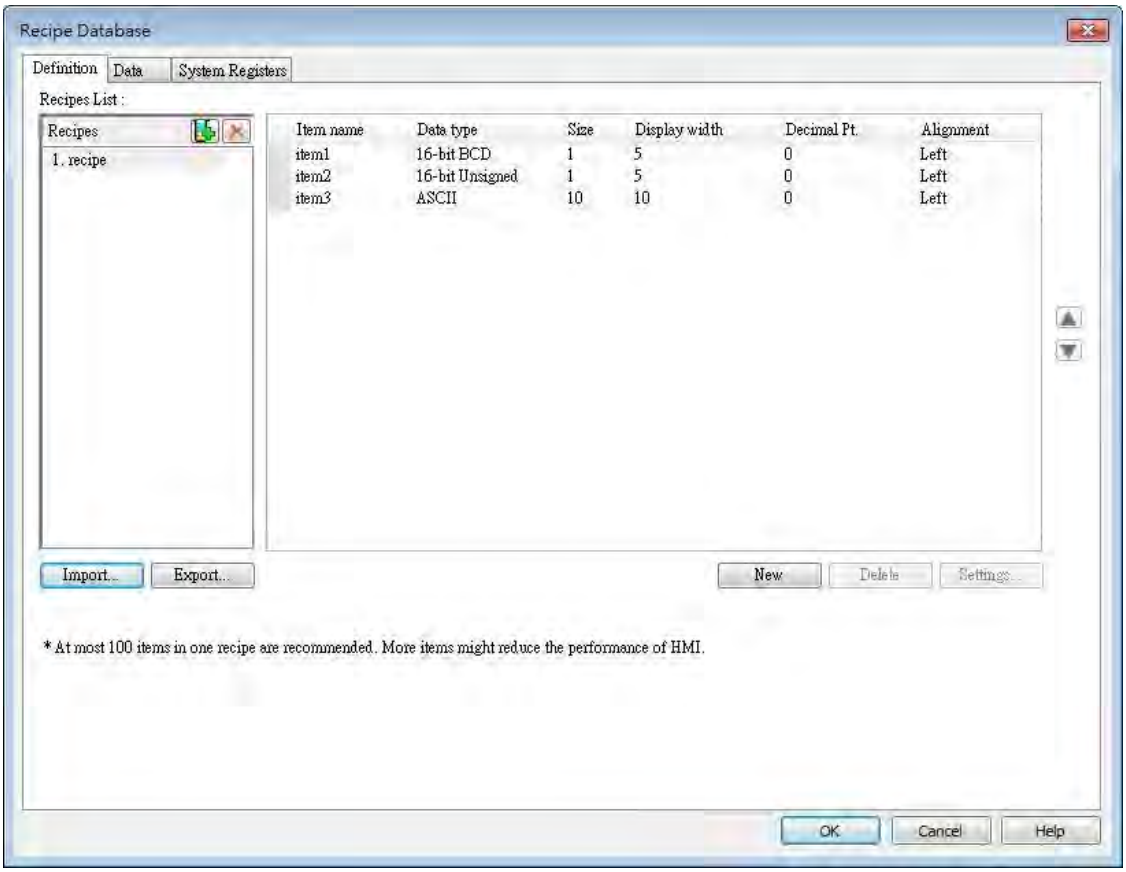
24.3. Recipe Database Settings

In EasyBuilder Pro, open [Data/History] » [Recipe Database] to define recipes, set transfer method and edit recipe data.

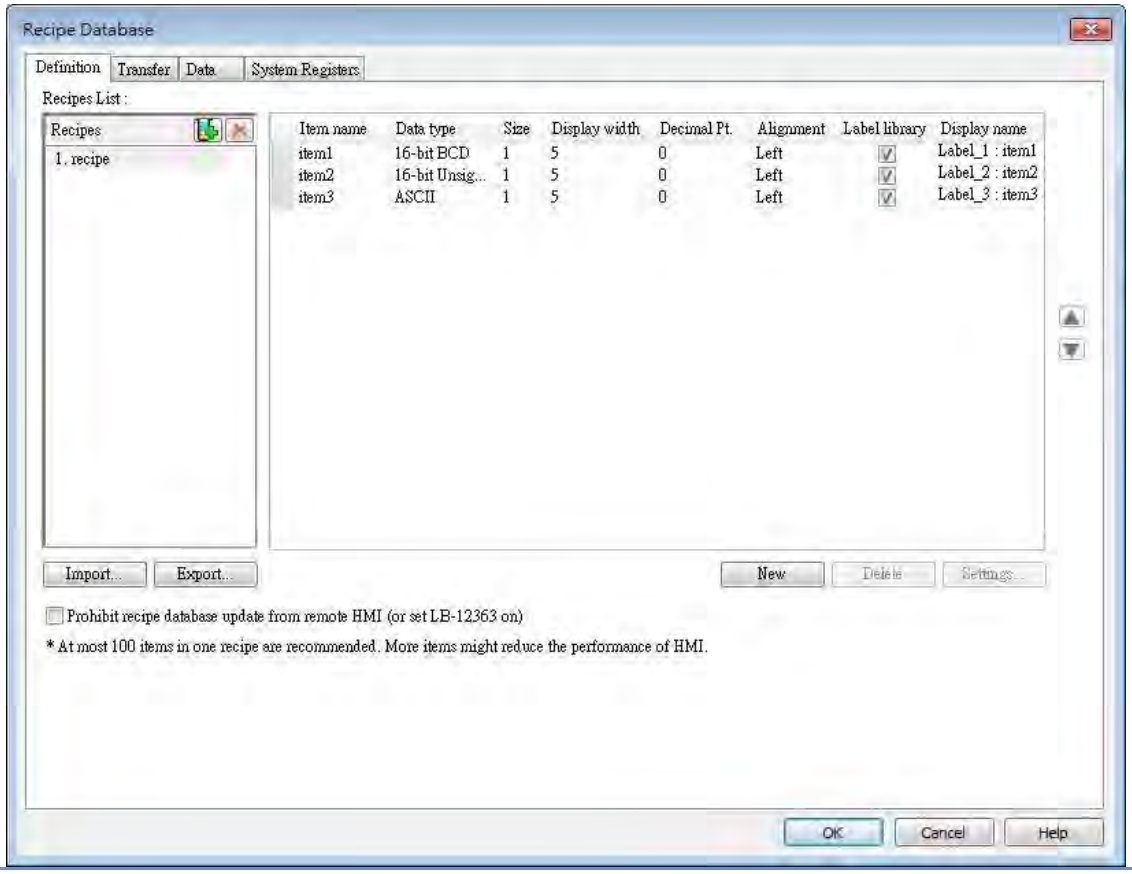
24.3.1. Definition

In Definition tab configure recipe database settings including recipe name, item name, data type…etc.

eMT, iE, XE, mTV Series



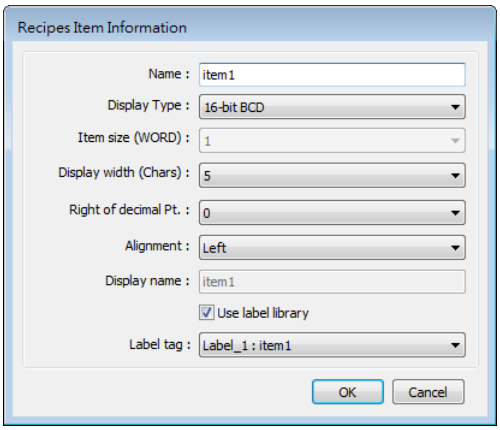
cMT / cMT X Series



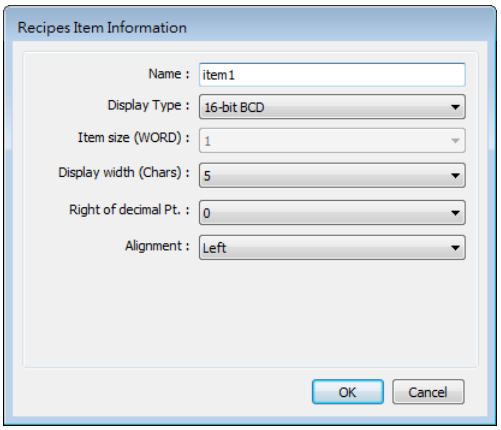
Setting	Description
Recipes List	Add or delete a recipe, the maximum is 100 recipes.
New	Add a new item, the maximum is 1000 items.
Settings	Configure recipe settings, see the following description.
Delete	Delete the selected items.
Import	Import recipe definition.
Export	Export recipe definition. The exported file will not contain recipe records.

Click [Settings]:

cMT, cMT X Series



eMT, iE, XE, mTV Series



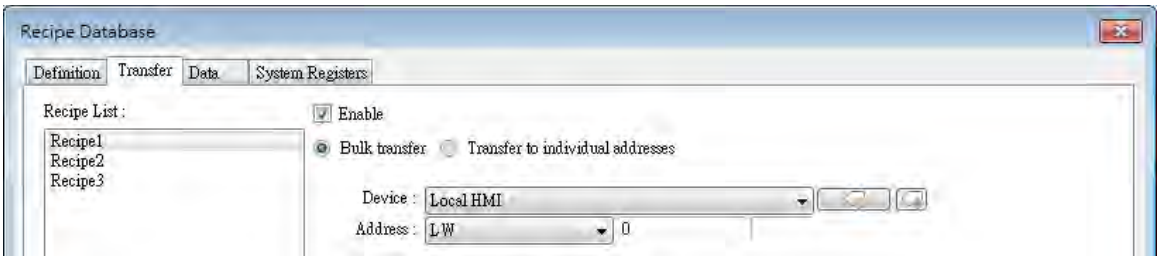
Setting	Description
Name	Enter recipe item name.
Display Type	Setting item data type.
Item size (WORD)	Setting the size of the item.
Display with (Chars)	Setting the number of characters of the item to be displayed.
Right of decimal pt.	Setting the decimal place when displaying data.
Alignment	Setting the alignment when displaying data. [Align left], [Align center], and [Align right] can be selected.
Use label library	On cMT / cMT X series, setting display name is possible. The display name is by default the same as the item name or it may be a tag from Label Tag Library. Please see Chapter 16 Address Tag Library for more details.

24.3.2. Transfer

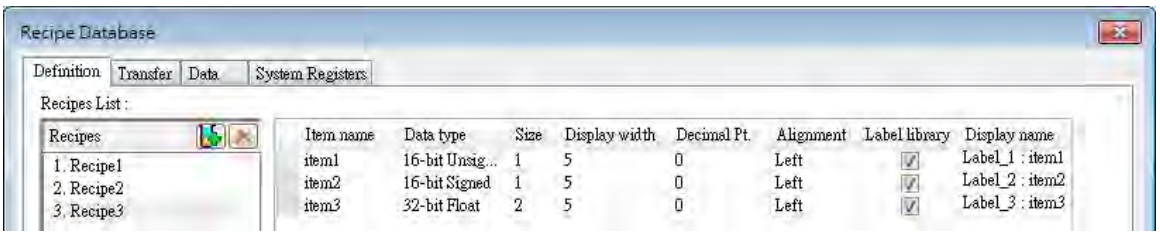
Transfer tab contains the settings for transferring records between the recipe database and designated destinations.

Bulk transfer

Transfer destinations of a record are consecutive addresses starting from the designated address.



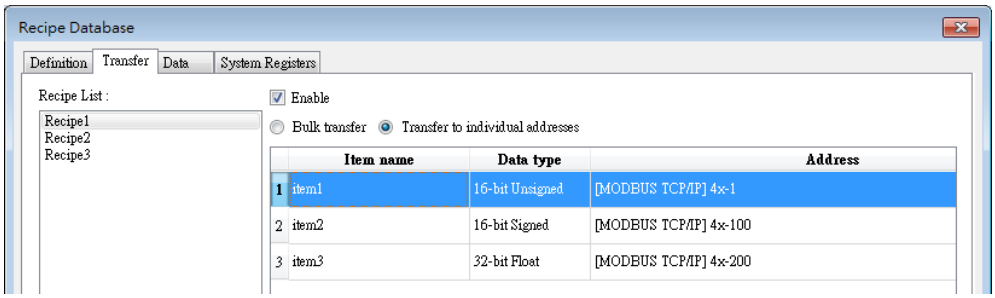
Under this mode, the data type settings of destination addresses must follow those of the recipe items. The following table shows the bulk transfer result when LW-0 is designated as the starting address given settings in the following figure.



Item Name	Destination	Data Type of Dest. Address
item1	LW-0	16-bit Unsigned
item2	LW-1	32-bit Signed
item3	LW-3	32-bit Float

Transfer to individual addresses

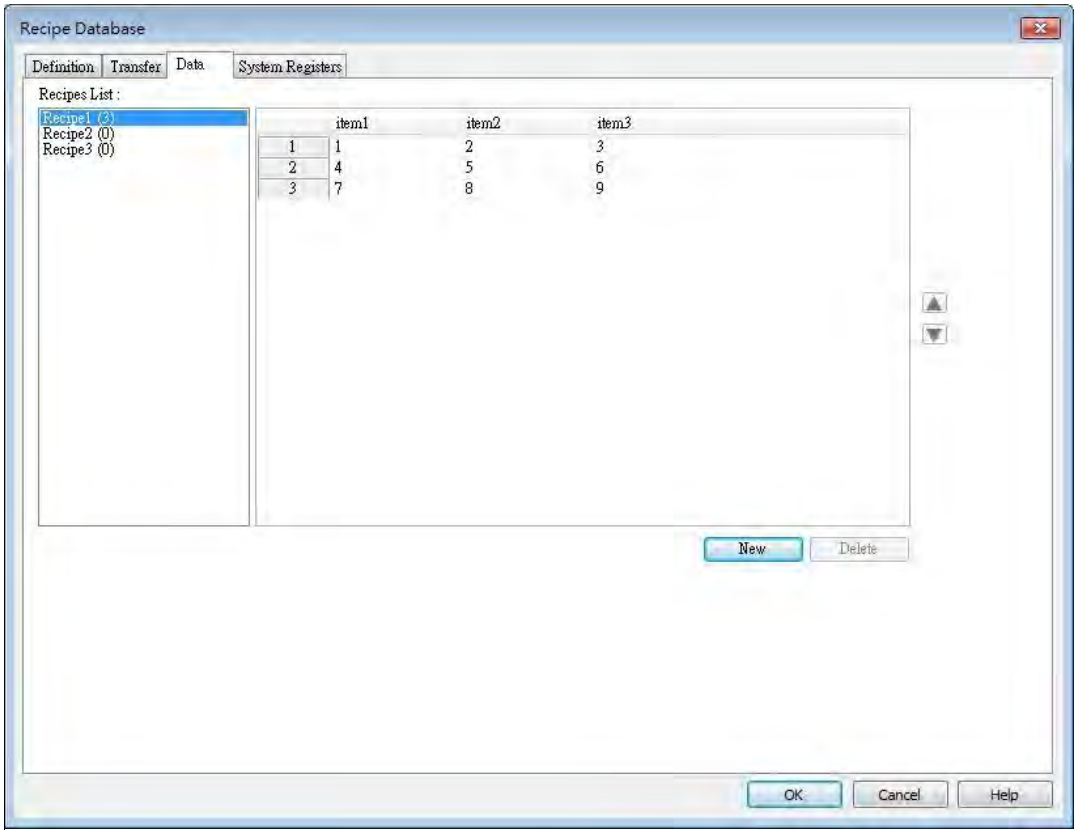
Transfer destination of each item within a record is configured individually.



Under this mode, the data type setting of each destination addresses must also follow those of the corresponding recipe item. In the settings shown above, the data type of address 4x-1 should be 16-bit Signed, and 4x-100 should be 32-bit Signed, and so on.

24.3.3. Data

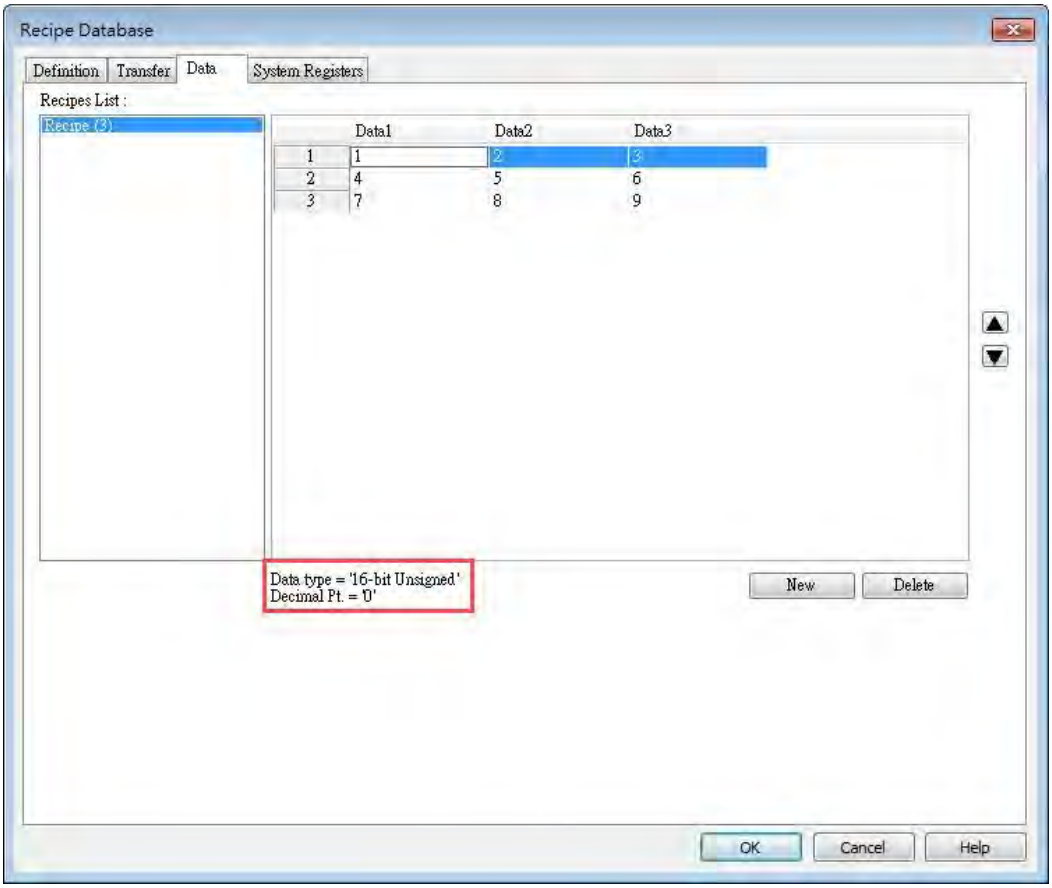
Data tab contains an editor for recipe content. Before recipe content can be edited in this tab, the data type of the items should be set in Definition tab.



In the example shown above, Recipe created in the Definition tab can be found. Items shown on the right hand side come from item names added in Definition tab. Data of each item name can be configured in this tab following the data type defined in the Definition tab.

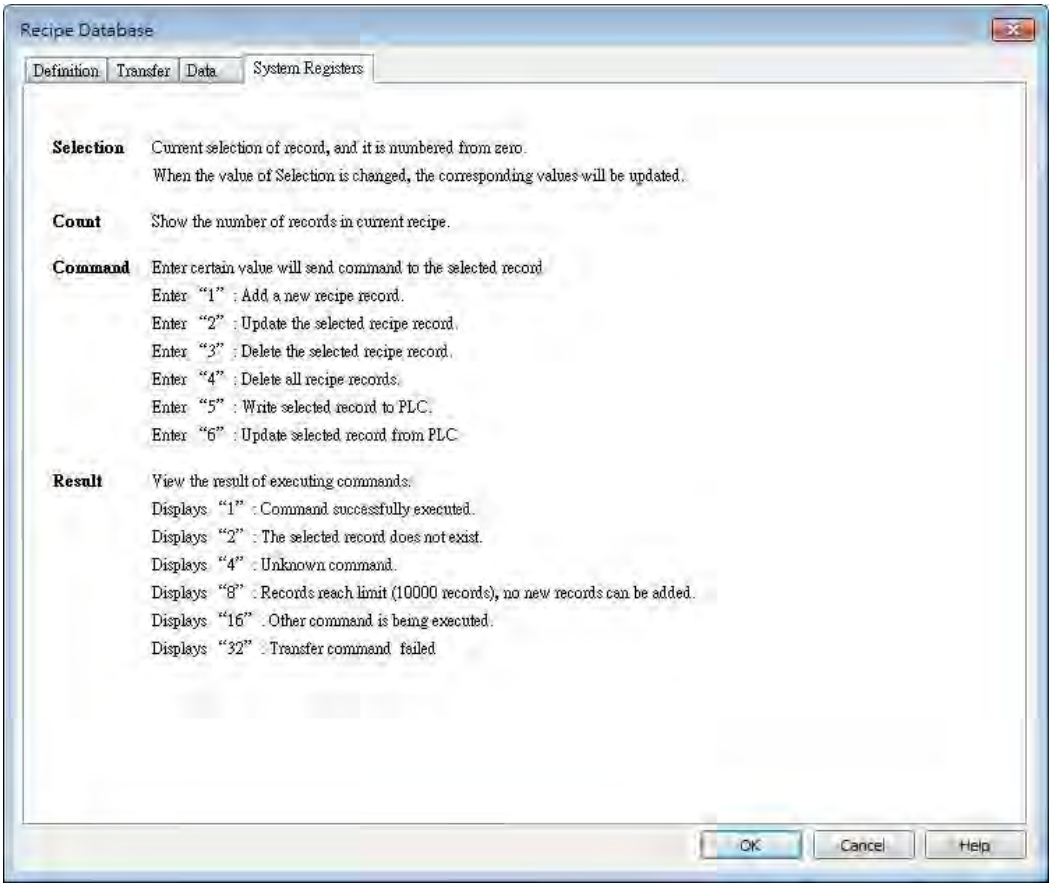
Setting	Description
Recipes:	The recipes created in Definition tab. The number enclosed in brackets shows the total number of records in the corresponding recipe.
Add	Adds records into the recipe according to the defined data type.
Delete	Deletes the edited content.
Up / Down Arrows	Moves the selected record upward / downward.

To define recipes according to the specified data type, click [Add] button above the record list to insert a new record and start editing each item. When click on the item, the item's data type will be shown under the record list. This helps users to fill in each item with appropriate value. Click [OK] to confirm and save the records.



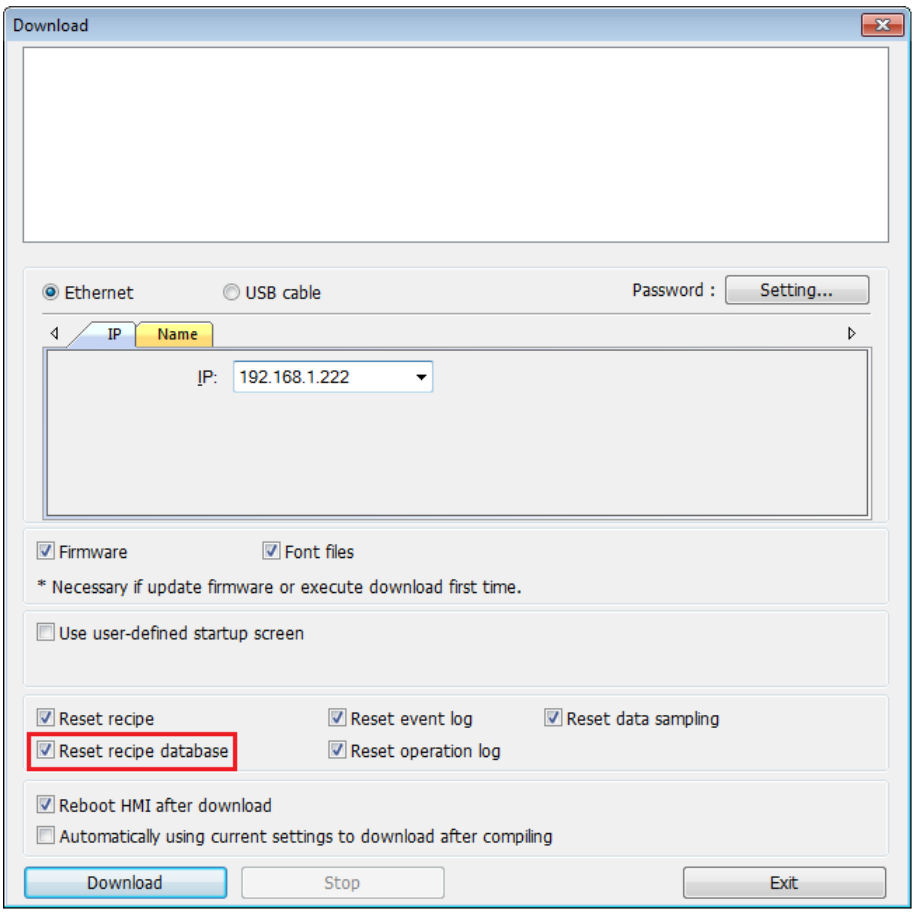
24.3.4. System Registers

In System Register tab, a list of recipe database related control addresses and their descriptions can be found.



Note

- The maximum number of Recipe Database acceptable in a project file is 100 Recipe Databases. A Recipe Database can contain 1000 items.
- Each recipe can hold a maximum of 10000 records.
- The maximum data length acceptable in a Recipe Database is 2000 words. Exceeding the limit can lead to compilation failure.
- The name of Recipe Database and recipe items should be in alphanumeric.
- The recipe records will be stored in the .exob file after compilation and will be downloaded to HMI. These recipes cannot be shared with other project files. If users need to modify the recipe contents and download them to the HMI, make sure that [Reset recipe database] check box is selected during download. If not, the recipe database in the HMI will not be updated.



Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

25. EasyConverter

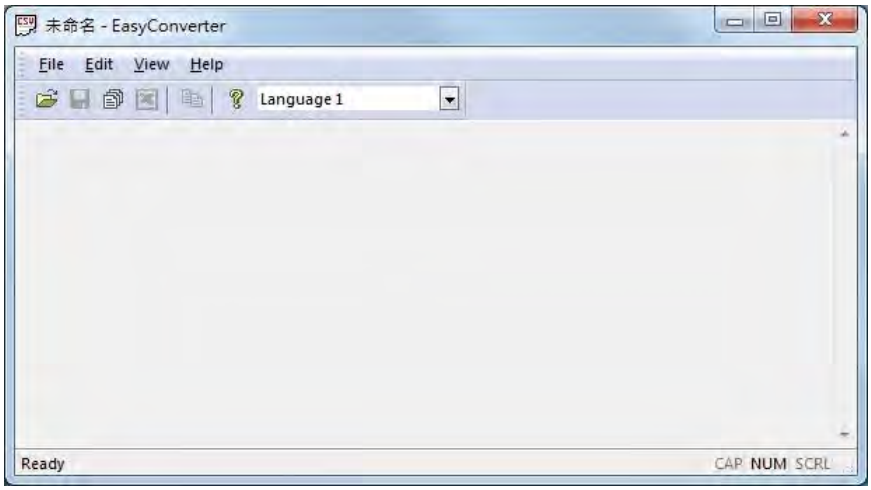
This Chapter explains how to use EasyConverter.

25.1. Overview 25-2
25.2. Converting Data Log File to Excel File 25-2
25.3. Converting Event Log File to Excel File 25-4
25.4. Converting Operation Log File to Excel File 25-6
25.5. Converting Multiple Files 25-7
25.6. Scaling Function 25-8
25.7. Batch File 25-9
25.8. Examination of Historical Data Integrity 25-13

25.1. Overview

EasyConverter reads the Data Log file, Event Log file, and Operation Log file in HMI and convert the files to Excel format.

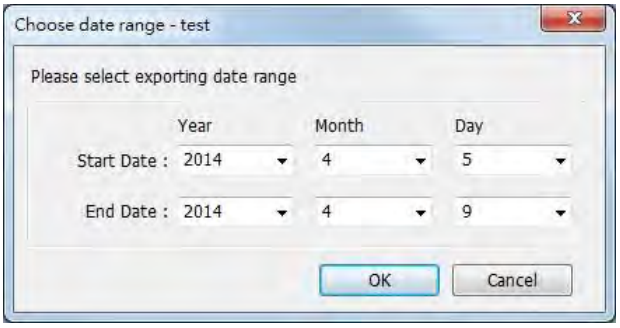
- From Utility Manager click [Data Conversion] » [EasyConverter].
- From EasyBuilder Pro menu select [Tool] » [Data/Event Log Converter].



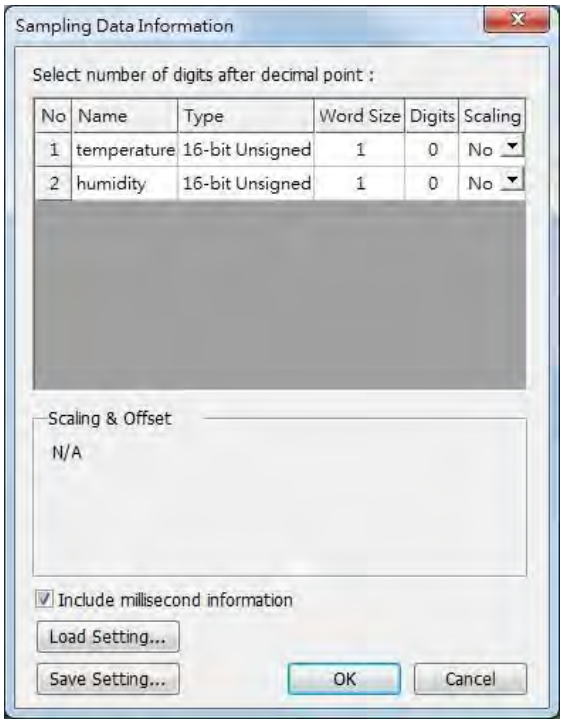
Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

25.2. Converting Data Log File to Excel File

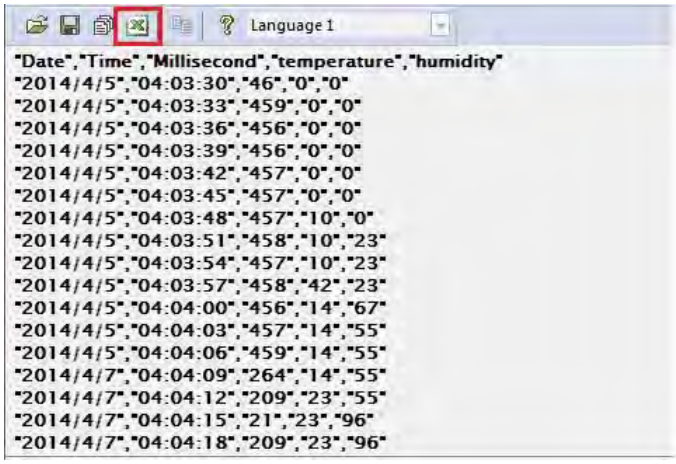
1. If the Data Log file format is .db, and the file includes data of more than one day, the data to be viewed can be specified by selecting a date range. (If the file format is .dtl, please skip this step.)



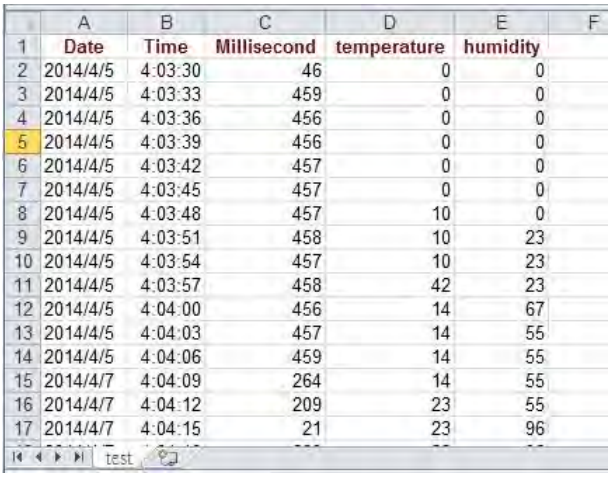
2. The following is the setting dialog box, please set based on actual needs.



3. Click [OK], the Data Log layout is shown in the following figure. Click [Export to Excel]. The file will be converted to Excel format.



4. The Excel layout is shown in the following figure.



	A	B	C	D	E	F
1	Date	Time	Millisecond	temperature	humidity	
2	2014/4/5	4:03:30	46	0	0	
3	2014/4/5	4:03:33	459	0	0	
4	2014/4/5	4:03:36	456	0	0	
5	2014/4/5	4:03:39	456	0	0	
6	2014/4/5	4:03:42	457	0	0	
7	2014/4/5	4:03:45	457	0	0	
8	2014/4/5	4:03:48	457	10	0	
9	2014/4/5	4:03:51	458	10	23	
10	2014/4/5	4:03:54	457	10	23	
11	2014/4/5	4:03:57	458	42	23	
12	2014/4/5	4:04:00	456	14	67	
13	2014/4/5	4:04:03	457	14	55	
14	2014/4/5	4:04:06	459	14	55	
15	2014/4/7	4:04:09	264	14	55	
16	2014/4/7	4:04:12	209	23	55	
17	2014/4/7	4:04:15	21	23	96	

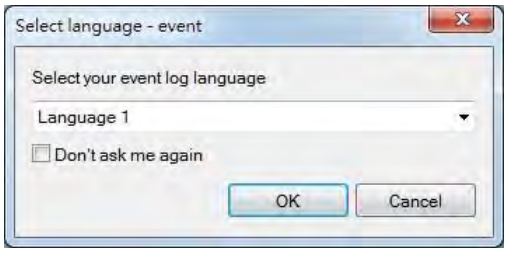
Note

- If the file requires over six million cells in Excel format, only partial data will be shown in EasyConverter. (The complete data will still be exported to xls / xlsx file.)
- The file will be automatically separated into different sheets in the xls / xlsx file under these conditions:
  1. Exceeds 60 thousand rows in a single sheet.
  2. Exceeds 1.5 million cells in a single sheet.

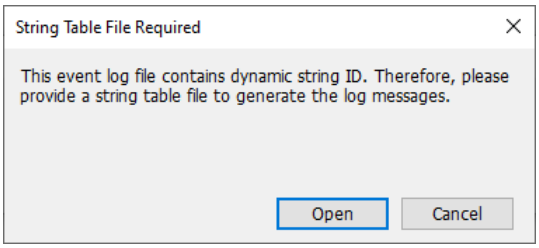
25.3. Converting Event Log File to Excel File

1. If the Event Log file format is .db, and the file includes data of more than one day, the data to be viewed can be specified by selecting a date range.
   (If the file format is .evt, please skip this step.)

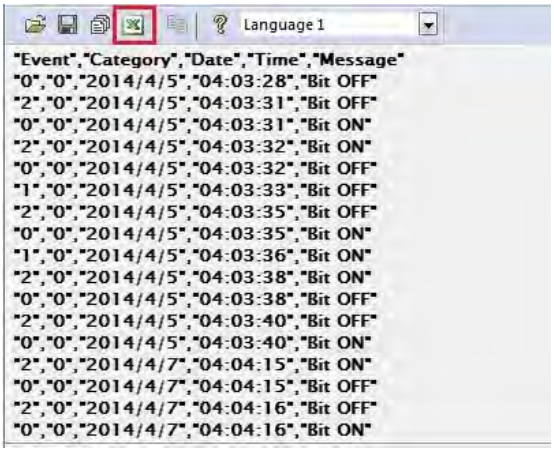
2. If the .db file of Event Log contains multiple languages, the language to be viewed can be specified. (If the file format is .evt, please skip this step.)



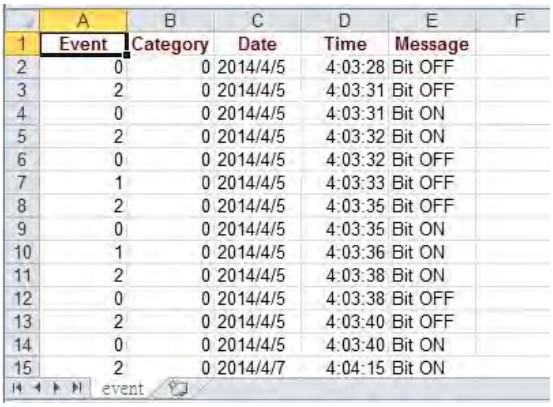
3. When the .db file of Event Log contains dynamic string ID, to correctly display the corresponding string of a triggered event, please manually select a .csv file path of string table.



4. Click [OK], the Event Log layout is shown in the following figure. Click [Export to Excel]. The file will be converted to Excel format.



5. The Excel layout is shown in the following figure.



	A	B	C	D	E	F
1	Event	Category	Date	Time	Message	
2	0	0	2014/4/5	4:03:28	Bit OFF	
3	2	0	2014/4/5	4:03:31	Bit OFF	
4	0	0	2014/4/5	4:03:31	Bit ON	
5	2	0	2014/4/5	4:03:32	Bit ON	
6	0	0	2014/4/5	4:03:32	Bit OFF	
7	1	0	2014/4/5	4:03:33	Bit OFF	
8	2	0	2014/4/5	4:03:35	Bit OFF	
9	0	0	2014/4/5	4:03:35	Bit ON	
10	1	0	2014/4/5	4:03:36	Bit ON	
11	2	0	2014/4/5	4:03:38	Bit ON	
12	0	0	2014/4/5	4:03:38	Bit OFF	
13	2	0	2014/4/5	4:03:40	Bit OFF	
14	0	0	2014/4/5	4:03:40	Bit ON	
15	2	0	2014/4/7	4:04:15	Bit ON	

event

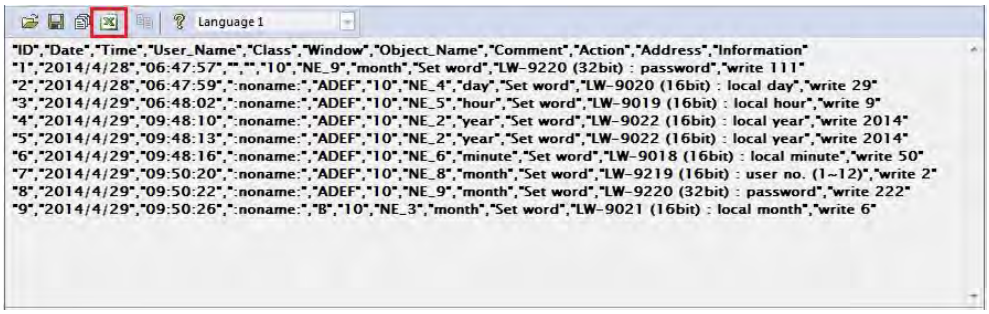
Note

- The "Event" column can be found. 0-> Event triggered; 1-> Event acknowledged; 2-> Event returns to normal.
- If the file requires over six million cells in Excel format, opening the file in EasyCoverter only partially shows the data. (The complete data will be exported to xls / xlsx file.)
- The file will be automatically separated into different sheets in the xls / xlsx file under these conditions:
  1. Exceeds 60 thousand rows in a single sheet.
  2. Exceeds 1.5 million cells in a single sheet.
- When a string table is used, and its .csv file is placed in the same folder as event log's .db file with the same filename (e.g. event log: event.db, string table: event.csv), then the string table can be automatically loaded when loading the .db in EasyConverter; that is, no prompt window pops up asking for the file path of the string table.
- In EasyConverter, the maximum number of rows allowed in a file is 285 thousand rows; that is, the file cannot be opened in EasyConverter after exceeding this limit.

25.4. Converting Operation Log File to Excel File

1. If the Operation Log file includes data of more than one day, the data to be viewed can be specified by selecting a date range.

2. Click [OK], the Operation Log layout is shown in the following figure. Click [Export to Excel]. The file will be converted to Excel format.



3. The Excel layout is shown in the following figure.

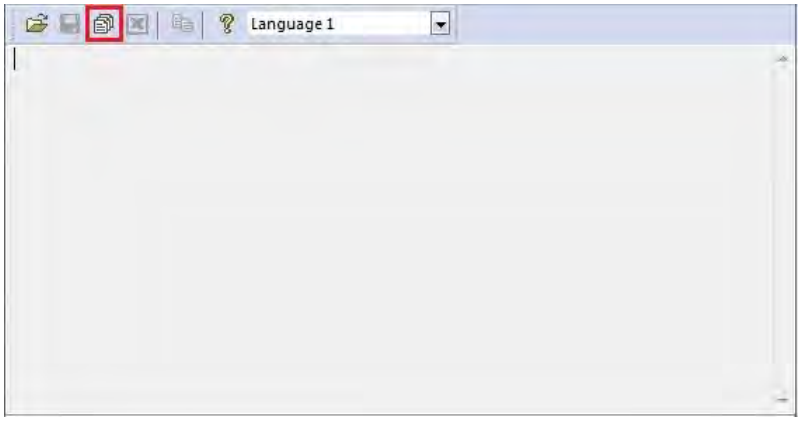
ID	Date	Time	User_Name	Class	Window	Object_Name	Comment	Action	Address	Information
1	2014/4/28	6:47:57			10	NE_9	month	Set word	LW-9220 (32bit) : password	write 111
2	2014/4/28	6:47:59	:noname:	ADEF	10	NE_4	day	Set word	LW-9020 (16bit) : local day	write 29
3	2014/4/29	6:48:02	:noname:	ADEF	10	NE_5	hour	Set word	LW-9019 (16bit) : local hour	write 9
4	2014/4/29	9:48:10	:noname:	ADEF	10	NE_2	year	Set word	LW-9022 (16bit) : local year	write 2014
5	2014/4/29	9:48:13	:noname:	ADEF	10	NE_2	year	Set word	LW-9022 (16bit) : local year	write 2014
6	2014/4/29	9:48:16	:noname:	ADEF	10	NE_6	minute	Set word	LW-9018 (16bit) : local minute	write 50
7	2014/4/29	9:50:20	:noname:	ADEF	10	NE_8	month	Set word	LW-9219 (16bit) : user no. (1~12)	write 2
8	2014/4/29	9:50:22	:noname:	ADEF	10	NE_9	month	Set word	LW-9220 (32bit) : password	write 222
9	2014/4/29	9:50:26	:noname:	B	10	NE_3	month	Set word	LW-9021 (16bit) : local month	write 6

Note

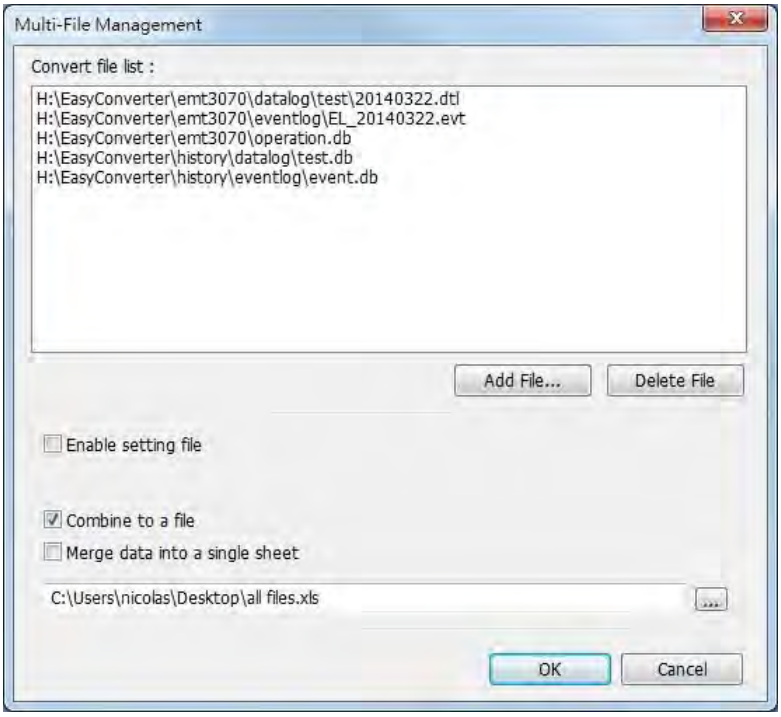
- If the file requires over six million cells in Excel format, opening the file in EasyConverter only partially shows the data. (The complete data will be exported to xls / xlsx file.)
- The file will be automatically separated into different sheets in the xls / xlsx file under these conditions:
  1. Exceeds 60 thousand rows in a single sheet.
  2. Exceeds 1.5 million cells in a single sheet.

25.5. Converting Multiple Files

1. Click [Multi-File] to open the following dialog box.



2. Click [Add File…] to add the files to be converted. If click [OK] without selecting [Combine to a file], the files will be exported to separate Excel files.



3. If [Combine to a file] is selected, the files will be separated into different sheets of one Excel file as shown in the following figure.

	A	B	C	D	E	F	G
1	Date	Time	Millisecond	temperature	humidity		
2	2014/3/22	6:36:52	260	2	1		
3	2014/3/22	6:36:55	250	6	3		
4	2014/3/22	6:36:58	250	10	6		
5	2014/3/22	6:37:01	300	13	8		
6	2014/3/22	6:37:04	280	17	10		
7	2014/3/22	6:37:07	250	21	13		
8							
9							
10							
11							
12							
13							
14							
15							

20140322 EL_20140322 operation test event

Note

- The files cannot be combined when the total size of the files exceeds 32MB.

25.6. Scaling Function

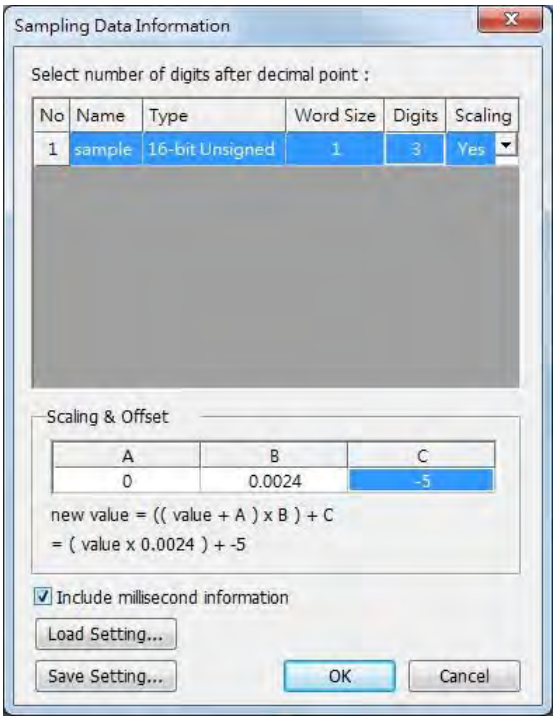
When opening a Data Log file, the scaling function can be set.

The equation of scaling new value = [(value + A) x B] + C, and users can set the values of A, B, and C.

A -> lower limit of the value ; B -> [(scaled max) - (scaled min) / (upper limit) - (lower limit)] ; C -> scaled min.

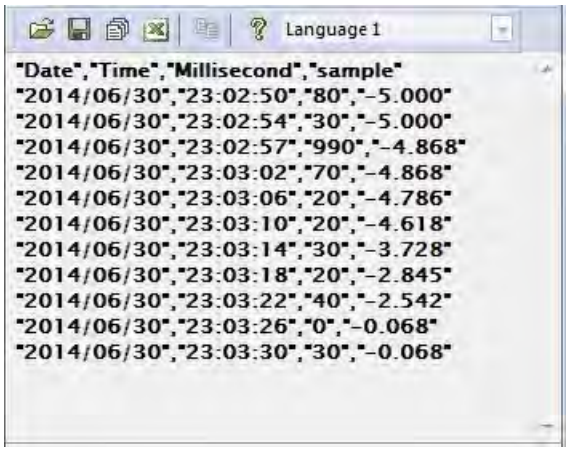
For example, here is a voltage data with a format of 16-bit unsigned (range: 0 ~ 4096).

To convert the data to volt, range form -5V to +5V, the new value = [(value + 0) x 0.0024] + (-5).



Before scaling:

After scaling:



The settings described above can be saved as a settings file in *.lgs format, and then loaded next time if needed.

25.7. Batch File

EasyConverter command line can execute batch file (.bat), and convert .dtl or .evt files into .xls or .csv files for export. It can also back up historical data from the database to the PC. In the batch file, the user can define the format of the exported file (ex: ASCII, Unicode, or UTF-8), and decide whether or not to include millisecond information or load settings file.
The following explains how to create batch file (.bat) and provides some relevant notes.

Parameters:
[/c{a,8,u}] [/t{0,1}] [/s "Format file"] ["Src file"] ["Dest file"]
[/d{0,1,2}] [/l{0,1,2}] [/h “Hostname”] [/n “Data Log Name”] [/g “Config File”] [/i “Time Interval”] [“Dest file”]

Example:
EasyConverter.exe /ca /t1 /s "C:\Format.lgs" "C:\Src.dtl" "C:\Dest.csv"
EasyConverter.exe /t1 /s "C:\Format.lgs" "C:\Src.dtl" "C:\Dest.xls"
EasyConverter.exe /d1 /l0 /h "hostname" /g "D:\config.ini" /i "5~2" "D:\test\"
EasyConverter.exe /d1 /l1 /h "hostname" /n “log000” /g "D:\config.ini" "D:\test\"

Parameter	Description
/c{a,8,u}	(Optional) Only required when exporting a CSV file. /ca, ASCII (Default) /c8, UTF-8 /cu, Unicode
/t{0,1}	(Optional) Select whether to include milliseconds.

	/t0, no millisecond information /t1, have millisecond information (Default)
/s	(Optional) To specified data format from source file. Specified /s: Need to specify "Format file". "Format file", File path of the imported *.lgs file. (e.g. "C:\Format.lgs")
/d{0,1,2}	(Optional) To specified database type to open. /d0, Open db file /d1, Connect to MySQL server /d2, Connect to MS-SQL server
/l{0,1,2}	(Optional) Converted log type, must have with MySQL/MS-SQL server. /l0, Event log /l1, Data log /l2, Operation log
/h	(Optional) To specified (HMI) hostname of table prefix, must have with MySQL/MS-SQL server. Specified /h: Need to specify "Hostname" "Hostname", hostname of HMI. (e.g. "cMT-XXXX")
/n	(Optional) To specified data log name of table prefix, must have when converting data log with MySQL/MS-SQL server. Specified /n: Need to specify "Datalog name" "Datalog name", data log name of HMI. (e.g. "log000")
/g	(Optional) Specifies the directory of the config.ini file. This parameter is necessary when /d1 or /d2 is used. e.g. /g ”D:\config.ini” The file can be downloaded by clicking this download icon.
/i	(Optional) To specified relative date interval range Specified /i: Need to specify "m~n" "m~n", convert relative date range from previous m days ago to previous n days, m > n. (e.g. "5~2") m: From m days ago at 0:00 to m days ago at 23:59. m~n: From m days ago at 0:00 to n days ago at 23:59. m~: From m days ago at 0:00 to the last record in the db. ~n: From the first record in the db to n days ago at 23:59. e.g. /i "m~n"

"Src file"	(Optional) The path of source file. (e.g. "C:\Src.dtl") Acceptable file type: .dtl, .evt, *.db
"Dest file"	(Optional) The path of destination file. (e.g. "C:\Dest.xls") Determine the format of the file extension, for .xls, .xlsx, *.csv file.

Note

- If the file name and path of "Dest file" is not specified in command line, the system will export the file to the same path as "Src file".

You can also find the commands above by entering the file path of EasyConverter.exe in Windows cmd.exe as shown in the following window.
Example: Enter “D:\EasyBuilder\EB Pro>EasyConverter.exe –h”

```
C:\Windows\System32\cmd.exe

D:\SVN_v6.05.02\SW>EasyConverter.exe -h

Usage:
[/c{a,8,u}] [/b{0,1}] [/t{0,1}] [/s "Format file"] ["Src file"] ["Dest file"]
[/d{0,1,2}] [/l{0,1,2}] [/h "Hostname"] [/n "Datalog name"] [/g "Config path"] ["Dest file"]

Example:
EasyConverter.exe /ca /b1 /t1 /s "C:\Format.lgs" "C:\Src.dtl" "C:\Dest.csv"
EasyConverter.exe /t1 /s "C:\Format.lgs" "C:\Src.dtl" "C:\Dest.xls"
EasyConverter.exe /d1 /l0 /h "cMT-XXXX" /g "C:\config.ini" "C:\Dest.xls"
EasyConverter.exe /d1 /l1 /h "cMT-XXXX" /n "log000" /g "C:\config.ini" "C:\Dest.xls"
EasyConverter.exe /d1 /l2 /h "cMT-XXXX" /g "C:\config.ini" "C:\Dest.xls"

/c{a,u,8} -- (Option) Only required when exporting a CSV file.
  /ca, ASCII (Default)
  /c8, UTF-8
  /cu, Unicode

/b{0,1} -- (Option) Add BOM (Byte Order Mark) to file header so that EXCEL can interpret non-ASCII strings correctly.
  /b0, do not write BOM
  /b1, write BOM (Default)

/t{0,1} -- (Option) Select whether or not to include milliseconds.
  /t0, no millisecond information
  /t1, have millisecond information (Default)

/s -- (Option) To specified data format from source file.
  Specified /s: Need to specify "Format file"

  "Format file", File path of the imported *.lgs file. (e.g. "C:\Format.lgs")

/v -- (Option) To validate if records within database are not modified.
  Validation mode only validate database, will not convert files

/d{0,1,2} -- (Option) To specified database type to open.
  /d0, Open db file
  /d1, Connect to MySQL server
  /d2, Connect to MS-SQL server

/l{0,1,2} -- (Option) Converted log type, must have with MySQL/MS-SQL server.
  /l0, Event log
  /l1, Data log
  /l2, Operation log

/h -- (Option) To specified (HMI)hostname of table prefix, must have with MySQL/MS-SQL server.
  Specified /h: Need to specify "Hostname"

  "Hostname", hostname of HMI. (e.g. "cMT-XXXX")

/n -- (Option) To specified data log name of table prefix, must have when converting data log with MySQL/MS-SQL server.
  Specified /n: Need to specify "Datalog name"

  "Datalog name", data log name of HMI. (e.g. "log000")

/i -- (Option) To specified relative date interval range
  Specified /i: Need to specify "m~n"

  "m~n", convert relative date range from previous m days ago to previous n days, m > n. (e.g. "5~2")

/g -- (Option) To specified connection info file path
  Specified /g: Need to specify connection info file path "Connection Info Path"
```

Example

To convert the file 20150919 stored in “D:\EasyBuilder\EB Pro\HMI_memory” from .dtl to .xls, and then save the file to the desktop, you can use the following command lines.

Scene 1: If the .bat file is placed in the same directory as EasyConverter, then the command line is:

EasyConverter.exe "D:\EasyBuilder\EB Pro\HMI_memory\20150919.dtl" "C:\Users\Desktop\20150919.xls"

Scene 2: If the .bat file is placed in a different directory from EasyConverter, the directory to store EasyConverter.exe. must be specified, and the command line will be:

"D:\EasyBuilder\EB Pro\EasyConverter.exe" "D:\EasyBuilder\EB Pro\HMI_memory\20150919.dtl" "C:\Users\Desktop\20150919.xls"

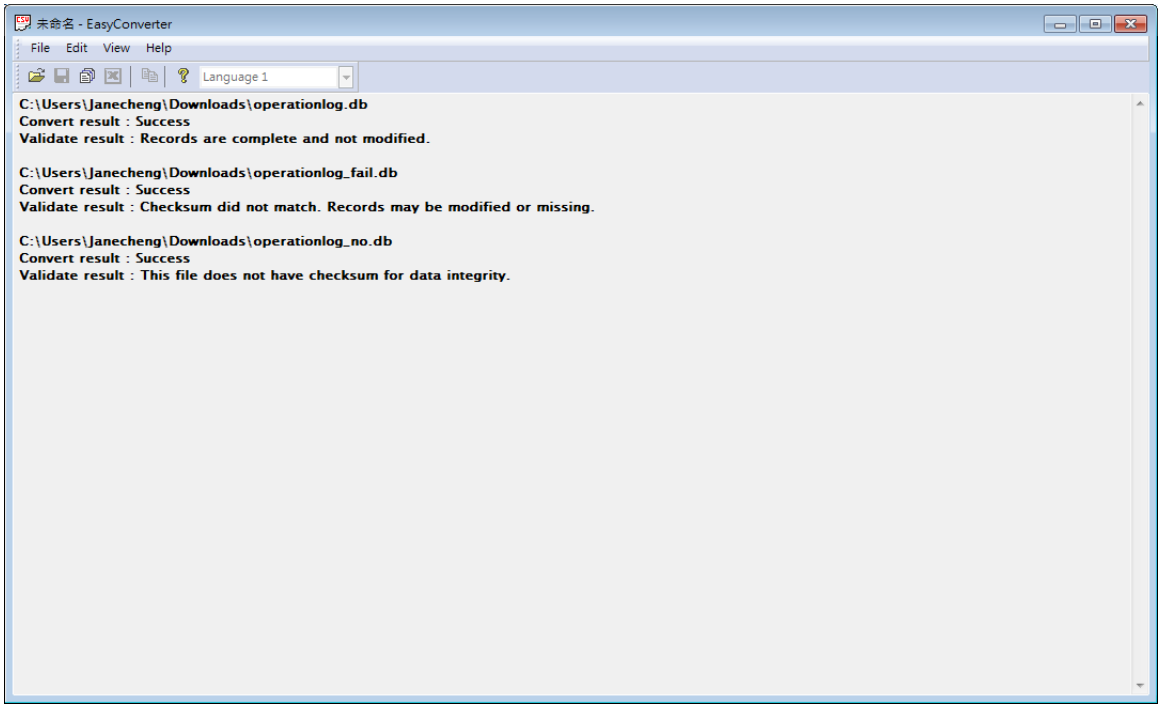
25.8. Examination of Historical Data Integrity

With EasyConverter, operation log file obtained from backup feature can be checked for data integrity by verifying its checksum. When EasyConverter opens a file that might have been tampered with, an alert window below will pop up.

In EasyConverter's status bar, the checksum verification results are illustrated with the following icons:

- The file does not contain checksum for examining data integrity.
- Checksum did not match. Records may have been modified or missing.
- All records are complete, and no modifications have been made.

The checksum verification applies to multi-file conversion as well.



File integrity may also be checked in command line. Use parameter /v to enter verification mode, and no conversion will take place under this mode.

e.g.

EasyConverter.exe /v "C:\Src.db"

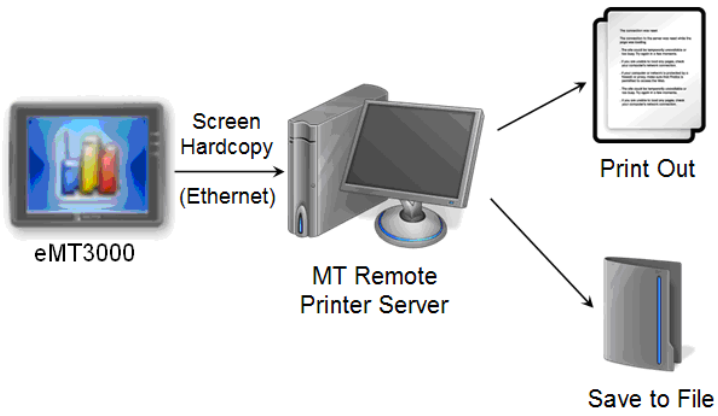
26. EasyPrinter

This chapter explains the setup steps of EasyPrinter.

26.1. Overview 26-2
26.2. Using EasyPrinter as a Printer Server 26-3
26.3. Using EasyPrinter as a Backup Sever 26-5
26.4. EasyPrinter Operation Guide 26-10
26.5. Convert Batch File 26-15

26.1. Overview

EasyPrinter is a Win32 application and can only run on MS Windows 7 SP1 / 8 / 10. It enables HMI to output screen hardcopies to a remote PC via Ethernet. The following explains how to use EasyPrinter.

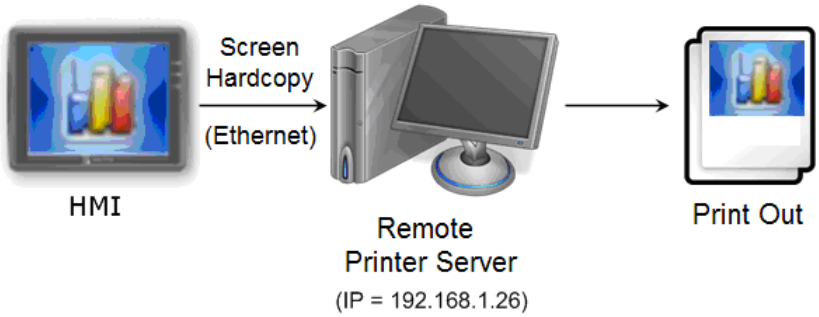


Here are some advantages of using EasyPrinter:

- EasyPrinter provides two modes of hardcopy output: [Print Out] and [Save to File]. Users can use either or both modes.
- Since EasyPrinter runs on MS Windows system, it supports most of the printers available on the market.
- Multiple HMIs can share one printer so users don't have to prepare printers for each HMI.

Additionally, EasyPrinter can also be a backup server. Users can use Backup objects on HMI to copy history files such as Data Sampling records and Event Log to a remote PC via Ethernet. Please see the following illustration:

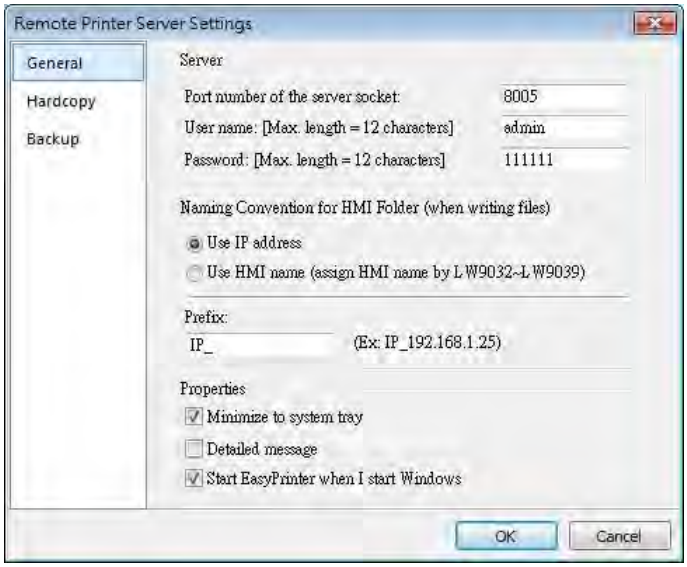
26.2. Using EasyPrinter as a Printer Server



Users can make screen hardcopies with a Function Key object. The hardcopies will be transferred to the Remote Printer Server via Ethernet and then printed out.

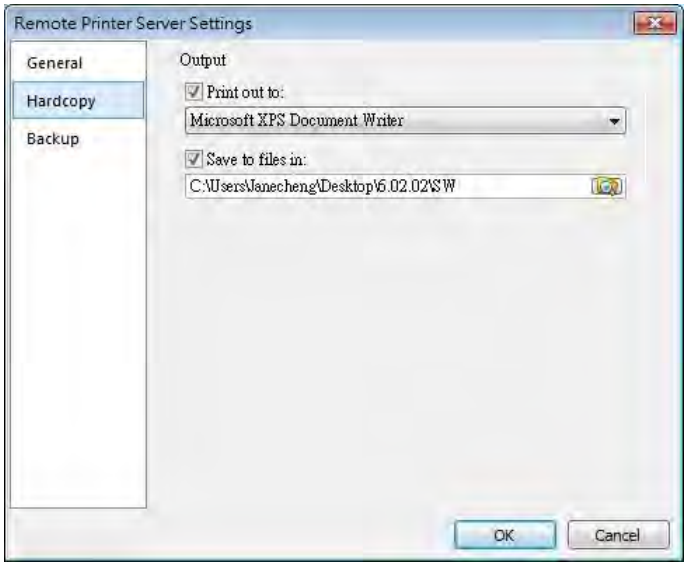
26.2.1. Setup Procedure in EasyPrinter

In EasyPrinter's main menu, select [Options] » [Settings] and the following dialog box appears:



1. Select [General] on the left hand side.
2. In [Server], set [Port number of the server socket] to "8005", [User name] to "admin" and [Password] to "111111". (These are default values.)
3. In [Naming Convention for HMI Folder], select [Use IP address] and enter "IP_" in the [Prefix] field.
4. In [Properties], select [Minimize to system tray] check box. Select [Start EasyPrinter when I start Windows] if necessary.

Set the print out location.

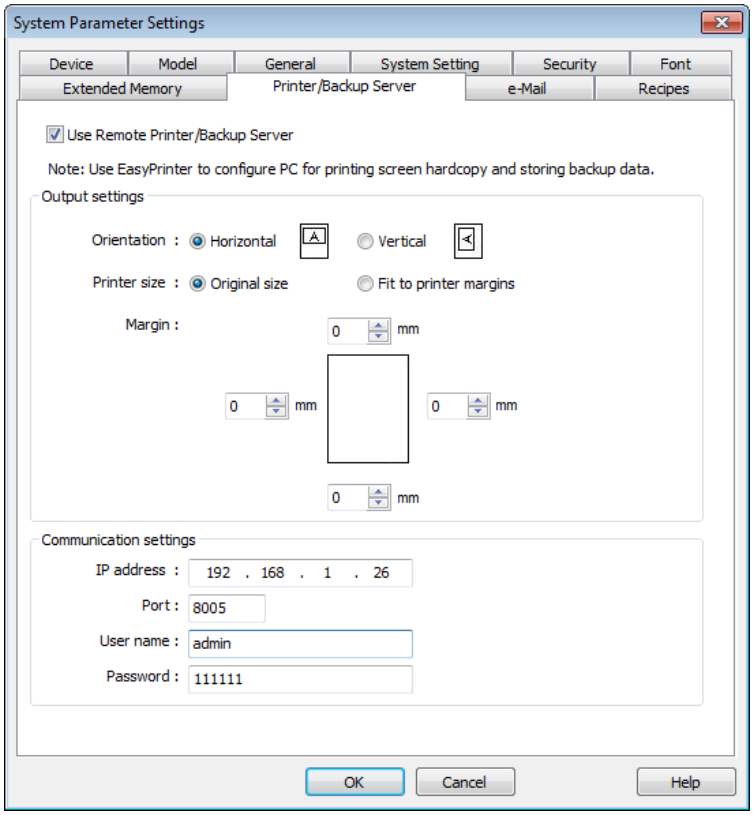


1. Select [Hardcopy] on the left hand side.
2. Under [Output] select [Print out to] and choose a printer as the output device for screen hardcopies. (The printer shown in the image above is an example; please select an actual printer located in your network environment.)
3. Select [Save to files in:] and browse for the directory to save the hardcopy file.
4. Click [OK] to confirm the settings.
5. In EasyPrinter main menu select [File] » [Enable Output] to output any incoming print request.

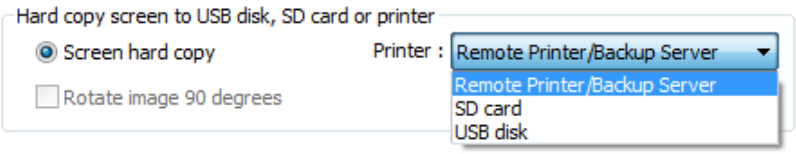
26.2.2. Setup Procedure in EasyBuilder Pro

The setting procedure of EasyPrinter in EasyBuilder Pro:

1. Open a new project or an existing project in EasyBuilder Pro.
2. In EasyBuilder Pro main menu select [Home] » [System Parameters] » [Printer/Backup Server] and select [Use Remote Printer/Backup Server] check box.



3. Under [Output settings] set appropriate values for left / top / right / bottom margins. (The margins are all set to 15mm in the example.)
4. Under [Communication settings] fill in the [IP address] of the printer server according to the settings in EasyPrinter. Set [Port] to “8005”, [User name] to “admin” and [Password] to “111111”.
5. Click [OK].
6. In EasyBuilder Pro main menu select [Objects] » [Button], select [Function Key], select [Screen hardcopy] and set [Printer] to [MT Remote Printer/Backup Server].



7. Place the Function Key object in the common window (window no. 4) so that screen hardcopies can be captured anytime when needed.
8. Compile and download the project to HMI. Press the Function Key object on the screen to make a screen hardcopy.

Note

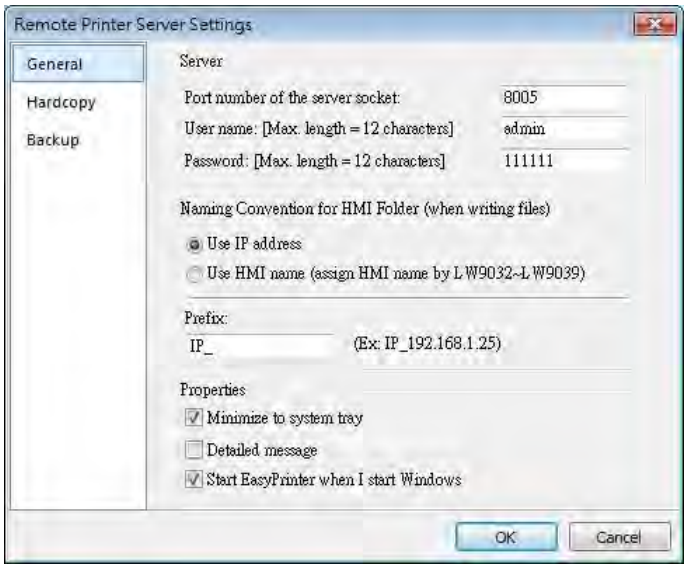
- A PLC Control object can also be used to make screen hardcopies.
- Alarm information cannot be printed via EasyPrinter.
- EasyPrinter can only communicate with HMI via Ethernet. Please check that the HMI has appropriate network settings.

26.3. Using EasyPrinter as a Backup Sever

Backup objects can upload historical data and Operation Log to remote backup server.

26.3.1. Setup Procedure in EasyPrinter

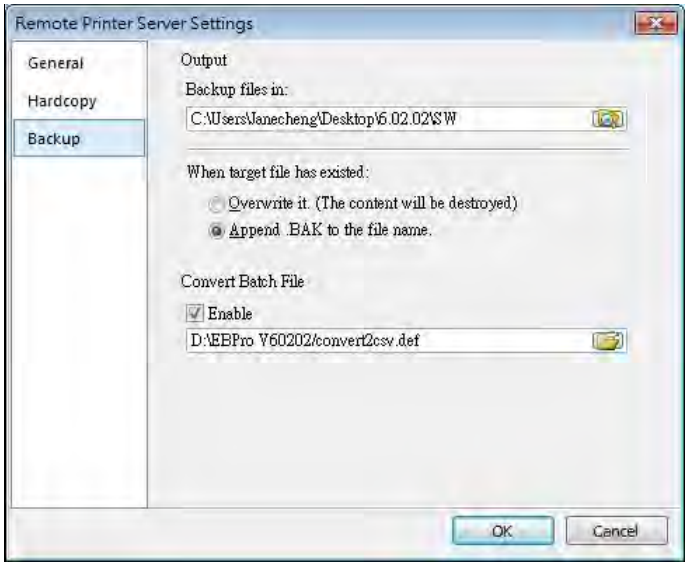
In EasyPrinter's main menu, select [Objects] » [Settings] and the following dialog box will appear:



1. Select [General] on the left hand side.
2. Under [Server] set [Port number of the server socket] to "8005", [User name] to "admin" and [Password] to "111111". (These are default values.)
3. Under [Naming Convention for HMI Folder] select [Use IP address] and enter "IP_" in the [Prefix] field.
4. Under [Properties] select [Minimize to system tray]. Select [Start EasyPrinter when I start

Windows] if necessary.

Set the backup location.

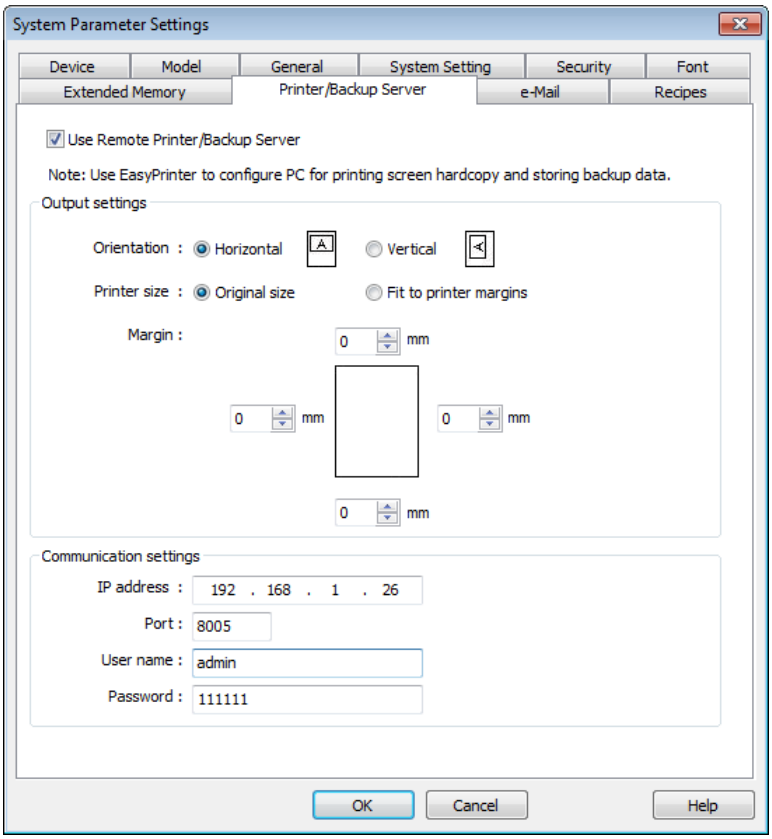


1. Select [Backup] on the left.
2. Under [Output] click the button to browse and select a storage directory of the incoming history files.
3. Click [OK] to confirm the settings.
4. In the main menu, select [File] » [Enable Output] to backup data in the selected directory.

26.3.2. Setup Procedure in EasyBuilder Pro

The setup procedure of EasyPrinter in EasyBuilder Pro:

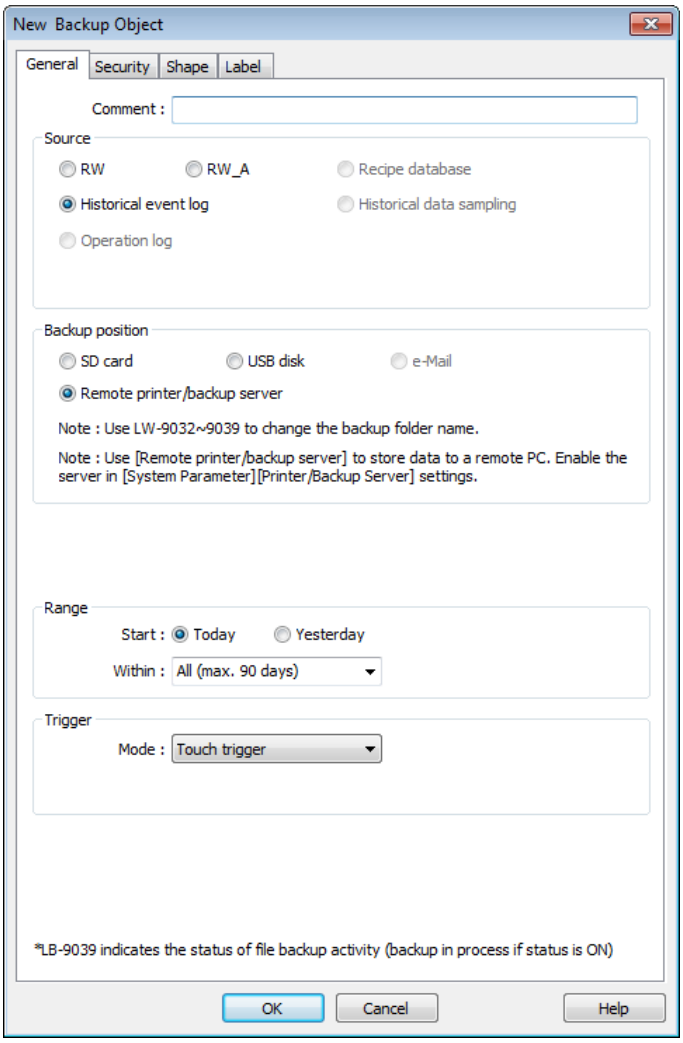
1. Open a new project or an existing project in EasyBuilder Pro.
2. In EasyBuilder Pro's main menu, select [Home] » [System Parameters] » [Printer/Backup Server] and select the [Use Remote Printer/Backup Server] check box.



3. Under [Communication settings] fill in the [IP address] of the printer server according to the settings in EasyPrinter. Set the [Port] to “8005”, [User name] to “admin” and [Password] to “111111”. (Note: These are default values.)
4. Click [OK].

Create a Backup object.

1. In EasyBuilder Pro’s main menu, select [Data/History] » [Backup] and the following dialog box appears:



2. Under [Source] select [Historical event log] (or [RW], [RW_A] if needed.)
3. Under [Backup position] select [Remote printer/backup server].
4. Under [Range] select [Today] and [All] (or other options if needed.)
5. Under [Trigger] select [Touch trigger].
6. Click [OK].
7. Place the [Backup] object in the common window (window no. 4), and users will be able to make backups anytime when needed.
8. Compile and download the project to HMI. Press the Backup object on the screen to make a backup of the history data.

Note

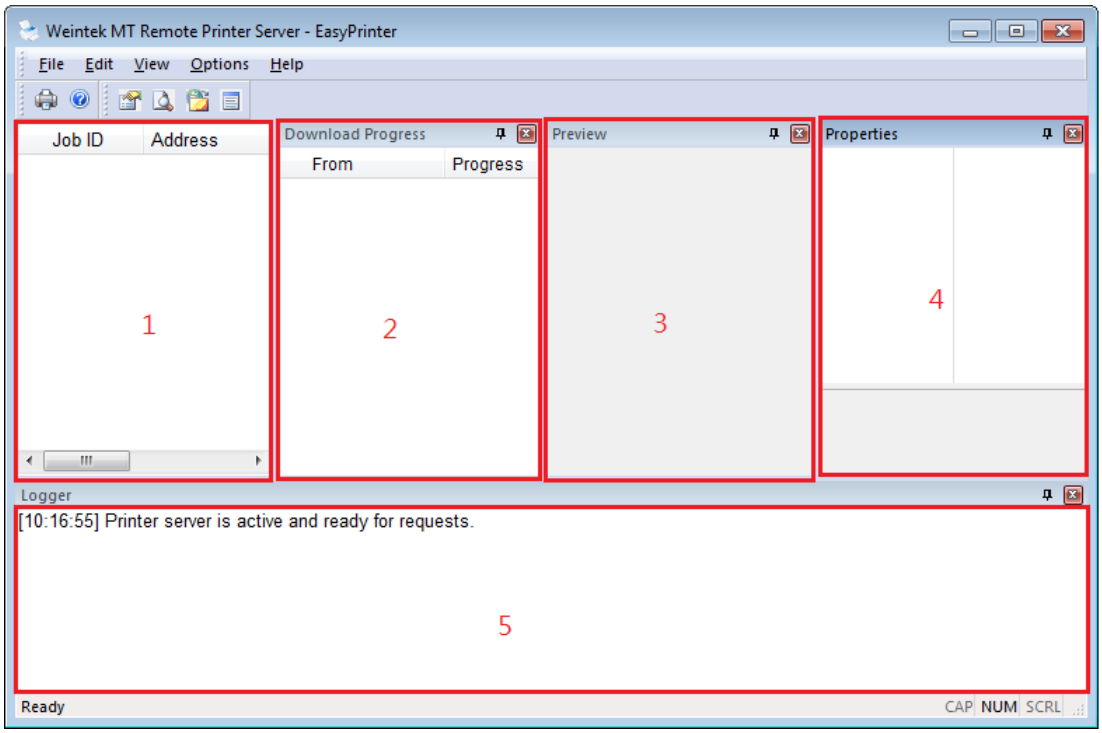
- The Backup object can also be triggered by a bit address.
- Users can arrange a Scheduler object, which turns a bit ON at the end of a week, to trigger the Backup object to automatically back up all history data.

26.4. EasyPrinter Operation Guide

The following introduces the interface and operation of EasyPrinter.

26.4.1. EasyPrinter Managing Window

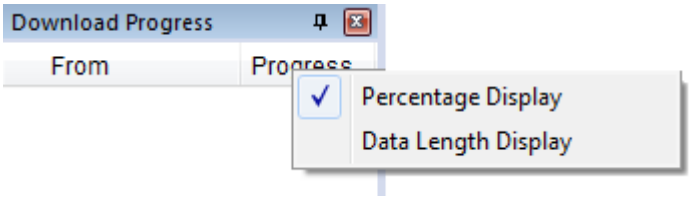
EasyPrinter main menu is divided into 5 parts as shown in the following figure:



Area	Name	Description
1	**Job List**	Lists all incoming tasks, such as screen hardcopy and backup requests.
2	**Download Progress**	Shows the download progress of incoming requests.
3	**Preview**	Shows the preview image of the screen hardcopy task selected from [Job List].
4	**Properties**	Shows the information about the task selected from [Job List].
5	**Logger**	Shows the time and message information of events such as incoming request, incorrect password, etc.

26.4.2. Operation Guide

The following describes the function of EasyPrinter menu items.

Menu	Description
File	**Enable Output** If selected, EasyPrinter processes the tasks one by one, otherwise, EasyPrinter stores the tasks in memory.
Edit	**Edit** Edits screen hardcopy by setting [Orientation], [Scaling] and [Margins]. **Delete** Deletes the selected tasks permanently. **Select All** Selects all tasks from [Job List].
View	**Properties Bar** Shows or hide the Property Window. **Preview Bar** Shows or hide the Preview Window. **Download Bar** In [Download Progress] Window, the mode to display download progress can be set by clicking the header of the [progress] column as shown in the following figure:  **Logger Bar** EasyPrinter can reserve up to 10,000 messages in Message Window. If a new message comes in, the oldest message will be deleted.
Options	Please see the following page.

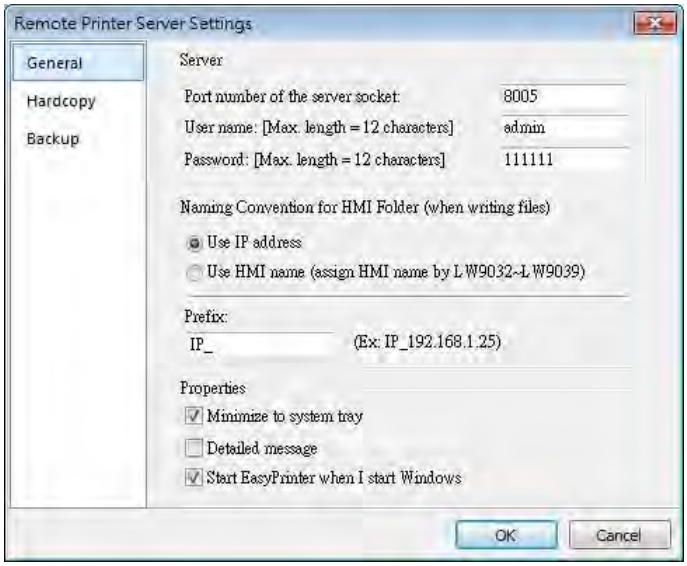
Note

- EasyPrinter can only reserve up to 128 MB of task data in memory. If the memory is full, any request coming in afterwards will be rejected. Users must either operate [Enable Output] or delete some tasks to make room for new tasks.

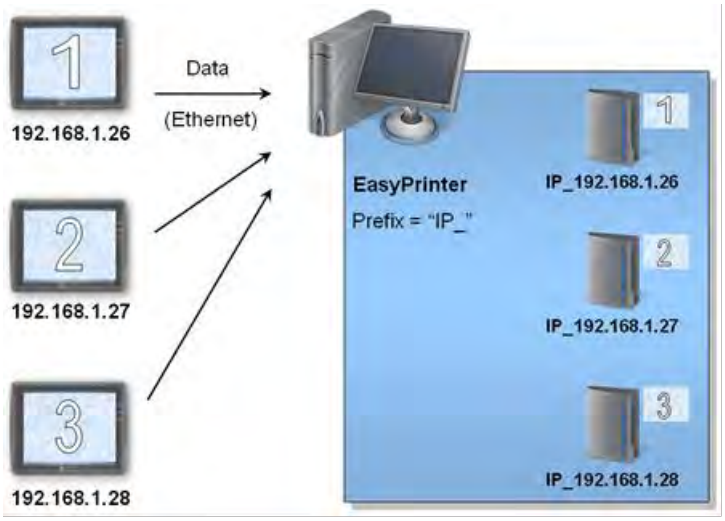
- The backup task is not editable.
- [Edit] is available only when a task is selected.
- [Delete] is available when at least one task is selected.

The following is the detail for [Options] » [Settings]

- In General Tab:



Setting	Description
Server	**Port number of the server socket** Sets the Ethernet port number to connect the HMI. Range: 1 ~ 65535. Default: 8005. **User name / Password** Sets the user name and password to let only authorized HMIs send requests to EasyPrinter.
Naming Convention for HMI Folder	EasyPrinter uses different folders to store files (e.g. hardcopy bitmap files, backup files) from different HMI. There are two ways to name the folders: **Use IP address** EasyPrinter names the folder as [Prefix] + [IP address] after the HMI at this IP address sends request.



Use HMI name

EasyPrinter names the folder in [Prefix] + [HMI name] after the HMI this name indicates sends request.

Properties

Minimize to system tray

If this check box is selected, the EasyPrinter shortcut icon will be placed in the system tray in PC. Double click the on icon in system tray to open EasyPrinter.

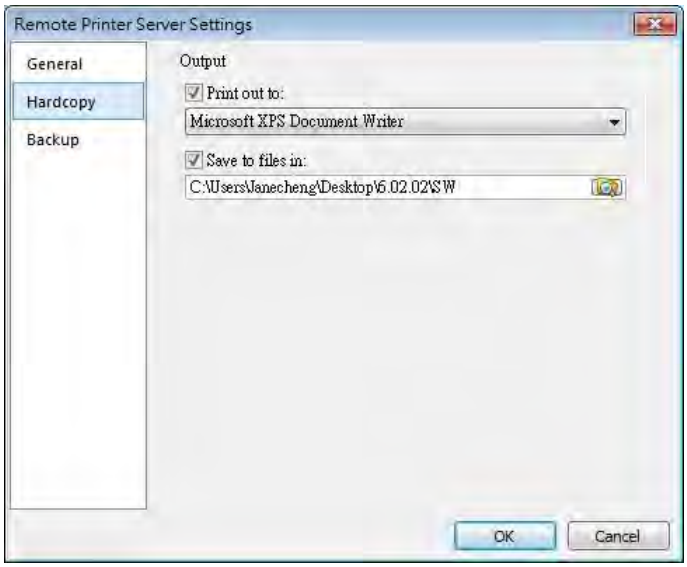
Detailed message

Select this check box to display more detailed messages about events in the message window.

Start EasyPrinter when I start Windows

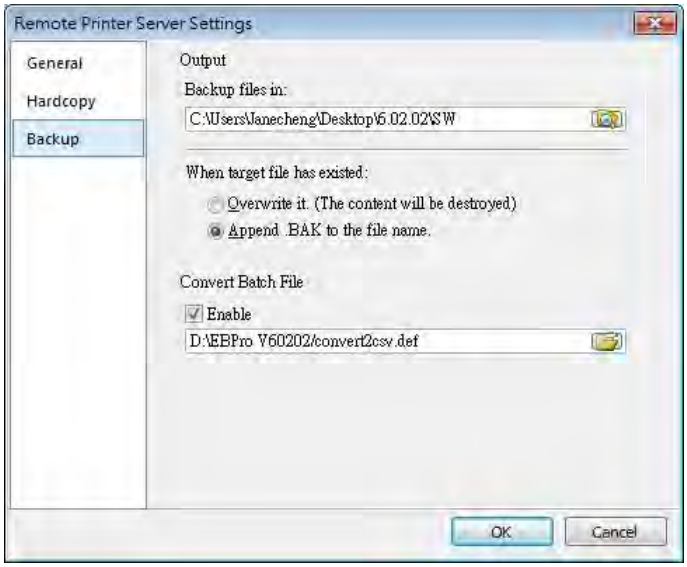
Select this check box to automatically run EasyPrinter when Windows starts.

- In Hardcopy Tab:



Setting	Description
Output	**Print out to** EasyPrinter prints out the hardcopy result with the specified printers. **Save to files in** EasyPrinter converts the hardcopy result into a bitmap file and saves it in the specified directory. The bitmap files are found at: [Specified Path] \ [HMI Folder] \ yymmdd_hhmm.bmp For example, when a hardcopy request is given at 17:35:00, 12/Jan/2009, the bitmap file will be named “090112_1735.bmp”. And if there is another bitmap file generated within the same minute, it will be named “090112_1735_01.bmp” and so on.

- In Backup Tab:

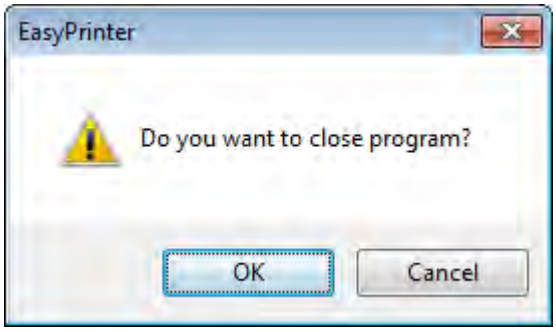


Setting	Description
Output	**Backup files in:** EasyPrinter stores the backup files to the specified path. The upper directory is the same: [*Specified Path*] \ [*HMI Name*] or [*IP address*] The lower directory: ● For Event Log files: \ eventlog\ EL_yyyymmdd.evt ● For Data Sampling files: \ datalog \ [*File name of the Data Sampling*] \ yyyymmdd.dtl

	• For Recipe files: \ recipe \ recipe.rcp or recipe_a.rcp • For Recipe Database: \ recipe \ recipe.db • For Operation Log: \ operationlog\ operationlog.db
Convert Batch File	Select [Enable] to convert the selected history file to .csv or .xls (Excel) format of Convert Batch Files.

Note

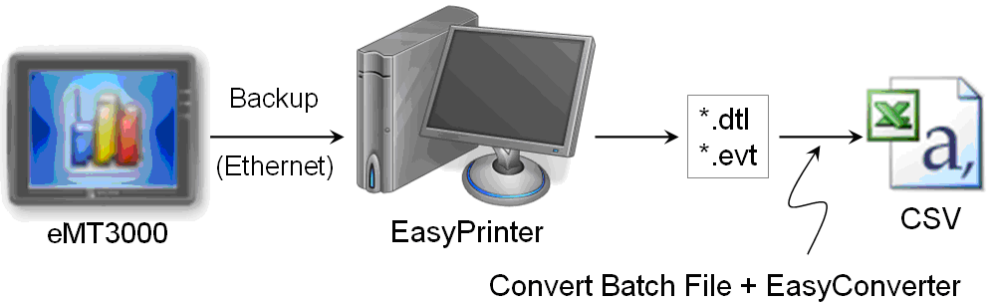
- System registers LW-9032 to LW-9039 can be used to specify HMI name.
- When closing EasyPrinter, the following dialog will appear.



26.5. Convert Batch File

EasyPrinter provides a conversion tool to convert the uploaded Data Sampling and Event Log history files to .csv files automatically. To do so, please select [Enable] under [Convert Batch File] to make EasyPrinter convert the history files.

In the following illustration, the conversion is actually executed by EasyConverter. EasyPrinter simply follows the criteria in Convert Batch File and activates EasyConverter with proper arguments to achieve the conversion.



Note

- EasyConverter is another Win32 application that converts history data into .csv or MS

Excel .xls files. Users can find it in the EasyBuilder Pro installation directory.

- Users requesting this function must ensure EasyPrinter and EasyConverter are placed in the same directory.

26.5.1. The Default Value of Convert Batch File

The following is the default Convert Batch File: convert2csv.def

Listing 1. Default Convert Batch File

```
1: "dtl", "EasyConverter /c $( Pathname)"
2: "evt", "EasyConverter /c $( Pathname)"
```

There are two lines in the file. Each line has two arguments separated by a comma and forms a criterion of how to process a specific type of files. The first argument stands for the extension name of the file type to be processed. The second argument stands for the command to be executed in console mode. Please note that "$(Pathname)" is a key word to inform EasyPrinter to replace it with the real name of the converted backup file. For example, if a Data Sampling history file named 20090112.dtl is uploaded and stored, EasyPrinter will send out the following command to a console window:

```
1: EasyConverter /c 20090112.dtl
```

A file named 20090112.csv is created.

The criteria of the default Convert Batch File:

1. Convert all Data Sampling history files (.dtl) into .csv files.
2. Convert all Event Log history files (.evt) into .csv files.

Note

- "$(Pathname)" in the second argument stands for the full path name of the file. In the previous case, EasyPrinter replaces it with:
  [Specified Path] \ [HMI Folder] \ [datalog] \ [Folder name of the Data-Sampling object] \ 20090112.dtl
- EasyPrinter interprets the Convert Batch File in line basis, that is, each line forms a criterion.
- Any two arguments should be separated by a comma.
- Every argument should be put in double quotes.
- Do not put any comma inside an argument.
- Supported parameters for batch file: $(PathName), $(HmiName), and $(IP) which stand for file path, HMI name, and HMI IP address.
- When [Start EasyPrinter when I start Windows] option is selected, please add in the file path where EasyConverter is placed, which is usually the EBPro installation directory in

your PC, as shown below.

1: "dtl", "C:\EBPro\EasyConverter /c $(Pathname)"
2: "evt", "C:\EBPro\EasyConverter /c $(Pathname)"

For more information, see “25 Easy Converter”.

26.5.2. Specialized Criteria

The specialized criterion are needed when:

- Upload file to a specific HMI, see listing 2.
- Identify the HMI by HMI name, see listing 3.
- Process differently to different Data Sampling, see listing 4.

(This can only be used for Data Sampling file with the file name “voltage”.)

The 3rd argument (“*”) indicates this criterion accepts the Data Sampling files that meet the criterion from any HMI. Users can also change the 3rd argument to “192.168.1.26”, “192.168.1.*”, or HMI name, etc. for narrowing the range of the target HMI.

Listing 2. Specialized Criterion for the HMI IP: 192.168.1.26

1: "dtl", "EasyConverter /c $(Pathname)", "192.168.1.26"

Listing 3. Specialized Criterion for HMI name: Weintek_01

1: "dtl", "EasyConverter /c $(Pathname)", "Weintek_01"

Listing 4. Specialized Creterion for Data Sampling file name: Voltage

1: "dtl", "EasyConverter /s Voltage.lgs $(Pathname)", "*", "Voltage"

26.5.3. The Format of a Convert Batch File

The following explains the arguments in a criterion.

File Type Command (line) HMI IP / Name Condition 1 Condition 2

- File Type
  This argument specifies the extension name of the uploaded file in this criterion.
  (e.g. “.dtl” for Data Sampling history files, “.evt” for Event Log history files)
- Command (line)
  The command EasyPrinter sends to a console window if the uploaded file meets the criterion.
- HMI IP / Name
  This argument specifies the HMI that meets the criterion.
- Condition 1
  This argument specifies the folder name of the Data Sampling files that meet the criterion.

This is not effective to other format of files.

- Condition 2

  Not used (Reserved for future use).

26.5.4. The Order of Examining Criterion

EasyPrinter examines criterion in descending order every time a file is uploaded. Once the file meets a criterion, it stops the examination and starts over for the next file. Therefore, users should place the criterion with a wider range downward in the Convert Batch File and place the more specific criteria upward. For example:

"evt", "EasyConverter /c $(Pathname)"
"dtl", "EasyConverter /c $(Pathname)"
"dtl", "EasyConverter /c $(Pathname)", "192.168.1.26"
"dtl", "EasyConverter /c $(Pathname)", "my_HMI_01"
"dtl", "EasyConverter /c $(Pathname)", "my_HMI_02"
"dtl", "EasyConverter /s Voltage.lgs $(Pathname)", "*", “Voltage"

The correct order of examination would be: (from bottom to top)

"dtl", "EasyConverter /s Voltage.lgs $(Pathname)", "*", “Voltage"
"dtl", "EasyConverter /c $(Pathname)", "my_HMI_02"
"dtl", "EasyConverter /c $(Pathname)", "my_HMI_01"
"dtl", "EasyConverter /c $(Pathname)", "192.168.1.26"
"dtl", "EasyConverter /c $(Pathname)"
"evt", "EasyConverter /c $(Pathname)"

27. Simulation in EasySimulator / cMT Viewer

This chapter explains how to use EasySimulator / cMT Viewer to run on-line / off-line simulation without having to start the simulator from EasyBuilder Pro.

27.1. EasySimulator 27-2
27.2. cMT Viewer 27-3

27.1. EasySimulator

27.1.1. Overview

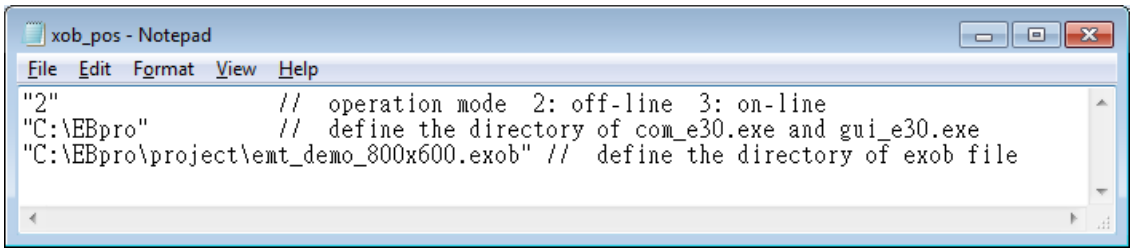
EasySimulator allows users to run on-line or off-line simulation for eMT/iE/XE/mTV models without having to start the simulator from EasyBuilder Pro. To do this, please prepare the required files and follow the steps to setup EasySimulator.

27.1.2. Steps to setup EasySimulator

1. Prepare the following required files and put them into the same folder.

- [driver] → [win32]
- com_e30.exe
- EasySimulator.exe
- gui_e30.exe
- sqlite3.dll
- xob_pos.def
- libcurl.dll
- libeay32.dll
- MFC71.dll
- mosquitto.dll
- mosquittopp.dll
- pthreadVC2.dll
- ssleay32.dll

2. Open **xob_pos.def** by using a text editing tool (e.g. Notepad) and edit the contents.



Line number	Description
1	"2" run an Off-line Simulation; "3" run an On-line Simulation.
2	The directories of the relevant files. (e.g. com_e30.exe, gui_e30.exe, EasySimulator.exe…etc.)
3	The full path of the .exob file.

3. Double click on EasySimulator.exe to start a simulation.
4. On-line / Off-line Simulation is displayed on the screen.

Note

- The required files can be found in the EasyBuilder Pro installation directory. Please install EasyBuilder Pro first then copy the required files to your PC.
- If EasySimulator.exe is not activated, please check if the installation directory is correct.
- If the “Failed to open project file: No such file or directory” dialog box appears, this indicates that there is an error of the .exob file path, please check again.

27.2. cMT Viewer

27.2.1. Overview

cMT Viewer allows users to run on-line / off-line simulation for cMT/cMT X models without having run EasyBuilder Pro itself.

27.2.2. Steps to setup cMT Viewer

1. Use a third-party software, e.g. WinRAR, to decompress cxob file twice and then a project file will be generated in the mt8000 folder.
2. Open Command Prompt and then use the commands below.

On-line Simulation

```
cMTViewer.exe -ns "path/to/project/file" -allinone
```

Off-line Simulation

```
cMTViewer.exe -fs "path/to/project/file" -allinone
```

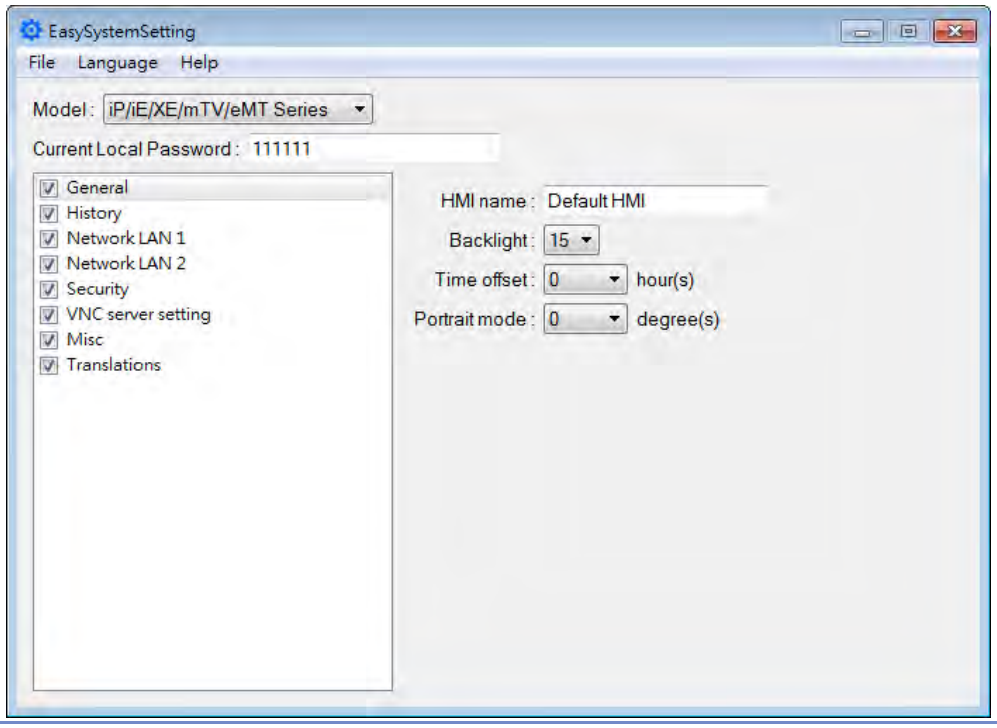
28. EasySystemSetting

This chapter explains how to use EasySystemSetting.

28.1. EasySystemSetting 28-2
28.2. Configuring EasySystemSetting..... 28-5

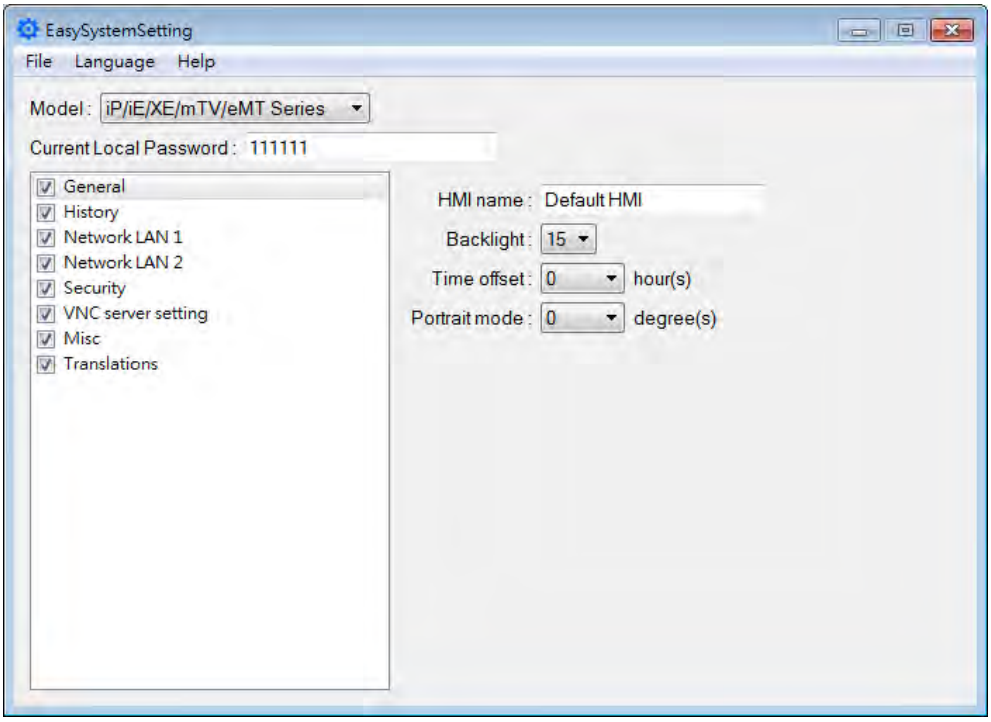
28.1. EasySystemSetting

EasySystemSetting allows updating hardware system settings by using a SD card or USB drive. EasySystemSetting can be opened by clicking EasySystemSetting.exe under EasyBuilder Pro installation directory, or clicking [Tool] » [SystemSetting] Editor in EasyBuilder Pro. The feature is available for HMI OS version 20131106 or later.



Setting	Description
File	[New] Restore default settings. [Load] Import and edit an existing .conf file. [Save] Export the configured data to a .conf file.
Language	Change the language used in EasySystemSetting.
Help	[About] Shows version information of EasySystemSetting.

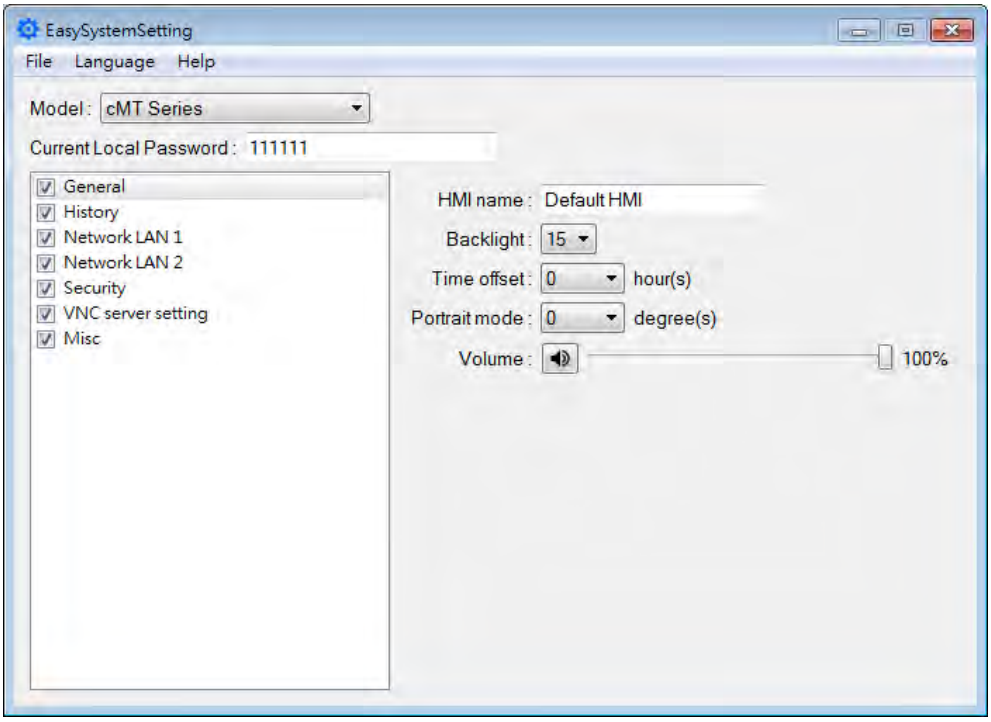
iP/iE/XE/mTV/eMT Series



Since the description of system settings can be found in Chapter 4, the following introduces only the General tab.

Setting	Description
Current Local Password	Please enter the password used to log in System Parameter Settings.
General	[HMI name] Enter HMI name. The name should only contain [A~Z], [a~z], [0~9], [-] and [.]. [Back light] Adjust LCD backlight brightness. [Time offset] Set the HMI RTC with offset. For example, if the current RTC time is 15:00:00, and the time offset is set to -3, the updated time will be 12:00:00. [Portrait mode] Set the display mode.

cMT/cMT X Series



Since the description of system settings can be found in Chapter 4, the following introduces only the General tab.

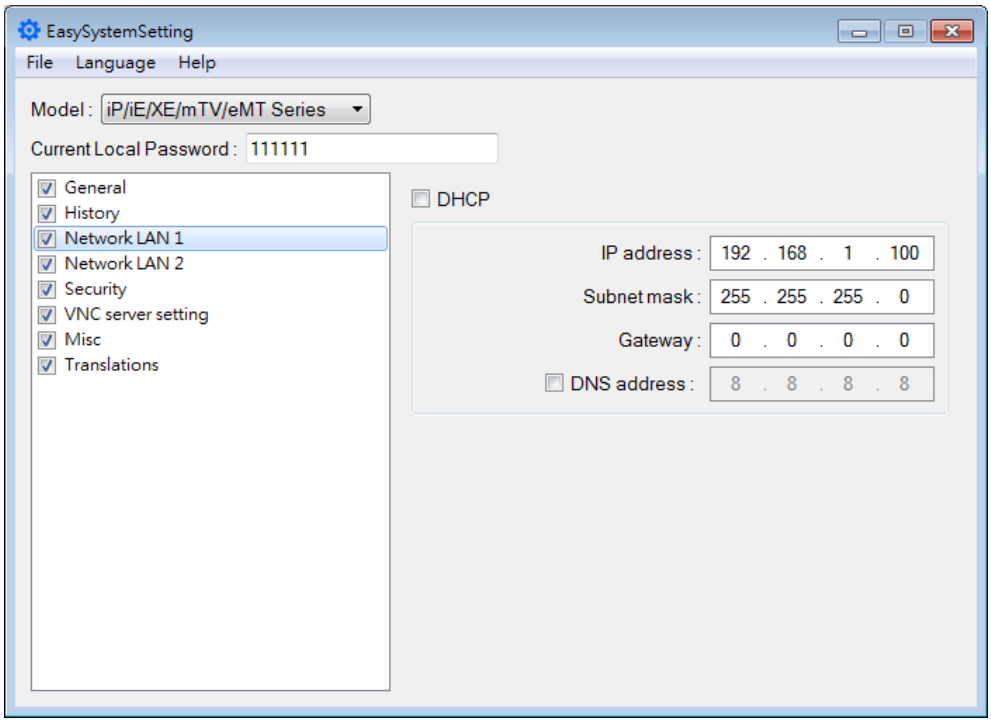
Setting	Description
Current Local Password	Please enter the password used to log in System Parameter Settings.
General	[HMI name] Enter HMI name. The name should only contain [A~Z], [a~z], [0~9], [-] and [.]. [Back light] Adjust LCD backlight brightness. [Time offset] Set the HMI RTC with offset. For example, if the current RTC time is 15:00:00, and the time offset is set to -3, the updated time will be 12:00:00. [Portrait mode] Set the display mode. [Volume] Adjust sound volume.

28.2. Configuring EasySystemSetting

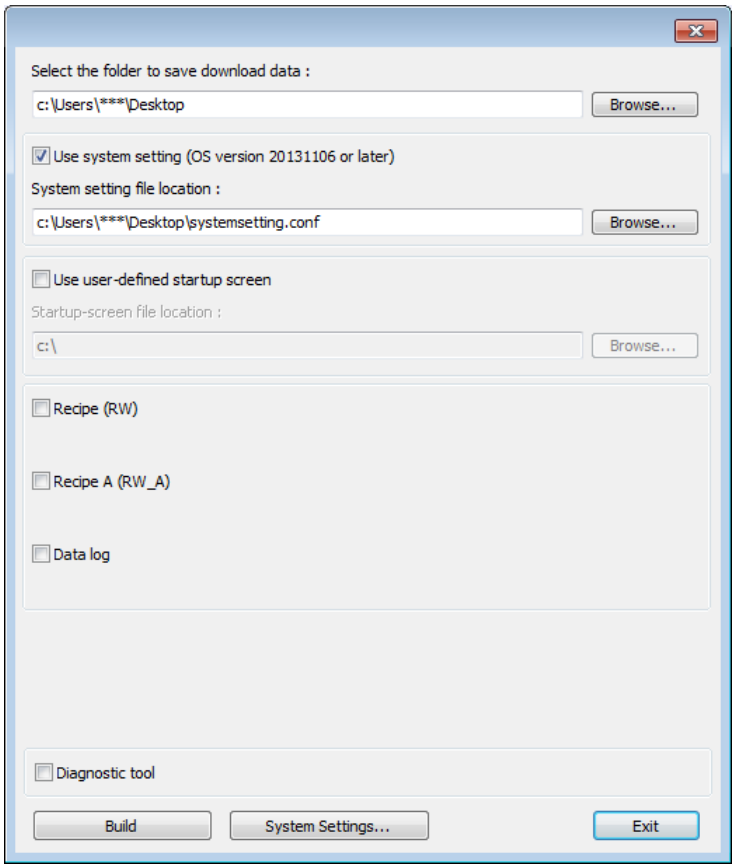
The following explains how to update HMI IP address by using SD card or USB drive.

iP/iE/XE/mTV/eMT Series

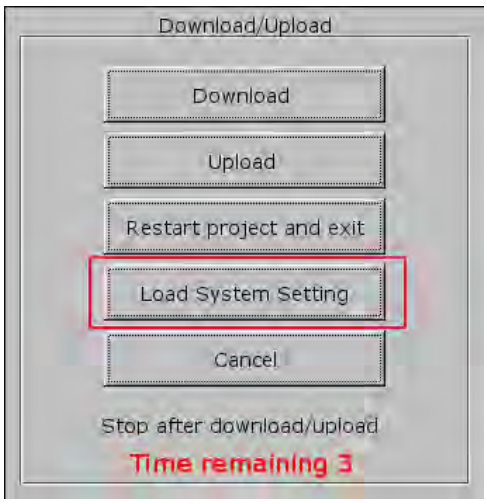
1. Open EasySystemSetting and change the Network settings as shown below.



2. Click [File] » [Save] to generate a “systemsetting.conf” file.
3. Close EasySystemSetting.
4. In EasyBuilder Pro click [Build Download Files], select [Use system setting], and browse for the systemsetting.conf file.



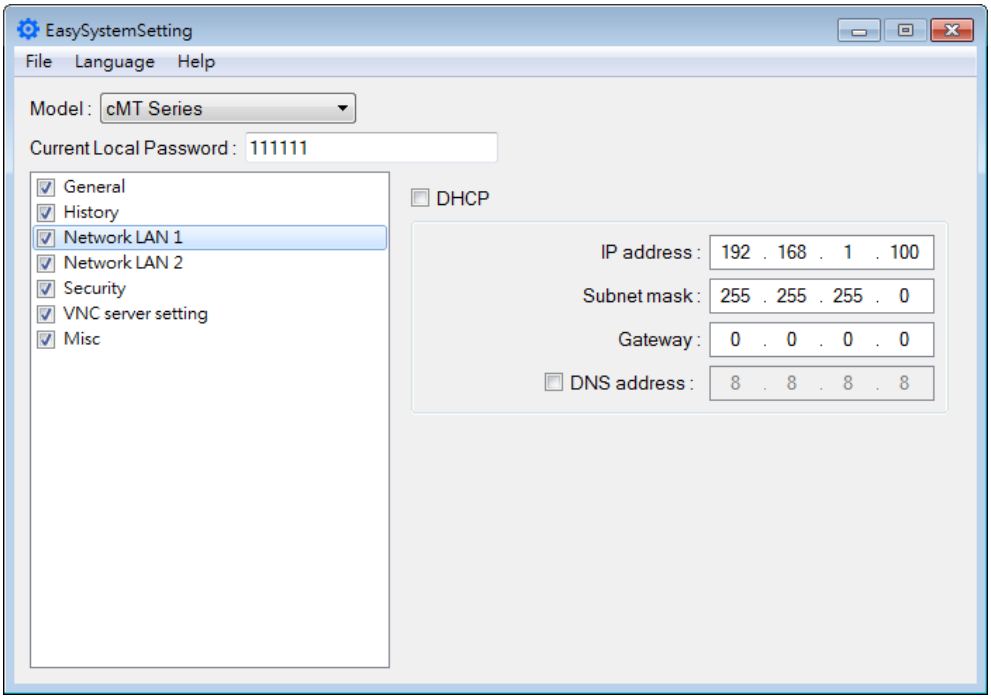
5. Click [Build] to generate the download file to be saved into the external device.
6. Insert the external device that stores the download file to HMI and the Download/Upload dialog box appears.



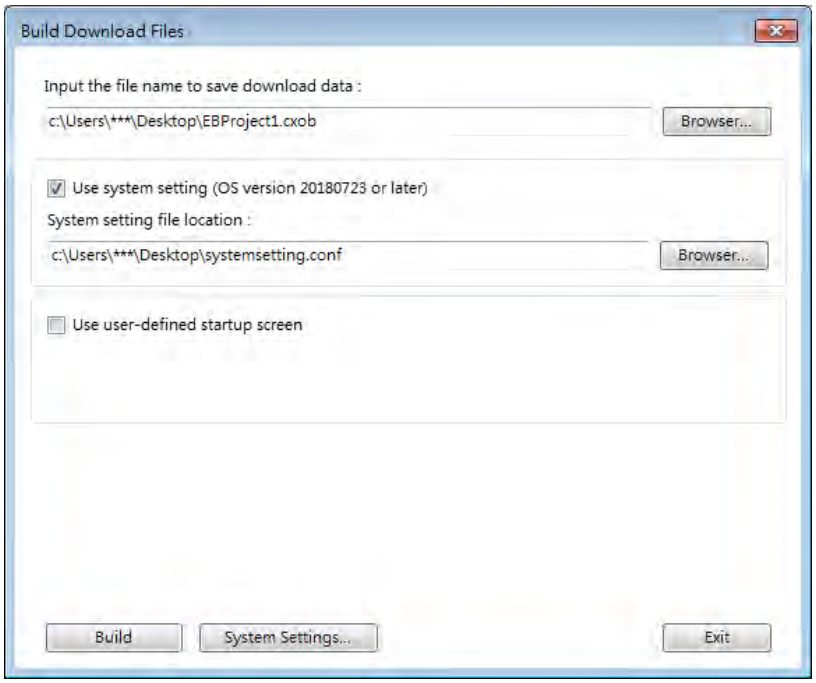
7. Press [Load System setting] and then the [Download Config Settings] message appears. The project file will be updated after finishing system settings.

cMT/cMT X Series

1. Open EasySystemSetting and change the Network settings as shown below.

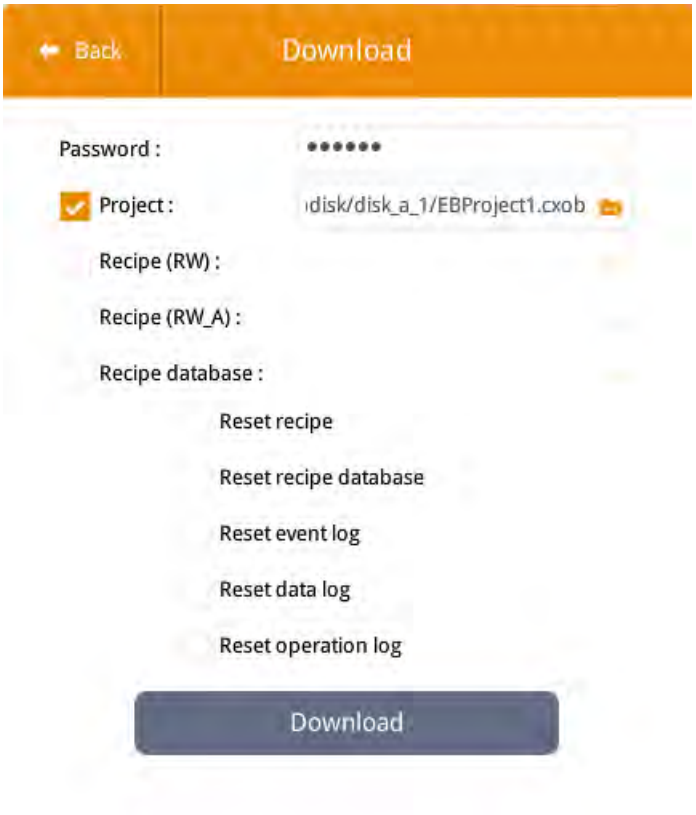


2. Click [File] » [Save] to generate a “systemsetting.conf” file.
3. Close EasySystemSetting.
4. In EasyBuilder Pro click [Build Download Files], select [Use system setting], and browse for the systemsetting.conf file.



5. Click [Build] to generate a compiled file and a system setting file with the same filename (EBProject1.cxob & EBProject1.conf). Save these files in the same directory in an external device.
6. Insert the external device to HMI and the Download/Upload dialog box appears; click

Download. Browse for the project file, enter the password, and then click Download.



Note

- System setting file can be downloaded using Utility Manager or EasyBuilder Pro.

29. Pass-through

This chapter explains how to set up Pass-through mode.

29.1. Overview 29-2
29.2. Serial Pass Through Ethernet Mode 29-2
29.3. Serial Pass-Through COM Port Mode 29-7
29.4. Pass-through Control 29-9
29.5. SIEMENS S7-200 PPI and S7-300 MPI Pass-through Settings 29-10
29.6. Ethernet Pass-Through 29-15

29.1. Overview

The Pass-through feature allows PC applications to control PLC via HMI. In this case the HMI is an adaptor.

The Serial Pass-Through feature provides two modes:

- Ethernet
- COM port

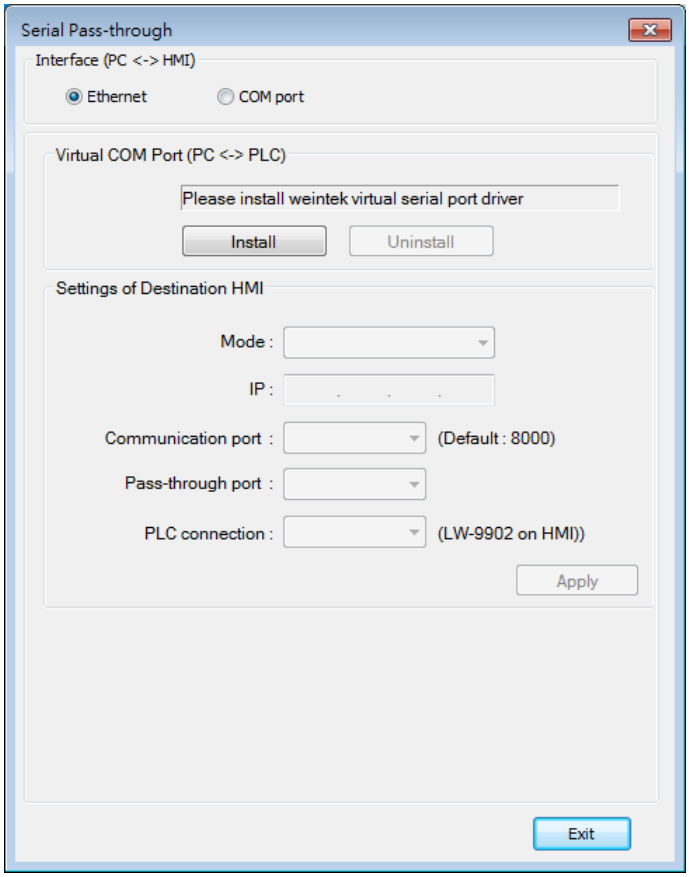
Click [Pass-through] in Utility Manager to open the setting dialog box.

29.2. Serial Pass Through Ethernet Mode

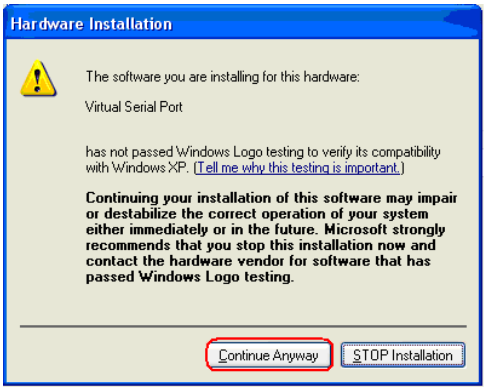
29.2.1. Steps to Install Virtual Serial Port Driver

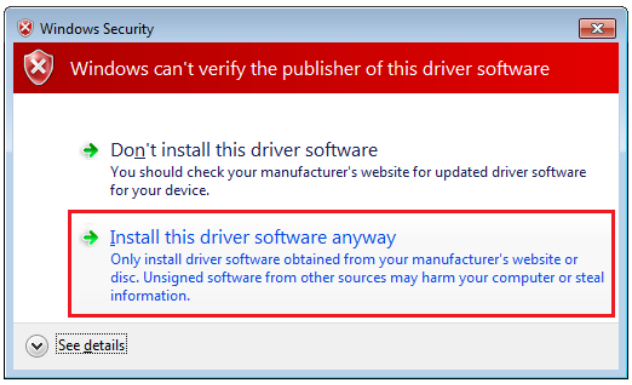
Before using [Ethernet] mode, please check if Weintek virtual serial port driver has been installed. Please always install the latest patch of Windows Update to ensure proper operation of virtual serial port driver.

1. Open Utility Manager to check if the driver has been installed. If it shows [Please install weintek virtual serial port driver], please click [Install].



2. If the dialog below pops up during installation asking for verification, please click [Continue Anyway].

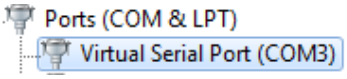




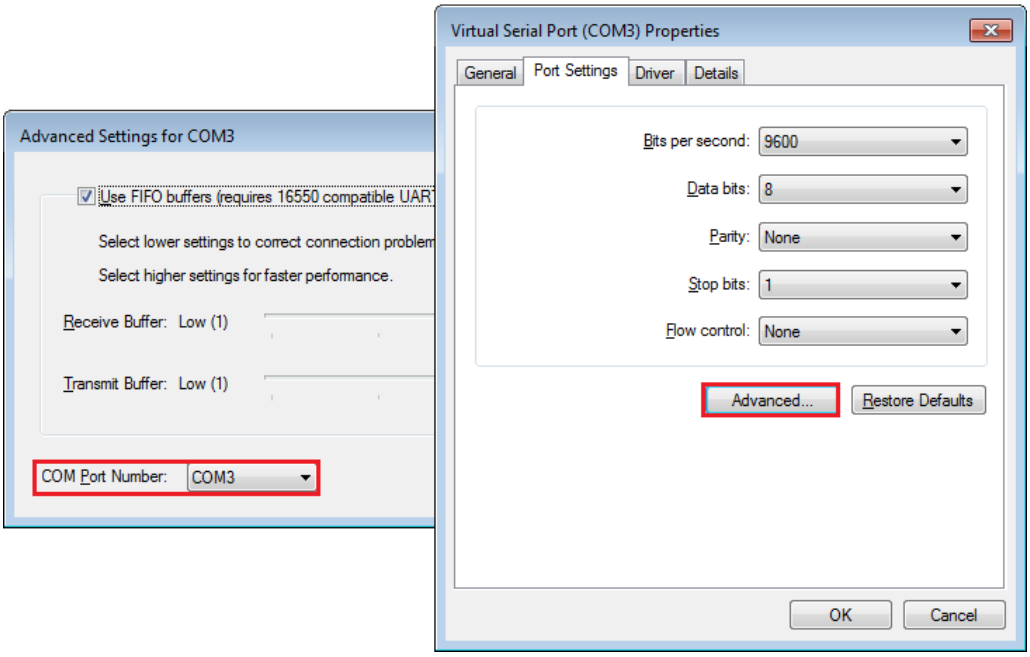
3. When finished, the [Virtual COM Port (PC <-> PLC)] field displays the virtual COM port used.

29.2.2. Steps to Change the Virtual Serial Port Number

1. Open [Device Manager] and find Virtual Serial Port.

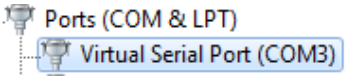


2. To change to another COM Port Number, double-click Virtual Serial Port and open [Port Settings] » [Advanced].

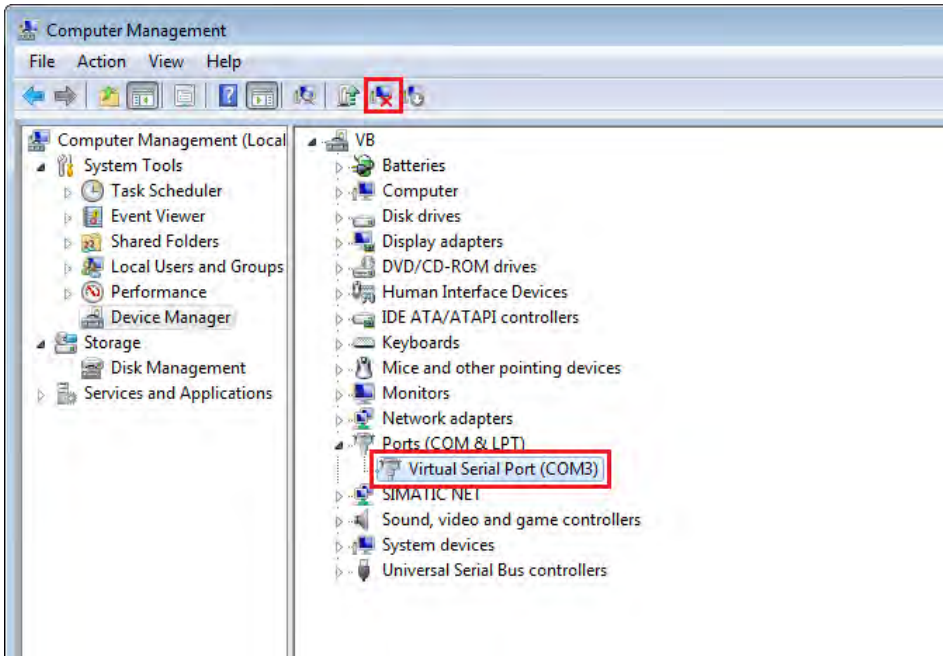


29.2.3. Steps to Uninstall the Virtual Serial Port

1. Open [Device Manager] and find Virtual Serial Port.



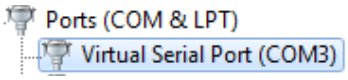
2. To uninstall the Virtual Serial Port, select it and click the [Uninstall] button in the Device Manager toolbar.



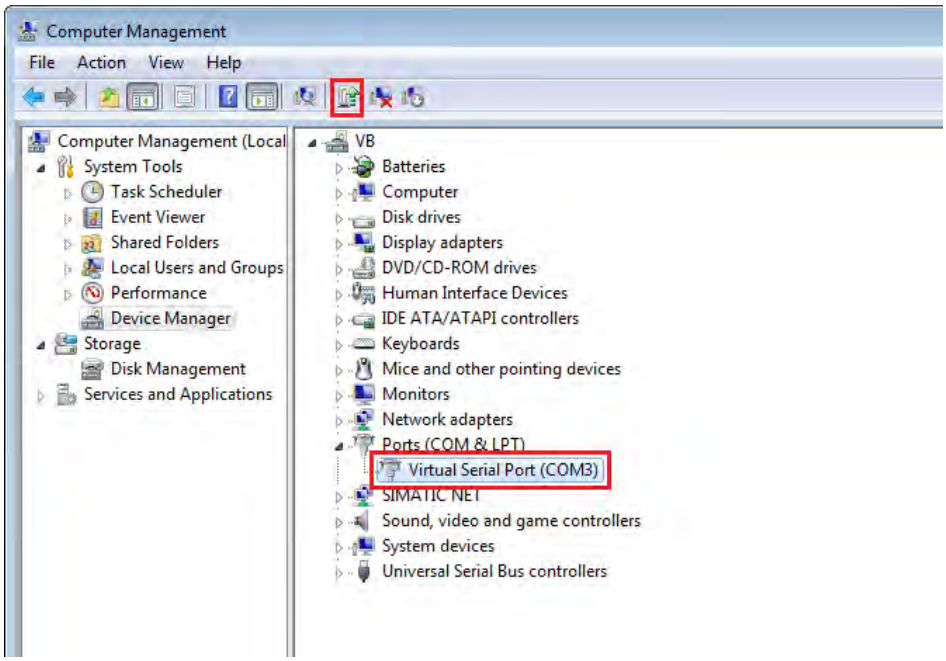
3. Click [OK] to uninstall this Virtual Serial Port.

29.2.4. Steps to Update Virtual Serial Port Driver

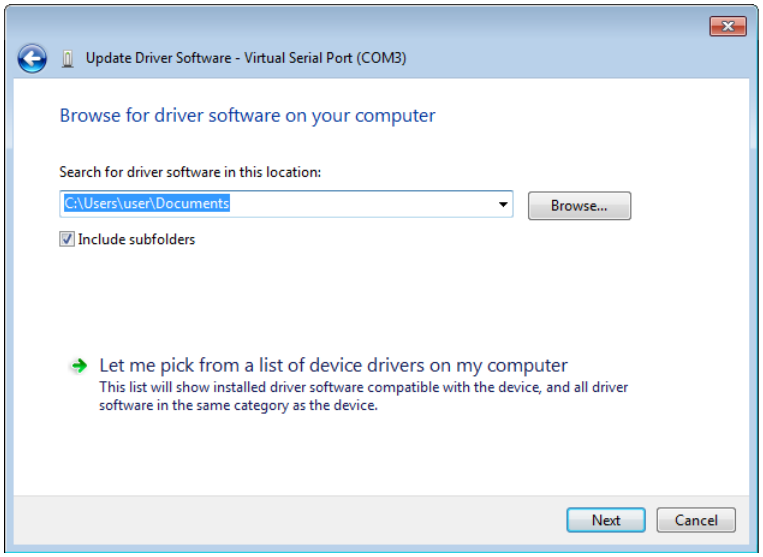
1. Open [Device Manager] and find Virtual Serial Port.



2. To update virtual serial port driver software, select Virtual Serial Port and click the [Update Driver Software] button in the Device Manager toolbar.



3. Browser for the directory of the driver, and then click [Next] to update the driver.



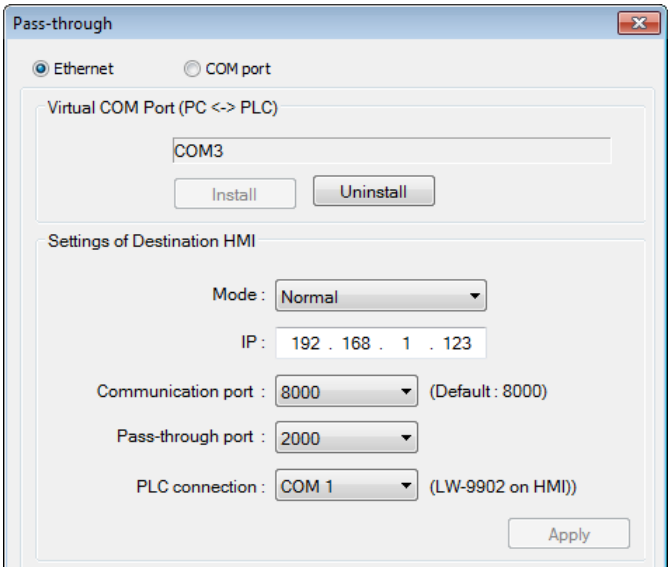
29.2.5. Settings of Ethernet mode

After installing the virtual serial port driver, follow the steps to use Ethernet mode of pass-through feature.

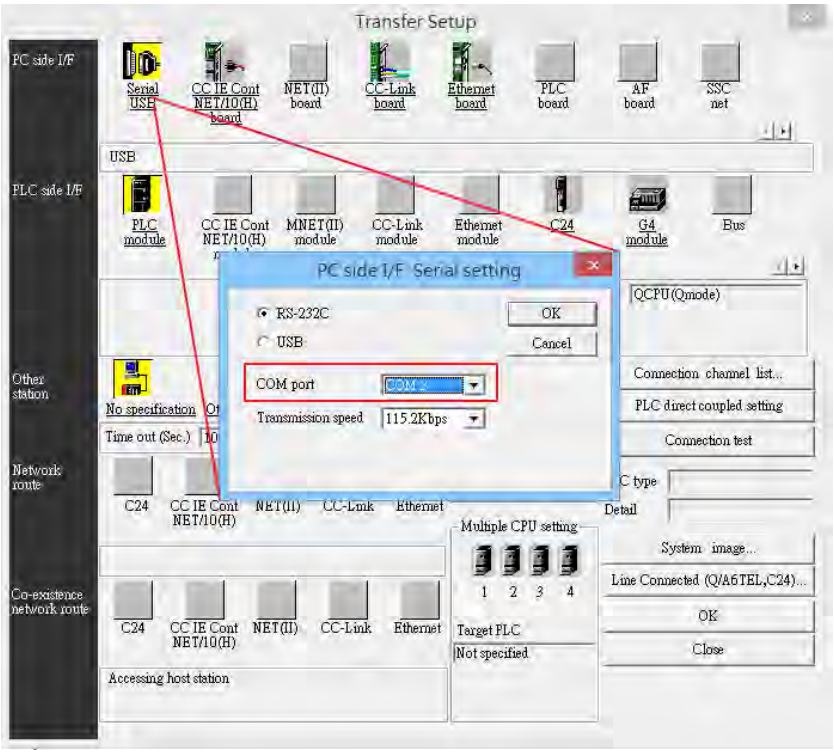
1. Set the IP address of the HMI connected with PLC.

2. Set the communication port and the serial port that connects HMI with PLC.

3. Click [Apply], to apply the settings.



4. When running PC application, set COM port to the used virtual serial port. For example, in Mitsubishi application, if the virtual serial port is COM 3, set [PC side I/F Serial setting] » [COM port] to COM 3.



5. With the correct configurations, upon execution of PLC application on PC, HMI will be

automatically switched to Pass-through mode. During Pass-through, the PLC is controlled by PC via the virtual serial port. Pass-through mode will be turned off when the application ends.

29.3. Serial Pass-Through COM Port Mode

[Source COM Port] The port connects HMI with PC.

[Destination COM Port] The port connects HMI with PLC.

To use [COM port] mode of Pass-through, please set the properties of Source COM Port and Destination COM Port correctly.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

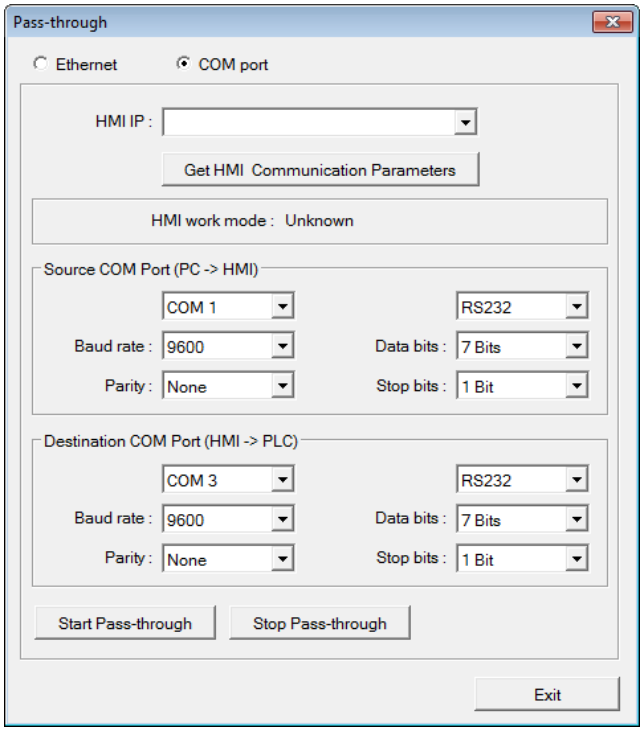
29.3.1. Settings of COM Port Mode

There are two ways to enable [COM port] mode of Pass-through feature.

- Using Utility Manager.
- Using system registers.
  LW-9901: pass-through source COM port (1 ~ 3: COM 1 ~ COM 3)
  LW-9902: pass-through destination COM port (1 ~ 3: COM 1 ~ COM 3)

29.3.2. Using Utility Manager

1. Click [Serial Pass-Through] button in Utility Manager to set the communication parameters as shown in the following figure.



Setting	Description
HMI IP	HMI IP address.
Get HMI Communication Parameters	Reads the settings of Source and Destination COM port. Click this button to update the communication parameters.
Source COM Port (PC->HMI) / Destination COM Port (HMI->PLC)	The communication parameters of Source and Destination COM Port are displayed. The settings will be applied when [Start Pass-through] is clicked.
Baud rate / Data bits / Parity / Stop bits	Source and Destination COM Port parameters should be set to be same. Since [Source COM Port] connects PC, select RS-232 mode in most situations; [Destination COM Port] connects PLC, so the setting depends on the PLC type, and can be one of RS-232, RS-485 2W, or RS-485 4W.

Note

- When pass-through feature is no longer needed, click [Stop Pass-through] to stop it. HMI will then resume communication with PLC.

There are three work modes of HMI.

Mode	Description
Unknown	The work mode before reading the settings of HMI.
Normal	The work mode after reading the settings of HMI. The HMI does not accept any data form the Source COM Port.
Pass-through	The work mode is "Pass-through." the PC connected via Source COM Port can control the PLC connected via Destination COM Port.

29.3.3. Using System Registers

Another way of enabling pass-through is by writing to LW-9901(Source COM port) and LW-9902 (Destination COM port).When the values of LW-9901 and LW-9902 match the conditions below, HMI will start Pass-through automatically:

- The values of LW-9901 and LW-9902 are 1 to 3 (1 to 3: COM 1 to COM 3).
- The values of LW-9901 and LW-9902 are different.

To change the communication parameters, just change the value in the related registers and set ON the appropriate registers: [LB-9030: update COM 1 communication parameters], [LB-9031: update COM 2 communication parameters] and [LB-9032: update COM 3 communication parameters]. HMI will then update the settings.

Note

- To stop Pass-through, change the values of LW-9901 and LW-9902 to 0.
- For some models (eMT, cMT…etc), COM1 RS232 RTS/CTS share the same pins (Pin no. 7 and 8) with COM3 RS232 RX/TX. When COM3 is not used in the device list, Pin no. 7 and 8 is used for COM1 RS232. To activate Pin no. 7 and 8 for COM3 RS232 RX/TX, please add COM3's Master Slave driver or Free Protocol driver into the device list in System Parameter Settings.

29.4. Pass-through Control

Generally speaking, during pass-through, HMI closes its connection with the PLC until the pass-through mode ends. However, certain PLC drivers allow communications between HMI and PLC in pass-through mode.

To see whether a driver supports concurrent communication, see "PLC Connection Guide".

Pass-through control is controlled by LW-9903. The following table shows valid LW-9903 values and their features.

LW-9903	Description
0 (Default)	Normal Mode. Communications between HMI and PLC in pass-through mode is allowed.
2	Stop Mode. No communications between HMI and PLC in pass-through mode

Note

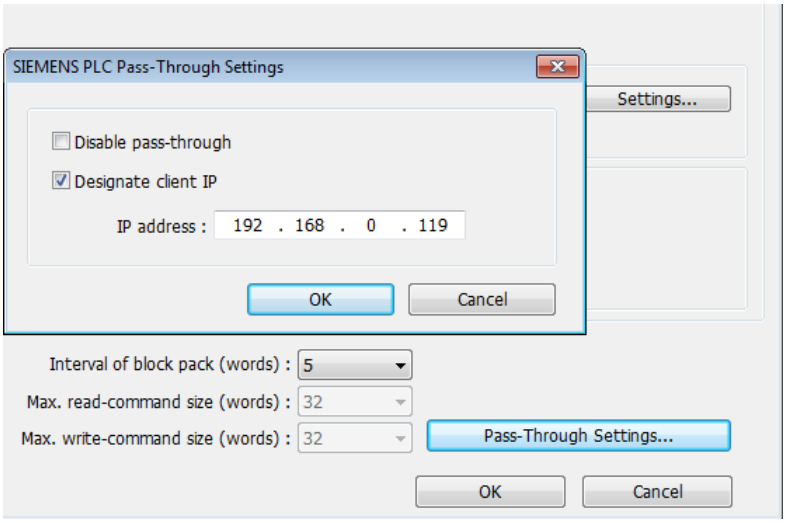
- Due to speed limitation, users may wish to set LW-9903 to 2 to enhance the speed of program download/upload in pass-through mode.

29.5. SIEMENS S7-200 PPI and S7-300 MPI Pass-through Settings

EasyBuilder Pro supports SIEMENS S7-200 PPI and S7-300 MPI pass-through feature.

29.5.1. EasyBuilder Pro Settings

Launch EasyBuilder Pro, go to [System Parameter Settings] » [Device list], and then add SIEMENS S7-200 PPI or S7-300 MPI device. Click [Pass-Through Settings] and the following dialog box appears.

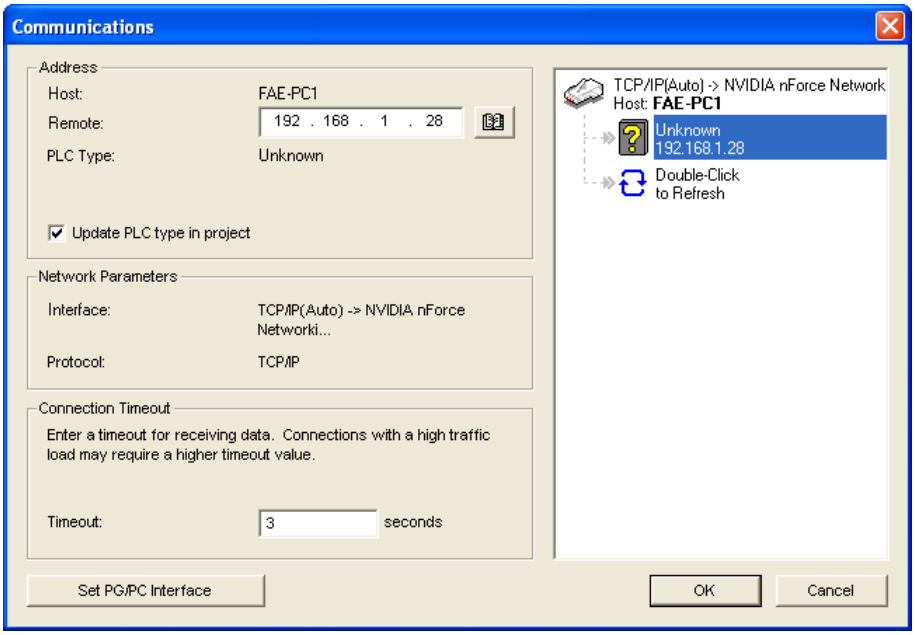


Setting	Description
Disable pass-through	Select this check box to disable pass-through mode. By default this check box is not selected.
Designate client IP	Designate client HMI IP address used in pass-through mode.

29.5.2. S7-200 PPI Connection

Confirm that the HMI used in pass-through communication is started and connected to the

network. Launch STEP 7 Micro/Win, open [Communications] dialog box, and then search for the HMI IP address. Connect the HMI to communicate.

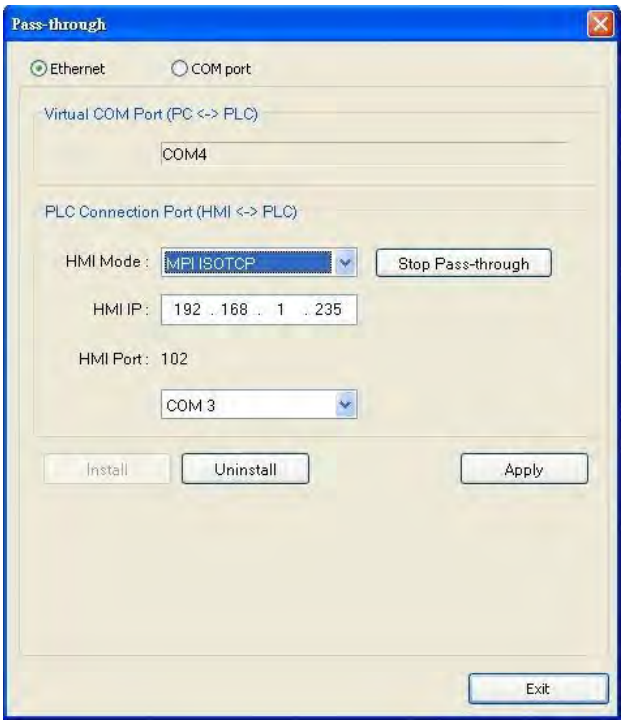


29.5.3. S7-300 MPI Connection

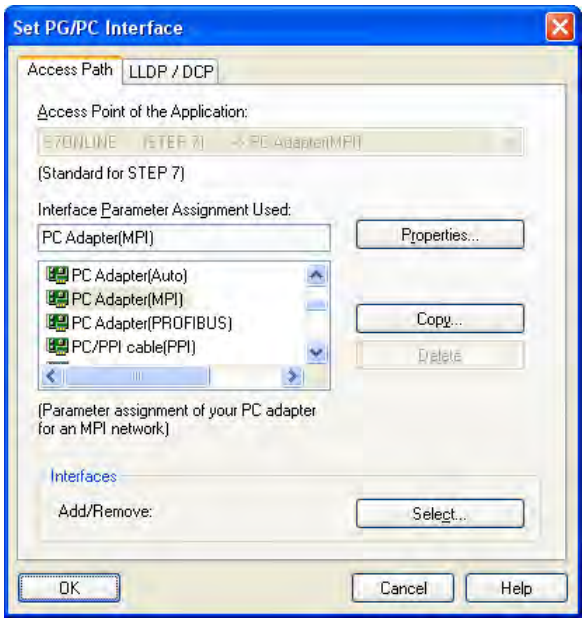
Connect via virtual COM port or Ethernet.

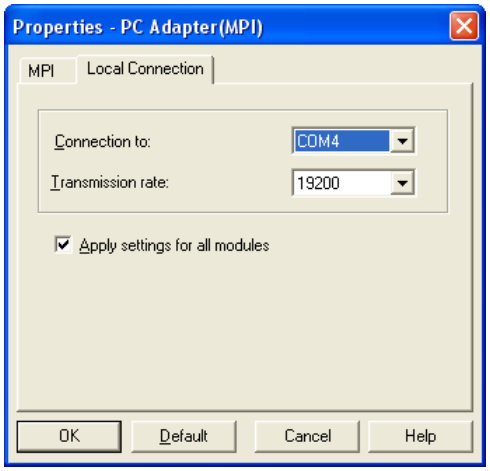
29.5.3.1. Virtual COM Port

1. In Utility Manager run [Pass-Through], in [HMI Mode] select “MPI ISOTCP” to install virtual serial port driver. Set the HMI IP address and the COM port that connects PLC, and then start Pass-through.



2. In STEP 7, go to [Option] » [Set PG/PC Interface]. Confirm that the interface used is "PC Adapter(MPI)", and then click [Properties]. Select the same COM port as the virtual serial port. In the example COM 4 is used.

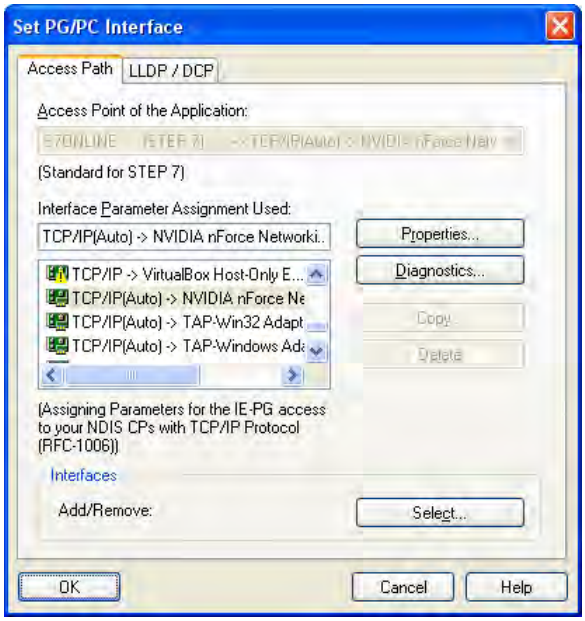




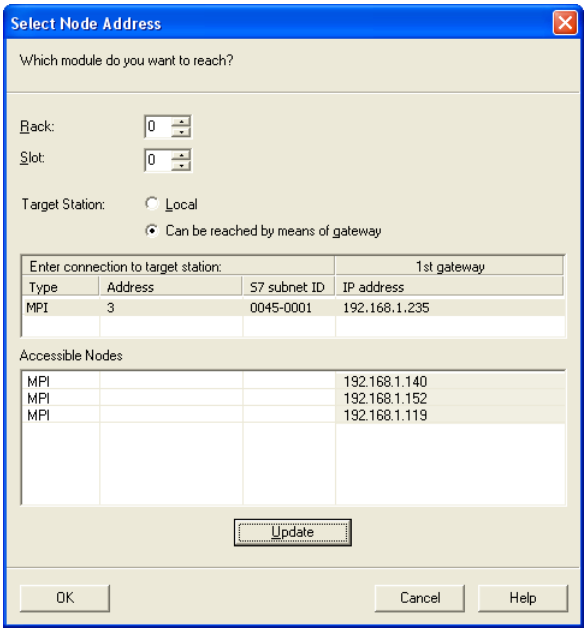
3. When finished, STEP 7 can be used to Upload / Download PLC program via HMI.

29.5.3.2. Ethernet

1. In STEP 7 go to [Option] » [Set PG/PC Interface]. As shown in the following figure, select "TCP/IP(Auto) -> *the name of the network interface card*".



2. Go to [PLC] » [Update station to PG], in [Target Station] select [Can be reached by means of gateway]. From left to right columns enter MPI, PLC station number, S7 Subnet ID, and HMI IP address. When finished, S7 can upload PLC program to STEP 7 via HMI.



29.5.4. Registers of SIEMENS Pass-through

System registers from LW-10850 to LW-10864 are used to set or indicate pass-through status of SIEMENS devices.

For more information see “22 System Registers”.

During pass-through mode, LW-10863 indicates errors and LW-10864 displays error code. The following table lists the error codes, the description of each code, and the possible reason. (The client usually refers to STEP 7 PLC program)

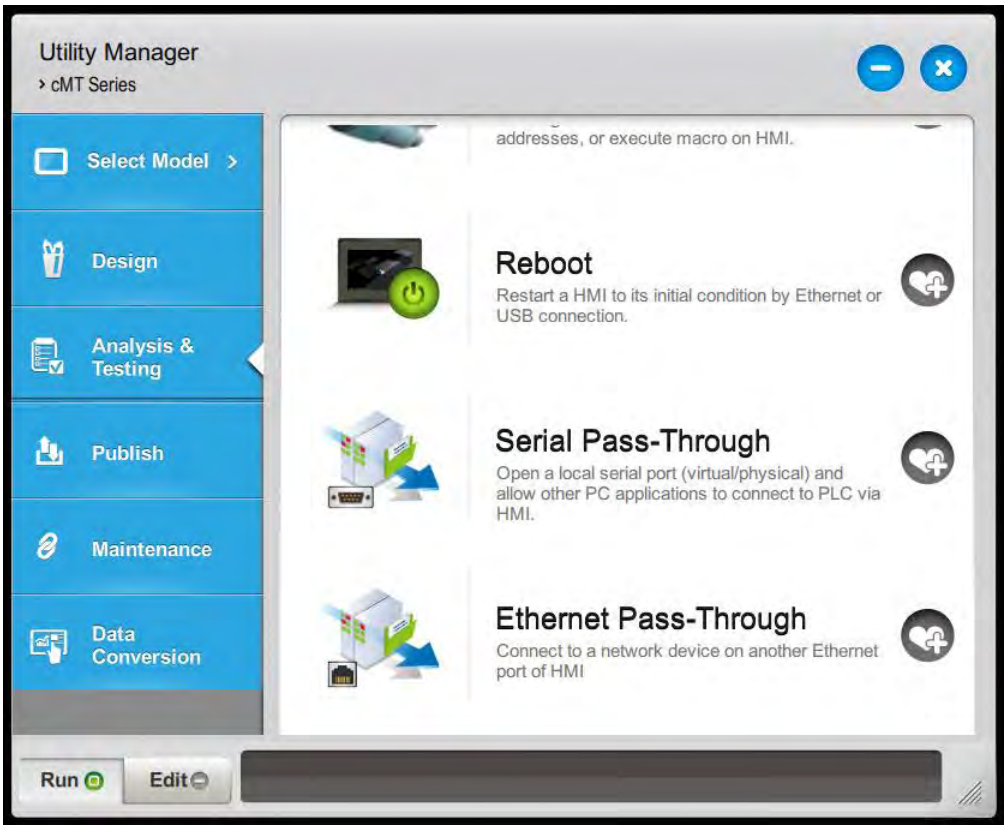
Error Code	Description	Possible Reason
0	Successfully executed	
1	Prohibit client from connecting HMI	HMI is already running pass-through and won’t accept any request from other client.
2	Prohibit client from connecting HMI	When LW-10850 is set to 1, the client IP for connecting HMI is different from the IP specified in LW-10858 ~ LW-10861.
3	Invalid communication protocol	Incorrect setting in LW-10853.
4	Invalid PLC station number	The PLC station number specified in LW-10852 does not exist.
5	Delayed communication	PLC connection failure.
6	Busy communication	PLC does not accept pass-through request, please confirm PLC settings.
7	Invalid pass-through request	Environment setup failure.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

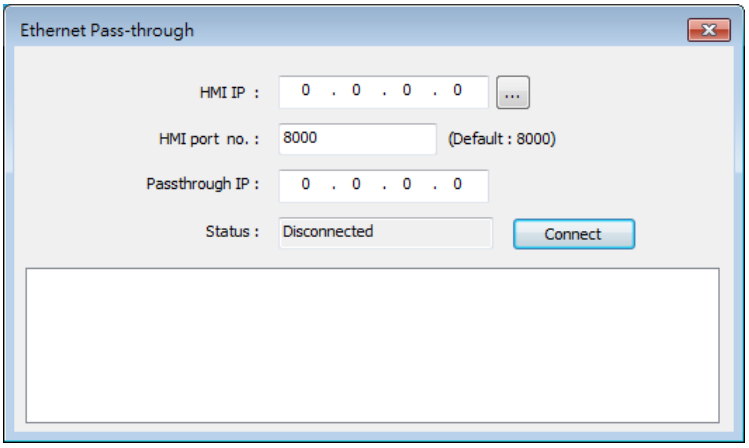
29.6. Ethernet Pass-Through

Ethernet Pass-through feature is available when both PLC and PC connect to HMI via Ethernet. Ethernet Pass-through allows running the application on PC to control PLC through HMI. This feature is only supported on cMT / cMT X Series models.

1. Launch Utility Manager, select cMT / cMT X Series, and then open Analysis & Testing tab in which Ethernet Pass-through can be found.



2. Click [Ethernet Pass-through] and then fill in the communication parameters.



Setting	Description
HMI IP	Enter HMI's IP address.
HMI port no.	Enter the port number used to connect to HMI. By default the port number is 8000.
Passthrough IP	Enter the IP address of the device to be controlled under Ethernet Pass-through mode.
Status	After filling in the communication parameters, click [Connect] to see the connection status.

Note

- One HMI can only use one Ethernet Pass-through IP.
- Ethernet Pass-through feature is disabled when system register LB-9044 (disable remote control) is ON.

30. Project Protection

This chapter explains the settings relevant to project protection.

30.1. Overview 30-2
30.2. EXOB/CXOB Password 30-2
30.3. Decompilation is Prohibited 30-3
30.4. Disable HMI Upload Function 30-3
30.5. Project Key 30-3
30.6. MTP File Password 30-4

30.1. Overview

EasyBuilder Pro provides a number of features to protect a project file.

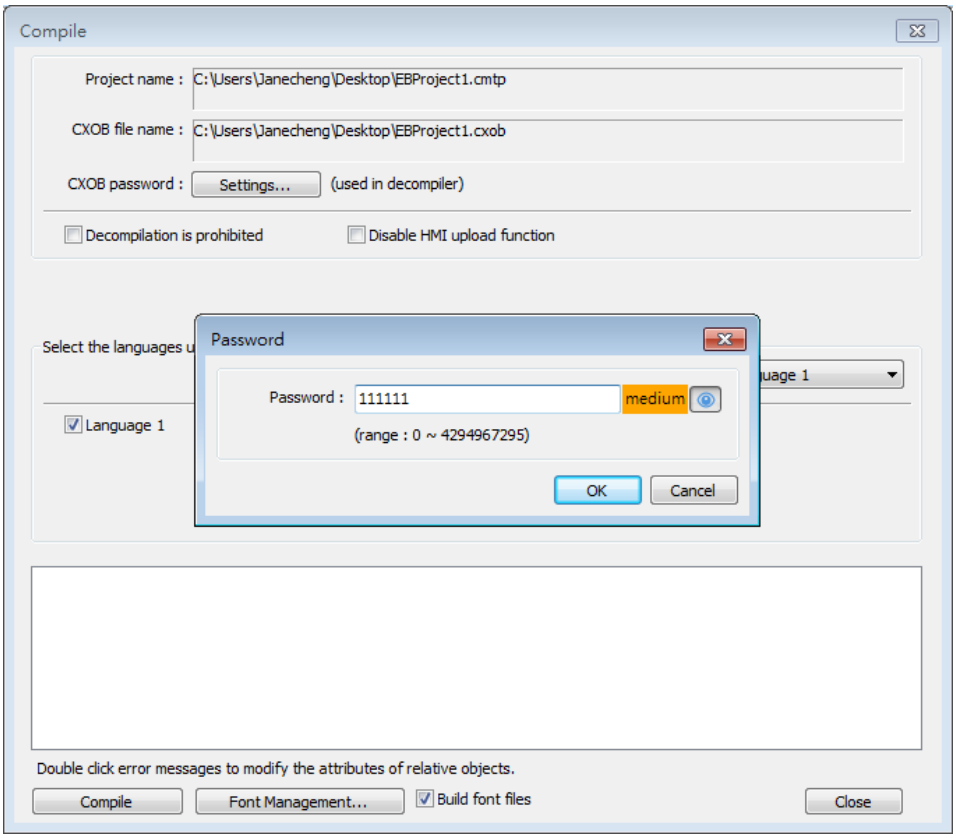
Note

- The protected projects cannot be decrypted by the factory since they are encrypted by users; therefore, please remember your password.

30.2. EXOB/CXOB Password

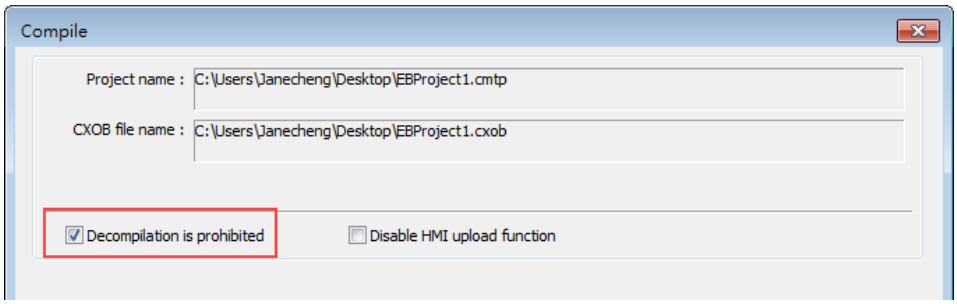
After editing a project (.emtp/.cmtp), users can compile the project to .exob/.cxob format. The .exob/.cxob file can be downloaded to HMI. Password can be set to protect the .exob/.cxob file in [EXOB/CXOB password] when compiling. (Password range: 0 ~ 4294967295)

A password is needed when attempting to decompile the .exob/.cxob file back to .emtp/.cmtp file. If the password is entered incorrectly for three times, please restart EasyBuilder Pro.



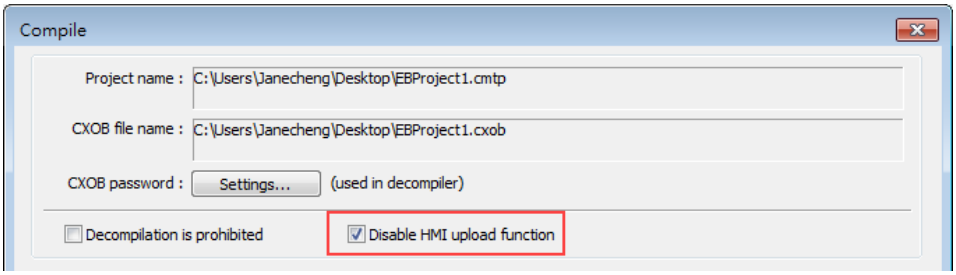
30.3. Decompilation is Prohibited

After a project (.emtp/.cmtp) is edited, users can compile the project to .exob/.cxob format. The .exob/.cxob file can be downloaded to HMI. If [Decompilation is prohibited] check box is selected when compiling, the setting in [EXOB/CXOB password] will be ignored. Furthermore, the .exob/.cxob file cannot be decompiled to .emtp/.cmtp file.



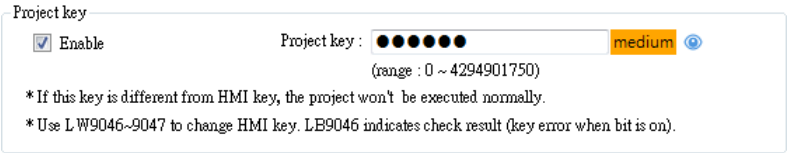
30.4. Disable HMI Upload Function

EasyBuilder Pro provides a system reserved register [LB-9033]. When this register is set ON, the .exob/.cxob file cannot be uploaded. When attempting to upload an .exob/.cxob file with this register set ON, the file obtained after uploading is 0 byte, and cannot be decompiled. Please reboot HMI for the changed setting to take effect.



30.5. Project Key

Projects can be restricted to run on a specific HMI.
The setting is in [System Parameters] » [General] » [Project key].



Enable and set the [Project key] (password range: 0 ~ 4294901750). System registers LW-9046 ~ LW-9047 (32-bit) can be used to set the [HMI key] for HMI. The values in LW-9046 and

LW-9047 cannot be accessed by a remote device. The .exob/.cxob file obtained after compiling can only be executed on HMI when [HMI key] and [Project key] match. If the keys don't match, LB-9046 is set ON. To change [HMI key], please reboot HMI.

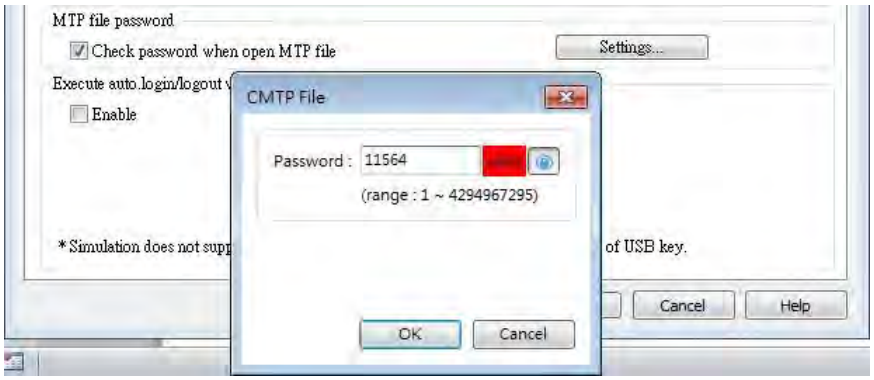
Note

- When **[HMI key]** and **[Project key]** don't match, HMI and PLC cannot communicate.

Click the icon to download the demo project. Please confirm your internet connection.

30.6. MTP File Password

After a project (.emtp/.cmtp) is edited, a password can be set to protect the .emtp/.cmtp file. In [System Parameters] » [Security] tab, enable [Project password] and click [Settings] (password range: 1 ~ 4294967295).
The password will be needed when attempting to open the .emtp/.cmtp file.



Note

- When using "Window Copy" function, if the source file is protected by MTP password, please enter the correct password for the system to execute window copy.

31. cMT Diagnoser

This chapter explains cMT Diagnoser which allows diagnostic operations in cMT / cMT X Series Model.

31.1. Overview 31-2
31.2. How to Start Diagnoser 31-2
31.3. cMT Diagnoser User Interface 31-4
31.4. Additional Notes 31-18

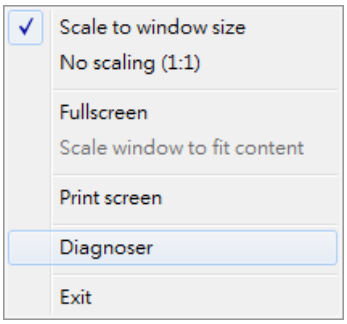
31.1. Overview

cMT Diagnoser can monitor data and status of all objects in the current window when it is connected to an HMI, show the status of individual communication packet, show communication status with MQTT server, and debug macros.

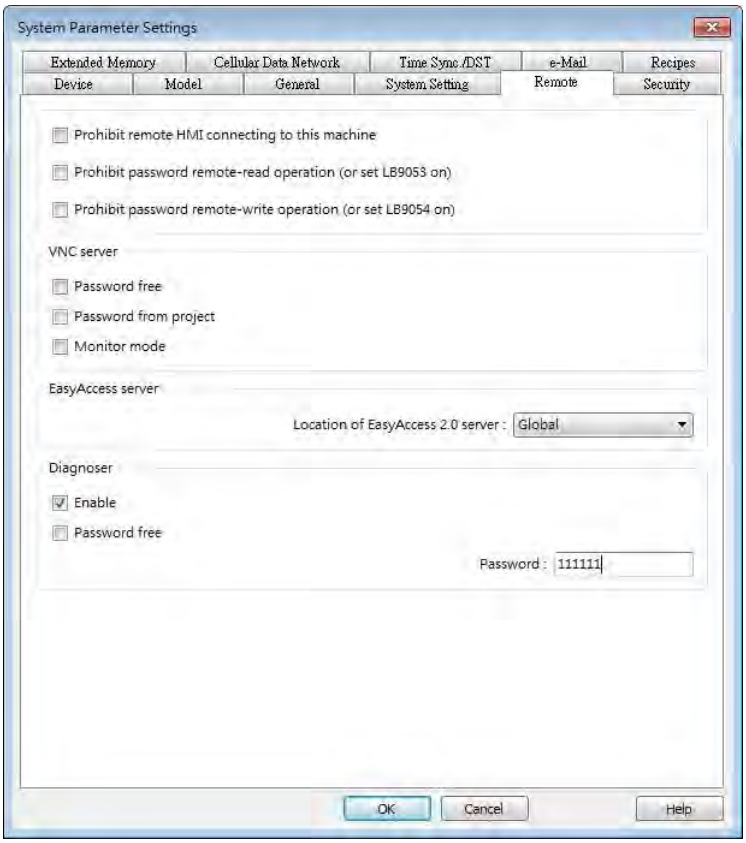
31.2. How to Start Diagnoser

cMT Diagnoser can be used in simulation mode or on HMI. When it is used on HMI, it can be opened from within cMT Viewer, or run an independent program named as cMT Diagnoser be directly opened on HMI.

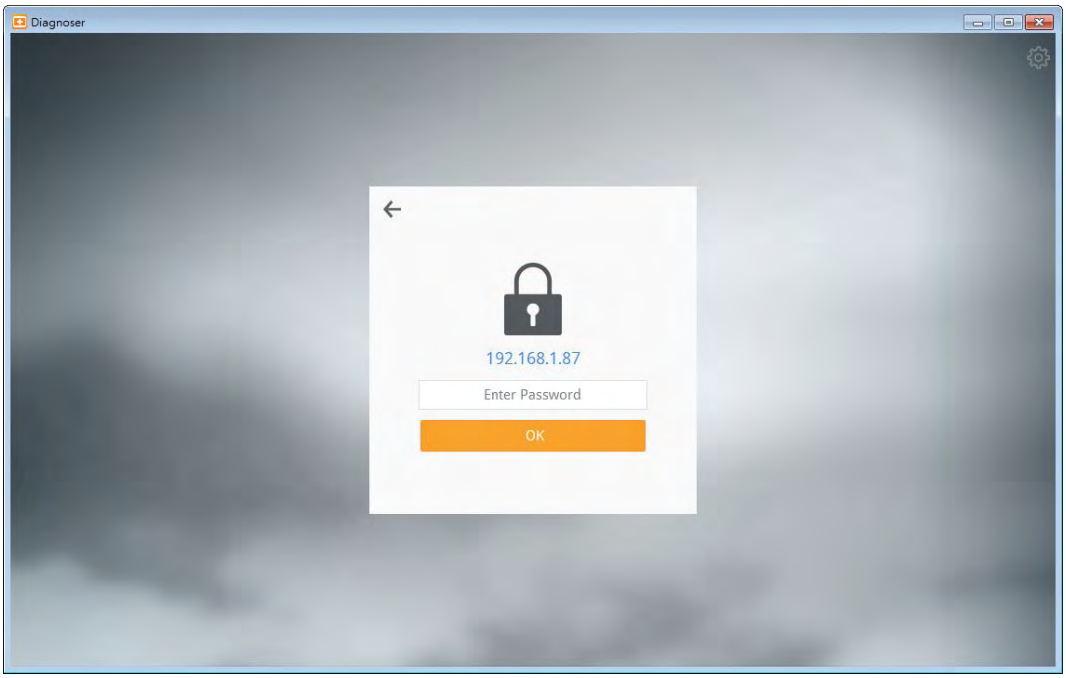
1. In on-line/off-line simulation, cMT Diagnoser can be opened by clicking the right mouse button in cMT Viewer simulation screen and then click Diagnoser.



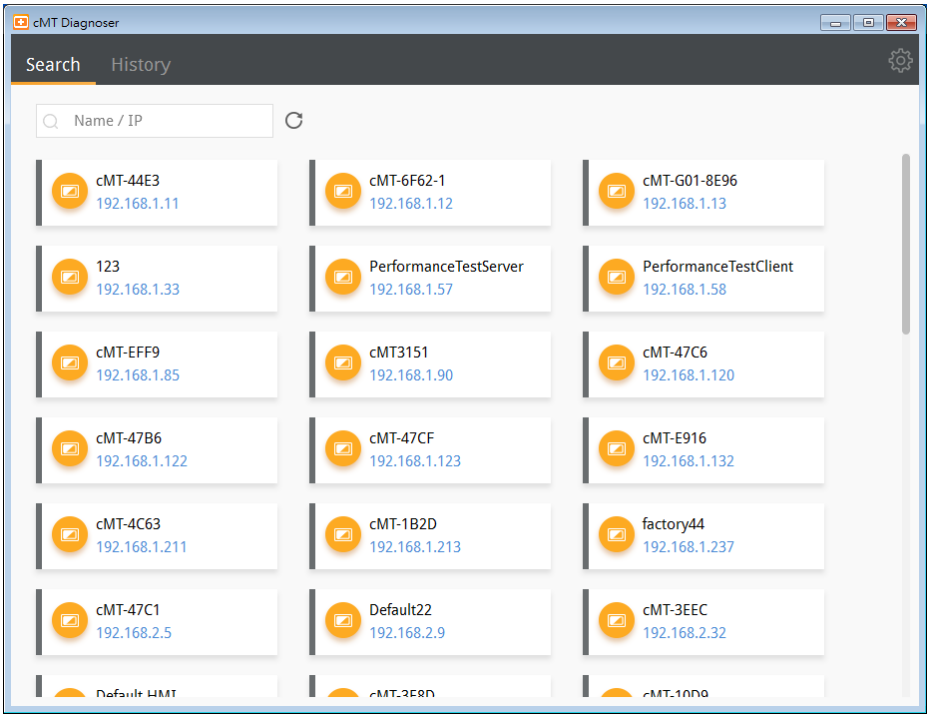
2. To permit connection to HMI from cMT Diagnoser, enable the Diagnoser option which is found in the [System Parameters] » [Remote] tab in EasyBuilder Pro.



After downloading the project to HMI, cMT Viewer can open cMT Diagnoser. If [Password free] is not selected in Remote tab in previous step, a password is required for running cMT Diagnoser.



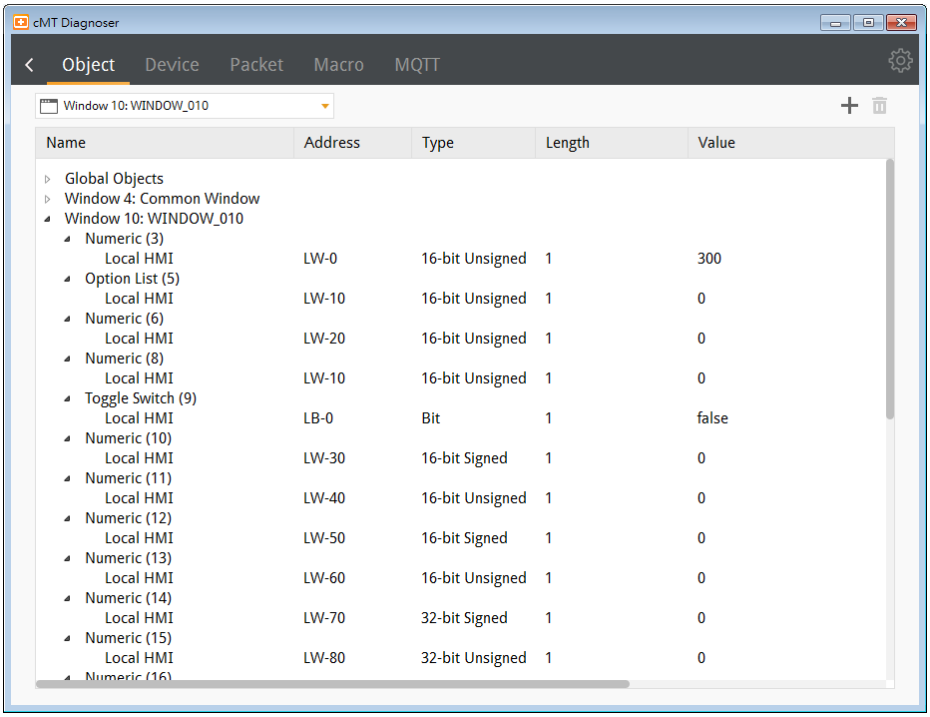
3. Execute cMTDiagnoser.exe, which can be run independently, in EasyBuilder Pro installation directory, and find an HMI to connect to.



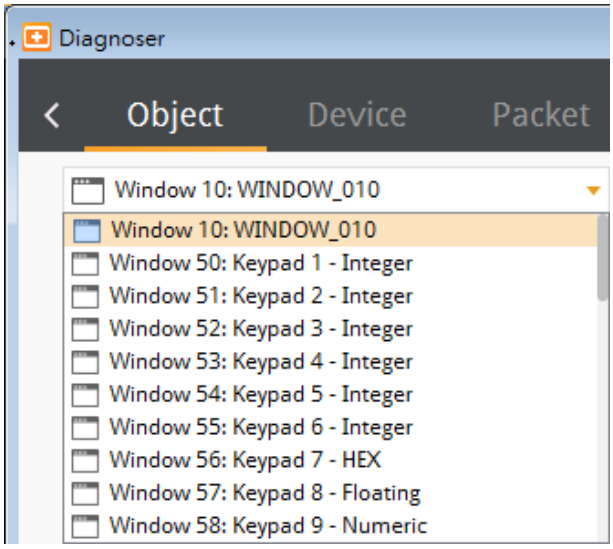
31.3. cMT Diagnoser User Interface

31.3.1. Object Settings

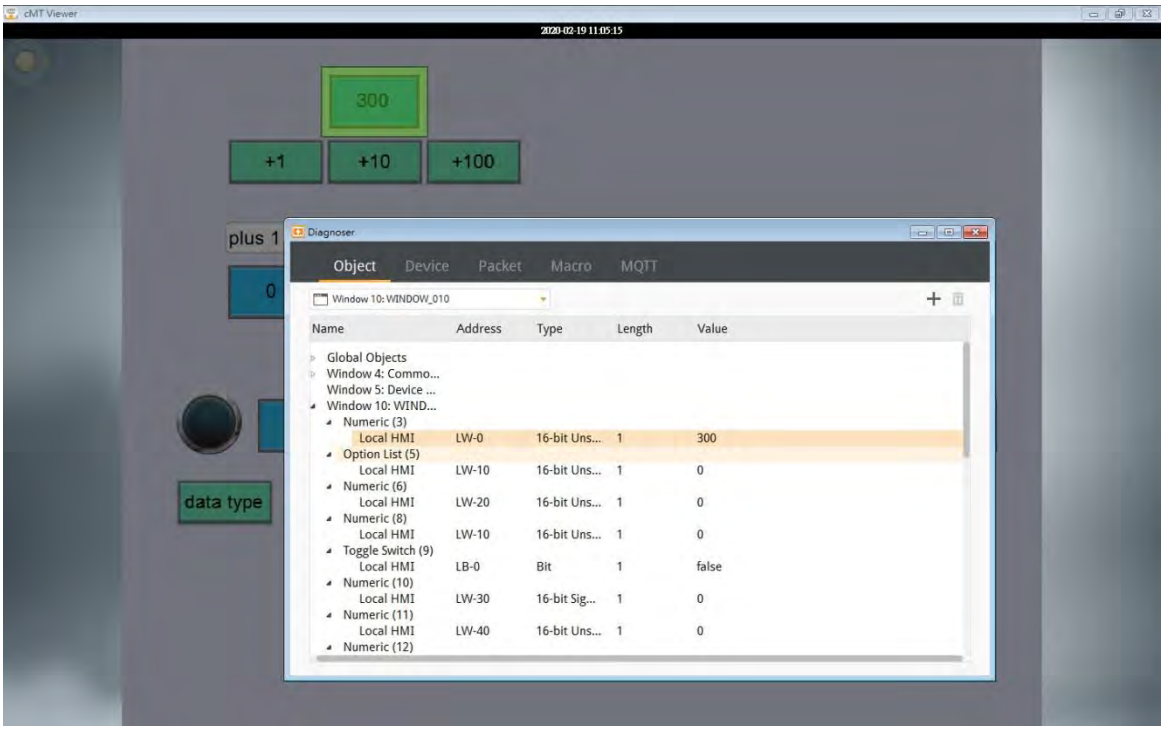
1. In the Object tab in cMT Diagnoser user interface, the objects used in the current window and their addresses, properties, and statuses are shown in a table.



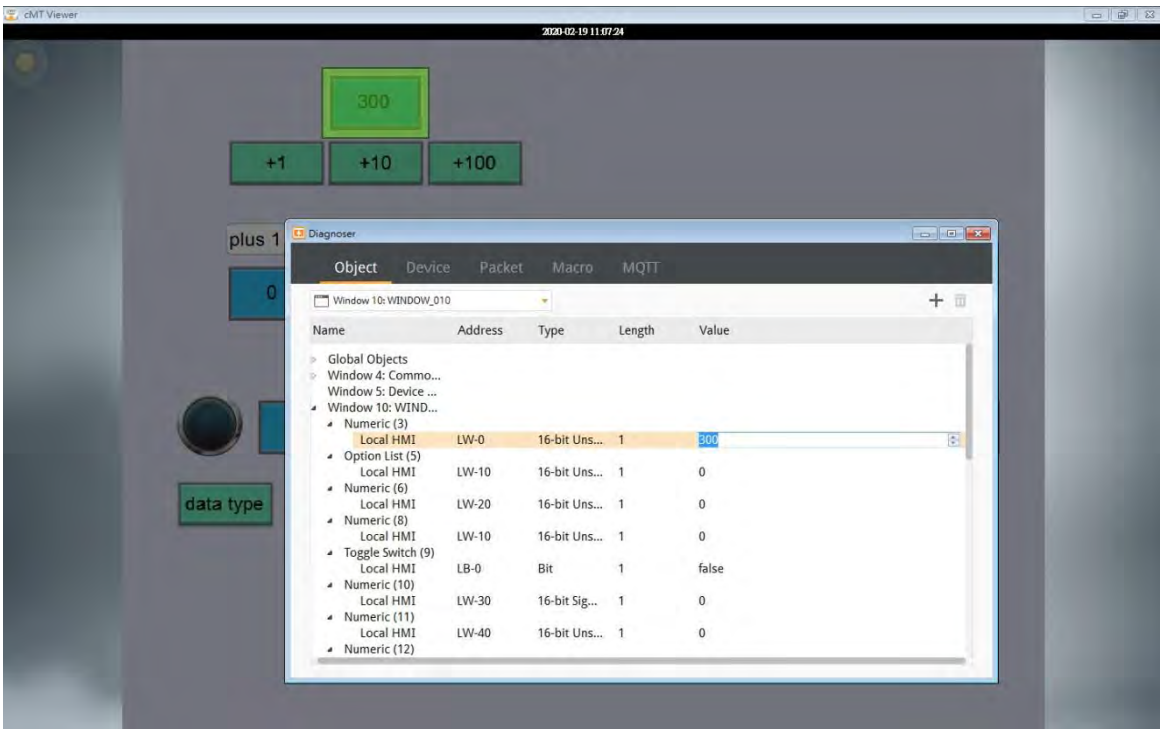
2. Selecting a window from the drop-down list can quickly change to another window.



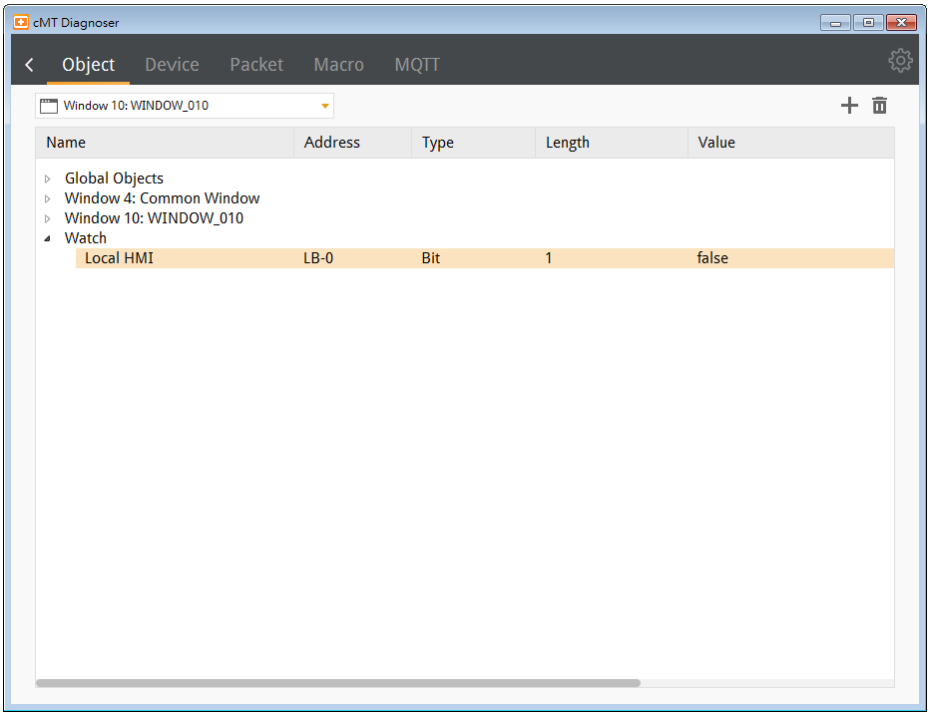
3. The object selected in the cMT Diagnoser window will be highlighted as shown below.



4. The value of an object can be changed by double-clicking on it in cMT Diagnoser window.

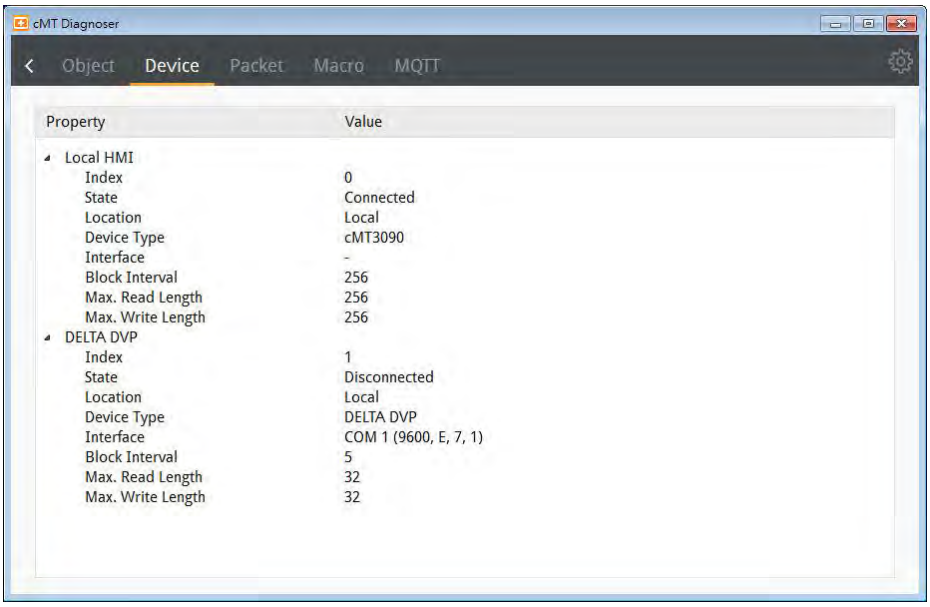


5. In cMT Diagnoser, watch addresses can be added or deleted. Click + to add a new address, or select an address and then click 🗑 to delete it.



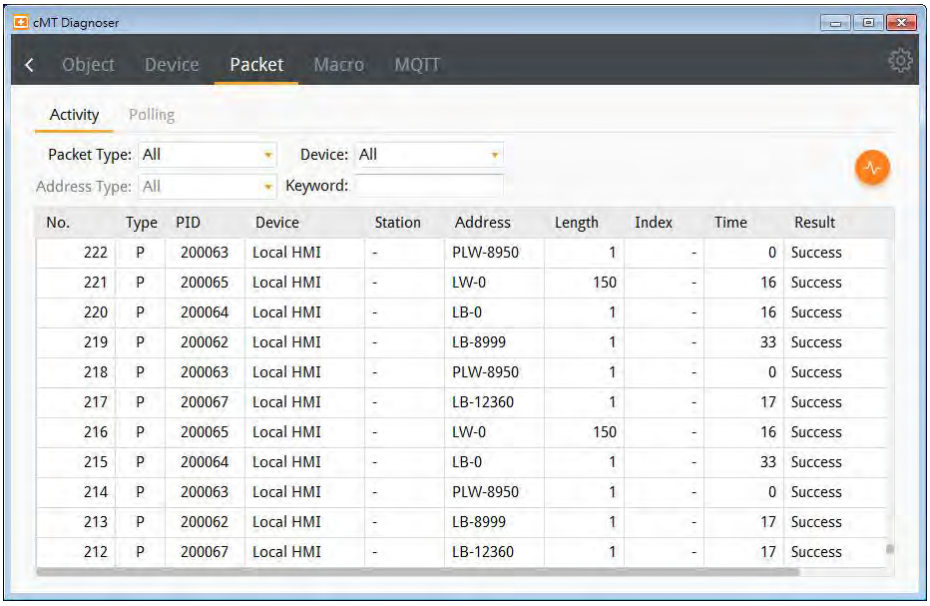
31.3.2. Device Settings

1. The state and property of the device communicating with HMI is shown in the Device tab.



31.3.3. Packet Settings

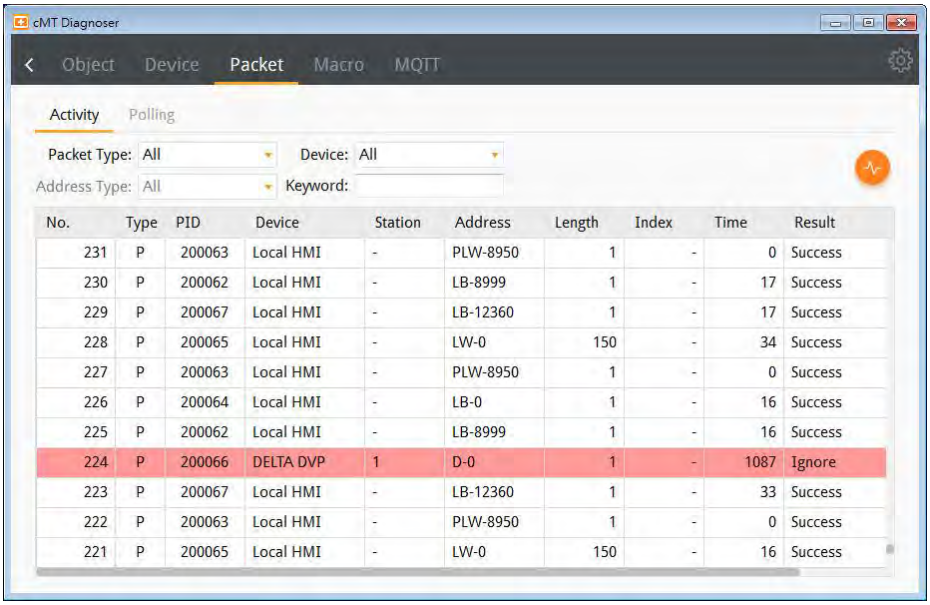
1. In the Packet tab, select the ranges and then click to observe communication status between HMI and PLC.



Item	Description
Packet Type	**All** Displays all packets in Activity area. **Read**

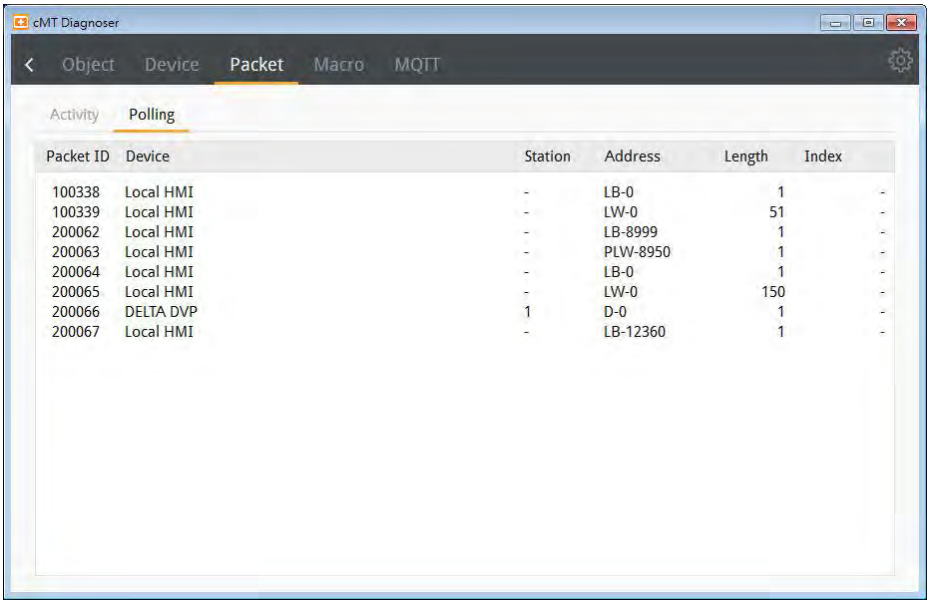
	Displays only Read packets in Activity area. **Write** Displays only Write packets in Activity area. **Polling** Displays only Polling packets in Activity area.
Device	**All** Displays the information of Local HMI and PLC. **Local HMI** Displays the information of Local HMI. **PLC** Displays the information of PLC.
Station	Selects the PLC station number to be displayed. (This function is disabled when selecting [All] in [Device].)
Address Type	Selects all or a preferred address type to be displayed. (This function is disabled when selecting [All] in [Device].)
Keyword	Displays information that contains the keyword entered.

When communication error occurs, information relating to the error will be highlighted in red in the Activity page, and the cause of error is shown in the Result column.



No.	Type	PID	Device	Station	Address	Length	Index	Time	Result
231	P	200063	Local HMI	-	PLW-8950	1	-	0	Success
230	P	200062	Local HMI	-	LB-8999	1	-	17	Success
229	P	200067	Local HMI	-	LB-12360	1	-	17	Success
228	P	200065	Local HMI	-	LW-0	150	-	34	Success
227	P	200063	Local HMI	-	PLW-8950	1	-	0	Success
226	P	200064	Local HMI	-	LB-0	1	-	16	Success
225	P	200062	Local HMI	-	LB-8999	1	-	16	Success
224	P	200066	DELTA DVP	1	D-0	1	-	1087	Ignore
223	P	200067	Local HMI	-	LB-12360	1	-	33	Success
222	P	200063	Local HMI	-	PLW-8950	1	-	0	Success
221	P	200065	Local HMI	-	LW-0	150	-	16	Success

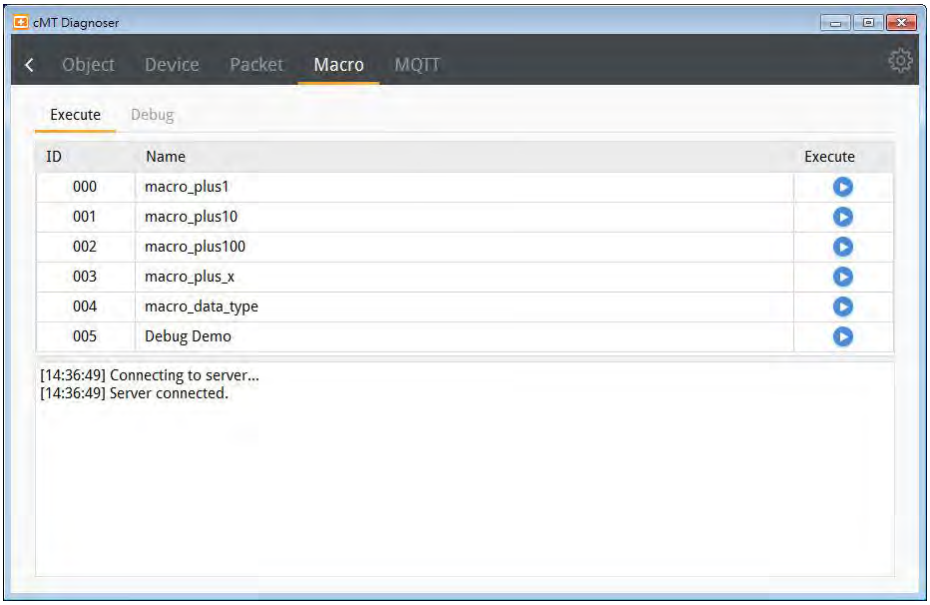
2. The Polling page shows the information of the polling packets periodically sent from HMI, including the start address, length…etc.



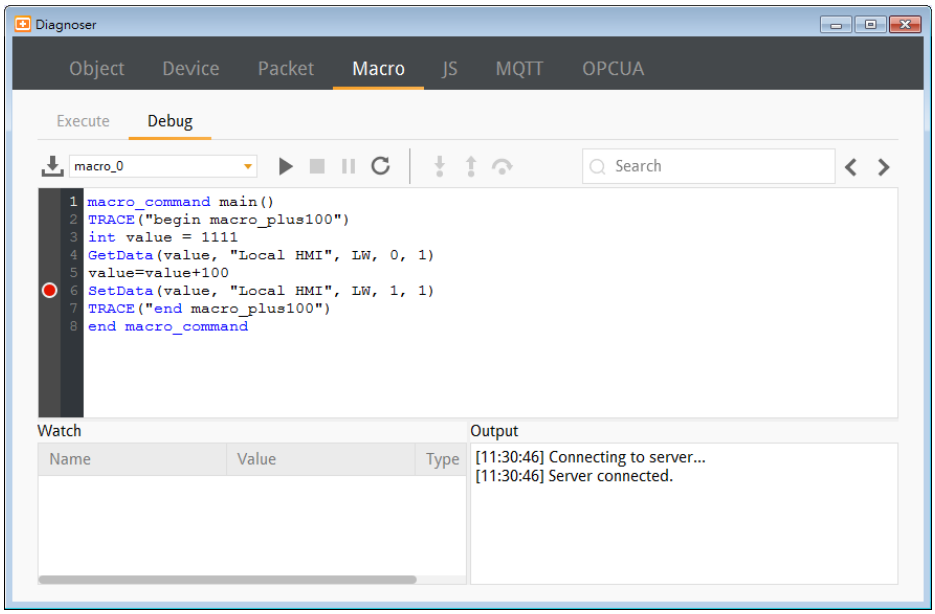
Item	Description
Packet ID	Use the Packet ID to find the invalid object.
Device	Displays HMI and PLC type.
Station	Displays PLC station number.
Address/Length	Displays the device type and the size of the packet.
Index	Displays the index register number of the object.

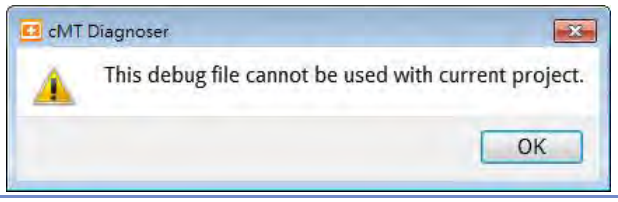
31.3.4. Macro Settings

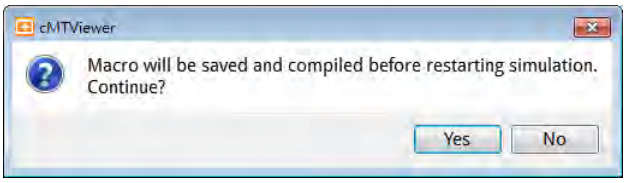
1. Execute and test macros.



2. Macro Debugger Tool.

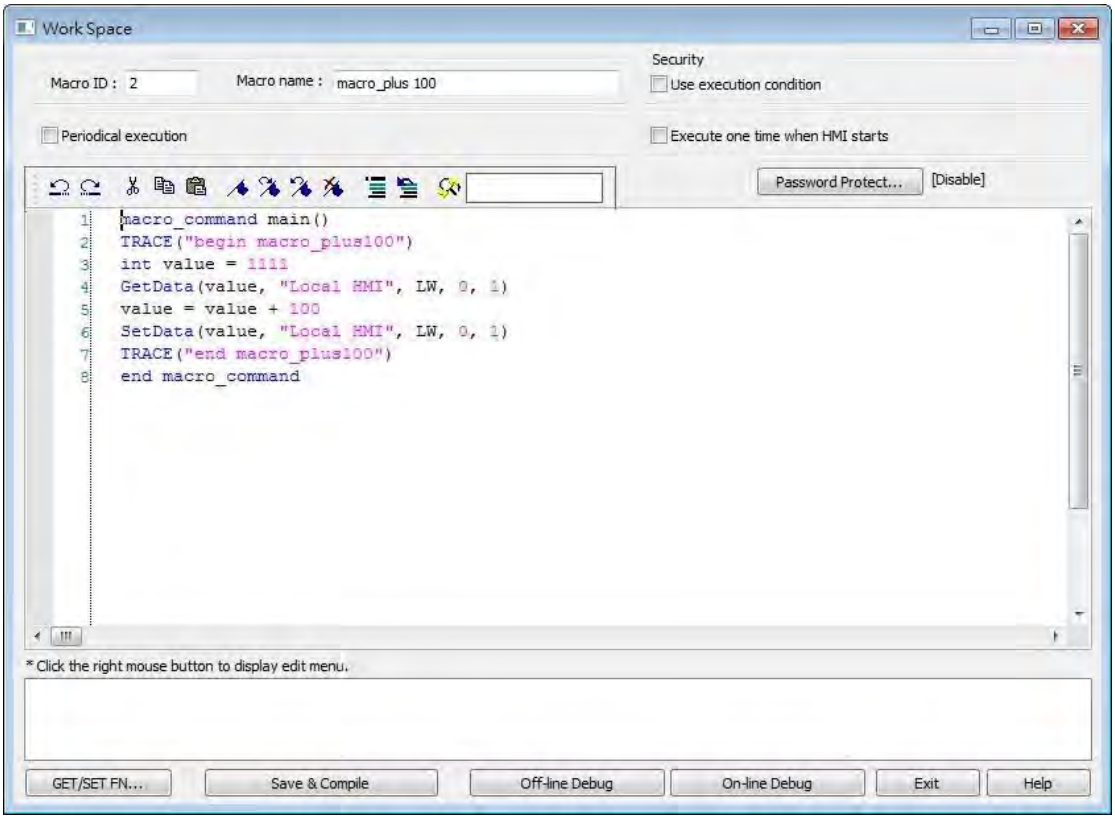


Item	Description
Load macro debug file	Macro Debug File can be generated when compiling a project file in EasyBuilder Pro, and it is placed in the same directory as the compilation file. Macro Debug File has a file name extension of ".debug". When an unmatched debug file is loaded, the following message window shows. 
Start debugging	Start debugging or execute macro from a breakpoint.
Stop debugging	Stop debugging.
Break	Pause macro debugging manually. A Breakpoint can be set at a statement by clicking the grey margin on the left-hand side of the statement. When a macro is launched, it will run until the Breakpoint, and wait for the next command.
Restart simulation	After running on-line / off-line simulation in macro work space, the macro can be revised directly in the Diagnoser window. After revising the macro, the simulation can be restarted by clicking Restart Simulation, and then the following confirmation window pops up.

	
Step Into	Run the macro line by line. If the line contains a function, the diagnoser will enter the called function and continue line-by-line debugging from there.
Stop Out	Return to the line where the current function was called.
Step Over	Run the macro line by line. If the line contains a function, the function will be executed and the result is returned without debugging each line.
Search	Search for a macro by entering keywords.
Watch	When debugging, variables can be added here for watching the change of data.
Output	Shows system information or TRACE output.

Macro Debug in cMT Diagnoser can be opened in the following ways, each with different privilege requirements and operation steps.

Way1. [On-line Debug] or [Off-line Debug] within macro editing work space.



Click [On-line Debug] or [Off-line Debug]. Software will generate a Macro Debug File. Also,

macro editing work space window will close and be replaced by cMT Viewer simulation and cMT Diagnoser, the latter of which loads the Macro Debug File automatically. When the macro is being debugged in cMT Diagnoser's, the macro codes can actually be modified.

Upon completing macro debugging, close the cMT Viewer simulation. The macro editing work space should appear again, and it will ask whether you want to update the macro with the changes made in cMT Diagnoser.

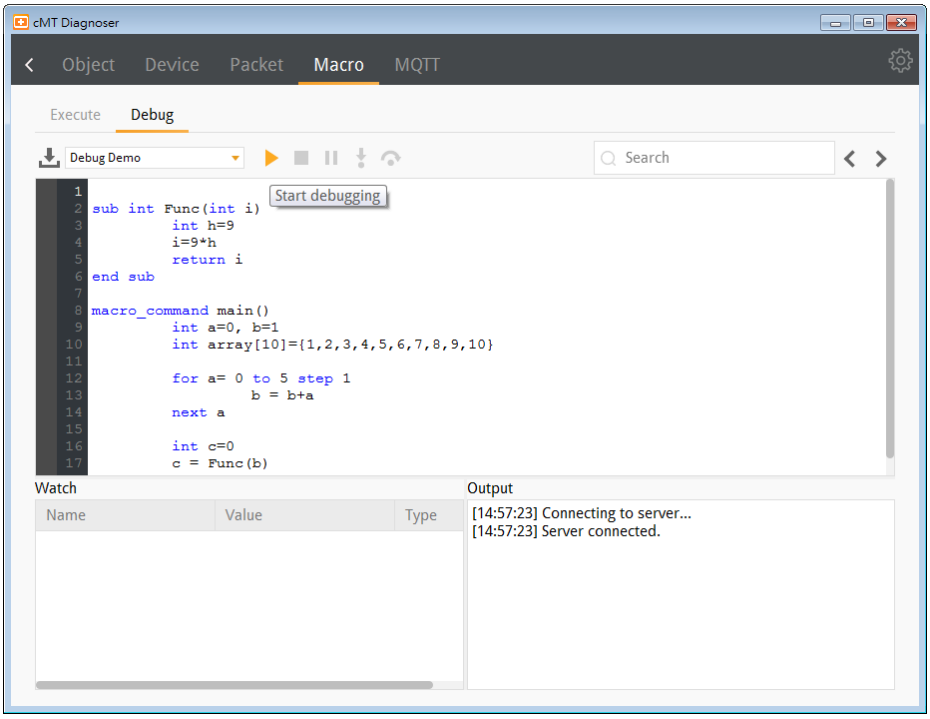
Way2. On-line simulation / off-line simulation in EasyBuilder Pro.
The Macro Debug File is automatically generated after compiling the project file. When cMT Viewer simulation window is shown by running simulation in EasyBuilder Pro, open cMT Diagnoser from the right-click menu. The system imports the debug file automatically when it detects one. In simulation mode, macro can only be debugged but not changed.

Note: There must be at least one macro in the project for the Macro Debug File to be generated during compilation.

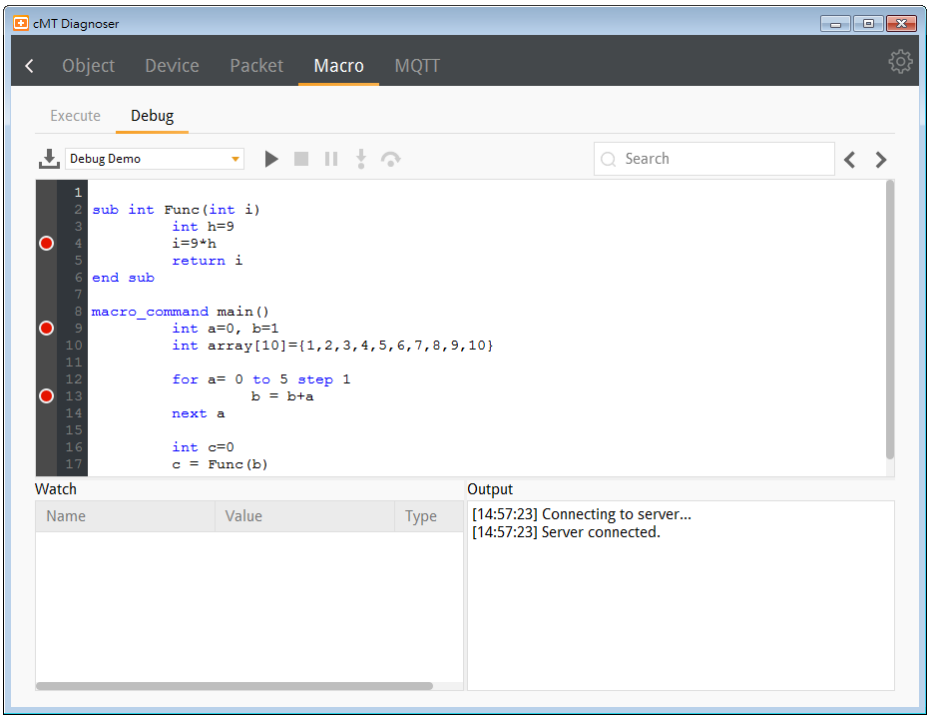
Way3. Downloading the project file to HMI and then running cMT Diagnoser.
The Macro Debug File should be manually imported in the Macro Debugger, and under this mode the macro can only be debugged but not changed.

An Example of Macro Debugging
The following steps show an example about how to debug macro.

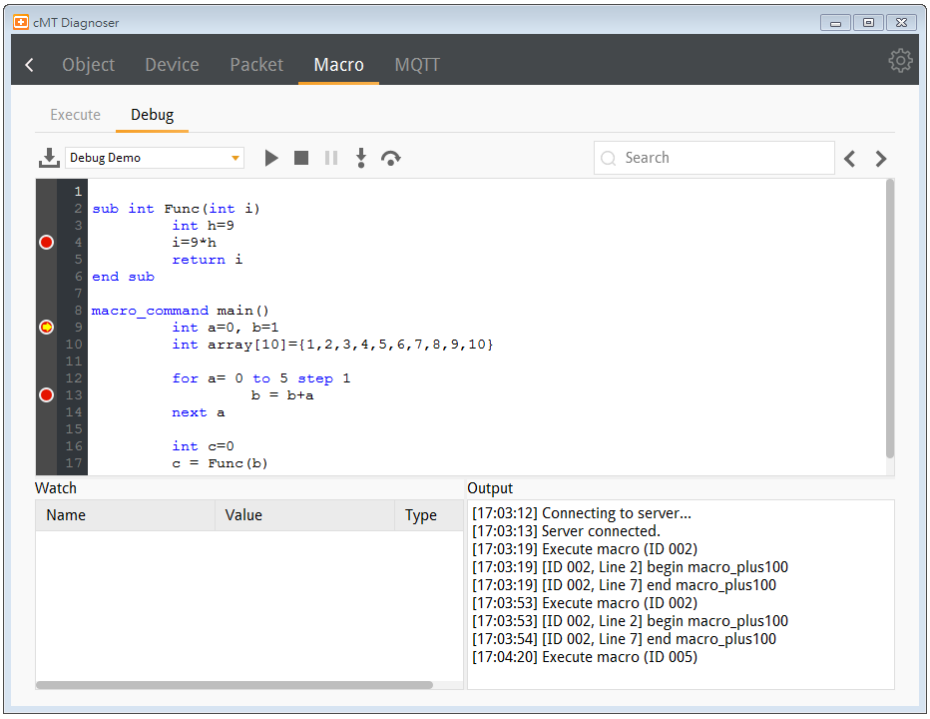
1. Click [Start Debugging].



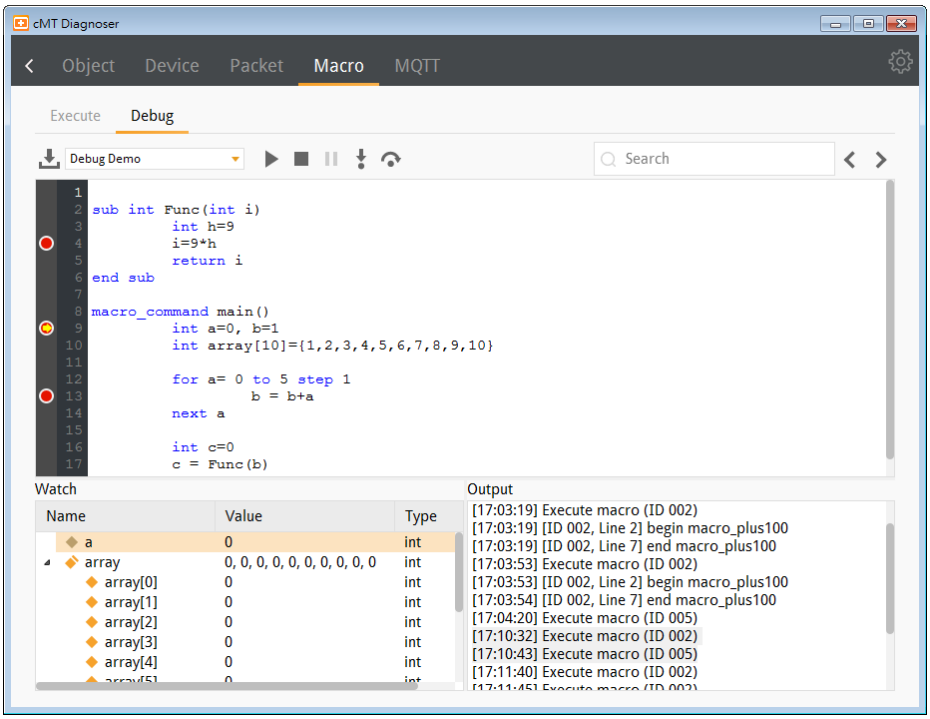
2. Set Breakpoints in the grey margin on the left-hand side of the lines. The Breakpoints are represented by red dots.



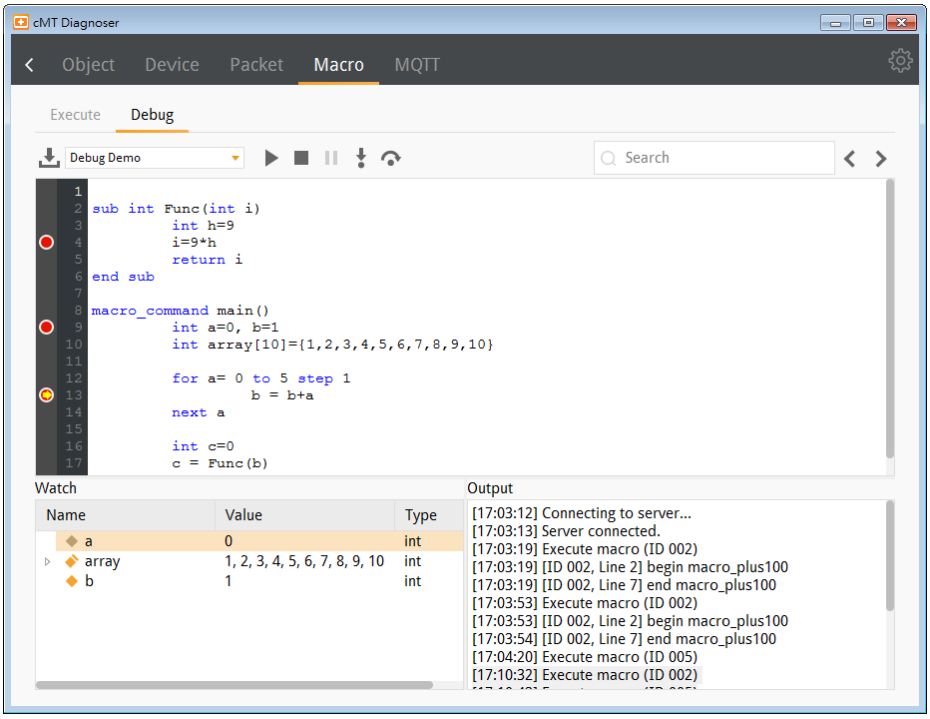
3. After debugging is started, it will stop at the first breakpoint, now marked by a yellow arrow.



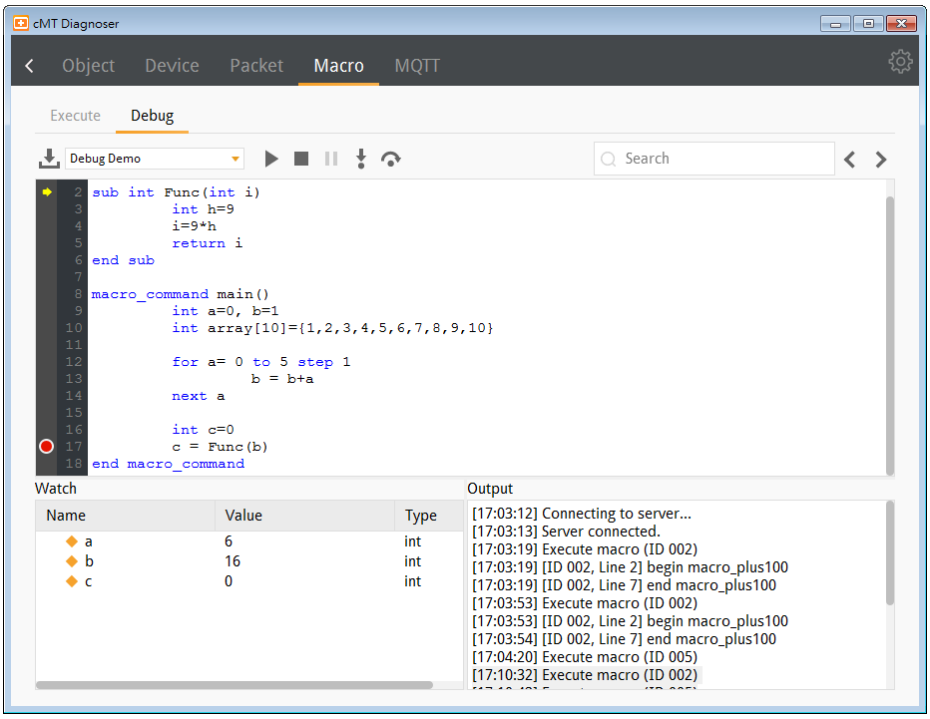
4. At this moment, variables can be added monitored in the Watch table as shown below. Contents of arrays can be monitored as well.



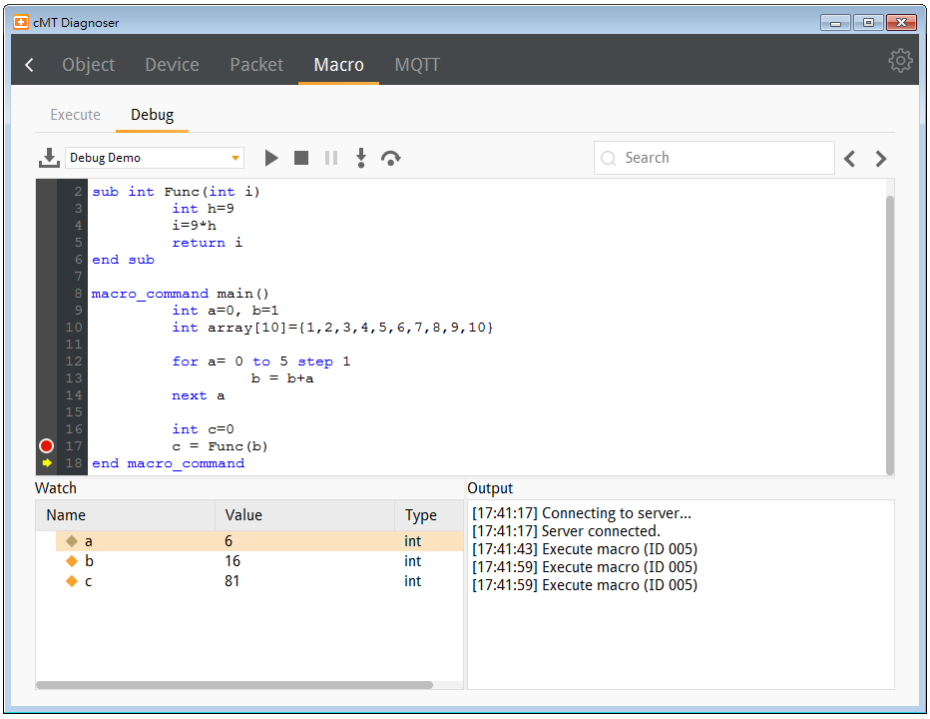
5. After the Start Debugging button is clicked once again, macro will run and stop at the next breakpoint. The changes of values are shown in the Watch list.



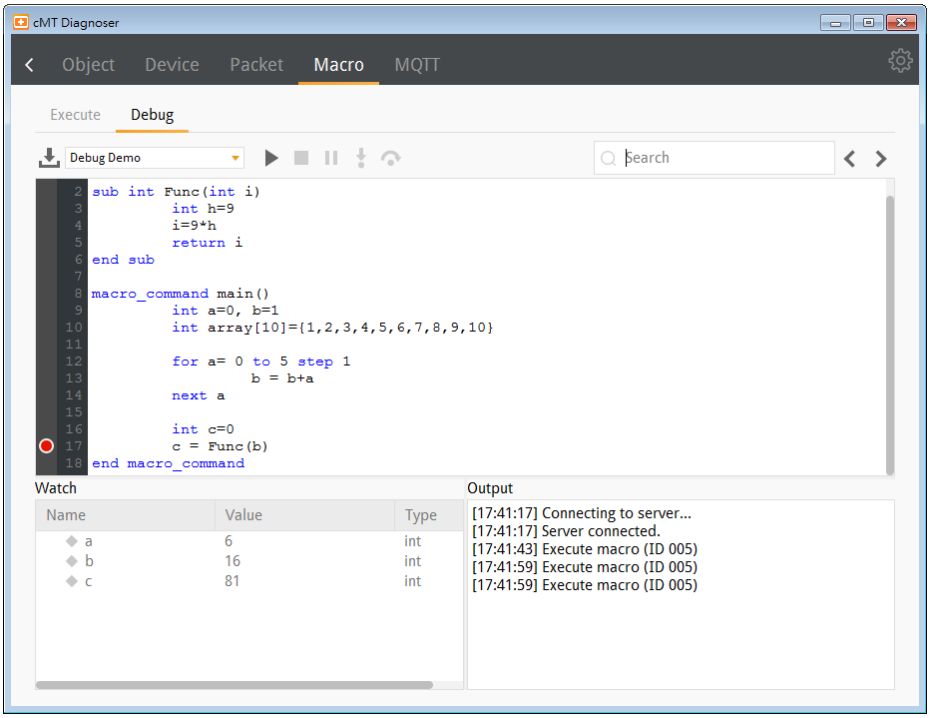
6. Click the Step Into button to run macro debugging line by line. If the line contains a function, cMT Diagnoser will enter the called function and continue line-by-line debugging from there. Changes of values can be observed in the Watch table.



7. Click the Step Over button to run the macro line by line. Now, if the line contains a function, the result will be returned without debugging each line within the function call. When debugging meets a Breakpoint within the function call though, it stops there.

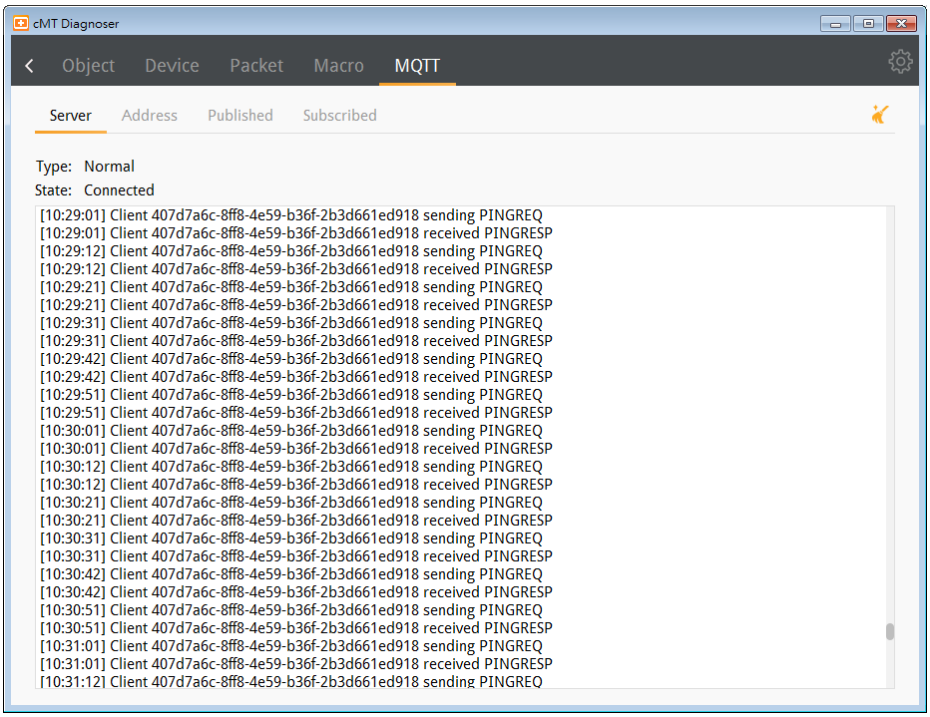


8. Click Stop Debugging button to stop.

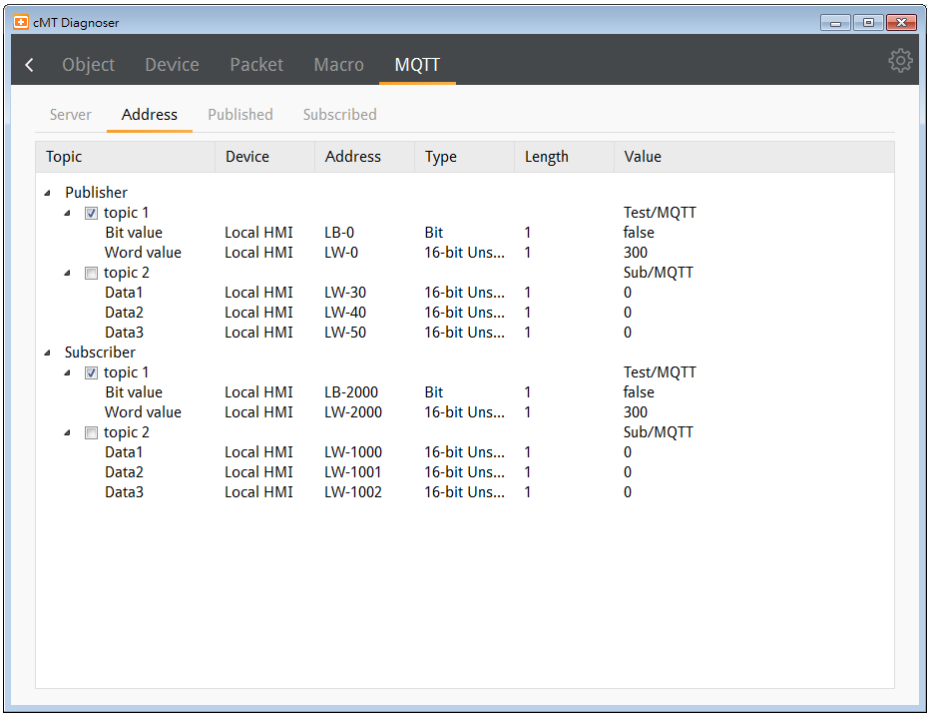


31.3.5. MQTT Settings

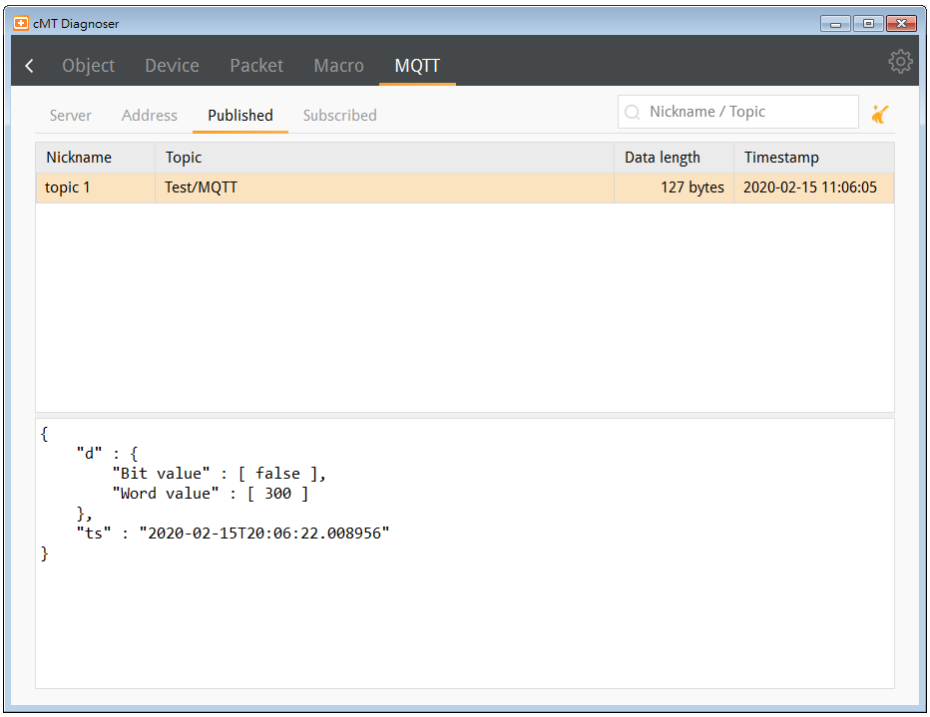
1. In Server tab the type and state of MQTT server is shown. Supported cloud services include: Normal / Azure IoT Hub / Sparkplug / Google Cloud IoT Core. Please note that AWS IoT is not supported.



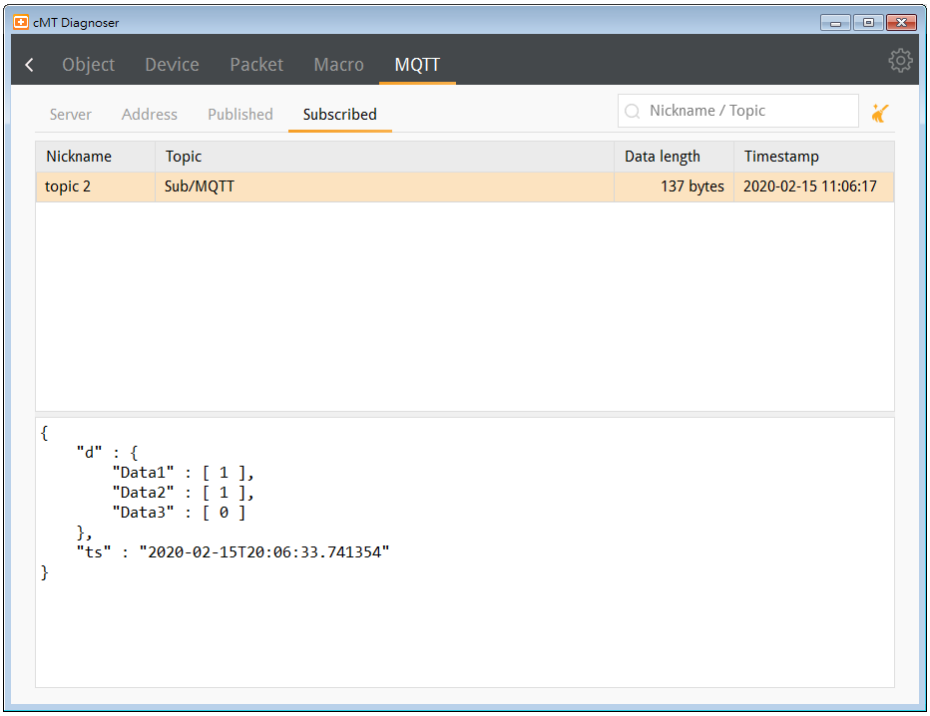
2. In Address tab the Topics published and subscribed by the HMI are shown. Selecting the checkbox of a topic in this tab can make cMT Diagnoser receive all MQTT messages published and subscribed, allowing cMT Diagnoser to change address data for easier MQTT debugging. Only normal servers are supported.



3. In Published tab the message published from HMI is shown. When Raw Data is used as content format, data will be shown in hexadecimal. Only normal servers are supported.

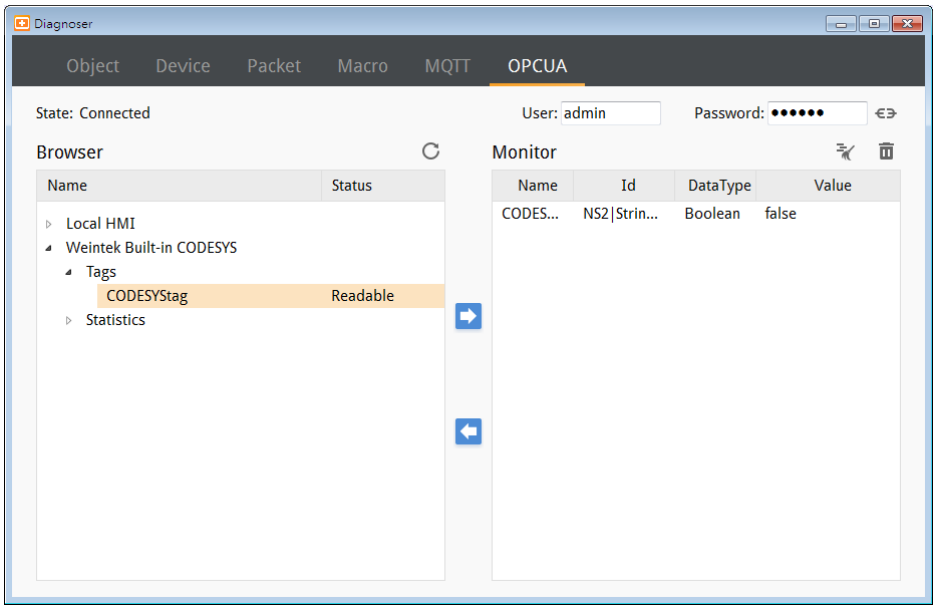


4. In Subscribe tab the message subscribed by HMI is shown. When Raw Data is used as content format, data will be shown in hexadecimal. Only normal servers are supported.



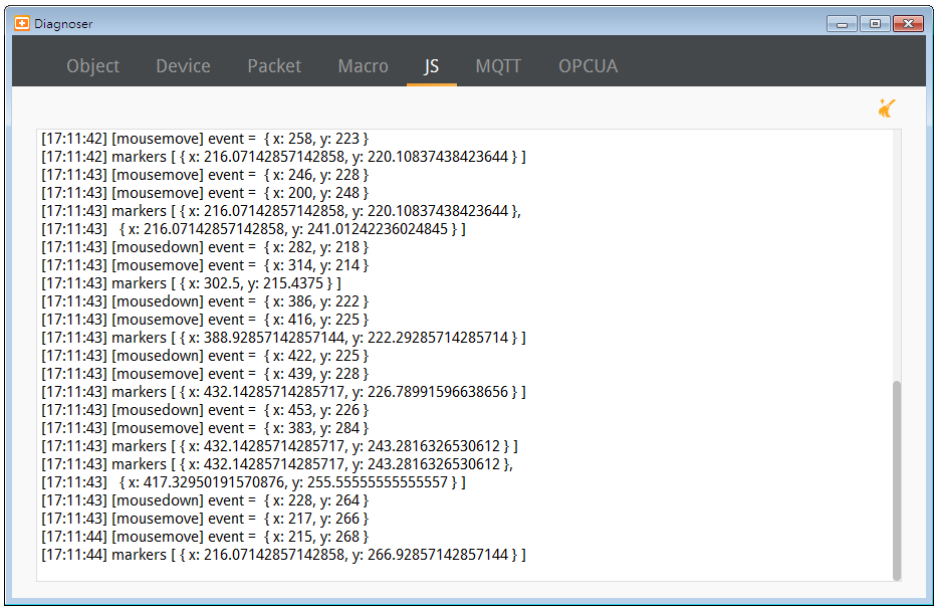
31.3.6. OPC UA Settings

1. Users can log in by entering the credentials to monitor values of OPC UA nodes.



31.3.7. JS Settings

1. In this tab users can view console messages of JS objects.

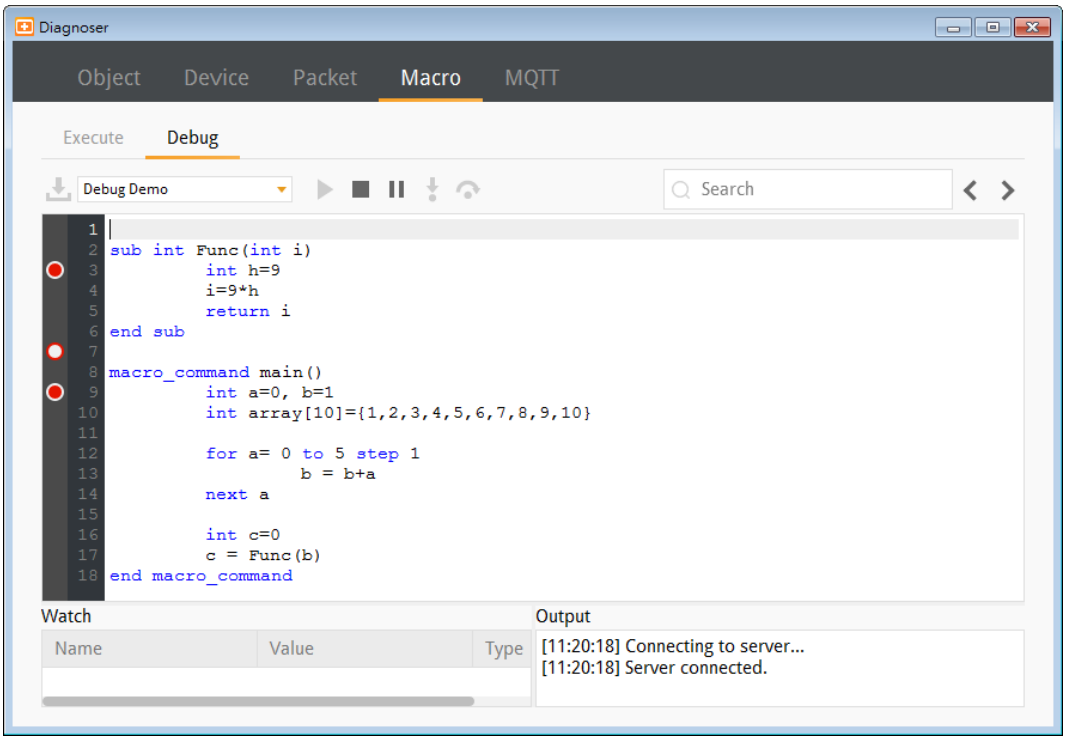


31.4. Additional Notes

1. cMT Diagnoser supports the monitoring all PLC addresses that HMI can access, including Tag PLC addresses.
2. Address values can be displayed in Decimal/HEX/Binary formats. Right-click anywhere on a monitor page to bring the menu up for the change.
3. When closing cMT Viewer with cMT Diagnoser opened, cMT Diagnoser will resume last state and pops up when cMT Viewer simulation window opens again. If Diagnoser is

closed before closing cMT Viewer, users should manually open Diagnoser in next simulation.

4. When [Password Protect] is enabled in Macro Manager, the Macro Debug File cannot be generated after compilation.
5. Password Protect can be enabled for an individual macro, and two modes are available: Encrypted and Read Only. To debug a password protected macro, a password is required to unlock the macro in cMT Diagnoser. If the macro is protected in Read Only mode, it can only be viewed but not changed when debugging. If the macro is protected in Encrypted mode, it cannot be viewed at all.
6. When a macro is not successfully compiled, it will be put into the Macro Under Development list and will not be contained in the Macro Debug File.
7. cMT Diagnoser can only debug one macro at a time. When debugging a macro, executing another macro will not be effective.
8. When the breakpoint is placed on irrelevant lines, such as beside an empty line , variable declaration without initial value assignment, the breakpoint will be represented by a hollow red dot as line 7 below shows.



32. FTP Server Application

This chapter explains how to use FTP Server.

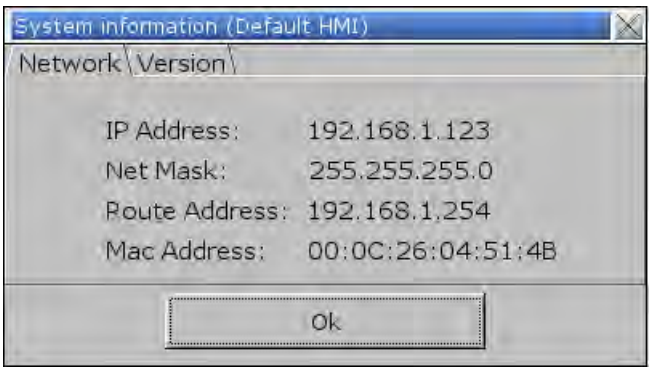
32.1. Overview 32-2
32.2. Steps to Log in FTP Server 32-2
32.3. Backup History Data and Update Recipe Data 32-3

32.1. Overview

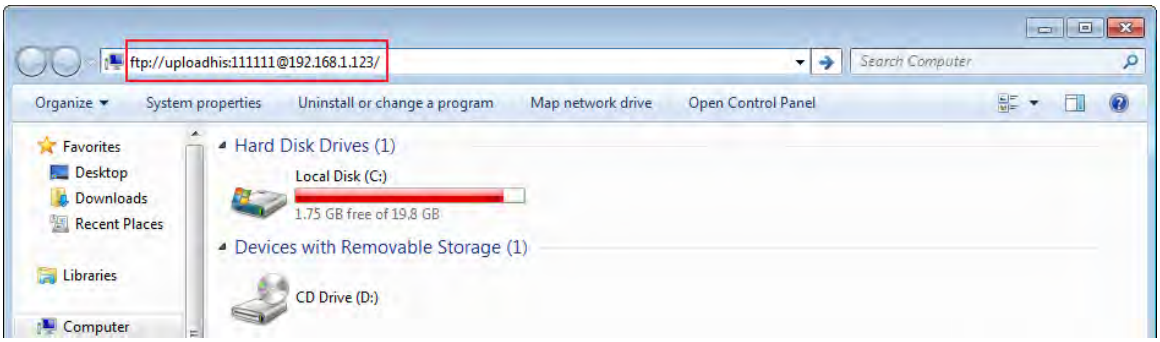
Apart from saving the history data from an HMI to your PC by using SD card, USB disk or EasyPrinter, the FTP Server can also do the backup. After downloading a project to the HMI, the FTP Server can be used to backup or update the history data and the recipe data, but not able to delete those data. Please disconnect from FTP before removing an SD card or USB disk, otherwise these external devices may not work when next time inserted.

32.2. Steps to Log in FTP Server

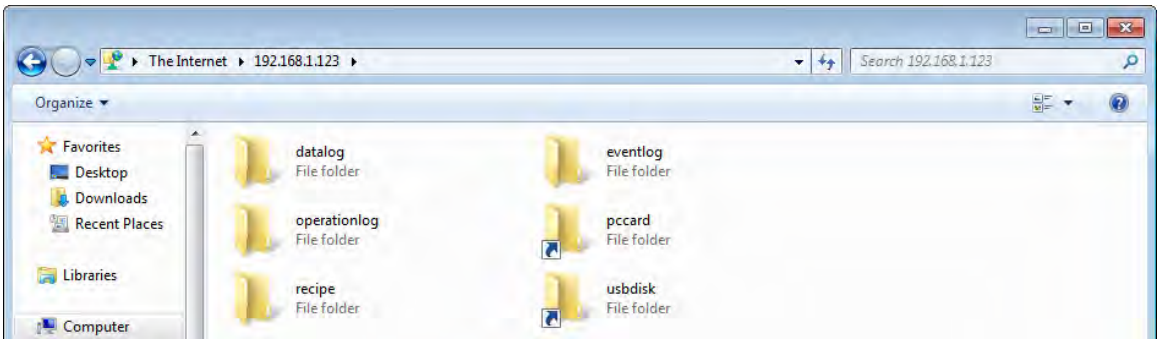
1. Before logging in FTP Server, please check the HMI IP address.



2. On PC, enter the HMI IP address: ftp://192.168.1.123 (example), then log in by the user name: uploadhis, and enter the HMI [history upload password] (if not changed, the default password is 111111). Or, directly enter ftp://uploadhis:111111@192.168.1.123/



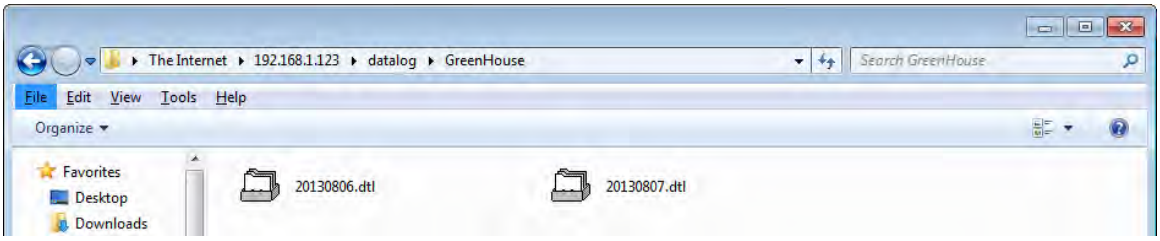
3. After entering the IP address,the ftp address: ftp://192.168.1.123 and its folders are shown as below:



32.3. Backup History Data and Update Recipe Data

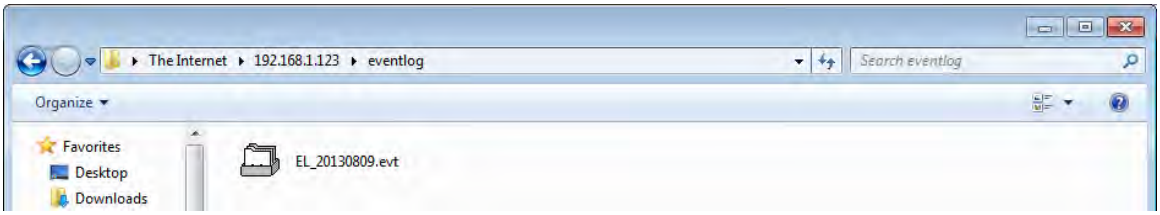
- Steps to backup Data Sampling records

1. Click “datalog” folder to view the files.
2. Click the file to check the records.
3. Copy and paste the needed files to your PC.



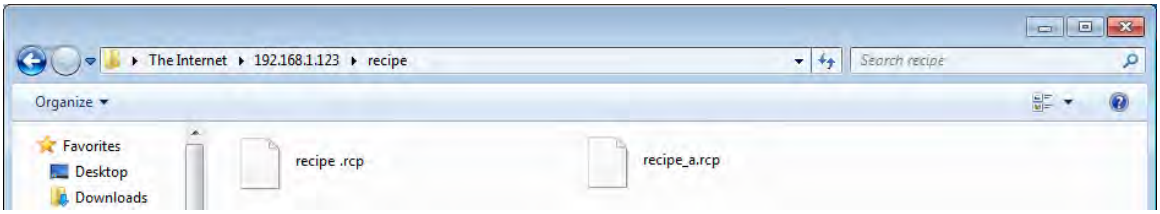
- Steps to backup Event (Alarm) Log records

1. Click “eventlog” folder to view the files.
2. Copy and paste the needed files to your PC.



- Steps to backup or update Recipe records

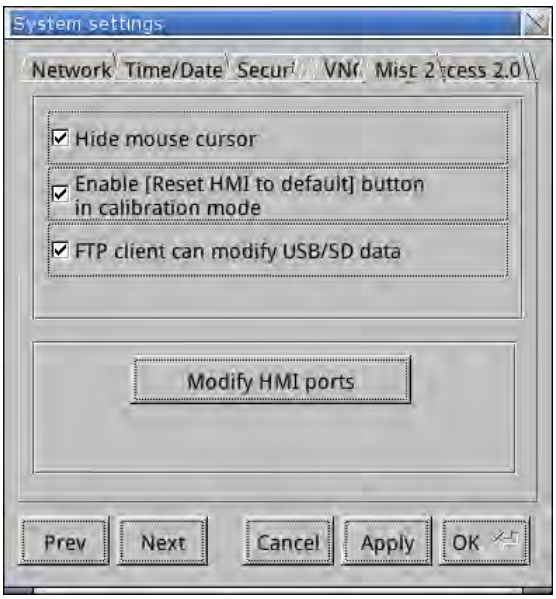
1. Click “recipe” folder to view the files.
2. Copy and paste the needed files to your PC.

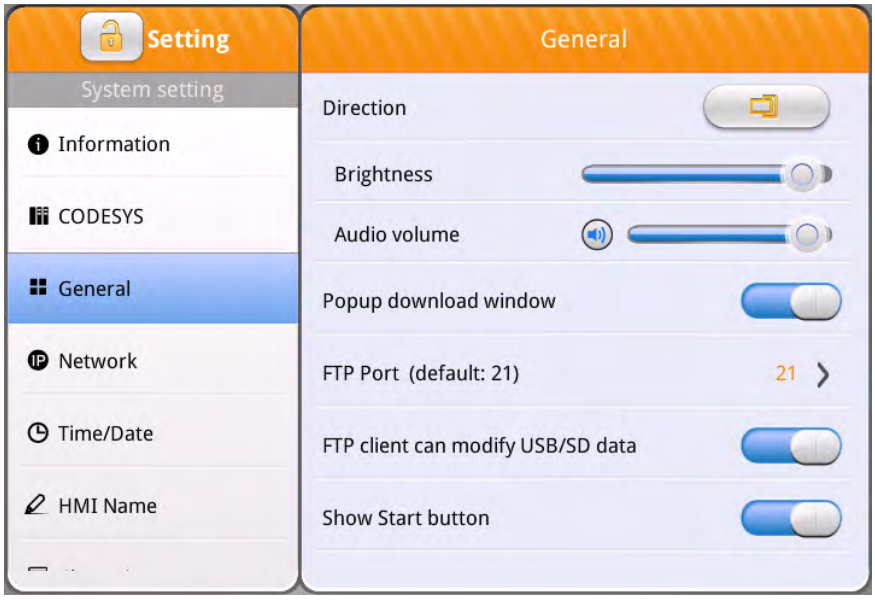


- Steps to modify data in USB/SD from FTP client

1. By default, [FTP client can modify USB/SD data] option is not selected. With this option selected, files in USB/SD can be deleted or modified from FTP client.

2. As shown below, this option can be found in system settings » Misc 2 tab on MT Series, and in system settings » General tab on cMT / cMT X Series.





Note

- Since the recipe data is automatically saved per minute, after updating “recipe.rcp” or “recipe_a.rcp”, the HMI must be rebooted within one minute or the saving will be failed.
- The HMI can be rebooted by the system registers: [LB-9047] (reboot HMI) and [LB9048] (reboot HMI protection). You can set [LB-9048] ON first, and then set [LB-9047] ON to reboot the HMI.
- When using FTP server, the date of modification for the files on HMI will be UTC+0, and when users view these files on PC, the date of modification is added by PC’s time zone, which can make the time different from the actual time.

33. EasyDiagnoser

This chapter explains how to use EasyDiagnoser.

33.1. Overview 33-2
33.2. Configuration 33-2
33.3. EasyDiagnoser Settings 33-3
33.4. Error Code 33-9
33.5. Window Adjustment 33-10

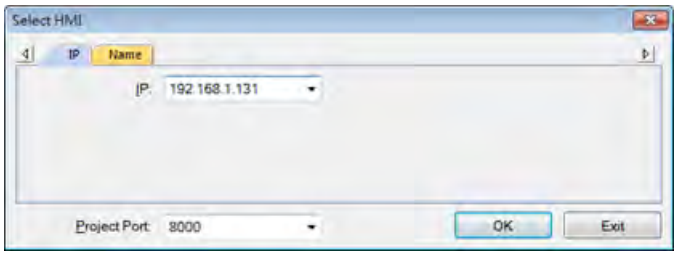
33.1. Overview

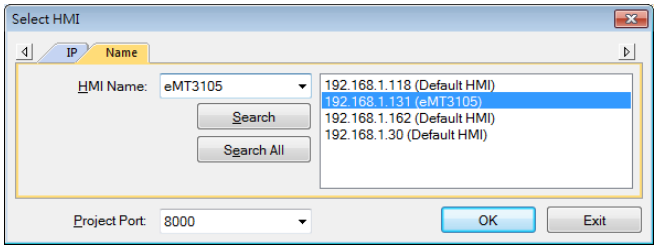
EasyDiagnoser is a tool for detecting the error in the communication of HMI with PLC.

33.2. Configuration

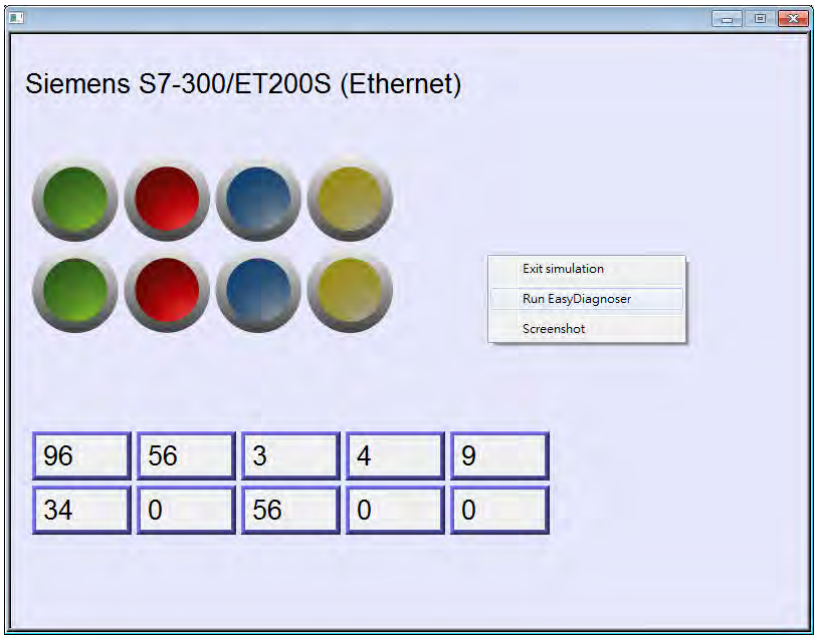
The following steps explain how to configure EasyDiagnoser.

1. Open Utility Manager and click EasyDiagnoser.
2. Set the HMI IP address. Enter IP address or click [Search All], and then enter [Project Port].

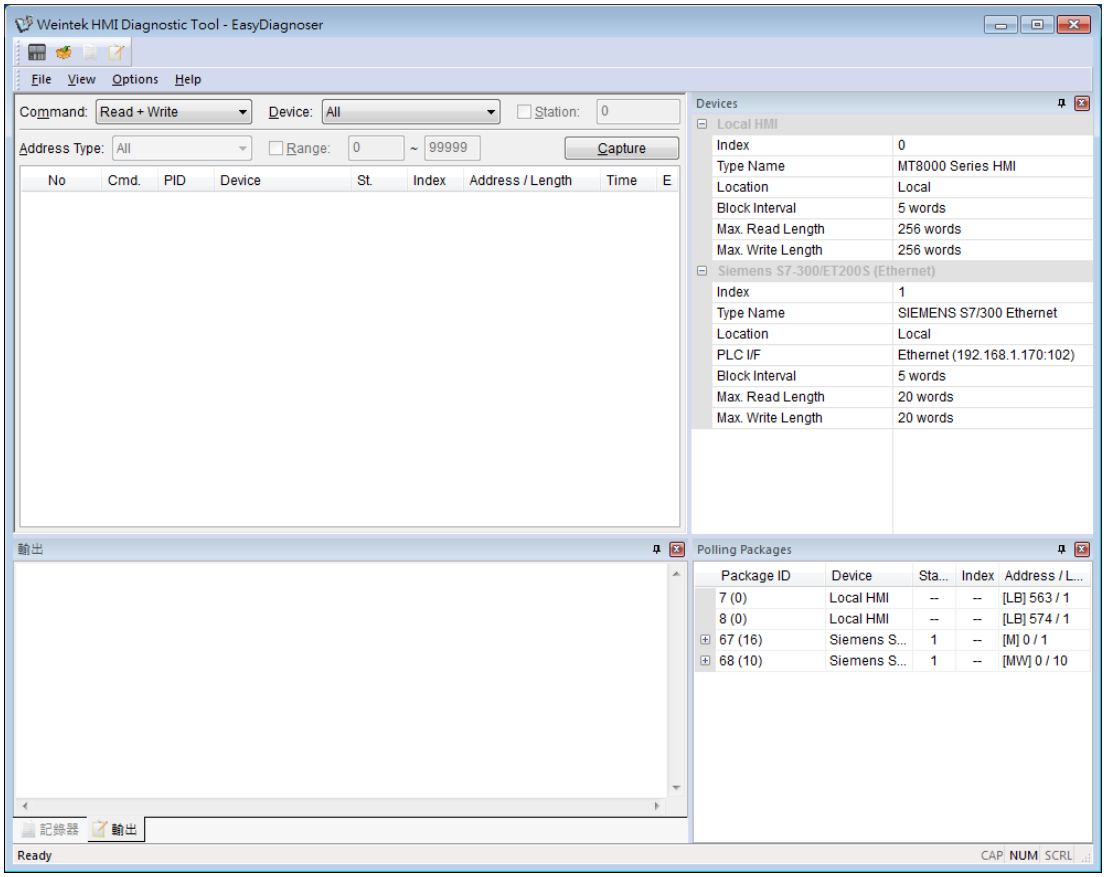




Or, during On-line simulation, right click and select [Run EasyDiagnoser] to open EasyDiagnoser. EasyDiagnoser will monitor the communication between PC and PLC.



3. When finished, click [OK], EasyDiagnoser dialog box is shown in the following figure.



33.3. EasyDiagnoser Settings

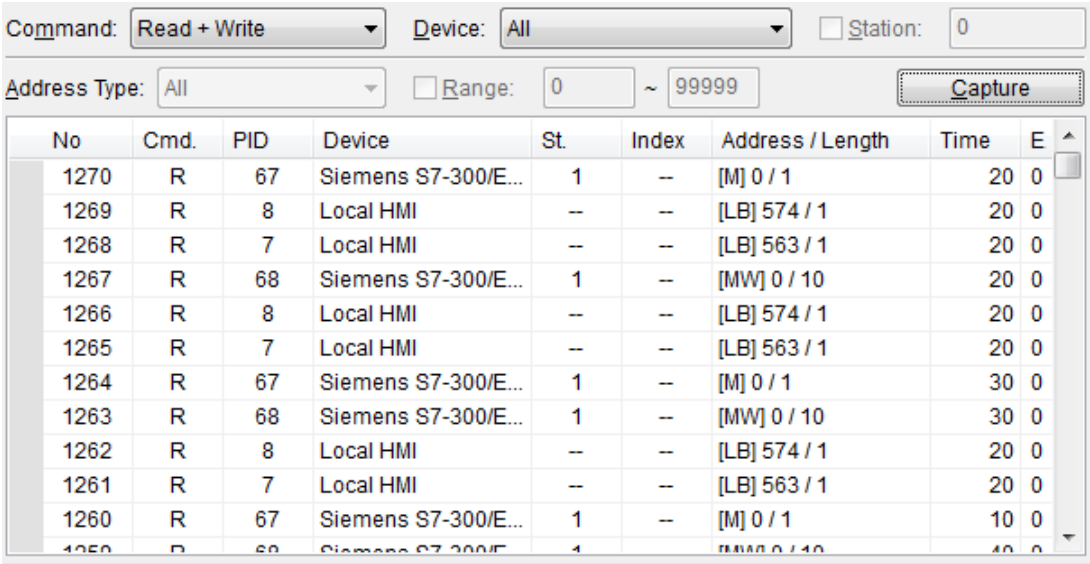
33.3.1. Main Menu

Item	Description
File	**Save As** The communication data can be saved as .xls file which can be opened by Excel. **Exit** Exits current file.
View	**Device Bar** displays Device window. **Package Bar** displays Package window. **Logger Bar** displays Logger window. **Output Bar** displays Output window.
Options	**Toolbars** displays toolbar icons of Device Bar, Package Bar, Logger Bar, and Output Bar.

	Status Bar displays information of CAP, NUM, and SCRL at the bottom of EasyDiagnoser window. **Update Package List** displays the Polling Package information of current page. **Show Object ID (HMI)** shows the ID of the objects on HMI as shown in the following figure. **Clear Activity List** clears all the information recorded during communication.
Help	Displays EasyDiagnoser version information.

33.3.2. Activity Area

In the activity area, users can observe the communication between HMI and PLC.



No	Cmd.	PID	Device	St.	Index	Address / Length	Time	E
1270	R	67	Siemens S7-300/E...	1	--	[M] 0 / 1	20	0
1269	R	8	Local HMI	--	--	[LB] 574 / 1	20	0
1268	R	7	Local HMI	--	--	[LB] 563 / 1	20	0
1267	R	68	Siemens S7-300/E...	1	--	[MW] 0 / 10	20	0
1266	R	8	Local HMI	--	--	[LB] 574 / 1	20	0
1265	R	7	Local HMI	--	--	[LB] 563 / 1	20	0
1264	R	67	Siemens S7-300/E...	1	--	[M] 0 / 1	30	0
1263	R	68	Siemens S7-300/E...	1	--	[MW] 0 / 10	30	0
1262	R	8	Local HMI	--	--	[LB] 574 / 1	20	0
1261	R	7	Local HMI	--	--	[LB] 563 / 1	20	0
1260	R	67	Siemens S7-300/E...	1	--	[M] 0 / 1	10	0

Item	Description
Command	**Read + Write** Displays Read and Write information in activity area. **Read** Displays only Read information in activity area.

	Write Displays only Write information in activity area.
Device	**All** Displays the information of Local HMI and PLC. ● If command is set to Read + Write, the Read and Write information of Local HMI and PLC will be displayed in the activity area. ● If command is set to Read, the Read information of Local HMI and PLC will be displayed in the activity area. ● If command is set to Write, the Write information of Local HMI and PLC will be displayed in the activity area. **Local HMI** Displays the information of Local HMI. ● If command is set to Read + Write, the Read and Write information of Local HMI will be displayed in the activity area. ● If command is set to Read, the Read information of Local HMI will be displayed in the activity area. ● If command is set to Write, the Write information of Local HMI will be displayed in the activity area. **PLC** Displays the information of PLC. ● If command is set to Read + Write, the Read and Write information of PLC will be displayed in the activity area. ● If command is set to Read, the Read information of PLC will be displayed in the activity area. ● If command is set to Write, the Write information of PLC will be displayed in the activity area.
Station	Selects the PLC station number to be displayed. (This function is disabled when selecting [All] in [Device]).
Address Type	Selects all or a preferred address type to be displayed. (This function is disabled when selecting [All] in [Device]).
Range	Sets the range of address types. (This function is disabled when selecting [All] in [Device]).
Capture	Click to start/stop capturing the communication message.
Error	Please see “33.4 Error Code”.

33.3.3. Polling Packages

Polling Packages

Package ID	Device	Station	Index	Address / Length
7 (0)	Local HMI	--	--	[LB] 563 / 1
8 (0)	Local HMI	--	--	[LB] 574 / 1
⊞ 67 (16)	Siemens S7-300/ET20...	1	--	[M] 0 / 1
⊞ 68 (10)	Siemens S7-300/ET20...	1	--	[MW] 0 / 10

Item	Description
Package ID	Uses the Package ID to check the error of the object.
Device	Displays HMI and PLC type.
Station	Displays PLC station number.
Index	Displays the index register numbers of the objects.
Address / Length	Displays the device type and the size of the package (in words).

Polling Packages

Object		Screen	ID	Address
7 (0)	Local HMI	--	--	[LB] 563 / 1
8 (0)	Local HMI	--	--	[LB] 574 / 1
⊟ 67 (16)	Siemens S7-300/ET...	1	--	[M] 0 / 1
▶ Toggle Switch		10	1	[M] 0
Toggle Switch		10	1	[M] 0
Toggle Switch		10	12	[M] 0
Toggle Switch		10	12	[M] 0
Toggle Switch		10	13	[M] 0
Toggle Switch		10	13	[M] 0
Toggle Switch		10	14	[M] 0
Toggle Switch		10	14	[M] 0
Toggle Switch		10	15	[M] 0
Toggle Switch		10	15	[M] 0
Toggle Switch		10	16	[M] 0

Item	Description
Object	Check the object in the package.
Screen	The window in the project where the object is placed.
ID	The ID number of the object.
Address	The address of the object.

Note

- Click [Package ID], the device station number will be displayed in the 3rd column.

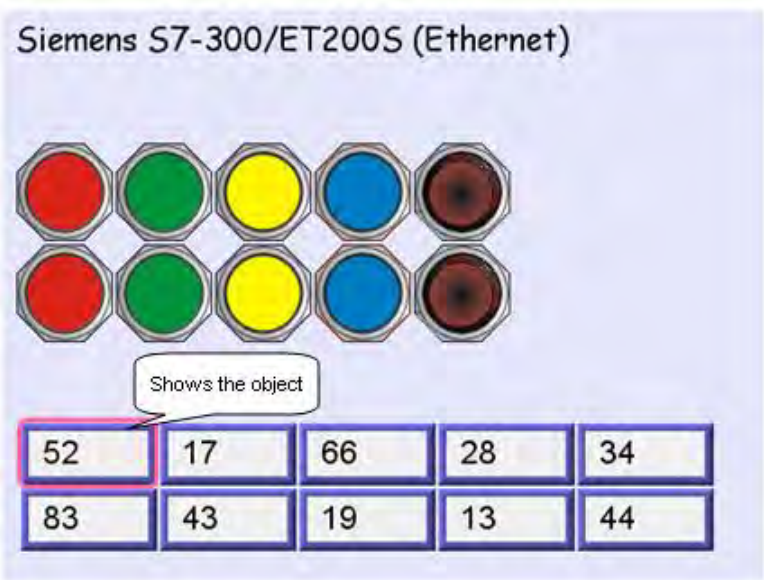
Polling Packages

Package ID	Device	Station	Index	Address / Length
7 (0)	Local HMI	--	--	[LB] 563 / 1
8 (0)	Local HMI	--	--	[LB] 574 / 1
67 (16)	Siemens S7-300/ET2...	1	--	[M] 0 / 1
68 (10)	Siemens S7-300/ET2...	1	--	[MW] 0 / 10

- Double click [Package ID] then select [object] to display the position of the object.
  For example, select [Numeric Input] and the screen no. displays 10.
  This shows that this object is in window no. 10 in the project and will be marked with pink frame on HMI as shown in the following figures

Polling Packages

Object		Screen	ID	Address
7 (0)	Local HMI	--	--	[LB] 563 / 1
8 (0)	Local HMI	--	--	[LB] 574 / 1
67 (16)	Siemens S7-300/ET20...	1	--	[M] 0 / 1
68 (10)	Siemens S7-300/ET20...	1	--	[MW] 0 / 10
Numeric Input		10	2	[MW] 0
Numeric Input		10	3	[MW] 2
Numeric Input		10	4	[MW] 4
Numeric Input		10	5	[MW] 6
Numeric Input		10	6	[MW] 8
Numeric Input		10	7	[MW] 10
Numeric Input		10	8	[MW] 12
Numeric Input		10	9	[MW] 14
Numeric Input		10	10	[MW] 16
Numeric Input		10	11	[MW] 18



33.3.4. Devices

Displays the information of HMI and PLC.

Devices

Local HMI	
Index	0
Type Name	MT8000 Series HMI
Location	Local
Block Interval	5 words
Max. Read Length	256 words
Max. Write Length	256 words
Siemens S7-300/ET200S (Ethernet)	
Index	1
Type Name	SIEMENS S7/300 Ethernet
Location	Local
PLC I/F	Ethernet (192.168.1.170:102)
Block Interval	5 words
Max. Read Length	20 words
Max. Write Length	20 words

33.3.5. Output (Macro debug)

With Macro Trace function, the executing status of Macro can be seen.

In the illustration below, for [ID 1, Ln 7] and [ID 1, Ln 12]

ID 1 represents Macro name.

Ln 7 and Ln 12 represent that data are in the 7th and 12th line of Macro.

Output

```
[ID 1, Ln 7] LW0 = 1
[ID 1, Ln 12] LW0 = 2
[ID 1, Ln 7] LW0 = 2
[ID 1, Ln 12] LW0 = 3
[ID 1, Ln 7] LW0 = 3
[ID 1, Ln 12] LW0 = 4
```

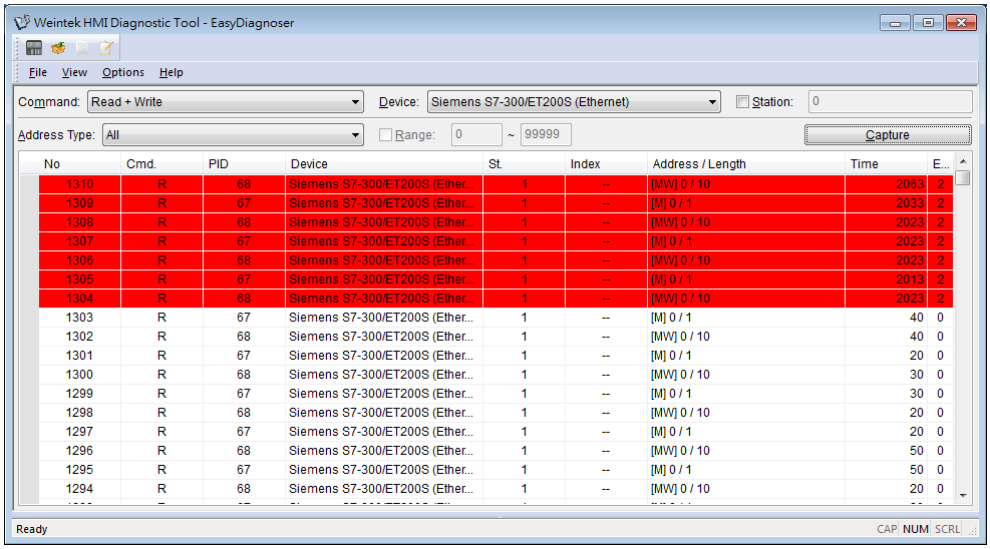
For more information, see “18 Macro Reference”.

33.4. Error Code

In the activity area, the reason of error can be found through the error codes listed below.

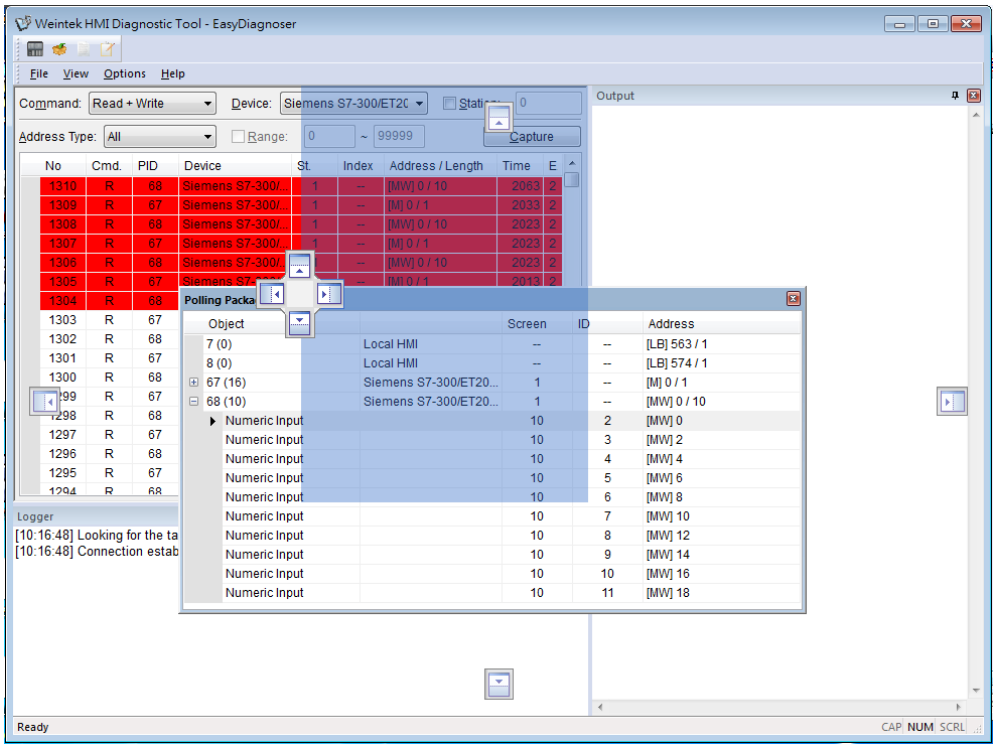
Error Code	Cause of Communication Error
0	Normal
1	The device is busy and not yet ready to process a command.
2	Communication error due to unexpected reason.
3	The device does not exist.
4	The device using the specified station number does not exist.
5	Incorrect address format.
6	Read/Write unsupported address.
7	The driver of the device does not exist.
8	The COM port does not exist.
9	Incorrect IP address or unable to connect the device.
10	Checksum error.
11	Unidentified command.
12	Ignore
20	The USB device is improperly connected.
21	The CAN Bus device is improperly connected.
22	No reply from the device.
23	Insufficient data read from the device before timeout.
24	The Conversion Tag used by the object does not exist or the content is incorrect.
25	HMI is not accepting any commands from a remote HMI.
251	Read/Write exceeding number of words from/to the register of the MODBUS device.
252	MODBUS device replies incorrect data format.
253	MODBUS device checksum error.

When error occurs, the error message will be shaded red as shown in the following figure.



33.5. Window Adjustment

Users can drag or use the smart docking icons in editing window to place the windows to a desired position.



Note

- EasyDiagnoser doesn't support Siemens S7/1200 (Ethernet) and Allen-Bradley Ethernet/IP (CompactLogix/ControlLogix) – Free Tag Names since both of the PLCs use tag.

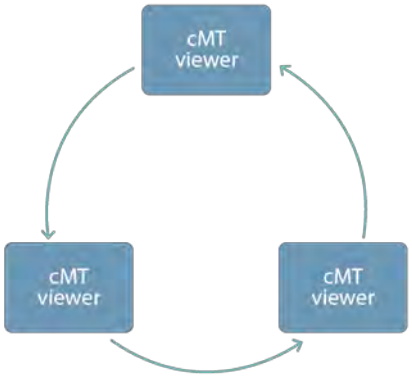
Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

34. Control Token (cMT / cMT X)

This chapter explains how to use Control Token.

34.1. Overview 34-2
34.2. Configuration 34-2
34.3. Demonstration 34-4

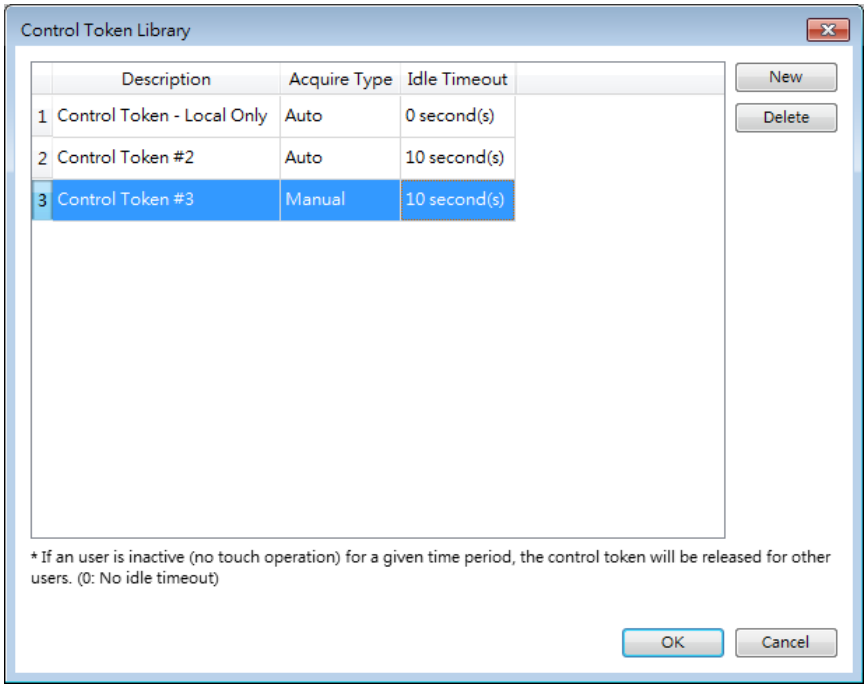
34.1. Overview



One cMT / cMT X HMI can be simultaneously controlled by multiple cMT Viewer clients. To ensure system safety by preventing an object to be controlled by multiple clients simultaneously, a control token can be set. Only one cMT Viewer client can acquire the control token at a time, and only the cMT Viewer client that acquires control token can operate the object. The rest of the clients can acquire the token one by one when the token is not occupied.

34.2. Configuration

Click [Project] » [Control Token] to open the settings window.



Setting	Description
Description	Specify the name of the selected control token. **Control Token – Local Only** When this token is selected, only the local HMI can operate the object. This token is available only for models that have built-in cMT Viewer: cMT3072, cMT3090, cMT3151, cMT-HDM…etc.
Acquire Type	Select an acquire type from [Auto] or [Manual]. **Auto** With this option selected, cMT Viewer will automatically acquire the control tokens set for all objects in the current window. The control token will be released after leaving this window. **Manual** The control token can be acquired or released by triggering an Action Trigger or a Combo Button.
Idle Timeout	Setting Idle Timeout helps avoid the issue that prevents other users from gaining operating right when the token possessor, having finished operation, does not release the control token. When the token possessor's HMI remains inactive (no touch operation) for a given time period, the control token will be

	released. The setting range allowed for Idle Timeout is 0~86400 seconds. For “Control Token- Local Only”, the Idle Timeout can only be zero and is not adjustable.

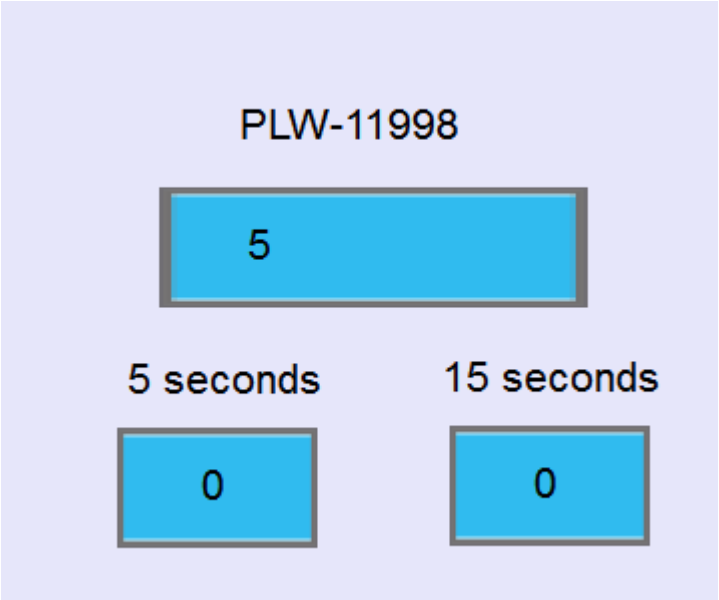
Note

- Control tokens are allocated on a first come first served basis, and a control token can be possessed by one cMT Viewer client at a time. Attempts made for acquiring a control token when it is unavailable will be put into a queue, and it becomes available, it will be passed to the next one in the queue.
- A cMT Viewer client can simultaneously possess multiple control tokens.
- In Auto mode, when changing to a new window, even if the object with the same control token appears in both previous and new window, the control token is reacquired upon entering the new window. If the object with the control token is placed in an overlay window or a common window, and the same overlay window is used in both previous and new window, then reacquiring control token will not take place.
- When cMT Viewer client is disconnected from the Server, and the connection is recovered within 30 seconds, reacquiring control token is not necessary. When the disconnection goes longer than 30 seconds, reacquiring control token is necessary.

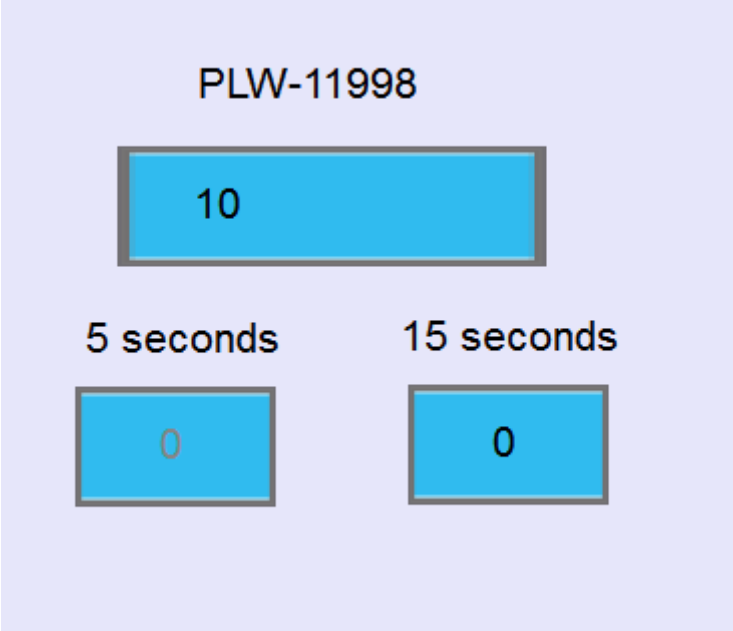
34.3. Demonstration

To monitor idle time countdown, use an ASCII object with system register PLW-11998. When two or more control tokens are used, this system register will count down from least idle time until elapsed.

In the following example, on the top, ASCII object shows idle time countdown from PLW-11998. The bottom two objects use two different control tokens, one with a 5-second idle time and the other 15-second. When running simulation, PLW-11998 will count down from five seconds.



After 5 seconds, the bottom left object turns grey due to expired control token, and another object has 10 seconds left for countdown.



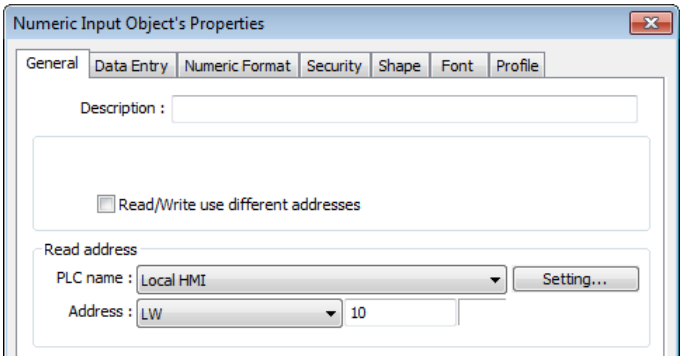
35. EasyWatch

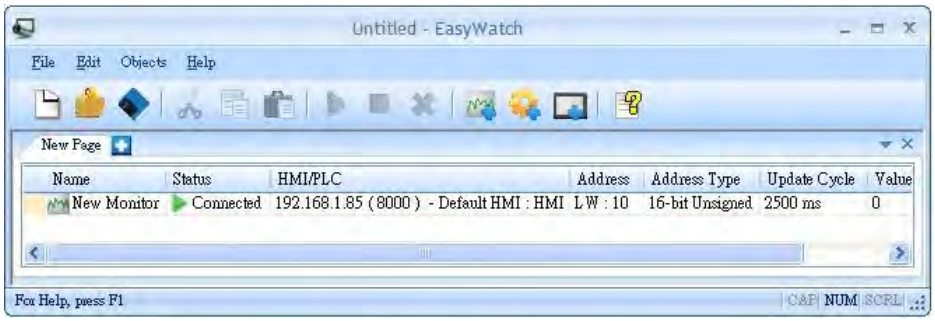
This chapter explains how to use EasyWatch.

35.1. Overview 35-2
35.2. Configuration 35-2
35.3. Monitor Settings 35-4
35.4. Macro Settings 35-8
35.5. HMI Manager 35-9
35.6. Object List 35-11

35.1. Overview

EasyWatch allows users to monitor the HMI or the PLC address values via Ethernet from the PC, or to invoke the Macro for debugging, remote monitoring, and controlling.
For example, In EasyBuilder Pro, after creating a Numeric Input Object, you can set its address to LW-10, and set the same address in EasyWatch. The value of LW-10 will be shown in EasyWatch when successfully connecting.





Note

- When the system register [LB-9044 (disable remote control)] or [System Parameter Settings] » [System Setting] » [Prohibit remote HMI connecting to this machine] is enabled, the feature of monitoring in EasyWatch will be unavailable.

35.2. Configuration

35.2.1. Basic Functions

Item	Description
File	**New:** Opens a new EasyWatch file. **Open:** Opens an existing EasyWatch file. **Save:** Saves an EasyWatch file. **Save As:** Saves an EasyWatch file to .ewt format.

	Exit: Exits EasyWatch.
Edit	**Cut:** Relocates the selected items to the clipboard. **Copy:** Copies the selected items to the clipboard. **Paste:** Pastes the items in the clipboard at the selected location.
Objects	**Add Object**: Adds new Monitor or Macro objects. **Delete Objects**: Selects the objects to be deleted, a dialog box appears, click [Yes] to delete. **Modify Object**: Changes the settings of the selected object. **HMI Manager**: Adds, modifies, or removes HMI settings. **Run**: Executes the selected object. **Stop**: Stops executing the selected object.
Help	**Help Topics:** Reference of how to operate EasyWatch. **About EasyWatch:** EasyWatch version information.

35.2.2. Quick Selection Tools

Item	Description
New	Opens a new EasyWatch file.
Open	Opens an existing EasyWatch file.
Save	Saves an EasyWatch file.
Cut	Relocates the selected items to the clipboard.
Copy	Copies the selected items to the clipboard.
Paste	Pastes the items in the clipboard at the selected location.
Run	Executes the selected object.
Stop	Stops executing the selected object.

Delete	Deletes the selected object.
Add Monitor	Adds a new address monitor.
Add Marco	Adds a new macro.
HMI	Opens HMI Manager to add, modify, or remove HMI settings.
Help Topics	Reference of how to operate the EasyWatch.

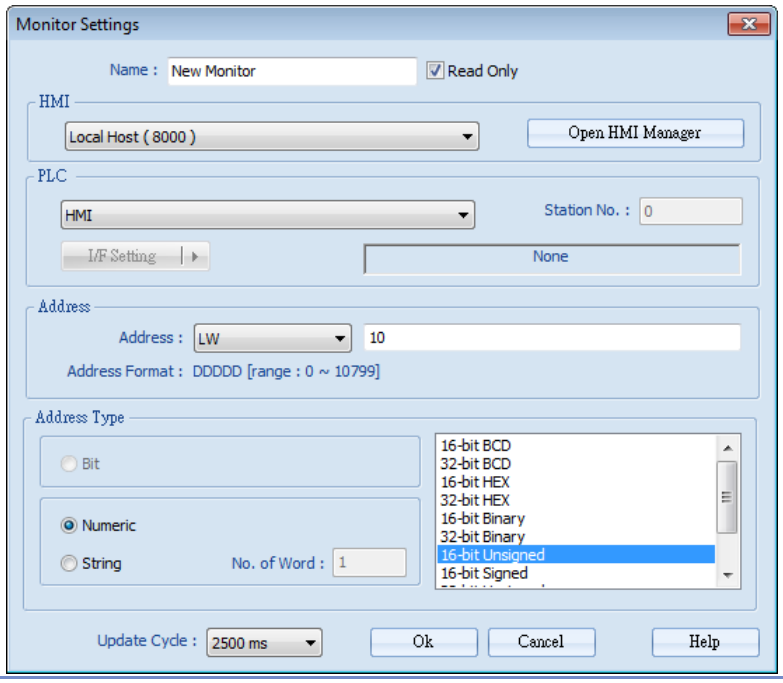
35.3. Monitor Settings

35.3.1. Adding a Monitor

There are two ways to create a Monitor Object.

- Select from the toolbar: [Objects] » [Add Object] » [Add Monitor].
- Select from the quick selection toolbar: [Add new address monitor].

35.3.2. Monitor Settings

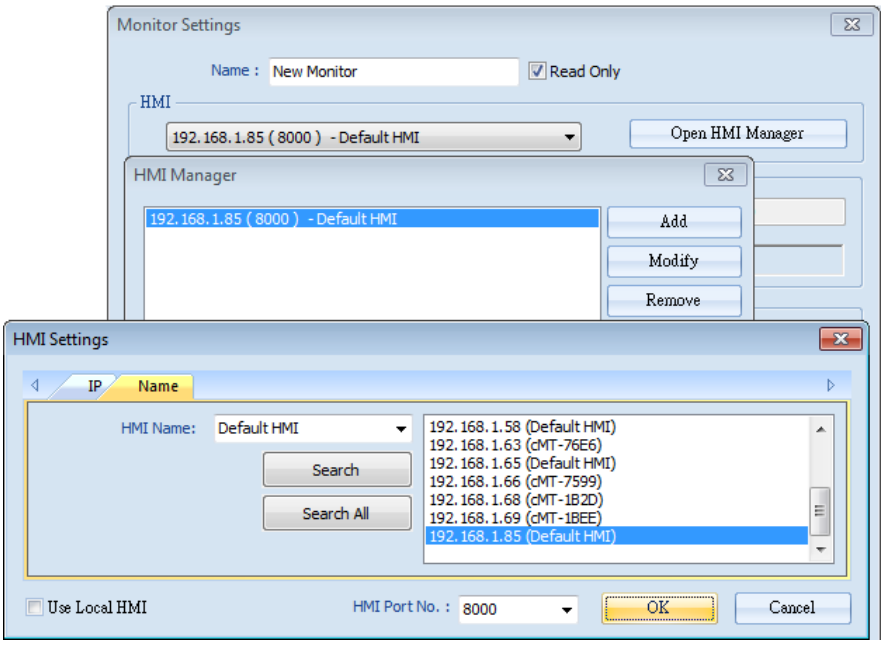


Setting	Description
Name	Enters an object name which is an unique name. **Read Only:** If an object is set to read only, its address value can't be edited.

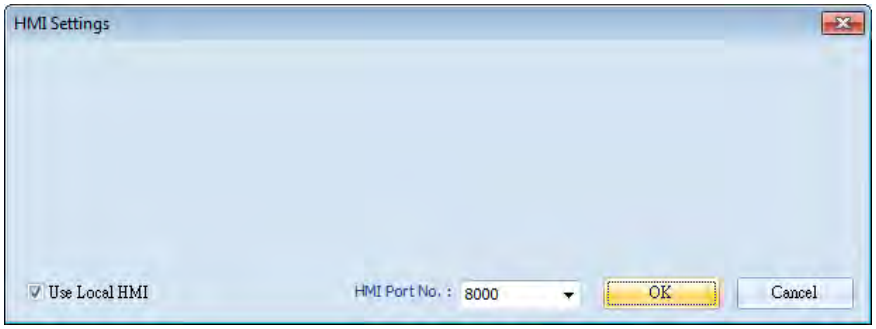
HMI	Select a HMI to monitor.
PLC	Selects the desired PLC to monitor and sets its type, station number, and connection method.
Address	Selects the desired object address to monitor and sets its address type.
Address Type	When the address type is set, the available formats of the address can be selected. When executing, the address will bedisplayed according to the selected format.
Update Cycle	Sets the update interval of the monitoring object. If many objects are executed simultaneously, the errors or delays may appear.

35.3.3. Steps to Add a New Monitor

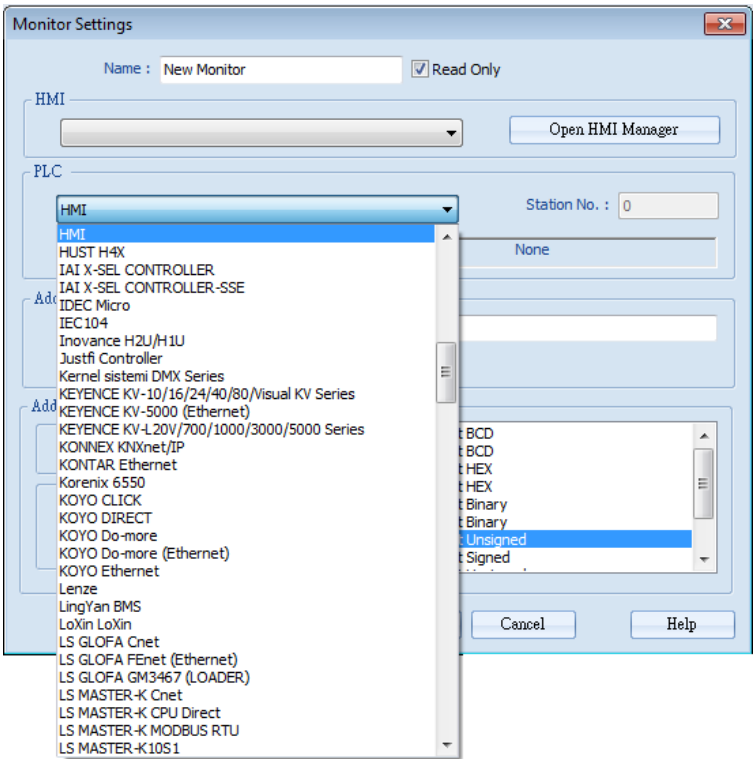
1. Select a target HMI. If the target HMI does not exist, click [Open HMI Manager] and click [Add] to search the HMI for adding.



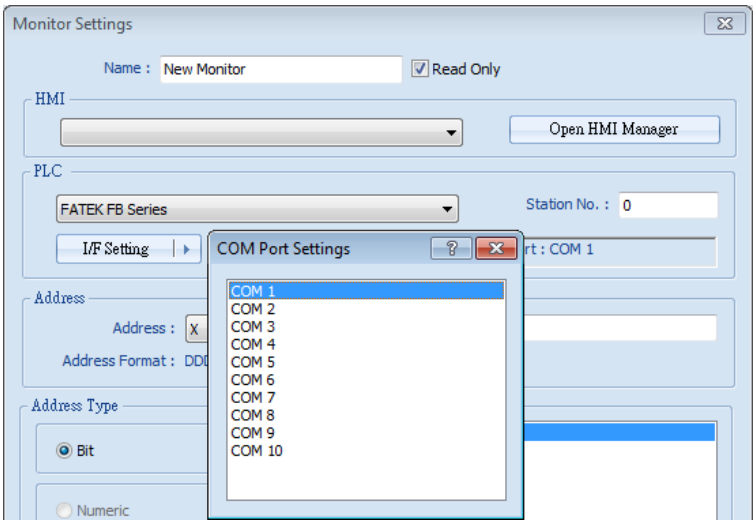
Or, select [Use Local HMI] checkbox to use the project on PC as a monitoring device.



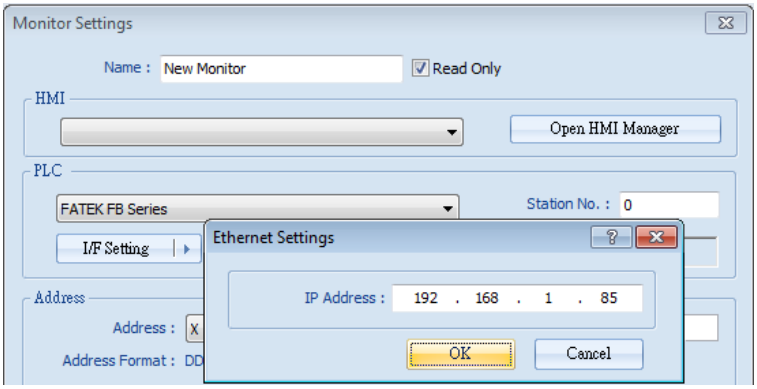
2. Select a target HMI or PLC. If one HMI is selected, you can directly control the local HMI.



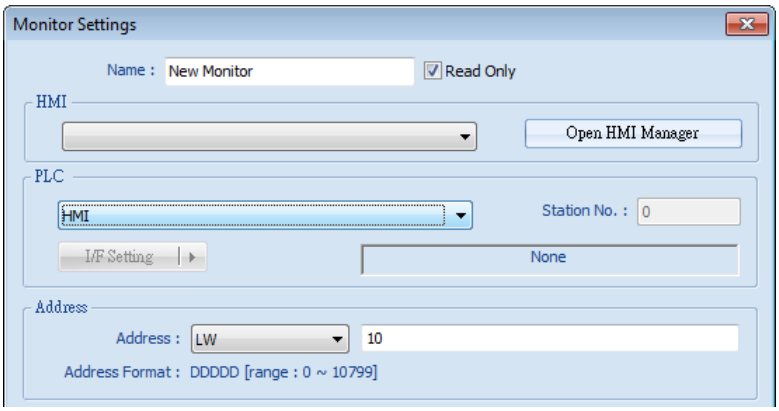
If one PLC is selected, click [I/F Setting] and select [Com Port] to choose a COM port.



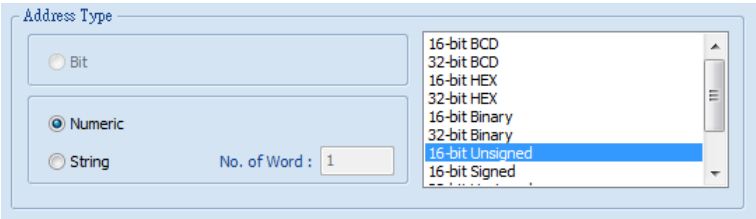
Or, click [I/F Setting] and select [Ethernet] to set the IP address.



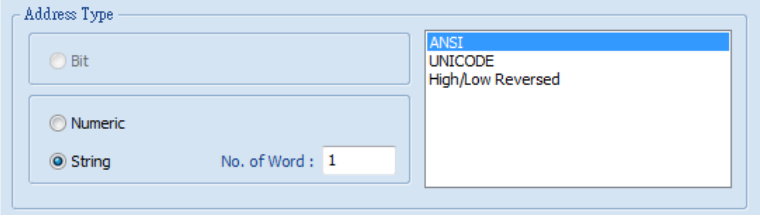
3. Set the object address and its address type for being monitored.



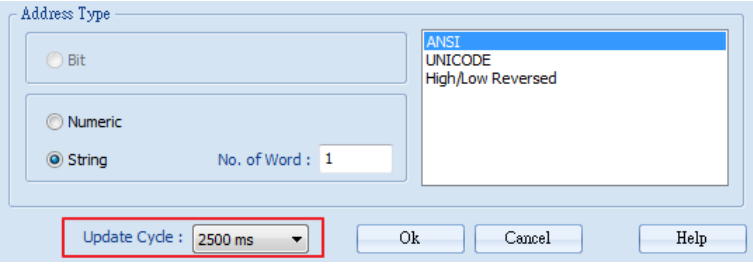
4. When a Word address is selected, set the address type to [Numeric] or [String].
[Numeric]: Select the data format of the monitor address.



[String]: Select the data format from [ANSI], [UNICODE], and [High/Reversed]. Set [No. of Word] for reading the number of WORD.



5. Set the update interval of the monitor object. The range can be set from 500ms to 5000ms.



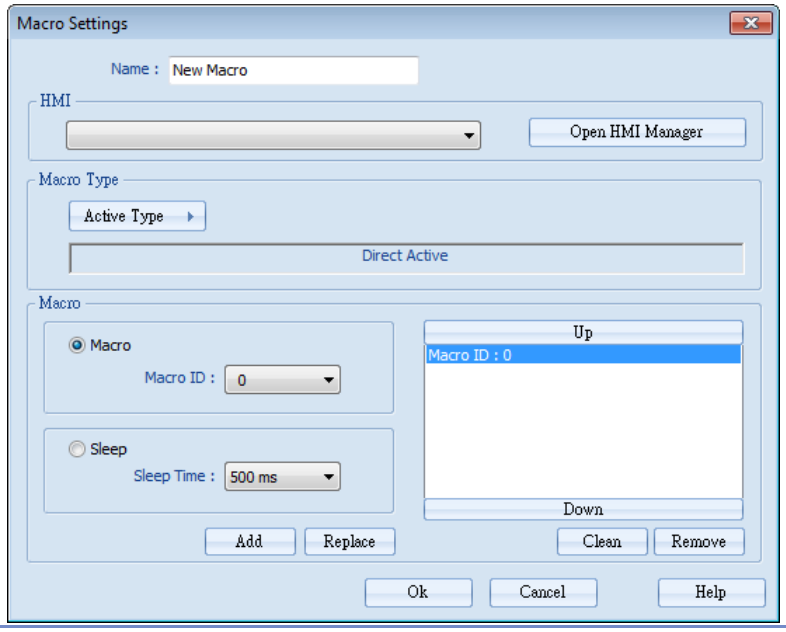
35.4. Macro Settings

35.4.1. Adding a Macro

There are two ways to create a Macro object.

- Select from the toolbar: [Objects] » [Add Object] » [Add Macro].
- Select from the quick selection toolbar: [Add new macro].

35.4.2. Macro Settings

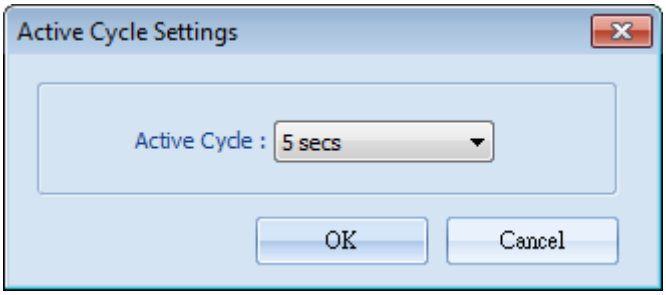


Setting	Description
Name	Enters an object name which is an unique name.
HMI	Selects a HMI to monitor.
Macro Type	The ways to execute the Macro included Direct Active and Cycle Active.
Macro	Each Macro Object can execute multiple macros. The time interval between the executions of two macros can be changed.

35.4.3. Steps to Add a New Macro

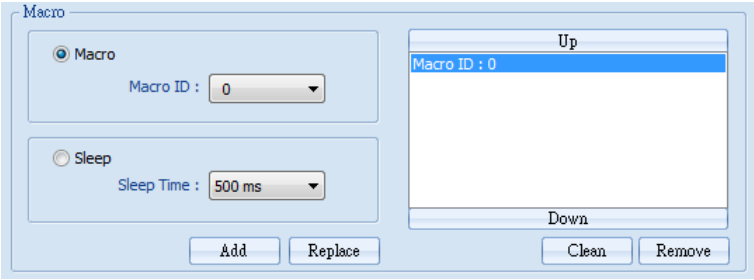
1. Select a target HMI. If the target HMI does not exist, add a new device, see “35.3.3 Steps to Add a New Monitor”.
2. Set Active Type under Macro Type to [Direct Active] or [Cycle Active].
   [Direct Active]: Directly executes Macro once.
   [Cycle Active]: Set the interval of executing Macro.

For example, if [Cycle Active] is set to 5 seconds, when executing a macro, the next time to execute the macro object will be 5 seconds later.

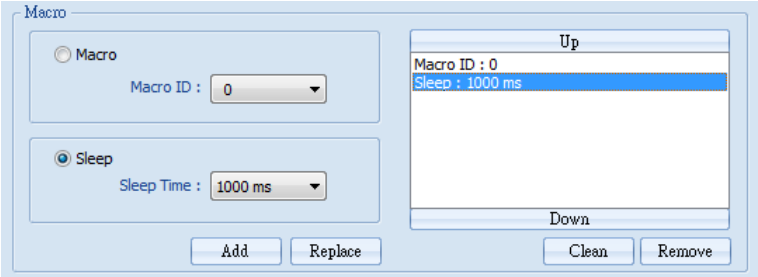


3. Set Macro to [Macro] or [Sleep].

[Macro]: Select the Macro ID for execution, and click [Add] to add the Macro to the list.



[Sleep]: Set the time interval between the executions of two Macros. Click [Add] or [Replace] to add or replace the Macros from the list.



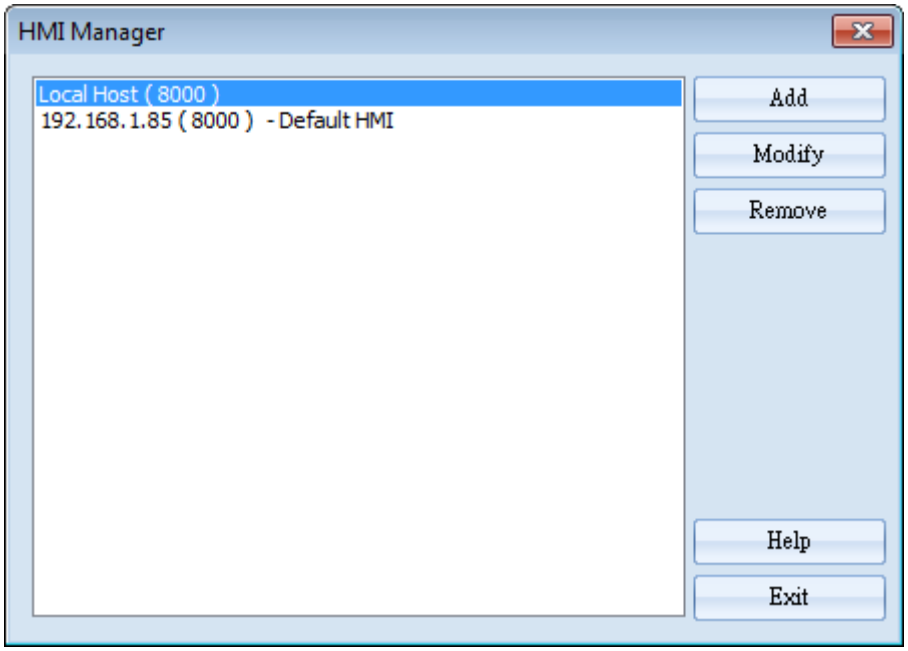
35.5. HMI Manager

35.5.1. Opening HMI Setting

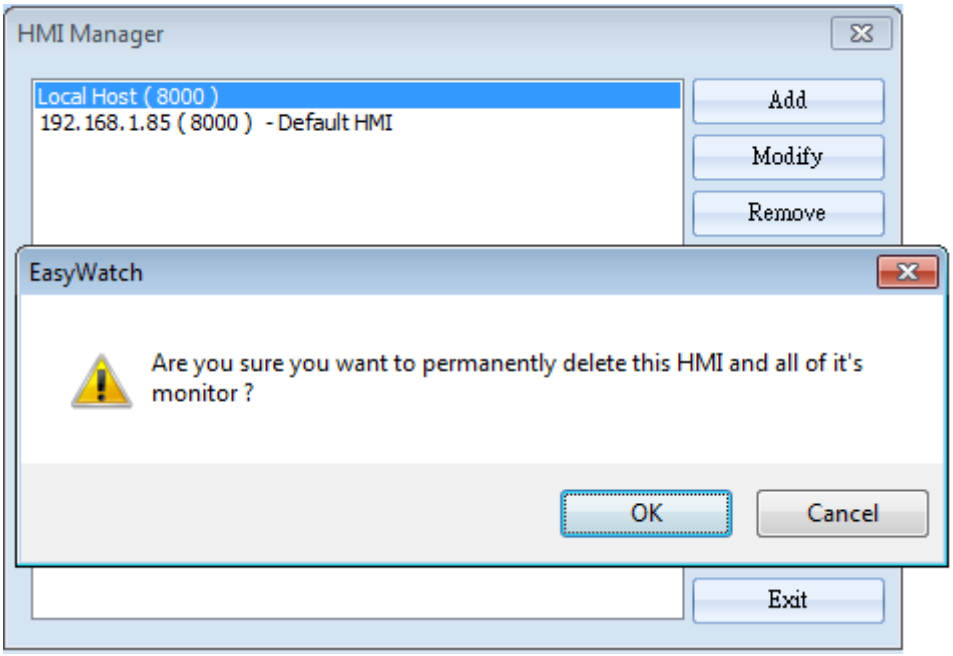
There are two ways to open the HMI Setting.

- Select from the toolbar: [Objects] » [HMI Manager].
- Select from the quick selection toolbar: [Open HMI manager].

35.5.2. HMI Manager



Setting	Description
Add	Select a target HMI. If the target HMI does not exist, add a new device, see “35.3.3 Steps to add a new monitor”.
Modify	Modify the settings of the HMI.
Remove	Remove the HMI.



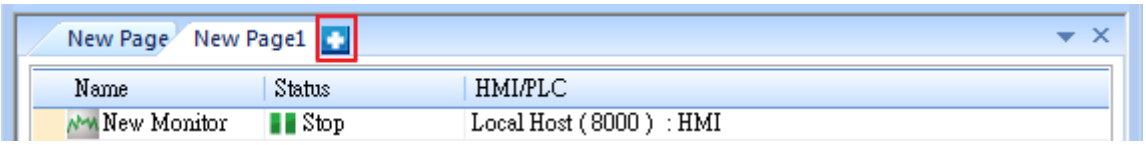
35.6. Object List

35.6.1. Object List Columns

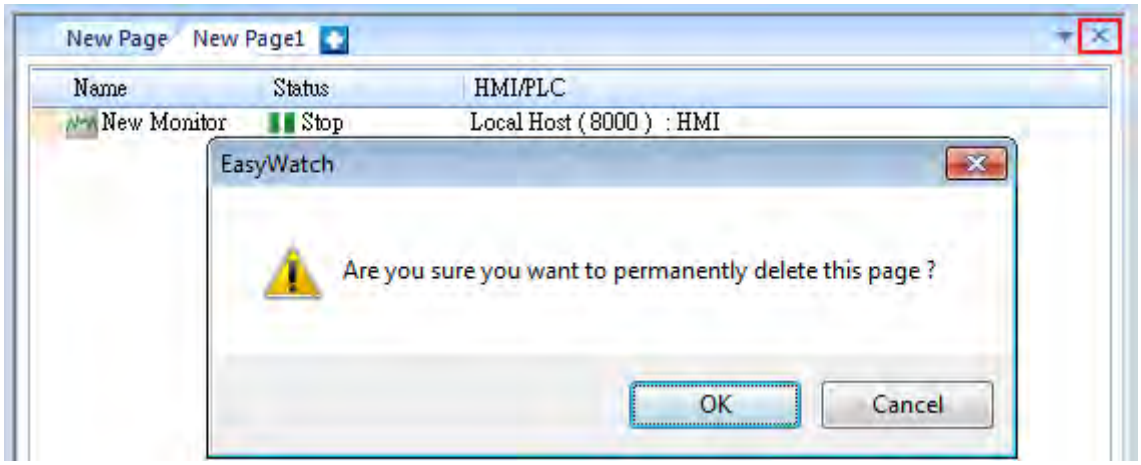
Setting	Description
Name	Displays the object name. The small icons of the names are for users to identify the objects.
Status	Displays the status of the objects: [Connecting], [Connected], and [Stop]. If HMI is not connected or Port No. is incorrect, the error message “HMI Not Found” will appear. For Monitor Objects, if the address is incorrect, “Address Error” message will appear.
HMI / PLC	Displays the information of HMI / PLC that is currently operated by the objects.
Address **Address Type**	For Monitor Objects, the relevant address setting is displayed.
Update Cycle	Set the update interval of the monitor object.
Value	For Monitor Object, if the status shows [Connected], the current HMI address value will be displayed. Modifying the value is also available when the Read-Only checkbox is unselected. For Macro Object, if set to [Direct Active], there will be an [Active] button in this column for clicking to directly execute a macro.

35.6.2. Editing Object List

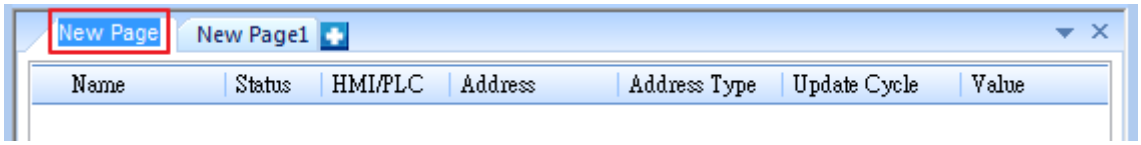
- Adding a new page: Click the icon to add a new page.



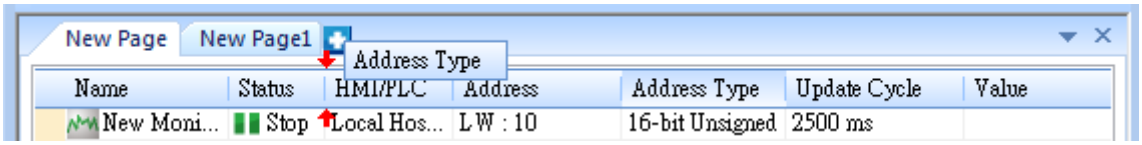
- Deleting a page: Click the icon and confirm the deletion.



- Renaming the page: Double-click on the page name and enter a new name.



- Positioning the column headers: Drag and drop the column headers to the desired location.



Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

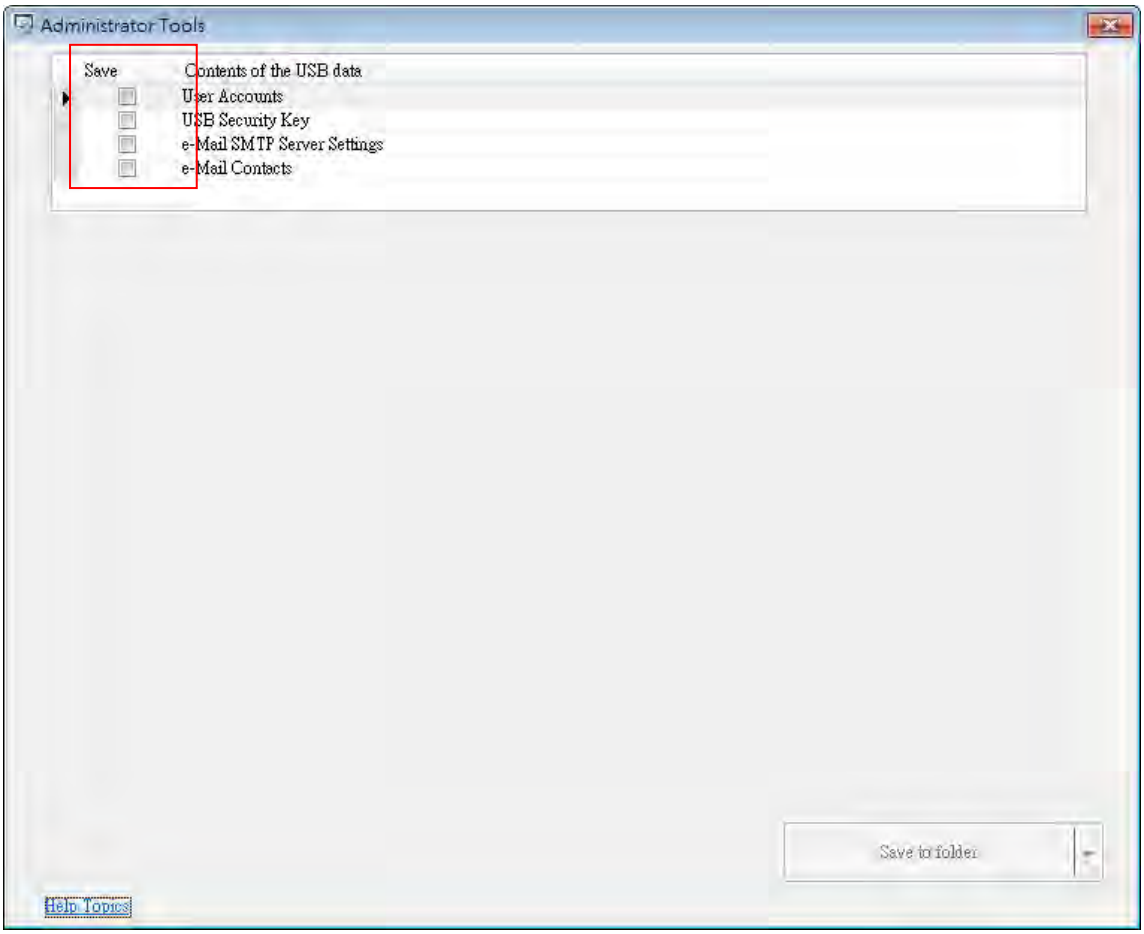
36. Administrator Tools

This chapter explains how to setup Administrator Tools.

36.1. Overview 36-2
36.2. User Accounts 36-2
36.3. USB Security Key 36-6
36.4. e-Mail SMTP Server Settings 36-9
36.5. e-Mail Contacts 36-10

36.1. Overview

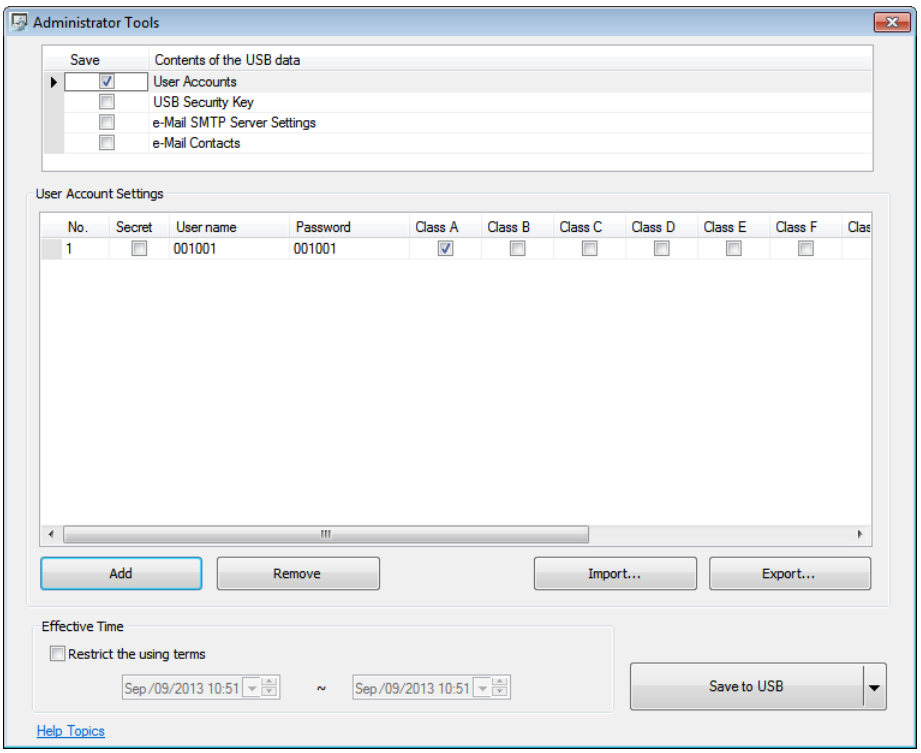
Administrator Tools allows storing the data of [User Accounts], [USB Security Key], [e-Mail SMTP Server Settings], and [e-Mail Contacts] to USB. With EasyBuilder Pro user accounts and e-Mail function, the data built can be imported to HMI by Function Key object set to “Import user data / Use [USB Security Key]”. The portability and convenience is greatly improved. Launch Administrator Tools, select the check boxes in [Save] column to enable the selected functions introduced in this chapter.

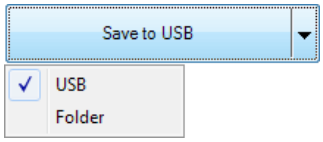


36.2. User Accounts

36.2.1. User Accounts Settings

Select [User Accounts] check box and complete the relevant settings.



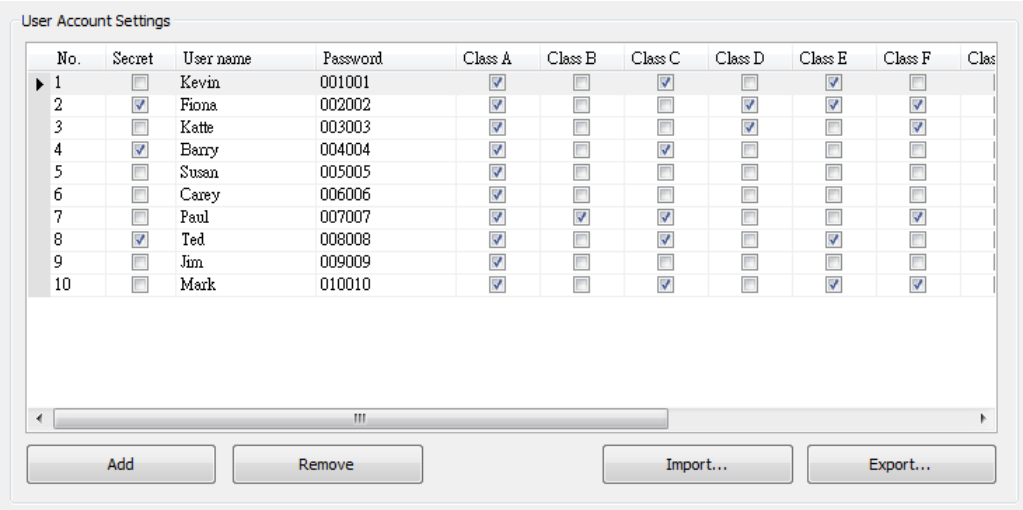
Settings	Description
Secret	Select the check box to set secret accounts.
User Name	Sets User Name. *Note 1
Password	Sets User Password. *Note 1
Class A to L	Sets user privilege.
Add	Adds a new account. *Note 2
Remove	Deletes an existing account.
Import	Imports user account data.
Export	Exports user account data.
Effective Time	If [Restrict the using terms] is not selected, it is available to import data anytime. If select [Restrict the using terms] check box, and set an effective time period, the importing of data to HMI can only be done in the time limit specified. When time expired, data cannot be imported, please reset the data with this tool again.
Save to USB	Saves data to USB. To save to a specific folder, click "▼" button and select "Folder". 

Note

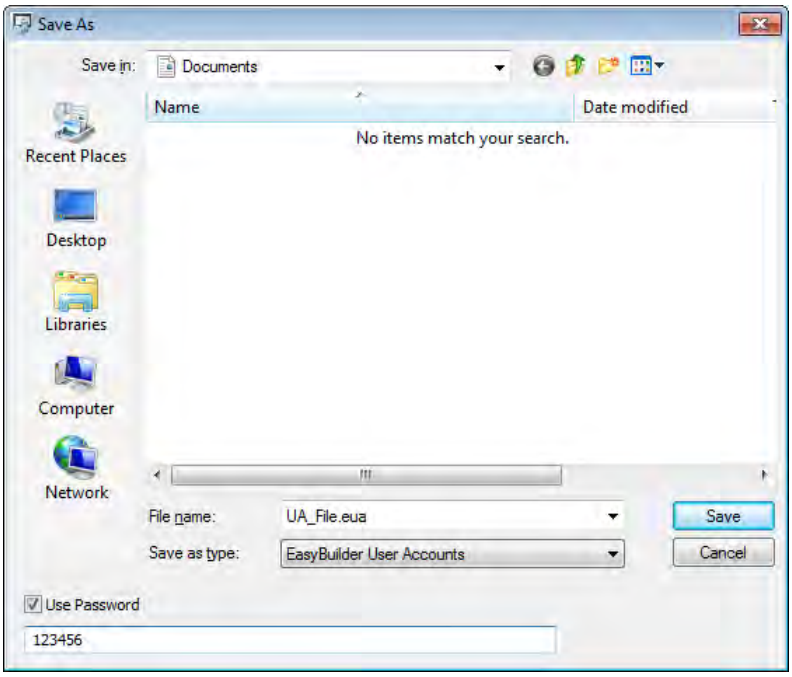
1. Only allows letters, numbers, “-” or “_”, case-sensitive.
2. A maximum of 127 user accounts can be added.

36.2.2. Steps to Set User Accounts

1. Click [Add] to create a new account. Click [Remove] to delete the selected account. Select [Secret] check box to define the account as a secret user. Type in [User name] and [Password] and check the privilege from [Class A] to [Class L] check boxes.

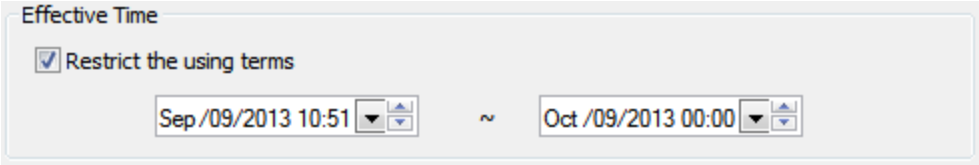


2. To back up the data, click [Export]. Click [Use Password] to protect the data, next time when click [Import] to import the backup data, a password is required.

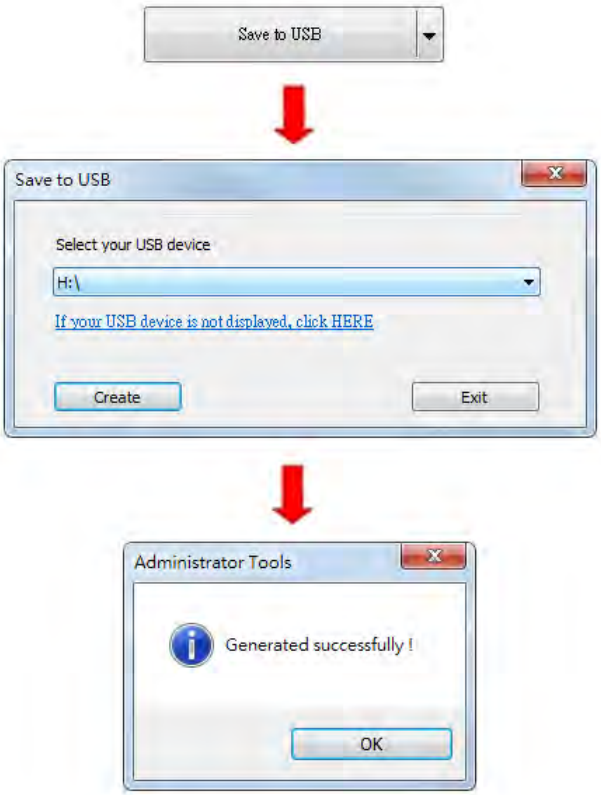


3. If under [Effective Time], the [Restrict the using terms] check box is selected, only during the specified time period can the users import account data to HMI. If not selected, users

can import data to HMI at any time.



4. When finished, click [Save to USB], select the location of USB and then click [Create]. If successful, the "Generated successfully!" message is shown.

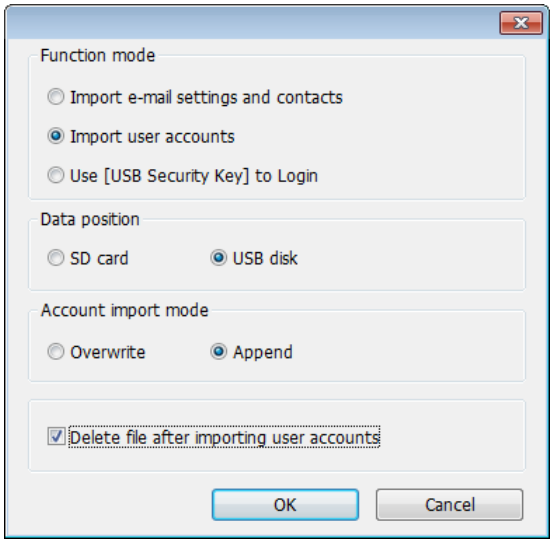


36.2.3. Steps to Import Accounts Using EasyBuilder Pro

The following steps explain how to create a Function Key to import data in EasyBuilder Pro.

1. Select "Import user data/Use [USB Security Key]" in Function Key setting dialog box, and then click [Settings].

2. Under [Function mode] select [Import user accounts]. Select the device that stores the data in [Data position]. Select [Overwrite] under [Account import mode]; HMI will only store the account data imported this time. Select [Append], HMI will store the accounts imported this time and those already exist. Select [Delete file after importing user accounts] check box to delete the source files after importing.

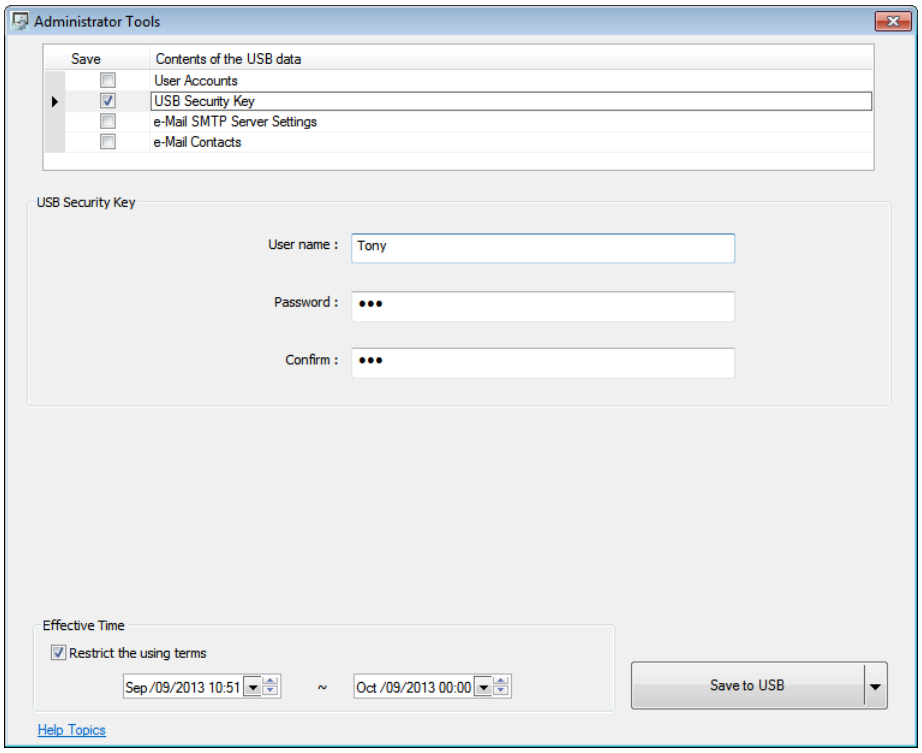


Click the icon to download the demo project that explains how to import user accounts by using Function Key. Please confirm your internet connection.

36.3. USB Security Key

36.3.1. USB Security Key Settings

With the predefined user login information, the USB Security Key can be used to log in directly.



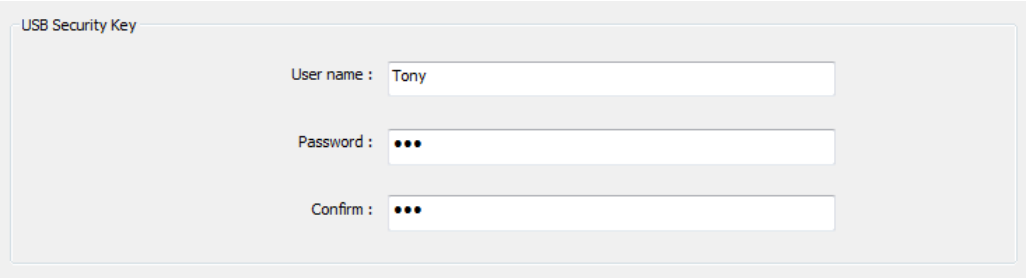
Settings	Description
User Name	Sets User Name. *Note 1
Password	Sets User Password. *Note 1
Confirm	Confirms User Password.
Effective Time	Logs in using USB Security Key during the specified time period. If not specifying Effective Time, log in at any time.
Save to USB	Saves data to USB.

Note

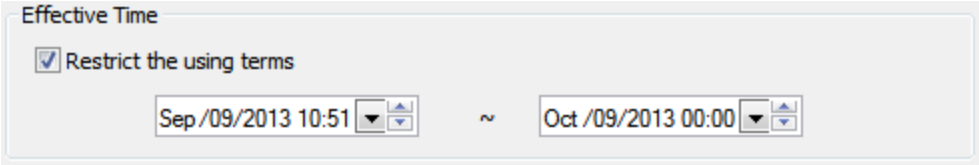
1. Only allows letters, numbers, "-" or "_", case-sensitive.

36.3.2. Steps to Set USB Security Key

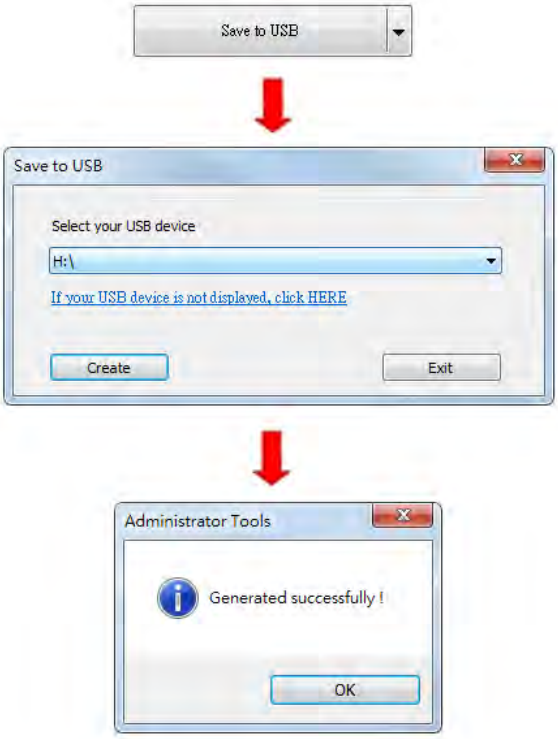
1. Type in the existing user name and password. Type the password again in [Confirm] field for password confirmation.



2. Under [Effective Time] if [Restrict the using terms] check box is selected, only during the specified time period can users log in using USB Security Key. If not selected, users can log in using USB Security Key at any time.



3. When finished, click **[Save to USB]**, select the location of USB and then click **[Create]**. If successful, the "Generated successfully!" massage is shown.

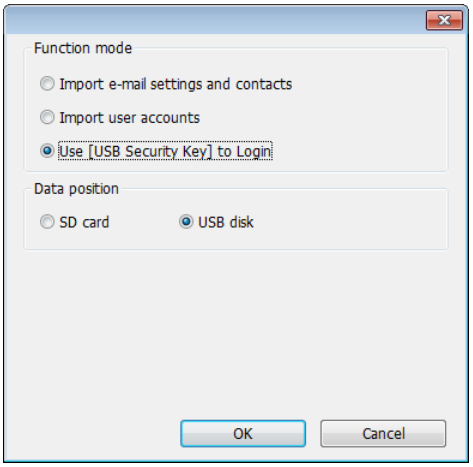


36.3.3. Steps to Set USB Security Key Using EasyBuilder Pro

The following steps explain how to create a Function Key to enable USB Security Key in EasyBuilder Pro. By touching the object, the USB Security Key is enabled for login.

1. Select "Import user data/Use [USB Security Key]" in Function Key setting dialog box, and then click [Settings].

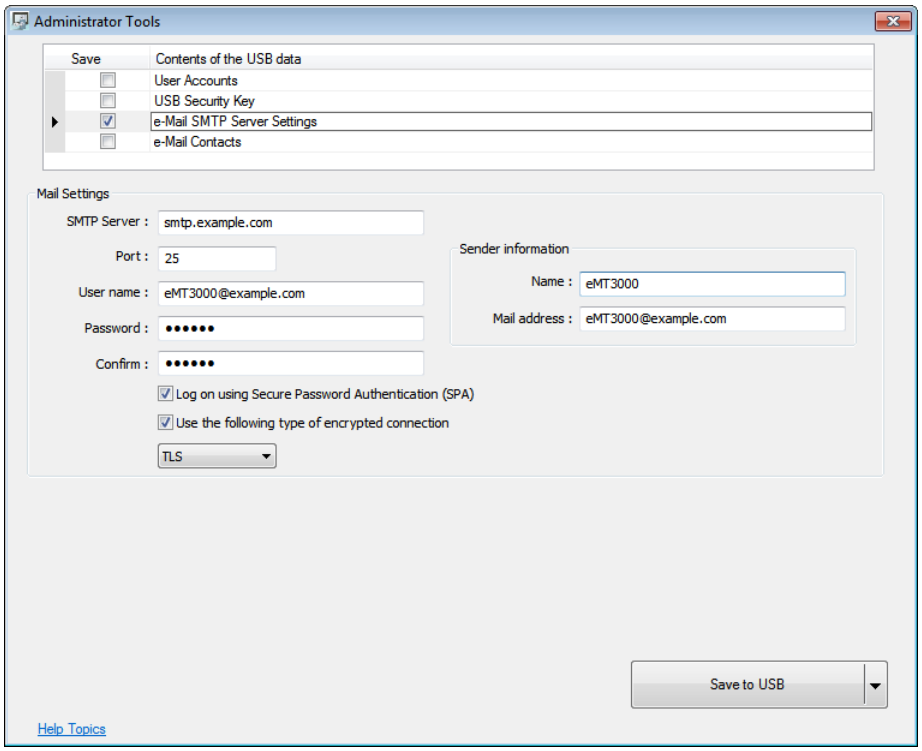
2. Under [Function mode] select [Use USB Security Key to Login]. Select the device that stores the data in [Data position].



Click the icon to download the demo project that explains how to enable login using USB Security Key by using Function Key. Please confirm your internet connection.

36.4. e-Mail SMTP Server Settings

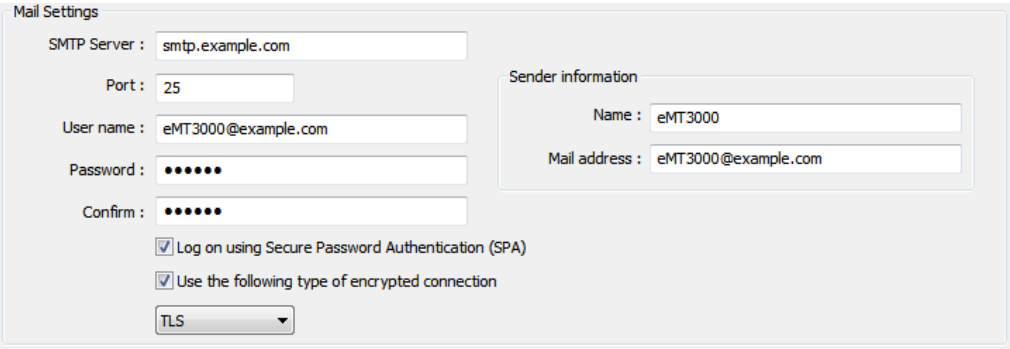
Select [e-Mail SMTP Server Settings] check box to complete the relevant settings.



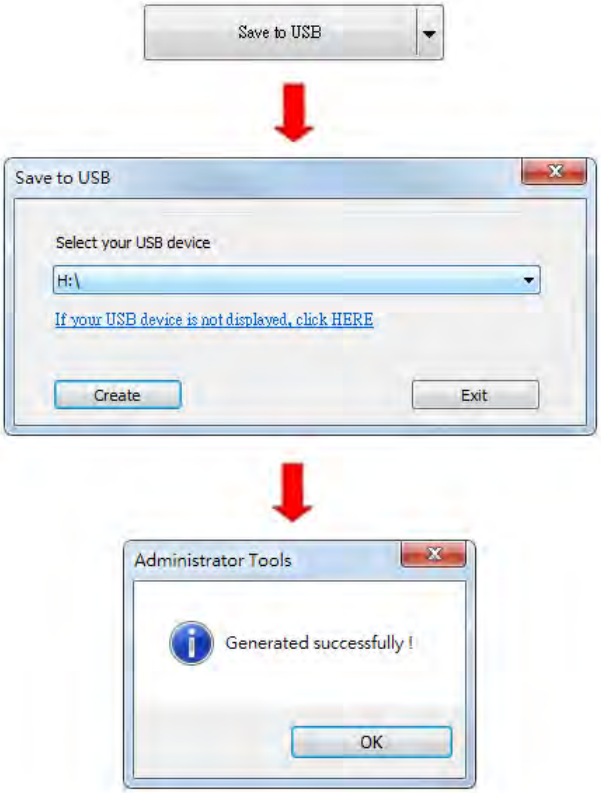
Mail Settings	Description
SMTP Server	Specifies SMTP Server.
Port	Specifies SMTP Server port number.
User name	User e-mail account name.
Password	User e-mail account password.
Confirm	Confirm user e-mail account password.
Sender Information	**Description**
Name	The sender name displayed when mail received.
Mail address	The sender address displayed when mail received.
Save to USB	Saves data to USB.

36.4.1. Steps to set e-Mail SMTP Server Settings

1. Set the settings as shown in the following figure.



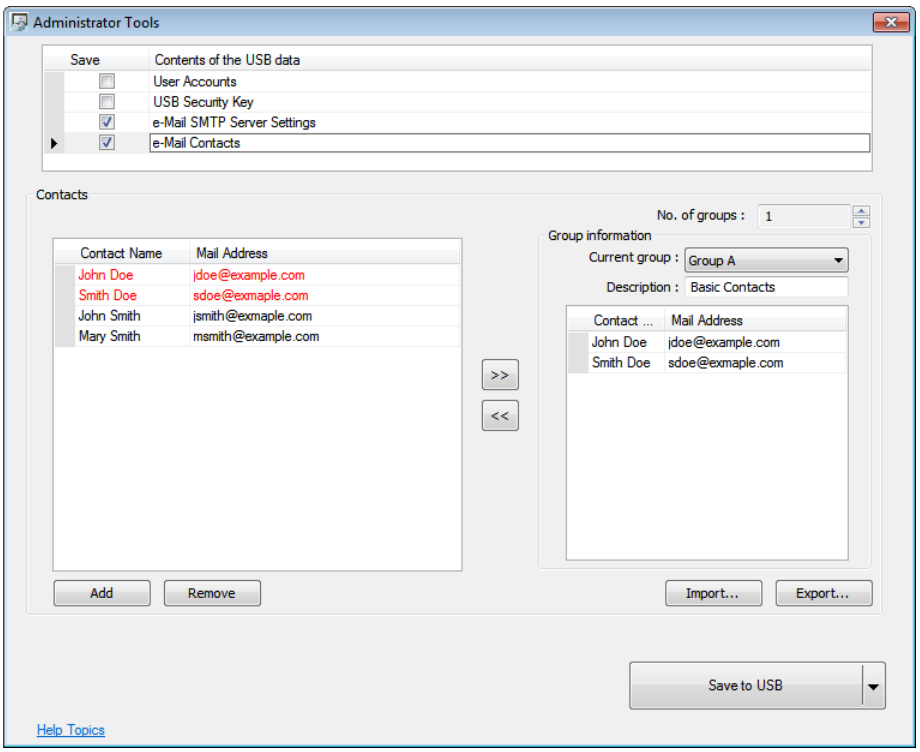
2. When finished, click [Save to USB], select the location of USB and then click [Create]. If successful, the "Generated successfully!" massage is shown.



36.5. e-Mail Contacts

36.5.1. e-Mail Contacts Settings

Select [e-Mail Contacts] check box to complete the relevant settings.



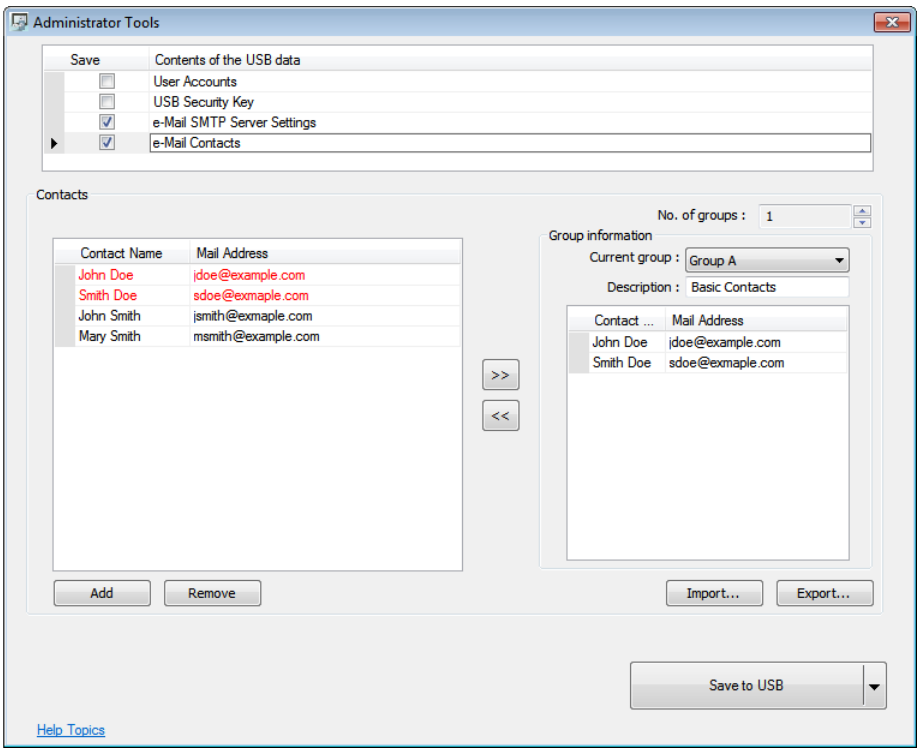
Settings	Description
Add	Adds a new contact. *Note1
Remove	Removes a contact.
No. of groups	The number of groups. *Note2
Current group	The name of current group. *Note3
Description	Group description.
Import	Imports contact information.
Export	Exports contact information.
Save to USB	Saves data to USB.

Note

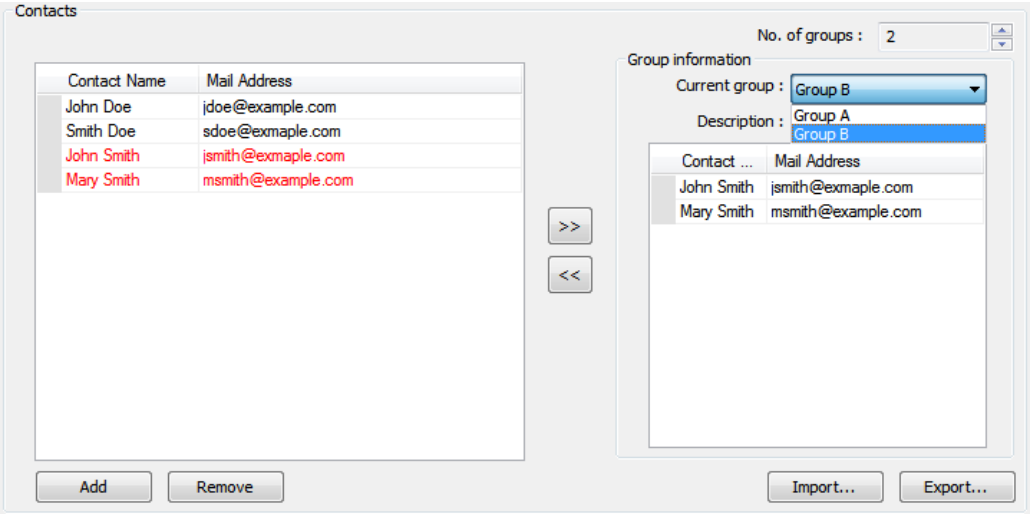
1. A maximum of 256 contacts can be added.
2. A maximum of 16 groups can be added. (Group A to Group P)
3. From Group A to P, when [No. of groups] is “1”, only Group A will exist, when added to “2”, Group A and Group B will exist, and so on.
4. UTF-8 characters are allowed in contact names.

36.5.2. Steps to set e-Mail Contacts

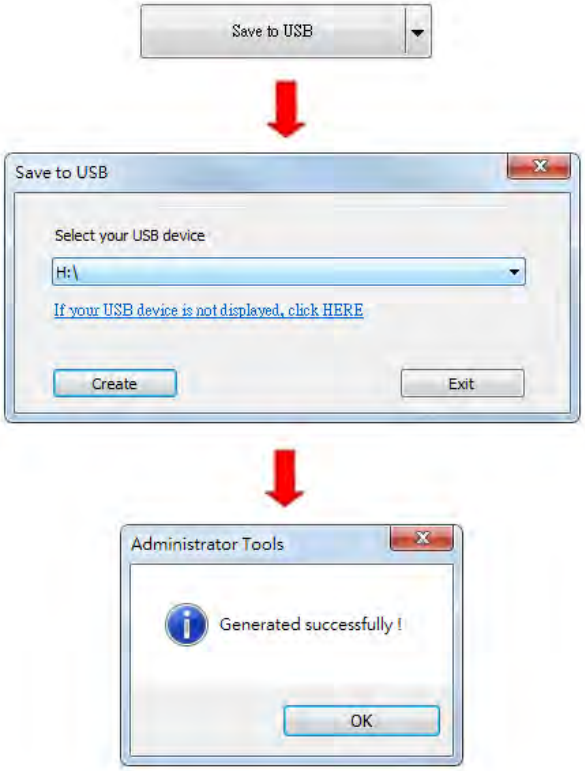
1. Click [Add] to add in all contacts.
2. Add the contacts to Group A, the added contacts are displayed in red font.



3. Press the up or down arrows of the spin box of [No. of groups] to add new groups. If add to “2”, Group B can be found. Repeat step 1 and step 2 to add contacts into groups.



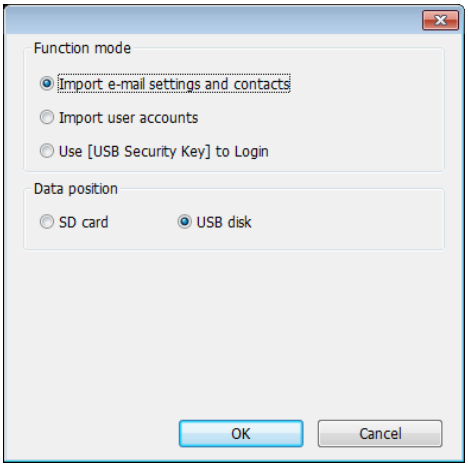
4. After adding all the e-mail contacts, click [Export] to back up the data for future use and modification. Next time when needed, click [Import] to import the backup data.
5. When finished, click [Save to USB], select the location of USB and then click [Create]. If successful, the “Generated successfully!” massage is shown.



36.5.3. Steps to Import e-Mail Settings and Contacts Using EasyBuilder Pro

The following steps explain how to create a Function Key to import e-Mail contacts.

1. Select "Import user data/Use [USB Security Key]" in Function Key setting dialog box, and then click [Settings].
2. Under [Function mode] select [Import e-mail settings and contacts]. Select the device that stores the data in [Data position].



Click the icon to download the demo project that explains how to import e-mail settings and contacts by using Function Key. Please confirm your internet connection.

37. MODBUS Gateway

This chapter explains how to use MODBUS Gateway and configure address mapping tables.

37.1. Overview 37-2
37.2. Steps to Create an Address Mapping Table 37-2
37.3. Notes about Configuring Address Mapping 37-5

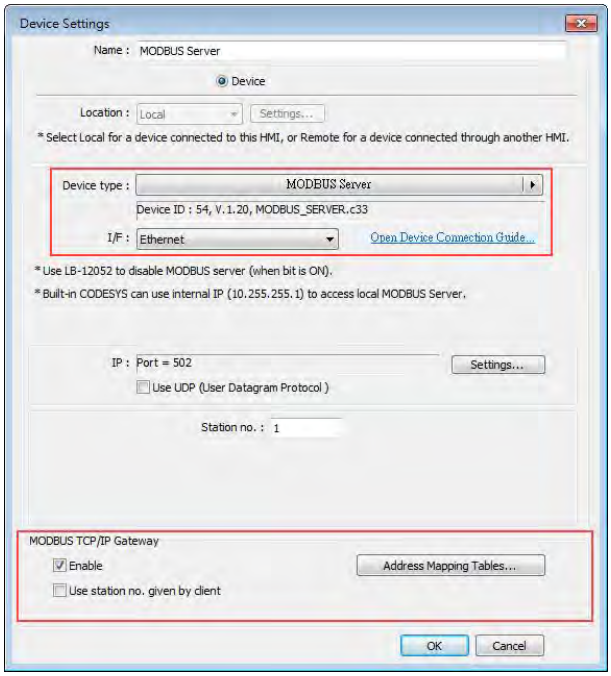
37.1. Overview

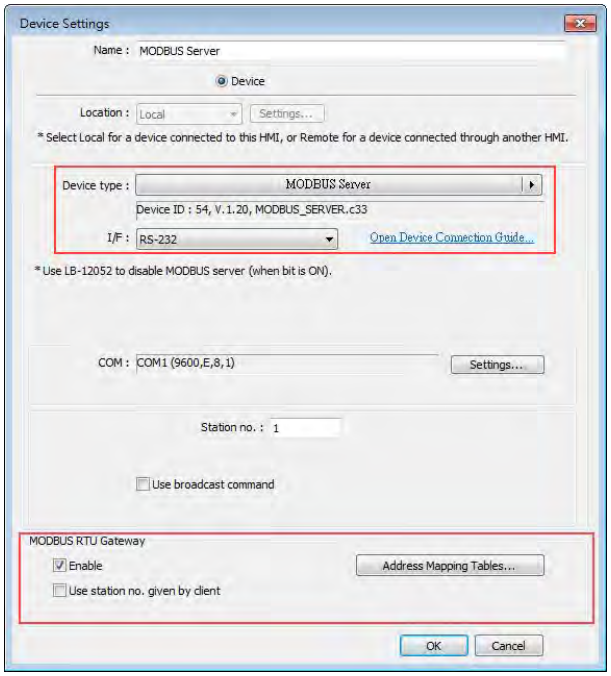
To access the data of the PLC connected to HMI with SCADA software (Supervisory Control and Data Acquisition), the former way was to transfer PLC data to the HMI's local address first, and then use MODBUS TCP/IP or MODBUS RTU protocol on PC to read HMI local address to get PLC data. Now by using MODBUS Gateway provided by EasyBuilder Pro, the mapping of MODBUS address to PLC address can be defined first, and then one can directly use MODBUS TCP/IP or MODBUS RTU protocol to access PLC data.

37.2. Steps to Create an Address Mapping Table

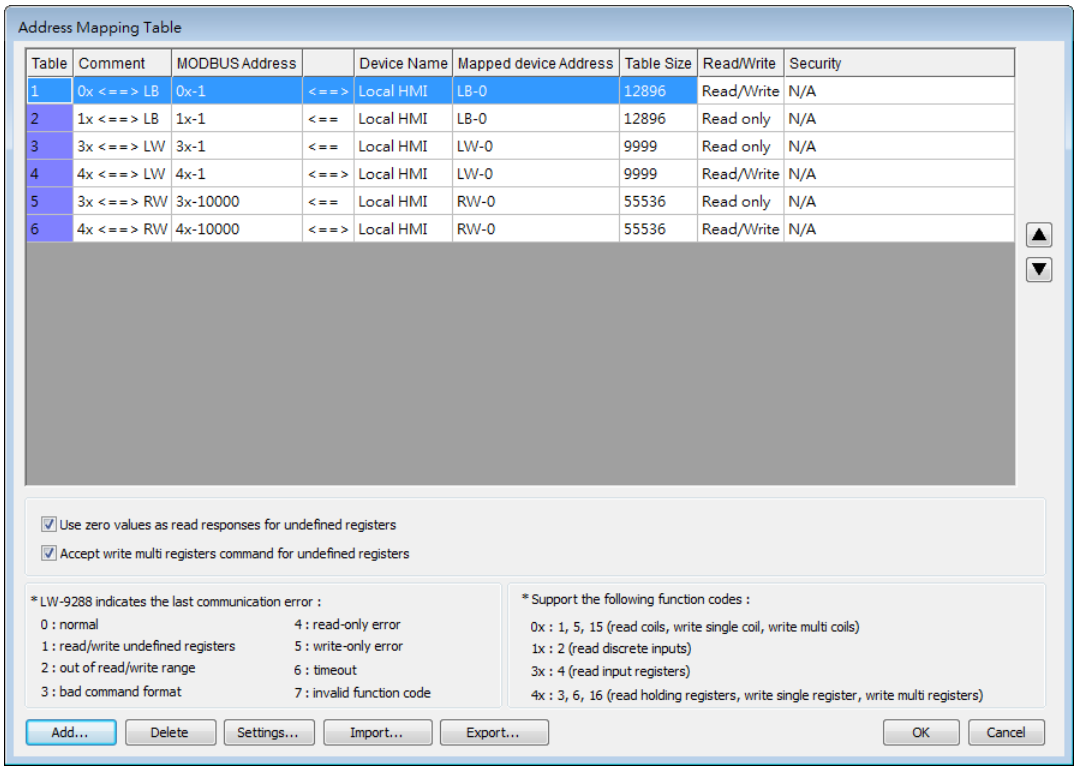
To create an Address Mapping Table, please follow the steps:

1. In [System Parameter Settings] » [Device] tab, add the PLC device. (In the example FATEK FB Series is used).
2. Add MODBUS Server, select Ethernet or COM port interface according to the MODBUS Client and enable [MODBUS TCP/IP Gateway] / [MODBUS RTU Gateway] as shown below.

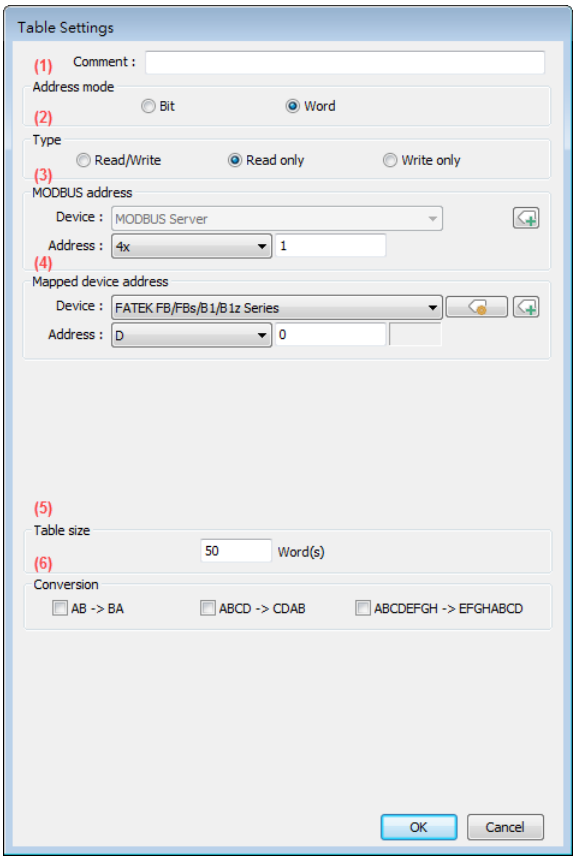




3. Click [Address Mapping Tables] button and the following default tables will be displayed. Modify the tables if needed or add new tables.



4. In the example below, to access the data in the 50 consecutive registers of FATEK FB Series PLC starting from register D-0, configure the settings as shown in the following figure.



(1) Select the address mode, in this example select [Word].
(2) Select the access type, in this example select [Read/Write].
(3) Set the start address of MODBUS, in this example set to “4x-1”.
(4) Set the start address of the mapped PLC, in this example set to “D-0”.
(5) Set the size of address mapping table, in this example set to “50”.
(6) Select high/low byte swap or high/low word swap if necessary.

Table	Description	MODBUS Address		PLC Name	Mapped PLC Address	Table Size	Read/Write
1	Access D0 ~ D49	4x-1	<==>	FATEK FB Series	D-0	50 Word(s)	Read/Write

The above figure shows that MODBUS Server 4x-1 ~ 4x-50 registers are mapped to FATEK FB Series PLC D-0 ~ D-49 registers.

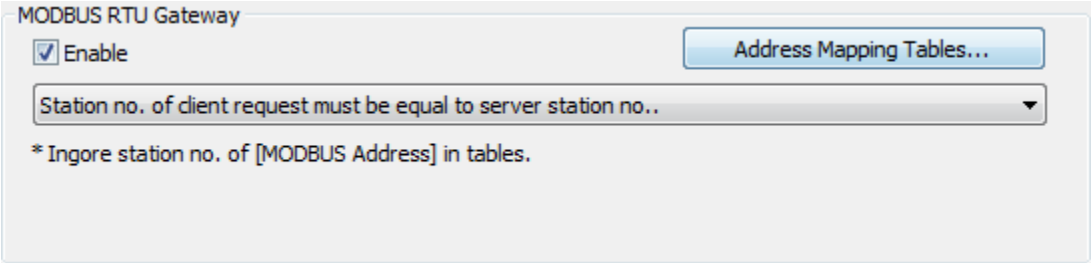
5. When finished, the data of FATEK FB Series PLC D-0 ~ D-49 registers are now accessible by using MODBUS protocol to send read / write command to 4x-1 ~ 4x-50 registers.

37.3. Notes about Configuring Address Mapping

- UDP is not supported when MODBUS TCP/IP Gateway feature is used.
- MODBUS RTU Gateway feature is only supported on cMT / cMT X Series models.
- The station number of MODBUS client request can be handled differently in the following modes, please select a mode suitable for your application:

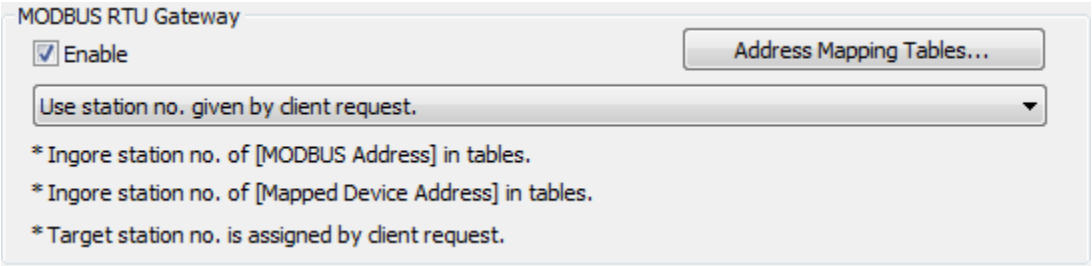
  The station no. of client request must be equal to server station no.

  Use the station number of MODBUS Server.



  Use station no. given by client request.

  Use the station number of MODBUS Client. Only one MODBUS Client can be connected to MODBUS Server at a time, and the clients should use different station numbers.

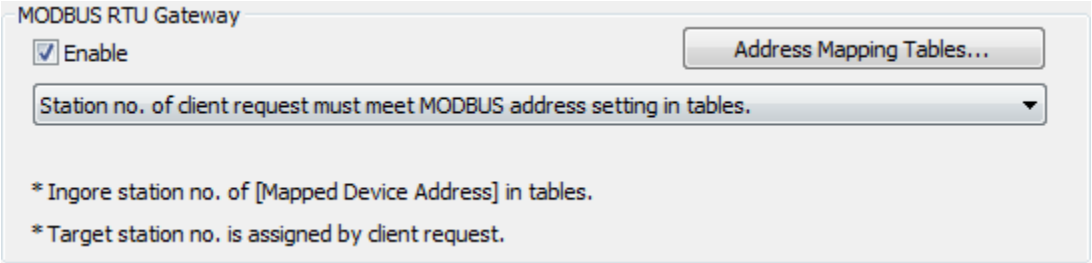


  Station no. of client request must meet MODBUS address setting in tables.

  Use the station number of the MODBUS address in the address mapping table.

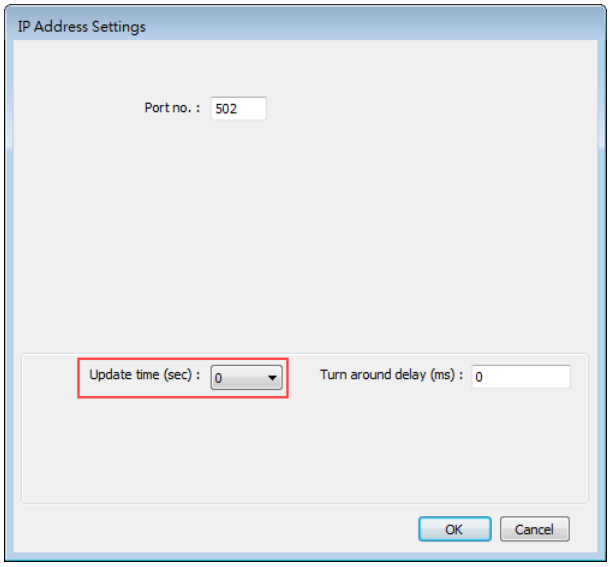
  Example: Two MODBUS Clients, 4x-1#100 and 4x-2#100, are connecting to MODBUS Server using station no. 1 and station no. 2. Under this mode, multiple clients can simultaneously connect to the server.

  If no station number is assigned for the MODBUS address, the default station number of the device will be used.

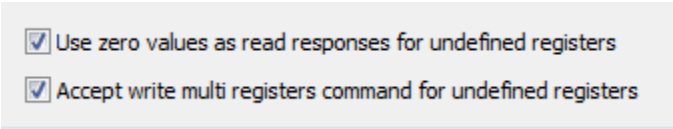


- Update time (sec) can be set to enable caching for MODBUS Server when multiple clients

are frequently reading data. By default, the setting is 0 and the caching feature is not enabled. Settings greater than 0 can enable caching. At the first time the server receives a command from the client, it gets data from PLC. If the update time is set to 1 second, and the client requests for data in the same register again, within a second, the server will use the PLC data previously obtained as response.



- When the registers of the client device are not consecutive, and the user would like to read / write all data at a time, the following two options are provided for users to decide how to handle the undefined registers in the mapping table.



- The clients are not allowed to read / write 16-bit data (high / low word) of a 32-bit register.
- System register LW-9288 is used to indicate if data transfer has been correctly executed. The following error codes represent:

Value	Definition
0	Normal
1	Read or write the register that is not defined in the Address Mapping Table.
2	Read or write a range of registers that is not within the range defined in a single Address Mapping Table. (Or, read / write a register that is defined in other Address Mapping Table.)
3	The command format does not follow MODBUS TCP/IP protocol.
4	Modify a read-only register.

5	Read a write-only register.
6	Cannot get the correct reply from PLC within the specified time range.
7	Use a function code that is not supported by MODBUS Server.

- If [MODBUS Gateway] is enabled, EasyBuilder Pro will cancel the original mapping between MODBUS Server and HMI register. That includes:
  (1) 0x, 1x mapped to LB
  (2) 3x, 4x mapped to LW, RW
  Therefore, to access data in LB or LW register via 0x, 1x, 3x, 4x, configure the Address Mapping Table again. The following figure is an example.

Table	Description	MODBUS Address		PLC Name	Mapped PLC Address	Table Size	Read/Write
1	0x <==> LB	0x-1	<==>	Local HMI	LB-0	12400 Bit(s)	Read/Write
2	1x <==> LB	1x-1	<==	Local HMI	LB-0	12400 Bit(s)	Read only
3	3x <==> LW	3x-1	<==	Local HMI	LW-0	9999 Word(s)	Read only
4	4x <==> LW	4x-1	<==>	Local HMI	LW-0	9999 Word(s)	Read/Write
5	3x <==> RW	3x-10000	<==	Local HMI	RW-0	55536 Word(s)	Read only
6	4x <==> RW	4x-10000	<==>	Local HMI	RW-0	55536 Word(s)	Read/Write

38. EasyDownload

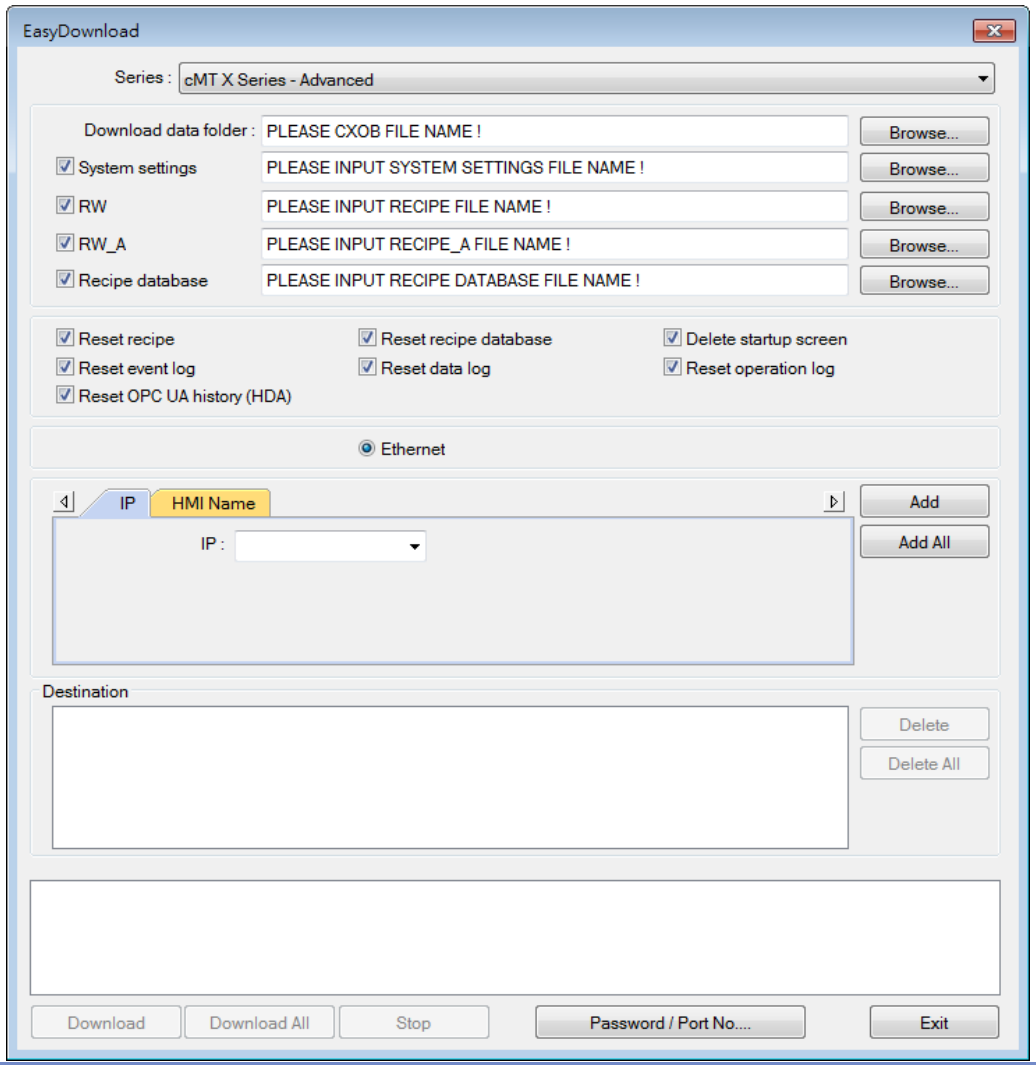
This chapter explains how to setup EasyDownload.

38.1. Overview 38-2
38.2. Configuration 38-2

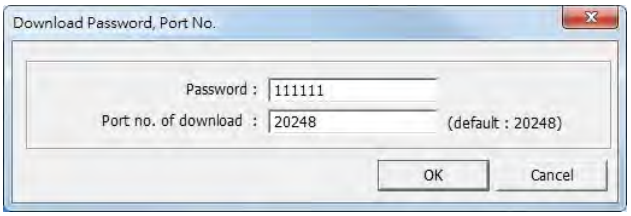
38.1. Overview

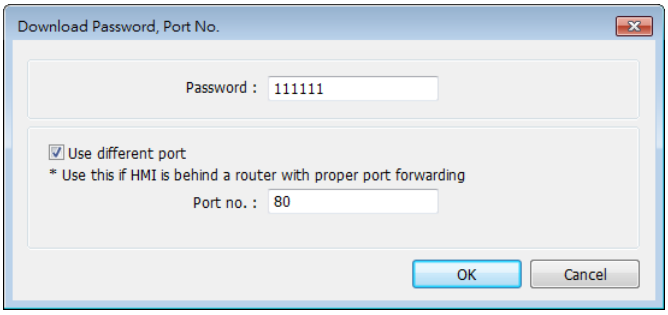
EasyDownload allows downloading the project data file built in EasyBuilder Pro via Ethernet or USB cable. In EasyBuilder Pro main menu select [Project] and then select [Build for SD card/USB disk download] to build the data file before running EasyDownload.

38.2. Configuration



Settings	Description
Series	Select a Series of HMI models.
Download data folder	Browse for the project data file to download. For cMT Series / cMT X Series / IIoT Gateway Series models, select a .cxob file. For non-cMT Series models, select a corresponding folder.
RW	Browse for the recipe data (.rcp) to download.

RW_A	Browse for the recipe data (.rcp) to download.
Recipe database	Browse for the recipe database (.db) to download.
Reboot HMI after download	If selected, HMI will reboot after downloading.
Reset recipe **Reset recipe database** **Delete startup screen** **Reset event log** **Reset data log** **Reset Operation Log**	The selected files will be erased before downloading.
USB cable	Download the file to HMI via USB cable. Please make sure that the USB driver is correctly installed.
Ethernet	Download the file to HMI via Ethernet.
IP	Enter the target HMI IP address.
Name	Enter the target HMI name.
Search	Search by HMI name.
Search All	Search from all the HMI names on the same network.
Add	Add the selected HMI in the Search field to the Destination field.
Add All	Add all the HMI in the Search field to the Destination field.
Destination	A list of all the HMI for download.
Delete	Delete the selected HMI in the Destination field.
Delete All	Delete all the HMI in the Destination field.
Download	Click to start download to the selected HMI in the Destination field.
Download All	Click to start download to all the HMI in the Destination field.
Password / Port No.	Enter the password and the port number of download set in HMI system settings. Non-cMT Series:  cMT / cMT X Seires:



Note

- Take eMT3000 Series as an example, the built download data directory has the following structure. Please select the parent directory of the generated file when download.

Parent directory	First subdirectory	Second subdirectory
emt3000	001	
	002	
	Pub	driver
		font

- The parent directory name changes according to the model used.
- Downloading to multiple HMI is only supported by Ethernet.
- When downloading the project to multiple HMI at a time, all the HMI must use the same password and port number.
- The downloading process goes from the top of Destination list to the bottom. Only when the downloading of one HMI is done will the downloading of next HMI start.
- To avoid long waiting time for an off-line HMI, the wait time is 3 seconds.
- EasyDownload can download project to 512 HMIs at a time; that is, the maximum allowable number of HMI that can be added into the Destination list is 512.

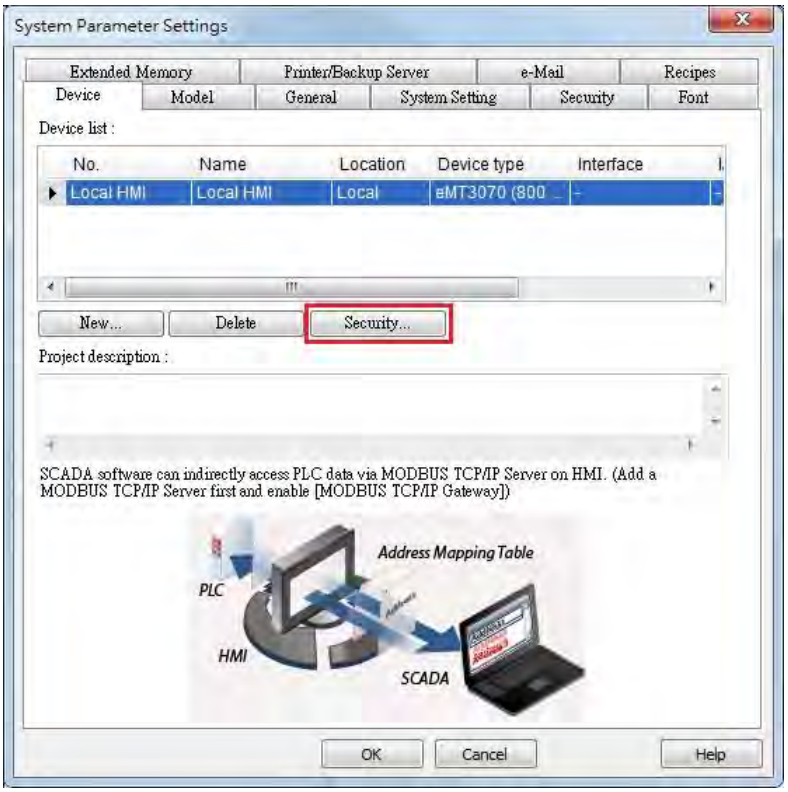
39. Data Security

This chapter explains how to setup Data Security.

39.1. Overview 39-2
39.2. Configuration 39-2

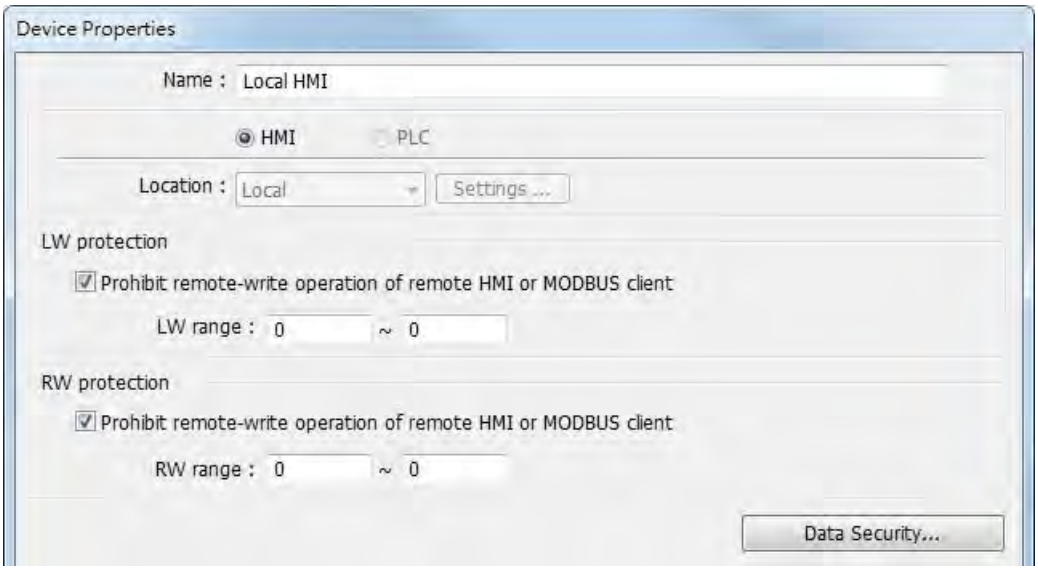
39.1. Overview

Data Security allows setting restrictions on the write operations that modify local Word or Bit register data. To do so, open [System Parameter Settings] » [Device] tab, select [Local HMI] and then click [Security...] button.



39.2. Configuration

The following is the settings dialog box:



Setting	Description
LW protection \ Prohibit remote-write operation of remote HMI or MODBUS client	If selected, a remote HMI or MODBUS client will not be able to write to the specified LW addresses.
RW protection \ Prohibit remote-write operation of remote HMI or MODBUS client	If selected, a remote HMI or MODBUS client will not be able to write to the specified RW addresses.

Click [Data Security] button to set the restrictions on the write operations that modify local Word or Bit register data.

39.2.1. Word address settings

Set the restrictions relevant to local Word addresses.

Setting	Description
Description	Enter the description or memo about this setting.
Address Mode	Select [Word] to set the relevant attributes.
Min. value	Set the minimum value that can be written to the designated word address.
Max. value	Set the maximum value that can be written to the designated word address.
Disable remote-write operation	If selected, the remote HMI will not be able to write to the protected address.
Used only in remote-write operation	If selected, the range between [Min. value] and [Max. value] is only used to restrict the value written by a remote device.
Use min. value when write value is less than min. value	If selected, when the written value is less than [Min. value], the system will write the specified minimum value instead. If not selected, when the written value is less than [Min. value], the system will keep the original value.
Use max. value when write value is more than max. value	If selected, when the written value is greater than [Max. value], the system will write the specified maximum value instead.

	If not selected, when the written value is greater than [Max. value], the system will keep the original value.
Notification	When the written value is not within the specified range between [Min. value] and [Max. value], the system will trigger the designated notification bit address.

As shown in the preceding figure, the remote HMI will not be able to write to LW-0, and when the value written to the local address is greater than 10, the value 10 is written instead, and the notification bit LB-0 will be set ON.

39.2.2. Bit address settings

Set the restrictions relevant to local Bit addresses.

Setting	Description
Description	Enter the description or memo about this setting.
Address Mode	Select [Bit] to set the relevant attributes.
Set ON only	If selected, the designated bit address can only be set ON.
Set OFF only	If selected, the designated bit address can only be set OFF.

Disable remote-write operation	If selected, the remote HMI will not be able to write to the protected address.
Used only in remote-write operation	If selected, the specified condition is only used to restrict the write operation by a remote device.
Notification	When enabled and: ● Select [Set ON], the system will trigger the notification bit address when attempting to set OFF the protected bit address. ● Select [Set OFF], the system will trigger the notification bit address when attempting to set ON the protected bit address.

As shown in the preceding figure, the remote HMI can only set ON LB-10, while the local HMI is not restricted. If the remote HMI attempts to set LB-10 OFF, the system will trigger notification bit LB-0 ON.

40. Web Streaming

This chapter explains how to setup Web Streaming.

40.1. Overview 40-2
40.2. Configuration 40-2
40.3. Operation 40-2
40.4. Applicable Models 40-3

40.1. Overview

With Web Streaming function, video from the USB camera on HMI can be streamed and viewed in web browsers, and third party devices.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

40.2. Configuration

Instead of setting an object in EasyBuilder Pro, Web Streaming is controlled by system registers. To enable Web Streaming, please set bit addresses or use macros to add the following system registers into the project file.

LB-12356: open(set on)/close(set off) web streaming server
LB-12357: web streaming server status. (on: opened / off: closed)

Set LB-12356 on to start Web Streaming, and set LB-12356 off to stop Web Streaming.
LB-12357 indicates the streaming status. When LB-12357 is on, Web Streaming is enabled and running. When LB-12357 is off, Web Streaming is stopped.
Web Streaming cannot start when the USB camera is not connected.

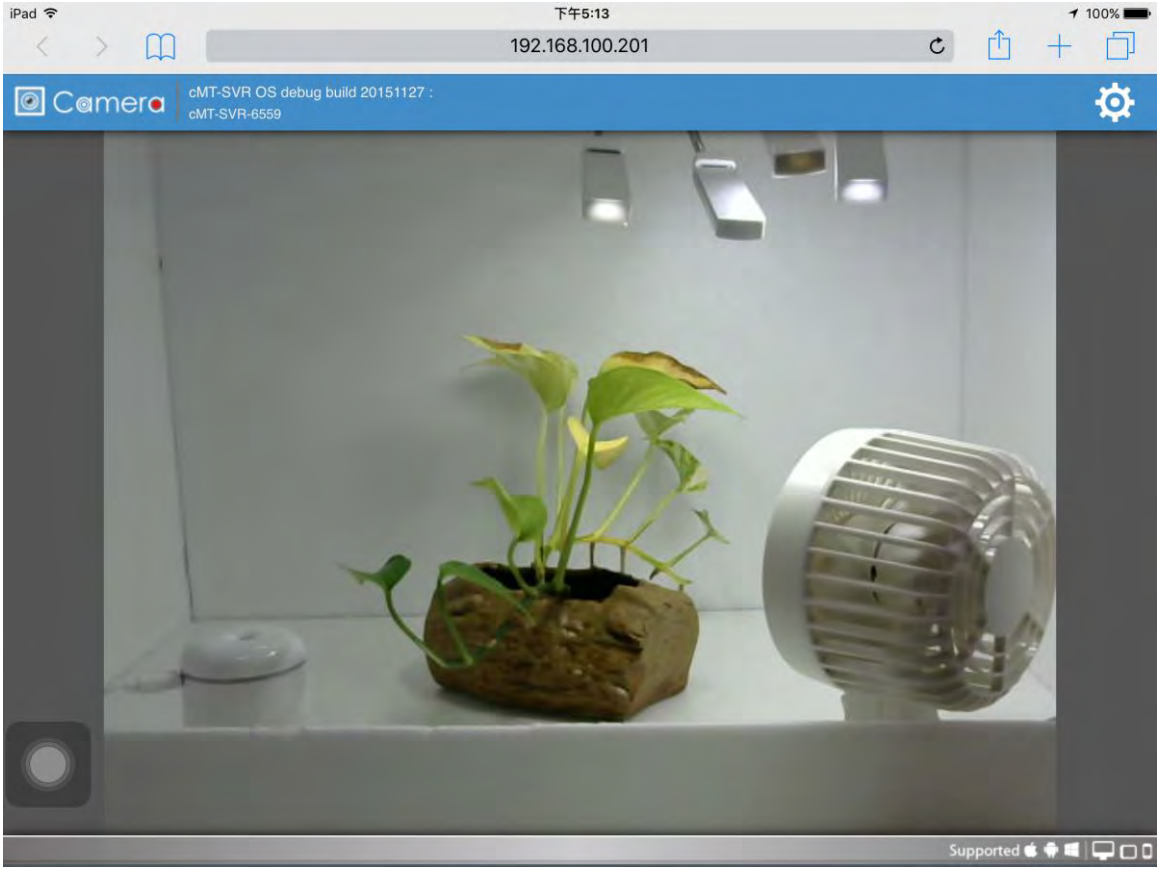
40.3. Operation

First, please make sure that the Web Streaming browser is enabled and running. To stream video on a browser, enter HMI's IP address with port 8080. For example, if the IP address is 192.168.1.100, please enter:

http://192.168.1.100:8080

Example 1

The following is a still image of video stream on iPad's Safari of a plant monitoring application.



Note

- When using EasyAccess 2.0, the HMI's IP address is the virtual IP address obtained in the software.

40.4. Applicable Models

Web streaming feature is supported on eMT/mTV/XE/cMT/cMT X Series.
Please update the OS versions to those listed in the table below. Models not listed here support web streaming without the need for updating OS.

Model	OS Version
eMT Series	20150525 or later
XE Series / eMT3070B	20150225 or later
mTV	20150508 or later
cMT-SVR	20151117 or later
cMT3090	20171024 or later
cMT3151	20171129 or later

41. Energy

This chapter explains how to setup Energy Demand Settings to monitor and record energy consumption and calculate future energy demands.

41.1. Energy Demand Setting 41-2
41.2. Energy Demand Display 41-6

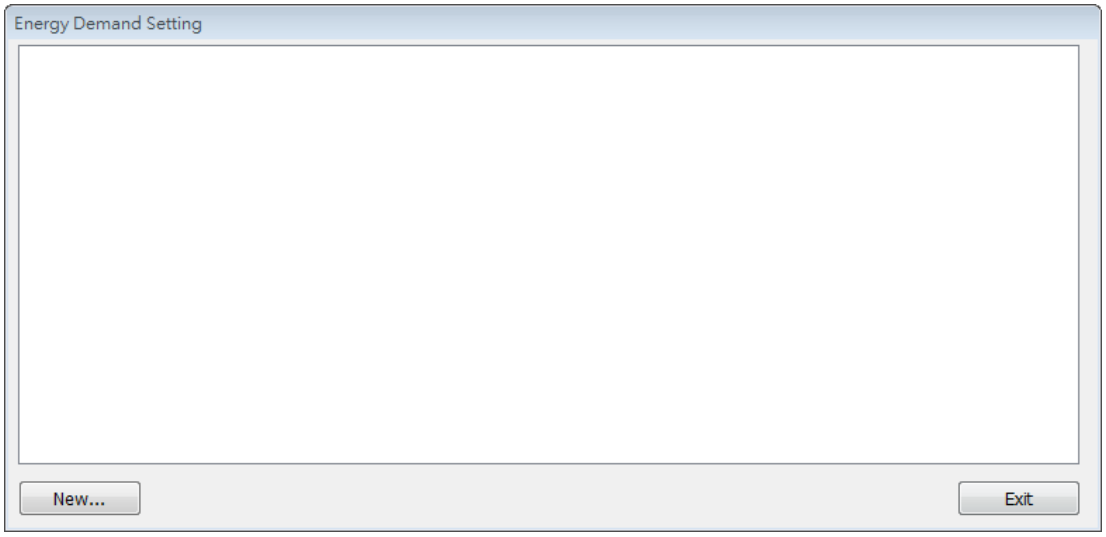
41.1. Energy Demand Setting

41.1.1. Overview

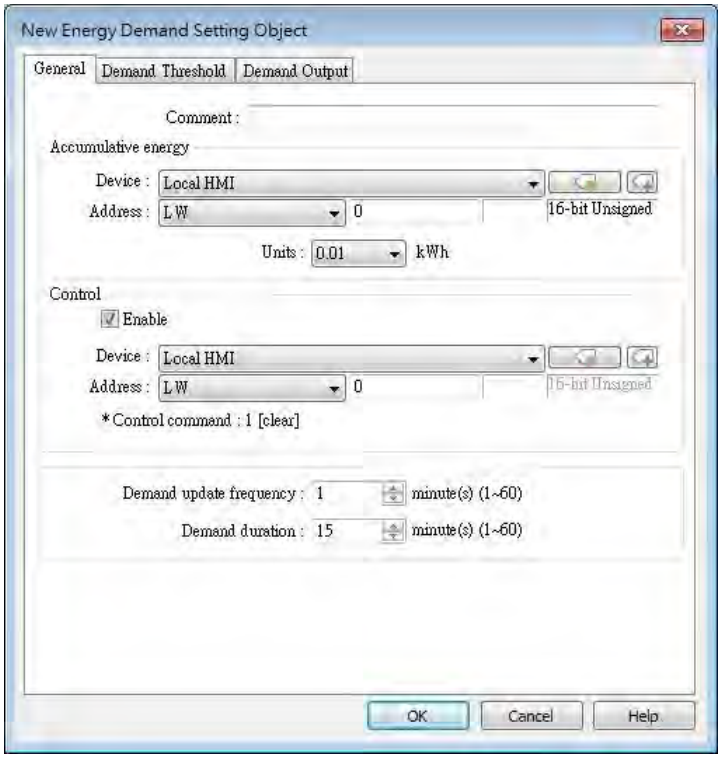
By monitoring the recorded energy consumption in a specified period, the Energy Demand Setting feature can calculate future energy demands, and help saving energy.

41.1.2. Configuration

Click [Data/History] and then click [Demand Setting] to open the settings dialog box. Configure General and Demand Threshold settings and click OK; an Energy Demand Setting object will be created.



General Tab

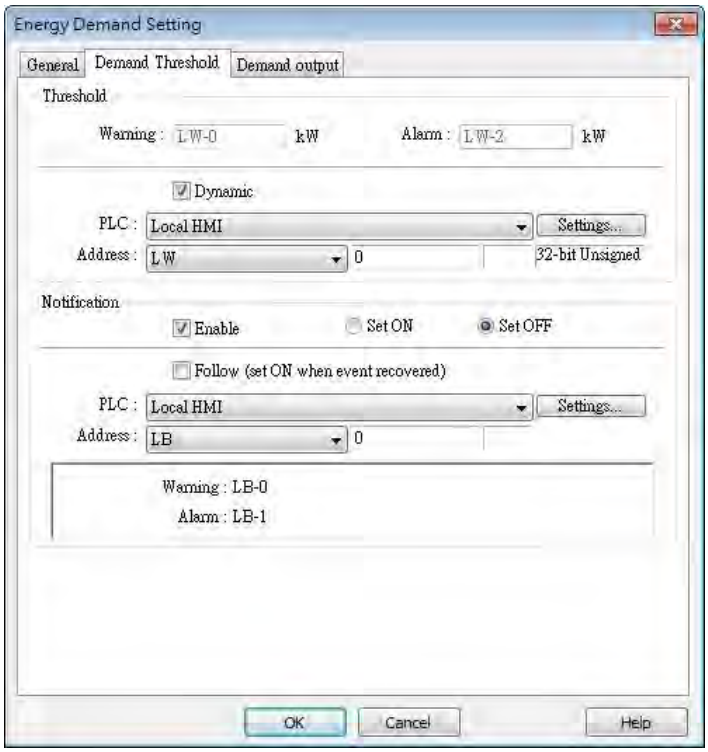


Setting	Description
Accumulative energy address	This address records energy consumption. The unit can be 0.1/0.01/0.001 kWh (kilowatt-hours).
Control	Giving control command 1 in the designated address can clear the energy consumption record.
Demand update frequency	The frequency to record energy consumption, the range is from 1 to 60 minutes.
Demand duration	The frequency to calculate energy demand. The range is from 1 to 60 minutes.

Note

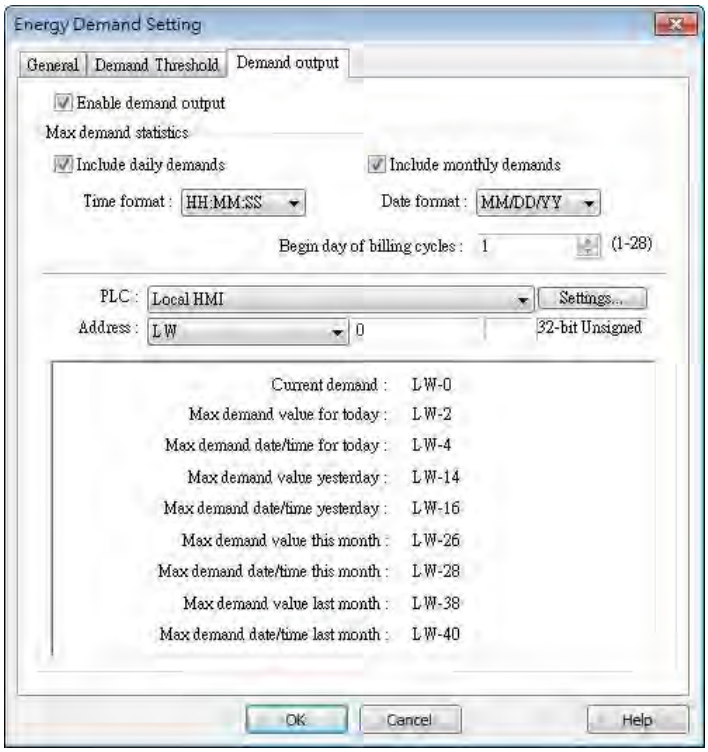
- Please note that Demand Duration (T) must be an integral multiple of Demand Update Frequency (t).

Demand Threshold



Setting	Description
Threshold	When the values in the specified addresses reach the values specified in Warning and Alarm fields, the warning and alarm will be triggered. The threshold limits can be dynamically changed at runtime.
Notification	When the estimated energy demand reaches the threshold limit, the status of the specified bit address will change accordingly.
Follow	If selected, when the estimated energy demand falls less than the threshold limit, the status of the notification bit address will return to its original state.

Demand Output



Setting	Description
Enable demand output	Opens [Max. demand statistics] settings.
Max demand statistics	The maximum energy demand of today/yesterday, and current month/last month, can be recorded in the corresponding addresses. The time/date format, and the beginning day of the billing cycle, can be specified.

41.2. Energy Demand Display

41.2.1. Overview

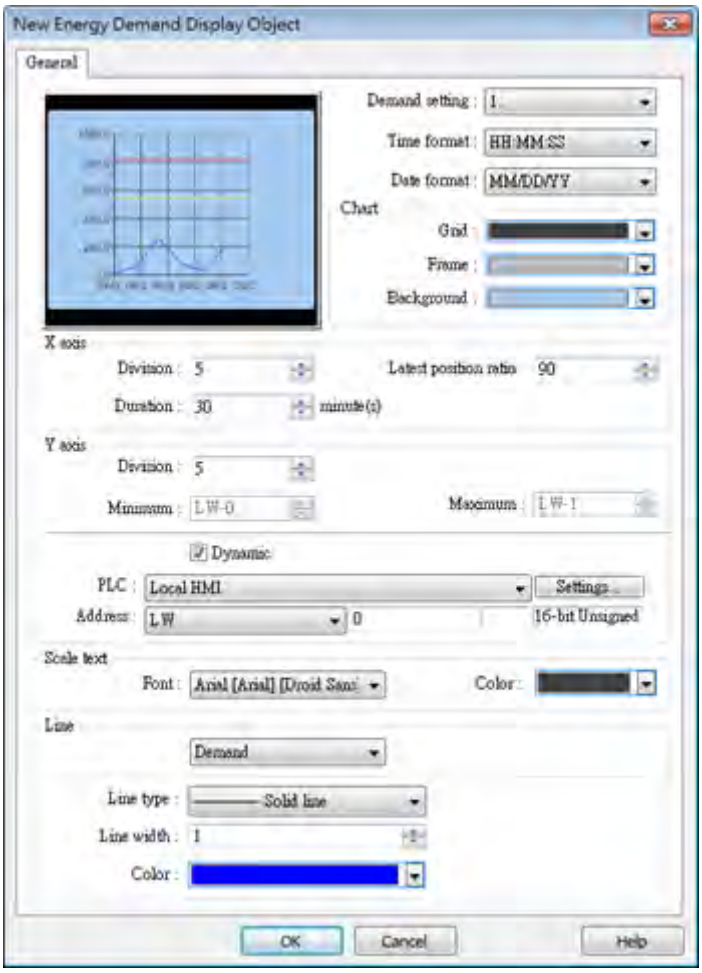
Energy Demand Display object graphs the result from Energy Demand Setting object at runtime.

The font, grid and watch line style can be specified, and the threshold limits can be shown in the graph.

41.2.2. Configuration

Click [Data/History] and then click [Demand Display] to open the settings dialog box. Configure the attributes and click OK; an Energy Demand Display object will be created.

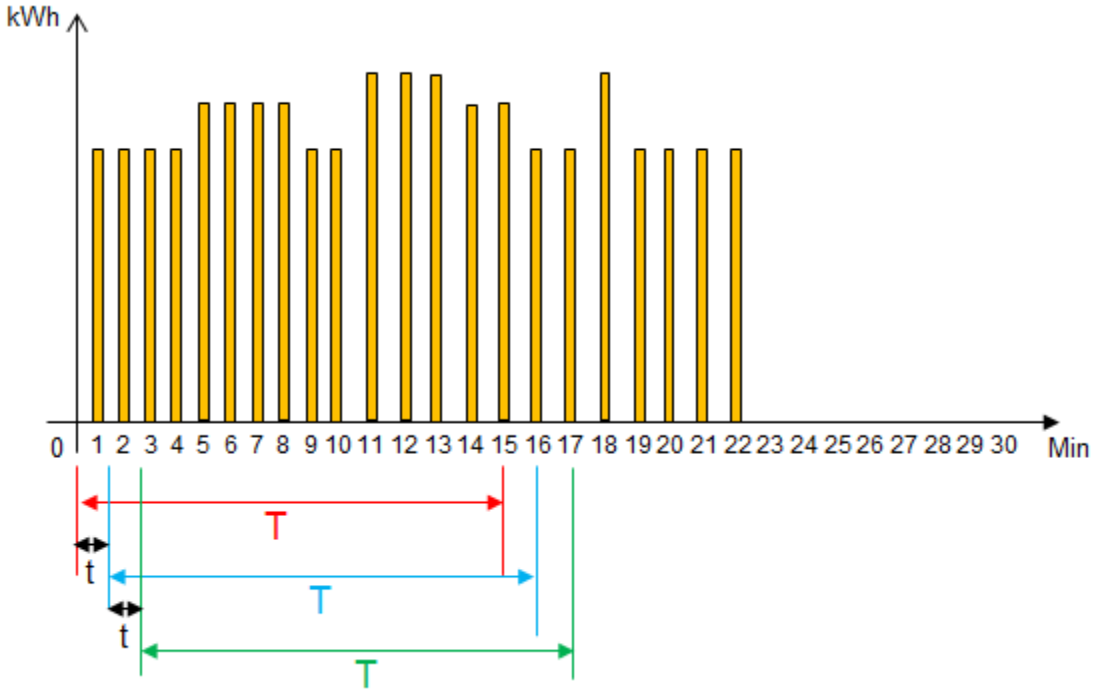
General Tab



Setting	Description
Object index	Select an existing Energy Demand Setting object as the data source of Energy Demand Display object.
Time/Date format	Set Time and Date format.
Chart	Set the colors of Grid, Frame, and Background.
X axis	Set the number of divisions and time duration on the X axis. Latest position ratio: The ratio represents a position along the X axis, where 50% represents the middle and 100% represents the right end. This field sets the position to mark the latest data, and the acceptable range is: 50%~100%.
Y axis	Set the number of divisions, maximum and minimum scale values on the Y axis. The limits can be changed dynamically at runtime.
Scale text	Set the font and color of the scale text.
Line	Three watch lines marking Demand, Alarm, and Warning can be shown. The type, width, and color of the lines can be customized.

Example 1

The following example illustrates the relationship between Demand Duration (T) and Demand Update Frequency (t) mentioned in Energy Demand Setting guide above.



1. As shown in the above figure, when t=1, the frequency to record energy consumption will be once per minute. When T=15, each 15 minutes the sum of the energy consumption measured every minute (t=1) will be calculated.
2. From the 1st to the 15th minute (red zone), the energy consumption measured each minute will be added up to get the total sum. The total sum times 4 (15 minutes is a quarter of an hour) to obtain a value (kWh) indicating the estimated energy demand.
3. From the 2nd to the 16th minute (blue zone), the energy consumption measured each minute will be added up to get the total sum. The total sum times 4 (15 minutes is a quarter of an hour) to obtain a value (kWh) indicating the estimated energy demand.
4. From the 3rd to the 17th minute (green zone), the energy consumption measured each minute will be added up to get the total sum. The total sum times 4 (15 minutes is a quarter of an hour) to obtain a value (kWh) indicating the estimated energy demand.
5. The Energy Demand Display object will graph the estimated energy demand.
6. If t=3 and T=15, the estimated energy demand will be: Sum of the latest 5 records (15/3=5) times 4 (15 minutes is a quarter of an hour).
7. If t=5 and T=30, the estimated energy demand will be: Sum of the latest 6 records (30/5=6) times 2 (30 minutes is half an hour).

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

42. IIoT

This chapter explains how to use IIoT protocols.

42.1. MQTT 42-2
42.2. OPC UA Server 42-28

42.1. MQTT

42.1.1. Overview

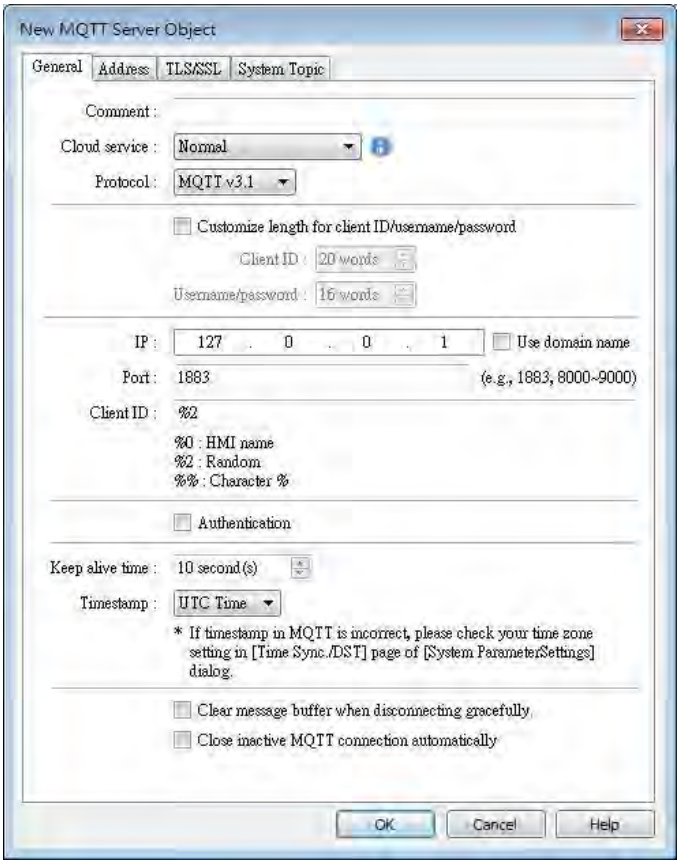
MQTT object can publish messages to an MQTT server, or subscribe to topics to receive messages from an MQTT server. HMI can serve as an MQTT server as well. When HMI serves as an MQTT server, it does not send message to another MQTT server.

42.1.2. Configuration

Click [Object] » [IIoT] » [MQTT] in the menu to open the settings dialog box.

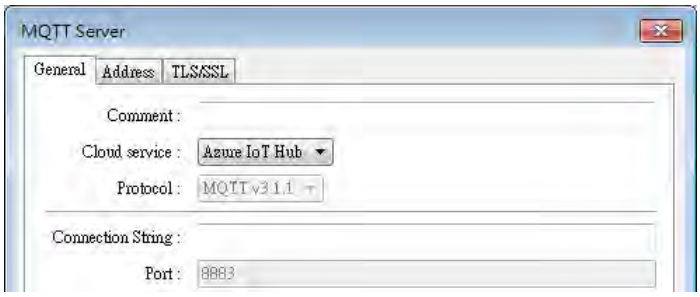
42.1.2.1. Server Settings

General Tab



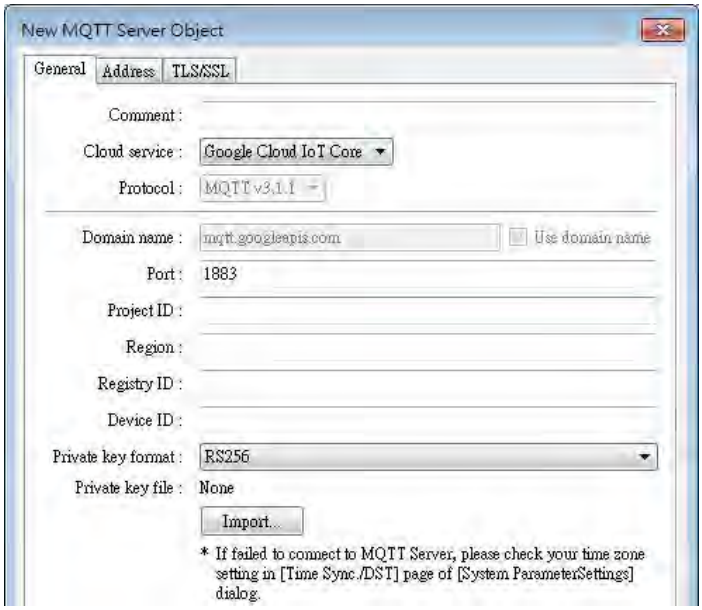
Setting	Description
Cloud service	**Normal** Use general MQTT publish-subscribe service. **AWS IoT** Use AWS IoT as a Broker, and use Thing Shadows service. For more information, please find “AWS IoT User Manual”. **Sparkplug B** Sparkplug B is a specification designed based on the characteristic features of IoT applications. It helps define topics and messages that are not specified by standard MQTT, and allows non-MQTT terminal devices to transfer data with MQTT Server through Edge of Network, which can be HMIs in this architecture. Please see “Sparkplug B Quick Start Guide” for more information. **Azure IoT Hub** Use Microsoft Azure IoT Hub as a Broker. Using this

service can simplify setting step to entering a Connection String. The Connection String can be found in Microsoft Azure > IoT devices.



Google Cloud IoT Core

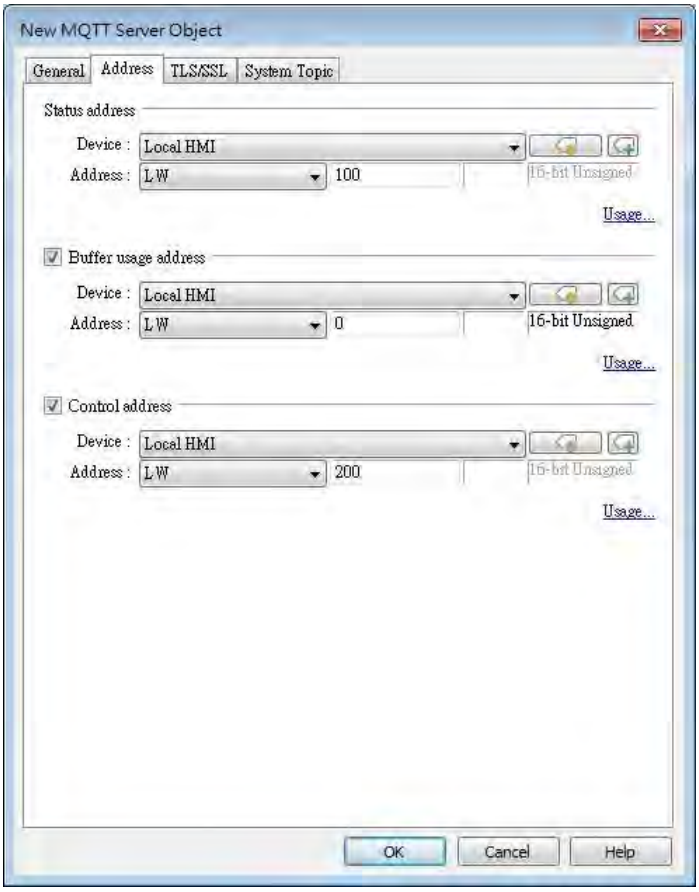
Use Google Cloud IoT Core as a Broker. Using this service requires filling in connection parameters and authentication credentials.



Protocol	Supports MQTT v3.1 and v3.1.1.
Customize length for Client ID/ username/password	Client ID: The upper limit is 128 words. Username/Password: The upper limit is 256 words.
IP	Enter the MQTT Server IP address for receiving the message. If 127.0.0.1 is entered, HMI will run a MQTT server locally.
Use domain name	A domain name can be used as MQTT server's IP address.
Port	Enter the MQTT Server port number for receiving the message.
Client ID	Enter the Client ID. Variables can be used for Client ID, for example, entering %0 will make the HMI Name to be the

	Client ID.
Authentication	If selected, connecting MQTT Server will require [Username] and [Password].
Username	Enter the username for connecting MQTT Server.
Password	Enter the password for connecting MQTT Server.
Keep alive time	When MQTT Server does not receive the message from HMI passing the specified time, the HMI will be identified as disconnected. Note: When running simulation, the message may be delayed, but the delay will not exceed the [Keep alive time]. The message from the HMI will be sent immediately.
Timestamp	**Local time** Use local HMI time for timestamp. **UTC time** Use UTC+0 (coordinated universal time) for timestamp. When the timestamp is shown incorrectly, please go to [System Parameters] » [Time Sync. / DST] tab to set the time zone.
Clear message buffer when disconnecting gracefully	This option is selected by default. With it selected, when disconnecting gracefully (by entering 2 for the command in MQTT's control addresses), message buffer will be cleared. Messages in the buffer will be retained when this option is not selected.
Close inactive MQTT connection automatically	In this mode, the connection will be automatically terminated if there's no data update for a specified period of time. The connection will resume once any data update occurs. The user can choose to publish initial values / topic list only at the first connection. In this mode, the start and stop commands are disabled.

Address Tab



Setting	Description
Status address	LW-n: Displays the connection status to MQTT Server. Value / Description 0 / Not attempting to connect to MQTT Server. 1 / Disconnected and can't connect to MQTT Server. 2 / Connection succeeded. LW-n+1: Error indicator. Value / Description 0 / No error 1 / Unknown error 2 / Failed to connect 3 / Access denied 4 / Not allowed port number for built-in MQTT server

5	Unresolvable domain name
6	Buffer overflowed
32	Incorrect client ID
48	Failed to verify certificate
256	Still connecting

Buffer usage address

Messages that have not been sent are stored in the buffer. The maximum buffer capacity is 10000 messages. The buffer capacity is measured in percentage (%), rounded up.
LW-n: Shows buffer usage.

Control address

LW-n: Controls the operation of MQTT Server.

Value	Description
0	Ready
1	Start
2	Stop
3	Update

LW-n+1: Sets the IP address of MQTT Server.
LW-n+5: Sets the port number of MQTT Server.
LW-n+6: Sets the Client ID for connecting MQTT Server.
LW-n+26: Enables / Disables authentication.

Value	Description
0	Disable
1	Enable

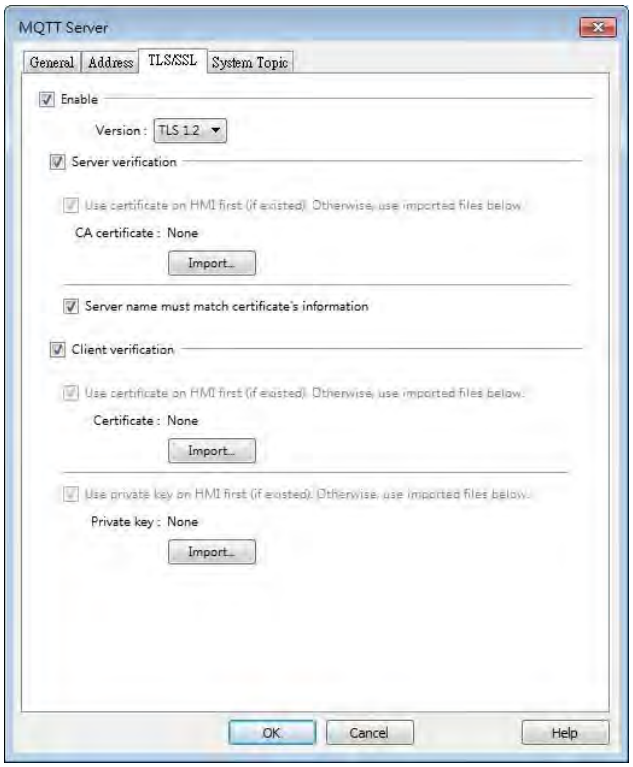
LW-n+27: Sets the username for connecting MQTT Server.
LW-n+43: Sets the password for connecting MQTT Server.
LW-n+59: Sets the domain name for connecting MQTT Server.

When Azure IoT Hub is used, the control addresses are as below:
LW-n: Controls the operation of MQTT Server.

Value	Description
0	Ready
1	Start
2	Stop
3	Update

	LW-n+1: Sets the Connection String (128 words).

TLS/SSL Tab

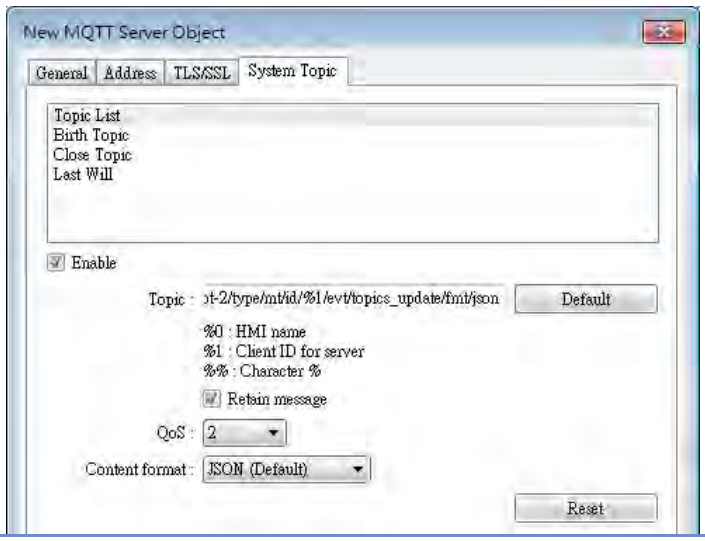


Setting	Description
Enable	Enable TLS/SSL authentication. TLS version can be selected from: TLS 1.0, TLS 1.1, and TLS 1.2. To use TLS 1.1 and TLS 1.2, HMI OS version must be 20180323 or later.
Server verification	**Enable** Verify whether the server certificate is signed by CA (Certificate Authority). Server certificate is sent from server during connection. **Server name must match certificate's information** Verify whether the server's domain name or IP matches the records in the server certificate. Domain name and IP records are stored in Subject Alternative Name of the certificate.
Client verification	Private key and client certificate is required for server to authenticate the client.

System Topic

Several system topics can be enabled for HMI to publish. When a system topic is enabled for an

HMI, the subscribers of that topic can view the list of available topics and connection status of that HMI.



Setting	Description
Topic List	List of topic sent from HMI to the server upon connection.
Birth Topic	The message sent from the HMI after it is connected to the server.
Close Topic	The last message sent from the HMI before it disconnects knowingly from the server.
Last Will	The message received by the subscriber to Last Will when connection between HMI and server is lost ungracefully. HMI updates its Last Will message when it connects to the server.
Topic	The actual topic name of the system topic.
Retain Message	When this checkbox is selected, the MQTT server will save the latest message.
QoS	MQTT provides three levels of reliability, which are known as quality of service (QoS). The reliability of the message determines the persistence of the message. QoS 0: At most once, messages are not persistent. QoS 1: At least once. QoS 2: Exactly once.
Content Format	JSON (Default): Use default content. The following are the defaults of each system topic. Actual context-dependent values are shown in red:

Topic list:

```
{
    "d" : {
        "topics" : [
            {
                "compression" : "Compression Type",
                "nickname" : "Topic Name",
                "topic" : "Topic"
            },
            {
                "compression" : "Compression Type",
                "nickname" : "Topic Name",
                "topic" : "Topic"
            }
        ]
    },
    "ts" : " Current Time "
}
```

Contents in the topics vary according to the actual topic settings. The above is an example for the case of two topics.

Birth Topic:

```
{
    "d":{
        "connected":true
    },
    "ts":"Current Timestamp"
}
```

Close Topic:

```
{
    "d":{
        "connected":false
    },
    "ts":" Current Timestamp"
```

```
}

Last Will:
{
    "d":{
        "connected":false
    }
}
```

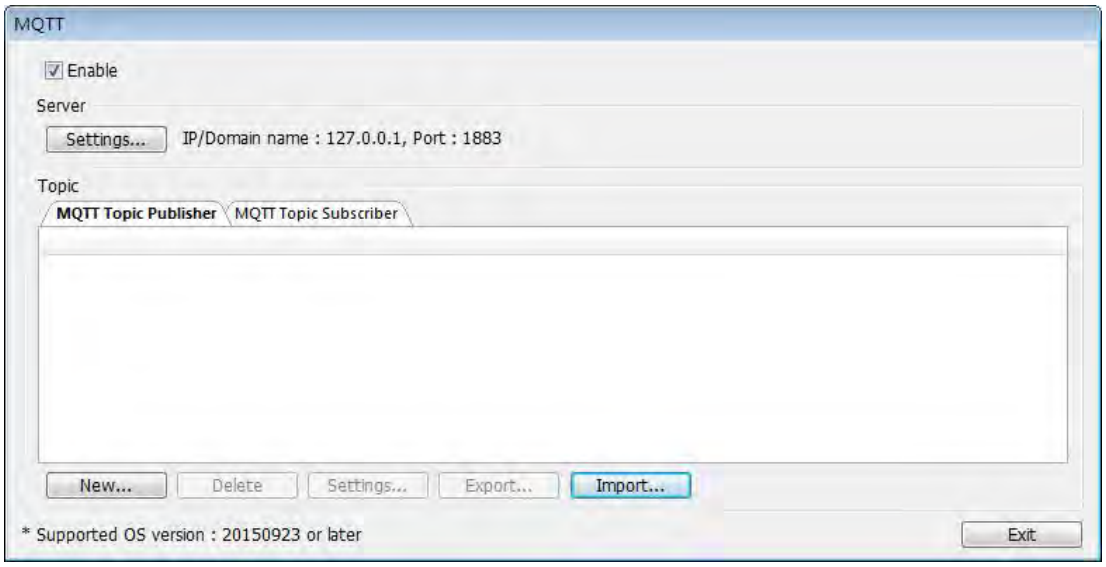
JSON (Advanced): Use user-defined content.

Note

- Please note that System Topics tab is not supported when using Sparkplug B, Azure IoT Hub, and Google Cloud IoT Core cloud services.

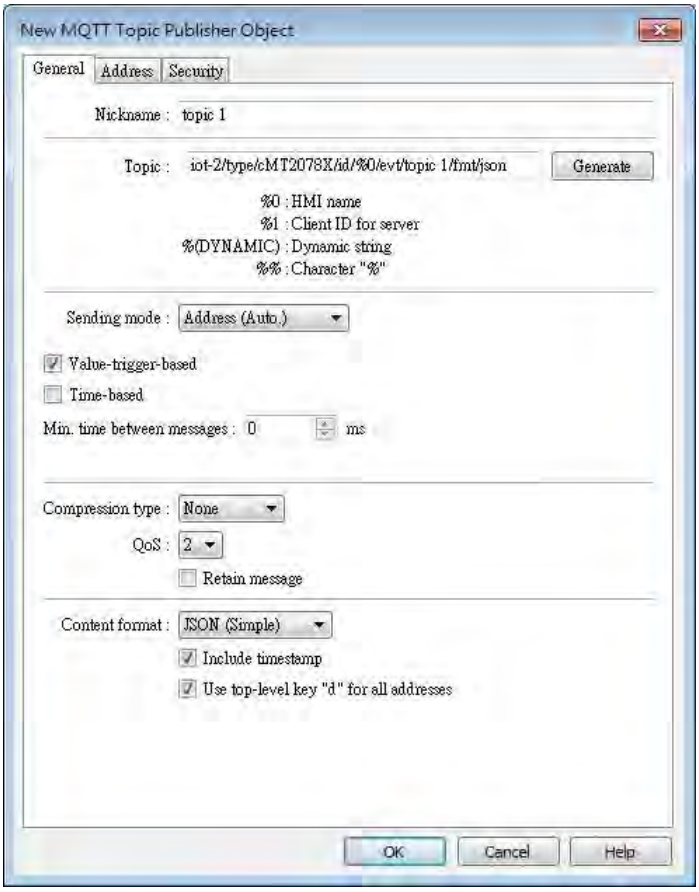
Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

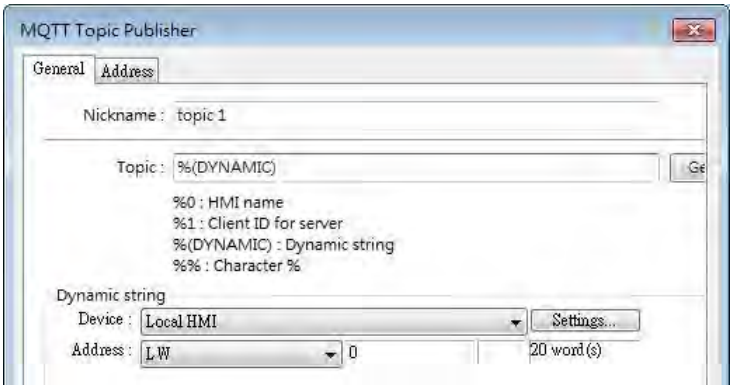
42.1.2.2. MQTT Topic Publisher



Click [New] to open General and Address settings, or click [Import] / [Export] to import or export an existing *.csv file. The maximum allowable number of topics is 255.

General Tab



Setting	Description
Nickname	Enter the nickname of the MQTT Topic for easier reference.
Topic	Specify the format of the message topic sent to MQTT Server. Variables can be used for Topic. Entering %(DYNAMIC) in the Topic field opens Dyanmic String group box for designating a word address. %(DYNAMIC) can include multiple topic levels. For example: myhome/groundfloor.  When Azure IoT Hub is used, users can only specify the last topic level.

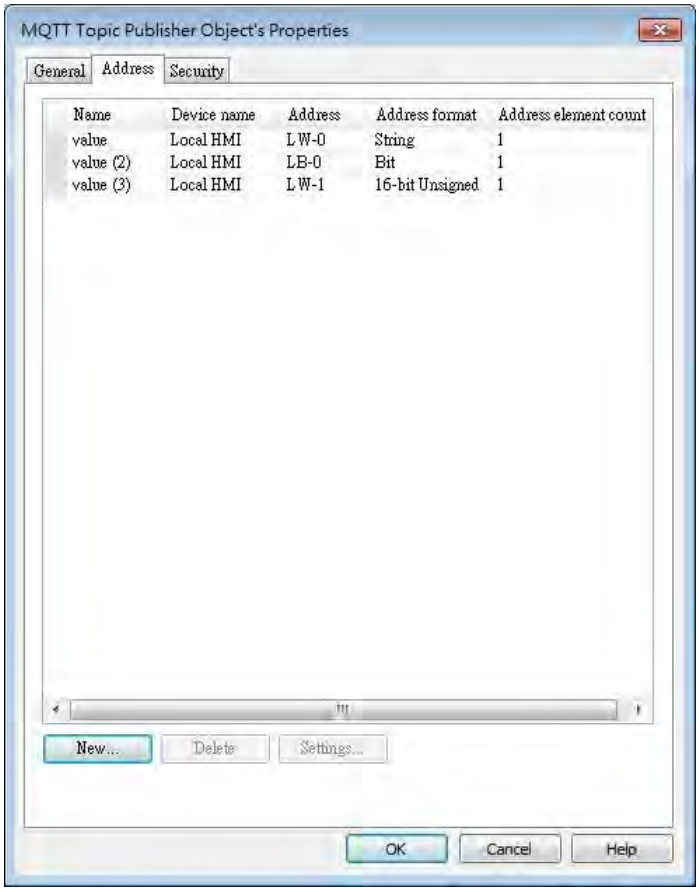
Sending mode	**Address (Auto.)** Value-trigger-based: Sends MQTT message when any value changes. Time-based: Sends MQTT message in a time-based manner. Min. time between messages: Sends MQTT message in a specified time interval. When the value changes within a period of time shorter than the specified time interval, the message will be stored in the buffer and then sent after the set wait time. This prevents the publisher from sending messages too frequently. **Address (Bit trigger)** Sends MQTT message when a designated bit is triggered. **Event (Alarm) Log** The topic source can be an Event Log. MQTT message can be sent when a single event or any event in a specific category occurs.
Compression type	The message will be compressed before being sent, and decompression is needed before reading the message. Messages in MQTT can be compressed / decompressed in zlib, gzip, or with DEFLATE algorithm.
Retain message	If selected, the MQTT server will save the latest message.
Include timestamp	This option is available only when the format used is [JSON (simple)]. Selecting this option can include timestamp in the message.
Use top-level key “d” for all addresses	This option is available only when the format used is [JSON (simple)]. When selected, the message format is as below: `{ "d": { "addressName1": ..., "addressName2": ... }, "ts": ... }` When not selected, the message format is as below:

	{
	"addressName1": ...,
	"addressName2": ...,
	"ts": ...
	}
	As shown in the above figure, when this option is not selected, ts and address names become keys of the same level. Therefore, please avoid using ts as an address name in this case.

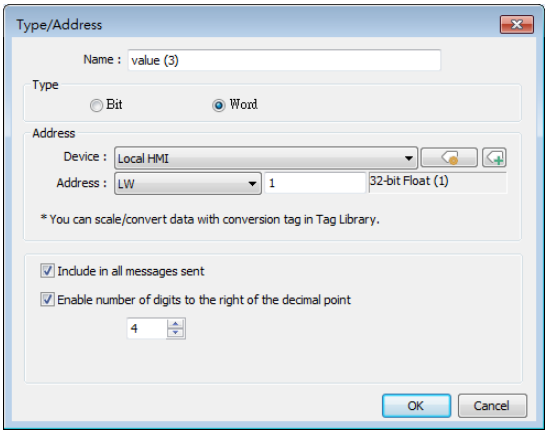
QoS	MQTT provides three levels of reliability, which are known as quality of service (QoS). The reliability of the message determines the persistence of the message. 0: At most once, messages are not persistent. 1: At least once. 2: Exactly once.
Content Format	The supported formats are: Raw Data: Data in bytes. JSON (Simple): JSON format with all data put in JSON member “d”. JSON (Advanced): JSON format with flexible JSON structure.

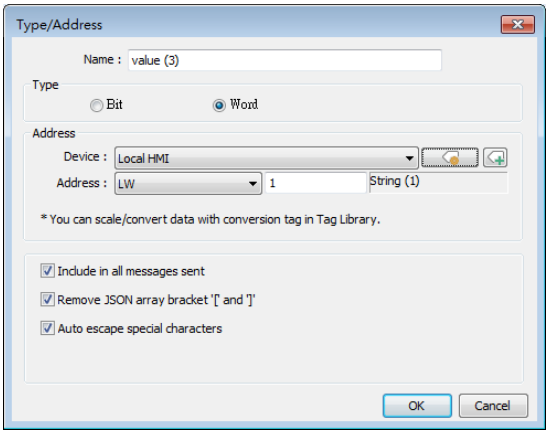
Address Tab

The following explains the address settings for [Raw Data] and [JSON (Simple)] content formats.



Setting	Description
New	Add the source of the topic. The length of each address can be specified respectively.
Delete	Delete the address.
Setting	Change the name and address.





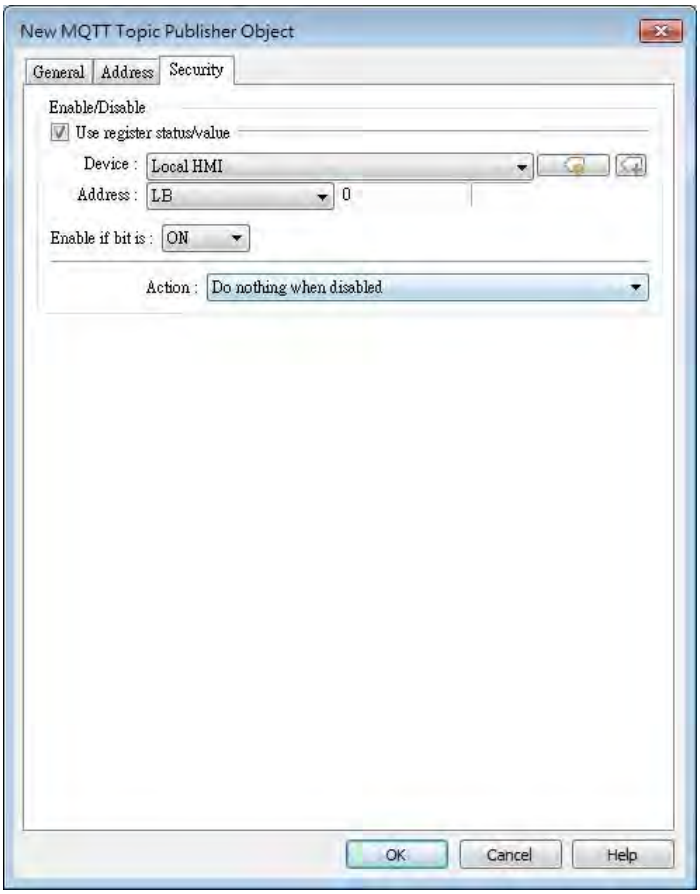
Setting	Description
Remove JSON array bracket "[" and "]"	For JSON formatted messages, selecting this option can remove bracket "[" and "]".

Enable number of digits to the right of the decimal point	When data type is Float, the number of digits after the decimal point can be specified.
Auto escape special characters (cMT / cMT X Series)	This option is available when the data type is String. JSON formatted messages may contain special characters (e.g. " and \) which can lead to JSON parsing errors. With this option selected, special characters in a string can be escaped (e.g. change from " to \" and change from \ to \\) for successful message parsing.

Note

- Maximum tag length: 255 words.

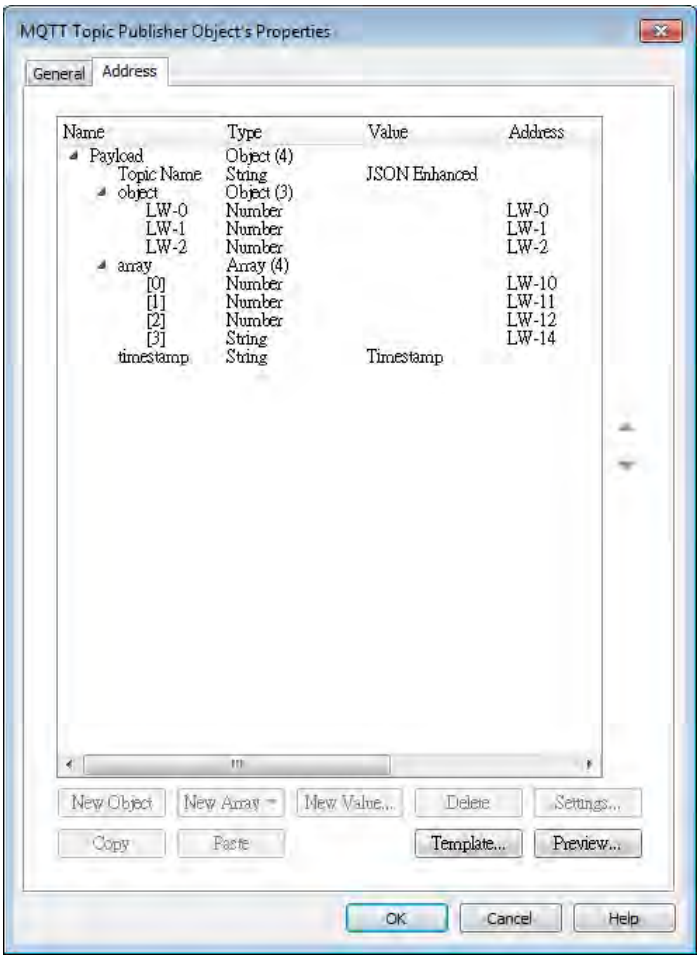
Security Tab



Messages will be published only when the state of the designated address meets the set condition. As shown above, the message will be published when LB-0 is ON.

Address Tab [JSON (Advanced)]

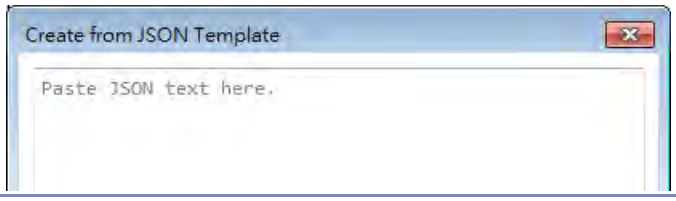
The following explains address settings for [JSON (Advanced)] content format. This is a nested format that allows using objects or arrays, and customizing timestamp and data name. Using this format provides a more flexible way of using MQTT.



When configure the settings as shown in the above screenshot, the received MQTT message by the subscriber is as below.

```
{
  "Topic Name" : "JSON Enhanced",
  "Object" : {
    "LW-0" : [ 1 ],
    "LW-1" : [ 2 ],
    "LW-2" : [ 3 ]
  },
  "Array" : [ [ 4 ], [ 5 ], [ 6 ], [ "AABBCCDD" ] ],
  "timestamp" : "2019-02-19T06:52:13.846038"
}
```

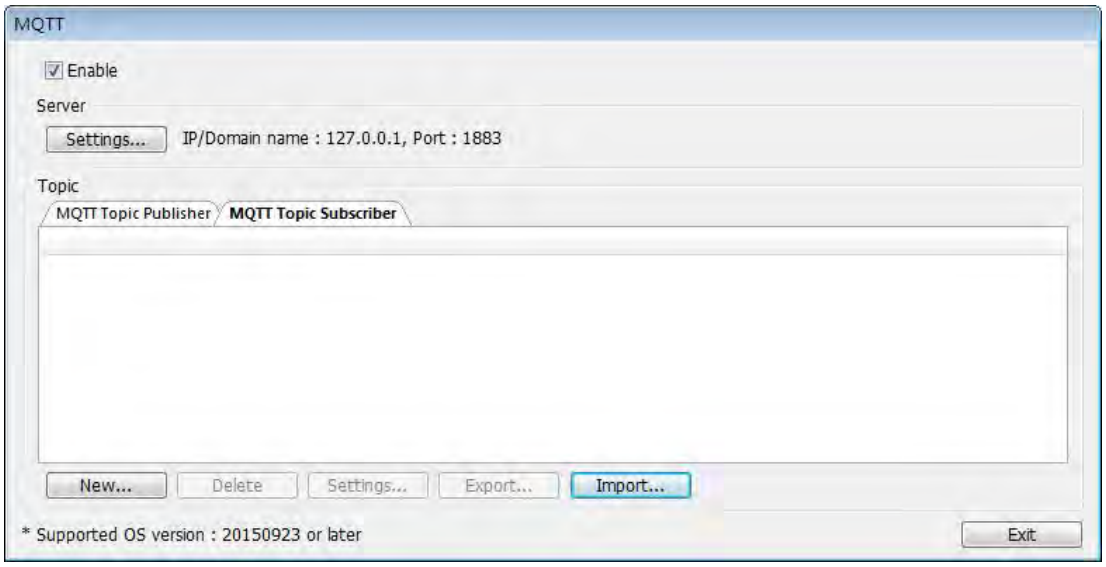
Setting	Description
New Object	Add a new object. The name, type and value of each item under the object can be configured. Items under the object are enclosed in curly brackets { }.

New Array	Add a new array. An array may contain multiple items but the name of the item is automatically generated and is unchangeable. Items under the array are enclosed in square brackets [].
New Value	Add a new number, string, or timestamp. When the new value is a number or a string, fixed value can be selected, or an address can be designated as the data source.
Delete	Delete the selected item.
Settings	Configure the selected item. When the selected item is an object or an array, the user may only change its name. When the selected item is contained in an object or an array, its parameters can be configured.
Copy	Copy the selected item.
Paste	Paste the copied item to the selected row.
Template	By pasting JSON string into the window, the system will automatically adjust the data structure setting according to JSON structure, saving time for users. 
Preview	Preview the JSON data in a reader-friendly format.

Note

- Maximum number of nodes for a Topic is 512 (payload included). Maximum tag length is 255 words.

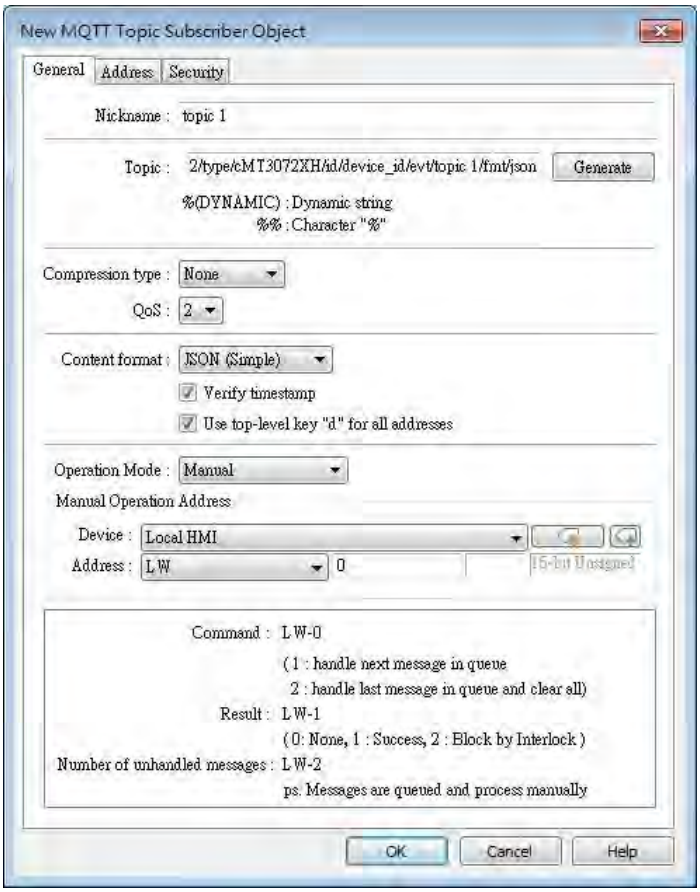
42.1.2.3. MQTT Topic Subscriber

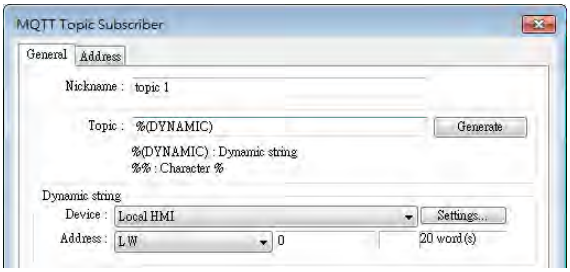


Click [New] to open General and Address settings, or click [Import] / [Export] to import or export an existing *.csv file. The maximum allowable number of topics is 255.

General Tab

The following explains the address settings for [Raw Data] and [JSON (Simple)] content formats.



Setting	Description
Nickname	Enter the nickname of the MQTT Topic for easier reference.
Topic	Subscribe to a topic in MQTT Server. The topic name can be dynamic. Entering %(DYNAMIC) in the Topic field opens Dynamic String group box for designating a word address. %(DYNAMIC) can include multiple topic levels. For example: myhome/groundfloor.  When Azure IoT Hub is used, users can only specify the last topic level, and the topic level should be the same as in MQTT Topic Publisher.
Compression type	Configure with the same setting as MQTT Topic Publisher.
Verify timestamp	When timestamp is included in the message, selecting this option will verify whether the timestamp is increasing, and update will occur when the timestamp does increase; otherwise, the message will be treated as expired message and update will not occur.

Use top-level key "d" for all addresses	When selected, the message format is as below:
	`{`
	`  "d": {`
	`  "addressName1": ...,`
	`  "addressName2": ...`
	`  },`
	`  "ts": ...`
	`}`
	When not selected, the message format is as below:

```
{
    "addressName1": ...,
    "addressName2": ...,
    "ts": ...
}
```

Please use appropriate setting according to the data source.

QoS

MQTT provides three levels of reliability, which are known as qualities of service (QoS). The reliability of the message determines the persistence of the message.
0: At most once, messages are not persistent.
1: At least once.
2: Exactly once.

Content Format

Raw Data: Unformatted raw data.
JSON (Simple): Single layer JSON format.
JSON (Advanced): JSON format with user-defined JSON structure.

Operation Mode

Operation mode for subscribing topics can be selected.
Process immediately: Write the value to the designated address immediately after receiving subscribed data.
Manual: Place the subscribed data in a queue before processing the data manually. The queue can hold 100 records.

Control address (Manual)

LW-n: Command

Value	Description
1	Write the oldest data in the buffer to the designated address. If there are 10 records in the buffer, the user can enter command 1 for ten times to write the data to the address sequentially.
2	Write the latest data in the buffer to the designated address, and then clear all data in the buffer.

LW-n+1: Execution Result

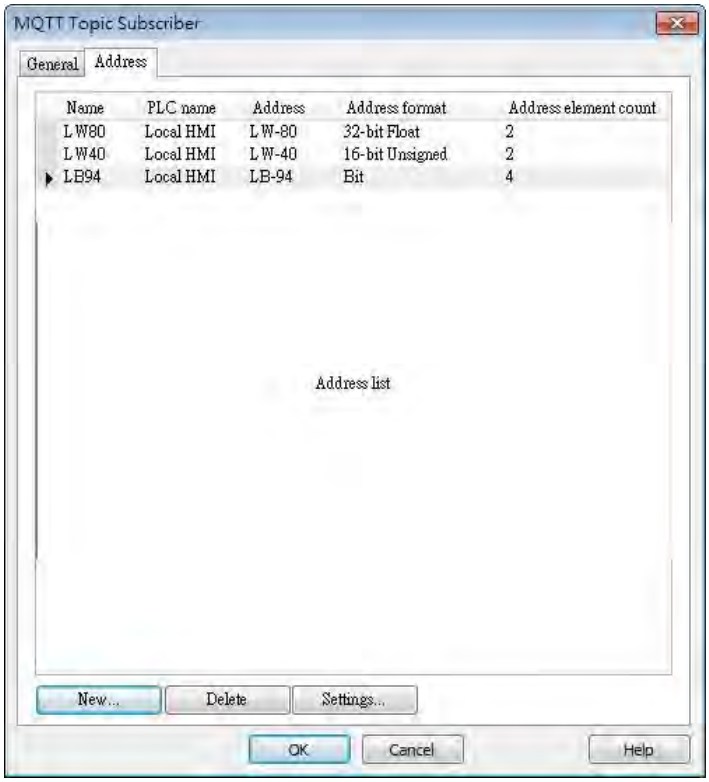
Value	Description
0	The buffer is currently empty.

1	The command is executed successfully.
2	The topic subscription is blocked so command execution failed. (See Security tab in this chapter.)

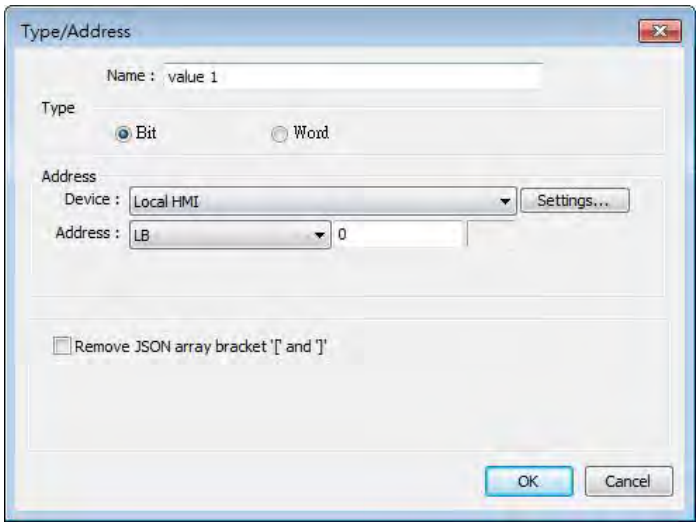
LW-n+2: Number of unhandled messages

Displays the number of messages in the buffer.

Address Tab

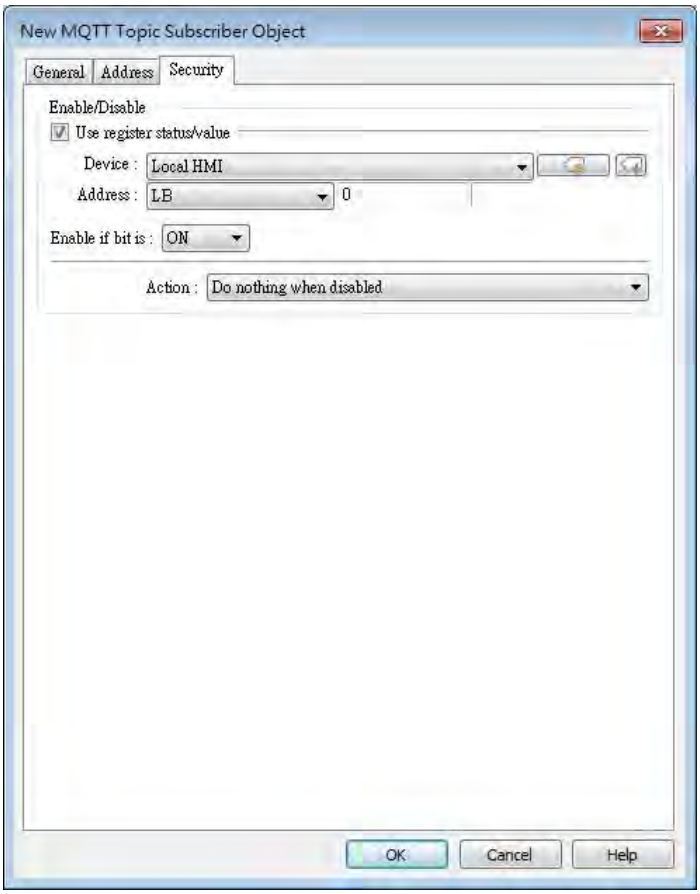


Setting	Description
New	Add the destination address of the subscribed topic. The length of each address can be specified respectively.
Delete	Delete the address.
Setting	Change the name and address.



Setting	Description
Remove JSON array bracket "[" and "]"	For JSON formatted messages, selecting this option can remove bracket "[" and "]".

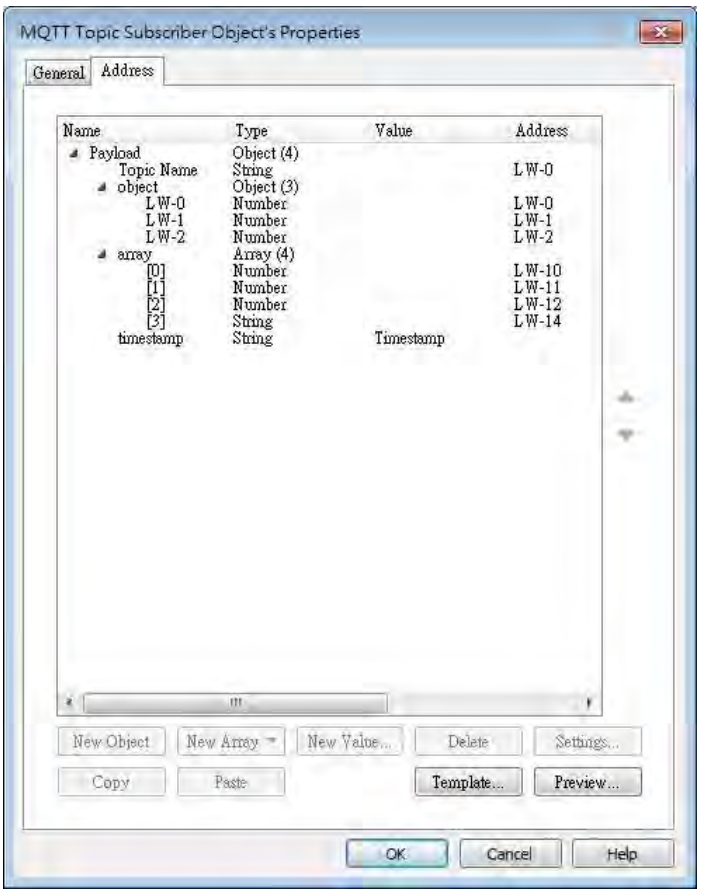
Security Tab



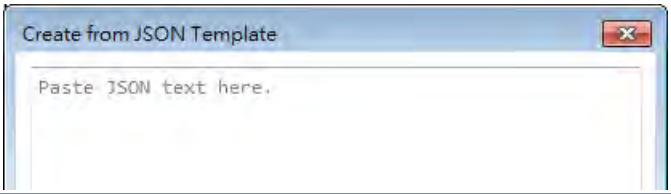
Messages will be subscribed only when the state of the designated address meets the set condition. As shown above, the message will be subscribed when LB-0 is ON.

Address Tab [JSON (Advanced)]

The following explains address settings for [JSON (Advanced)] content format. This is a nested format that allows using objects or arrays, and customizing timestamp and data name. Using this format provides a more flexible way of using MQTT.



Setting	Description
New Object	Add a new object. The name, type and value of each item under the object can be configured. Items under the object are enclosed in curly brackets { }.
New Array	Add a new array. An array may contain multiple items but the name of the item is automatically generated and is unchangeable. Items under the array are enclosed in square brackets [].
New Value	Add a new number, string, or timestamp. When the new value is a number or a string, fixed value can be selected, or an address can be designated as the data source.
Delete	Delete the selected item.
Settings	Configure the selected item. When the selected item

	is an object or an array, the user may only change its name. When the selected item is contained in an object or an array, its parameters can be configured.
Copy	Copy the selected item.
Paste	Paste the copied item to the selected row.
Template	By pasting JSON string into the window, the system will automatically adjust the content according to JSON structure, saving time for users. 

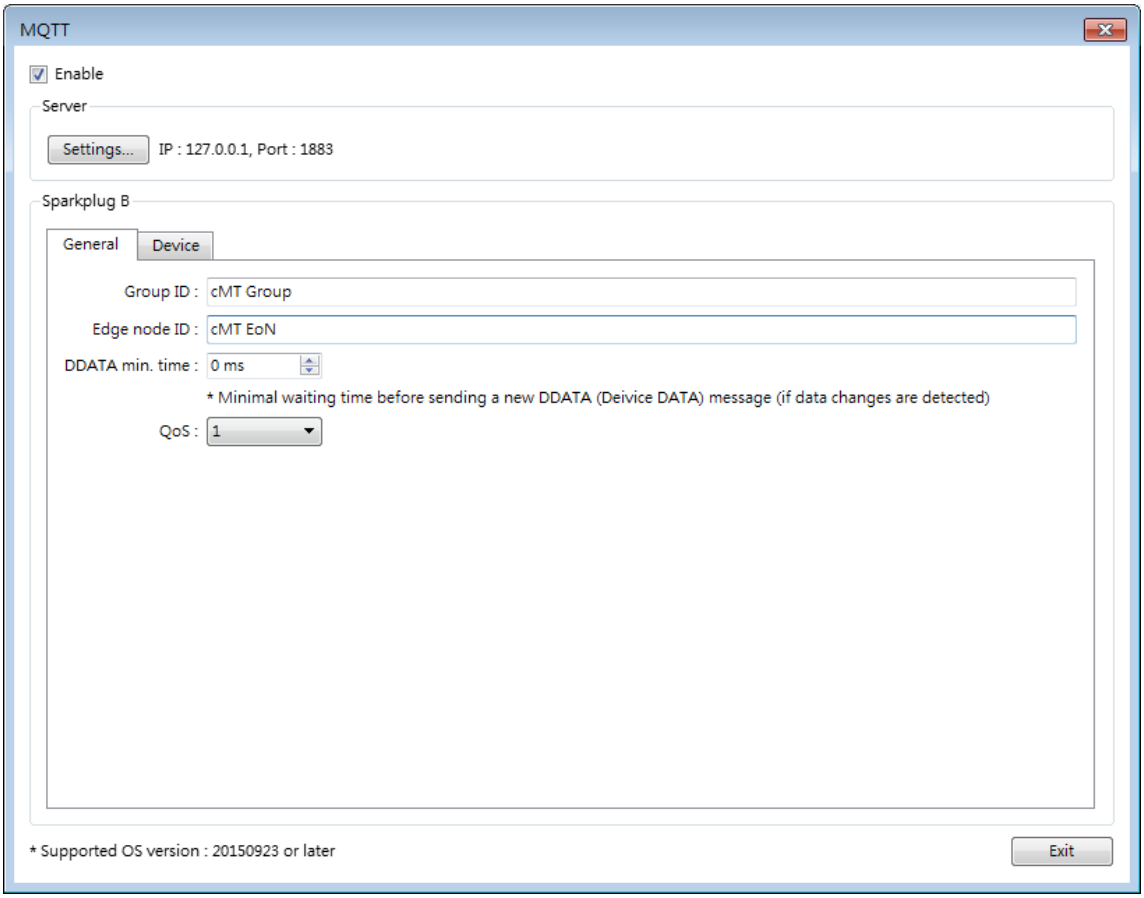
Note

- Amazon Web Service (AWS) IoT Core supports standard MQTT protocol. However, please note the following restrictions:
  1. The maximum number of layers in a topic is 8 (iot-2/type equals to 2 layers).
  2. Authentication in General tab is not supported, please use TLS/SSL.
  3. Supports only Qos 0 and Qos 1.
  4. Retaining the latest message in MQTT server is not supported.

42.1.2.4. Sparkplug B

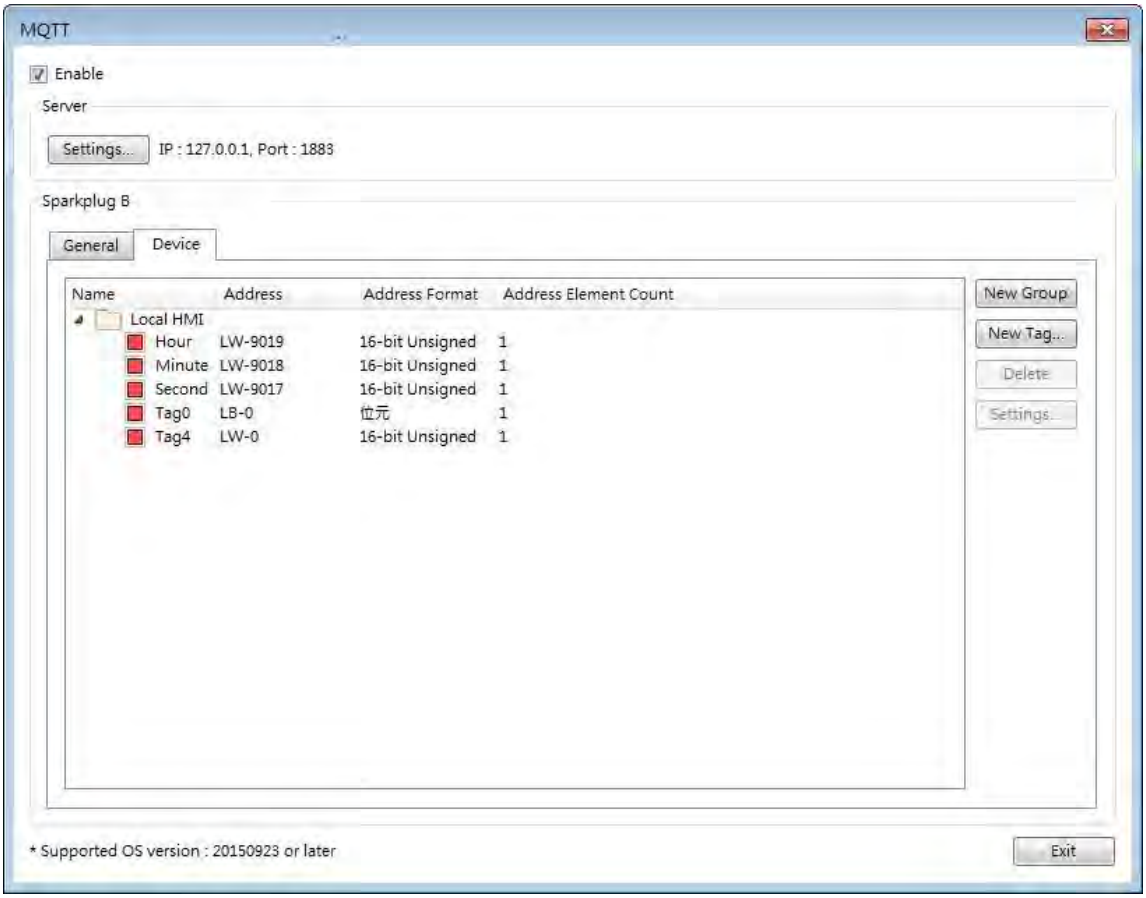
General settings and Device settings for cloud service Sparkplug B are as shown below.

General Tab



Setting	Description
Group ID	The group ID that identifies the group in which the Edge of Network Nodes belong to.
Edge ID	The ID that identifies a specific Edge of Network Node.
DDATA min. time	The minimum-wait-time duration before a new DDATA (Device DATA) message is sent when data change is detected.
QoS	MQTT provides three levels of reliability, which are known as qualities of service (QoS). The reliability of the message determines the persistence of the message. 0: At most once, messages are not persistent. 1: At least once. 2: Exactly once.

Device Tab



Setting	Description
New Group	Add a group to manage the tags.
New Tag	Add the tags of this EoN node monitored by MQTT engine. Please note that the Name field should not be blank.
Delete	Delete an existing group or tag.
Settings	Configure an existing group or tag.

Click the icon to download the demo project. Please confirm your internet connection before downloading the demo project.

42.2. OPC UA Server

42.2.1. Overview

OPC UA (Unified Architecture) is a communication technology often used in industrial automation fields. OPC UA features cross-platform interoperability, unified access, standardized communication, and security. In this architecture, cMT / cMT X Series HMI models with built-in OPC UA server play a key role as Communication Gateway, and allow OPC UA clients to access HMI or PLC data by subscribing to tags to receive real-time updates. This new architecture can help you achieve vertical integration.

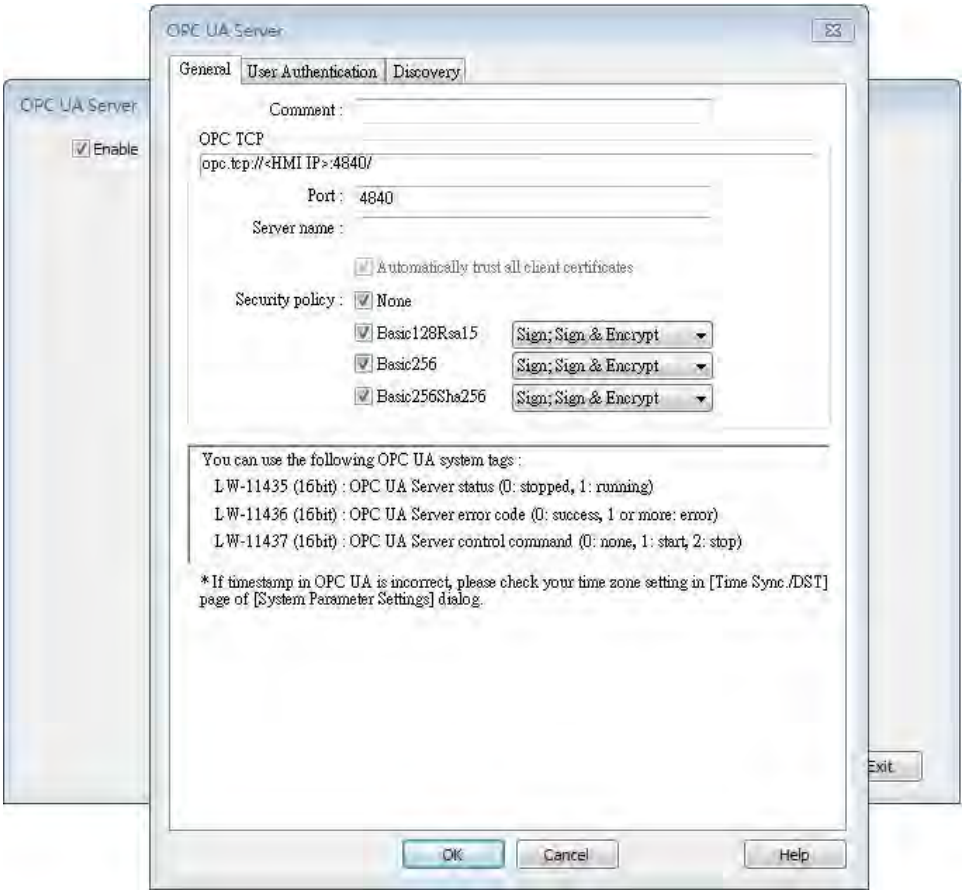
Hardware & Software requirements:

- HMI Model: cMT / cMT X Series models. *A license must be loaded for cMT-SVR / cMT-SVR-200 and cMT-HDM / cMT-FHD / cMT-FHDX.
- Software: EasyBuilder Pro V5.06.01 or later
- Recommended OPC UA Client: Unified Automation UaExpert

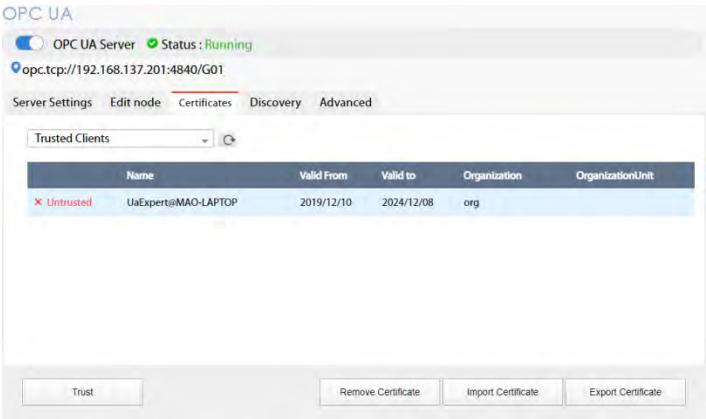
42.2.2. Configuration

Click [Object] » [IIoT] » [OPC UA Server] in the menu to open the settings dialog box.

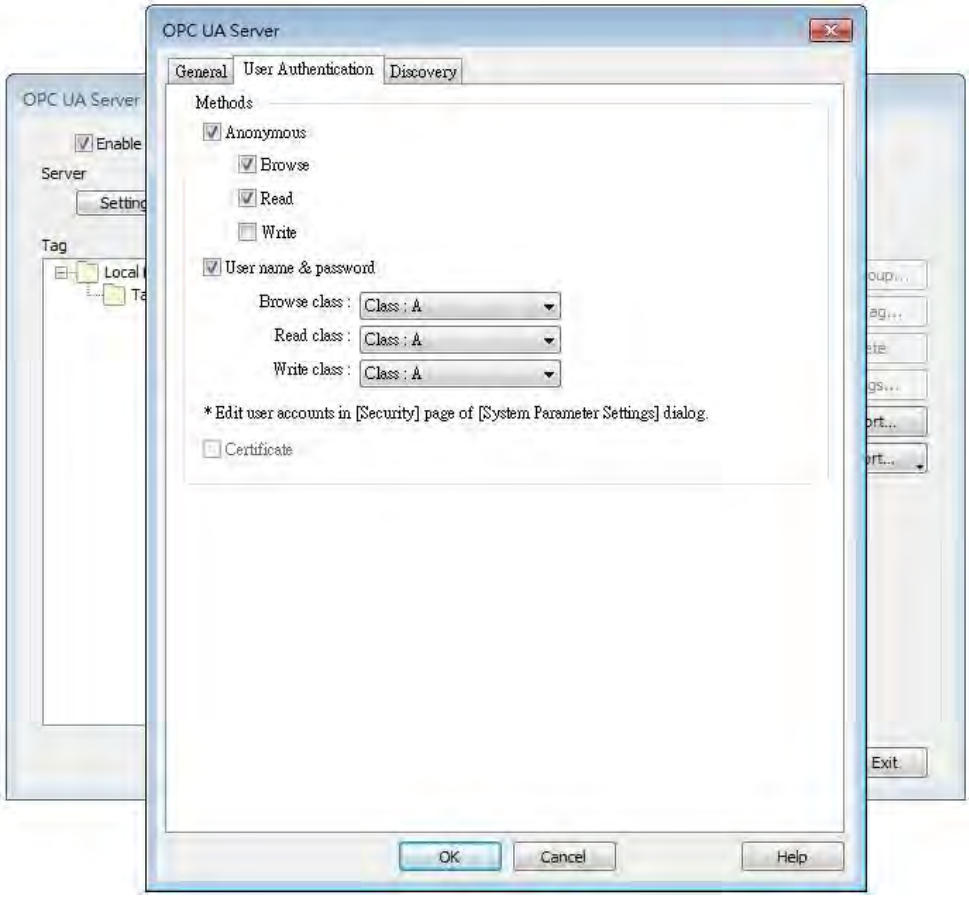
General Tab

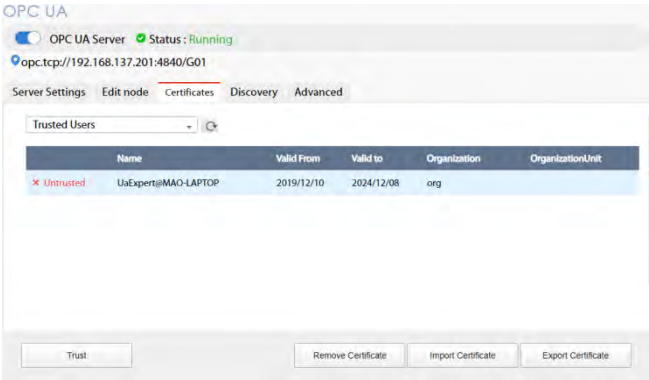


Setting	Description
Comment	The description about the OPC UA Server.
OPC TCP	The URL of the server.
Port	The port number for the clients to connect with OPC UA Server. The default port number is 4840.
Server name	The server name, this field is allowed to remain blank. **[Automatically trust all client certificates]** This option is enabled by default, but may be toggled for cMT Gateway series only. When this option is disabled, all OPC UA clients will be refused connection unless their corresponding client certificates have been trusted in the OPC UA web interface like shown below:

	 Note: According to OPC UA specification, an OPC UA client, in making a connection, will use a client certificate which will be checked by the OPC UA server for its legitimacy. The exception is if the security policy allows for the “None” option.
Security Policy	Configure the security policy and available algorithms that can be used by clients.

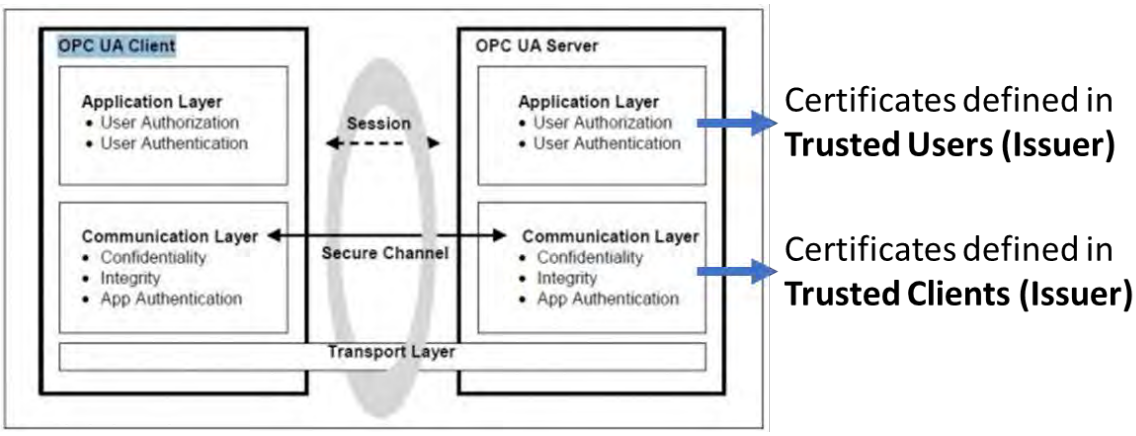
User Authentications Tab



Setting	Description
Methods	**Anonymous** Grant Browse, Read, Write permissions to anonymous login by selecting the checkboxes. **User name & password** Use the same user name and password as HMI. The permissions are granted to the security classes specified in System Parameter Settings » Security. **Certificate** This option is available only for cMT Gateway series. OPC UA client may use certificates as authentication method instead of username-and-password method to login. Use web interface to configure trusted/untrusted user certificates, as shown below: 

Note

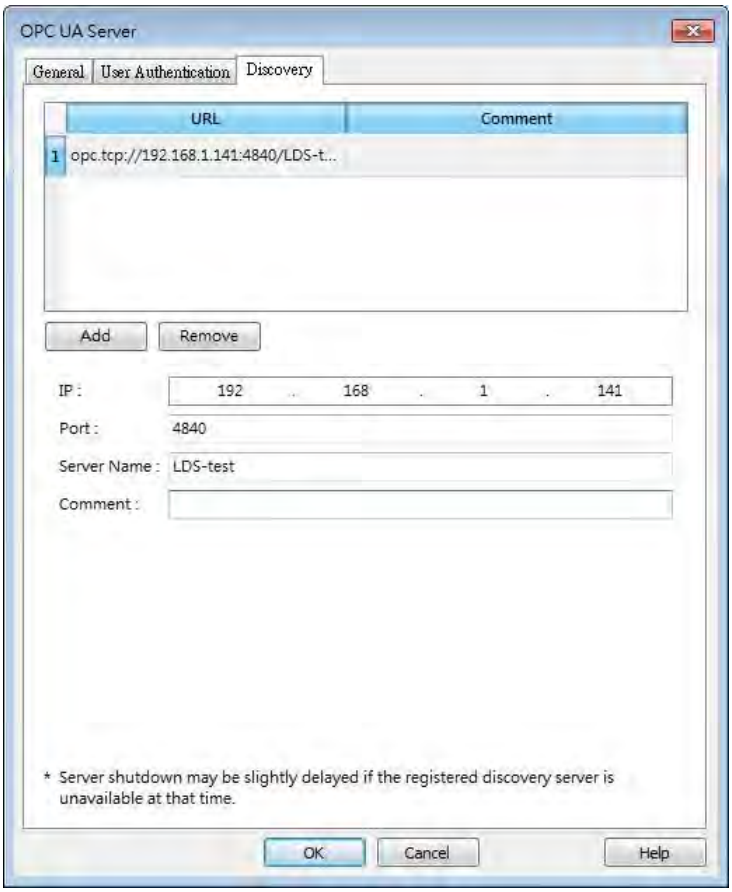
- OPC UA security layers can be split into
  (1) communication layer (e.g. SecurityPolicy)
  (2) application layer, as shown in the image below:



Security Layers (from http://wiki.opcfoundation.org/index.php/File:SecurityLayers.jpg)

- Client certificate is at communication layer and its use is required when using SecurityPolicy other than None.
- User certificate is at application layer and using it is one of the ways for authentication.

Discovery Tab



When configured, OPC UA server will register to the Local Discovery Server (LDS).
OPC UA Discovery service is used to simplify server location maintenance when there are many OPC UA servers in the network. An OPC UA client can access one LDS Server and obtain all registered OPC UA server.

Setting	Description
IP	IP address of the OPC UA client.
Port	Port number used by the OPC UA client.
Server Name	Server name of the OPC UA client.
Comment	A memo on the server and will not influence communication.

Example 1

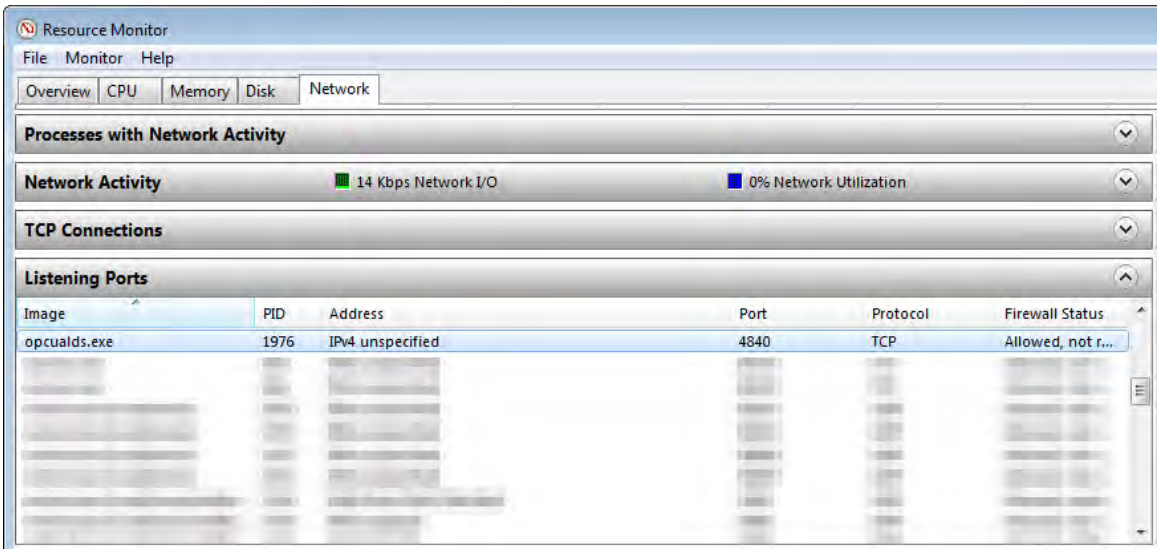
The following is an example showing how to set up Discovery service.

1. Install Local Discover Server (LDS) on a PC (for example, the PC name is DESKTOP-ABCD). Download the LDS provided by OPC Foundation from the link below: https://opcfoundation.org/developer-tools/developer-kits-unified-architecture/local-discovery-server-lds/

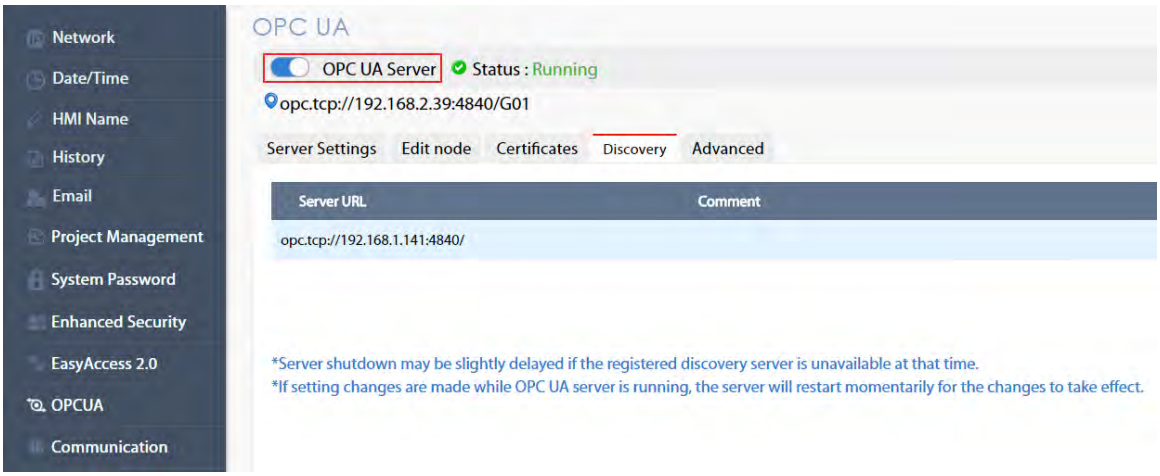
2. If the DNS service of router cannot resolve the HMI name to IP address, the HMI name should be changed to the IP address of the HMI. For example: If HMI IP address is 192.168.1.100, then the HMI name should be 192.168.1.100 or 0.0.0.0.
3. On the PC with OPC UA LDS installed, please manually copy the certificate from folder "C:\ProgramData\OPC Foundation\UA\pki\rejected\certs" (Folder for rejected certificates) to folder "C:\ProgramData\OPC Foundation\UA\pki\trusted\certs" (Folder for trusted certificates).
4. Launch the software of OPC UA Client, enter the name of the PC with OPC UA LDS installed or its IP address to obtain all the registered OPC UA servers.

When Discovery does not work properly, please:

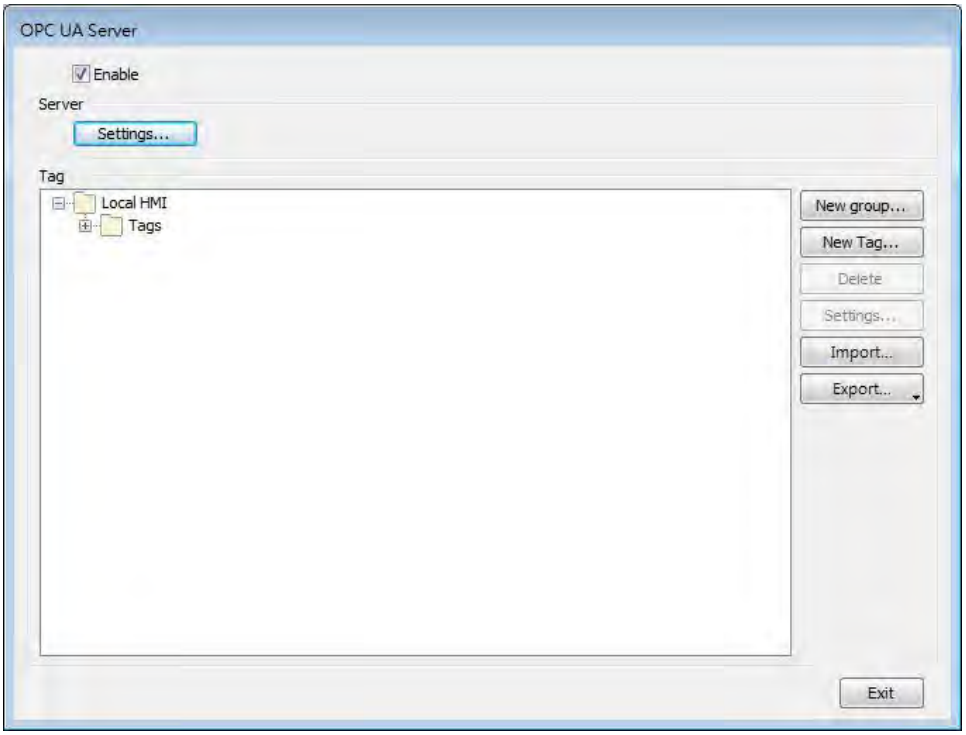
1. Open Windows Task Manager » Performance » Resource Monitor » Network » Listening Ports, and find the port number used by opcualds.exe. As shown in the following screenshot, in this example the PC's opcualds.exe uses port 4840.

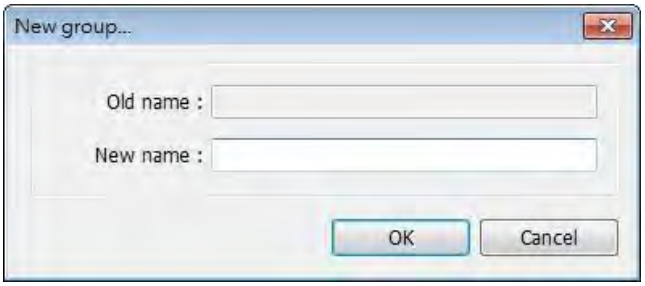


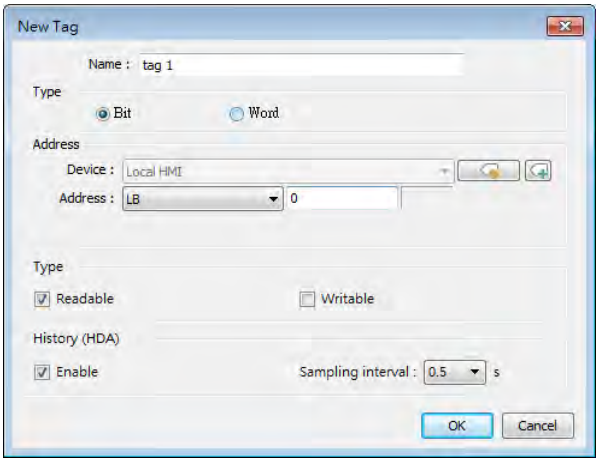
2. Enter HMI's IP address in the web browser, and enter the password to log in. Open OPC UA settings page and restart OPC UA Server. Please note that OPCUA settings tab is only supported on cMT Gateway Series models.



Tag



Setting	Description
New group	 Add a new group for managing tags.

New Tag	 Add a new tag for the client to monitor or control. The name must be entered, and the address can be Readable or Writeable. **History(HDA)** Enable OPC UA HDA.
Delete	Delete an existing group or tag.
Settings	Set an existing group or tag.
Import	Import a tag file. Applicable import formats are: *.xlsx, *. xls, *.csv, *. xml
Export	Export current tags. Applicable export formats are: Excel format or XML format.

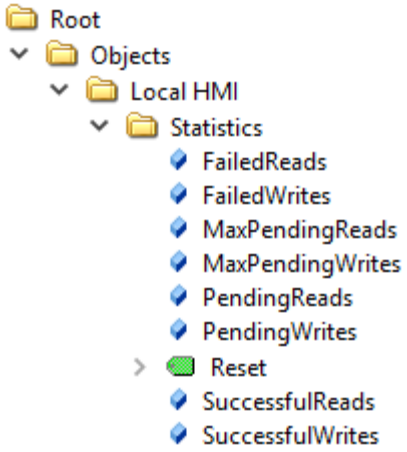
Note

- When downloading the project file to HMI, please make sure that the HMI time and time-zone settings are correct. Otherwise, the client program may not be able to authenticate, and the communication may fail due to authentication error caused by incorrect certificate valid time.

Click the icon to watch the demonstration film. Please confirm your internet connection before playing the film.

42.2.3. Device Statistics

Device-specific statistical data can be found in “Statistics” node, as shown below:



Meaning of each node:

Node Name	Description
FailedReads	No. of failed read commands. If it is not zero, there may be communication errors.
FailedWrites	No. of failed write commands. If it is not zero, there may be communication errors.
MaxPendingReads	Max. no. of pending read commands.
MaxPendingWrties	Max. no. of pending write commands.
PendingReads	No. of pending read commands in the queue. If the number stays high for a long time, it means the communication module is not able to process all commands in time. OPC UA nodes may update slower. Under extreme circumstances (e.g. >30), OPC UA node may not update in a long time.
PendingWrites	No. of pending write commands. Write commands have higher priority than read commands. If PendingWrites stays high, it will affect read commands.
Reset	Reset all statistical data.
SuccsessfulReads	No. of successful read commands.
SuccsessfulWrites	No. of successful write commands.

42.2.4. Limitation

The limitation of OPC UA server are listed below:

Item	Description
OPC UA Profile	UA 1.02 Standard UA Server Profile, including but not limited to * Core Server Facet

	* UA-TCP UA-SC UA-Binary * SecurityPolicy – None * Enhanced DataChange Subscription Server Facet * Standard DataChange Subscription Server Facet * Embedded DataChange Subscription Server Facet * User Token – X509 Certificate Server Facet * User Token – User Name Password Server Facet * Standard DataChange Subscription Server Facet * Embedded DataChange Subscription Server Facet See Profile Reporting Visualization Tool by OPC Foundation for more details.
Security policies	None Basic128Rsa15 Basic256 Basic256Sha256
Number of nodes	15 000
Max. array size	255
Read cache	100ms (Cache will be used for 100ms from previous read)
Max. client sessions	100
Max. subscription per session	64
Min. publishing interval	100ms
OPC UA HDA	Supports up to 50 node addresses with each node address can storing up to 10000 HDA data records. What constitutes a node address? Each HDA-enabled node is considered to take up the number of node addresses that is equal to the Element Count setting. If the data type is String, it is the No. of word setting instead. When the remaining space in HMI memory is less than 10%, the system will delete the earliest data and store the latest data. The system will stop deleting data when the remaining space increases exceeding 10%.
Performance (Values may change for different hardware/EBPro version)	
Max. Read/Subscribe	Built-in registers (e.g. LW): 27000 words/second (WPS)

Throughput (Security: None)	MODBUS RTU@9600bps: 500 WPS MODBUS RTU@115200bps: 4000 WPS MODBUS TCP/IP: 10000 WPS Tested Environment EBPro version: V6.02.02.242 cMT-G02 OS version: 20180917 Test uses as many as words as possible in one node (using array) to optimize reading.

Note

- Examples showing how to count OPC UA HDA node addresses:
  If there are 50 nodes (node1, node2…node50) and each node maps to one bit only(Element Count is 1), all the nodes altogether takes up 50 node addresses.
  If a node maps to a 16-bit unsigned array of size 50 (which is when Element Count is 50), every element in the array takes up a node address, so the number of node addresses taken by this node is 50.
  If a node maps to a string where No. of word is set to 50, the number of node addresses taken by this node is 50.

43. JS Object and JS Library

This manual explains the usage and syntax of JS Object and JS Library.

43.1. Overview 43-2
43.2. JS Object 43-3
43.3. JS Resource 43-5
43.4. Demonstrations 43-7
43.5. Limitations 43-20
43.6. How a JS object operates behind the scenes 43-20

43.1. Overview

JS objects can be used to meet the requirements in HMI applications that may not be achieved with only built-in features in EasyBuilder Pro. By programming in JavaScript, the look and behavior of objects can be freely controlled to achieve the intended goals.

43.1.1. Usage Suggestions

Use the built-in features in EasyBuilder Pro where possible, and use JS objects only when the requirements are difficult or impossible to fulfill by using the built-in features.

43.1.2. Hardware and Software Requirements

- EasyBuilder Pro V6.05.01 or later
- Applicable JavaScript version: ECMAScript 2017 (Not including SharedArrayBuffer and Atomics)
- Eligible HMI model: cMT X Series
- JS objects are not supported on 32-bit Android devices.

43.1.3. Warning

JS objects provide powerful customization features, but using them incorrectly can result in system error or performance degredation. Please use JS objects carefully.

43.1.4. JS Object SDK

For more information on SDK supported by JS object, visit this link:
https://dl.weintek.com/public/Document/JS_Object_SDK/Current/index.html

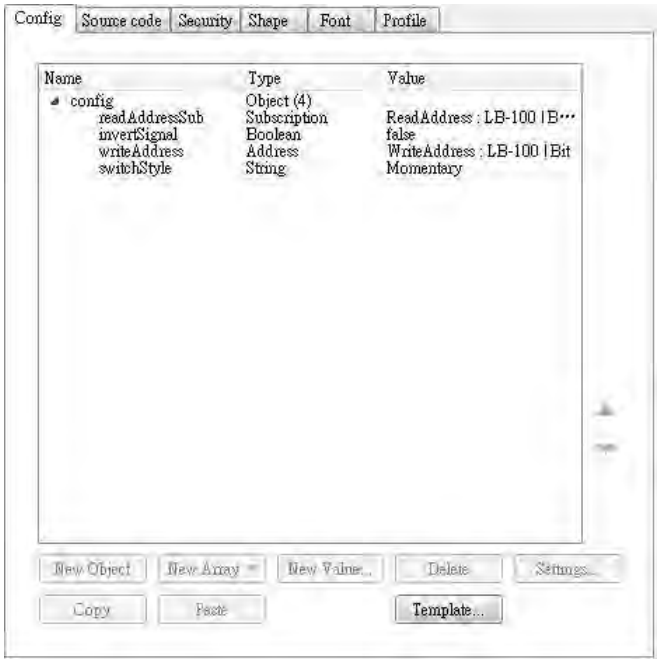
43.2. JS Object

Config Tab

JS object property configuration may contain the following types of data:

1. Address
2. Subscription
3. The parameters defined by users according to the needs to determine the runtime behavior of a JS object.

Configuration will be wrapped into a JavaScript object: 'config' and injected to the JS object at runtime ⇨ *this.config*.

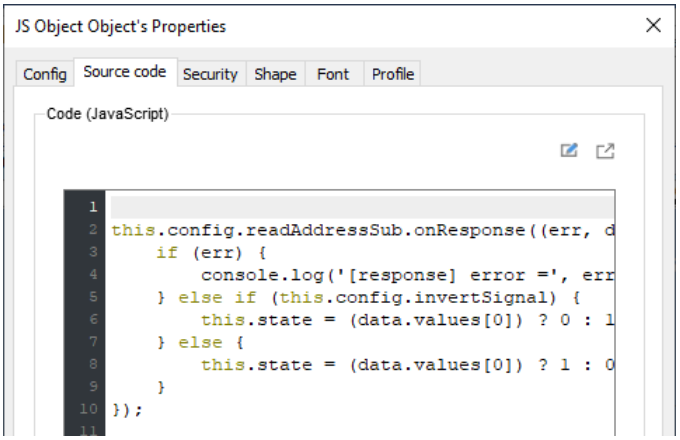


Setting	Description
New Object	Add a new JavaScript object which can have multiple properties and each property has a name and a value.
New Array	Add a new array.
New Value	Add a new property with the following data types to choose from: **String** Maximum allowable size is 1000 words. **Number** Supports 64-bit float. **Boolean** May be True or False.

	Address The address of the device. **Subscription** Address subscription. Address subscription is used to monitor data change of designated device data. Once data changes, the system will notify the subscriber through the callback function registered in Subscription.onResponse method.
Delete	Delete a selected item.
Settings	Edit a selected item.
Copy	Copy the selected item.
Paste	Paste the copied item to the selected place.
Template	Show current configuration in JSON format in an editing window. After editing the JSON content and then clicking the [OK] button, the (JSON) content will be parsed and converted back to the configuration. Users who are familiar with JSON can expedite configuration process using this.

Source Code Tab

The source code is written in JavaScript and it determines the runtime behavior of the JS object.

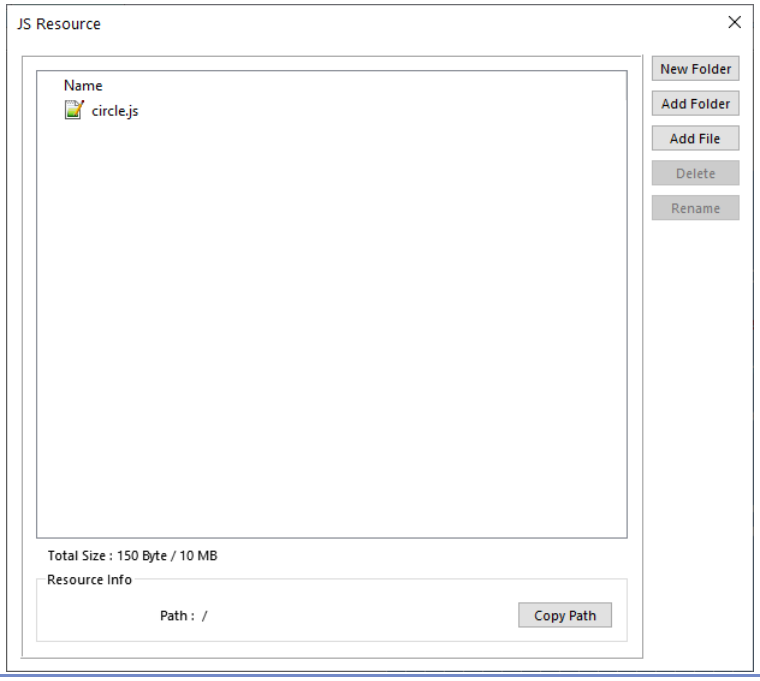


Setting	Description
Compile	Compile the source code to check if it works properly.
	Edit the source code in a separate pop-up window.

43.3. JS Resource

When using external JS modules for JS objects, please add the JS modules into EasyBuilder Pro project's JS Resource first, and then the modules can be imported in the [source code] of JS objects.

The screenshot below shows an example of JS Resource, and the settings are explained in the table below.



Setting	Description
Name	The name of the imported file or folder.
New Folder	Create a new folder.
Add Folder	Import a folder.
Add File	Import a file.
Delete	Delete a file or a folder.
Rename	Rename the folder or JS Resource.
Copy Path	Copy a selected path of a file or a folder for use in the source code of JS object.

Note

- Size limit of JS Resource: 10MB.

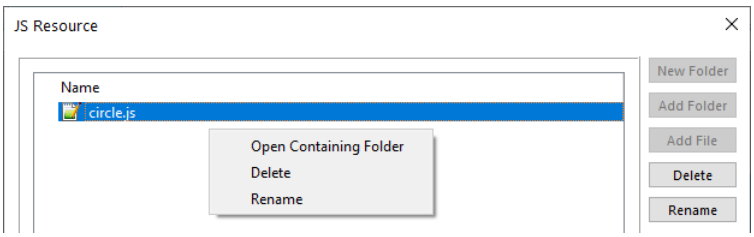
43.3.1. Temporary Folder

When opening a project, EasyBuilder Pro will create a temporary folder and extract JS Resource

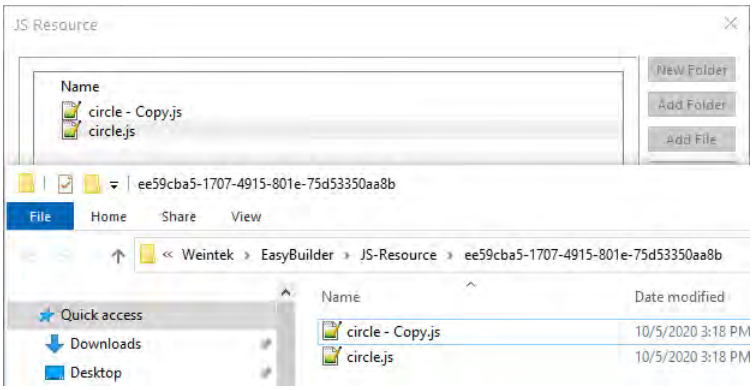
content in this folder. When saving a project, EasyBuilder Pro will synchronize the content in the temporary folder to the project.

43.3.1.1. Opening a Temporary Folder

In JS Resource window select a file or a folder and then right-click the mouse button to open a menu and select [Open Containing Folder].



Changes made in the folder will also be shown in the JS Resource window.



Note

- The temporary folder is generated automatically when opening the project, and is deleted automatically when closing the project.
- The name of the temporary folder is not fixed and is randomly generated when opening the project.
- Making changes in the temporary folder will turn the project into Modified state and the content in the temporary folder will only be synchronized to the project when the user saves the project.

43.4. Demonstrations

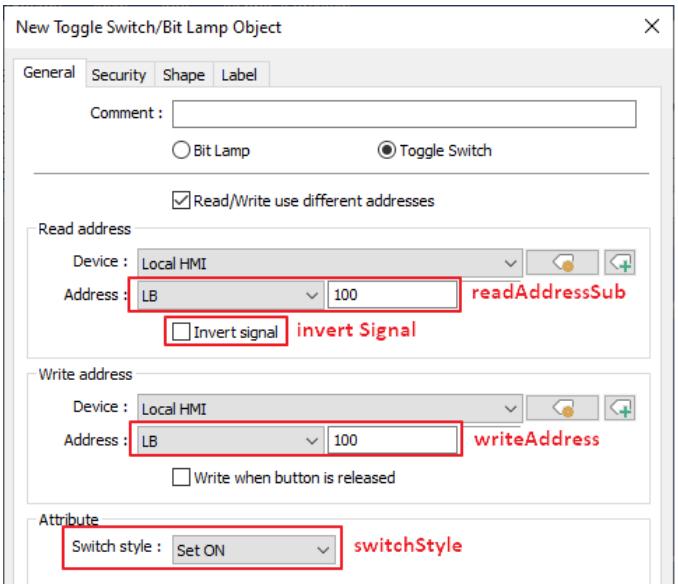
This chapter demonstrates some examples aiming to help users learn how to use JS object. For more information on the functions and classes used for the program code, please see JS Object SDK

43.4.1. Mimic a [Toggle Switch] object

This section aims to familiarize users with the configuration by making JS objects work as Toggle Switches. The configuration steps are as below:

1. Design a JS object that can read the designated address and invert signal.
2. Enable the ability to set the designated address ON for the JS object.
3. Enable the ability to set the designated address OFF for the JS object.
4. Enable the ability to invert the state of the designated address for the JS object.
5. Enable momentary control feature for the JS object.

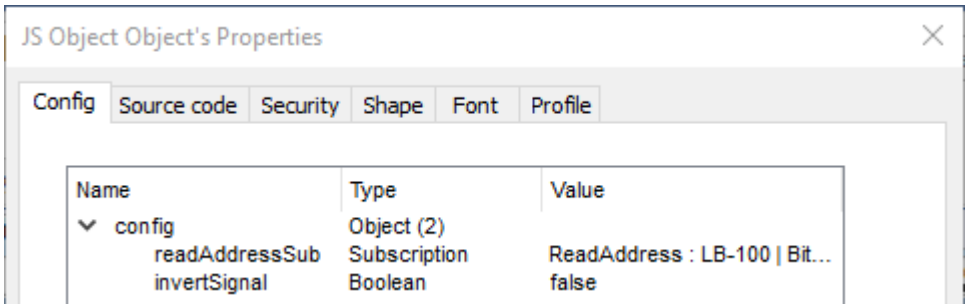
To design a JS object to achieve the functions of a Toggle Switch, the property names of the JS object that correspond to the settings of a Toggle Switch are shown below.

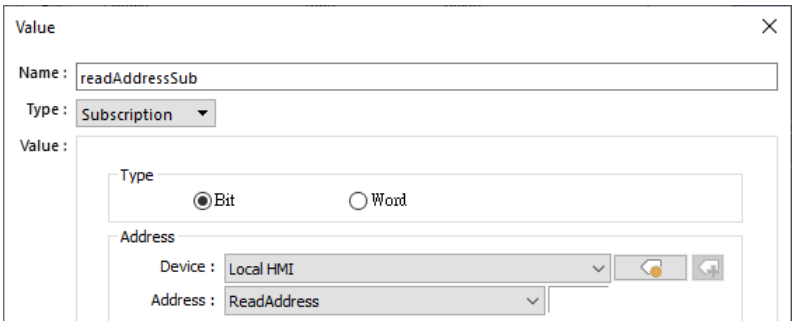


43.4.1.1. JS Object that Reads Address and Invert Signal

This section explains how to create a JS object to read the designated address and invert signal. To achieve the goal, two properties should be added: *readAddressSub* and *invertSignal*.

Config Tab



Setting	Description
readAddressSub	Read the state of the designated bit address. Set [Type] to [Subscription], set [Value Type] to [Bit], and set address to [LB-100]. 
invertSignal	Select whether or not to invert signal. Set [Type] to [Boolean], set [Value] to [false] (don't invert signal)

Source Code Tab

This section explains the source code line by line.

```
1
2  this.config.readAddressSub.onResponse((err, data) => {
3      if (err) {
4          console.log('[response] error =', err.message);
5      } else if (this.config.invertSignal) {
6          this.state = (data.values[0]) ? 0 : 1;
7      } else {
8          this.state = (data.values[0]) ? 1 : 0;
9      }
10 });
11
```

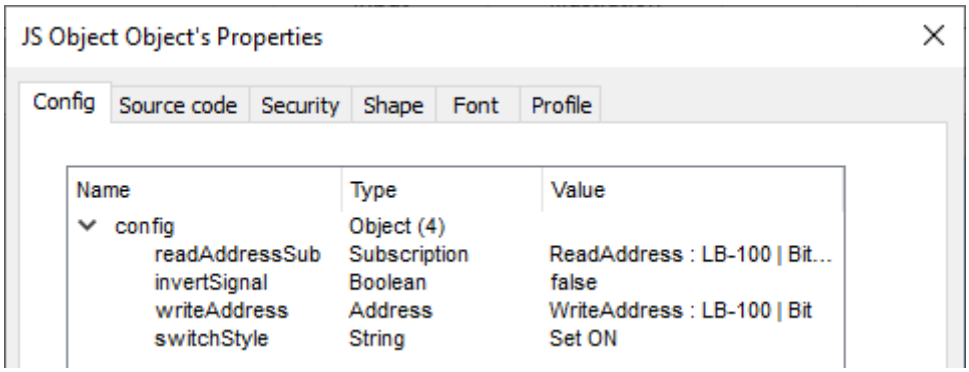
- Line2: ***'this'*** is the JS object. Through ***'this.config'***<object> to obtain the value (***'readAddressSub'*** and ***'invertSignal'***) added in [Config] tab.
- Line2: Call 'onResponse' function of the ***'this.config.readAddressSub'*** < Subscription > and register response callback to get notification when the value of the read address changes.
- Line 3~9: Act according to the response callback.
- Line 3~4: Check whether an error has occurred. Output an error message to the debugging console if an error has occurred (variable err has meaningful value).
- Line 5~8: If variable err is Null, it means that reading from the Read Address is successful and change in data has been detected. Then, set JS object state according to the data read and the ***'invertSignal'*** property.

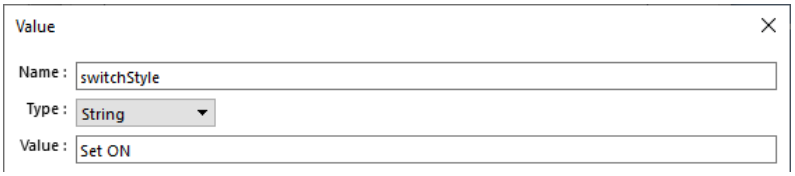
Data in Read Address	Invert Signal	JS Object State
0	False	0
1	False	1
0	True	1
1	True	0

43.4.1.2. Toggle Switch ON Mode

Following the preceding section, this section explains how to make a JS object work as a Toggle Switch that can set ON. To achieve the goal, another two properties should be added: *writeAddress* and *switchStyle*.

Config Tab



Setting	Description
readAddressSub	Read the state of the designated address.
invertSignal	Invert output signal.
writeAddress	Set the address to write to. When the user presses this object, the value will be written to this address according to the value of ***'switchStyle'***.
switchStyle	Set the switch mode and set [Value] to [Set ON]. 

Source Code Tab

```
this.config.readAddressSub.onResponse((err, data) => {
    if (err) {
        console.log('[response] error =', err.message);
    } else if (this.config.invertSignal) {
        this.state = (data.values[0]) ? 0 : 1;
    } else {
        this.state = (data.values[0]) ? 1 : 0;
    }
});

var mouseArea = new MouseArea();
this.widget.add(mouseArea);

mouseArea.on('mousedown', (mouseEvent) => {
    if (this.config.switchStyle === 'Set ON') {
        driver.setData(this.config.writeAddress, 1);
    }
});
```

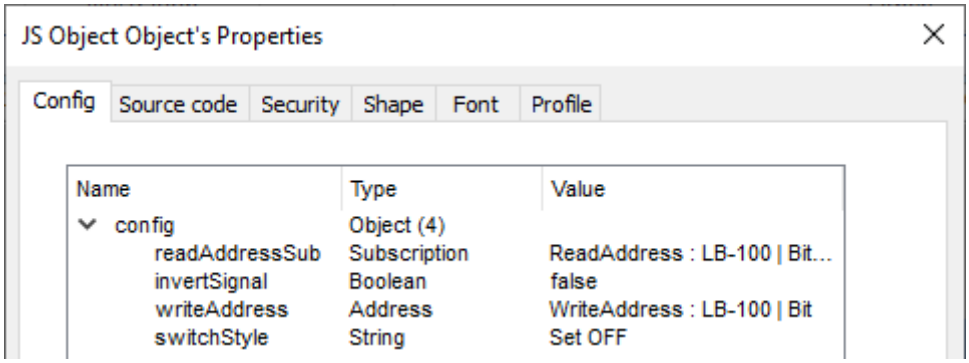
Use MouseArea object to carry out write operation when the object is pressed or released.

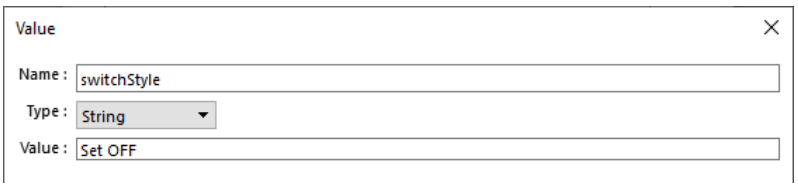
- Line 1~11: Please see Ch43.4.1.1 in this user manual.
- Line 12: Create a MouseArea object named ***'mouseArea'***.
- Line 13: The ***'this.widget'*** <Container> represents the graphical widget of JS object. This widget is responsible for image display and user interaction.
- Line 13: Add ***'mouseArea'*** to JS object.
- Line 15: Register ***'mousedown'*** event listener with ***'mouseArea'***. When the user presses the object, ***'mouseArea'*** will notify the listener with a MouseEvent object.
- Line 15~19: The listener to ***'mousedown'*** event.
- Line 16~18: When [Switch Style] is 'Set ON', value 1 will be written into [Write address] using driver module's setData Method.

43.4.1.3. Toggle Switch OFF Mode

Following the preceding sections, this section explains how to make a JS object work as a Toggle Switch that can set OFF.

Config Tab



Setting	Description
readAddressSub	Read the state of the designated address.
invertSignal	Invert output signal.
writeAddress	Set the address to write to. When the user presses this object, the value will be written to this address according to the value of ***'switchStyle'***.
switchStyle	Set the switch style and set [Value] to [Set OFF]. 

Source Code Tab

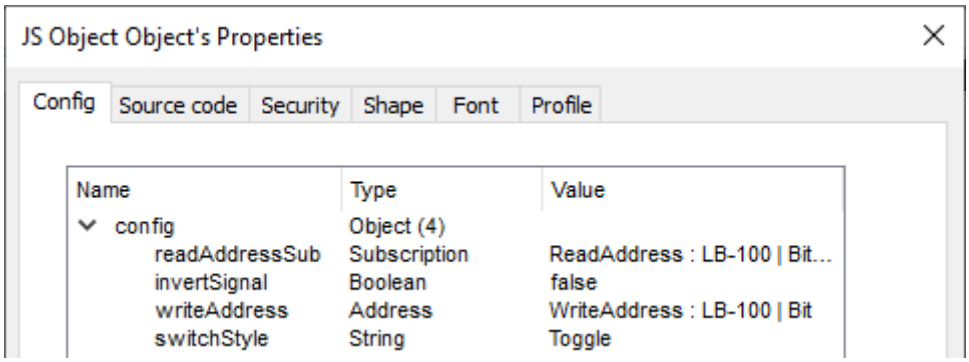
```
1
2  this.config.readAddressSub.onResponse((err, data) => {
3      if (err) {
4          console.log('[response] error =', err.message);
5      } else if (this.config.invertSignal) {
6          this.state = (data.values[0]) ? 0 : 1;
7      } else {
8          this.state = (data.values[0]) ? 1 : 0;
9      }
10 });
11
12 var mouseArea = new MouseArea();
13 this.widget.add(mouseArea);
14
15 mouseArea.on('mousedown', (mouseEvent) => {
16     if (this.config.switchStyle === 'Set ON') {
17         driver.setData(this.config.writeAddress, 1);
18     } else if (this.config.switchStyle === 'Set OFF') {
19         driver.setData(this.config.writeAddress, 0);
20     }
21 });
22
```

- Line 1~17: Please see Ch43.4.1.2 in this user manual.
- Line 18~19: When [Switch Style] is 'Set OFF', value 0 will be written into [Write address].

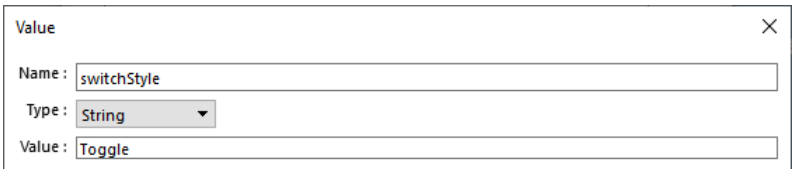
43.4.1.4. Toggle Switch Toggle Mode

Following the preceding sections, this section explains how to make a JS object work as a Toggle Switch in Toggle mode, which inverts the state of the designated address.

Config Tab



Setting	Description
readAddressSub	Read the state of the designated address.

invertSignal	Invert output signal.
writeAddress	Set the address to write to. When the user presses this object, the value will be written to this address according to the state of ***'readAddressSub'*** and the value of ***'switchStyle'***.
switchStyle	Set the switch style and set [Value] to [Toggle]. 

Source Code Tab

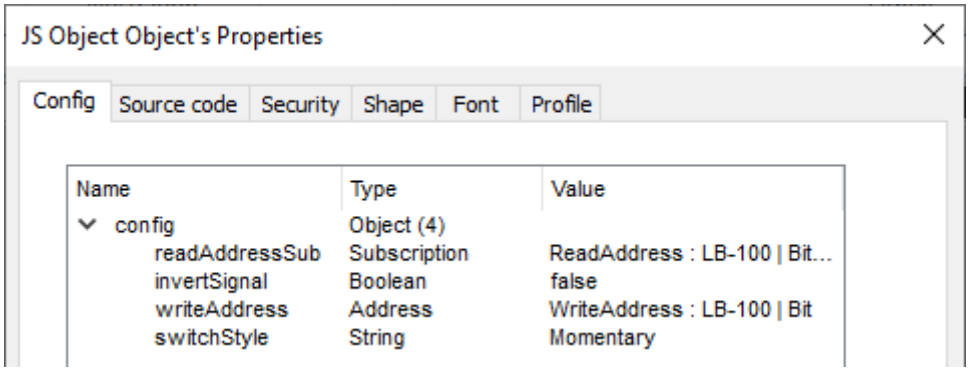
```
1
2  this.config.readAddressSub.onResponse((err, data) => {
3      if (err) {
4          console.log('[response] error =', err.message);
5      } else if (this.config.invertSignal) {
6          this.state = (data.values[0]) ? 0 : 1;
7      } else {
8          this.state = (data.values[0]) ? 1 : 0;
9      }
10 });
11
12 var mouseArea = new MouseArea();
13 this.widget.add(mouseArea);
14
15 mouseArea.on('mousedown', (mouseEvent) => {
16     if (this.config.switchStyle === 'Set ON') {
17         driver.setData(this.config.writeAddress, 1);
18     } else if (this.config.switchStyle === 'Set OFF') {
19         driver.setData(this.config.writeAddress, 0);
20     } else if (this.config.switchStyle === 'Toggle') {
21         let currentValue = this.config.readAddressSub.latestData.values[0];
22         let newValue = (currentValue) ? 0 : 1;
23         driver.setData(this.config.writeAddress, newValue);
24     }
25 });
```

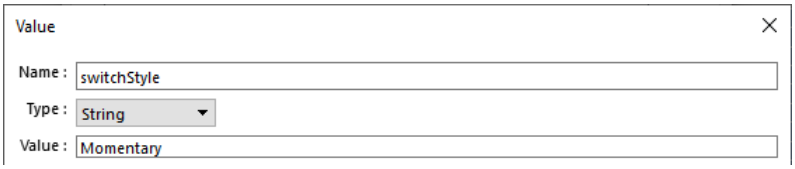
- Line 1~19: Please see Ch43.4.1.3 in this user manual.
- Line 20~23: When [Switch Style] is 'Toggle', [Write address] will be set to an inverted state.

43.4.1.5. Toggle Switch Momentary Mode

Following the preceding sections, this section explains how to make a JS object work as a Toggle Switch in Momentary mode.

Config Tab



Setting	Description
readAddressSub	Read the state of the designated address.
invertSignal	Invert output signal.
writeAddress	Set the address to write to. When the user presses this object, value 1 will be written to this address and when the user releases this object, value 0 will be written to this address.
switchStyle	Set the switch mode and set [Value] to [Momentary]. 

Source Code Tab

```
1
2  this.config.readAddressSub.onResponse((err, data) => {
3      if (err) {
4          console.log('[response] error =', err.message);
5      } else if (this.config.invertSignal) {
6          this.state = (data.values[0]) ? 0 : 1;
7      } else {
8          this.state = (data.values[0]) ? 1 : 0;
9      }
10 });
11
12 var mouseArea = new MouseArea();
13 this.widget.add(mouseArea);
14
15 mouseArea.on('mousedown', (mouseEvent) => {
16     if (this.config.switchStyle === 'Set ON') {
17         driver.setData(this.config.writeAddress, 1);
18     } else if (this.config.switchStyle === 'Set OFF') {
19         driver.setData(this.config.writeAddress, 0);
20     } else if (this.config.switchStyle === 'Toggle') {
21         let currentValue = this.config.readAddressSub.latestData.values[0];
22         let newValue = (currentValue) ? 0 : 1;
23         driver.setData(this.config.writeAddress, newValue);
24     } else if (this.config.switchStyle === 'Momentary') {
25         driver.setData(this.config.writeAddress, 1);
26     }
27 });
28
29 mouseArea.on('mouseup', (mouseEvent) => {
30     if (this.config.switchStyle === 'Momentary') {
31         driver.setData(this.config.writeAddress, 0);
32     }
33 });
34
```

- Line 1~23: Please see Ch43.4.1.4 in this user manual.
- Line 24~25: When [Switch Style] is 'Momentary', value 1 will be written into [Write address].
- Line 29~33: When the user releases the object, value 0 will be written into [Write address].

43.4.2. Examples of JS Resource Usage

This section aims to familiarize users with the ways to import JS module in JS Resource using JS objects.

JS object supports the following ways to import modules:

1. ES6 dynamic import: Use import() function. Please note that ES6 static import is not supported by JS object.
2. CommonJS: Use require() function.

43.4.2.1. ES6 Dynamic Import

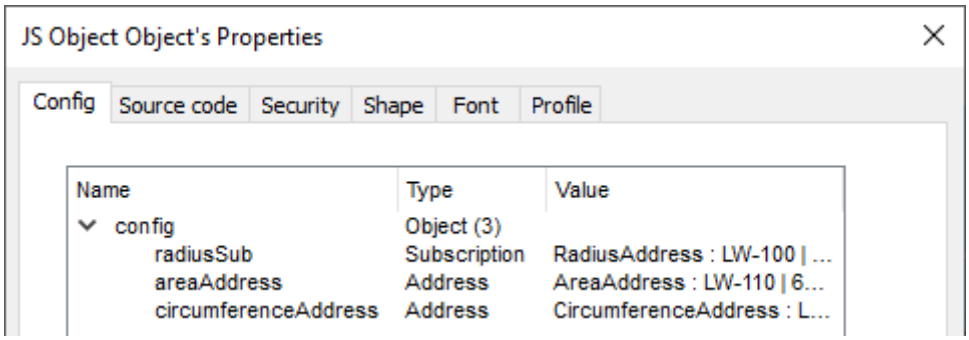
The following demonstration shows how to import JS modules using ES6 dynamic import.

In the JS object, when radius changes, the area and circumference of the circle can be calculated using the ***area*** and ***circumference*** functions in JS module **(*circle.js*)**, and the result is output to the designated addresses.

The program code of circle.js:

```
// circle.js

export function area(radius) {
  return Math.PI * radius * radius;
}

export function circumference(radius) {
  return 2 * Math.PI * radius;
}
```

1. Settings for creating a JS object are shown below:



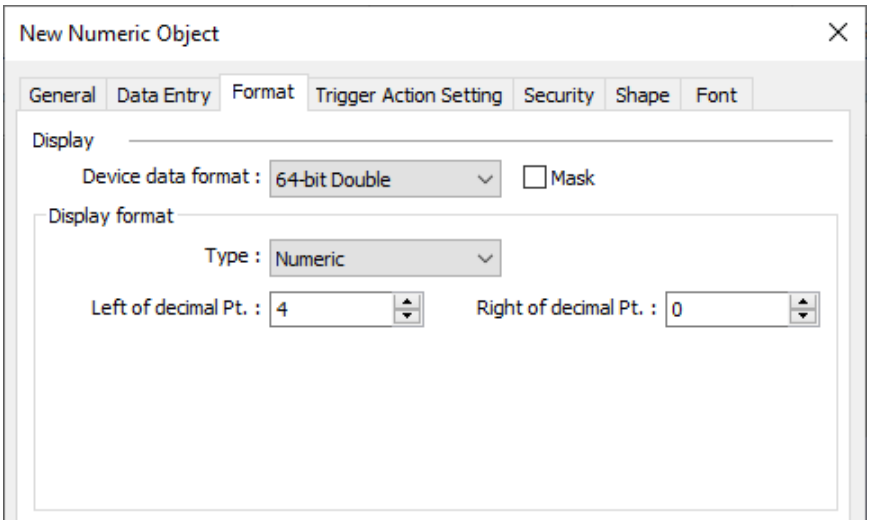
Setting	Description
radiusSub	Obtains current radius value and monitors its changes. (Select LW-100, 64-bit Double)
areaAddress	The address for outputting the calculation result of the area of the circle. (Select LW-110, 64-bit Double)
circumferenceAddress	The address for outputting the calculation result of the circumference of the circle. (Select LW-120, 64-bit Double)

2. Edit the program code as shown below in [Source code] settings page:

```
1  const circle = await import('/circle.js');
2
3  this.config.radiusSub.onResponse((err, data) => {
4      if (err) {
5          console.log('[response] error =', err.message);
6      } else {
7          let radius = data.values[0];
8          let area = circle.area(radius);
9          let circumference = circle.circumference(radius);
10         driver.setData(this.config.areaAddress, area);
11         driver.setData(this.config.circumferenceAddress, circumference);
12     }
13 });
```

- Line 1: Import the '/circle.js' file in JS Resource and bind the result to the ***circle***.
- Line 3: Call ***'this.config.radiusSub'*** 'onResponse' function to register response callback for notification of changes in [circle radius].
- Line 8: Calculate area of the circle by calling the method in ***circle.js*** and store the result in ***area*** variable.
- Line 9: Calculate circumference of the circle by calling the method in ***circle.js*** and store the result in ***circumference*** variable.
- Line 10~11: Output the calculation result of the area and the circumference of the circle respectively to 'areaAddress' and 'circumferenceAddress'.

3. In the project, create three Numeric objects (64-bit Double) and set the address to LW-100 (Radius), LW-110 (Area), and LW-120 (Circumference).



4. When values are entered in LW-100 (Radius), the results are automatically updated at LW-110 (Area), and LW-120 (Circumference).

Note

- To import JS module's "default export", please refer to the code below.

```
var myModule = (await import('my-module.js')).default;
```

43.4.2.2. CommonJS

The following demonstration shows how to import JS modules using require() function. This JS object behaves the same as using ES6 but in this section the external JS module is imported using CommonJS.

The program code of circle.js:

```
// circle.js

exports.area = function (radius) {
  return Math.PI * radius * radius;
};

exports.circumference = function (radius) {
  return 2 * Math.PI * radius;
}
```

1. Property settings are the same as explained in the preceding section.
2. Edit the program code as shown below in [Source code] settings page:

```
1  const circle = require('/circle.js');
2
3  this.config.radiusSub.onResponse((err, data) => {
4      if (err) {
5          console.log('[response] error =', err.message);
6      } else {
7          let radius = data.values[0];
8          let area = circle.area(radius);
9          let circumference = circle.circumference(radius);
10         driver.setData(this.config.areaAddress, area);
11         driver.setData(this.config.circumferenceAddress, circumference);
12     }
13 });
14
```

- Line 1: Import the '/circle.js' file in JS Resource and bind the result to the ***circle***.
- Line 3~13: See Ch43.4.2.1 in this user manual.

3. In the project, create three Numeric objects (64-bit Double) and set the address to LW-100 (Radius), LW-110 (Area), and LW-120 (Circumference).
4. When values are entered in LW-100 (Radius), the results are automatically updated at LW-110 (Area), and LW-120 (Circumference).

43.5. Limitations

1. Size limit of a JS object's source code: 100 KB.
2. Size limit of a JS Resource file: 10MB
3. Size limit of the memory usage of a JS context(*): 20 MB.

(*) A JS context is a sandboxed execution context with its own set of built-in objects and functions. And, all the JS objects in the same window share one JS context, that is, share the same memory heap and global object.

43.6. How a JS object operates behind the scenes

1. Create JS context.
2. Create JS object.
3. Initialize JS object 'config' (⇨ *this.config*).
4. Initialize JS object 'widget' (⇨ *this.widget*).
5. Wait for the reply from all Subscriptions.
6. Wrap JS object source code into an async. function and call it.

Appendix A.

Comparison of HMI Software Features

- eMT Series: eMT3070B, eMT3105P, eMT3120A, eMT3150A
- cMT Series: cMT-SVR-100, cMT-SVR-102, cMT-SVR-200, cMT-SVR-202, cMT-iV5, cMT3151, cMT3090, cMT3071, cMT3072, cMT3103, cMT-HDM, cMT-FHD, cMT-FHDX-220
- cMT X Series: cMT3092X, cMT3152X, cMT3072X, cMT3072XH, cMT3102X, cMT3161X, cMT3162X, cMT2078X, cMT2108X, cMT2158X, cMT2166X, cMT1106X
- cMT Series-Gateway: cMT-G01, cMT-G02, cMT-G03, cMT-G04, cMT-CTRL01
- mTV Series: mTV-100
- iE Series: MT8050iE, MT8053iE, MT6070iE, MT6071iE, MT8070iE, MT8071iE, MT8073iE, MT8100iE, MT8101iE, MT8102iE, MT8103iE
- iER Series: MT8070iER1
- XE Series: MT8090XE, MT8091XE, MT8092XE, MT8121XE, MT8150XE
- iP Series: MT6051iP, MT8051iP, MT6071iP, MT8071iP, MT6103iP, MT8102iP

Please note that the table below is for reference only and may not include all features. Some features are version-dependent, so please confirm again by cross-checking with the software or consulting Weintek service when in doubt.

iP/iE/eMT/mTV/XE Series

Category	Model / Function	iP	iE	eMT	mTV	XE	Note
EasyBuilder	Project size limit	22.5MB	22.5MB	64MB	64MB	64MB	
	USB Disk Download	Y	Y	Y	Y	Y	
Function	64-bit Data Type	N/A	N/A	N/A	N/A	N/A	
	3G/4G Dongle	Y	N/A	N/A	N/A	N/A	1
	Control Token	N/A	N/A	N/A	N/A	N/A	
	Dual USB	Y	Y	N/A	N/A	N/A	
	e-Mail	N/A	Y	Y	Y	Y	
	Enhanced Security Mode	Y	Y	Y	Y	Y	

Comparison of HMI Software Features

Category	Function \ Model	iP	iE	eMT	mTV	XE	Note
	Ethernet Printer	N/A	N/A	N/A	N/A	N/A	
	FTP	N/A	N/A	N/A	N/A	N/A	
	LDAP	N/A	N/A	N/A	N/A	N/A	
	String Table	N/A	Y	Y	Y	Y	
	Time Synchronization (NTP)	N/A	Y	Y	Y	Y	
	USB Tethering	N/A	Y	Y	Y	Y	2
	User-defined Startup Screen	Y	Y	Y	Y	Y	
	VNC Server	Y	Y	Y	Y	Y	3
	Web Streaming	N/A	N/A	Y	Y	Y	4
	Web Interface	N/A	N/A	N/A	N/A	N/A	
	EasyWeb 2.0	N/A	N/A	N/A	N/A	N/A	
	WebView	N/A	N/A	N/A	N/A	N/A	
Hardware	Audio (Speaker)	N/A	N/A	eMT3120 eMT3150	N/A	N/A	10
	Audio (Other)	N/A	N/A	N/A	HDMI	USB	
	CAN Bus	N/A	N/A	Y	N/A	8091XE 8092XE	
	MPI	Y	Y	Y	Y	Y	
	WiFi	N/A	8103iE	N/A	N/A	N/A	
	Vibration Feedback	N/A	N/A	N/A	N/A	N/A	
Object	Action Trigger	N/A	N/A	N/A	N/A	N/A	
	BACnet Schedule	N/A	N/A	N/A	N/A	N/A	
	Barcode Scanner (Android Camera)	N/A	N/A	N/A	N/A	N/A	5
	Combo Button	Y	Y	Y	Y	Y	
	Condition	N/A	N/A	N/A	N/A	N/A	
	Date/Time	Y	Y	Y	Y	Y	
	Dynamic Drawing	Y	Y	Y	Y	Y	
	Dynamic Scale	Y	Y	Y	Y	Y	
	File Browser	Y	Y	Y	Y	Y	
	Flow Block	Y	Y	Y	Y	Y	

Comparison of HMI Software Features

Category	Function \ Model	iP	iE	eMT	mTV	XE	Note
	JS Object	N/A	N/A	N/A	N/A	N/A	
	Media Player	N/A	N/A	eMT3105 eMT3121 eMT3150	N/A	Y	6
	PDF Reader	N/A	N/A	N/A	N/A	N/A	
	Picture Viewer	Y	Y	Y	Y	Y	
	Pie Chart	Y	Y	Y	Y	Y	
	PLC Web Browser	N/A	N/A	N/A	N/A	N/A	11
	Table	Y	Y	Y	Y	Y	
	Touch Gesture	N/A	N/A	N/A	N/A	N/A	
	Video In - IP Camera	N/A	N/A	Y	N/A	Y	7
	Video In - USB Camera	N/A	N/A	Y	Y	Y	12
	Video In - Video Input	N/A	N/A	eMT3121 eMT3150	N/A	N/A	
	VNC Viewer	N/A	Y	Y	Y	Y	8
IIoT/Energy	Demand Display	N/A	N/A	N/A	N/A	N/A	
	Demand Setting	N/A	N/A	N/A	N/A	N/A	
	MQTT	N/A	Y	Y	Y	Y	
	MQTT – Advanced JSON	N/A	N/A	N/A	N/A	N/A	
	MQTT - AWS IoT, Sparkplug B, Azure IoT Hub, Google Cloud IoT Core	N/A	N/A	N/A	N/A	N/A	
	OPC UA Client	Y	Y	Y	Y	Y	
	OPC UA Server	N/A	N/A	N/A	N/A	N/A	
Data/History	Circular Trend Display	Y	Y	Y	Y	Y	
	Database Server	N/A	N/A	N/A	N/A	N/A	
	Event Bar Chart	N/A	N/A	N/A	N/A	N/A	
	Operation Log Settings	Y	Y	Y	Y	Y	
	SQL Query	N/A	N/A	N/A	N/A	N/A	
	Recipe Database	Y	Y	Y	Y	Y	
	Recipe Import / Export	Y	Y	Y	Y	Y	
	Recipe View	Y	Y	Y	Y	Y	

Comparison of HMI Software Features

Category	Function \ Model	iP	iE	eMT	mTV	XE	Note
Other	CODESYS	N/A	N/A	N/A	N/A	N/A	
Utility	cMT Viewer App	N/A	N/A	N/A	N/A	N/A	
	cMT Diagnoser	N/A	N/A	N/A	N/A	N/A	
	EasyAccess 1.0	Y	Y	Y	Y	Y	
	EasyDiagnoser	Y	Y	Y	Y	Y	
	EasyPrinter	Y	Y	Y	Y	Y	
	EasySimulator	Y	Y	Y	Y	Y	
	EasySystemSetting	Y	Y	Y	Y	Y	9
	EasyWatch	Y	Y	Y	Y	Y	
Weincloud	Dashboard	N/A	N/A	N/A	N/A	N/A	
	EasyAccess 2.0	Y	Y	Y	Y	Y	
	EasyAccess 2.0 Built-in License	N/A	8053iE 8073iE 8103iE	N/A	N/A	8092XE	

cMT Series

Category	Function \ Model	cMT-SVR	cMT3071 cMT3072	cMT3090	cMT3103	cMT3151	cMT-HDM cMT-FHD	cMT-G01 cMT-G02 cMT-G03 cMT-G04 cMT-CTRL01	cMT3092X cMT3102X cMT3152X cMT3162X	cMT-FHDX-220	cMT3072X cMT3072XH	cMT2166X cMT3161X	cMT2078X cMT2108X cMT2158X	cMT1106X	Note
EasyBuilder	Project size limit	32MB	64MB	64MB	64MB	64MB	64MB	32MB	64MB	64MB	64MB	64MB	64MB	64MB	
	USB Disk Download	N/A	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
Function	64-bit Data Type	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	3G/4G Dongle	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	1
	Control Token	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	N/A	
	Dual USB	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Y	Y	Y	Y	Y	Y	
	e-Mail	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	Enhanced Security Mode	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	Ethernet Printer	N/A	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	FTP	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	

Category	Function \ Model	cMT-SVR	cMT3071 cMT3072	cMT3090	cMT3103	cMT3151	cMT-HDM cMT-FHD	cMT-G01 cMT-G02 cMT-G03 cMT-G04 cMT-CTRL01	cMT3092X cMT3102X cMT3152X cMT3162X	cMT-FHDX-220	cMT3072X cMT3072XH	cMT2166X cMT3161X	cMT2078X cMT2108X cMT2158X	cMT1106X	Note
	LDAP	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A	
	String Table	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	Time Synchronization (NTP)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	USB Tethering	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	2
	User-defined Startup Screen	N/A	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	VNC Server	N/A	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	3
	Web Streaming	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	N/A	4
	Web Interface	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	EasyWeb 2.0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Y	Y	Y	Y	Y	Y	
	WebView	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Y	Y	Y	Y	Y	N/A	
Hardware	Audio (Speaker)	N/A	N/A	Y	N/A	Y	N/A	N/A	cMT3092X cMT3152X cMT3162X	N/A	N/A	Y	cMT2158X	N/A	10
	Audio (Other)	N/A	N/A	N/A	N/A	N/A	HDMI	N/A	N/A	HDMI	N/A	N/A	N/A	N/A	
	CAN Bus	N/A	Y	Y	Y	Y	N/A	N/A	Y	N/A	Y	N/A	N/A	N/A	
	MPI	Y	Y	Y	Y	Y	Y	Y	Y	Y	cMT3072XH	N/A	N/A	N/A	
	WiFi	cMT-SVR-200/202	N/A	N/A	Y	N/A	N/A	cMT-G02	cMT3102X (M02 module)	N/A	N/A	N/A	N/A	M02 module	
	Vibration Feedback	N/A	N/A	N/A	N/A	N/A	N/A	N/A	cMT3162X	N/A	N/A	N/A	N/A	N/A	
Object	Action Trigger	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	N/A	
	BACnet Schedule	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	Barcode Scanner (Android Camera)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	N/A	N/A	N/A	5
	Combo Button	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	Condition	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A	
	Date/Time	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	Dynamic Drawing	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	Dynamic Scale	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	

Category	Function \ Model	cMT-SVR	cMT3071 cMT3072	cMT3090	cMT3103	cMT3151	cMT-HDM cMT-FHD	cMT-G01 cMT-G02 cMT-G03 cMT-G04 cMT-CTRL01	cMT3092X cMT3102X cMT3152X cMT3162X	cMT-FHDX-220	cMT3072X cMT3072XH	cMT2166X cMT3161X	cMT2078X cMT2108X cMT2158X	cMT1106X	Note
	File Browser	N/A	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	Flow Block	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	JS Object	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Y	Y	Y	Y	Y	N/A	
	Media Player	N/A	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	N/A	6
	PDF Reader	N/A	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	Picture Viewer	N/A	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	Pie Chart	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	PLC Web Browser	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Y	N/A	Y	N/A	N/A	N/A	11
	Table	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	Touch Gesture	N/A	N/A	N/A	N/A	N/A	N/A	N/A	cMT3152X cMT3162X	N/A	N/A	Y	N/A	N/A	
	Video In - IP Camera	N/A	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	N/A	7
	Video In - USB Camera	N/A	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	N/A	12
	Video In - Video Input	N/A	N/A	N/A	N/A	Y	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	VNC Viewer	N/A	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	8
IIoT/Energy	Demand Display	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	N/A	N/A	N/A	
	Demand Setting	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	N/A	N/A	N/A	
	MQTT	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	MQTT – Advanced JSON	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	MQTT - AWS IoT, Sparkplug B, Azure IoT Hub, Google Cloud IoT Core	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A	
	OPC UA Client	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	OPC UA Server	License	Y	Y	Y	Y	License	Y	Y	License	Y	N/A	N/A	N/A	
Data/History	Circular Trend Display	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	Database Server	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A	N/A	N/A	
	Event Bar Chart	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	N/A	N/A	N/A	
	Operation Log Settings	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	

Category	Function \ Model	cMT-SVR	cMT3071 cMT3072	cMT3090	cMT3103	cMT3151	cMT-HDM cMT-FHD	cMT-G01 cMT-G02 cMT-G03 cMT-G04 cMT-CTRL01	cMT3092X cMT3102X cMT3152X cMT3162X	cMT-FHDX-220	cMT3072X cMT3072XH	cMT2166X cMT3161X	cMT2078X cMT2108X cMT2158X	cMT1106X	Note
	SQL Query	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A	N/A	N/A	
	Recipe Database	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	Recipe Import / Export	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
	Recipe View	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	
Other	CODESYS	N/A	License	License	N/A	License	N/A	cMT-CTRL01	License	License	License	N/A	License	N/A	
Utility	cMT Viewer App	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	N/A	
	cMT Diagnoser	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	EasyAccess 1.0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	EasyDiagnoser	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	EasyPrinter	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	EasySimulator	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
	EasySystemSetting	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	Y	Y	Y	9
	EasyWatch	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
Weincloud	Dashboard	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Y	Y	Y	Y	Y	Y	
	EasyAccess 2.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	EasyAccess 2.0 Built-in License	cMT-SVR-102/202	cMT3071: N/A cMT3072: Y	Y	Y	Y	N/A	N/A	cMT3092X cMT3152X	N/A	N/A	N/A	N/A	N/A	

Notes

USB Tethering is not supported on MT6071iE

OS Support

No	Function	OS Version
1	3G/4G Dongle	iP: 20201007 cMT-SVR: 20151127 cMT3071/3072/3090/3103/3151: 20180723 cMT-HDM/FHD/FHDX: 20190130 or later

2	USB Tethering	iE Type A: 20160707 iE Type B: 20160325 eMT: 20160601 mTV: 20160527 XE: 20160503 cMT-SVR: 20160518 cMT3090/3103/3151/HDM: 20160412 or later
3	VNC Server	cMT3090/3151: 20160412 or later
4	Web Streaming	cMT-SVR: 20151127, cMT3151:20171129 or later
5	Barcode Scanner Android	Android cMTViewer V1.7.0 or later
6	Media Player	cMT-HDM: no 1920*1080 resolution cMT3072X(H): 20200929
7	Video-in IP Camera	eMT: 20160601 XE: 20160503 cMT3072X(H): 20200929
8	VNC Viewer	iE Type A: 20160707 iE Type B: 20160325 eMT: 20160601 mTV: 20160527 XE: 20160503 all cMT series: 20181211 or later
9	EasySystemSetting	all cMT series: 20180723 or later
10	Audio Output	USB Audio Output OS Support: XE 20190325
11	PLC Web Browser	cMT3072X(H):20200723, cMT3092X/3152X/3162X: 20201202
12	Video In – USB Camera	cMT3072X(H): 20200929

Appendix B. cMT Viewer Limitations

A cMT / cMT X series HMI screen accessed locally, by physically on the HMI screen itself, or remotely, from another device running cMT Viewer or VNC viewer. However, please note that some features do not work when connecting remotely from a cMT Viewer client (iOS, Android, PC, cMT-iV5/iV6), even if they work on the local cMT / cMT X machine. Please take this into consideration when designing projects.

Features that do not work remotely on cMT Viewer:

- File Browser
- Media Player
- Video In: USB Camera
- Video In: IP Camera (This feature is supported only in cMT Viewer that runs on Windows PC, and a decoder is required.)
- Video In: Video Input (analog)
- Picture View
- PDF Reader
- VNC Viewer
- Selected features of Function Key: Screen hardcopy, Import user data/User [USB Security Key]
- Selected features of PLC Control
  - i. Screen hardcopy: File are saved in the external drive on HMI
  - ii. Backlight: Backlight control of HMI screen
- Selected features of Operation Log
  - i. Operation Log printing: Logs are saved in the external drive on HMI

Appendix C. PC Simulation Limitations

Running on-line or off-line simulation when designing an EasyBuilder Pro project can help users find out if the project will work as planned. However, some features only works on HMI, in which case the project should be downloaded to HMI in order to see the effects. Please take this into consideration when designing projects.

When running simulation on PC, the historical data is stored in their respective destination folder that is placed under EasyBuilder Pro installation directory:

- HMI_memory: when the designated destination is HMI memory.
- SD_card: when the designated destination is SD card.
- usb1: when the designated destination is USB Disk.
- usb2: when the designated destination is USB Disk2. (Applicable only for models that support two USB disks.)

Features that do not work when running on-line/off-line simulation on PC:

cMT/cMT X Series

- Media Player
- Video In: USB Camera
- Video In: IP Camera
- Video In: Video Input (analog)
- PDF Reader
- VNC Viewer
- PLC Web Browser

eMT/iE/XE/mTV Series

- Video In: USB Camera
- Video In: IP Camera
- Video In: Video Input (analog)
- VNC Viewer

Appendix D. Notes on the Use of 64-bit Data Type

cMT / cMT X Series models offer 64-bit data type support. The following explains limitations on the use of 64-bit data type.

- Input ranges (Numeric object, Set Word object…etc)
  i. unit64: 0 ~ 2^48
  ii. int64: -2^48 ~ 2^48
  iii. double: -2^48 ~ 2^48

- int64 / unit64

Double is the internal data type used in computation, while it stores a mantissa with 52-bit precision. As a result , 15-digits integers and a portion of 16-digits integers may be stored exactly, but storing numbers exceeding that may result in lost precision.
In EasyBuilder Pro, inputs exceeding 48-bit may result in error; therefore, the input limit is defined as 48-bit.

- double

In EasyBuilder Pro, the upper/lower input limit was formerly determined by the max/min of double (+/- 1.79 * 10^308), which is nearly limitless in practicality. For consistency, the upper/lower input limit is also defined as -2^48 ~ 2^48.

Example:
The value 144,115,188,075,855,872 (a 64-bit unsigned int. value) read from PLC may be displayed as 144,115,188,075,855,870 by Numeric as precision is lost.

Other circumstances:
As in the example above, the same problem may occur when using Recipe Import / SQL Query / Database (Data Sampling), where data is from an external source.

Please note that for double, as a floating-point number, precision issue is inherent. It should not be assumed that all numbers in the range +2^48 ~ -2^48 is guaranteed to display exactly. Explanations above should apply to the use of int64 / uint64 only.