

# Silver Series



## INSTALLATION AND OPERATION MANUAL



1241 Bundy Boulevard  
Winona, Minnesota 55987 USA

Phone: (507) 454-5300  
Fax: (507) 452-4507  
Web: [www.watlow.com](http://www.watlow.com)



## **COPYRIGHT NOTICE**

This manual is a publication of Maple Systems, Inc., and is provided for use by its customers only. The contents of the manual are copyrighted by Maple Systems, Inc.; reproduction in whole or in part, for use other than in support of Maple Systems equipment, is prohibited without the specific written permission of Maple Systems.

The copyright of EasyBuilder-5000 and its related software belongs to Weintek Labs, Inc.

## **WARRANTY**

Maple Systems warrants each product to be free from electrical and mechanical defects in materials and workmanship for a period of two years from the date of shipment. This warranty does not apply to defects in the Products caused by abuse, misuse, accident, casualty, alteration, negligence, repair not authorized by Maple Systems, use on current or voltages other than specified by Maple Systems, or application or installation not in accordance with published instruction manuals. This warranty is in lieu of any other warranty either expressed or implied.

Maple Systems' liability is limited to the repair or replacement of the Product only, and not costs of installation, removal, or damage to user's property or other liabilities. If Maple Systems is unable to repair or replace a nonconforming Product, it may offer a refund of the amount paid to Maple Systems for such Product in full satisfaction of its warranty obligation. Maximum liability of Maple Systems is the cost of the Product.

Information furnished by Maple Systems, Inc., is believed to be accurate and reliable. However, no responsibility is assumed by Maple Systems for the use of this information nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication, or otherwise, under any patent or patent rights of Maple Systems, Inc. Maple Systems retains the right to revise or change its products and documentation at any time without notice.

## **IF SERVICE IS REQUIRED**

Contact Watlow Customer Service at (507) 454-5300 for instructions on how to return a product and how to obtain a Return Materials Authorization (RMA) number.

Package the unit in its original packaging container or, if unavailable, any suitable rigid container. If a substitute container is used, surround the unit with shock absorbing material; damage in shipment is not covered by the warranty. Include a letter with the unit describing the difficulty and designating a contact person.

All returns will be tested to verify customer claims of noncompliance with the product warranty. Improper return packaging, which makes verification impossible, will void the warranty. Products passing the tests will be returned "AS IS" to the customer.

If noncompliance is verified and is not due to customer abuse or the other exceptions described with product warranty, Maple Systems will, at its option, repair or replace the Product returned to it, freight prepaid, which fail to comply with the foregoing warranty, provided Maple Systems is notified of such noncompliance within the two-year warranty period.

## **APPLICATIONS ASSISTANCE**

This manual is designed to provide the necessary information for trouble-free installation and operation of your new Operator Interface Terminal (OIT). However, if you need assistance, please call Watlow at (507) 494-5656 between 7 a.m. and 5 p.m. Central Standard Time (CST). Ask for an Applications Engineer. Or you may e-mail your questions to [wintechsupport@watlow.com](mailto:wintechsupport@watlow.com).

# Table of Contents

EZware-5000 Support . . . . .	1	Creating a Sample Project . . . . .	36
OIT Models Supported . . . . .	1	Setting the System Parameters . . . . .	36
PLCs Supported . . . . .	1	Creating a Popup Window . . . . .	40
About Your Documentation. . . . .	1	Creating a Startup Window . . . . .	48
Conventions . . . . .	1	Finishing Up . . . . .	54
What You Need . . . . .	2	<b>Chapter 5 - Simulator Mode . . . . .</b>	<b>59</b>
OIT Basics . . . . .	2	The Simulation Screen . . . . .	59
What is a Silver Series OIT? . . . . .	4	<b>Chapter 6 - Using EZware-5000 . . . . .</b>	<b>61</b>
List of Features. . . . .	5	Overview . . . . .	60
<b>Chapter 1 - Installation of OITs . . . . .</b>	<b>7</b>	The Project Manager . . . . .	60
Unpacking the Unit . . . . .	7	Communications & Password Settings . . . . .	62
Managing Electrostatic Discharge . . . . .	7	Editor Settings . . . . .	63
CE Compliance . . . . .	7	Transfer Settings . . . . .	63
NEMA Rating . . . . .	7	Simulation Settings. . . . .	63
Environmental Considerations . . . . .	7	Viewing Data Log. . . . .	65
Safety Precautions . . . . .	8	Easy Printer . . . . .	70
Control Panel Design Guidelines. . . . .	8	Converting Data/Event Log. . . . .	70
Control Panel Grounding . . . . .	9	Recipe Extended Mem. Editor . . . . .	71
Connect OIT Chassis Ground to Control Panel. . . . .	9	Pass Through Mode . . . . .	76
Power Supply Selection. . . . .	11	The EasyConverter Application . . . . .	76
Cable Routing and Noise Immunity. . . . .	11	Managing Projects . . . . .	78
Installation. . . . .	12	Editing and Creating Screen Objects. . . . .	82
Connect the OIT to Power . . . . .	12	Basic Editing Commands . . . . .	86
Panel Preparation . . . . .	14	Grouping Objects . . . . .	90
Mount the OIT to the Panel. . . . .	14	Layering Objects . . . . .	91
Configuration Wiring . . . . .	15	Nudging Objects . . . . .	93
Connect the OIT to the PC for Configuration . . . . .	15	Making Objects Same Size . . . . .	97
Configuration . . . . .	15	Making Objects Same Color. . . . .	98
General Settings . . . . .	101	Automatically Reboot OIT. . . . .	112
Save and Compile the Project. . . . .	112	<b>Chapter 7 - Creating Windows . . . . .</b>	<b>116</b>
<b>Chapter 2 - OIT Local Setup. . . . .</b>	<b>17</b>	Window Fundamentals. . . . .	117
Factory Configuration. . . . .	17	Opening a Window . . . . .	117
Calibrating the Touchscreen . . . . .	17	Creating a New Window . . . . .	117
Getting into Local Setup . . . . .	17	Assigning Underlay Windows . . . . .	121
Changing the System Settings. . . . .	18	How to Display Underlay Windows . . . . .	122
Viewing System Information . . . . .	22	Rules That Apply to Underlay Windows . . . . .	122
Transferring Projects and Data from USB/CompactFlash . . . . .	23	Deleting a Window . . . . .	123
<b>Chapter 3 - Connect the OIT to the PLC or Controller . . . . .</b>	<b>25</b>	Using Base Windows . . . . .	123
COM Ports for the HMI5080/HMI5104/HMI5121 . . . . .	26	How to Display Base Windows . . . . .	123
COM Ports for the HMI5056 . . . . .	27	Using the Common Window. . . . .	129
Connecting to COM1 and COM3 on Port A . . . . .	28	Changing the Active Common Window. . . . .	132
Connecting to COM1 and COM2 on Port B . . . . .	32	Using the Fast Selection Window. . . . .	133
Connecting multiple PLC/controllers serially using daisy chaining . . . . .	33	Using the Fast Selection Key . . . . .	135
Connecting one or more PLC/controllers via Ethernet . . . . .	33	Changing screens using the Fast Selection window . . . . .	136
System Message Window . . . . .	138	<b>Chapter 8 - Creating Graphic Objects . . . . .</b>	<b>140</b>
<b>Chapter 4 - Creating Your First Project . . . . .</b>	<b>35</b>	Drawing Objects . . . . .	140
Before You Begin . . . . .	35	Using the Drawing Tools . . . . .	140
Connecting OIT to Computer . . . . .	35	Using Text . . . . .	144
Starting EZware-5000 . . . . .	35		

Predefined Shapes and Pictures. . . . .	146	Monitoring Alarms with the Alarm/Event	
Using a Predefined Shape . . . . .	146	Log . . . . .	263
Using a Predefined Picture . . . . .	148	Displaying Alarms/Alarm Display Object. . . . .	266
Graphics Libraries . . . . .	150	Displaying Alarms using the Alarm Bar	
What are 'states'? . . . . .	150	Object. . . . .	267
Using Shape Libraries . . . . .	150	Using Events . . . . .	269
Using Picture Libraries . . . . .	154	Monitoring Events With the Event Log	
Using Group Libraries . . . . .	157	Object. . . . .	269
Using Sound Libraries . . . . .	160	Displaying Events Using the Event	
		Display Object. . . . .	269
<b>Chapter 9 - Creating and Using Databases and</b>			
<b>Languages. . . . .</b>	<b>164</b>	<b>Chapter 14 - Data and Recipe Transfer Objects</b>	<b>272</b>
Creating and Using the Tag Library . . . . .	164	Using the Data Transfer Object. . . . .	271
Importing and Exporting the Tag		Using the Recipe Transfer (Data Transfer, Trig-	
Library. . . . .	165	ger Based) Object . . . . .	272
Using the Tag Library. . . . .	165	Creating a Recipe . . . . .	273
Creating the Label Library . . . . .	166	Backup Part. . . . .	274
Setting Different Fonts for Different Lan-			
guages . . . . .	168	<b>Chapter 15 - Macros . . . . .</b>	<b>277</b>
Importing and Exporting the Label		Using Macros . . . . .	277
Library. . . . .	169	Macro Sample and	
Using the Label Library. . . . .	169	Implementation . . . . .	277
Using Languages with the Label		Variables, Declarations and Memory	
Library. . . . .	170	Usage . . . . .	278
		Memory Usage: . . . . .	278
<b>Chapter 10 - Representing Data with Graphics</b>		Variable Declarations. . . . .	279
<b>Objects. . . . .</b>	<b>172</b>	Variable Initialization . . . . .	279
General Properties of Objects. . . . .	172	Array initialization . . . . .	280
Using Index Registers . . . . .	182	Reserved Words. . . . .	281
Using Internal Data Memory of OIT . . . . .	184	Operator. . . . .	281
Representing PLC Coil Registers . . . . .	192	Order of Precedence . . . . .	282
The Bit Lamp Object . . . . .	192	Main Functions and Sub-	
The Set Bit Object . . . . .	196	functions . . . . .	282
The Toggle Switch Object. . . . .	197	Local and Global Variables . . . . .	282
Representing PLC Data Registers. . . . .	199	Creating Variable Arrays . . . . .	283
The Word Lamp Object . . . . .	199	Using Macros Within Recipes . . . . .	283
The Set Word Object . . . . .	202	Statements, Conditions &	
The Slider Object. . . . .	203	Expressions . . . . .	284
The MultiState Switch Object. . . . .	205	Function Calls and Passing Parameters	
The Numeric Display Object. . . . .	209	246	
The Numeric Input Object . . . . .	211	Reading & Writing External Registers in	
The ASCII Display Object . . . . .	216	a Macro . . . . .	286
The ASCII Input Object. . . . .	217	Precautions, Tips & Tricks when Using	
The Moving Shape Object . . . . .	218	Macros . . . . .	288
The Animation Object . . . . .	222	Compiler Errors & Error Codes . . . . .	288
The XY Plot Object . . . . .	224		
<b>Chapter 11 - Using and Creating Keypads . . .</b>	<b>240</b>	<b>Chapter 16 - Miscellaneous Objects. . . . .</b>	<b>291</b>
How to Create a Keypad . . . . .	240	Misc. Objects . . . . .	291
Displaying and Using a Keypad . . . . .	241	Scheduler Object . . . . .	293
		Time Set Tab . . . . .	293
<b>Chapter 12 - Bar Graphs, Meters and Trends .</b>	<b>246</b>	Prohibit Tab. . . . .	295
Creating Bar Graphs . . . . .	246	Media Player . . . . .	296
Creating Meter Displays . . . . .	248		
Creating Trend Displays & Data Sampling		<b>Index . . . . .</b>	<b>303</b>
Objects . . . . .	251		
History Data Display Part . . . . .	258		
<b>Chapter 13 - Capturing Alarms and Events . . .</b>	<b>264</b>		
Using Alarms. . . . .	264		

## Important Information Regarding Watlow Part Numbers

When you purchased your Operator Interface Terminal and accessories from Watlow, you ordered and received them under Watlow part numbers. The parts are also labeled with Maple Systems' part numbers and are referred to by those part numbers throughout this manual and other support documents. The table below correlates the Watlow and Maple Systems part numbers.

Watlow P/N	Maple P/N	Description
TS00-0070-0000	HMI5070TH	7.0" (800 x 480) 65,536 (16-bit) color TFT LCD and an analog-resistive touchscreen.
TS00-0100-0000	HMI5100T	10.0" (800 x 480) 65,536 (16-bit) color TFT LCD and an analog-resistive touchscreen.
0830-0750-0000	4010-0005	25W power supply (for the TS00-0070-0000 only) Input: 85-264 VAC, Output: 24VDC, 1.1A, CE/UL rated
0830-0751-0000	4010-0006	60W power supply Input: 85-264 VAC, Output: 24VDC, 2.5A, CE/UL rated
0219-0374-0000	7446-0185-5	Com. cable, 5-foot, OIT COM1 [2-wire 485] to bare wires for Watlow EZ-ZONE® controller screw terminals
0219-0375-0000	7431-0110	Splitter adapter – for connecting to both COM1 [RS-232] and COM2 [RS-232]
0219-0376-0000	7431-0111	Splitter adapter – for connecting to both COM1 [RS-485, 2-wire] and COM3 [RS-485, 2-wire]
0219-0377-0000	7431-0112	Splitter adapter – for connecting to both COM1 [RS-485, 4-wire] and COM3 [RS-485, 2-wire]
0219-0378-0000	7431-0113	Splitter adapter – for connecting to both COM1 [RS-485, 4-wire] and COM3 [RS-232]
0219-0379-0000	7431-0114	Splitter adapter – for connecting to both COM1 [RS-485, 2-wire] and COM3 [RS-232]
0210-0382-0000	7431-0115	USB cable, 5 feet, PC to OIT for downloading projects
0830-0753-0000	2203-0002	Package of 5 ea. protective screen covers for the TS00-0070-0000
0830-0754-0000	2203-1001	Package of 3 ea. protective screen covers for the TS00-0100-0000

---

## Introduction - Welcome

Welcome to the Maple Systems' Silver Series Operator Interface Terminals (OITs) from Watlow. Using graphic OITs has never been easier. This powerful family of graphics operator interface terminals connect to Watlow controllers and other devices such as programmable logic controllers (PLCs) to provide the human-machine interface in industrial applications. The Silver Series has several features not found in other graphic OITs. This manual explains the operation of the Silver Series OITs and how to implement the many available features using the EZware-5000 Configuration Software.

Watlow and Maple Systems have worked together to bring you this product. Please contact Watlow for support, and be sure to utilize any resources referenced in this manual on the Watlow website ([www.watlow.com](http://www.watlow.com)) and the Maple Systems Website ([www.maplesystems.com](http://www.maplesystems.com)).

### EZware-5000 Support

#### OIT Models Supported

For the latest list of Silver Series Models supported by Watlow, please visit the Watlow website at [www.watlow.com](http://www.watlow.com).

#### PLCs Supported

For the latest list of PLCs and controllers supported by the Silver Series OITs, please visit our [website](#).

### About Your Documentation

Maple Systems provides many resources to allow you to get the most out of your Silver Series OIT.

*Silver Series Operation Manual* (shipped with EZware-5000 as a PDF file) - describes installation, general operation and features of the Silver Series using EZware-5000 configuration software..

*Watlow Silver Operator Interface Terminal Manual Addendum* - contains specific instructions and examples on communicating with and creating interfaces for Watlow controllers.

*Controller Information Sheets* - important information specific to each supported protocol.

*EZware-5000 On-line Help* - covers the operation of EZware-5000. Always available by clicking **HelpTopics** from the **Help** menu in EasyBuilder, or press **F1** from any dialog.

For more information about these and other training sources, visit the Maple Systems [website](#).

### Conventions

When using EZware-5000, there are usually several ways to perform a task. For example, if you want to copy a graphics object, you can:

Click the **Copy** command on the Edit menu.




Click the **Copy** button on the Standard toolbar.

Press the CTRL + C keys on your computer.

In most cases, we will describe each method when the task is first discussed. The menu method is then used whenever the task is used in later procedures. Other conventions used in this manual are listed in the following table.

Convention	Meaning
<b>Bold</b>	Characters that you must type exactly as they appear. For example, if you are directed to type <b>a:\setup</b> , you should type all the bold characters exactly as they are printed.
<i>Italic</i>	Placeholders for information you must provide. For example, if you are directed to type <i>filename</i> , you should type the actual name for a file instead of the word shown in italic type. Italics are also used to indicate a glossary term.
ALL CAPITALS	Directory names, file names, key names, and acronyms
KEY1+KEY2	A plus sign (+) between key names means to press and hold down the first key while you press the second key.
click	Refers to clicking the primary mouse button (usually the left mouse button) once.
Double-click	Refers to quickly clicking the primary mouse button (usually the left mouse button) twice.
Right-click	Refers to clicking the secondary mouse button (usually the right mouse button) once. Right-clicking usually opens shortcut menus.

The following table identifies symbols and margin icons.

Icon	Meaning
	Identifies a procedure.
	Indicates a reference to additional information.
	Indicates an important note.

## What You Need

The following items are needed to configure and operate your OIT.

Configuration Software	EZware -5000
Configuration Cable <sup>1</sup>	7431-0104 ( Ethernet Crossover cable for HMI5056T, HMI5080T, HMI5104T/XH, HMI5121T/X, HMI5150X models) 7431-0115 (USB download cable for HMI5043N, HMI 5056N, HMI5070NH/TH and HMI5100N/T models)
Personal Computer <sup>2</sup>	User Provided
Power Cable	6030-0009 2-conductor 18 AWG, shielded, no connectors
24VDC Power Supply	User Provided
PLC	User Provided
Controller Information Sheet	Maple Systems provides Controller Information Sheets which contain important information specific to each PLC. <b>Please locate the sheet that corresponds to your PLC on our website.</b>
Communication Cable (HMI to PLC)	Refer to our web site ( <a href="http://www.maple-systems.com">www.maple-systems.com</a> ) for a list of available cables.

<sup>1</sup>Allows you to connect OIT directly to PC Ethernet port or to USB port to download/upload projects into the OIT.

<sup>2</sup>Computer requirements include a Pentium 1.7 GHz or higher processor, 512 MB RAM, 200 MB available Hard Disk Space, SVGA or higher resolution monitor, Keyboard and Mouse, available Ethernet port or USB port (for downloading project), at least one RS232 port for online simulation, Windows XP, 2000 or Vista.

## OIT Basics

*Operator Interface Terminals* (OITs) provide much more versatility than traditional mechanical control panels. An OIT allows a plant floor operator to monitor current conditions of a control system and, if necessary, to initiate a change in the operation of the system. OITs connect to programmable logic controllers (PLCs) typically through the PLC's serial communications port. The OIT can be programmed to monitor and/or change current values stored in the data memory of the PLC.

OITs can have either text-based or graphics-based displays. A text-based OIT can display printable text characters but no graphics. Some text-based OITs can display text characters in various sizes. A graphics-based OIT can display printable text characters of varying fonts and sizes and graphics shapes such as icons, bitmaps, or pictures. Using pictures instead of words or characters often greatly simplifies the operation of the OIT, making the OIT much more intuitive to use.

Some OITs use touchscreen displays while others use a membrane-style keypad. Membrane-style keyboards are best used in applications in which the keypad is likely to become dirty. Touchscreen displays are placed over the OIT screen thus providing much more flexibility than typical membrane-style keypads. Because of this, switches can be created on a touchscreen that appear only when needed.

The Maple Systems Silver Series OITs are graphics-based touchscreen OITs. Before we get any further into the operation of these OITs, it is necessary to define some terms that will be used throughout this manual.

### **Projects**

The OIT has two basic segments of internal memory. The *code memory* contains the information required by the OIT that controls how it operates such as the features supported and how it communicates to a PLC. The OIT programmer does not have the ability to change code memory. The *project memory* pertains to all of the window screens created and any other features that the OIT programmer can create using the EZware-5000 configuration software. Therefore, the term *project* is used to designate the file that is sent to the OIT from the EZware-5000 software.

### **Objects**

An *object* is any action that the OIT performs while it is communicating to the PLC. In order to get the operator interface terminal to 'do anything', you must program the OIT with objects. Objects perform actions such as display text or graphics, write a value to a PLC register, or display an alarm. Objects most often are graphics shapes that are to be displayed on the OIT screen. For example, a *Text Object* is used to display text on the OIT. But objects are also used to configure the OIT to perform some action. For example, a *PLC Control Object* tells the OIT to continuously monitor a PLC register that is used by the PLC to request a new window. Some objects can display a graphics shape on the OIT screen and perform some action. For example, a *Toggle Switch Object* creates a graphic object on the OIT that when pressed, activates a bit in the PLC.

### **Graphics Object**

A *graphics object* is any text, icon, or picture that can be displayed on the OIT. Graphics objects are further defined by how they are composed or created. A *Text Object* is a graphics object that displays text on the OIT screen. A *Bitmap Object* is a graphics object that displays a bitmap on the OIT screen. Bitmaps are files stored in the OIT to display pictures. A *Shape Object* is a graphics object that displays a shape on the OIT screen. Shapes are also files stored in the OIT to display pictures. Shapes differ from bitmaps in that shapes are stored using a vector-based file format whereas bitmaps use a pixel-based file format. Each format has its advantages and disadvantages. We will not go into any more detail about bitmap objects and shape objects until later in this manual. For now, think of them as objects used to display pictures on the OIT screen. Finally, a *Group Object* is the most complex type of graphics object. It is a combination of other objects. Briefly, a group object consists of one or more objects that are 'grouped' together and stored as one object. A good example is a keypad, which is really a combination of several keys each designed to perform a specific task. When grouped together, a keypad can be stored as a group object for use in other projects or windows.

### **Windows**

A *window* is a screen that can be displayed on the OIT. Windows can be full-sized to completely cover the OIT display or partially sized. Any partially sized window is usually referred to as a *popup window*. Windows can appear on the OIT display by a request from the PLC or by a press from the touchscreen. Windows can be configured to any size. Once a window is displayed, it can be moved around the OIT display, removed from the display, or minimized to an icon. Windows can even overlap each other. Each window can display graphics objects and there is no limit to the number of graphics objects that can be placed on each window. The Silver Series is capable of storing up to 1999 windows, but the actual limit is determined by the total amount of memory used for the application. A more in-depth discussion of windows is covered in later chapters. For now, think of windows as screens that can be displayed on the OIT.



## What is a Silver Series OIT?

The Silver Series OITs by Maple Systems are graphics operator interfaces designed to connect to PLCs in an industrial environment. The displays are covered with analog resistive touchscreens designed for harsh industrial environments. The touchscreens use the latest in touchscreen technology enabling the OIT programmer to create switches that are very fine in resolution. Unlike many other touchscreen OITs on the market, the Silver Series OITs are not limited to a fixed number of cells in which switches can be created. The OIT programmer can create as many switches of varying sizes and shapes as he wishes, limited only by the total amount of memory available for the project.

Three LED indicators are provided on the face of the Silver Series OITs to provide instant feedback to the OIT operator of the current operating condition of the OIT.










LED Indicator	Purpose
PWR LED (yellow)	indicates if power is applied to the OIT
CPU LED (green)	indicates if the OIT is operating correctly
COM LED (red)	indicates communications activity on PLC port






















The Silver Series OITs have three serial ports, which provide a connection to a PLC using RS-232 or RS-485 communications and an Ethernet connection for PLC communication and project upload and download. The serial ports also provide the ability to use the EZware-5000 configuration software in Simulation Mode enabling the OIT programmer to test his project on the PC instead of downloading the project to the OIT.







The Silver Series is powered using +24VDC. Finally, a reset switch is provided on the back of the OIT to reinitialize the OIT if an operational failure occurs. The Silver Series is designed for industrial environments and carries a NEMA 4X (indoor only) rating as well as CE compliance for noise immunity and emissions.

## List of Features

The next chapter will guide you through the creation of your first project. Before you proceed, you may wish to read this brief list of some of the features offered in the Silver Series OIT.

Icon	Name	Description
	Bit Lamp	Creates a graphics object to reflect the current status of a PLC bit.
	Word Lamp	Creates a graphics object to reflect the current state of a multi-state PLC data register.
	Set Bit	Creates a touchscreen graphics object that represents a two-state switch. When pressed it sets/resets a PLC bit.
	Set Word	Creates a touchscreen graphics object that represents a multi-state switch. When pressed it can place a constant value in a PLC register or jog the value.
	Toggle Switch	Creates a touchscreen graphics object that represents a two-state switch changing state (picture) based upon a PLC bit. When pressed, it can control another PLC bit.
	Slider Object	Creates a touchscreen graphics object that changes the state according to the position of a slider switch.
	Multi-State Switch	Creates a multi-state touchscreen graphics object that changes state (picture) according to the value in a PLC data register. When pressed, it sends a value(s) to another PLC register.
	Function Key	Creates a touchscreen graphics object or runs a macro.
	Numeric Input	Displays a number stored in a PLC register. The number can be changed using a numeric keypad.

	Numeric Data	Displays a number stored in a PLC register.
	ASCII Input	Displays ASCII characters stored in a PLC register. Characters can be changed using an alphanumeric keypad.
	ASCII Data	Displays ASCII characters stored in a PLC register.
	Moving Shapes	Creates a multi-state graphics object, which changes state (picture) and position on the screen according to a value in a PLC register.
	Animation	Creates a multi-state graphics object, which changes state (picture) on the screen according to a value in a PLC register. The positions on the screen are predefined.
	Indirect Window	Configures the OIT to monitor PLC data registers to display for a specific window popup by a PLC word address
	Direct Window	Displays a Window based on a bit in a PLC register.
	Alarm Displays	Creates alarms to display alarms sent from the Alarm/Event Log Object.
	Alarm Bar	Displays alarms detected by the Alarm/Event Log Object on a single horizontal scrolling line.
	Data Transfer (triggered)	Transfers data to the specified PLC registers based on a touch or PLC bit status.
	Data Transferred (time-based)	Transfers data to the specified PLC registers on a timed basis.
	Event Displays	Displays messages detected in the Alarm/Event Log and can acknowledge the message(s).
	Data Sampling (Data Logging)	Stores data for trending and archiving.
	Alarm/Event Log	Contains the data for detecting alarm conditions.
	Trend Displays	Creates a trend graph. Samples data in a single or multiple 16-bit PLC register and plots the data on a time graph.
	Bar Graph Displays	Creates a bar graph with alarm monitoring.
	Meter Displays	Creates a scale meter.
	Recipe Transfer	Transfers data to the specified PLC registers.
	System Message	Customizes the content of system-generated messages.
	PLC Control	Configures the OIT to monitor PLC data registers to display full window screens.
	History Data Display	Displays historical data in a tabular format.

	Backup	Copies recipe, event or historical data to one of the USB memory devices.
	Data Block Display	Displays the data in a series of registers as a line graph.
	XP Plot	Creates an XY Plot.
	Media Player	Displays a video file.
	Option List	Creates a drop-down list box
	Scheduler	Performs an action based upon a time schedule.

# Chapter 1 - Installation of OITs

## Before You Begin

Please read the following for proper handling of your new OIT.

### ***Unpacking the Unit***

Carefully unpack the OIT. Please read any instructions or cautions that appear on the shipping container. Check all material in the container against the enclosed packing list. Maple Systems, Inc. will not accept responsibility for shortages against the packing list unless notified within 30 days. The equipment and its accessories were inspected and tested by Maple Systems before shipment; all of the equipment should be in good working order. Examine the equipment carefully; if any shipping damage is evident, notify the carrier immediately. You are responsible for claim negotiations with the carrier. Save the shipping container and packing material in case the equipment needs to be stored, returned to Maple Systems, or transported for any reason.

### ***Managing Electrostatic Discharge***

It is best NOT to remove the rear enclosure of the OIT. When the rear part of the enclosure is removed, the circuitry inside is exposed to possible damage by electrostatic discharge during handling. Minimize the possibility of electrostatic discharge by:

- Discharging personal static by grounding yourself prior to handling the OIT
- Handling the OIT at a static-free grounded workstation
- Connecting the frame ground (FG) connector of the OIT to a clean earth ground
- Placing the OIT in an anti-static bag during transport

### ***CE Compliance***

The Silver Series OITs have been tested to conform to European CE requirements per Council Directive 89/336/EEC. The European Union created these requirements to ensure conformity among products traded in those countries. Specifically, the Silver Series OITs meet or exceed the noise emissions and immunity requirements as set forth in the EN50081 (Emissions) and EN50082 (Immunity) standards. These products are designed to withstand electrical noise in harsh industrial environments. They also conform to requirements that limit electrical emissions. However, this does not guarantee that the products will be totally immune from possible malfunction in cases where severe electrical noise occurs. Therefore, we strongly recommend that you follow the guidelines outlined in this chapter for proper wire routing and grounding to insure the proper operation of the Silver Series OIT.

### ***NEMA Rating***

The Silver Series OITs are rated for NEMA 4X/12 (indoor only) or IP65 installations. This means that when the OIT is properly mounted to a panel or other enclosure, the front enclosure of the OIT will provide protection to the inside of the panel from splashing water, wind blown dust, rain, or hose-directed water. The OIT must be installed according to the instructions in this chapter to be properly sealed.

### ***Environmental Considerations***

The Silver Series is designed to operate in temperatures from 0-45° C. It is intended for indoor installations and not designed for outdoor applications. Avoid installing the Silver Series in environments with severe mechanical vibration or shocks. Do not install the OIT in enclosures with rapid temperature variations or high humidity. Either will cause condensation of water inside the device and eventual damage to the OIT.

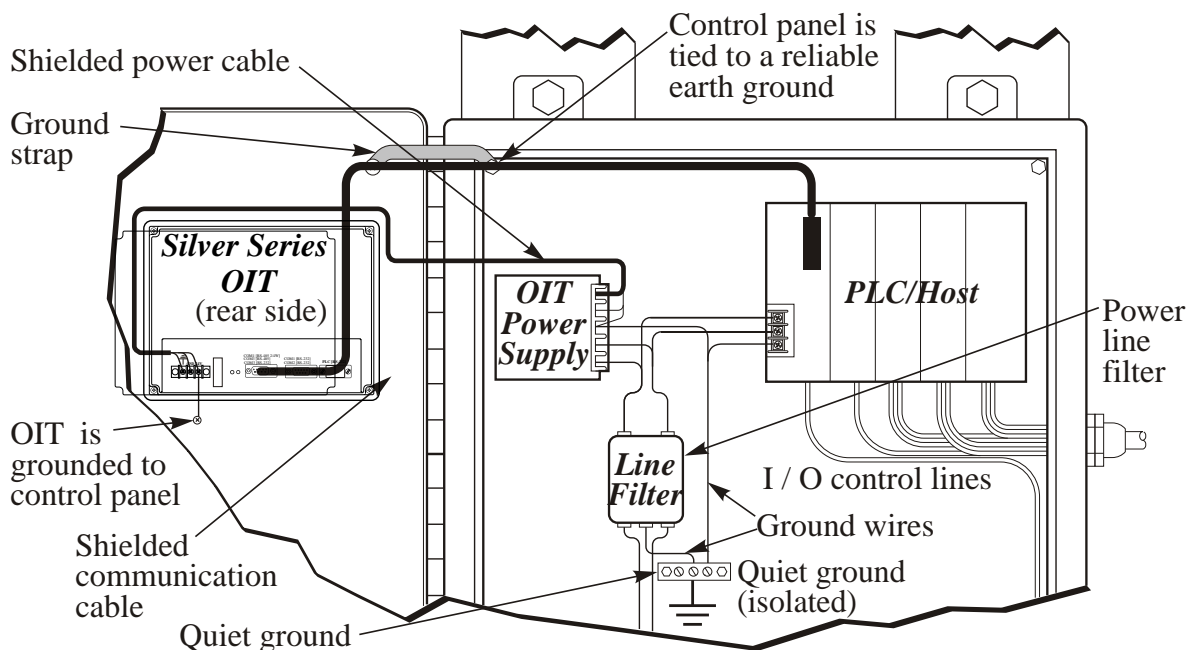
## Safety Precautions

Please observe the following precautions when installing the Silver Series OIT. Failure to comply with these restrictions could result in loss of life, serious personal injury, or equipment damage.

	<p><b>Warning:</b> Do not operate the OIT in areas subject to explosion due to flammable gases, vapors, or dusts.</p>
	<p><b>Warning:</b> Do not connect the OIT to an AC power source. You will cause permanent damage to the OIT.</p>
	<p><b>Warning:</b> Do not attempt to use a DC power supply that does not meet OIT power requirements. You may cause malfunction or permanent damage to the OIT.</p>
	<p><b>Warning:</b> Do not power the OIT with a DC power supply used for inductive loads or for input circuitry to the programmable logic controller. Severe voltage spikes caused by these devices may damage the OIT.</p>

## Control Panel Design Guidelines

Pay careful attention to the placement of system components and associated cable routing. These items can significantly enhance the performance and integrity of your control application.



Control Panel Example

## Control Panel Grounding

The control panel should be connected to a good, high-integrity earth ground both for safety considerations and shielding purposes. Maple Systems cannot overemphasize the importance of good grounding. If you fail to use good grounding procedures during installation, sporadic malfunction of the OIT may occur:

Connect the OIT's chassis ground terminal to a reliable earth ground with a low-resistance path.

Route all earth ground wires that lead from the OIT, the PLC, the power supply, and the line filter to a central earth ground point such as a barrier strip. This will ensure that no ground current from one device influences the operation of the other devices.

Connect the OIT chassis ground terminal to the control panel door using a heavy-gauge short braided cable or ground wire to minimize resistance.

Connect the power cable's shield wire to the OIT's chassis ground terminal.

Connect the control panel to earth ground using a copper grounding rod close to the OIT and control panel.

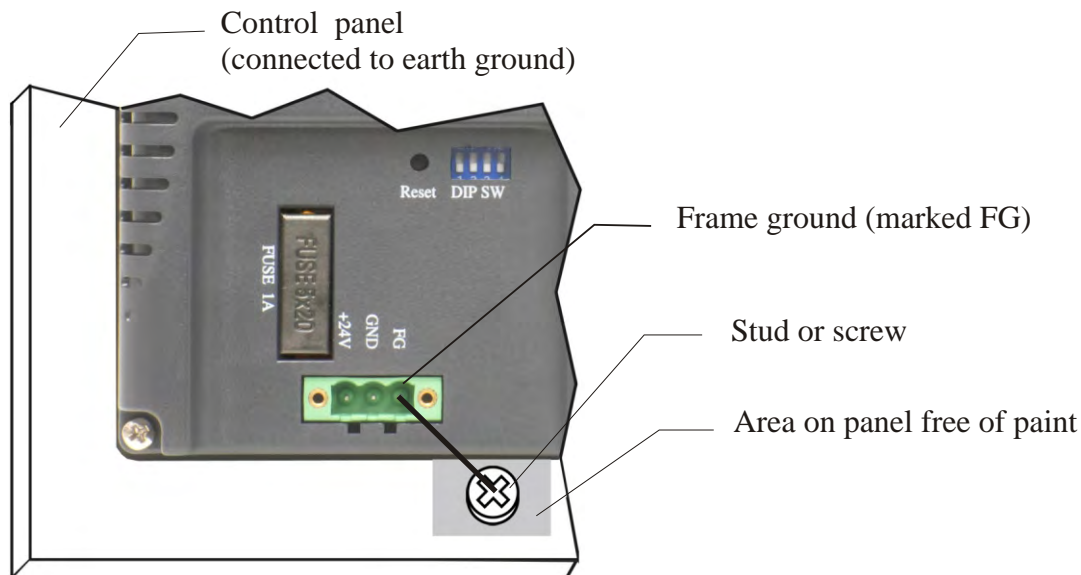
Hinged doors on control panels do not provide a long-term electrical connection to the rest of the enclosure. Corrosion develops over time and prevents good electrical contact. For this reason, a separate wire braid should be installed from the hinged control panel to the rest of the enclosure.

For a more in-depth overview of ground wiring techniques, refer to technical note #1027, "OIT Ground Wiring and Electrical Noise Reduction," which you can find on our web site.

### Connect OIT Chassis Ground to Control Panel

To reduce the possibility of electrical interference, connect the chassis ground terminal of the OIT to a clean earth ground. If the control panel is metal, make sure it is properly grounded. Then connect a *short* heavy-gauge wire (#18 AWG) from the chassis ground terminal of the OIT to a mounting bolt on the control panel door. The mounting bolt must have good electrical contact to the control panel; scrape away any paint that may be covering the panel to provide a good connection.

If the control panel is made of a non-conductive material, it is essential that you connect the chassis ground terminal of the OIT to a clean earth ground point located close to the panel.



Typical Chassis Connection

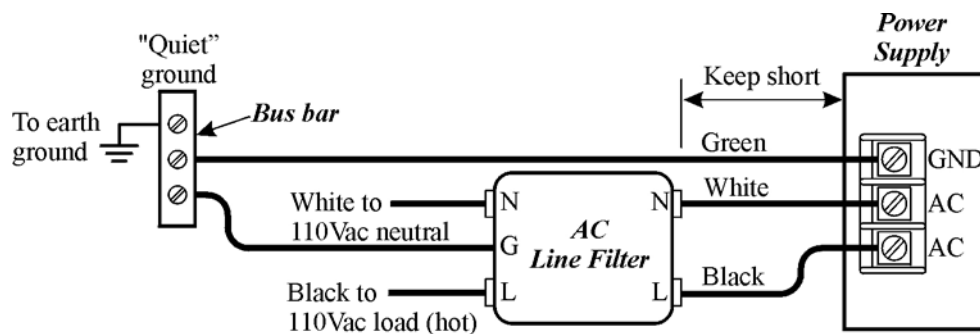
## Power Supply Selection

The power supply used to power the OIT should provide an output of +24 VDC 5% measured at the OIT power terminal block. A 24VDC regulated power supply dedicated to the OIT is required. Consult the datasheet of your particular OIT for current requirements.

The power cable for the OIT should be 18AWG, 2-conductor wire with a shield drain wire and protective shield (foil or braid). The shield drain wire must be connected to earth ground at both ends of the cable. Please refer to the "Connect the OIT to Power" section for more information.

A power line filter installed at the AC input to the OIT power supply is highly recommended as a safeguard against conducted RF noise, which is often present on factory power lines. The wires connecting the output of the power line filter to the power supply should be kept as short as possible to minimize any additional noise pickup. The case of the power line filter should be connected to a quiet earth ground. The power line filter should have a current rating of at least 3 Amps with common mode and differential mode attenuation.

Do not use the power supply used to provide power to the OIT to power switching relays, solenoids, or other active devices.



**Power Line Filter Connection**

## Cable Routing and Noise Immunity

Follow these guidelines when routing cable to the OIT:

Always route the OIT communication cable and the power cable away from any AC voltage or rapidly switching DC control lines.

Never bundle the OIT cables together with 120VAC power wires or with relay wiring.

Try to keep at least 8 inches (20 cm) of separation between the OIT cables and other power wiring. If voltages greater than 120VAC are used in the system, greater separation is required.

If the OIT cables must come near AC wiring, make sure they cross at 90 degrees.

Run AC power wires in a separate grounded conduit to reduce electrical noise interference.

Keep the cable lengths for the OIT as short as possible. Do not coil excess cable and place it next to AC powered equipment.

Cover any equipment used in the enclosure that operates at high frequency or high current levels with a grounded metal shield.

## Installation

It is necessary to follow all installation procedures described in this chapter for electrical noise immunity and CE compliance.

Your Maple Systems OIT is designed to connect easily to your PLC. External rear connectors provide quick connections for power, communications and programming wiring.

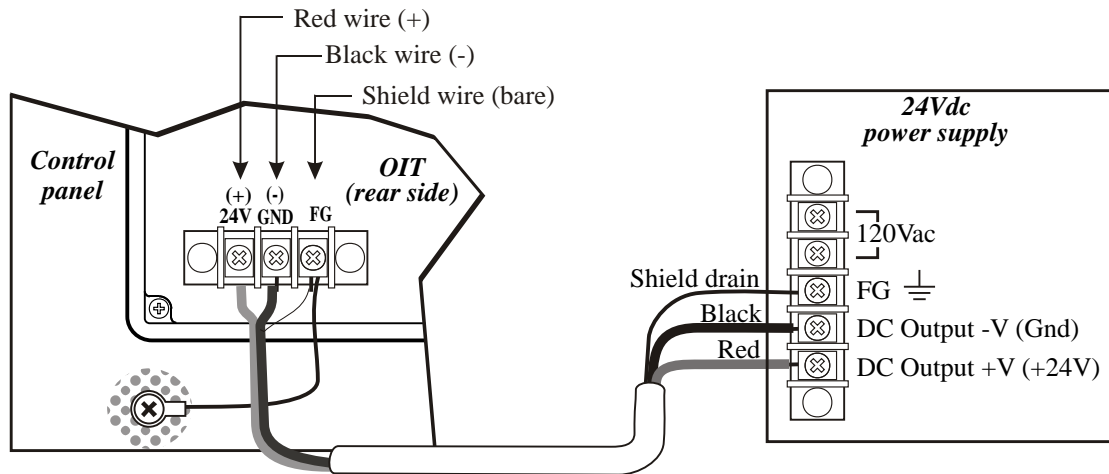
There are two connectors for serial communications on the rear of the OIT. Both of these are D-subminiature 9-pin connectors.

Use the supplied separate 3-position terminal block to provide power to the OIT.

### Connect the OIT to Power

The power cable for the OIT should be 18AWG, 2-conductor wire with a shield drain wire and protective shield foil. You may buy cable P/N 6030-0009 by the foot from Maple Systems to make these.

Always run the DC ground wire directly back to the signal return of the power supply. *Do not use the chassis ground wire as your signal return.*



### OIT Power Inputs

#### ► To connect the OIT to power:

1. Connect the power cable to the OIT
  - a. Strip the power cable shield to expose 2" of the black and red wires.
  - b. Strip about ¼" of insulation from the black and red wires.
  - c. Thread the black and red wires through the ferrite core. The shield wire must be outside.
  - d. Connect the red wire to the DC positive (+) input of the OIT power terminal.
  - e. Connect the black wire to the DC negative (-) input of the OIT power terminal.
  - f. Connect the power cable shield wire to the OIT power terminal's chassis ground input.
2. Route the power cable to the OIT power supply. The power cable should not be any longer than necessary.
3. Install the power supply wires as follows (with colors shown for Maple Systems cable P/N 6030-0009):

Color	Power Supply	OIT
Red	+Output/+24 Vdc	+24V
Black	-Output/+24 Vdc return	GND
Shield	Case ground	FG





*The power connector on the Silver Series is a terminal block with wire clamps. Lugs are not required.*



**Typical Power Wiring**

### **Panel Preparation**

A metal panel or mounting surface with a minimum thickness of 15 gauge (0.059 inch/3.3mm) if cold-rolled steel or hardened steel, or 10 gauge (0.101 inch/2.6mm) if aluminum alloy (6061-T6 preferred) is required. Thinner panels or surfaces may bow between the mounting clamps and not form a seal with the gasket.

The area of the panel or mounting surface where the gasket comes into contact must be flat and free of scratches, pits, and other features that prevent the gasket from sealing properly. If the panel or mounting surface is not uniform, thick, flat, stiff, or smooth enough, then a sealant such as silicone may be required.

---

**WARNING:** *The OIT requires a stiff, flat, smooth mounting surface free of blemishes to seal properly to NEMA 4.*

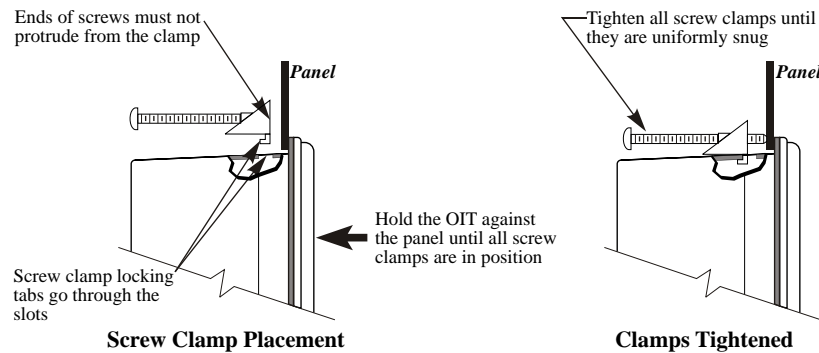
---



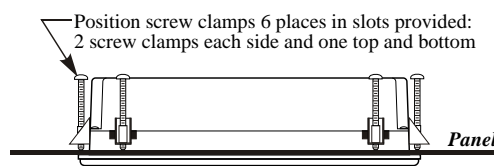
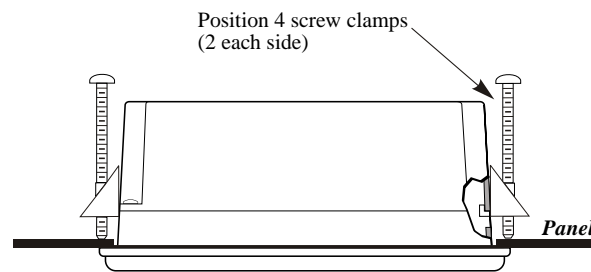
*Clean and deburr the panel cutout before the OIT is installed.*

---

## Mount the OIT to the Panel



## Installing Screws on the OIT



### STEPS:

1. Prepare the screw clamps for the OIT by positioning the metal brackets at the midpoints of the screws. Position the screws so that the ends don't protrude from the plastic portions.

2. Set the OIT in the panel cutout and hold it in place until all clamps are in position.
3. Tighten the screw clamps until all are uniformly snug.

**CAUTION:** Do not over-tighten the screws beyond snugness, or you may damage the housing, or cause the touchscreen to malfunction.

*REINSTALLATION:* Because the gasket will take a “set” to the panel, be sure to reinstall the OIT to the same panel cutout when a NEMA 4 seal is required. For best results, also replace the gasket itself.

## Configuration Wiring

The OIT must be configured for a particular protocol before use. The *EZware* software is used for configuring the OIT. For detailed instructions on installing and using the software, please refer to the software documentation section of this manual.

### ***Connect the OIT to the PC for Configuration***

To configure the OIT using Maple System's configuration software, you will need either an Ethernet crossover cable, Maple P/N 7431-0104 or a USB download cable, Maple P/N 7431-0115 depending upon the OIT model used. If using a switch or router between the PC and the OIT, use a straight-through or crossover cable as required by the switch or router.



*If an Ethernet port is not available on the PC, you may be able to install a USB Ethernet adapter. You can also transfer the project to the OIT with a Compact Flash or USB memory device. Refer to Chapter 2, OIT Local Setup.*

---

**Please Use This Page For Your Notes:**

## Chapter 2 - OIT Local Setup

### Factory Configuration

Each OIT arrives from the factory with a demo project file that illustrates some of the most popular features of the OIT. Please follow the directions enclosed in *Chapter 4, Creating Your First Project*, to configure your OIT for the PLC that you are using.

The OIT also has a black reset push button and a four position DIP switch located through an access hole on the back of the enclosure. The reset switch can be used to reinitialize the OIT if the OIT malfunctions. Only Dip switch 1 has functionality. Dip switch 1 puts the OIT into Touch Calibration mode. For normal operation, all of the DIP switches should be set to the OFF position.

This chapter describes how to use the Silver Series' local setup options. The local setup allows touchscreen calibration, transferring projects to and from the OIT, configuring the OIT's IP settings, passwords, time and date, screen contrast, as well as managing the storage of recipe and history files. Additionally, memory and other system information can be viewed.

### Calibrating the Touchscreen

The touchscreen of the OIT is fully calibrated before it leaves the factory so you shouldn't need to adjust it. However, with time the touchscreen may need to be recalibrated.

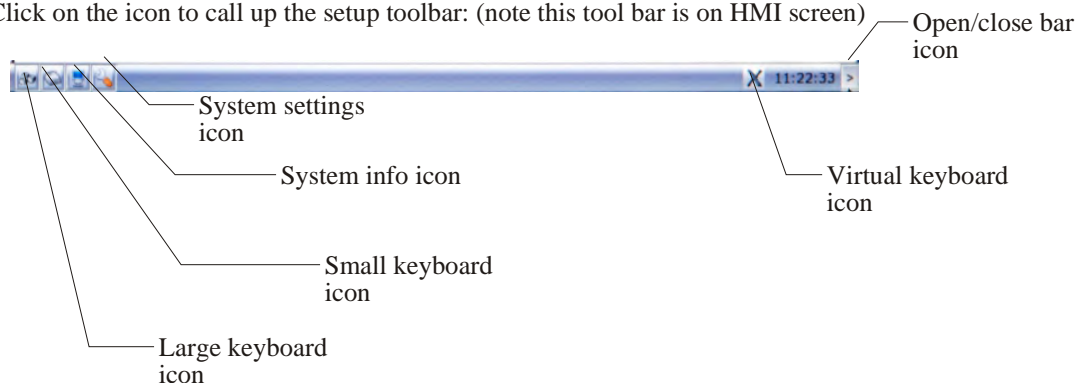
#### ► To calibrate the touchscreen:

1. Turn Dip Switch 1 ON.
2. Cycle power to the OIT.
3. After the OIT displays a crosshair cursor, you are prompted to touch the cursor for for:
  - a. Top left position
  - b. Top right position
  - c. Bottom right position
  - d. Bottom left position
  - e. Center position.
4. The OIT will then resume initializing.

### Getting into Local Setup

Accessing the local setup menus requires a USB-mouse be connected to the OIT. Using the mouse, point to the lower right-hand corner of the OIT. The open bar icon will appear.

Click on the icon to call up the setup toolbar: (note this tool bar is on HMI screen)



*Please Note: the above tool bar is located on the HMI.*

## Changing the System Settings

### ► To change the system settings on the OIT:

1. Open the Setup Toolbar as described above, and click the **System Settings** icon.
2. A dialog will be displayed requesting the *Local* password. The default password is 111111.



*You may have to move windows around a bit to gain access to the virtual keyboard. You may also have to click inside the password field again.*

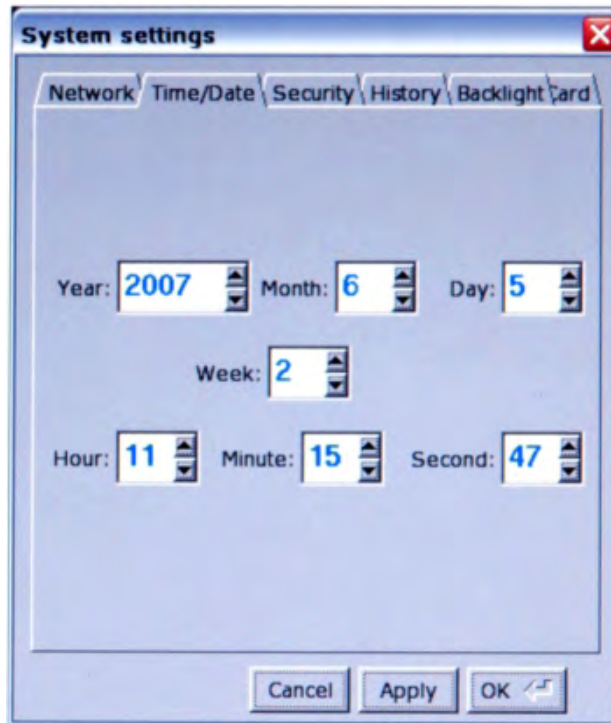
---

3. Once the correct password has been entered, the System Settings dialog is displayed. For all tabs, the Cancel, Apply, and OK buttons are available.

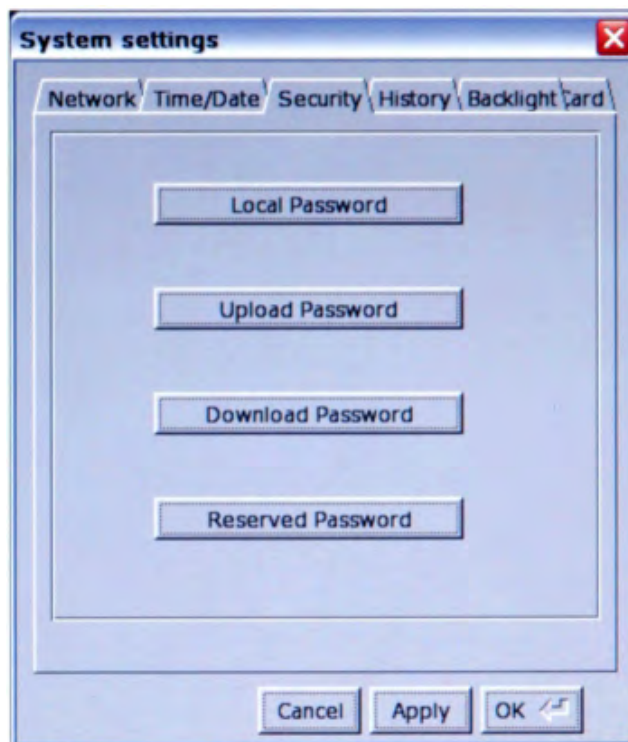
Cancel	Close the System Settings dialog without saving any changes.
Apply	Save the current settings without closing the System Settings dialog.
OK	Save the current settings and close the System Settings dialog.

4. Click on the **Network** tab to configure network settings. The Network dialog is displayed.
5. The Network tab allows configuration of the IP settings. The default setting is *Auto Get IP Address*. Use this setting if the OIT will provide an IP address by a DHCP server. If connected to a DHCP server, this dialog will display the IP settings obtained by the OIT. The IP fields are disabled, and are not editable.
6. If using a static IP address, use the *IP Address Get From Below* option. The IP fields are enabled. Enter the *appropriate settings* for your network.

- Click the **Time/Date** tab to configure time/date settings. The time/date dialog is displayed.



- Configure appropriate *time* and *date* settings.
- Click on the **Security** tab to display the security settings dialog. Here, you can select your system passwords.





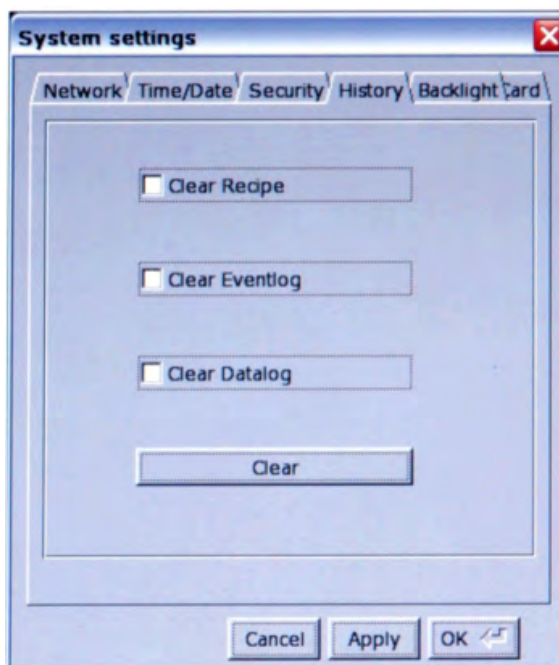
10. Configure your password settings.

Local Password	The password required to enter local setup.
Upload Password	The password required to <i>upload</i> data from the OIT to a PC or memory module.
Download Password	The password required to <i>download</i> data to the OIT from a PC or memory module.
Upload (History) Password	The password required to <i>upload</i> history from the OIT to a PC or memory module.

11. The new password must be entered, and then entered again to confirm. As the password is entered into the confirm field, an indicator will show if the two passwords match.



12. Click on the **History** tab to display the History dialog. This is where data stored in the OIT can be cleared.



---

 This will not clear data stored on a Compact Flash or USB device.

---

Clear Recipe	The OIT's stored <i>recipe</i> data will be cleared when the Clear button is clicked.
Clear Eventlog	The OIT's stored <i>event log</i> data will be cleared when the Clear button is clicked.
Clear Datalog	The OIT's stored <i>data log</i> data will be cleared when the Clear button is clicked.

13. Click on the **Backlight** tab to display the Backlight dialog. The Backlight tab adjusts the brightness of the OIT's backlight. Use the mouse or touchscreen to roll the wheel. Rolling to the left decreases the brightness, rolling to the right increases the brightness.




---

 System tags *Backlight Up*, *Backlight Down*, and *Backlight Index* can be used to adjust the brightness at runtime.

---

14. Click on the **CF Card** tab to display the CF card dialog. The CF Card tab displays

information about the Compact Flash module. If no CF card is present, the dialog will show None. Otherwise, volume data about the CF card is displayed.

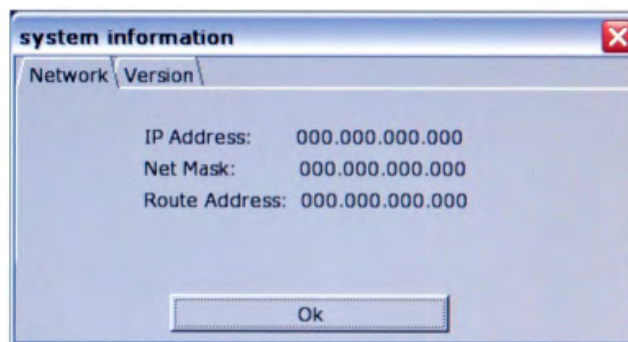


The Eject button will cause the OIT to ignore the CF device. It will not actually eject the card from the OIT. If the Eject button is pressed, the CF card must be removed and re-inserted before the OIT will recognize. Note that USB devices will not be displayed.

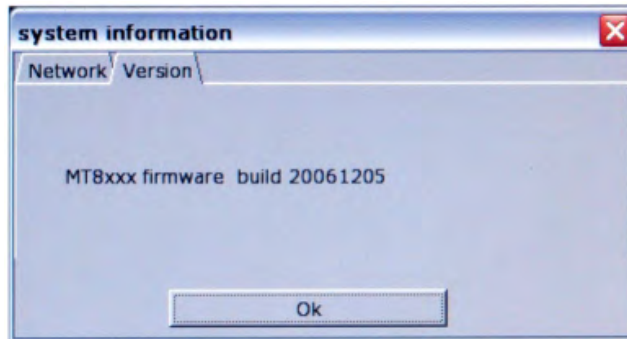
### Viewing System Information

► **To view system information:**

1. Open the Setup Toolbar as described above, and click the System Information icon. A dialog will be displayed with two tabs.
2. Click on the **Network** tab to display the current IP settings.




- Click on the **Version** tab to display the current firmware information.



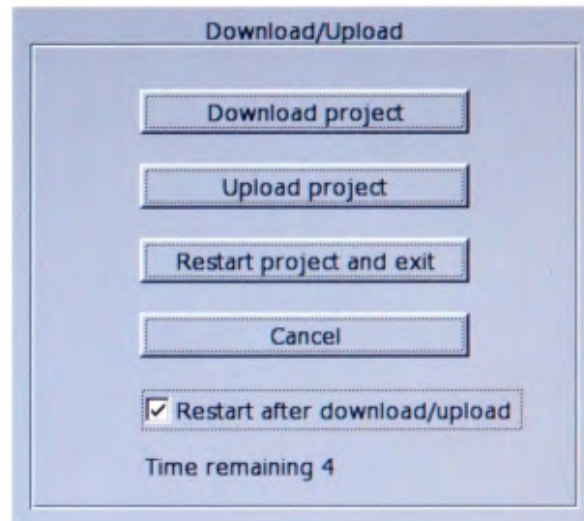
### ***Transferring Projects and Data from USB/CompactFlash***

When a USB or Compact Flash device is inserted into the Silver Series OIT, a dialog is displayed. The same dialog is displayed for transferring projects or data.

---

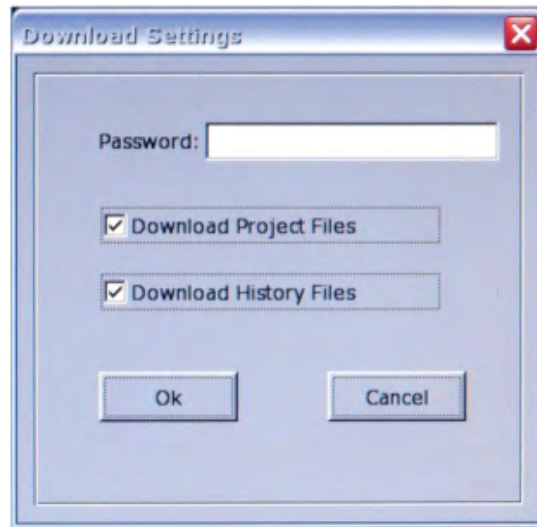
 *Data is placed on the USB or CF device by EasyBuilder or Project Manager.*

---

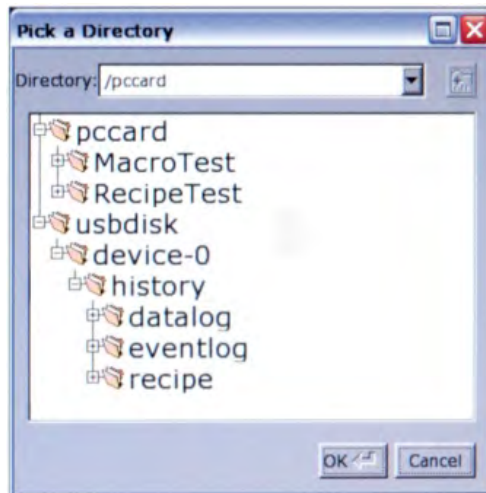


Download Project	Transfer data from the USB/CF to the OIT
Upload Project	Transfer data from the OIT to the USB/CF
Restart Project and exit	Close the dialog and restart the OIT
Cancel	Close the dialog without transferring
Restart after download/upload	When checked, the OIT will restart after the transfer
Time Remaining	The dialog will automatically close after 10 seconds if no selection is made; shows the time remaining

When Download is selected, a dialog will appear requesting the Download password and what data to download.



Password	Enter the <i>Download</i> password.
Download Project Files	When checked, the OIT will check the specified folder for <i>project</i> data, and transfer it if it exists.
Download History Files	When checked, the OIT will check the specified folder for <i>history</i> data, and transfer it if it exists. History data includes Recipe , Event Log, and Data Log data.



## Chapter 3 - Connect the OIT to the PLC or Controller

The Silver Series family of OITs can connect to one, two or more PLC/controllers. This is accomplished via two 9-pin D-sub serial communications connectors (Port A and Port B) and one Ethernet port. Each PLC/Controller has its own wiring requirements. Maple Systems offers OIT-to-PLC/Controller communication cables that connect directly to software controlled COM1, which is available on either Port A or Port B for most PLC/Controllers that are built to any length and tested for high reliability. The Silver Series family has three software controlled communications ports located on two physical connectors, designated as Port A and Port B (see Figures 1 & 2, below).

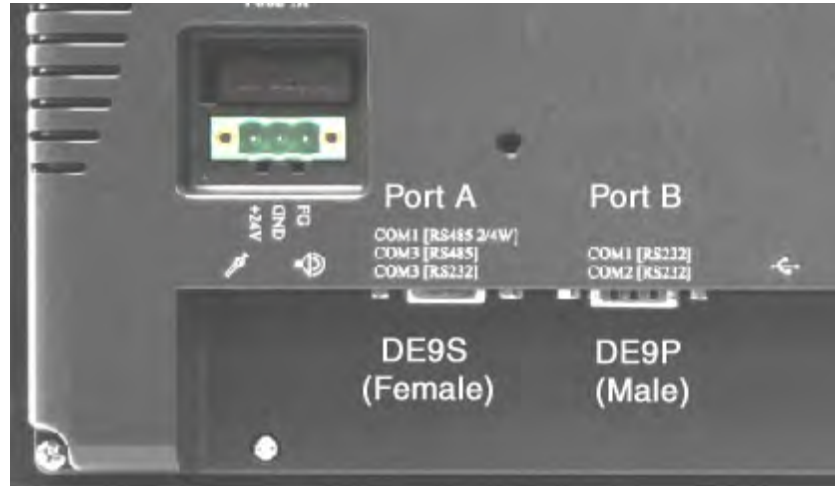


Figure 1 - Com Ports - Rear View



Figure 2 - Com Ports - Bottom View

The three software controlled ports are COM1, COM2, and COM3. Each software controlled communications port can only be selected for one communication type. Once a communication type is selected for one of these software controlled ports, that port cannot be used for another communication type. Each of these communication ports has between one and three communication types. COM1 can be RS232, RS485 4-Wire, or RS485 2-Wire. COM2 is strictly RS232. COM3 is either RS232 or RS485 2-Wire. COM1 cables are readily available for purchase from Maple Systems. COM2 and COM3 cables are custom-made according to wiring diagrams that can be found on Maple Systems' web site. Connecting multiple PLCs and controllers may require a splitter.

**COM Ports for the HMI5080/HMI5100/HMI5104/HMI5121/HMI5150**

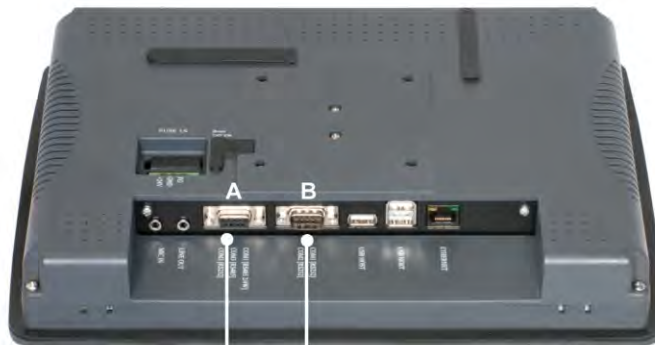
**Port A**



DE9S

- COM1 [RS-485 2 wire]
- COM1 [RS-485 4 wire]
- COM3 [RS-232]
- COM3 [RS-485 2 wire]

Pin#	COM1 [RS-485 2 wire]	COM1 [RS-485 4 wire]	COM3 [RS-232]	COM3 [RS-485 2 wire]
1	TX/RX-	RX-		
2	TX/RX+	RX+		
3		TX-		
4		TX+		
5	Signal Ground			
6				TX/RX-
7			TXD	
8			RXD	
9				TX/RX+



**Port B**



DE9P

- COM1 [RS-232]
- COM2 [RS-232]

Pin#	COM1 [RS-232]	COM2 [RS-232]
1		
2	RXD	
3	TXD	
4		TXD
5	Signal Ground	
6		RXD
7	RTS	
8	CTS	
9		

### COM Ports for the HMI5043/5056/5070

**Port A**



DE9S

- COM1 [RS-485 2 wire]
- COM1 [RS-485 4 wire]
- COM3 [RS-232]
- COM3 [RS-485 2 wire]

Pin#	COM1 [RS-485 2 wire]	COM1 [RS-485 4 wire]	COM3 [RS-232]	COM3 [RS-485 2 wire]
1	TX/RX-	RX-		
2	TX/RX+	RX+		
3		TX-		
4		TX+		
5	Signal Ground			
6				TX/RX-
7			TXD	
8			RXD	
9				TX/RX+



**Port B**



DE9P

- COM1 [RS-232]
- COM2 [RS-232]

Pin#	COM1 [RS-232]	COM2 [RS-232]
1		
2	RXD	
3	TXD	
4		TXD
5	Signal Ground	
6		RXD
7	RTS	
8	CTS	
9		

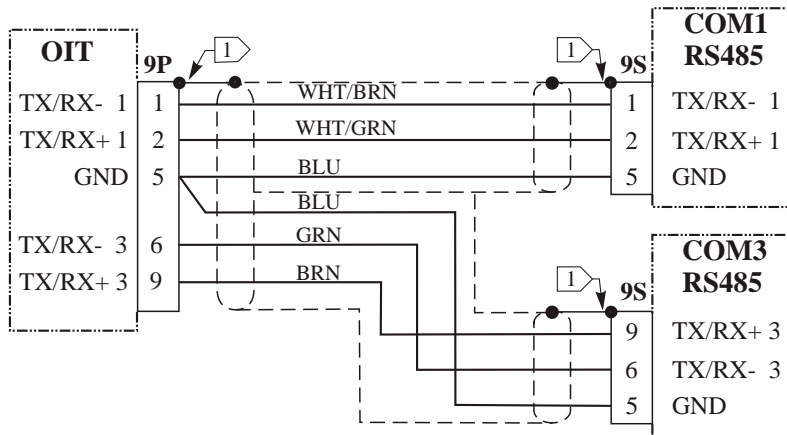
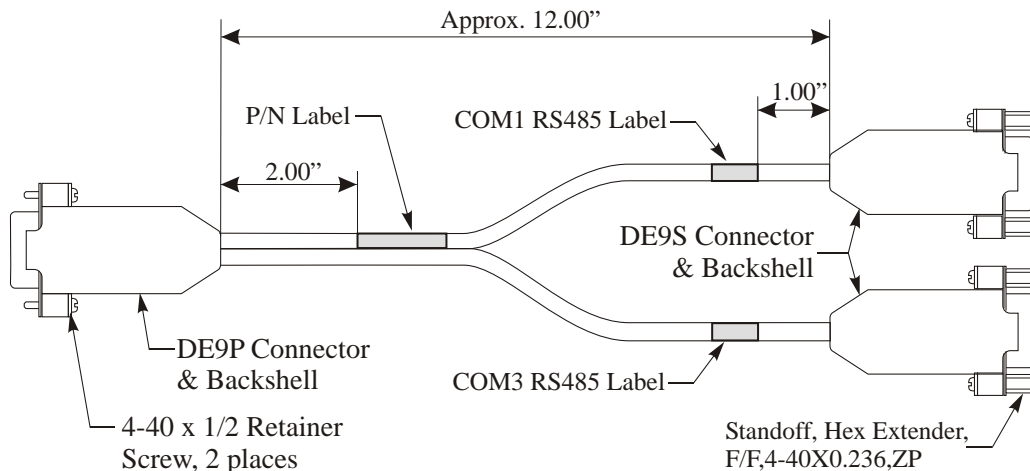


### Connecting to COM1 and COM3 on Port A

Maple Systems Silver Series family of OITs is capable of connecting to multiple PLC/controllers. If you choose to take advantage of this feature, and one of your PLC/controllers uses either RS485-4 Wire or RS485-2 Wire communication, one of four splitters P/N 7431-0111, 7431-0112, 7431-0113, or 7431-0114 will be needed.

#### Scenario 1 – One RS232 PLC/Controller, COM1 configured for RS485-2 Wire

When using two RS485-2 Wire PLC/Controllers, with COM1 configured for RS485-2 Wire and COM3 configured for RS485-2 Wire, splitter P/N 7431-0111 is needed.

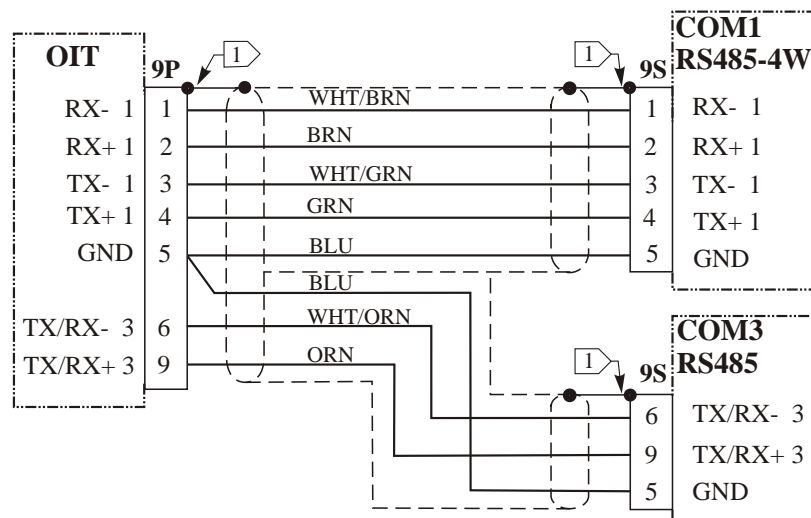
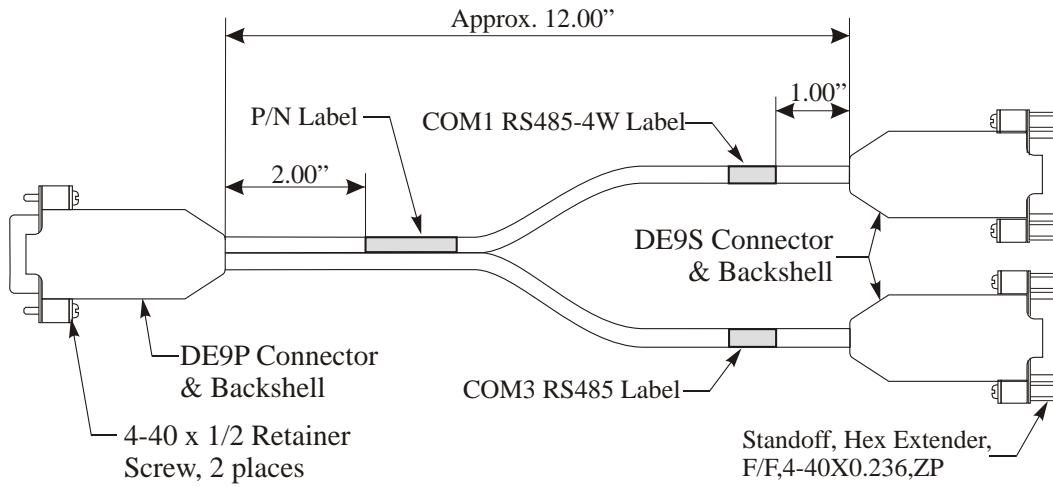


**NOTES:**

- 1 Shield wire must be terminated to connector metal shell.

### Scenario 2

When using one RS485-4 Wire and one RS485-2 Wire PLC/Controller, with COM1 configured for RS485-4 Wire and COM3 configured for RS485-2 Wire, splitter P/N 7431-0112 is needed.

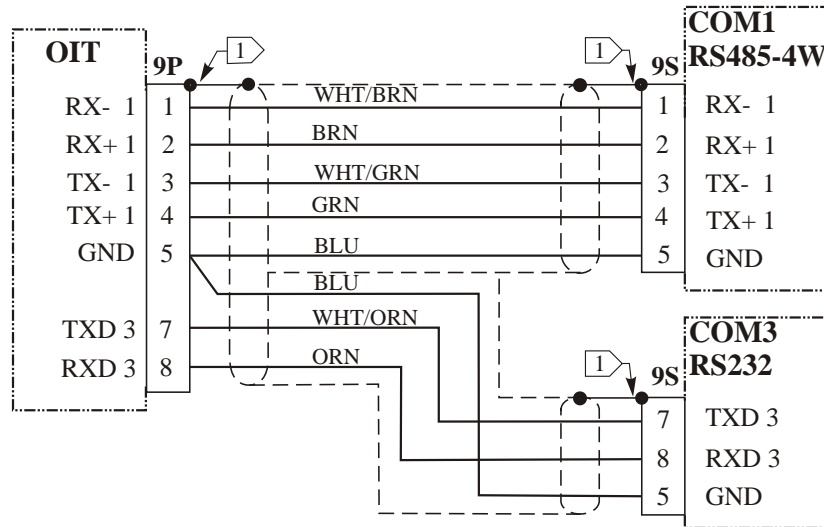
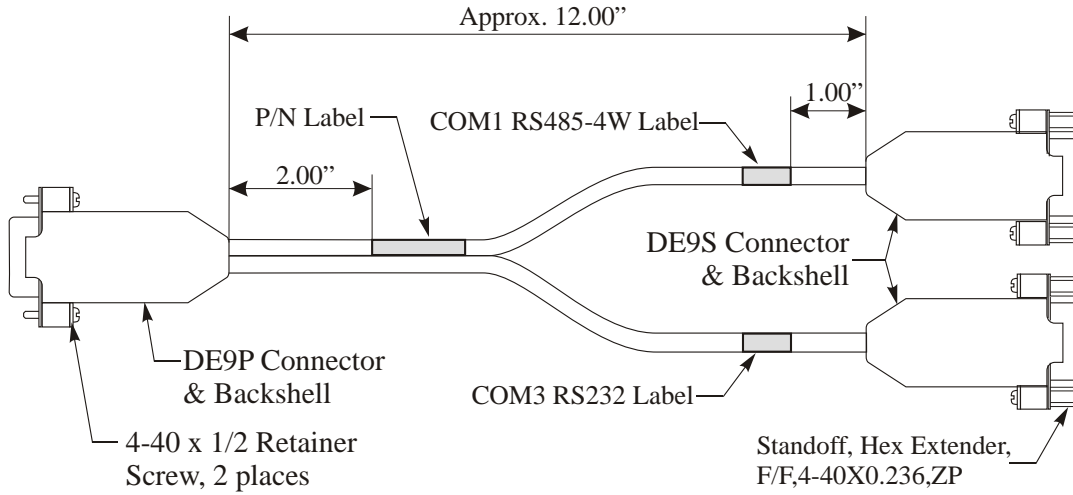


**NOTES:**

- ① Shield wire must be terminated to connector metal shell.

### Scenario 3

When using one RS485-4 Wire and one RS232 PLC/Controller, with COM1 configured for RS485-4 Wire and COM3 configured for RS232, splitter P/N 7431-0113 is needed.

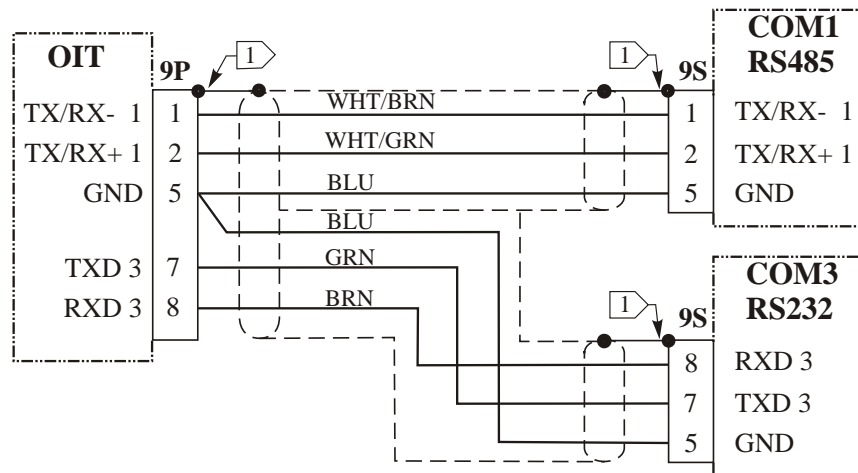
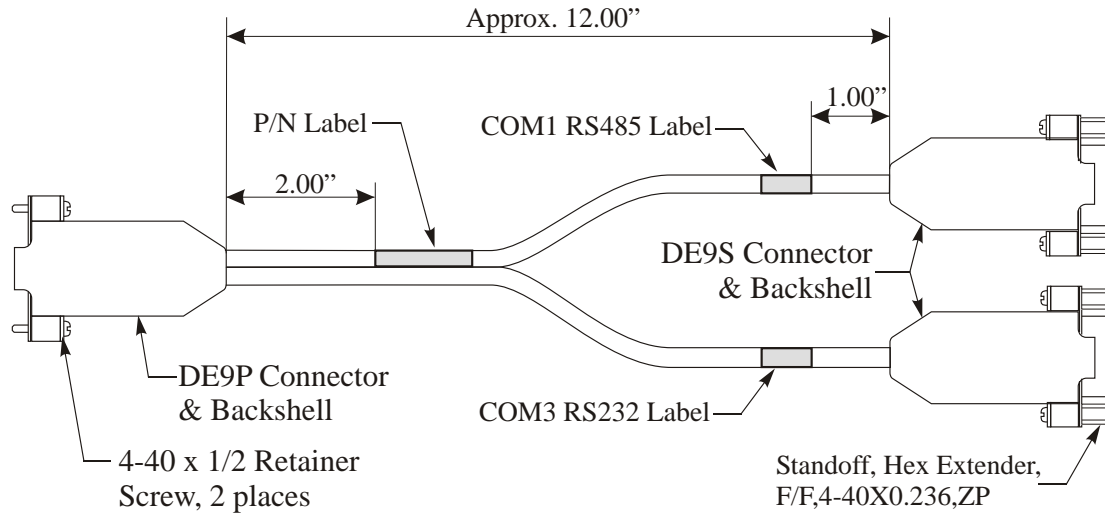


**NOTES:**

- 1 Shield wire must be terminated to connector metal shell.

### Scenario 4

When using one RS485-2 Wire and one RS232 PLC/Controller, with COM1 configured for RS485-2 Wire and COM3 configured for RS232, splitter P/N 7431-0114 is needed.

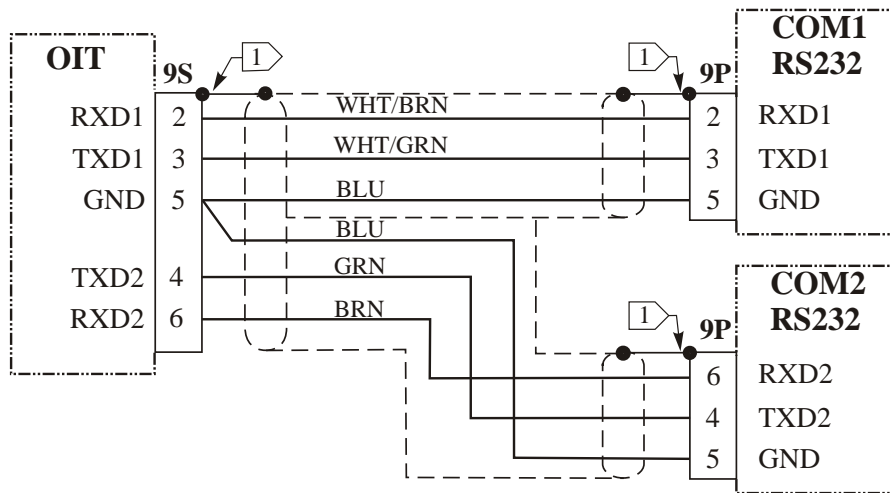
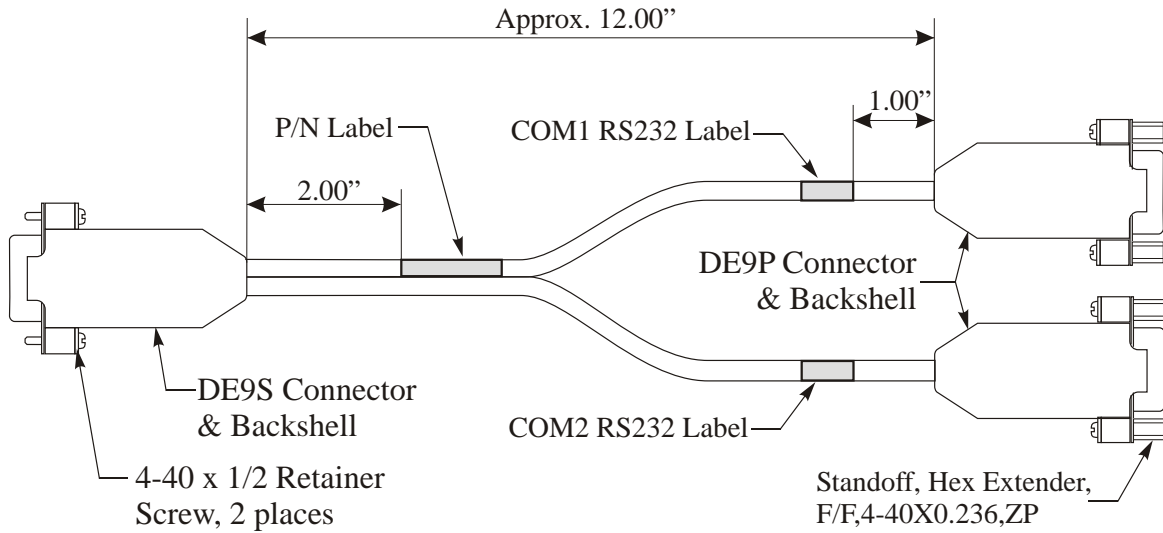


**NOTES:**

① Shield wire must be terminated to connector metal shell.

### Connecting to COM1 and COM2 on Port B

If both controllers use RS232 communication then only one splitter, P/N 7431-0110 is needed.



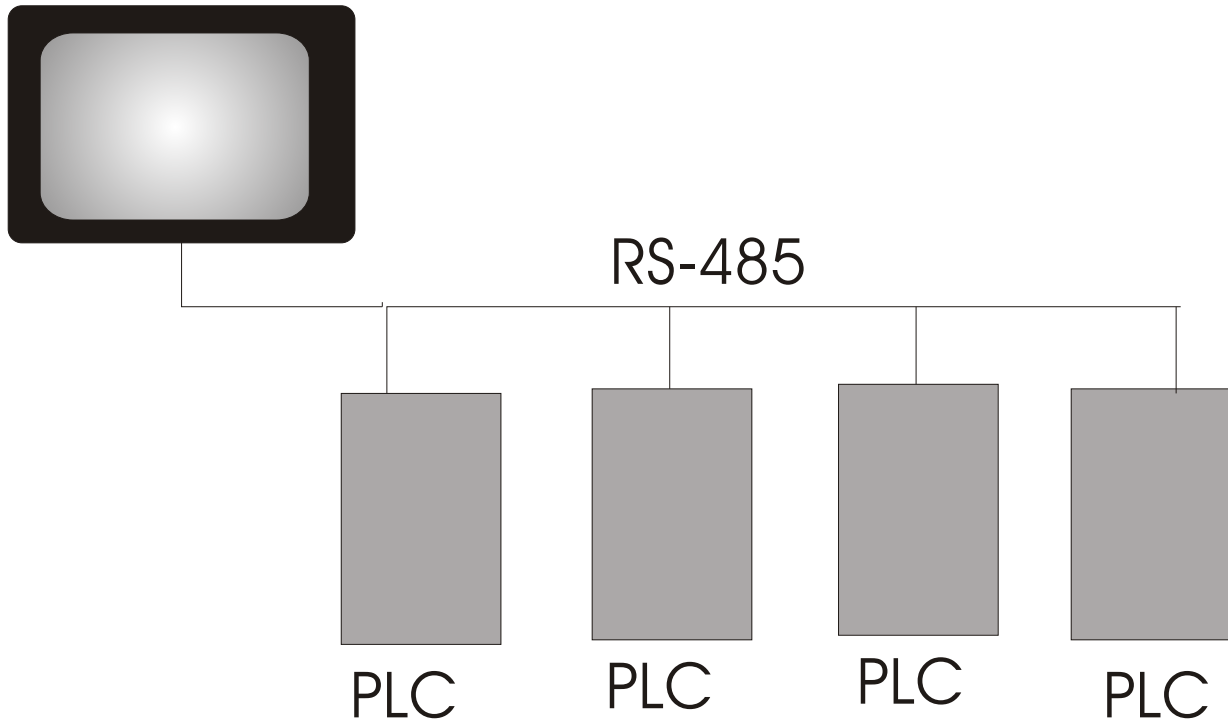
**NOTES:**


- 1) Shield wire must be terminated to connector metal shell.

### Connecting multiple PLC/controllers serially using daisy chaining

The RS485 4-wire and RS485 2-wire communications ports of the Silver Series family supports multi-drop connections. A daisy-chain connection is made from the OIT to the first PLC/controller, and then from the first PLC/controller to the second and so on (see below). The network may need biasing resistors – refer to the PLC/controller documentation for more information.

## Silver Series OIT



 Refer to the Help Section of EasyBuilder-5000, under the How do I Section... for more information.

### Connecting one or more PLC/controllers via Ethernet

The Silver Series OITs have a single RJ45 Ethernet connector, configured for 10BaseT and 100BaseT operation. This port is for both communicating with a PC for configuration purposes, as well as for communicating with supported Ethernet PLCs and protocols.

The connector is wired just like the RJ45 Ethernet connector on a PC. The Ethernet connection to a PLC/controller will vary with each PLC/controller. Some will require a straight-through connection, while others require a crossover connection. See Maple Systems' website for cable drawings of Ethernet cables.

#### ► To connect a single PLC/controller to an OIT:

1. Attach one end of the Ethernet cable to the RJ45 connector on the OIT
2. Attach the other end to the RJ45 connector on the PLC/controller

A router or switch is required to connect multiple PLCs/controllers to a single OIT. In general, standard straight-through type cables can be used when connecting to a PLC/controller through a router or switch. Some routers and switches may be *auto-sensing*. In this case, either type of cable may be used.

**Please Use This Page For Your Notes**

## Chapter 4 - Creating Your First Project

Often the best way to learn about new software is to jump right in. This chapter will step you through the process of installing the EZware-5000 configuration software and then using the software to create a sample project that can be downloaded to your OIT. We won't go into much detail about how each feature works. The purpose of this chapter is only to provide you with an overview of the process of creating a functional OIT that can communicate to a PLC. For our sample project, we will configure the OIT using the local OIT Memory, but you may feel free to select whichever protocol driver you intend to use.

By the end of this chapter, you should be able to:

- Install EZware-5000 configuration software.
- Create a sample project with two windows and several graphics objects.
- Save a project, compile a project and download the project to the OIT.
- Verify that the OIT is functioning properly.

### Before You Begin

Before you install EZware-5000, make sure your computer meets the following minimum system requirements:

- Pentium-based 1.7 GHz or higher processor
- 512 MB of RAM (more memory improves performance)
- 200 MB available hard disk space
- VGA or higher-resolution monitor set for 16 Bit color 1024 x 768 pixel mode
- Microsoft mouse or compatible pointing device
- One available Ethernet port
- Microsoft Windows XP, 2000 or Vista

### Connecting OIT to Computer

Before you start your first project, the OIT should be connected to the computer so that the project can be downloaded after creating it. You should also connect the PLC that you are using to the OIT so that you can test the operation of the OIT after you have finished creating this sample project.

#### ► To connect your OIT to the computer

1. Connect a +24VDC power supply to the OIT.
2. Connect a crossover Ethernet cable between the OIT and PC.
3. Apply power to the OIT.

### Starting EZware-5000

Before you can create a sample project, you must start the configuration software. The EZware-5000 software has two main applications:

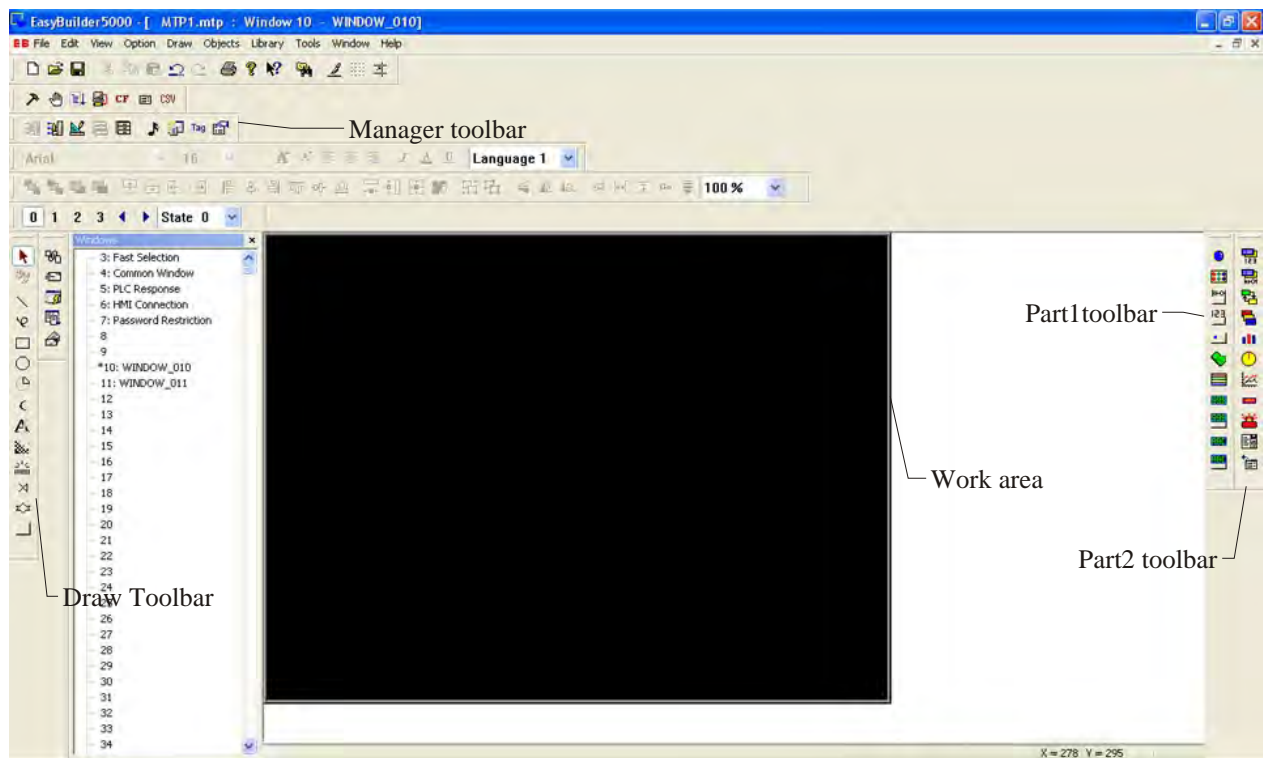
- **EasyBuilder** – used to create the project downloaded to the OIT.
- **Project Manager** - used to place the OIT into different operating modes.

#### ► To start the EZware-5000

1. From the Windows Task Bar, click the **Start** button, point to **Programs**, and then click the **Maple Systems-EZware-5000-Project Manager**.
2. On the Project Manager dialog box, click **EasyBuilder5000**.
3. The Welcome to EasyBuilder dialog box appears. Select the correct *OIT model*.
4. Click **OK** to display the System Parameters screen of EasyBuilder.



The following illustration shows the various sections of EasyBuilder.



## Creating a Sample Project

This section walks you through the creation of an EasyBuilder project named MTPPrj1. Once downloaded to the OIT, this basic configuration allows the OIT to connect to the PLC, display a startup screen, and display a screen containing one PLC register monitor when a switch on the startup screen is pressed.

Although we strongly recommend that you perform the following steps to create this sample project, the project is already included in your EasyBuilder software with the following filenames:

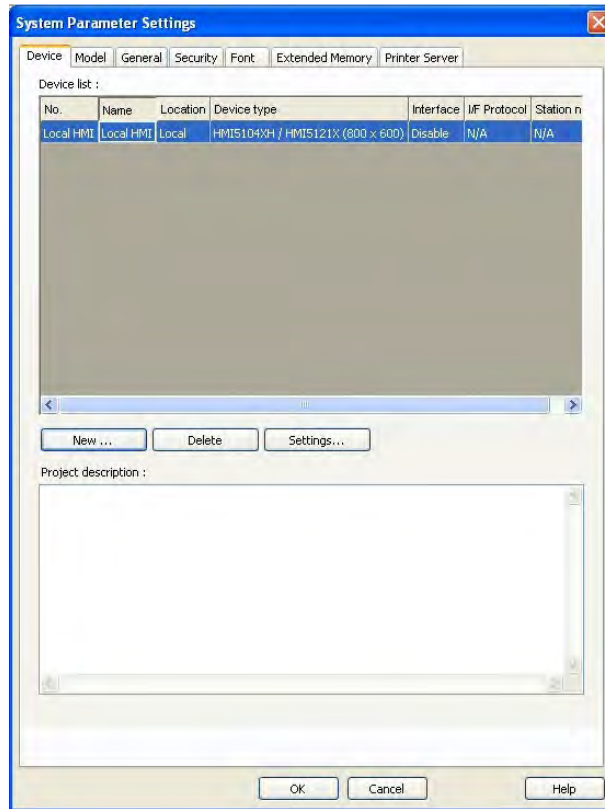
- HMI5056.MTP-sample project for the HMI5056
- HMI5070.MTP-sample project for the HMI5070
- HMI5080.MTP-sample project for the HMI5080
- HMI5104.MTP-sample project for the HMI5104
- HMI5121.MTP-sample project for the HMI5121
- HMI5150.MTP-sample project for the HMI5150

## Setting the System Parameters

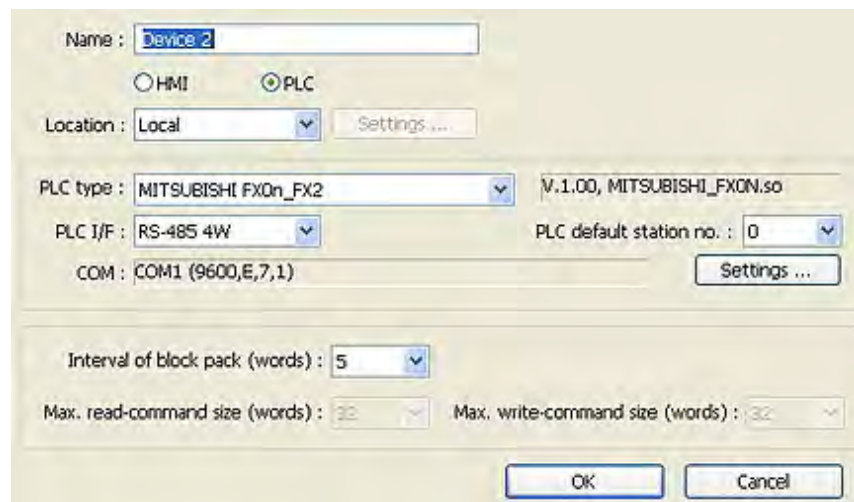
Whenever you begin a new project, you should always set the system parameters before you create any windows. System parameters determine the basic operating conditions of the OIT such as what type of PLC it is connecting to.

- ▶ To edit the System Parameters
  1. Click the Edit menu on the main screen of EasyBuilder.

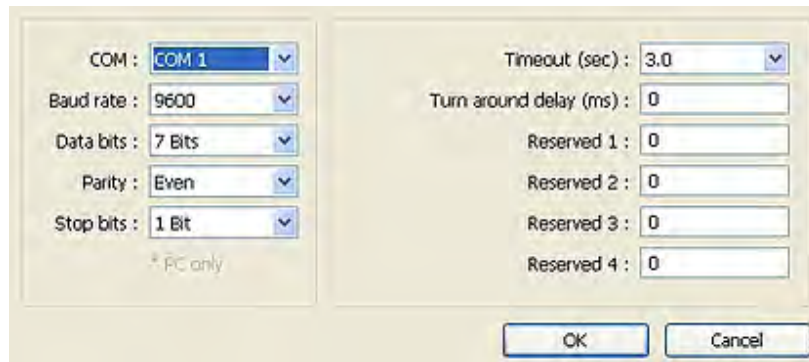
- At the bottom of the menu, click **System Parameters**. The System Parameters dialog box appears.



- The dialog box has six tabs: Device, Model, General, Security, Font and Extend Memory. Select the **Device** tab.
- Click **New...** The Printer Server Dialog appears.



5. Select *your controller* from the drop down menu next to PLC Type: Click **Settings...** The Com Port Settings dialog appears.



The image shows a 'Com Port Settings' dialog box with the following fields and values:

COM :	COM 1	Timeout (sec) :	3.0
Baud rate :	9600	Turn around delay (ms) :	0
Data bits :	7 Bits	Reserved 1 :	0
Parity :	Even	Reserved 2 :	0
Stop bits :	1 Bit	Reserved 3 :	0
	* PC only	Reserved 4 :	0

Buttons: OK, Cancel

6. Set *Com Port Settings* that match your PLC. Reserved settings are specific to each type of PLC. Refer to the Controller Information Sheet for that PLC for more details. Click **OK** twice to return to System Parameter Settings dialog

7. Now click on the **General** tab.



8. In the Option section, for the Startup Window no: drop-down, select **10. WINDOW\_010**

9. Click **OK** to return to the EasyBuilder main screen.



*For more information, consult your PLC operations manual or Maple Systems Controller Information Sheets available on our website at [www.maple-systems.com](http://www.maple-systems.com).*

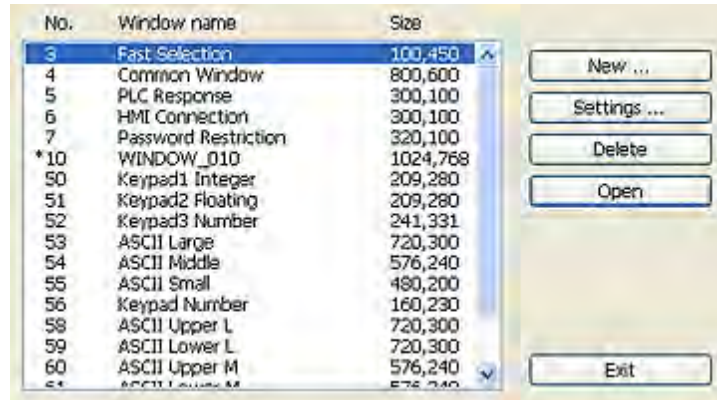
## Creating a Popup Window

We will first configure Window #11 as a popup window. We will be using a function key in the startup window to call this popup window; however, this window must first exist before we can map the function key to it, which is why we are creating the popup prior to creating the startup window.

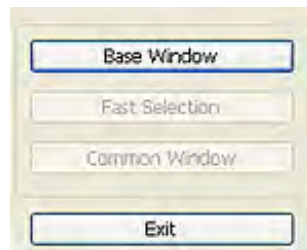
Multiple popup windows can be displayed on a full screen window. The windows may overlap each other or be moved anywhere onscreen. This section will show how to create a scale and a numeric register that displays the current value of the scale. You will also create an increment and decrement key to change the value in the scale meter.

► To create **Window#11**

1. From the Window menu, click **Open Window**. The Open Window dialog box appears.



2. Click **New...** The Select Window Style dialog box appears.



3. Click **Base Window**. The Window Setting dialog box appears:

Name : Window\_011

Window no. : 11

**Size**

Width : 210      Height : 150

**Frame**

Width : 4      Color :

**Background**

Color :      Pattern : ...

Filled      Pattern color :

**Underlay window**

Bottom : None

Middle : None

Top : None

**Popup window**

Start pos.    X : 60      Y : 60       Monopoly

4. Type **WINDOW\_011** in the Name: field and **11** in the Window no: field.
5. In the Size section, enter **210** for the width. Enter **150** for the height.
6. Click the pull down box from the Background Color box. The Color dialog box appears.
7. Click on the white color box, then click **OK**. The color box will reflect the color that you have chosen.
8. In the Popup window section, enter **60** for the X position. Enter **60** for the Y position.
9. Click **OK**. The Open Window dialog will reappear.
10. In the Open Window dialog, highlight **Window 11**, then click on the **Open** button. Window 11 now appears on your work area, configured to the size indicated above.

► **To create a meter display on Window#11**

1. From the Objects menu, click **Meter Display**. The New Meter Display Object dialog box appears.

The screenshot shows the 'New Meter Display Object' dialog box with the 'General' tab selected. The 'Description' field contains 'Meter Display'. Under the 'Read address' section, the 'PLC name' is set to 'Allen-Bradley CompactLogix/FlexLogix', the 'Device type' is 'Tx,ACC', and the 'Address' is '0'. There is an unchecked checkbox for 'Index register' and a dropdown menu set to '16-bit Unsigned'.

2. Type **Meter Display** in the Description box.
3. Click the **Outline** tab to configure the meter style.

The screenshot shows the 'New Meter Display Object' dialog box with the 'Outline' tab selected. The 'Degree' section has 'Start degree' at 270 and 'End degree' at 90. The 'Background' section has a color selector, a 'Profile' dropdown, and checkboxes for 'Full circle' and 'Transparent'. The 'Tick marks' section has 'Enable' checked, 'Color' dropdown, 'Coordinate' checked, 'Main scale' at 6, 'Sub. scale' at 1, and 'Length' at 6. The 'Pointer' section shows a pointer preview, 'Arm style ...' button, 'Color' dropdown, 'Length' at 3, and 'Width' at 4. The 'Pin point' section has 'Radius' at 1, 'Inner' dropdown, 'Frame' dropdown, and radio buttons for 'Circle', 'Rectangle' (selected), and 'Image'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

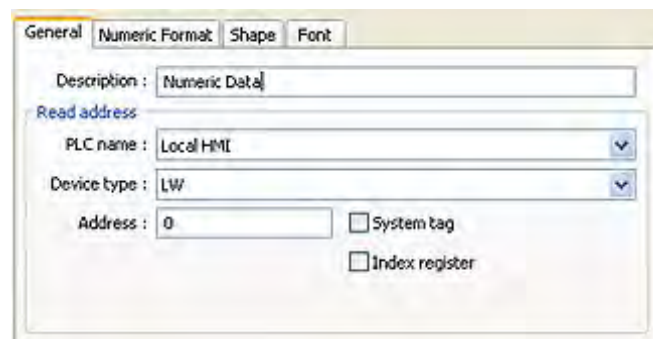
4. In the Start Degree, type **270**
5. In the End Degree box, type **90**.
6. Uncheck "Full Circle".
7. Check **Enable** in the tick marks section.
8. Select black as the tick marks color.
9. Select main scale **6**, sub. scale **1**, length **6** for tick marks.
10. Select pointer arm style and length.
11. Click on the Limits tab.
12. Set value **0**, Span **100**.
13. Enable range limits and select *low*, *mid* and *high* colors. Width = 3.
14. Click **OK**.
15. Click the work area to place the meter.
16. On the main screen of EasyBuilder, you will see a white square outline that is attached to your cursor in the work area. This represents the meter display just created. Click to place the meter display in Window #11.
17. Now, resize the meter so that it is a better fit on the popup window. Double click on the meter display to re-display the attribute dialog.
18. Click the **Profile** tab to display the profile form.



19. Enter the following values:  
**X: 45**  
**Y: 13**  
**Width: 120**  
**Height: 120**
20. Click **OK**.

► **To create a numeric register on Window#11**

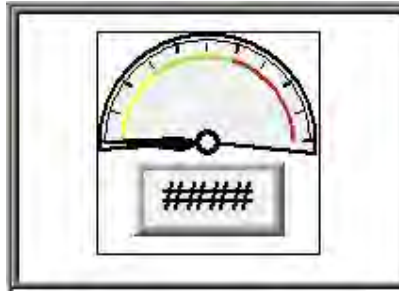
1. From the Objects menu, click **Numeric/ASCII - Numeric Display**. The New Numeric Display dialog box appears.



2. Type **Numeric Data** in the Description box.
3. Click **OK**.
4. On the main screen of EasyBuilder, you will see a white rectangle outline that is attached to your cursor in the work area. This represents the numeric register just created. Click to place



the numeric register on Window #11 somewhere underneath the scale meter. A numeric register box will appear.



► **To create an increment key on Window#11**

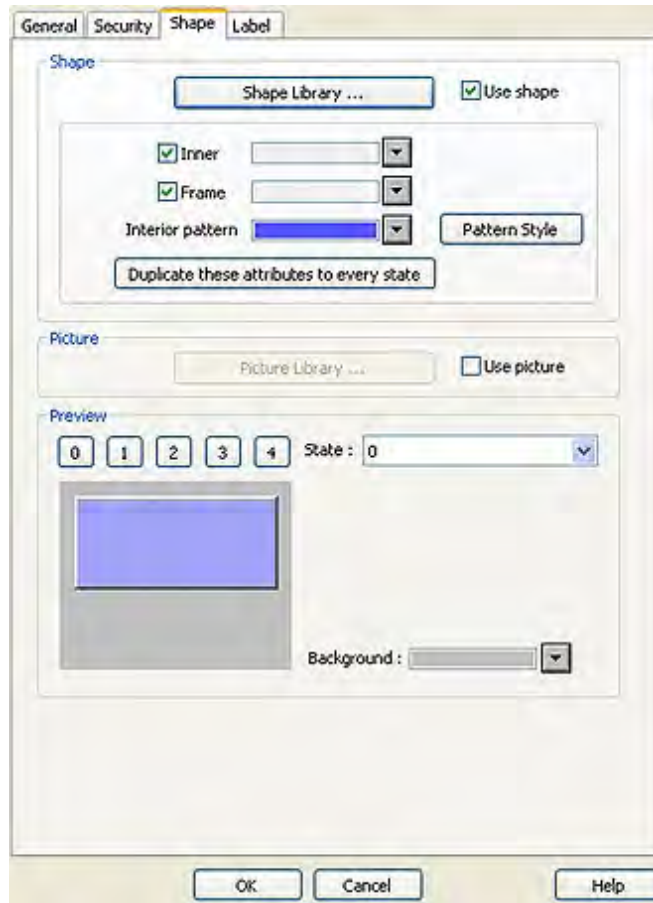
1. From the Objects menu, click **Button-Set Word**. The New Set Word Object dialog box appears.

The dialog box has the following fields and settings:

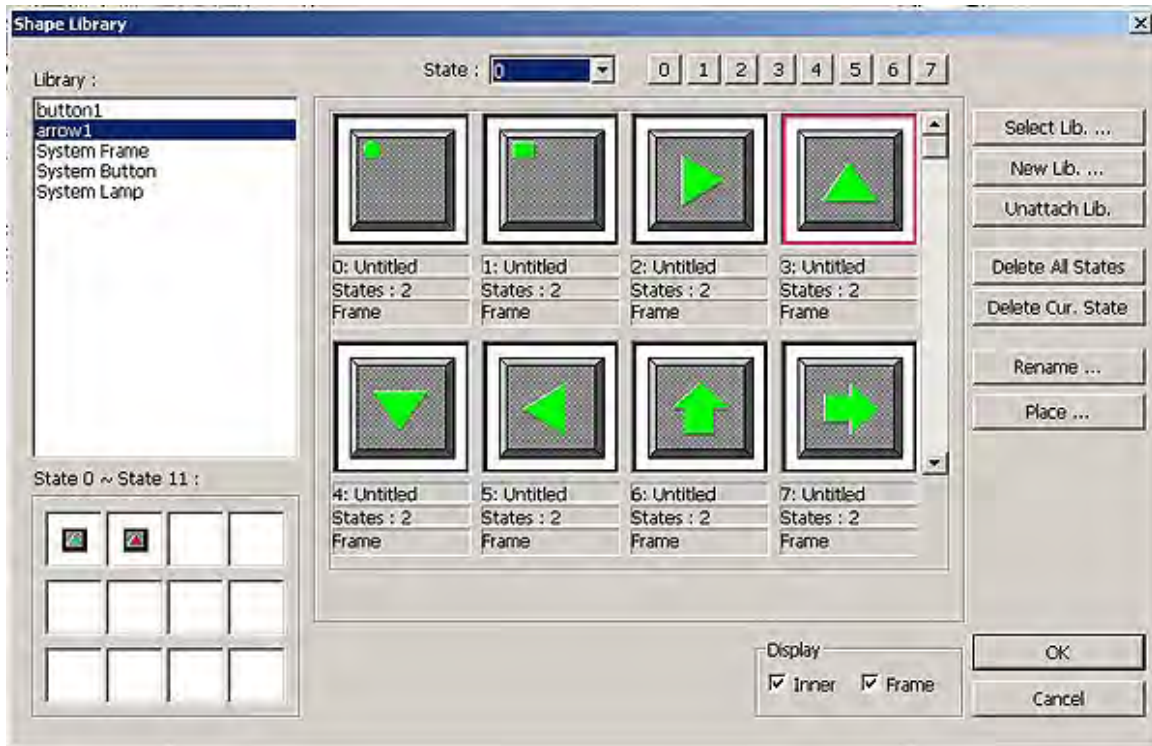
- General** tab selected.
- Description: [Empty]
- Write address** section:
  - PLC name: Local HMI
  - Device type: LW
  - Address: 0
  - System tag:
  - 16-bit Unsigned: selected
  - Index register:
- Notification** section:
  - Enable:
- Attribute** section:
  - Mode: Press and hold increment (JOG++)
  - Inc. value: 1
  - Upper limit: 100
  - JOG delay: 0.5 second(s)
  - JOG speed: 0.1 second(s)
- Buttons: OK, Cancel, Help

2. Type **Increment Button** in the Description box.
3. Click on the pull down box of the Mode: box and select **Press and hold increment (JOG++)**
4. Enter **1** in the Inc. value box.
5. Enter **100** in the Upper limit box.
6. Click on the pull down box of JOG Delay and select **0.5 second**.

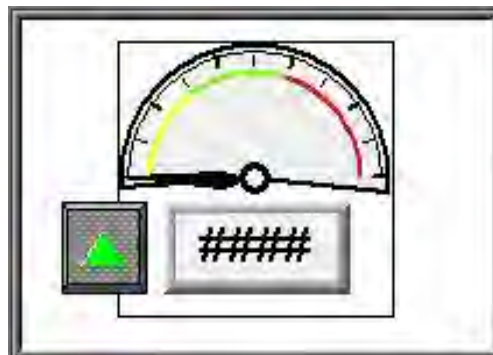
7. Click the **Shape** tab.



8. Check **Use shape**, then click **Shape library**. The Shape Library dialog box appears.



9. Click **Select Lib...** The Open dialog box appears.
10. Select **arrow1.plb** from the list of shape libraries. Then click **Open**.
11. Click **arrow1** in the Shape library box.
12. Scroll through the selections to selection 3. Click the shape, then click **OK**. The shape will appear in the Shape tab.
13. Click **OK**.
14. On the main screen of EasyBuilder, you will see a white square outline that is attached to your cursor in the work area. This represents the increment key just created. Click to place the increment key on Window #11.
15. Use the blue handles surrounding the key to drag the increment key to the desired size and then place to the left of and below the numeric object.



► **To create a decrement key on Window#11**

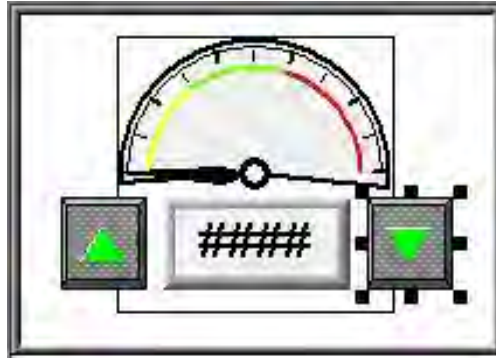
1. We are going to create the decrement key by copying the increment key and then making changes to the attributes.
2. Click the increment key to highlight it.
3. From the Edit menu, click **Copy**.

4. From the Edit menu, click **Paste**. A copy of the increment key appears in the upper left corner of the work area. Deselect the increment key by clicking on any blank space in the work area.
5. Double-click on the second increment key. The Set Word Object's Attribute dialog box appears.

The image shows the 'Set Word Object's Attribute' dialog box with the 'General' tab selected. The 'Description' field is set to 'Decrement button'. Under the 'Write address' section, 'PLC name' is 'Local HMI', 'Device type' is 'LW', and 'Address' is '0'. The bit width is set to '16-bit Unsigned'. There are checkboxes for 'System tag' and 'Index register', both of which are unchecked. In the 'Notification' section, the 'Enable' checkbox is also unchecked. Under the 'Attribute' section, the 'Mode' is set to 'Press and hold decrement (JOG-)', 'Dec. value' is '1', 'Bottom limit' is '0', 'JOG delay' is '0.5 second(s)', and 'JOG speed' is '0.1 second(s)'. At the bottom, there are 'OK', 'Cancel', and 'Help' buttons.

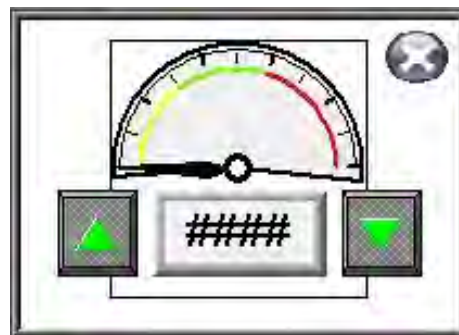
6. In the General tab section, change the Description to **Decrement Button**.
7. Click on the pull down box of the Mode Attribute: and select **Press and hold decrement (JOG—)**.
8. Enter **0** in the Attribute Bottom limit box.
9. Click the **Shape** tab.
10. Click **Shape library**. The Shape Library dialog box appears.
11. Scroll through the selections to selection 4. Click the shape, then click **OK**. The shape should appear in the Shape tab.
12. Click **OK**.

13. Drag the decrement button to the bottom right area of your window.



► **To create a close window button**

1. Finally, we must create a button that will close the popup window. From the Objects menu, click **Function Key**. The New Function Key Object dialog box appears.
2. Type **Close Window** in the Description box.
3. Click the **Close Window** radio button.
4. Click the **Shape** tab to display the shape dialog.
5. Unclick the **Use shape** checkbox and click the **Use picture** checkbox. Click the **Picture library button**. The Picture Library dialog box appears.
6. Click the **Select Lib. Button** on the bottom of this dialog box. Select the **computer.flb** library, then click **Open**.
7. Under the Library: list box, you will now see computer. Click on **computer** to display this library.
8. Scroll through the selections to selection 21. Click the **picture**, then click **OK**. The color box will reflect what you have chosen.
9. Now click the **Label** tab and make sure **Use Label** checkbox is not checked. Click **OK**.
10. On the main screen of EasyBuilder, you will see a white square outline that is attached to your cursor in the work area. This represents the function key just created. Move this key to the upper right corner of the screen and click to place.
11. Now double-click on this key to redisplay the attribute box. Click on the **Profile** tab.
12. Enter **X:170** and **Y:7** for position values. Enter **Width: 33** and **Height: 28** for the size values.
13. Click **OK**.



### **Creating a Startup Window**

We will now configure Window #10 as the startup screen. The OIT can store up to 1999 predefined windows but screens 0-9 are reserved. Therefore, the first window screen that is available for configuration is Window #10. By default, this is the startup screen when you begin a new project. This section will show how to place text in the window and how to create two function keys that will open and close Window #11.

To Display Window #10, from the Window menu, click 1 MTP1-Window10. Window 10 will now appear in your workspace.

► **To place text on Window#10**

1. From the Draw menu, click **Text**. The New Text Object dialog box appears.

Use label library  
 Use bitmap font

**Attribute**

Font : Arial

Color :

Size : 16

Align : Left

Blink : None

Italic     Underline

**Movement**

Direction : No movement

**Content**

This is the startup screen.

Tracking    Duplicate this label to every state

2. Click the pull down box from the Color box. The Color palette appears.



3. Click on the white color box, then click **OK**. The color box will reflect the color that you have chosen.
4. Select **16** in the size box.
5. Select **Left** in the Align box.
6. Double-click the word 'text' in the Content box and type "**This is the Startup Screen.**"
7. Click **OK**.
8. On the main screen of EasyBuilder, you will see a white rectangle outline that is attached to your cursor in the work area. This represents the text box just created. Center the rectangle somewhere on the top third portion of the work area and then click. The text box "This is the Startup Screen" will appear.

► To create a function key on Window#10

1. From the Objects menu, click **Button-Function Key**. The New Function Key Object dialog box appears.

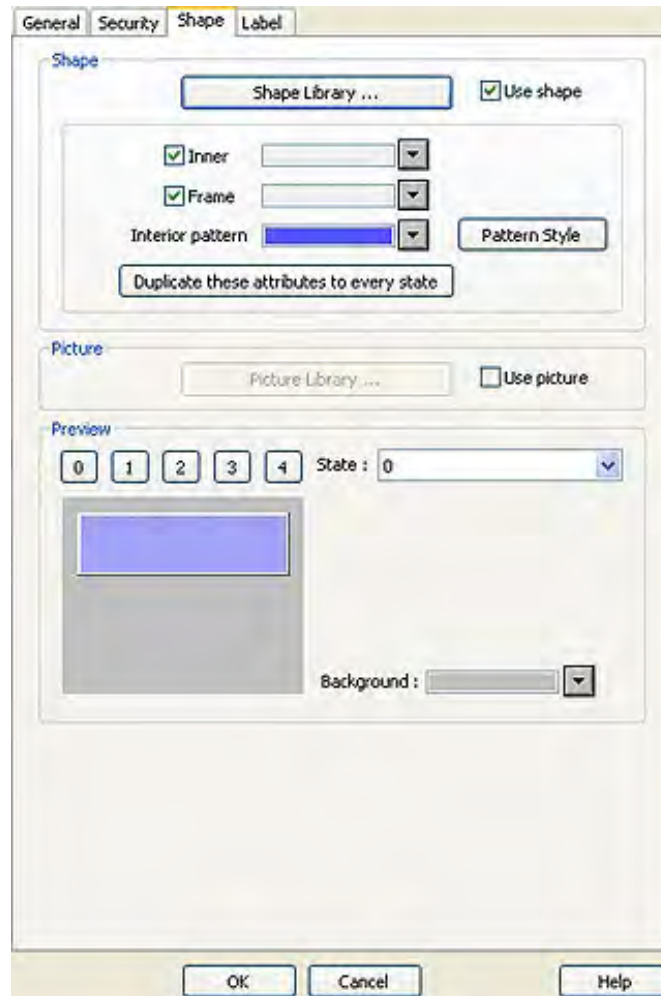


The image shows the 'New Function Key Object' dialog box with the 'General' tab selected. The dialog has four tabs: 'General', 'Security', 'Shape', and 'Label'. The 'Description' field is empty. There is a checkbox for 'Activate after button is released' which is unchecked. Under the 'Window' section, there are three radio buttons: 'Change full-screen window' (selected), 'Change common window', and 'Display popup window'. The 'Window no.' dropdown menu is set to '50. Keypad Integer'. Below this, there are two radio buttons: 'Return to previous window' and 'Close window', both unchecked. The 'ASCII mode' section has four radio buttons: '[Enter]', '[Backspace]', '[Clear]', and '[Esc]', all unchecked, and one more radio button for '[ASCII]' which is also unchecked. There are two more radio buttons: 'Execute macro' and 'Hard copy', both unchecked. The 'Notification' section has a checkbox for 'Enable' which is unchecked. At the bottom, there are three buttons: 'OK', 'Cancel', and 'Help'.

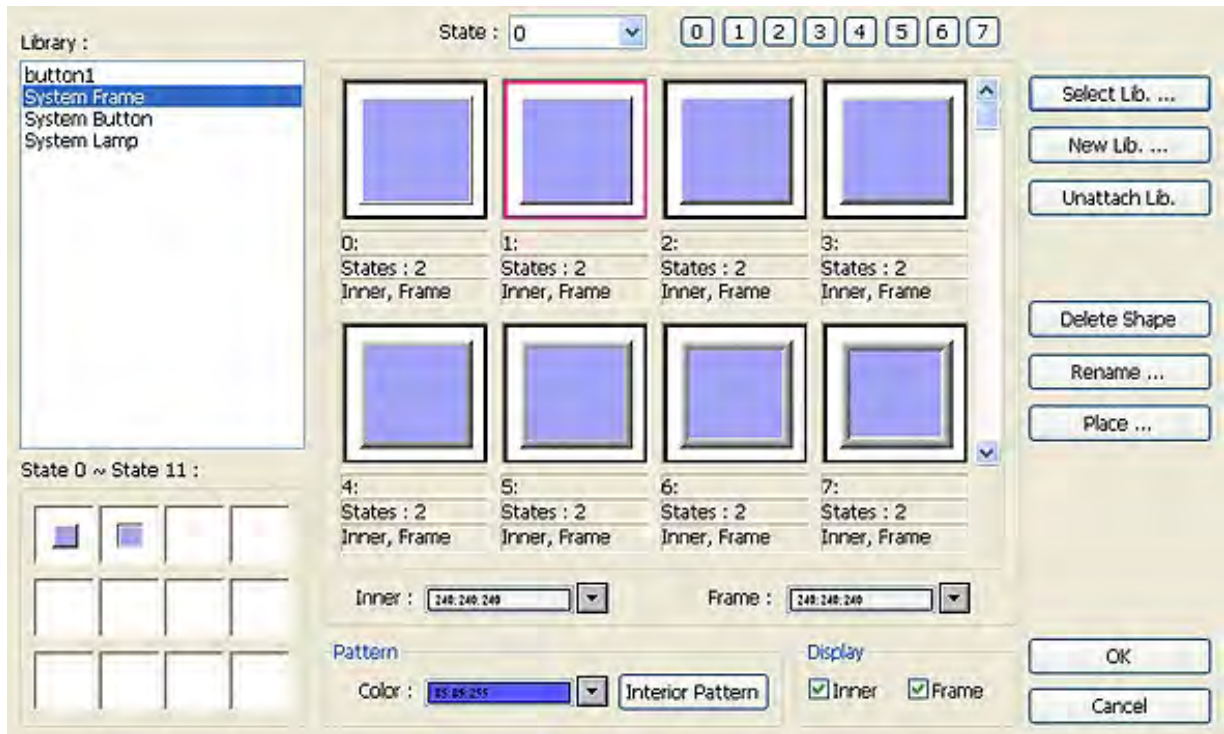
2. Type **On Button** in the Description box.
3. Click Display Popup Window.
4. Type **11** in the Window No. Box.



5. Click the **Shape** tab to display the shape dialog..



6. Check the **Use shape** checkbox, and then click **Shape library**. The Shape Library dialog box appears.

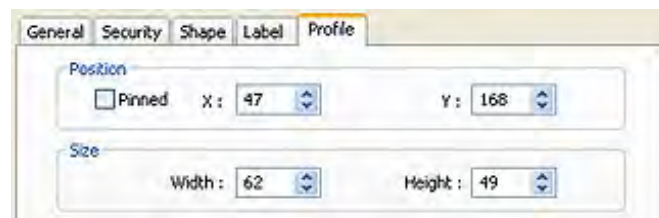


7. Click **button1** in the Shape library box.
8. Scroll through the selections to selection 23. Click the shape, then click **OK**. The shape will appear in the Shape tab.

- Click the **Label** tab. The label dialog appears.



- Check the **Use Label** checkbox.
- Click the pull down box from the Color box. The Color dialog box appears.
- Click on the black color box, then click **OK**. The color box will reflect what you have chosen.
- Select **16** in the Size box.
- Select **Left** in the Align box.
- Type **ON** in the Content box..
- Click **OK**.
- On the main screen of EasyBuilder, you will see a white square outline that is attached to your cursor in the work area. This represents the function key just created. Center the square somewhere on the bottom left portion of the work area and then click. The function key will appear.
- Double-click on the function key. The Function Key Object's Properties box appears.



19. Click on the **Profile** tab.
20. Enter **47** for the X position. Enter **168** for the Y position.
21. Enter **62** for the Width. Enter **49** for the Height.
22. Click **OK**.
23. On the main screen of EasyBuilder, you will see the function key that you just created move and/or change size. This function key is used to display Window #11.

► **To create a second function key on Window#10**

1. We could create the second function key by repeating the steps for the first function key. Instead, however, we will take advantage of the first function key and copy it.
2. Click the first function key to highlight it.
3. From the Edit menu, click **Copy**.
4. From the Edit menu, click **Paste**. A copy of the function key appears in the upper left corner of the work area. Deselect the function key by clicking on any blank space in the work area.
5. Double-click on the second function key. The Function Key's Object Attribute dialog box appears.
6. In the General tab section, change the Description to **Off Button**.
7. Click on **Close Window**. Enter **10** for Window number
8. Click on the **Label** tab. Type **OFF** in the Content box.
9. Click on the **Profile** tab.
10. Enter **196** for the X position. Enter **168** for the Y position.
11. Click **OK**.
12. On the main screen of EasyBuilder, you will see the second function key that you just created move to the lower right hand side of the work area. This function key is used to close Window #11.

You have finished configuring your first window. It will look similar to the picture below:




You have now done your part in creating this sample project. It is now time for EZware -5000 to do its part.

## Finishing Up

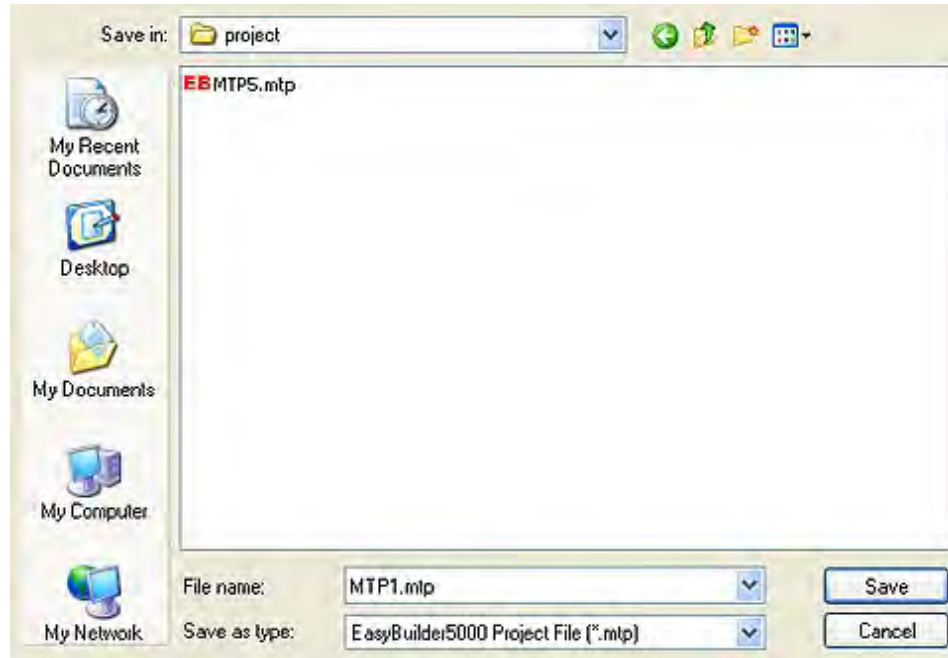
There are still a few steps, which must be completed before you can test your first project. In this section, you will:

- save the project onto your computer hard drive
- compile the project into a format that can be understood by the OIT
- download the project to the OIT
- verify that the OIT operates as expected
- exit the EZware-5000 software

 If you haven't already done so, now would be a good time to connect the OIT to the computer and to connect the PLC that you are using to the OIT. For more information, consult the first part of this chapter or see "Installation of OITs" later on in this manual.

### ► Saving your first project

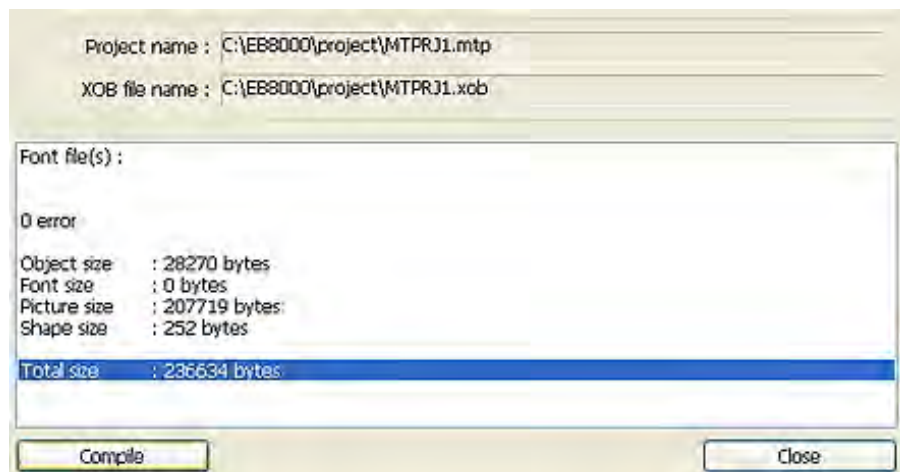
1. From the File menu, click **Save As**. The Save As dialog box appears.



2. Use the drop-down menus to locate the project folder.
3. In the File name text box, type **MTP1**.
4. Click **Save**. The file is saved onto your computer hard drive and the main screen of EasyBuilder reappears.

### ► Compiling your first project

1. From the Tools menu, click **Compile**. The Compiling dialog box appears.



2. Click **Compile**. EasyBuilder will compile your project and display error results.
3. If no errors occur, click **Close**. The main screen of EasyBuilder reappears. If errors have occurred, repeat the steps in the **Creating a Sample Project** section.

► **Downloading your first project**

It is necessary to determine the IP Address of the OIT. By default, the OIT is configured to get a dynamic IP address from a DHCP server. This setting can be changed, To determine or change the OIT's IP address:

1. Plug a USB mouse into the OIT.
2. Apply power to the OIT.
3. Move the mouse pointer to the lower right corner of the OIT screen. A menu bar will appear, Click the wrench icon on the left side of the bar.
4. On the Network tab of the System Settings dialog, the OIT's IP address will be shown. If set for *Auto Get IP Address*, the OIT's current IP setup information is displayed but is not changeable. To change the IP address, select *IP Address Get From Below* and enter the data.
5. Once the IP Address has been established, the project can be downloaded. From EasyBuilder's Tools menu, select **Download**. Or, click the **Download** icon.
6. Enter the OIT's *IP address* and *Download password*. The default download password is **111111**. The default address is 192.168.0.201, Subnet 255...0, Gateway 192.168.0.254.
7. Make sure that the **Firmware** and **Reboot HMI After Download** boxes are checked.
8. Click the **Download** button. If the OIT is properly connected to the PC, the status box on the download dialog will update with messages indicating the progress of the transfer. The OIT itself will also display messages reflecting its status.
9. When the download is complete, the OIT will restart. Click the **Exit** button on the Download dialog.

**CONGRATULATIONS.** You have completed your first EZware-5000 project.

**Please Use This Page For Your Notes:**

## Chapter 5 - Simulator Mode

As you saw from creating the sample project in the last chapter, downloading any changes you make to the OIT can take time. To decrease the amount of time required to download a project to the OIT, uncheck the Firmware option on EasyBuilder's download dialog.

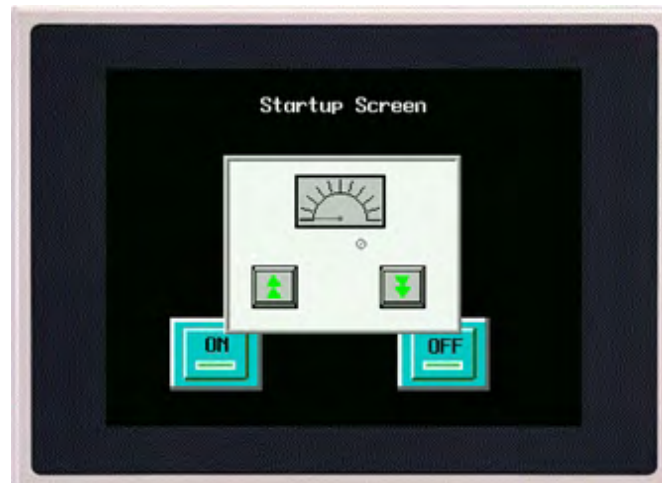
A better way of testing changes made to a project is to use the computer to simulate the operation of the OIT.

This chapter shows you how to dramatically save time when creating and testing a project by putting the computer into Simulation Mode. In Simulation Mode, the computer displays a screen that simulates what is seen when the OIT executes the project file. You can use the mouse on your computer to simulate the operation of the touchscreen on the OIT. Rather than waiting for the download to test each change you make to your project, you can now instantly switch to Simulation Mode to see if your project is working as expected.

You can enter Simulation Mode in three ways. Using the sample project that you created in the last chapter, we will guide you through the three methods.

### The Simulation Screen

All modes of simulation use the same screen.



For our example, we are showing the sample project as it would appear after the ON function key has been pressed. When the Simulation Mode screen first appears, it will always show the startup screen of the project. You can test any objects you have created which require touchscreen input by clicking on them. In this example, clicking on the increment key simulates the key as it is pressed on the touchscreen. Simulation Mode always uses the compiled version of the project that you have created (the \*.xob file). Therefore, you must save and compile any changes you make to the project before you can simulate it.

Simulating the OIT is done in on-line or off-line mode.

On-line simulation is most useful when a project is near completion and you need to test the interaction of the project with the PLC. It is also very convenient when making minor changes to a project because these changes can be instantly checked for operation.

Off-line simulation does not require any connection to the OIT. Because of this, off-line simulation is most often used when starting a project. You can quickly test new ideas and create preliminary screens without the OIT. Off-line simulation is also great for demonstrating the operation of the OIT and becoming familiar with the operation of the OIT before it is installed. If you want to simulate the interaction with a PLC when off-line, it can be done by creating additional windows on the OIT which allow PLC data input.



#### ► To use off-line simulation mode from EasyBuilder:

1. If the project has been modified since the last time it was saved and compiled, or if it has not yet been compiled, it must be saved and compiled before the simulator can start.



2. From the Tools menu, click **Off-line Simulation** or click the Off-line Simulation icon in the Standard toolbar.
3. The Simulation Mode screen will appear.
4. To end off-line simulation, right click in the simulation screen and click **Exit**.



► **To use on-line simulation mode from EasyBuilder**

1. If the project has been modified since the last time it was saved and compiled, or if it has not yet been compiled, it must be saved and compiled before the simulator can start.
2. From the Tools menu, click **On-line Simulation** or click the On-Line Simulation icon in the Standard toolbar.
3. The Simulation Mode screen will appear.
4. To end on-line simulation, right click in the simulation screen and click **Exit**.

Simulation mode can also be started from Project Manager. Project Manager requires only the compiled project file (\*.xob). This can be useful in situations in which you want to demonstrate changes made to a project for a client without giving him access to your project file. It also allows you to rapidly switch from simulating one project to another without having to open, save, and compile each project.

► **To use off-line simulation mode from Project Manager**

1. Start the Project Manager software.
2. Click **Offline-Simulator**. The Open Project box appears.
3. Click on the compiled project file that you wish to simulate.
4. Click **Open**. The simulation screen will appear.
5. To end off-line simulation, right click in the simulation screen and click **Exit**.

► **To use on-line simulation mode from Project Manager**

1. Start the Project Manager software.
2. Click the **Online-Simulator**. The Open Project dialog box appears.
3. Click on the compiled project file that you wish to simulate.
4. Click **Open**. The simulation screen appears.
5. To end on-line simulation, right click in the simulation screen and click **Exit**.

Online simulation requires a direct connection between the PC and the controller. It is not necessary to have the OIT connected.

The OIT's COM1 RS232 connection is wired the same as a standard PC 9-pin serial connector. Therefore, the standard Maple Systems' COM1 RS232 cable for the selected PLC/controller can be used with online simulation. Connect the cable end marked "HMI" to the PC's serial port, and connect the other end to the PLC/controller.



*The above method will work with a standard PC serial COM port, regardless of the COM number assigned to the PC serial port. EasyBuilder supports RS232 communication on ports COM1-COM9. COM3 and higher are intended for use with online simulation, since the OIT itself supports RS232 communication on COM1 and COM2 only.*



*If simulating with RS422 or RS485 communication, the PC will require some mechanism to support RS422/485. A communication card can be installed in the PC, a USB communications module can be used, or an external RS232/RS485 converter can be used. In any case, a standard Maple Systems cable may not work.*

If simulating a project using Ethernet communications, an Ethernet cable between the PC and the PLC/controller is needed. Consult the PLC/controller documentation for details.

Now that you are familiar with using simulation mode, the next chapter guides you through the fundamental operation of three primary segments of the EZware-5000 configuration software.

---

## Chapter 6 - Using EZware-5000

### Overview

The EZware-5000 software is composed of three separate applications which are accessible from the EZware-5000 folder: EasyBuilder, Project Manager, and EasyConverter. EasyBuilder is the application software used to create a project file. Project Manager is a utility application that puts the OIT into different operating modes. EasyConverter changes certain binary files created by the OIT into text files.

By the end of this chapter, you should be quite familiar with the operation of Project Manager and be able to maneuver around EasyBuilder easily. This will pave the way for actually creating graphics objects in later chapters.



*EasyBuilder 5000 includes extensive online help. From any dialog box within the software, click on the **HELP** button, or press the computer's **F1** key.*

---

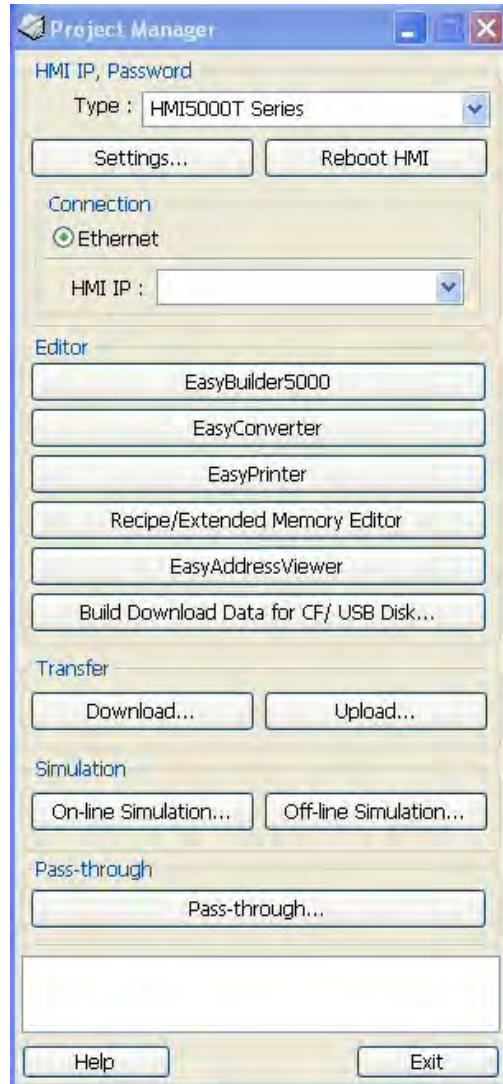
### The Project Manager

The Project Manager utility sets the communications parameters that are used by the computer when communicating to the OIT. The utility can also be used to:

- Start EasyBuilder
- Write project data to a CompactFlash or USB device for transfer to an OIT
- Download project data to the OIT or upload project data from the OIT
- Start a project simulation session.

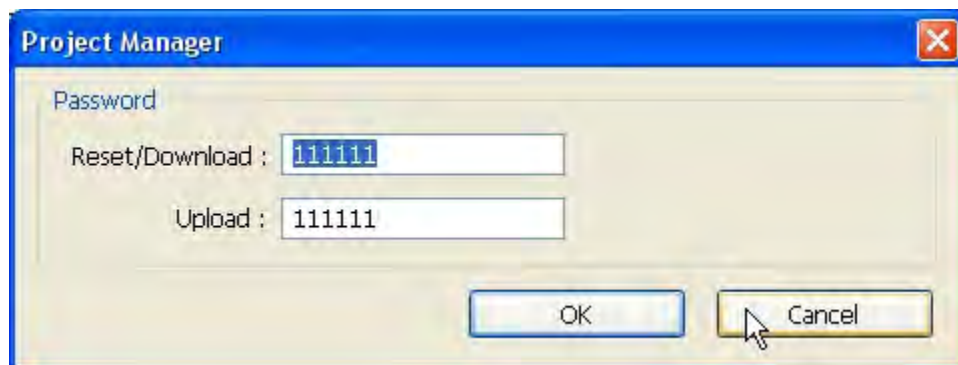
To access Project Manager, click on the Windows Start button and select **All Programs-Maple Systems-EZware-5000-Project Manager**.

The following illustration shows the various commands available:



## ***Communications & Password Settings***

### **Settings Button**



1. **Download/ Upload** passwords (default 111111)

## Reboot HMI

Connects to the OIT, checks password and resets the OIT in order to update system files.

## Editor Settings

### EasyBuilder-5000

Brings up EasyBuilder-5000 editing software for project creation and editing.

### Build Download Data for CF/USB Disk

Select the folder to save download data :

PLEASE INPUT DIRECTORY NAME !

Sources

Project  
PLEASE INPUT XOB FILE NAME !

Recipe (RW)  
PLEASE INPUT RECIPE FILE NAME !

Recipe A (RW\_A)  
PLEASE INPUT RECIPE\_A FILE NAME !

Data log  
PLEASE INPUT DATA LOG FILE NAME !

Data can be downloaded to the Silver Series OITs using both CompactFlash and USB jump drives. The Build Download Data function builds the data to do this.

#### To build download data for CompactFlash or USB jump drive:

1. Insert CF card or USB jump drive to OIT.
2. Start Project Manager.
3. Click **Build Download Data for CF/USB Disk...** The Project Manager dialog appears.
4. Insert CF card or USB jump drive into PC. Note the folder that represents the USB/CF device.
5. Click **Browse...** to select the *folder* that represents the USB/CF device.
6. Check the boxes that will be the sources for your data and browse to the appropriate file for that source.
7. Click **Build**.

## Easy Converter Application

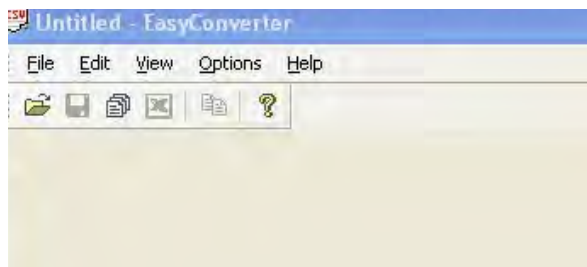
The EasyConverter application is used to view data log files (\*.dtl) and event log files (\*.evt) that are created by the OIT during operation. Data log files are files the OIT creates when capturing data using the Data Sampling object. Event log files are history files the OIT creates when using the Event Log object. Both files are in binary format to minimize the amount of file space required.

To view these files with a computer, you must use the EasyConverter application. In addition, EasyConverter can take a data or event log file and convert it into a CSV (Comma Separated Value) file that can be easily imported into other applications such as Microsoft Excel. For more information on how to do data logging, see Chapter 12 “Bar

Graphs, Meters, and Trends”, section “Creating Trend Displays & Data Sampling Objects”. For more information on how to use Event Log files, see Chapter 13 “Capturing Alarms and Events”, section “Monitoring Alarms with the Alarm/Event Log”.

## To start the EasyConverter Application

1. From the Tools menu, click **Data/Event Log Converter**. Or click the Data/Event Log Converter icon in the Project toolbar. You can also access the EasyConverter application from the Project Manager. The EasyConverter application appears:



2. Click **File...Open** to display the Open dialog box. Search for the data log file (\*.dtl) or event log file (\*.evt) that you wish to view, then click the Open button.

### *Please Note:*

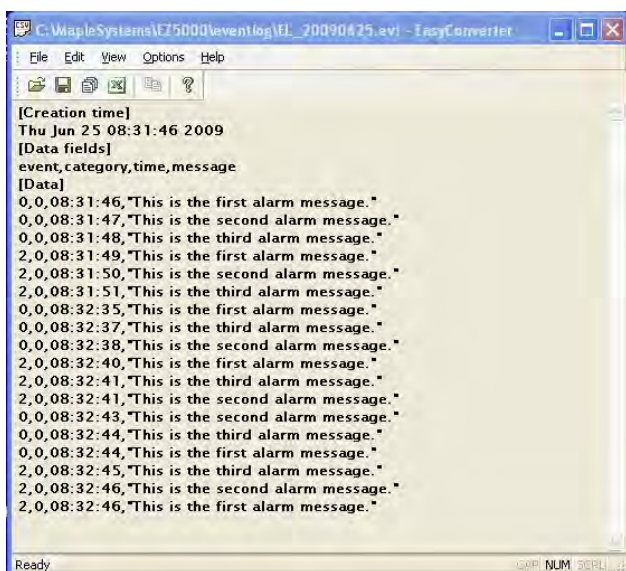
*These files are captured by the OIT and stored in internal OIT memory, an attached SD card or USB flash drive. You must copy these files into your computer before using the EasyConverter utility. If you are running a project that is capturing data log or event log files using the Offline Simulator, the EZware-5000 application automatically creates new folders on your computer and creates the files.*

*Data log files are stored in a folder using the name that you assigned in the Data Sampling Object attribute box (ex: C:\MapleSystems\EZ5000\datalog\History Data1\20090618.dtl).*

*Event log files are stored in an eventlog folder (ex: C:\MapleSystems\EZ5000\eventlog\EL\_20090625.evt).*

## Viewing an Event Log File (\*.evt) using EasyConverter Application

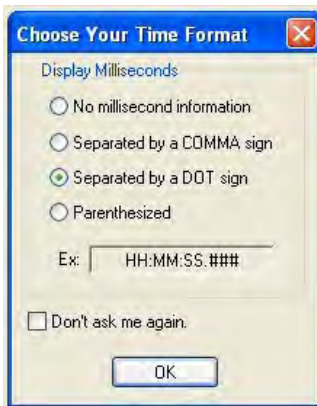
1. From the EasyConverter application, click File...Open and locate the Event Log file (\*.evt):



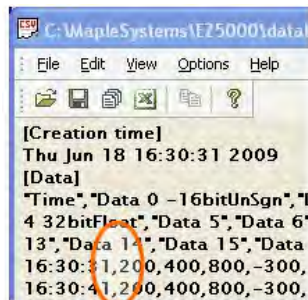
2. **[Creation time]**: the text line below the creation time heading shows the date and time that the file was initially created. Format is Day of Week, Month, Day of Month, Time (24Hr format), and Year.
3. **[Data fields]**: the text line below this header is used to label and identify each column of data under the [Data] header. The Data fields header shows the event number, category of event, time event occurred in 24HR format, and the event (alarm) message recorded.
4. **[Data]**: this is where the actual data is recorded. Note that this is read only- the EasyConverter application does not allow any edits.

## Viewing a Data Log File (\*.dtl) using EasyConverter Application

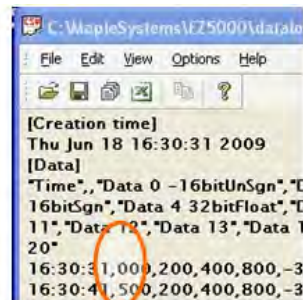
1. From the EasyConverter application, click **File...Open** and locate the Data Log file (\*.dtl). As the file is loaded, the following popup window appears:



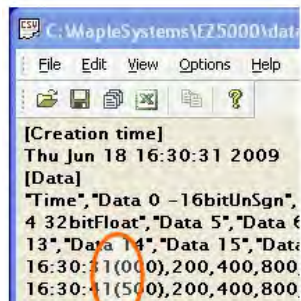
2. **Display Milliseconds:** This option determines how the time value for each data record is displayed. In this window you have four options:



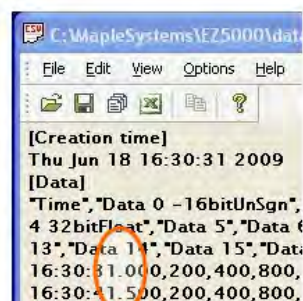
No Millisecond Option



Separated by Comma Option

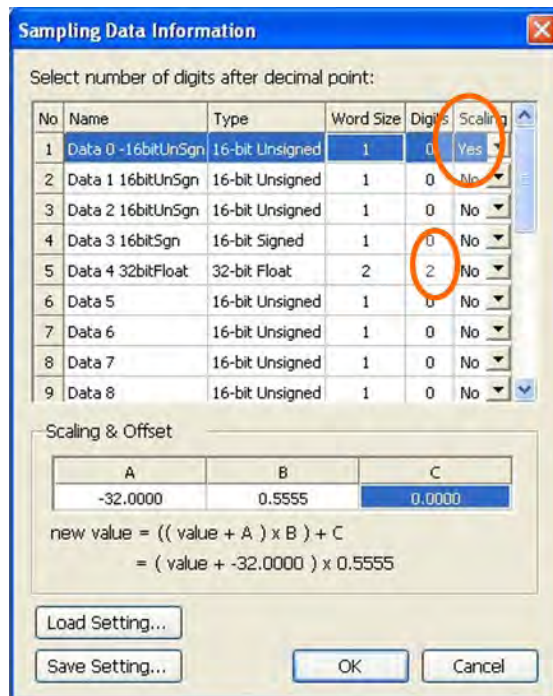


Parenthesized Option



Separated by Dot Option

3. **Decimal display and Scaling option:** The dialog box below allows you to select different formats for the data before it is displayed:

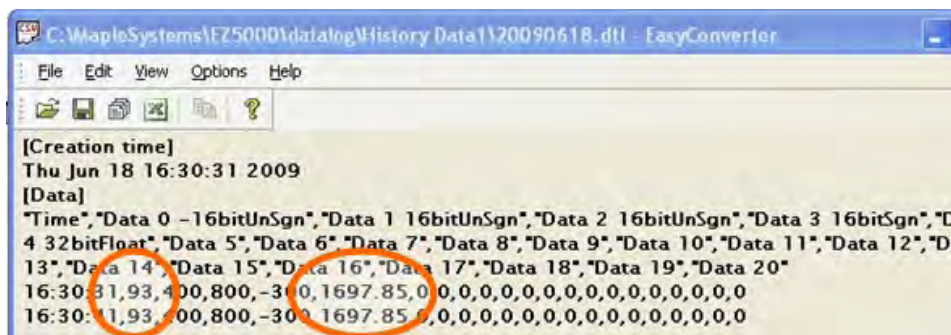


4. For this example, we have elected to use the linear scaling formula on Data 0 to convert the raw data (which is measuring temperature in °F) to °C.

Also, the raw data collected for Data 4 is a floating point number calculated to three decimal places. However, we are only concerned with a two decimal accuracy, so we change the 'Digits' field from 3 to 2. If we want to use these same changes on other files, **click the Save Setting...** button to save the changes as a conversion (\*.lgs) file. When opening other data log files, we can apply the same changes by clicking on the Load Setting button and selecting the proper conversion file.

**Click OK** to continue.

5. The data log file is shown:



In our example, notice that the value for Data 0 {200°F} has changed to 93°C and the value for Data 4 has two decimal places {1697.85} instead of three {1697.850}.

6. **[Creation time]:** the text line below the creation time heading shows the date and time that the file was initially created. Format is Day of Week, Month, Day of Month, Time in 24Hr format and Year.

7. **[Data]:** the text line below this header is used to label and identify each column of data recorded. The header shows the time each record was captured followed by the title given for each data column.

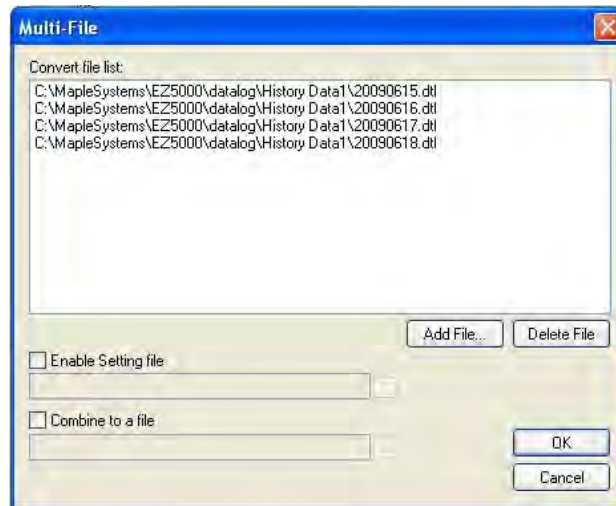
*Please Note: this is read only-- the EasyConverter application does not allow any edits.*



## Converting a Data /Event Log File to a CSV file using EasyConverter Application

After opening a Data/Event Log file into EasyConverter, you can easily export the file into a CSV file. Note: Microsoft Excel is required to convert the file.

1. Click **File...Export** to Excel or click on the Excel icon
2. Microsoft Excel is activated and you should see the data in a spreadsheet. Simply click **File...Save As** to save the file.
3. Converting multiple files: EasyConverter also has a feature that allows you to select multiple Data/Event Log files and save them to one CSV file. To do this click **File...Multi-File** or **click on the Multi-File icon**.

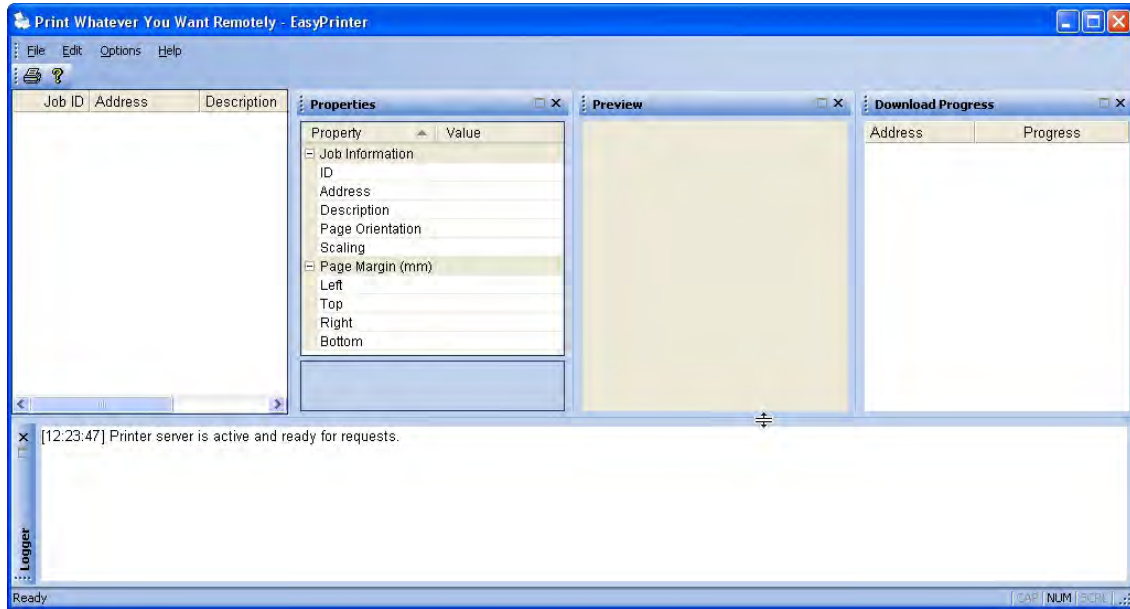


4. Click **Add File...**, then select which Data/Event Log files you wish to combine.
5. **Enable Setting file**: check this box to select a conversion (\*.lgs) file that performs a linear scaling conversion or decimal point conversion on the selected files.
6. **Combine to a file**: check this box to combine the files and put them into an Excel spreadsheet.

**Please Use This Page For Your Notes:**

## Easy Printer

The Easy Printer button allows monitoring of the print server through Project Manager. The EasyPrinter utility must be running on the PC that is hosting the networked printer from which you wish the Silver Series OIT to print.

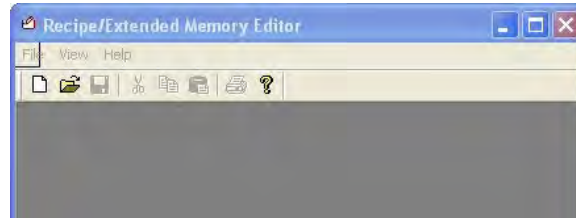


## Recipe/Extended Memory Editor

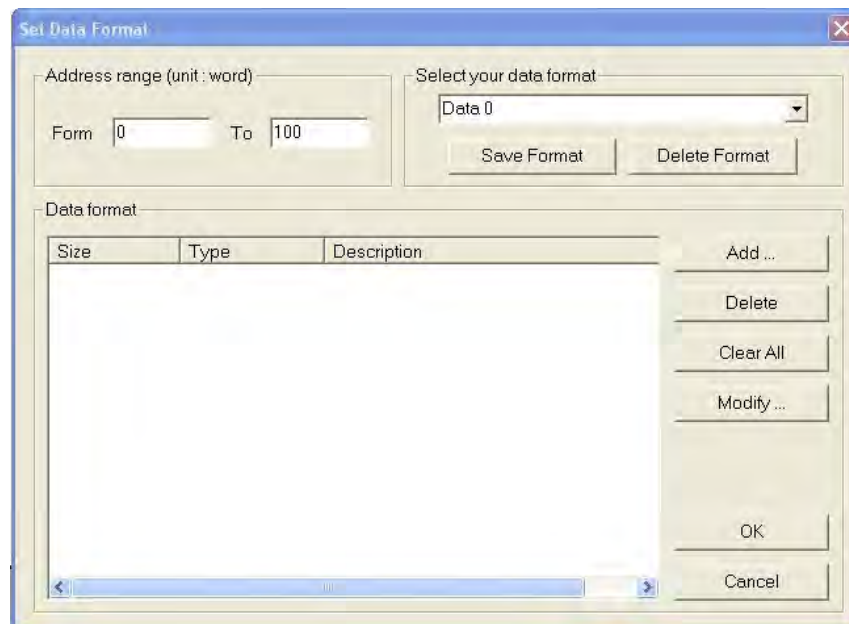
The Recipe/Extended Memory Editor is an application that allows you to create, view, and modify recipe (\*.rcp) files and Extended Memory (\*.emi) files. You can also use this editor to import/export recipe and extended memory files to CSV formatted files which can then be shared by other applications.

To create a new Recipe/Extended Memory file:

From the Project Manager application, click **Recipe/Extended Memory Editor**:



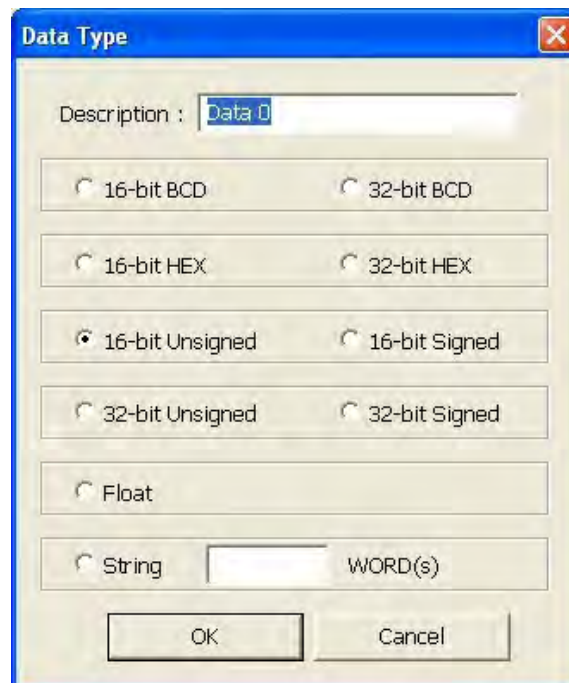
1. Click **File . . . New**. The following dialog box appears:



2. Address range (unit:word) frame -- use this frame to select the total number of RW (recipe word) registers or EMx (Extended Memory) registers that are used for the recipe or extended memory file. In the example above, RW0 thru RW100 will be used.

3. Data format frame- use this area to enter how each data register element is be interpreted by the OIT. The data collected and stored in recipe/extended memory files is in binary format and does not contain any information about how each data should be interpreted by the OIT (for example, is the data in 16-bit signed integer format or BCD?). Therefore, the data format is used to define how each data element is formatted (ie. Signed vs Unsigned, 16-bit vs 32-bit, Float, String, etc.).

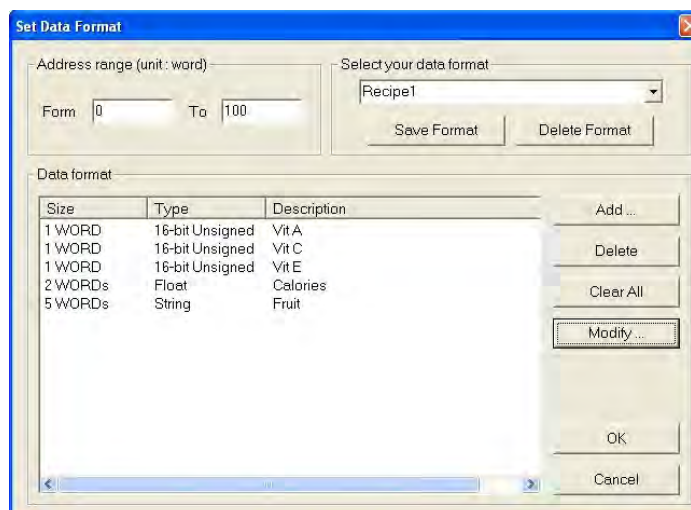
- a. Add: Click to add a new data register format. The following dialog box appears:



In the Data Type Dialog Box you can enter a description of the Data Type or Preset Value. You can also select which format that the OIT will use to interpret the data.

**Please Note:** For String format, you must enter the total number of 16-bit register words used for the character string.

- b. Delete: If you wish to delete an item from the list, highlight a data element and click delete to remove from list.
- c. Clear All: click to erase entire list of data elements.
- d. Modify: highlight a data element and click Modify to revise the name of the data element or format.
- e. Below is an example of a list of data elements created:



4. Select your data format frame- use to select or create a data format template that can be used for other recipes or extended memory files. In the example above, the data elements configured in the 'Data format' frame have been saved to a template called 'Recipe1'. Click OK.

5. A table of data elements appears that shows the recipe or extended memory table:

ID	ADDRESS	Vit A	Vit C	Vit E	Calories	Fruit
0	0	0	0	0	0	
1	10	0	0	0	0	
2	20	0	0	0	0	
3	30	0	0	0	0	
4	40	0	0	0	0	
5	50	0	0	0	0	
6	60	0	0	0	0	
7	70	0	0	0	0	
8	80	0	0	0	0	
9	90	0	0	0	0	
10	100	0	0	0	0	

This table has several columns with headers:

- a. ID: this is the ID number for each record.
- b. Address: this is the starting RW address or EMx address for each record.
- c. Data headers: these are the descriptions used for each data element created in the step above.

6. Edit the table by clicking on each data element field and entering the preset value. Please note that you are only allowed to enter data according to the format defined for each data element.

Here is an example of a completed table:

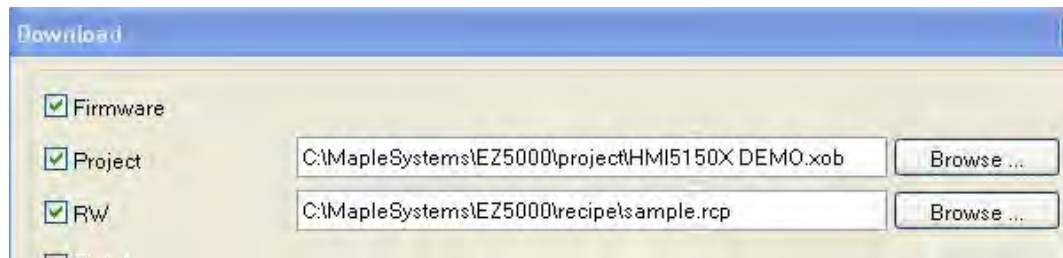
ID	ADDRESS	Vit A	Vit C	Vit E	Calories	Fruit
0	0	10	23	13	46.5	APPLES
1	10	25	67	37	14.7	ORANGES
2	20	12	434	5	67.9	PEARS
3	30	123	4	15	13.5	CHERRIES
4	40	12	155	43	34.7	GRAPES
5	50	76	23	42	23.5	PINEAPPLE
6	60	5	32	75	98.9	MELON
7	70	5	43	83	19.24	KIWI
8	80	100	90	24	598	PEACHES
9	90	7	83	64	12.34	BANANAS
10	100	34	77	102	8.64	GRAPEFRUIT

7. Click File...Save with the following options:

- a. Save file as \*.rcp for recipes.
- b. Save file as \*.emi for extended memory files. Note: You must use file names which correspond to the ten extended memory areas preconfigured in the OIT (ex. em0.emi, em1.emi, ..., em9.emi).
- c. Save file as \*.csv to import file into applications such as Microsoft Excel.

8. Once the file is saved, you can copy the recipe file to your OIT by:

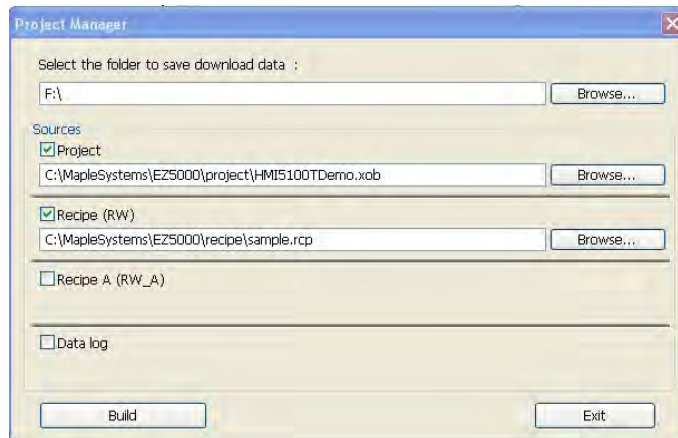
- a. Procedure 1-- Downloading the file using the Project Manager:



- i. With the computer connected to your OIT, click Transfer...Download.
- ii. On the Download dialog box, check the RW enable box, then select the recipe file.

- b. Procedure 2 -- Downloading using a SD card or USB Flash Drive:

- i. Attach an SD card or USB Flash Drive to your computer.
- ii. In Project Manager, click Build Download Data for CF/USB Disk button:



- iii. Select the folder to save download data: Enter the SD card or USB Flash drive location.
- iv. Check the Recipe (RW) enable box, then select the recipe file.
- v. Click the Build button. A folder called 'mt8000' is created on the root directory of your drive.
- vi. Remove the SD or USB drive and connect it to your OIT.

**Please Note:** if you are using more than one USB device, make sure that the USB Flash drive is connected first (this identifies the USB Flash Drive as USB1).

## ***Transfer Settings***

### **Download**

Sends a compiled project file stored on the computer to the OIT.

### **Upload**

Receives a compiled project file from the OIT to be stored on the computer.

For more information about printing, see the Systems Configuration Settings/Printer tab section later in this chapter.

## ***Simulation Settings***

### **On-line Simulation**

Opens a compiled project (\*.xob) in on-line simulator mode.

### **Off-line Simulation**

Opens a compiled project (\*.xob) in off-line simulator mode.



## Pass-through mode

The pass-through mode allows serial communications from the PC to the PLC, though the OIT. This can eliminate the need to disconnect the PLC from the OIT when it is necessary to configure the OIT.

### To use the pass-through mode:

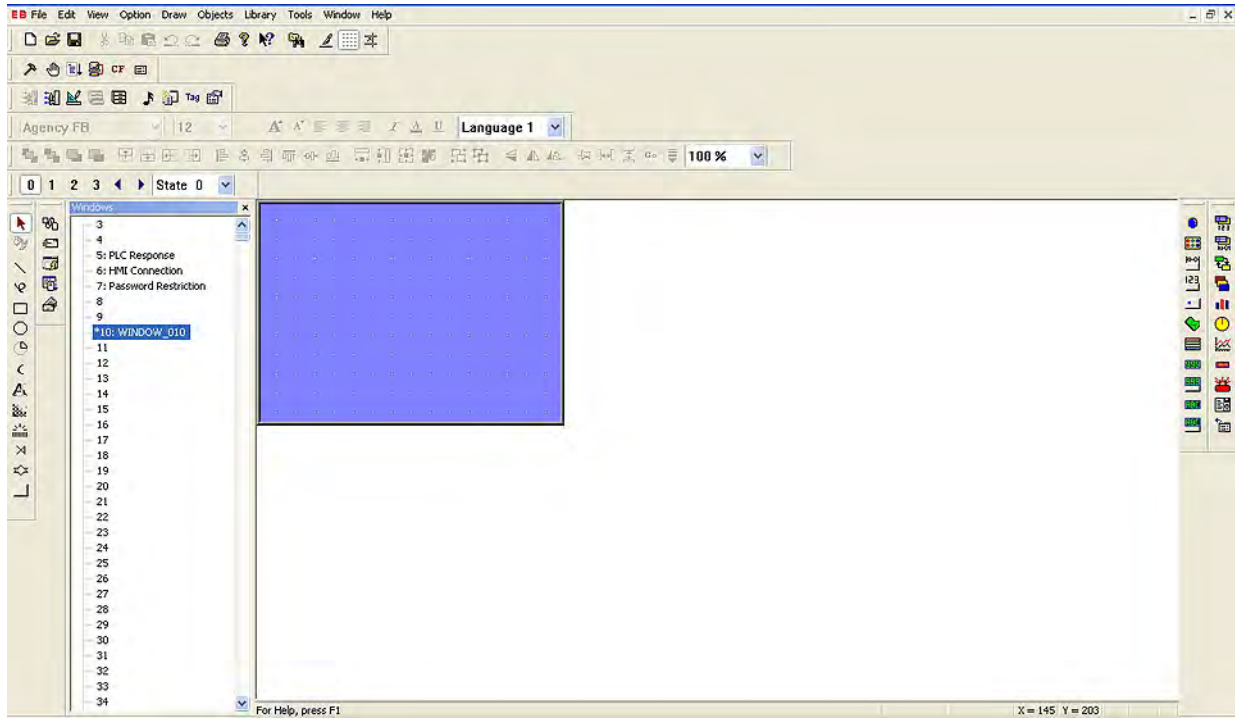
1. HMI/IP: Specify the **IP Address** of the OIT that is to be used in passthrough mode or click **Get HMI Communication Parameters** to read the current communication settings for the OIT.
2. HMI work mode: Indicates the current mode of the OIT. After the communication parameters have been received, the mode will be either Normal (OIT set to communicate with PLC) or Passthrough (OIT is in pass-through mode).
3. Source COM Port: Set Com settings for the OIT that will be connected to the PC.
4. Destination COM Port: Set COM settings for the OIT that will be connected to the PLC.
5. Click **Start Pass-through** to begin passthrough and **Stop Pass-through** to end. Click **Exit** to exit pass-through mode.

That's all there is to Project Manager.

## The EasyBuilder Application

This section guides you in how to operate the EasyBuilder application; however, it does not show you how to program your OIT or how to create graphics objects. These topics are reserved for later chapters. This section shows the fundamental operation of EasyBuilder from saving files, printing projects, and selecting the target PLC to showing how graphics objects can be easily manipulated in the EasyBuilder work area. When you have completed this chapter, you will be better able to use the features that are explained in later chapters.

The following illustration is used for reference to the following sections:



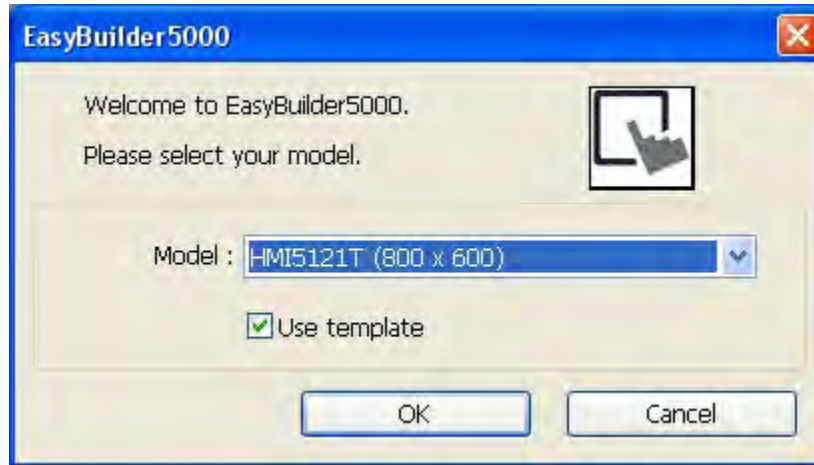
## Managing Projects

Like most Windows™ application software, EasyBuilder will open, save, close, and print files using the standard windows format.

### Opening, Editing Projects

To create a new project 

1. On the File Menu, click **New** or click the **New** icon in the Standard toolbar. The EasyBuilder dialog box appears.

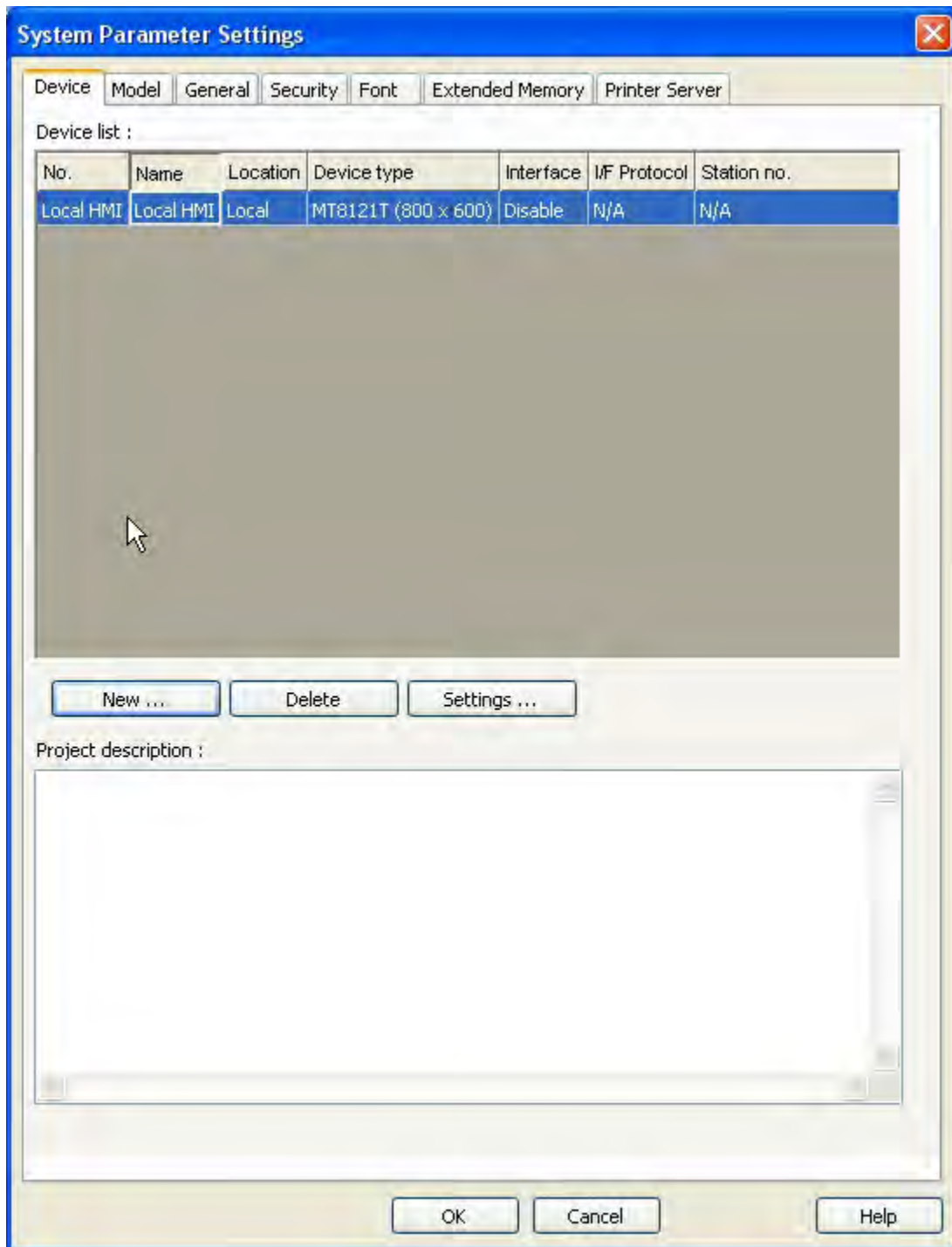


2. Select the *OIT model* you intend to use with your project.



*A project's display mode may be changed after the initial selection has been made. To set Display Mode select **Edit-System Parameters...**, and select the **Model** tab. To change the Display Mode, resize the window's resolution to fit the mode selected.*

3. Click **OK**. The System Parameter Settings dialog appears.



- This is where you will select your PLC or controller. Click **New...** to bring up the Device Properties dialog.

**Device Properties**

Name : Device 1

HMI  PLC

Location : Local Settings ...

PLC type : MITSUBISHI FX0n/FX2  
V.1.00, MITSUBISHI\_FX0N.so

PLC I/F : RS-485 4W PLC default station no. : 0

COM : COM1 (9600,E,7,1) Settings ...

Use broadcast command

Interval of block pack (words) : 5

Max. read-command size (words) : 32 Max. write-command size (words) : 32

OK Cancel

- Select the PLC option button and then select the *PLC type* from the drop down menu. Configure any necessary COM ports and click **Settings...** to bring up the COM Port Settings dialog.

**COM Port Settings**

COM : COM 1

Baud rate : 9600

Data bits : 7 Bits

Parity : Even

Stop bits : 1 Bit

Timeout (sec) : 3.0

Turn around delay (ms) : 0

Send ACK delay (ms) : 0

Parameter 1 : 0

Parameter 2 : 0

Parameter 3 : 0

OK Cancel

6. Configure the *appropriate COM port settings* for your PLC (for more information, see the *Controller Information Sheet* for your PLC) and click **OK**.
7. Adjust *Interval of Block pack* settings, if necessary. This can be adjusted later by selecting System Parameters-Settings.
8. Click **OK**. You are now ready to begin working in EZware-5000 to create your EasyBuilder project.

► **To open an existing project**



1. On the File Menu, click **Open** or click the **Open** icon in the Standard toolbar. The Open dialog box appears.
2. Click on the project file you intend to open.
3. Click **Open**. The main screen of EasyBuilder appears with the initial screen of the project displayed.

► **To close a project**

1. On the File Menu, click **Close**.
2. If changes have been made to the project file, EasyBuilder will ask you if you would like to save the project. Then the main screen of EasyBuilder will remain but with no work area displayed. You must now use the Open or New commands to edit a project.

► **To save an existing project**

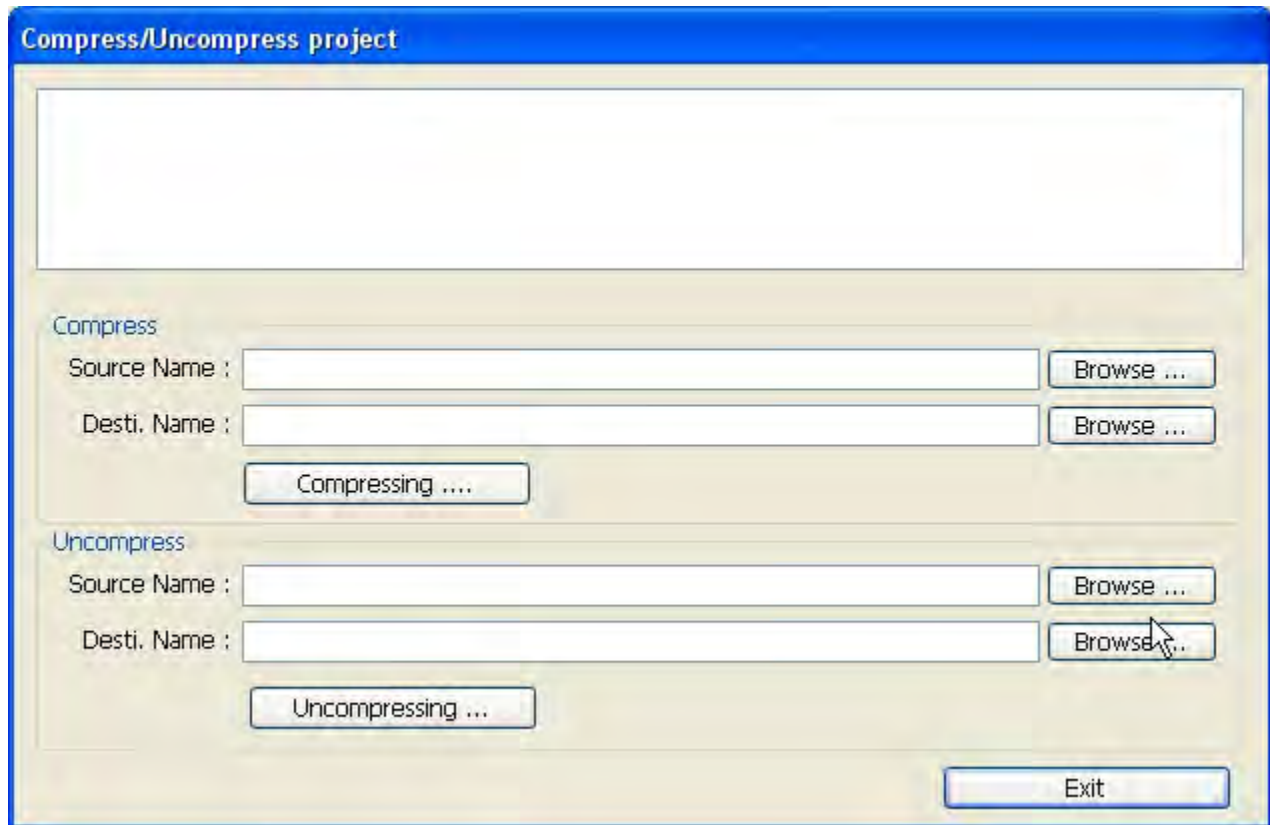


1. On the File Menu, click **Save** or click the **Save** icon in the Standard toolbar.
2. If the project already has a name, then the project will automatically be saved. If this is a new project, then the **Save As** dialog box appears.
3. Enter a file name and then click **Save**.
4. The main screen of EasyBuilder reappears.

► **To save a project using the compress feature**

The compress feature allows you to save a project in compressed format so that the project data takes less space on your hard drive. This utility also will save the graphics libraries associated with the project into the project file. This facilitates sending a copy of the project to another person who has EZware in order to download the project into an OIT.

1. From the Tools menu, select **Compress/Uncompress....** The Compressing dialog box appears.



2. In the **Compress** frame box under **Source Name**, select the project (\*.mtp) file that you wish to compress. File must have a .mtp extension.
3. Click **Compress...** to begin. The utility will compress the project file and all related graphics libraries into a single file.

► **To extract a compressed project file**

1. From the Tools menu, select **Compress/Uncompress....** The Compressing dialog box appears.
2. In the **Uncompress** frame box under **Source Name**, select the project (\*.cmp) file that you wish to uncompress.
3. Click **Uncompress...** to begin. The utility will extract the project file and all graphics libraries from the \*.cmp file.

► **To exit EasyBuilder**

On the File Menu, click **Exit** -- or click on the standard windows **Close** icon in the upper right corner.

## **Editing and Creating Screen Objects**

This section shows how to manipulate graphics objects that are placed onto the work area of EasyBuilder. We will use examples from sample projects that are included with the EasyBuilder software: HMI5043.MTP, HMI5057.MTP, HMI5070.MTP, HMI5080.MTP, HMI5100.MTP, HMI5104.MTP, HMI5121.MTP and HMI5150.MTP. Please load one of these project files and have EasyBuilder ready before you begin this section.

We will refer to Window\_12 of the project. To display Window\_12 on EasyBuilder, perform the following steps.

► **To open a window**

1. On the Window Menu, click **Open Window**. The Open Window dialog box appears.
2. Click **Window\_12..**

3. Click **Open**. Window\_12 appears in the work area of EasyBuilder.



## Display Options

Before we begin describing some of the commands that can be used to edit graphics objects, there are a few commands that apply to the general work area of EasyBuilder.

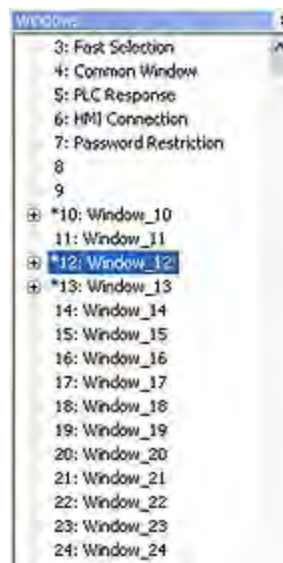
EasyBuilder provides a **Window Tree** that can be used to easily maneuver between windows of a project. It also allows you to select any object that is on the window that is currently displayed.

### ► To display/hide the Window Tree

1. On the View Menu, click **Window Tree**

### ► Using the Window Tree to display window screens Windows

1. The Window Tree combines into a single tree a list of all of the windows in a project, and all of the objects on each window. Using the 5056DEMO.MPT project, the following should be seen :

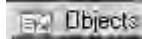


2. Notice that the tree displays the title for each window next to the window number. This makes it easy to determine windows have been created for a project. The asterisk (\*) next to Window #10 indicates that this window is currently open.
3. To display an open window, click on the *window number* in the Window Tree. To display a window that is not yet open, double-click on the *window number*. This will automatically open the window and display it.

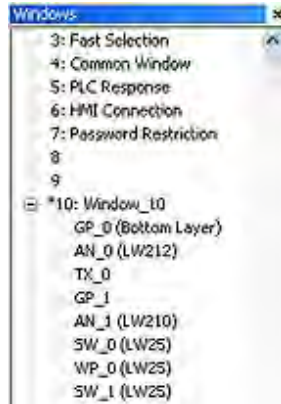


4. To change any settings or close an open window, click on the *window number* to select it. Then right-click anywhere inside the Window Treebar to display a pop-up dialog box. Select **Close** to close the window or **Setting** to change any of the window settings.
5. Finally, to create a new window click on any window number that is not currently used. Then, right-click inside the Window Tree and click **New**.

► **Using the Window Tree to change object attributes**



1. On the Window Tree, click + adjacent to any window number. The tree will expand to list the objects on that window, as shown:



2. The Object ID of each objects will be listed. For example, TX\_0 is Text Display #0, SW\_0 is Set Word #0, etc.
3. To highlight a particular object on the screen, click on the Object ID in the list. This allows the quick selection of a particular object, especially on crowded screens or screens where objects may be overlapping..
4. To display the object's attribute dialog box, double-click the *Object ID* in the Window Tree.

► **Using the grid function**



One option is to have the work area covered with grid lines. These grid markings can be helpful when trying to align objects that are created on the work area. If grid lines are selected, you can decide what size each grid is.

1. On the Option Menu, click **Grid/Snap**.
2. Click the **Display** box or click the **Grid** icon in the Standard toolbar.



*If you still don't see the grid after you have enabled it, make sure that the color of the grid doesn't match the background color of the window currently displayed.*

► **Changing the grid size & color**

1. On the Option Menu, click **Grid/Snap**
2. Select the *X spacing* and *Y spacing* for grid size.



*Spacing is measured in pixels.*

3. Select the *Grid color* in the drop down menu.

Another option is to display what are known as Object ID tags. Every time you create a new object in EasyBuilder, an Object ID tag is assigned to the new object. This is done for several reasons:

- Object ID tags are required by EasyBuilder to differentiate each object created.
- If an error occurs during the compile process, EasyBuilder can refer to the window and object ID to indicate the object that is causing problems.

► **To enable/disable Object ID tags**

1. On the Option Menu, click **Function Property**. The Function Property dialog box appears.



2. Click the **Display Object ID** checkbox to display tags.
3. Click **OK**.

If you enable the Grid on the work area, you may also take advantage of a useful feature called Snap. The Snap option causes all objects placed into the work area to fall along the boundaries set by the grid lines. The Snap option can help provide a more ordered appearance to graphics objects.

► **To use the snap option** 

1. The **Grid** option must be enabled.
2. On the **Option** Menu, click **Grid/Snap**. The Grid/Snap Settings dialog box appears.



3. Click the **Snap** checkbox.
4. Click **OK**.

Finally, you will find that you can move objects by selecting them with the mouse cursor and dragging them to a new location. However, you may find it difficult to select an object *without* accidentally moving it just a little. At times this may be frustrating, so you may wish to disable the move feature using the mouse and only move an object by changing its X and Y position parameters in the Profile tab of the object's attributes box. This is the intent of the Fix Objects feature.

► **To enable the fix objects command**

1. On the Option Menu, click **Grid/Snap**. The Grid/Snap Settings dialog box appears.
2. Click the **Fix Objects** checkbox.
3. Click **OK**. Movement by mouse cursor is now disabled.

► **To zoom in on the selected window** 

1. Use the zoom pull-down menu at the top right of the screen. Zoom levels range from 50% to 200% in 25% increments.

► **To change the state of the selected window**



1. Use the zoom pull-down menu at the top right of the screen.
2. The objects will appear as they would in the selected state.

## Basic Editing Commands

► **To select a graphics object**



1. On the Edit Menu, click **Select** or click the mouse cursor icon in the Draw toolbar.
2. Click on the graphics object. For example, using Window\_12, click on **Fill Level**. This causes the text box to be selected, with small square blocks around the edges indicating the boundaries of the object. Deselect the object by clicking somewhere else in the work area.



3. When a graphics object is selected, it can then be modified, copied, deleted, or moved to a new location.

► **To select multiple graphics objects**

1. On the Edit Menu, click **Select** or click the mouse cursor icon in the Draw toolbar.
2. Click and hold down the left mouse button at the upper left corner of the graphics objects you wish to highlight. For example, using Window\_12, click to the left and above of **Motor Speed**.
3. Move the mouse cursor to the lower right corner of the highlighted objects. Notice that a rectangle is formed as you do this. In this example, move the mouse to the right and below **Fill Level**.
4. Release the mouse button. The rectangle outline changes to small clear square blocks around the perimeter of the objects selected. Now both objects are selected.



*If you do not completely enclose any graphics objects you wish to highlight, then they will not be selected.*

5. Alternatively, you may select objects by holding down the CTRL key while selecting each object until all objects are selected.
6. When several graphics objects are selected, they can easily be moved, deleted or copied together.

► **To select next object**



1. This option can be used to easily select an object that is underneath another object on the screen. To use this option, you must first select the overlapping object.
2. On the Edit Menu, click **Select Next Object** or click on the **Select Next Object** icon on the Standard toolbar. You can also right-click on the top-most object on the screen. This will display a menu with a list of the overlapping objects on the bottom. Then check the object that you wish to access.

- Once the object you are trying to access is selected, you can proceed to modify its attributes or delete it.

► **To select all objects**

- On the Edit Menu, click **Select All Objects**.
- Small, clear, square blocks appear around the perimeter of all the objects on the window.
- You can now easily move, delete or copy the entire window screen to a new window.

► **Using the Undo  and Redo  commands**

- The Undo command is used to cancel the last command or action that you made. For example, select the **Motor Speed** text box in Window\_12.
- Press the **DELETE** key on your keyboard to delete the text box.
- From the Edit menu, click **Undo** or click the Undo icon from the Standard toolbar. You can also press CTRL+Z. The deleted text box reappears.
- The Redo command is used to cancel the Undo command. For example, if you decided that you really did want the **Motor Speed** text box deleted, click the **Redo** command.

► **Using the Cut , Copy , and Paste  commands**

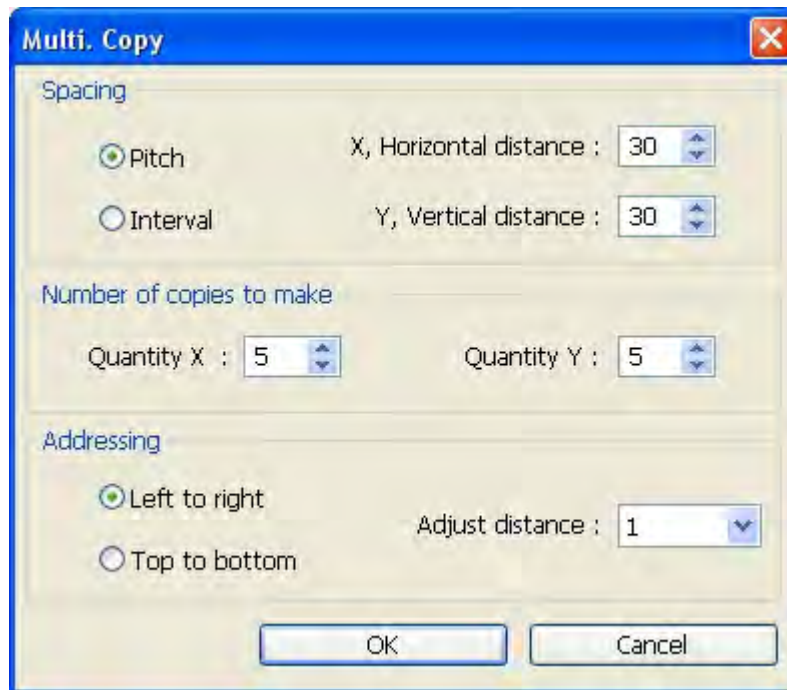
- These commands are all selected from the **Edit** menu or by clicking the appropriate icon in the Standard toolbar.
- Select the graphics object or objects you wish to cut or copy.
- Click **Cut** to copy and remove the graphics object(s) from the work area or click **Copy** to copy the graphics object(s). Using Window\_12, select the **Motor Speed** text box and then press CTRL+X to cut the object from the work area.
- Objects cut or copied from one window can be pasted into other windows. Once the object has been selected and cut or copied, open another window, then paste the object into that window.
- In this example, paste the Motor Speed text box back into Window\_12 by pressing CTRL+V. The pasted object appears highlighted in the upper leftmost corner of the work area.
- Move the text box by clicking on the highlighted object and dragging it back to the original position.

► **Using the Multi-Copy command**

The Multi-Copy command is used to make multiple copies of a single object on a screen. This is most useful when creating new keyboards or data tables. In addition to copying the object, this command will automatically assign new PLC memory addresses to each object, (applies only to Bit-type objects).

- Select the *object* you wish to copy.

- From the Edit menu, select **Multi. Copy...**. The Multi. Copy dialog box appears.



- Modify the settings according to your requirements.
- Click **OK**. The main screen of EasyBuilder is redisplayed with the Multi. Copy command executed.

Multiple objects can be created according to these parameters:

**Pitch vs. Interval:** This setting affects how EasyBuilder places the copies of an object on the screen relative to each other. The Pitch setting will interpret the X Distance setting as the distance from the left side of an object to the left side of the next object and the Y Distance setting as the distance from the top of one object to the top of another object.

The Interval setting will interpret the X Distance setting as the distance from the right side of one object to the left side of the next object. The Y Distance setting is the distance from the bottom of one object to the top of the next object.

**X Distance, Y Distance:** These settings determine the spacing (in pixels) between objects.

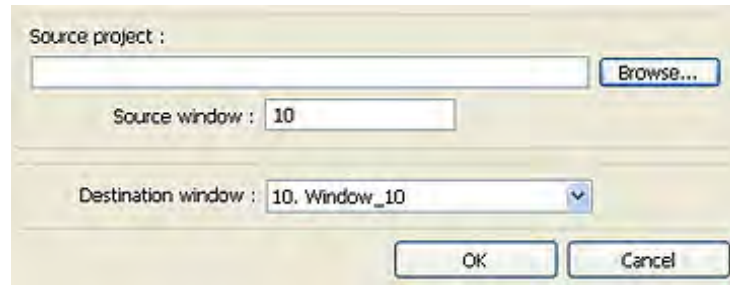
**Quantity X, Quantity Y:** These settings determine the number of copies to be made along the X axis (horizontal) and the Y axis (vertical).

**Adjust Distance:** This is the offset used when assigning a new PLC memory coil address to each new copy. For example, if 1 is selected, then EasyBuilder will add one to the PLC memory coil address for each new copy made.

#### ► Using the Window Copy command

The Window Copy command is used to copy a window from an existing project into the project that you are currently editing. This feature reduces time spent creating new projects since you can now use windows that you created for other similar projects.

1. From the Edit menu, select **Window Copy...** The Window Copy dialog box appears.



2. In the **Source Project:** box, specify the path to the project that contains the window screen you wish to copy. Use the **Browse...** command button if you are unsure of the location or name of the project file.
3. In the **Source Window** box, enter the number of the window screen you wish to copy.
4. In the **Destination Window** box, enter the number of the window screen in the project you are currently editing that you wish to copy the contents of the source window to.
5. Click **OK**. The main screen of EasyBuilder is redisplayed with the Window Copy command executed.

► **To delete a graphics object(s)**

1. Select the object or objects you wish to delete.
2. Press the **DELETE** key or from the **Edit** menu, click **Delete**.



*You can restore the deleted object by using the Undo command.*

► **To move a graphics object(s)**

1. Select the object or objects you wish to move.
2. When you place the mouse cursor over a highlighted object, the cursor changes to a crosshair. This indicates that you are able to move the selected object. Move the object by clicking on it and dragging the object to the proper position on the work area.

► **To resize a graphics object**

1. Select the object to be resized.
2. To resize the object, move the mouse cursor over one of the small black squares. The cursor changes to a double-arrow icon to indicate that it is in resizing mode.
3. Click and drag the mouse to resize the object.
4. Objects can also be resized by changing the width and height attributes in the Profile tab of the Attributes dialog box, (see below).

► **To change attributes of a graphics object**

1. Select the object to be changed, then select **Change Attribute** from the **Edit** menu. You can also double-click the object.
2. The object's attribute dialog box is displayed. Object Attributes defines what the object is or how it behaves, (i.e. size, position, color, etc.). Click **OK** to accept any changes made or **Cancel** to cancel any changes.

► **To view object attributes of multiple graphics objects on a window**

1. There will be times when you may want to quickly determine what PLC data registers are tied to which objects in a window. This can easily be done using the object attributes command.

- For this example, open Window\_11 to display the popup window of the sample project.



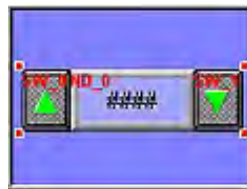
- On the Edit menu, click **Select all objects**.
- From the Edit menu, click **Change Attribute**. The Object Attribute dialog box is displayed.



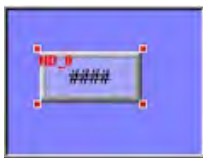
- All of the objects that use PLC data registers or the internal memory of the OIT are listed in this dialog box according to their Object ID tag. Alongside each ID tag is the PLC or OIT memory identifier.

## Grouping Objects

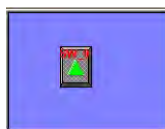
When creating graphics on a window, you may create a complex graphic that is actually composed of several simpler objects overlaid onto each other. This numeric display with increment and decrement button is an example of grouped objects.



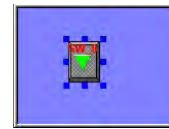
This numeric display is actually composed of three separate objects:



a numeric display



an increment button



a decrement button

These objects are then overlaid to represent the scale. *Grouping* the three objects together makes it easier to move or copy the scale meter. Instead of copying each separate object, the scale meter can be assigned as a group, then copied as though it were one object.

► **To group objects** 

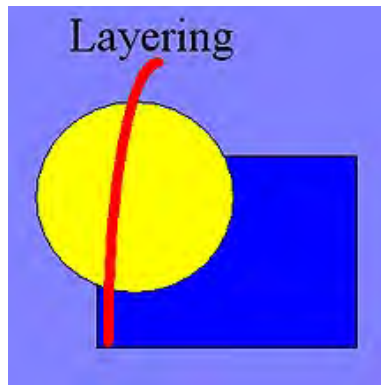
1. Select the objects to be grouped.
2. From the Edit menu, click **Group** or click the **Group** icon from the Manager toolbar.
3. All objects within the group can now be copied, deleted, or moved by clicking on the group.

► **To ungroup objects** 

1. Select the objects to be ungrouped.
2. From the Edit menu, click **UnGroup** or click the **UnGroup** icon from the Manager toolbar.
3. All objects within the group can now be copied, deleted, or moved separately.

## Layering Objects

Graphics objects can be overlaid upon each other. When 'layered' the graphics object on the topmost layer will be completely seen. How much of the other graphics objects are seen depends on what is on top of them. The following layer commands help to position the overlaid objects exactly how you want them. To better illustrate how layering works we will use a rectangle object, a circle object and an arc object that are overlaid on each other in Window\_12 of the sample project.

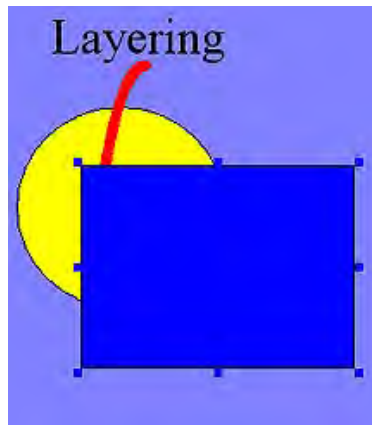


► **Using the top**  **and bottom**  **layer commands**

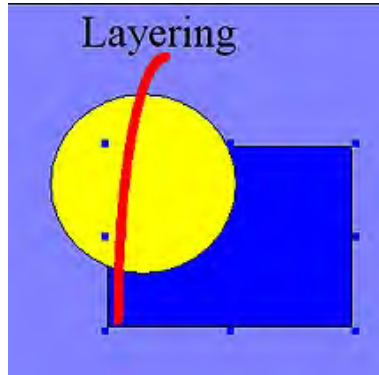
1. Select one of the layered objects. In this example, select the layered rectangle from Window\_12.



- From the Edit menu, click **Layer**, then **TopLayer** or click the **TopLayer** icon from the Manager toolbar. Note that the rectangle completely overlaps the circle and the arc.

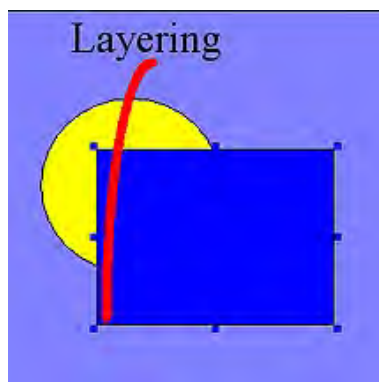


- From the Edit menu, click **Layer**, then **BottomLayer** or click the **BottomLayer** icon from the Manager toolbar. Note that the rectangle recedes behind the circle and the arc.



► Using the next  and previous  layer commands

- These two commands can move an object through the various layers of other objects.
- Select one of the layered objects. Again, select the layered rectangle from Window\_12.
- From the Edit menu, click **Layer**, then **PreviousLayer** or click the **PreviousLayer** icon from the Manager toolbar. Note that the rectangle covers the circle but not the arc.



- From the Edit menu, click **Layer**, then **NextLayer** or click the **NextLayer** icon from the Manager toolbar. Note that the rectangle recedes behind the circle and the arc.

Normally, an object that is controlled by a PLC Register (i.e., a Word Lamp, Bit Lamp, Animation, etc.) is brought to the Top Layer when the value in the PLC Register changes. This behavior can be changed by the Object Layout

option on the General tab of the System Parameters dialog. A setting of Control sets the behavior as outlined above. A setting of Nature (default setting) will result in the object remaining at the layer assigned during development.

 *The Object Layout setting is global - affects all PLC-controlled objects in the application.*

## Nudging Objects

Nudging is used to fine-tune the movement of objects in the work area of EasyBuilder. Using the nudge feature on a selected object will move that object in the specified direction either by one pixel or by the grid setting amount.

► Using the nudge top , bottom , left , and right  commands

1. Select one of the layered objects or a group of objects in the work area of EasyBuilder.
2. From the Edit menu, click **Nudge**, then select the direction of the nudge or click the appropriate icon from the Manager toolbar.

## Aligning Objects

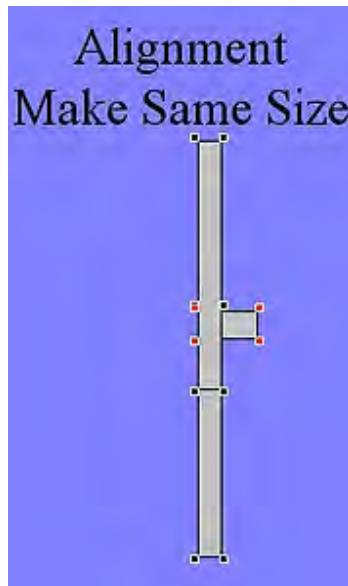
Alignment can be used to quickly align two or more objects. Aligning justifies objects to the desired position of the last-selected object. The last select object will have "handles" of a different color than the other selected objects. To better illustrate, refer to the left side of Window\_12 of the sample project:



► Using the align left  command

1. Select the objects you wish to align. For this example, select the three rectangle objects of Window\_12.

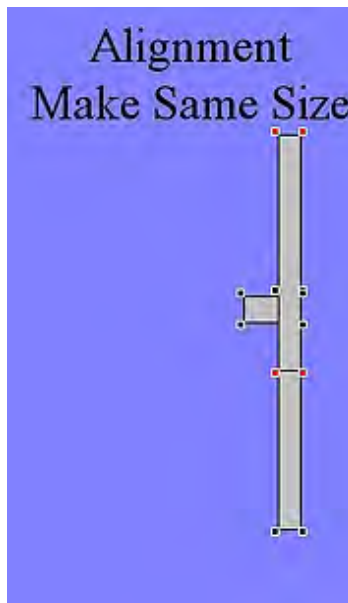
- From the Edit menu, click **Align**, then **Left** or click the appropriate icon from the Manager toolbar.



- From the Edit menu, click **Undo** to put the objects back in their original position.

► Using the align right  command

- Select the objects you wish to align. For this example, select the three rectangle objects of Window\_12
- From the Edit menu, click **Align**, then **Right** or click the appropriate icon from the Manager toolbar.

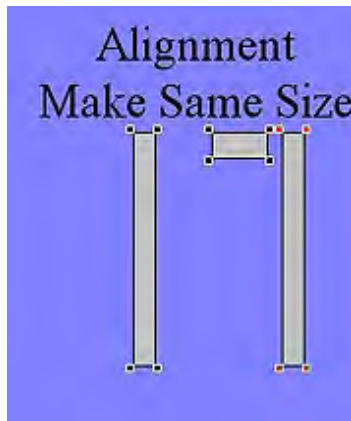


- From the Edit menu, click **Undo** to put the objects back in their original position.

► Using the align top  command

- Select the objects you wish to align. For this example, select the three rectangle objects of Window\_12.

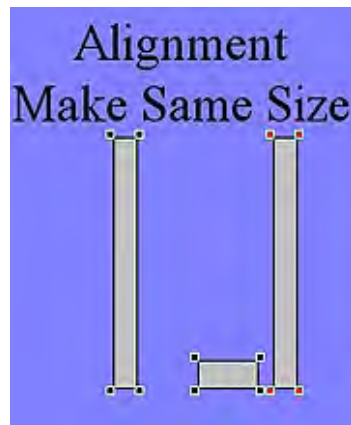
- From the Edit menu, click **Align**, then **Top** or click the appropriate icon from the Manager toolbar.



- From the Edit menu, click **Undo** to put the objects back in their original position.

► Using the align bottom  command

- Select the objects you wish to align. For this example, select the three rectangle objects of Window\_12
- From the Edit menu, click **Align**, then **Bottom** or click the appropriate icon from the Manager toolbar.

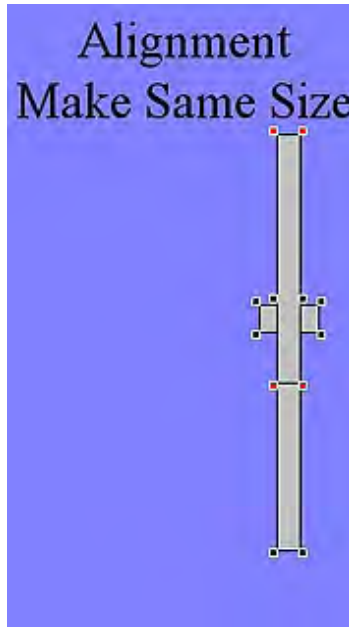


- From the Edit menu, click **Undo** to put the objects back in their original position.

► Using the align horizontal center  command

- Select the objects you wish to align. For this example, select the three rectangle objects of Window\_12.

- From the Edit menu, click **Align**, then **Horiz. Center** or click the appropriate icon from the Manager toolbar.




---

 *Centering is based on the horizontal center of the left-most object.*

---


- From the Edit menu, click **Undo** to put the objects back in their original position.

► **Using the align vertical center**  **command**

- Select the objects you wish to align. For this example, select the three rectangle objects of Window\_12.
- From the Edit menu, click **Align**, then **Vert. Center** or click the appropriate icon from the Manager toolbar.




---

 *Centering is based on the vertical center of the top-most object.*

---

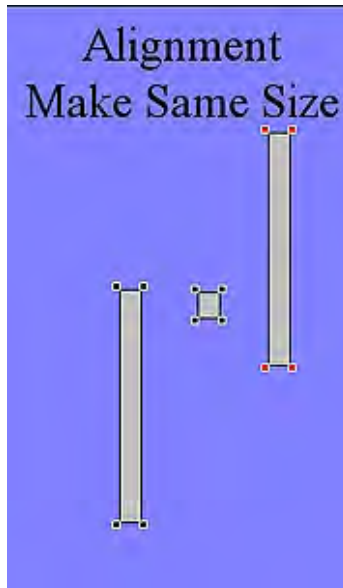
- From the Edit menu, click **Undo** to put the objects back in their original position.

## Making Objects the Same Size

This feature is handy if you want to quickly make two or more objects the same size. Objects are sized to the size of the last-selected object. The last-selected object will have "handles" of a different color than the other selected objects. This is most often used when you are trying to overlap objects that must be the same size. To better illustrate, refer to the left side of Window\_12 of the sample project:

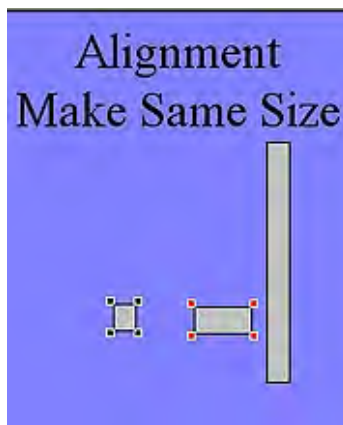
► The make same width  command

1. Select the objects you wish to make the same width. For this example, select the left and center rectangle objects of Window\_12
2. From the Edit menu, click **Make Same Size**, then **Width** or click the appropriate icon from the Manager toolbar.



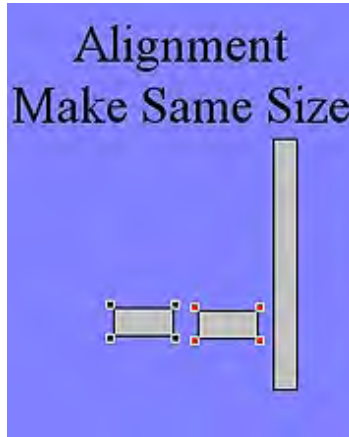
► The make same height  command

1. Select the objects you wish to make the same height. For this example, select the left and center rectangle objects of Window\_12
2. From the Edit menu, click **Make Same Size**, then **Height** or click the appropriate icon from the Manager toolbar.



► The make same size  command

1. Select the objects you wish to make the same size. For this example, select the left and center rectangle objects of Window\_12
2. From the Edit menu, click **Make Same Size**, then **Both** or click the appropriate icon from the Manager toolbar.

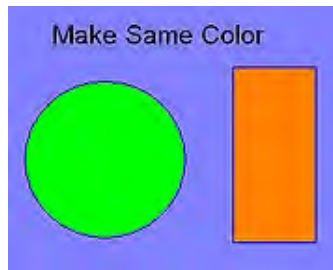


### Making Objects the Same Color

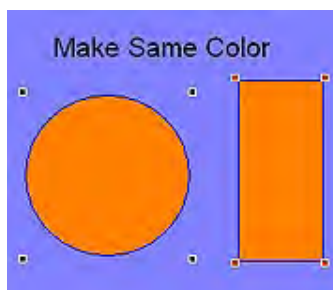
This feature can be used to make two or more drawn objects the same color.

► **To make objects the same color:**

1. Select the objects that you wish to make the same color. EZware will choose the color of the last selected object.

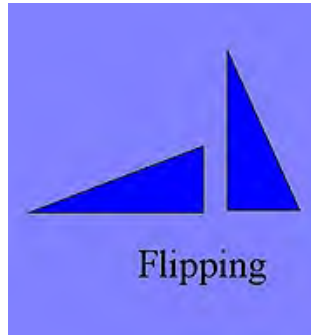


2. Select **Edit-Make Same Color**. The selected objects will all change color to the same color as the last selected object.



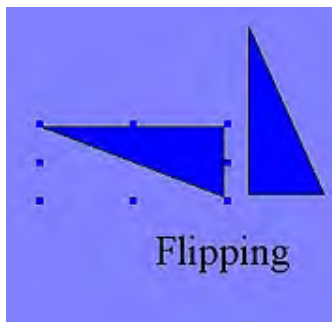
## Flipping Objects

These three commands allow you to quickly 'flip' or position an object in a new direction. To better illustrate, we will again refer to Window\_12 of the sample project.



► To flip vertically 

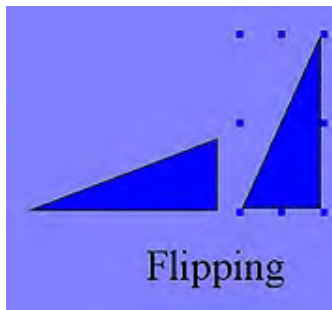
1. Select the object you wish to flip vertically. For this example, select the left triangle object of Window\_12.
2. From the Edit menu, click **Flip Vertical** or click the appropriate icon from the Manager toolbar.



3. From the Edit menu, click **Undo** to put the left triangle object back in its original position.

► To flip horizontally 

1. Select the object you wish to flip horizontally. For this example, select the right triangle object of Window\_12.
2. From the Edit menu, click **Flip Horizontal** or click the appropriate icon from the Manager toolbar.

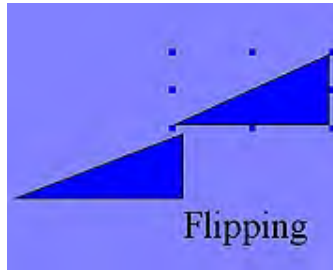


3. From the Edit menu, click **Undo** to put the right triangle object back in its original position.



► **To rotate** 

1. Select the object you wish to rotate. For this example, select the right triangle object of Window\_12.
2. From the Edit menu, click **Rotate 90 degrees** or click the appropriate icon from the Manager toolbar.



► **To Lock an Object** 

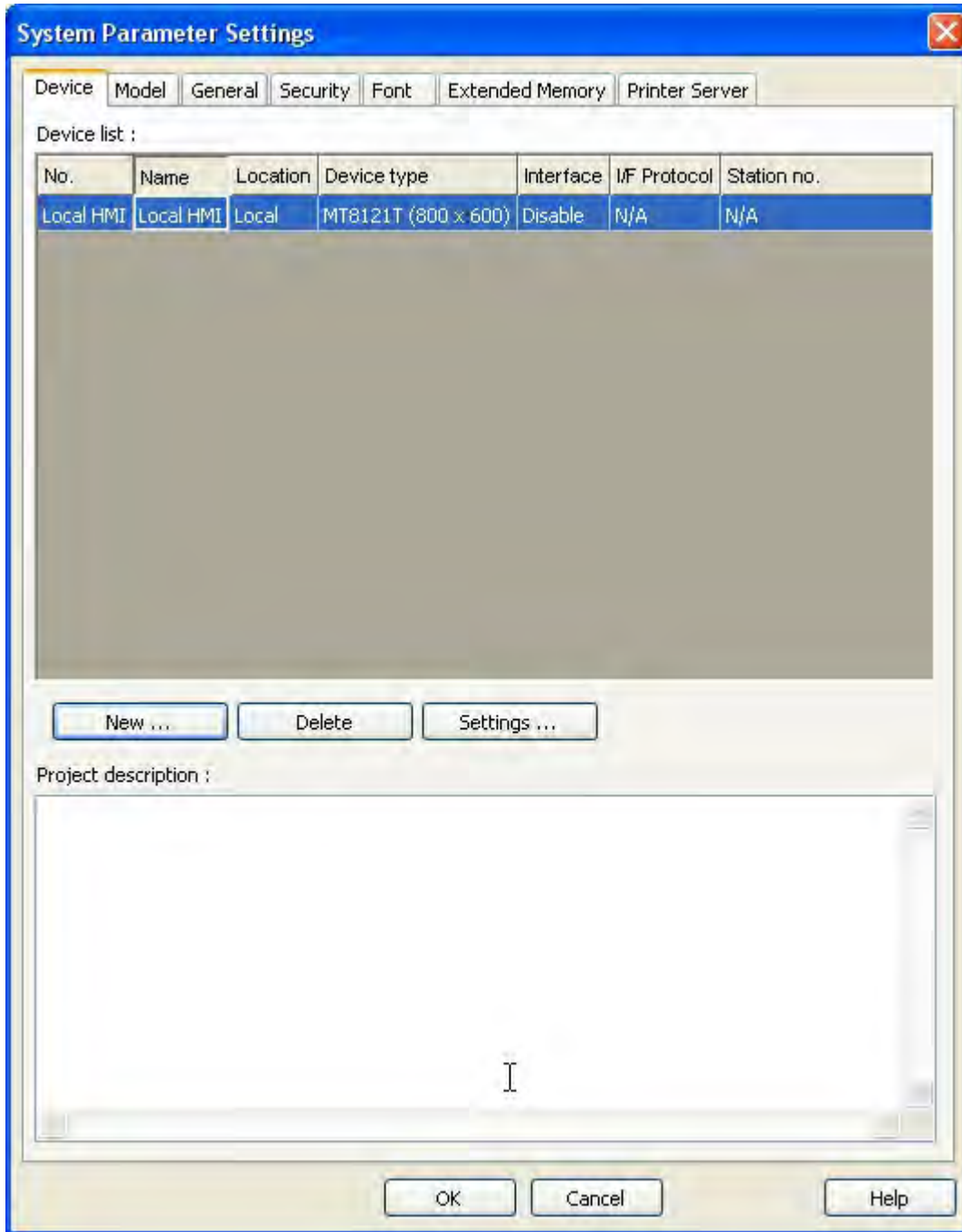
Each object position and size can be locked by using the Pin button. When the object is locked, its position and size cannot be changed.

1. Select the object you wish to lock.
2. From the Edit menu, click **Pinned** or click the appropriate icon from the Manager toolbar.
3. To unlock the object, deselect **Pinned** from the Edit menu or click the appropriate icon from the Manager toolbar.

## General Settings


The final section of this chapter shows all the settings or parameters that can be configured using EasyBuilder. From the Edit menu, click **System Parameters**. The System Parameter Setting dialog box appears. The dialog box has seven tabs: Device, Model, General, Security, Font/Language, Extend Memory and Printer Server.

## PLC Settings Configuration



► **To configure PLC settings**

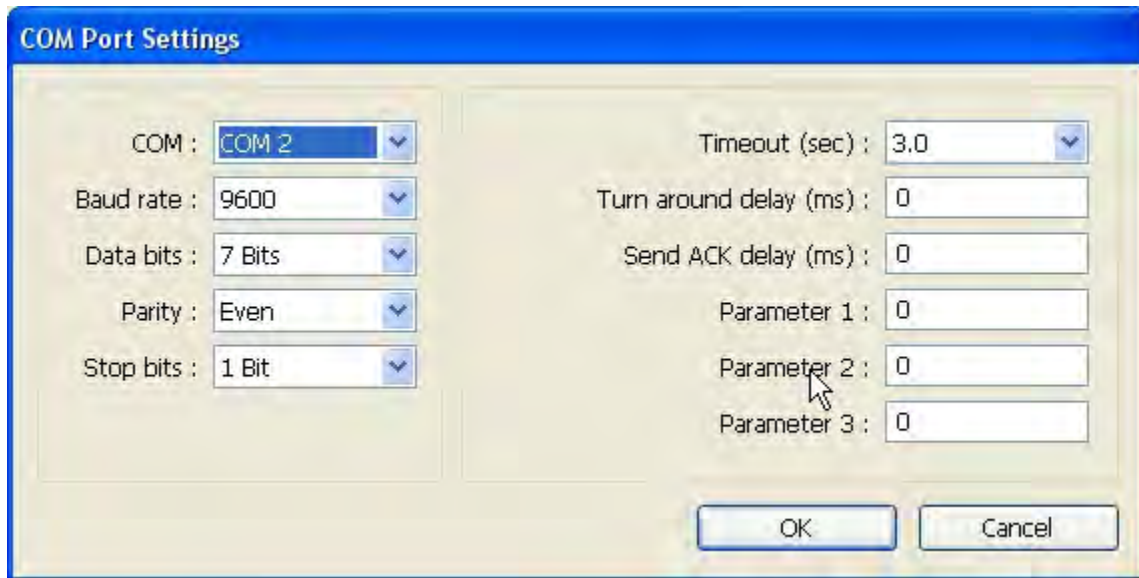
1. Click **New...** The Device Properties dialog appears.

 You can “export” a project created for one PLC brand to a different PLC brand by loading the project file into EasyBuilder, then entering a different PLC type in this box. EasyBuilder will go through your entire project and attempt to change each reference to PLC data registers to a logical selection for the new PLC protocol; however, we strongly recommend that you review these changes to assure that they are satisfactory.

2. Select the PLC button and then select your *PLC* from the drop-down menu.
3. There are two choices for Location - either *Local* or *Remote*. Select **Local** if the PLC will be connected directly to the OIT. Select **Remote** if the PLC will be connected to a remote OIT. If Remote is selected, then click Settings and configure the appropriate IP settings.


4. IP Address : Enter the *address* of the remote OIT.
5. Port No. enter the *TCP/IP port* assigned to the remote OIT during its configuration. Click **OK**.
6. Select the appropriate *PLC* type from the drop down menu.
7. There are four selections for PLC I/F:
  - RS-232**
  - RS-485 2W**
  - RS-485 4W**
  - Ethernet**

8. Select COM port and PLC station number (see Controller information Sheet for your PLC) and then click **Settings...** The COM Port Setting dialog appears.



9. Select the appropriate COM port..
10. Select *baud rate*, *parity*, *data bits*, and *stop bits*. Use Maple Systems' Controller Information Sheets or the PLC manufacturer's operations manual for information on the communications parameters required by the PLC. These parameters have to match the PLC settings.
11. Click **OK**.
12. **Interval of Block Pack (Words)**: Determines how many words are read from the PLC in one communication cycle. For more information on this feature, see *PLC Block Pack* at the end of this chapter.
13. Click **OK**.

## Model Settings Configuration



The screenshot shows the 'Model' configuration window. It features several tabs: 'Device', 'Model', 'General', 'Security', 'Font', 'Extended Memory', and 'Printer Server'. The 'Model' tab is selected. The configuration is organized into three main sections:

- HMI model:** A dropdown menu showing 'HMI5104XH / HMI5121X (800 x 600)'. Below it, 'HMI station no' is set to '0' and 'Port no.' is set to '8000' (noted as 'used as MODBUS server's port no.').
- Timer:** A section with a 'Clock source' dropdown set to 'HMI RTC'.
- Printer:** A section with a 'Type' dropdown set to 'None'.

1. HMI Model: Select your *HMI Model*.
2. HMI Station No: See Controller Information Sheet for more information.
3. Port No: Assign a *TCP/IP port* so that another OIT can access this port. For more information, see the "*Using Remote HMIs*" section.
4. Clock Source: Select *internal HMI RTC* or *External Device*.
5. Printer Type: Select *type of printer*.

## Auto Resizing Using the Model Tab

This tab can also be used to auto resize your project. The project and all objects and graphics will be automatically proportionately resized.

► **To auto resize your project using the Model tab in the System Parameters dialog:**

1. In the Model tab, select a different *display size* from the HMI model drop-down box and click **OK**.
2. If the new display is smaller than the existing display, the project will automatically resize. If the new display size is larger than the existing display, the Resize Popup Windows/ Objects dialog appears.



3. Check **Resize pop-up windows** and **Resize objects**, and then click **OK**. The project will automatically resize.

## General Settings Configuration

1. **Fast select button attribute:** Enables or disables the fast selection window (which is designated as window number 3). Settings sets the look of the button including shape and color. Position sets the fast selection button on either the lower left or lower right portion of the display.

### 2. Screen Saver:

**Backlight Saver** sets the number of minutes (none-255) of inactivity after which the backlight is turned off.

Note: Backlight is re-triggered when the screen is pressed or when an alarm occurs (if the "Enable back light when alarm occurs" option is selected).

**Screen Saver** sets the number of minutes (none-255) that the screen goes untouched before activated.

Note: If the value "none" is selected, then the screen saver is disabled.

**Saver Window No:** designates the window that will be used as the screen saver.

## 3. Option:

**Startup Window No:** Selects the window number that will be displayed every time the OIT is started up.

**Extra No of events:** The default number of events in the system is 1000.

Note: Users can add more records, up to 10,000 using this setting.

**Common window:** Determines if objects in the common window are placed above/below the base window.

**Keyboard caret color:** Sets the color of the cursor.

**Object layout:** Determines layout/display of objects using Control Mode or Nature Mode. (see note below)

**RW\_A enabled:** Enables an extra 64K of recipe data.

*Please Note: Normally, an object that is controlled by a PLC Register (i.e., a Word Lamp, Bit Lamp, Animation, etc.) is brought to the Top Layer when the value in the PLC Register changes. This behavior can be changed by the Object Layout option on the General tab of the System Parameters dialog. A setting of Control sets the behavior as outlined above. A setting of Nature (default setting) will result in the object remaining at the layer assigned during development.*

4. **Keyboard:** Lists the windows that have been configured as keyboards.

5. **Add:** Opens a window list from which you can select an additional window to be available as a project keyboard.

6. **Delete:** deletes the selected keyboard from the list.

## Security

The screenshot shows the 'System Parameter Settings' dialog box with the 'Security' tab selected. The dialog contains a table of settings for 12 users. Each user row includes an 'Enable' checkbox (all checked), a 'Password' field (all containing '0'), and checkboxes for operable classes A through F (class 'A' is checked for all users).

User	Enable	Password	A	B	C	D	E	F
User 1	<input checked="" type="checkbox"/>	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User 2	<input checked="" type="checkbox"/>	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User 3	<input checked="" type="checkbox"/>	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User 4	<input checked="" type="checkbox"/>	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User 5	<input checked="" type="checkbox"/>	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User 6	<input checked="" type="checkbox"/>	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User 7	<input checked="" type="checkbox"/>	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User 8	<input checked="" type="checkbox"/>	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User 9	<input checked="" type="checkbox"/>	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User 10	<input checked="" type="checkbox"/>	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User 11	<input checked="" type="checkbox"/>	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User 12	<input checked="" type="checkbox"/>	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

At the bottom of the dialog are 'OK', 'Cancel', and 'Help' buttons.



The Security tab determines the table of user passwords. Up to 12 sets of password can be set where one or all users can be set or disabled. Each user can be individually enabled, and has a unique password and combination of any six Access Classes, A-F. Only numbers are allowed for password.

When an object is created, it can be assigned an Access Class (A-F). When that object is activated (for example, a button is touched), the OIT compares the access classes allowed for the current user to the access class specified for the object. If the access class of the object is within the classes allowed, the object's action is executed.

The access class for an object is configured in the User *Restriction* section of the Security tab for that object. So, for example, if you wish to set the access class for a set bit object, click on the security tab of that objects properties dialog, as shown below:

The object can be configured to handle insufficient security in one of two ways:

- The object can be hidden while security is not sufficient.
- The OIT can display the message defined as Message 1 in the System Message object.

There are a number of local OIT addresses that can be used with the security features.

Address	Description	Data Type	Notes
LB9050	Reset security level to 0 (no security)	Boolean	Set momentarily to trigger.
LB9060	Invalid password	Boolean	Set by the OIT when an invalid password is entered. Must be explicitly reset.
LB9061	Load new passwords	Boolean	Loads new passwords as defined by LW9500-9523
LW9219	User level	16-bit signed	Set to 1-12. Determines which password is used.

LW9220	Password	32-bit unsigned	Enter password into this address to change the security level
LW9222	Current access classes	16-bit signed	Indicates the Access Classes currently allowed. Bit 0 indicates Class A; Bit 5 indicates Class F.
LW9500, LW9501	New password, user 1	32-bit unsigned	Loaded into security system when LB9061 is set momentarily
LW9502, LW9503	New password, user 2	32-bit unsigned	Loaded into security system when LB9061 is set momentarily
LW9504, LW9505	New password, user 3	32-bit unsigned	Loaded into security system when LB9061 is set momentarily
LW9506, LW9507	New password, user 4	32-bit unsigned	Loaded into security system when LB9061 is set momentarily
LW9508, LW9509	New password, user 5	32-bit unsigned	Loaded into security system when LB9061 is set momentarily
LW9510, LW9511	New password, user 6	32-bit unsigned	Loaded into security system when LB9061 is set momentarily
LW9512, LW9513	New password, user 7	32-bit unsigned	Loaded into security system when LB9061 is set momentarily
LW9514, LW9515	New password, user 8	32-bit unsigned	Loaded into security system when LB9061 is set momentarily
LW9516, LW9517	New password, user 9	32-bit unsigned	Loaded into security system when LB9061 is set momentarily
LW9518, LW9519	New password, user 10	32-bit unsigned	Loaded into security system when LB9061 is set momentarily
LW9520, LW9521	New password, user 11	32-bit unsigned	Loaded into security system when LB9061 is set momentarily
LW9522, LW9523	New password, user 12	32-bit unsigned	Loaded into security system when LB9061 is set momentarily

To log out, use a Set Bit to momentarily set bit LB9050.

## Font



Determines the font for non-ASCII strings. To add a Windows TrueType font that is on your computer, click **Add...** and select the font from the drop down menu that appears, and click **OK**.

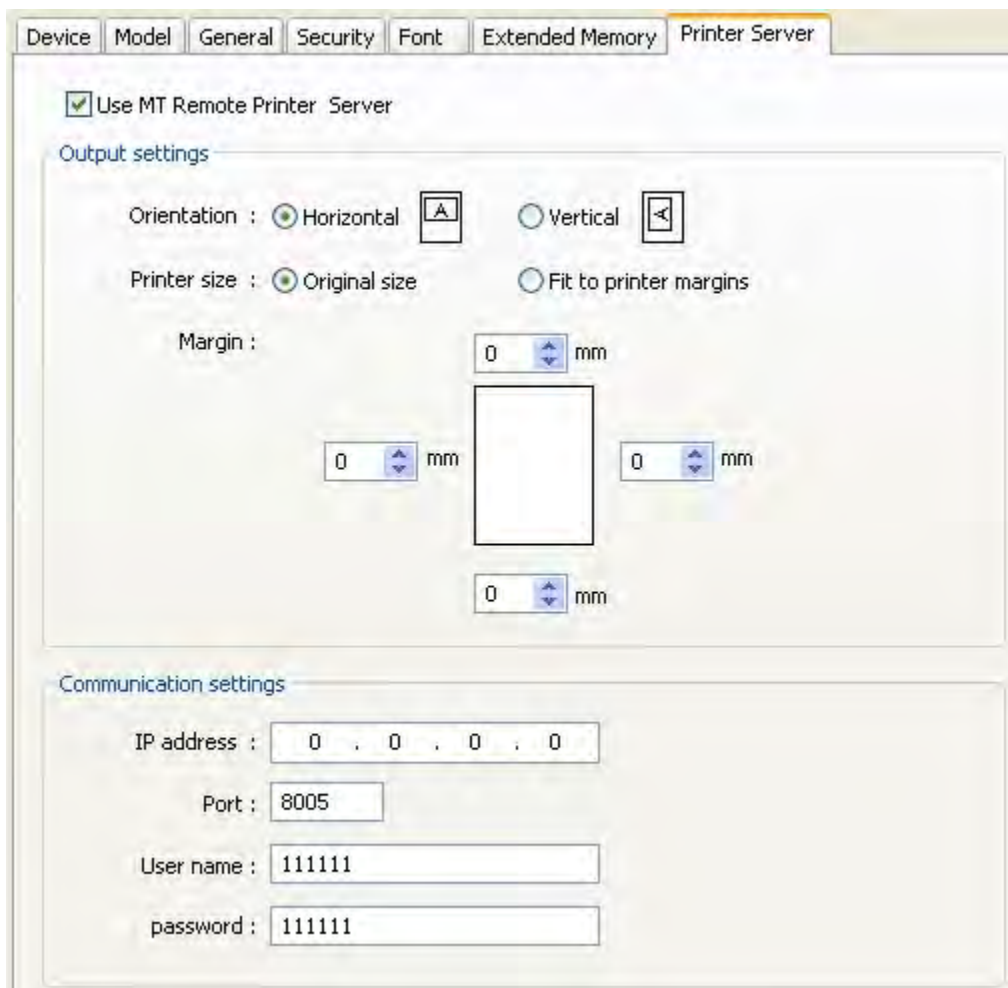


## Extended Memory

Device	Model	General	Security	Font	Extended Memory	Printer Server
EM0	File name :	em0.emi	<input type="radio"/> CF card	<input checked="" type="radio"/> USB 1	<input type="radio"/> USB 2	
EM1	File name :	em1.emi	<input type="radio"/> CF card	<input checked="" type="radio"/> USB 1	<input type="radio"/> USB 2	
EM2	File name :	em2.emi	<input type="radio"/> CF card	<input checked="" type="radio"/> USB 1	<input type="radio"/> USB 2	
EM3	File name :	em3.emi	<input type="radio"/> CF card	<input checked="" type="radio"/> USB 1	<input type="radio"/> USB 2	
EM4	File name :	em4.emi	<input type="radio"/> CF card	<input checked="" type="radio"/> USB 1	<input type="radio"/> USB 2	
EM5	File name :	em5.emi	<input type="radio"/> CF card	<input checked="" type="radio"/> USB 1	<input type="radio"/> USB 2	
EM6	File name :	em6.emi	<input type="radio"/> CF card	<input checked="" type="radio"/> USB 1	<input type="radio"/> USB 2	
EM7	File name :	em7.emi	<input type="radio"/> CF card	<input checked="" type="radio"/> USB 1	<input type="radio"/> USB 2	
EM8	File name :	em8.emi	<input type="radio"/> CF card	<input checked="" type="radio"/> USB 1	<input type="radio"/> USB 2	
EM9	File name :	em3.emi	<input type="radio"/> CF card	<input checked="" type="radio"/> USB 1	<input type="radio"/> USB 2	

Used to set parameters for saving data to CompactFlash and USB memory storage.



## Printer Server



Device Model General Security Font Extended Memory **Printer Server**

Use MT Remote Printer Server

Output settings

Orientation :  Horizontal   Vertical 

Printer size :  Original size  Fit to printer margins

Margin :

0 mm

0 mm

0 mm

0 mm

Communication settings

IP address : 0 . 0 . 0 . 0

Port : 8005

User name : 111111

password : 111111

1. Check the **Use Remote Printer Server** checkbox to enable the OIT to print to a printer connected to a PC on the network.

 The EasyPrinter utility (included with EasyBuilder5000) must be running on the PC with the printer.

2. In the Output Settings section, select *page layout* (horizontal or vertical), print area size and margins.
3. In the Communications Settings section, set the settings for the PC hosting the printer.

## PLC Block Pack

This feature allows the OIT programmer to enhance the update rate of data received from the PLC when the PLC registers displayed on a screen are 'blocked' close together. The default setting is 0. This means that the OIT sends a command to the PLC requesting data for each register that is displayed on screen. For instance, if you have five data fields configured on the OIT to monitor %R1, %R2, %R3, %R4, and %R5 of a GE PLC, then the OIT sends five separate commands to the PLC to update these fields.

This is the slowest method of updating the registers, but it does have the advantage that the update rate does not depend on having the PLC registers all together, (R1-R5). In other words, if monitoring %R1, %R10, %R65, %R156, and %R2048, the update rate would be exactly the same as monitoring %R1-%R5.

The "PLC block pack" feature allows you to take advantage of any groups or 'blocks' of registers that allow the OIT to send a smaller number of update commands to the PLC to get the information needed. The OIT will request data for up to 32 contiguous PLC registers depending upon the PLC block pack setting and the addresses of the PLC

registers monitored on screen. The PLC block pack # represents the maximum number of 'skips' between registers allowed before the OIT considers a PLC register to be part of a different block.

For example, suppose we wish to monitor the following 7 PLC registers on one screen: %R1, %R2, %R4, %R7, %R12, %R14 and %R100. If PLC block pack is set to 0, the OIT sends seven separate commands to get the data. If PLC block pack is set to 1, the OIT checks the spacing between requested registers. If the spacing is more than 1, then a separate command is sent. In this example, the OIT sends four commands- one command to read R1, R2, and R4; one command to read %R7, one command to read %R12 and %R14, and one command to read %R100.

With most PLC protocol drivers, more time is spent with 'overhead' data than getting the actual data that you need. With the example above, setting the block pack number to 1 probably reduces the time required to update the PLC data by as much as 40%! So properly setting the PLC block pack can have a major impact on the update rate of the OIT.

If the PLC block pack setting were adjusted to 4 or greater, the OIT sends only two commands- one command to read %R1, %R2, %R4, %R7, and %R12; and another command to get %R100. The update rate is now probably three times as fast!

You might think that the best setting is 10 so why not have the PLC block pack always on 10? However, in some situations, a setting of 10 may actually slow down the update rate. For example, if the PLC registers displayed on the screen are actually %R1, %R10, and %R40 with the PLC block pack is set to 10, the OIT will send two commands: one to get %R1 and %R10, another to get %R40. If the PLC block pack were set to 0, then the OIT would have sent three commands. In this case, however, the three commands probably take less time than the two commands because the OIT has to receive data for ten registers (%R1-%R10) in one command takes more time than the overhead associated with two commands. Therefore, you should experiment with this setting if you find that the update rate is longer than you would like.

Now that you know your way around EZware-5000 and its various applications, it is time to start creating your own project. The next seven chapters show you in detail, all of the features that you can use to create a user-friendly touchscreen graphics operator interface terminal. Along the way, we will attempt to provide you with clear definitions of each feature and, if necessary, some additional examples of how each feature might be used.

## Automatically Reboot OIT

EasyBuilder can automatically reboot the OIT after download.

### ► To enable automatic reboot of OIT after download

1. From the Tools menu, select **Download**. The Download dialog box is displayed.
2. Check the **Reboot HMI after download** checkbox.
3. Click **OK**.

## Save and Compile the Project

EasyBuilder can automatically save and compile the project when downloading or simulating.

### ► To enable automatic save and compile

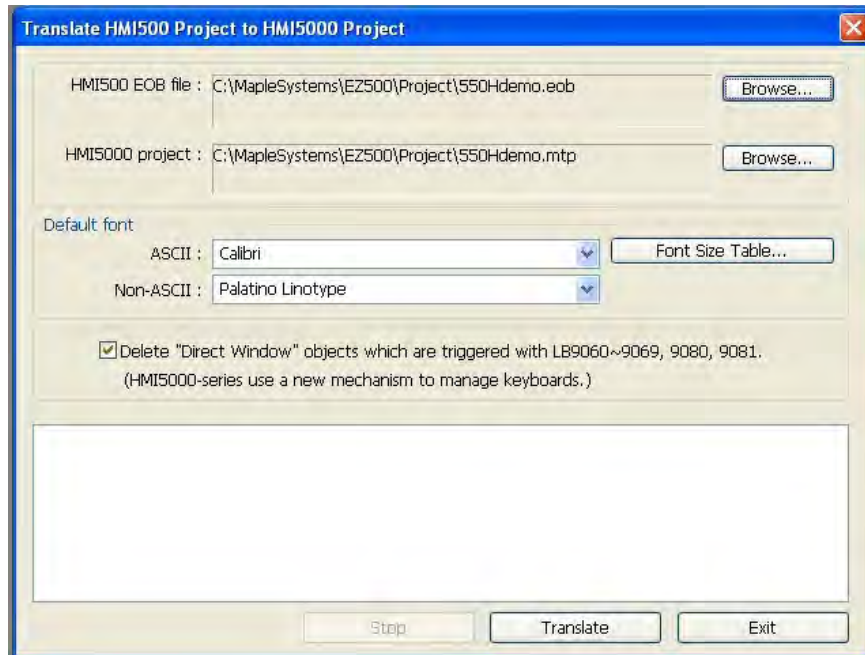
1. From the Option menu, select **Function Property**. The **Function Property** dialog box is displayed.
2. Check the **Automatic save and compile when download and simulate** checkbox.
3. Click **OK** to exit the **Function Property** dialog box.

## EZWare-500 project translation utility

EZware-5000 has a translation utility that will convert projects created in EZware-500 {for HMI550T/TE, HMI530T/TE, and HMI520T/TE models} into a project format that can be read and modified using the EasyBuilder-5000 software. Below is a brief description of how to use this utility (for more information, download technical notes on this subject from Maple Systems' website):

### To convert a \*.eob (EZware-500 project) into a \*.mtp (EZware-5000 project)

From the Tool menu, click Translate HMI500 Project:



1. Click the Browse... button for the HMI500 EOB file and select the EZware-500 project file you wish to convert.
2. By default, the converted file is automatically placed into the same directory. To modify the name of the file or location, click the Browse... button for the HMI5000 project file.
3. Default font frame: All text objects in theEZware-500 project file use a proprietary fixed font. When converting to an EZware-5000 project, these text objects are converted to TrueType fonts. Use this section to specify which TrueType fonts you wish to use.
  - a. ASCII: This is the true-type font that is used on all text objects and labels.
  - b. Non-ASCII: International characters (such as ä) used in EZware-500 are converted using this selected font.

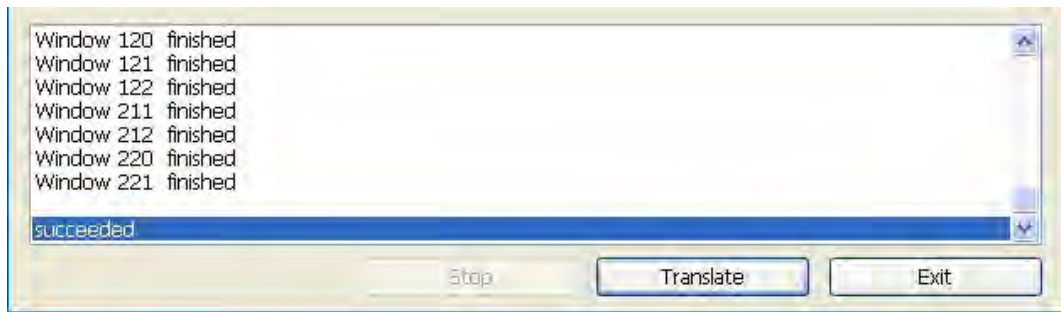
- c. Font Size Table...: select a specific EZware-5000 font size that will be used for each EZware-500 font size.



4. Delete “Direct Window” objects which are triggered with LB9060~9069, 9080, 9081, (HMI5000-series use a new mechanism to manage keyboards.): Check this box to delete any Direct Windows objects in the EZware-500 project file which are triggered by the internal memory bits listed above.

*Please Note: In EZware-500, these internal memory bits allowed a popup window to display based upon touching any Numeric Input or ASCII Input object located in a particular area of the screen. This feature is not available in EZware-5000. By default, this option is checked.*

5. Click the Translate button to begin conversion of the file. When done, you will see text in the comments field that indicates if the conversion is successful:



6. Click the Exit button to go back to the main screen of EasyBuilder-5000. Open the converted file as you would normally open any EZware-5000 project file.

*Please Note: due to some incompatibilities between the EZware-500 software and the EZware-5000 software, you should check each screen and test operation after conversion to make sure that your OIT works to your satisfaction.*

**Please Use This Page For Your Notes:**



## Chapter 7 - Creating Windows

This section shows how to create windows using EasyBuilder. To better illustrate some of the examples, please open the sample project included with the EZware-5000 configuration software.

### Window Fundamentals

An operator interface terminal wouldn't be very useful if all of the information to be displayed could only be placed onto one screen. Therefore, most OITs have multiple screens that you can use to display information. The Maple Systems Silver Series is capable of storing up to 1999 windows (actual limit is determined by memory requirements of each screen), giving you maximum flexibility in designing your operator interface. We prefer to call these screens 'windows' because they have several features not normally associated with screens:

- Windows can be created in any size. You can make the window full-sized so that it fits the entire area of the OIT display or you can create a window that partially covers the display.
- Windows can be overlaid on top of each other. All data on each window displayed is updated continuously regardless of whether or not it is covered up by another window.
- Windows can be moved around OIT display to allow portions of other windows to come into view.

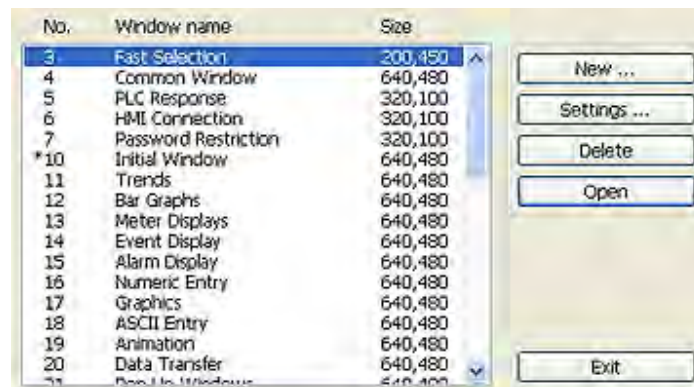
The Silver Series has four basic types of windows available for use: Base windows, a Common window, System Message window, and a Fast Selection window. Base windows are the windows that you will most often use. The common window and fast selection window are two windows reserved for special functions. System Message windows are reserved for custom messages from the controller. By the end of this chapter, you should be able to create these windows and use the many features available to them.

### Opening a Window

To view the contents of a window in EasyBuilder, it must first be opened. When you create or open an existing project file only the initial window is opened. To view any other windows that have already been created, you must first open the window. This can be done using the Window Tree (see Chapter 3, "Using EZware-5000", Display Options) or by performing the following:

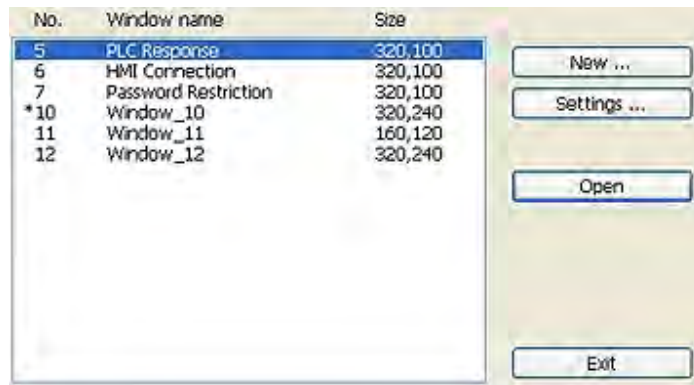
#### ► To open a window

1. From the Window menu, select **Open Window**. The Open Window dialog box appears.



2. Click on the Window you wish to open. Then click **Open**. Note: you can also open the window by double-clicking the window.
3. The Open Window dialog box closes and the opened window is displayed in the EasyBuilder work area.

Let's look at the Open Window dialog box again using the sample project.






The Open Window dialog box lists all of the windows currently created for the project. You will notice that in the sample project, three windows have been created: 10, 11, and 12. The asterisk next to Window #10 indicates that this window has already been opened. Each window is listed with the Window name and size. The window name is the name that you assign to the window when it is created. The window size is shown for quick reference. Finally, you will notice that the Open Window dialog box is also used to create a new window or change any of the settings for a window. Press the **Exit** command button to return to the main EasyBuilder screen.

By default, when a window is opened it will replace any window that was displayed in the work area of EasyBuilder. To switch between open windows, click the **Window** menu and select from the list of windows currently open. You can also *cascade* (select **Window-Cascade**) or *tile* (select **Window-Tile**) the open windows to see the windows at the same time.

EasyBuilder requires more resources from your computer every time you open another window. When many windows are open, the performance of the computer may be affected. Therefore, you may wish to close some of the windows until you need to edit them.

#### ► To close a window

1. You will notice two small icons located in the upper right hand corner of each window: the minimize icon , the maximize icon , and the close icon . To close a window, click the close icon associated with that window.

These icons show in the window if minimized. Otherwise, the icons are located on the rightmost end of the menu bar of EasyBuilder.

## Creating a New Window

Whenever you create a new project, several windows have already been predesignated by EasyBuilder:

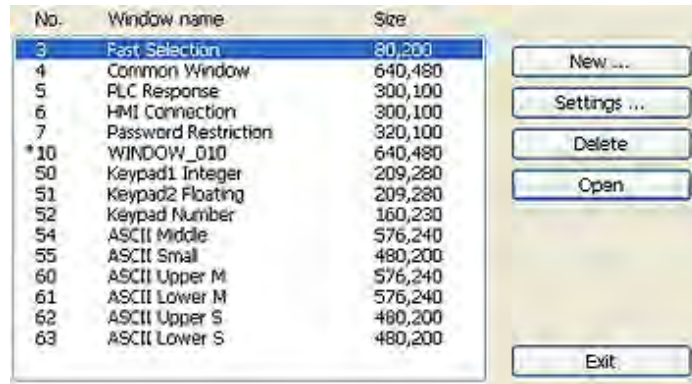
- #3: Fast Selection - The window controlled by the Fast Sel button that allows you to quickly jump to commonly accessed windows or objects.
- #4: Common Window - A window that displays graphics (such as a company logo) or objects that are always on every screen of the display. The common window runs below every other window that is being displayed on the OIT.
- #5: PLC Response - This is the window that is displayed when the OIT loses communications with the PLC.
- #6: HMI Connection - This is the window that is displayed with the OIT loses communications with a remotely connected OIT.
- #7: Password Restriction - This is a window that is displayed when a protected object or window access is attempted without the proper password.
- #10:Startup Screen - The window that displays whenever the OIT is started up.
- #50:Keypad1 Integer - A style of numeric keypad.
- #51:Keypad2 Integer - A style of numeric keypad.
- #54:ASCII Middle - A style of an ASCII keypad.
- #55:ASCII Small - A style of an ASCII keypad.

- #60:ASCII Upper M - A style of an ASCII keypad.
- #61:ASCII Lower M - A style of an ASCII keypad.
- #62:ASCII Upper S - A style of an ASCII keypad.
- #63:ASCII Lower S - A style of an ASCII keypad.

The remaining windows are available to be created as you desire for your project. In order to create new windows, the following steps must be performed:

► **To create a new window**

1. From the Window menu, select **Open Window**. The Open Window dialog box appears.



2. Click **New...** The Select Window Style dialog box appears.



3. Click **Base Window**. If the Fast Selection and Common Window buttons appear grayed out,

then it is because those windows have already been created. The Window Setting dialog box appears.

4. Modify the window parameters, then press **OK**. The Open Window dialog box reappears.
5. If you wish to open the window you just created, click on the window and click **Open**. Otherwise, click **Exit** to return to the main screen of EasyBuilder.

## Window Settings

Let's look more closely at the parameters you can change when creating a new window.



*When a window is initially created, the window settings can be edited to make changes; however, once a window has been created, the window number can not be changed. To change the window settings, highlight the desired window and click on the **Setting...** button.*

### Assigning a Window Name

The Name is a description box used to help you identify what the window is used for without having to actually open the window and look at the contents. Up to 50 characters can be entered into this field with space characters allowed.

### Assigning the Window Number

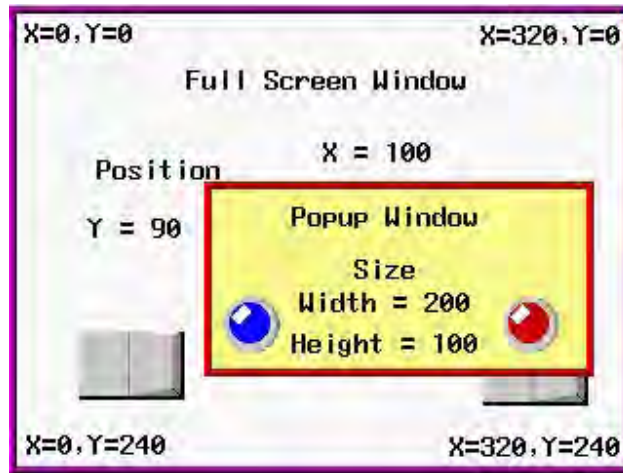
Although 1999 windows are available on the Silver Series, two are specifically reserved for the Common Window and the Fast Selection window and several are reserved for internal use. Therefore, you can assign #10-1999 to any window you create. When you initially create a window, EasyBuilder will automatically assign the lowest available number to the window. However, you can assign any number within the allowed range. In this manner, you can group windows together that may share some common traits.

### Assigning a Position

This is the starting position that the window goes to when it is initially called onto the OIT screen. The starting position is labeled Start Pos: in the Window Setting Dialog box. The X and Y positions refer to the pixel location of the OIT display at which the upper left hand corner of the window is to be displayed.

The default setting is X=0 and Y=0 which is the upper left hand corner of the OIT display. The X-axis refers to the horizontal location and the Y-axis refers to the vertical location. The HMI5056 has a 320 x 240 pixel display, so the ranges are X=0-319 and Y=0-239. The HMI5070 has a 480 x 234 pixel display, so the ranges are X=0-479 and Y=0-233. The HMI5080/5104 has a 640 x 480 pixel display, so the ranges are X=0-639 and Y=0-479. The

HMI5121 has a 600 x 800 pixel display, so the ranges are X=0-599 and Y=0-799. The HMI5150 has a 1024 x 768 pixel display, so the ranges are X=0-1023 and Y=0-767.



**Assigning Size of Window**

You can vary the size of a new window to create full screen or popup windows. Popup windows are most often used to display data that does not need to be on the OIT display all of the time. For example, you might want to configure a numeric keypad in one popup window that can be moved aside or closed when not needed. The keypad can be triggered to pop up when needed or appear with a keypress. The allowable range for the width is 100-320 (640 for the HMI5080/5104/5121) pixels. The allowable range for the height is 100-240 (234 for the HMI5056/5070, 480 for the HMI5080/5104/5121) pixels.

**Monopoly Feature**

The monopoly feature is used to ‘monopolize’ all touchscreen action that can occur on the OIT screen. For instance, suppose the OIT display is currently showing a full screen window with several touchscreen objects.



If a popup window is displayed which *does not* have the monopoly feature enabled, the OIT operator is able to press any function keys on the popup window or the full screen window since objects on both screens are active.



In this example, the OIT operator is able to press Key1-3 on the full screen window or the switch on Popup Window#21. If another popup window is displayed that does have the monopoly feature enabled, then Key1-3 on the full screen window will not respond when pressed.



Please note that the monopoly feature only disables touchscreen objects on the *full screen window*. The switch on Popup Window#21 can still be activated, as can any popup window displayed on the full screen.

The monopoly feature can be used to display a popup window with some action that the OIT operator must perform before being allowed to do some other action on the OIT. For example, you might construct a dialog box that asks the OIT operator if some step in the control process has been performed. The dialog box would have **Yes** and **No** function key options which, with the monopoly feature enabled, the OIT operator must press before continuing.

### Assigning Underlay Windows

An Underlay Window is a Base Window that is displayed at the same time as the Base Window that calls it. Each full-size Base Window can display up to three Underlay Windows. A Pop-Up Window cannot display any Underlay Windows. There may be times when it is desirable to place the same information on multiple (but not all) windows. Underlay Windows provide a means to accomplish that without the overhead of a Direct or Indirect Window, and without using the Common Window (which makes the information visible on all windows). Underlay windows also eliminate the need to actually place the same objects on multiple windows, which increases the memory required for the project. Each Base Window can display up to three Underlay Windows, which are called Bottom, Middle, and Top.

EasyBuilder allows you to assign up to three underlay windows to any base window that you are creating. First, create the underlay window; then select it from the pull-down list box when you change the settings of the base window.

### **How to Display Underlay Windows**

When you have common objects to display on multiple screens, use the Underlay Window feature to store these on underlay windows attached to a base window. This decreases the amount of memory required for the project. The base window “calls” the underlay window. Each full-size base window can display up to three underlay windows, which are called Underlay #1, Underlay #2, and Underlay #3. Objects on the underlay windows cannot be edited from the base window on which they are displayed.

A pop-up window cannot display any underlay windows.

To assign underlay windows to a base window, you must first create the underlay windows. Underlay windows are created the same way base windows are created. Then you can assign the underlay windows to the base window.

#### ► **To assign underlay windows to a base window**

1. From the Window menu, select **Open Window**. The Open Window dialog box appears.
2. Highlight the *base window* you wish to use.
3. Click the **Setting** command button. The Window Setting dialog box appears.
4. Click the *pull-down list box* for each Underlay Window to assign the underlay window screens to the base window.
5. Click **OK**. Click **Close** on the Window Setting dialog box.

### **Rules That Apply to Underlay Windows**

Please note the rules that apply when you use the underlay window feature:

- Only the objects of an underlay window are displayed on the base window. All background information about the underlay window (i.e. background color, frame color and size, etc.) are not displayed on the base window.
- Active objects can be ‘overlaid’ on top of each other over two or more underlay windows. For example, a Set Bit object that is configured to set an OIT local bit LB0 is located at X=20, Y=50 for Underlay Window #1. Another Set Bit object that controls LB1 is placed on Underlay Window #2 at the same location. When the Base Window displays these two windows, both objects will be active so that when they are pressed both LB0 and LB1 will be set.
- Static objects may be overlaid using underlay windows. However, any static object that is on the Base Window has the highest precedence and will be displayed over any static objects that are on the underlay windows. Active objects have higher precedence over static objects.
- Popup windows cannot display underlay windows.
- Underlay windows are always positioned from the top left corner of the screen. Any position settings you make when creating the underlay window are ignored when the window is displayed by a Base Window as an underlay window.

For more information and an example of how to use underlay windows, consult the EasyBuilder Help files.

### **Creating a Frame**

EasyBuilder provides the option of having a frame around any window that you create. You can a frame from 0-16 pixels wide in any of EasyBuilder's 65K available colors. The default setting is no frame.

### **Window Background**

You can also select a different background color for each window created. The default setting is black.

#### ► **To select a different background color**

1. From the Window menu, select **Open Window**. The Open Window dialog box appears.
2. Click on the window that you wish to change the background color. Click **Setting**. The Window Setting dialog box appears.
3. In the Background frame section, check the **Filled** box.
4. Click the pull-down arrow of the **Color** box. The Color dialog box appears.

5. Select from one of the 16 basic colors or add new colors from the customized list of colors. To create a customized color, click on one of the boxes in the Customized color box, then click **Customize color**.
6. The color table dialog box appears allowing you to select one of the colors available.
7. Click **OK** in the Color dialog box to go back to the Window Setting dialog box. Click **Close** to return to the EasyBuilder main screen.

Instead of using a solid background color, you can select a pattern that is displayed in the background of a window.

► **To select a background pattern**

1. From the **Window** menu, select **Open Window**. The Open Window dialog box appears.
2. Click on the window that you wish to change the background pattern. Click **Setting**. The Window Setting dialog box appears.
3. In the Background frame section, check the **Filled** box.
4. Click the pull-down arrow of the **Color** box. The Color dialog box appears.
5. Select from one of the 16 basic colors or add new colors from the customized list of colors. To create a customized color, click on one of the boxes in the Customized color box, then click **Customize color**.
6. The color table dialog box appears allowing you to select one of the colors available.
7. Click **OK** in the Color dialog box to go back to the Window Setting dialog box.
8. Click the pull-down arrow of the **Pattern Color** box and select a color. Note: to use the background pattern, the color selected for Pattern Color must be different then the color selected for the solid color.
9. Once a pattern color is selected, click **OK** in the Color dialog box to go back to the Window Setting dialog box.
10. Click the pull-down arrow of the **Pattern** box. The Pattern Style dialog box appears.
11. Select the pattern you wish to use, then click **OK**. The Window Setting dialog box reappears.
12. Click **OK** in the Color dialog box to go back to the Window Setting dialog box. Click **Close** to return to the EasyBuilder main screen.

## Deleting a Window



*Before any window can be deleted from a project, the window must be closed. To close a window, see the section earlier in this chapter on opening windows.*

► **To delete a window**

1. From the Window menu, select **Open Window**. The Open Window dialog box appears.
2. Highlight the window that you wish to delete.
3. Click the **Delete** command. A dialog box appears asking if you want to delete this window.
4. Click **Yes**. The dialog box disappears and the selected window is deleted.
5. Click **Close** in the Open Window dialog box to go back to the EasyBuilder main screen.



*If the Delete button does not appear when the window is highlighted, this means that this is a Window that can't be deleted. Typically, if you are unable to delete a window, it is because it is one of the predefined windows discussed earlier in this chapter.*

## Using Base Windows

Of the four types of windows, base windows are the most commonly used. A base window is used to create a full screen window or a popup (partially sized) window. Popup windows can also be moved about on the OIT display and can overlap each other. The number of objects that can be placed onto each base window is limited only by the total amount of memory available within the OIT.

### How to Display Base Windows

Base windows can be displayed on the OIT screen by using a function key to display the window or by using the PLC to call up the window.



## Using a Function Key

The Function Key Object is a graphic touch object that is placed onto a window to perform an action. Function keys have many purposes that are discussed in later sections, but two actions that a function key can perform are:

- Calling a full screen window
- Calling a popup window

### *Calling a full screen window*

If a function key object is created to call a full screen window, the window that is displayed replaces all other windows that are on display regardless of how many are open; therefore, think of calling a full screen window as performing two actions -- closing any open windows and displaying a full screen window.

#### ► To call a full screen window using a function key object

1. From the Objects menu, select **Function Key**. The New Function Key Object dialog box appears.
2. In the General Tab, click the Change Window option button.
3. In the same frame, enter the **Window No.** you want to call.
4. Select the **Shape** tab, click on the **shape** or **bitmap** checkbox, and then click on the **Shape library**. Select the shape or bitmap you wish to use to represent the function key and click **OK**.
5. Select the **Label** tab.
6. Check **Use label** and select a label for the function key.
6. Click **OK** to return to the main screen of EasyBuilder.
7. Place the function key object where you want it on the window you are editing.

You must select a window that is full size when using the Change Window function.

### *Calling a popup window*

If a function key object is created to call a popup window, the window that is displayed is generally overlaid over all other windows that are on display. Think of calling a popup window as opening another active window for display.

#### ► To call a popup window using a function key object

1. From the Objects menu, select **Function Key**. The New Function Key Object dialog box appears.
2. In the **General** Tab, check the **Display Popup Window option button**. Note that the "Close this popup window when change full-screen window" checkbox will cause the popup to close automatically whenever the base window changes.
3. In the same frame, enter the **Window No.** you want to call.
4. In the **Shape** tab, click on the **shape** or **bitmap** checkbox, then click on the **Shape library** or **Bitmap library**. Select the shape or bitmap you wish to use to represent the function key.
5. In the **Label** tab, select a label for the function key.
6. Click **OK** to return to the main screen of EasyBuilder.
7. Place the function key object where you want it on the window you are editing.

You must select a window that is less than full size when using the Popup Window function.

To display a full screen or popup window, press the touchscreen where the function key object is located. The window displays in the location on the OIT display as determined by the X and Y position in the Window Settings box for that window.



*For more information on using the Function Key object, see the chapter on "Representing Data with Graphics Objects".*

---

## Using the PLC to display a base window

Three objects are used by the PLC to call or display a base window. The PLC Control Object is used by the PLC to display full screen windows. The Direct and Indirect Window Objects are used by the PLC to display popup windows.

### *The PLC Control Object*

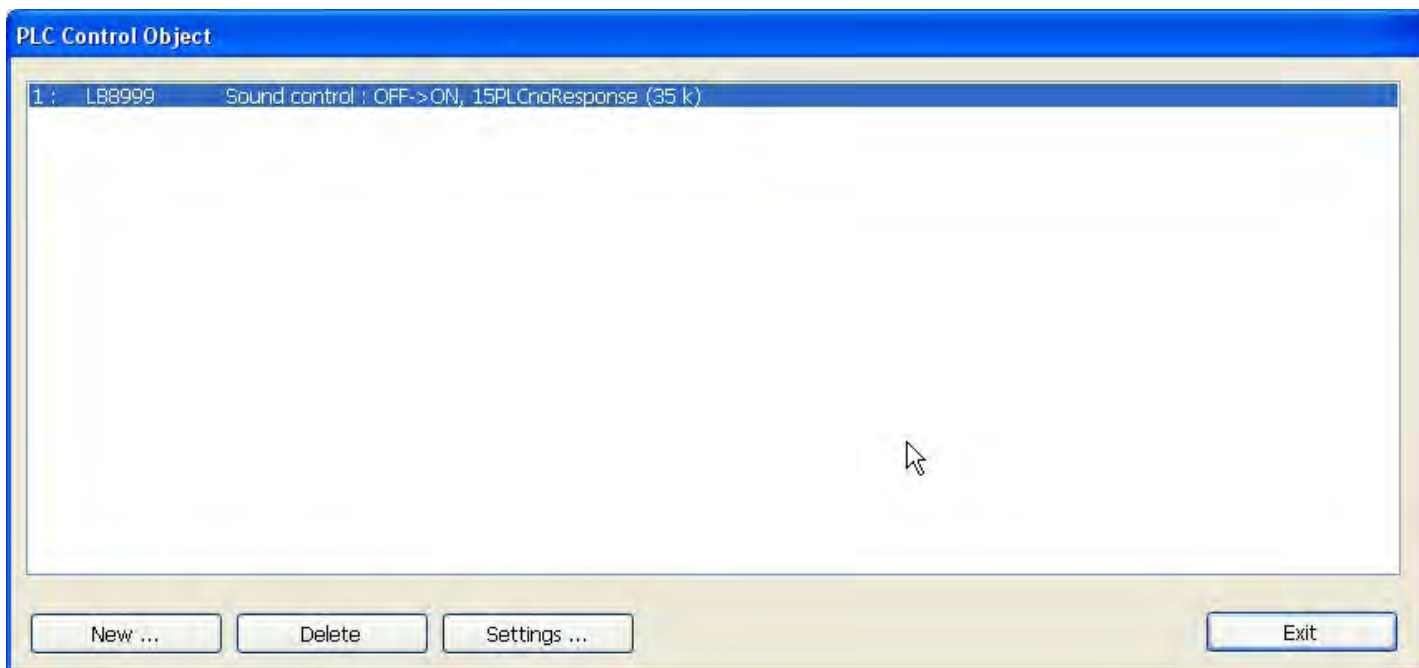
The PLC can display a full screen window by using the PLC Control Object. This object allows the OIT to continuously scan a PLC register to display a full screen window, which corresponds to the number in the PLC register.

Once the OIT displays a full screen that is requested by the PLC Control Object, the OIT will automatically write the number of the requested screen to the next consecutive register. For example, if you assign internal data register LW10 to a PLC Control Object/Change Window and the number 3 is placed into this register, then the OIT will display Screen #3. Finally, it will put the number 3 into LW11. This allows the PLC to confirm that the Screen has been properly displayed by the OIT.

You can create as many PLC Control Objects as you need -- each object is universal and is not dependent upon which OIT screen is currently on display.

► **To call a full screen window using the PLC Control object** 


1. From the Objects menu, select **PLC Control**. The PLC Control Object dialog box appears.



- Click **New...**. The PLC Control dialog appears.

- Select the **Device type** and **Device Address** for the actual PLC address you wish to monitor.

---

 *The OIT will write the number of the newly-displayed window to the next consecutive address after the address specified in this step.*

---

- Select the *format* from the drop down menu (16-bit or 32-bit unsigned, signed or BCD).
- Click **OK** to return to the PLC Control Object dialog box.
- You will see a new entry that lists the PLC address that is monitored by the OIT.

The PLC must only enter window numbers that represent full size windows when using the PLC Control Object function. The OIT automatically closes any open windows before the window called by the PLC Control Object is displayed.

### ***The Direct and Indirect Window Objects***

If you want the PLC to call up a popup window that is to be overlaid on top of other windows already open, then there are two methods used to do this:

- Direct Window Object
- Indirect Window Object

---

 *Unlike the PLC Control Object, the Direct and Indirect Window Objects are only active for the windows into which they are placed.*

---

The Direct Window Object is used to display a popup window using a PLC coil. The OIT continuously reads the value of the PLC coil to determine if it is set to 1. If so, then the predefined popup window is displayed.

The Indirect Window Object is used to display a popup window using a PLC data register. Similar to the PLC Control Object, the OIT continuously monitors the selected PLC data register. It will display any popup window that corresponds to the number placed into the PLC data register by the PLC.

The PLC must reference a window that is less than full size when using the Direct Window and Indirect Window objects.

► To call a popup window using the Direct Window Object



1. From the Objects menu, select **Direct Window**. The New Direct Window Object dialog box appears.

The image shows a dialog box titled 'General' with the following fields and options:

- Description : [Text box]
- Read address section:
  - PLC name : Local HMI [Dropdown]
  - Device type : LB [Dropdown]
  - Address : 0 [Text box]
  - Index : INDEX 0 (16-bit) [Dropdown]
  - System tag
  - Index register
- Attribute section:
  - Style : No title bar [Dropdown]
  - Window No. : 3. Fast Selection [Dropdown]

2. Select the PLC address according to *Device type* from the drop down menu and then enter the *address* .
3. Select the *Window No.* you want to call from the drop down menu under the Attribute heading.
4. Click **OK** to return to the main screen of EasyBuilder. The mouse cursor will have a square object that represents the size of the popup window.
5. Place the Direct Window object where you want the popup window to appear.
6. You can resize the object after you have placed it by clicking on the object and dragging on the sizing handles.

► To call a popup window using the Indirect Window Object



1. From the Objects menu, select **Indirect Window**. The New Indirect Window Object dialog box appears.

The dialog box is titled 'General' and contains the following fields and options:

- Description : [Text box]
- Read address:
  - PLC name : Local HMI [Dropdown]
  - Device type : LW [Dropdown]
  - Address : 0 [Text box]
  - System tag
  - Index register
  - 16-bit Unsigned [Dropdown]
- Attribute:
  - Style : No title bar [Dropdown]

2. Select the PLC address according to *Device Type* and *Address*.
3. Select the *format* from the drop down menu.
4. Click **OK** to return to the main screen of EasyBuilder. The mouse cursor will have a square object that represents the size of the popup window.
5. Place the Indirect Window object where you want the popup window to appear.
6. You can resize the object after you have placed it by clicking on it and dragging on the sizing handles.

## Tips and Suggestions

Having trouble deciding which method to use to display a base window? Here are some suggestions:

- **If you want to clear the OIT display of all open windows and display a new full screen window**, then use the PLC Control Object function or select the Change Window option in the Function Key Object. The PLC Control Object is global (meaning it does not matter which windows are currently on display), so the OIT will always monitor the PLC address that you have selected. The Function Key Object can be made local to one or more windows by placing the object on only those windows. It can also be made global (meaning the OIT operator can always change to this window no matter which window is currently displayed), by placing the Function Key Object on the Common Window or the Fast Selection Window in the Task Bar, (more about these later in this chapter).

- **If you want to display a popup window on one particular full screen window**, then try using the Popup Window option in the Function Key Object. This is particularly useful if you want the OIT operator to control the ability to display the popup window. If you want the PLC to determine when to display the popup window, then try using the Direct Window object.
- **If you want to display a popup window that can be shown on any full screen window**, then try using the Popup Window option in the Function Key Object. You should place the Function Key Object in the Fast Selection Window with the Task Bar enabled, (more information is available on the Task Bar later in this chapter). This will allow the OIT operator access to that popup window regardless of which full screen window is displayed. If you want to display the popup window only when some condition in the PLC has occurred, then try using the Direct Window object on the Common Window, (more about the common window later in this chapter).
- **If you want to display one of many possible popup windows on any full screen window**, then try using the Indirect Window object on the Common Window. You can then let the PLC determine which popup window should be displayed or you can create several Set Word objects to allow the OIT operator to select which window to look at, (more about the Set Word object in a succeeding chapter).

## Returning to a previous window

You can configure a function key to display the full-sized base window that was on the OIT screen before the currently shown window. Perform the following:

### ► Create a function key to return to previous window

1. From the **Objects** menu, select **Function Key**. The New Function Key Object dialog box is displayed.
2. Select the Return to Previous option button.
3. Configure the rest of the function key, and then click **OK**.
4. Place the function key object onto the full-sized base screen.



*Placing a 'return to previous' function key onto a popup window will not work.*

---

## Using a function key to close a window

You can configure a function key to close any popup window that is currently displayed on the OIT screen, and was called by a function key:

### ► Create a function key to close a window

1. From the Objects menu, select **Function Key**. The New Function Key Object dialog box is displayed.
2. Select the Close Window option button.
3. Configure the rest of the function key, and then click **OK**.
4. Place the function key object onto the popup window.



*Placing a 'close window' function key onto a full-size base screen will not work. To close a direct window, turn off the controlling bit. To close an indirect window, write a 0 to the controlling register.*

---

## Using the Common Window

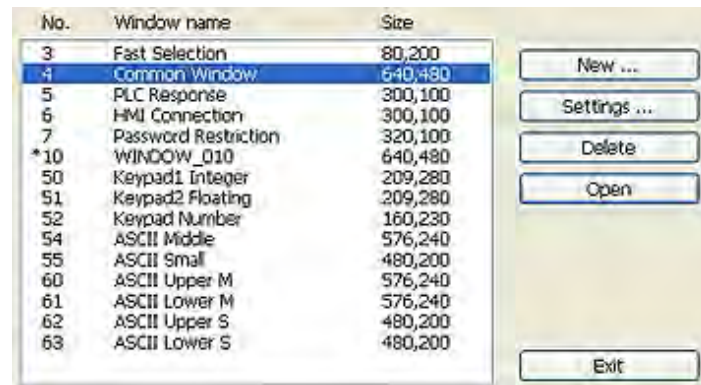
Your project might require that some data be displayed on the OIT screen at all times, regardless of which window(s) are displayed. For example, you may want to display a company logo on the OIT screen at all times, or you may want to display some critical data or an alarm message which should be seen no matter what windows are displayed.

Using base windows to display this information requires that you configure every full screen base window with the same graphics object. The common window however, is a predefined window in the OIT that you can enable to display this information. When created, the common window always operates in the background as a full screen window that overlays any full screen base window displayed.

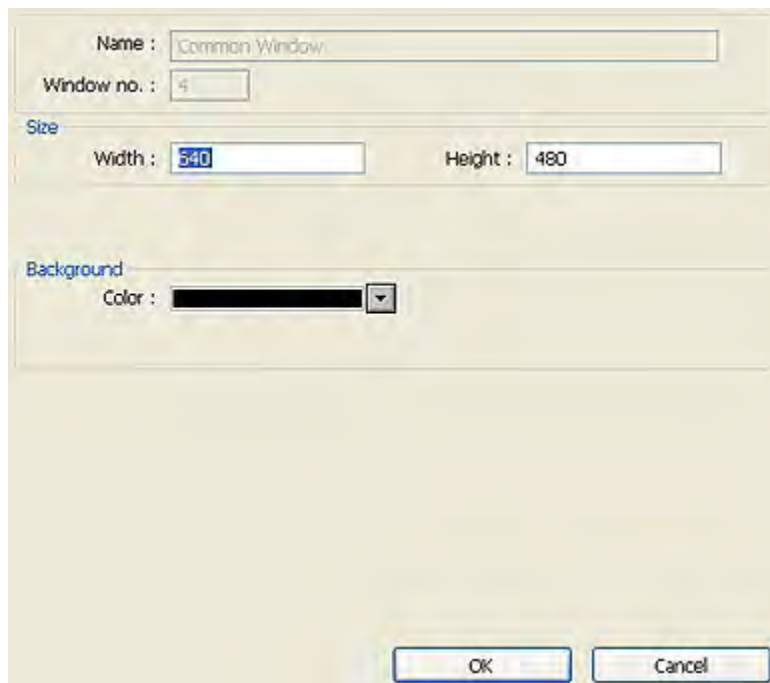
Whenever you create a new project, the common window (Window #4) is automatically created.

► **To Access the common window**

1. From the Window menu, select **Open Window**. The Open Window dialog box appears.



2. Highlight Window #4, **Common Window** and click **Settings**. The Window Setting dialog box appears.



3. As you can see, some of the parameters for a common window are disabled and cannot be changed. Window #4 is always reserved for the common Window. Although the Background settings are active, none of these settings affects the operation of the common window.
4. Press **OK**. The Open Window dialog box reappears with the Common Window selected.
5. If you wish to open the common window, click **Open**. Otherwise, click **Close** to return to the main screen of EasyBuilder.

### Displaying the common window above/below the base screen

This setting resolves the conflict that can occur if a graphics object on the common window occupies the same space on the OIT display as a graphics object on a base screen. Using the **Above base screen** attribute forces the graphics object on the common window to cover the base window graphics object. Using **Below base screen** has the opposite effect.



Common Window

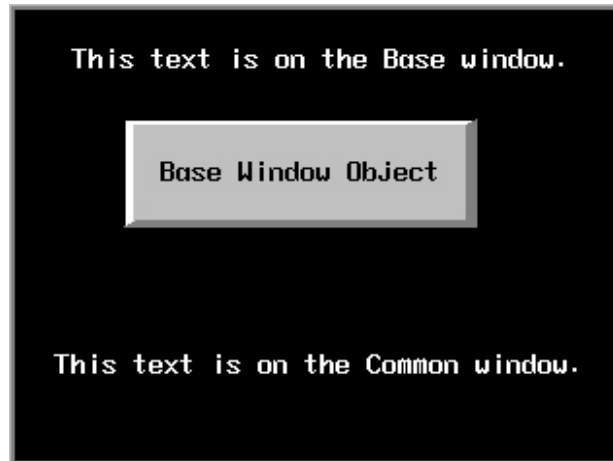


Base Window

When **above base screen** is set, the result is:



When **below base screen** is set, the result is:



► **To set the above/below base screen option for a common window**

1. From the Edit menu, select **System Parameters**. The Set System Parameters dialog box appears.
2. Select the **General** tab. In the Option frame box under common window, select either **Above** base window or **Below** base window for the Attribute box.
3. Click **OK** to return to the main screen of EasyBuilder.



*Active graphics objects (objects that display information or graphics according to a data value in a PLC register or coil) take precedence over passive graphics objects*



(objects such as circles, lines, rectangles, etc.). Therefore, a Set Word object on a base window will cover a Rectangle object on a common window which occupies the same space even with the **Above base screen** attribute enabled.

### Changing the Active Common Window

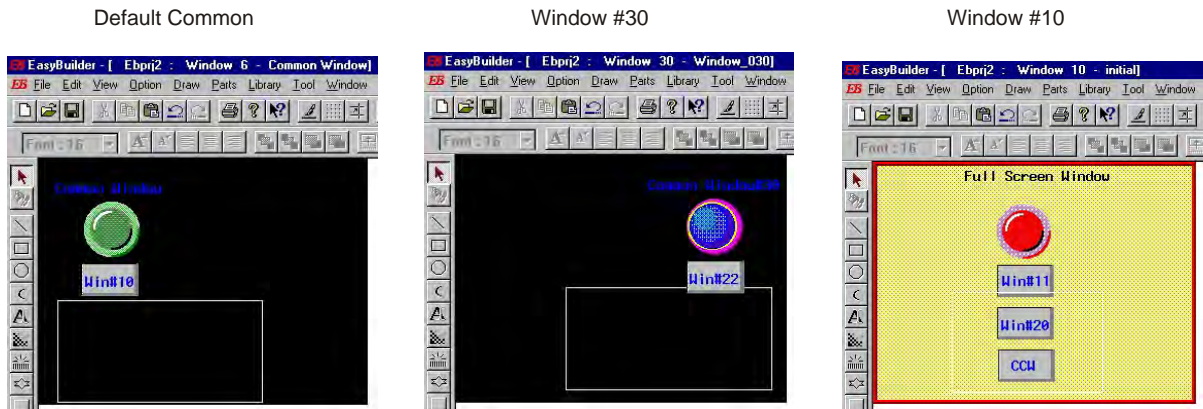
Although only one common window can be on the OIT display at one time, you do have the ability to create multiple common windows by using base windows as common windows. Then, with the help of a Function Key object, the OIT operator can change the active common window.

The ability to change the common window adds more flexibility to your project should you need it. For example, you may have a series of full screen windows which all need a keypad for entry. Another series of full screen windows may require a common alarm message. By changing the common window with each series of windows, you can customize each common window to contain only the graphics objects that are needed.

► **To change the active common window**

1. Create a common window.
2. Create a base window that is full screen. This will be used as another common window. Note that when a base window is used as a common window, the **Start Pos.**, **Style**, **Frame**, and **Background** settings are ignored by the OIT. For this example, let's use Window#30 as the alternate common window.
3. Create another full screen base window that can be displayed along with the common windows, (such as a startup window). Let's use Window#10, the initial window, for our example.
4. On Window#10, create a Function Key object that is used to change the common window. From the **Objects** menu, click **Function Key**. The New Function Key Object dialog box appears.
5. On the General tab, select the Change Common Window option button.
6. In the same frame, select **30** from the Window no drop down menu.
7. Click the **Shape** tab. Click **Use Shape....** and then click on the **Shape Library...** command. Select **Shape#19** from the **button1** library. Click **OK** to return to the Create Function Key Object dialog box.
8. Click the **Label** tab. Click the **Use label** checkbox.
9. For our example, type **CCW** (meaning **Change Common Window**) into the Content box. Click **OK** to exit the Create Function Key Object dialog box and go back to the EasyBuilder main screen.
10. Place the new Function Key object somewhere on Window #10.

Below is an illustration of what the default common window, Window#30 (alternate common window), and Window#10 (startup window) might look like:



When the OIT initializes, it displays the startup screen (Window #10) and the default common window.



To change common windows, we press the CCW function key that we have configured to change the common window to Window#30.



Window#30 remains the common window until another function key object is pressed that changes the common window or until the OIT is reset.

## Using the Fast Selection Window

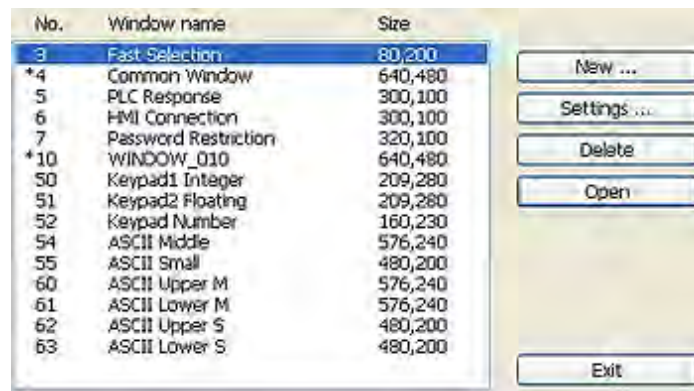
In the last section, you read that common windows are great for displaying information that should be displayed all the time, regardless of which base windows are active. You may also create projects that require a window that is always accessible (like a common window) *but not always displayed*. For example, you may want a numeric keypad available for any data entry; however, a keypad takes precious space on the OIT display. Ideally it should appear on screen only when a key was pressed and then, by pressing another (or the same) key, the keypad should disappear.

This is essentially the purpose of the Fast Selection window. The Fast Selection window can also be used as a menu key that allows the OIT operator to rapidly switch screens.

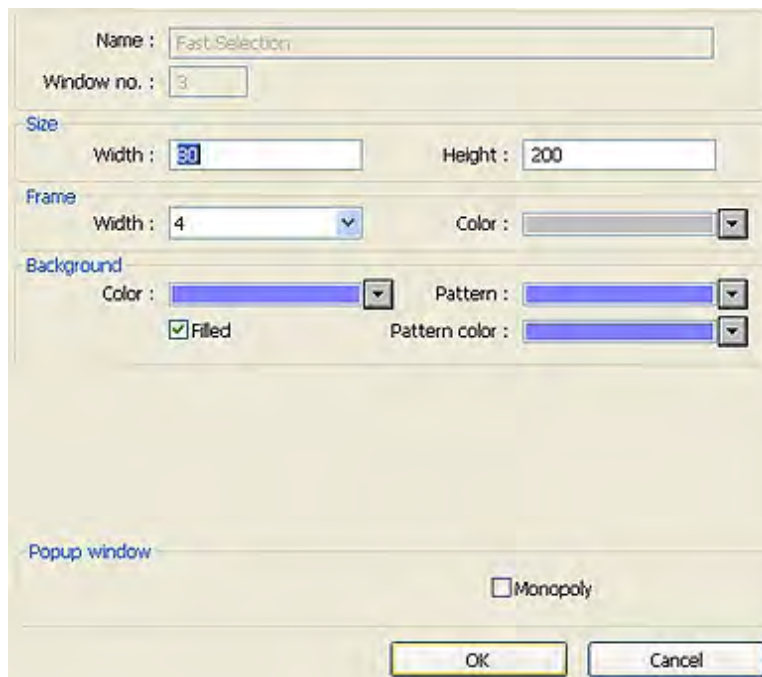
Whenever a new project is created, the fast selection window (Window #3) is automatically created.

► **To open a fast selection window:**

1. From the Window menu, select **Open Window**. The Open Window dialog box appears.



2. Highlight the Fast Selection Window and click **Settings**. The Window Setting dialog box appears.



3. The **Name:** and **Window No.:** are reserved to identify the Fast Selection window. The other parameters can be modified, (see the Window Settings section earlier in this chapter for more information). Notice that the default size is Width=80 and Height=200. The default was selected to create a 'sidebar' that contains function keys to display other windows. The example below illustrates how the Fast Selection Window can be used for this purpose.
4. Press **OK**. The Open Window dialog box reappears with the Fast Selection window selected.
5. If you wish to open the fast selection window, click **Open**. Otherwise, click **Close** to return to the main screen of EasyBuilder.

## Using the Fast Selection Key

The fast selection key is used to display the Fast Selection window. Pressing the key again causes the Fast Selection window to close.



## Creating the Fast Selection Key

The fast selection key must be activated in System Parameters in order to have it display on the OIT. The fast selection key will always display on the operator interface.

### ► To display the fast selection key

1. Select **Edit-System Parameters**. The System Parameters Setting Dialog appears. Select the General tab.



2. In the Fast selection button section, select **Enable** from the Attribute drop down menu.
3. Select either *right* or *left* from the Position drop down menu to select whether the button will appear in the right lower corner or left lower corner of the display.
4. Click **Settings...** Click the **Use shape** checkbox, and then select **Shape Library...**
5. Select a *shape* for your fast selection button and click **OK**.

6. Click **OK**.

### Changing screens using the Fast Selection window

One of the more useful purposes of the Fast Selection window is to use it as a ‘menu’ key for switching screens on the OIT display. The following example illustrates how you might create a Fast Selection window for this purpose. The example steps you through creating six full-screen base windows, enabling the Task Bar, and creating a Fast Selection window to call up the six screens.

We will start with the following six screens.



Next, create a new Fast Selection window using default settings. For this example, we will create six Function Key objects that change windows. Configure each **Function Key** object with the **Change Window** attribute and place the object into the Fast Selection window.



Finally, the Task Bar must be enabled, (see the next section for instructions). After downloading the project file to the OIT, the following initial screen should be shown.



Press the Fast Selection key to activate the Fast Selection window.



Press one of the Function Keys to switch to another full screen window.



Finally, pressing the Fast Selection Key will remove the Fast Selection Window until it is needed again.



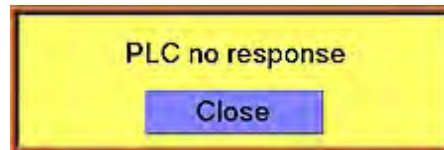
As you can see, the Fast Selection window, though not seen, is always readily available.



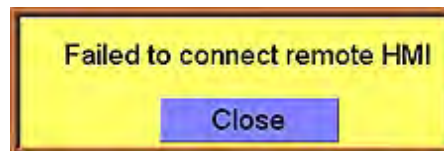
*The Fast Selection window can only call full screen windows not popup windows. Therefore, you cannot use Function Key objects with the Popup Window attribute in Fast Selection windows.*

## System Message Window

EasyBuilder has two Windows designated as System Message windows - Window #5 is designated as the PLC Response Window that pops up automatically when the OIT loses communications with the PLC.



Window #6 is designated as the HMI Response window, which will pop up automatically with the OIT loses communications with a another OIT that is remotely connected.



Both of these windows have been pre-configured with automated messages and a function key to close the window, so no changes are necessary.

**Please Use This Page For Your Notes:**



## Chapter 8 - Creating Graphic Objects

### Drawing Objects

To display any graphics objects on the OIT screen you must either create them using the drawing tools included with EasyBuilder, import them from another applications program, or select from the many choices available in the graphics libraries included with EasyBuilder.

This chapter shows you how to create, use, and save graphics objects in EasyBuilder. You have learned from previous chapters how graphics objects are placed onto windows. In this chapter, we concentrate on how to create graphics objects.

#### Using the Drawing Tools

Several drawing tools are provided in EasyBuilder to make it easier to create graphics objects. With these tools, you can create simple geometric shapes such as straight lines, circles, rectangles, and polygonal shapes. You can also combine several various geometric shapes to form complex shapes that can be stored into one of the graphics libraries for later retrieval.

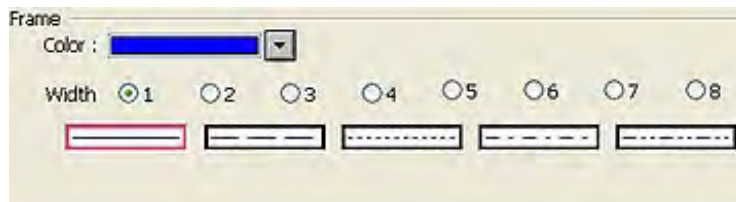
#### The Line Tool

Use the Line tool to create straight lines on a window. Each line that you create has three parameters associated with it: length, thickness, and color.

##### ► To create a line



1. From the Draw menu, click **Line** or click the **Line** icon in the Draw toolbar. The Properties dialog box appears.



2. Click the pull down box from the Color box. The Color dialog box appears.
3. Click on the appropriate color box, then click **OK**. The color box should reflect what you have chosen.
4. Click on the line thickness that you want to use.
5. Move the mouse cursor over to the work area of EasyBuilder onto the window that you are currently editing. The mouse cursor has changed from an arrow to a crosshair cursor.
6. Click to mark the beginning of the line. Move the mouse to the location where the end of the line should be. Click the mouse cursor again to mark the end of the line. The line object is formed and displayed on the window screen with small white square blocks around the boundaries of the line.
7. To continue creating more lines repeat the last step. To move the line just created, click on the **Select** option from the **Edit** menu or click the **Select** icon in the **Draw** toolbar. Move the line to the preferred location.
8. To change any of the attributes, double-click on the line to display the **Line Object's Attribute** dialog box.

#### The Rectangle Tool

The rectangle tool is used to create rectangles or squares. Each rectangle created has four parameters associated with it: size, thickness, color, and filled.

### ► To create a rectangle



1. From the Draw menu, click **Rectangle** or click the **Rectangle** icon in the Draw toolbar. The Attributes dialog box appears.
2. If the interior of the rectangle is to be filled with a color, check the **Filled** box. Two additional settings appear.
3. If you want to fill the rectangle with a solid color, then select a color for the **Interior** box. Click **Pattern Style** to display the **Pattern Style** dialog box. Select the upper left-most pattern for a solid color. If you want to use a pattern for the filled section of the rectangle, then select from the available patterns, and click **OK** to exit the **Pattern Style** dialog box. In the **Attributes** dialog box, select the **Pattern** color.
4. In the Frame section, click the pull-down arrow of the **Color** box and select a color. Click **OK** in the Color dialog box to go back to the Attributes dialog box.
5. Click on the line thickness that you want to use.
6. Move the mouse cursor over to the work area of EasyBuilder onto the window that you are currently editing. The mouse cursor has changed from an arrow to a crosshair cursor.
7. Click to mark a corner of the rectangle. Move the mouse to where the opposite corner of the rectangle should be. Click the mouse cursor again to mark the corner of the rectangle. The rectangle object is formed and displayed on the window screen with small white square blocks around the perimeter of the rectangle.
8. To continue creating more rectangles repeat the last step. To move the rectangle just created, click on the **Select** option from the **Edit** menu or click the **Select** icon in the **Draw** toolbar. Move the rectangle to the preferred location.
9. To change any of the attributes, double-click on the rectangle to display the **Rectangle Object's Attribute** dialog box.

## The Ellipse/Circle Tool

The ellipse tool is used to create ellipses or circles. Each ellipse created has four parameters associated with it: size, thickness, color, and filled.

### ► To create an ellipse



1. From the Draw menu, click **Ellipse/Circle** or click the **Ellipse/Circle** icon in the Draw toolbar. The Attributes dialog box appears.
2. If the interior of the ellipse is to be filled with a color, check the **Filled** box. Two additional settings appear.
3. If you want to fill the ellipse with a solid color, then select a color for the **Interior** box. Click **Pattern Style** to display the **Pattern Style** dialog box. Select the upper left-most pattern for a solid color. If you want to use a pattern for the filled section of the ellipse, then select from the available patterns, and click **OK** to exit the **Pattern Style** dialog box. In the **Attributes** dialog box, select the **Pattern** color.
4. In the Frame section, click the pull-down arrow of the **Color** box and select a color. Click **OK** in the Color dialog box to go back to the Attributes dialog box.
5. Click on the line thickness that you want to use.
6. Move the mouse cursor over to the work area of EasyBuilder onto the window that you are currently editing. The mouse cursor has changed from an arrow to a crosshair cursor.
7. Click to mark a corner of the ellipse. Move the mouse to where the opposite corner of the ellipse should be. Click the mouse cursor again to mark the ellipse. The ellipse object is formed and displayed on the window screen with small white square blocks around the perimeter of the ellipse.
8. To continue creating more ellipses repeat the last step. To move the ellipse just created, click on the **Select** option from the **Edit** menu or click the **Select** icon in the **Draw** toolbar. Move the ellipse to the preferred location.
9. To change any of the attributes, double-click on the ellipse to display the **Ellipse Object's Attribute** dialog box.

## The Arc Tool

The arc tool is used to create arcs. Each arc created has three parameters associated with it: size, thickness, and color.

► To create an arc



1. From the Draw menu, click **Arc** or click the **Arc** icon in the Draw toolbar. The Attributes dialog box appears.
2. In the Frame section, click the pull-down arrow of the **Color** box and select a color. Click **OK** in the Color dialog box to go back to the Attributes dialog box.
3. Click on the line thickness that you want to use.
4. Move the mouse cursor over to the work area of EasyBuilder onto the window that you are currently editing. The mouse cursor has changed from an arrow to a crosshair cursor.
5. Click to mark a corner of the arc. Move the mouse to where the opposite corner of the arc should be. Click the mouse cursor again to mark the arc. The arc object is formed and displayed on the window screen with small white square blocks around the perimeter of the arc.
6. To continue creating more arcs repeat the last step. To move the arc, click on the **Select** option from the Edit menu or click the **Select** icon in the **Draw** toolbar. Move the arc to the preferred location.
7. To change any of the attributes, double-click on the arc to display the **Arc Object's Attribute** dialog box.

## The Polygon Tool

The polygon tool is used to create polygons. Each polygon created has four parameters associated with it: size, thickness, color, and filled.


► To create a polygon



1. From the Draw menu, click **Polygon**. Or click the **Polygon** icon in the Draw toolbar. The Attributes dialog box appears.
2. If the interior of the polygon is to be filled with a color, check the **Filled** box. Two additional settings appear.
3. If you want to fill the polygon with a solid color, then select a color for the **Interior** box. Click **Pattern Style** to display the **Pattern Style** dialog box. Select the upper left-most pattern for a solid color. If you want to use a pattern for the filled section of the polygon, then select from the available patterns, and click **OK** to exit the **Pattern Style** dialog box. In the **Attributes** dialog box, select the **Pattern** color.
4. In the Frame section, click the pull-down arrow of the **Color** box and select a color. Click **OK** in the Color dialog box to go back to the Attributes dialog box.
5. Click on the line thickness that you want to use.
6. Move the mouse cursor over to the work area of EasyBuilder onto the window that you are currently editing. The mouse cursor has changed from an arrow to a crosshair cursor.
7. Click to mark a corner of the polygon. Move the mouse to where another corner of the polygon should be. Click the mouse cursor again to mark the corner. Continue clicking each corner until it is time to connect the last corner to the first corner of the polygon. Right click to complete the polygon. The polygon object is formed and displayed on the window screen with small white square blocks around the perimeter of the polygon.
8. To continue creating more polygons repeat the last step. To move the polygon, click on the **Select** option from the Edit menu or click the **Select** icon in the **Draw** toolbar. Move the polygon to the preferred location.
9. To change any of the attributes, double-click on the polygon to display the **Polygon Object's Attribute** dialog box.

## The Scale Lines Tool

The Scale Lines tool is used to create scales. Each scale created has four parameters associated with it: size, thickness, color, and filled.

► **To create a scale** 

1. From the Draw menu, click **Scale**. Or click the **Scale** icon in the Draw toolbar
2. Move the mouse cursor over to the work area of EasyBuilder onto the window that you are currently editing. The mouse cursor has changed from an arrow to a crosshair cursor.
3. Click to mark a corner of the scale. Move the mouse to where the opposite corner of the scale should be. Click the mouse cursor again to mark the corner of the scale. The scale object is formed and displayed on the window screen with small white square blocks around the perimeter of the scale.
4. From the Edit menu, select **Change Attribute**. The Scale Object's Attribute dialog box appears.
5. In the Frame section of the Style tab, click the pull-down arrow of the **Color** box and select a color. Click **OK** in the Color dialog box to go back to the Scale Object's Attribute dialog box.
6. Click on the line thickness that you want to use.
7. In the Scale section, select from one of the six **Styles**:



8. Enter the number of divisions in the **Division:** box. For **Up**, **Down**, **Full**, and **3/4 Size** enter the **Meter Length**.
9. Adjust the size and position of the scale in the Profile tab, if necessary.
10. Click **OK**. The Scale Object's Attribute closes and the main screen of EasyBuilder reappears with the Scale Object displayed. Scale lines are most often used when creating bar graphs, scale meters, and trend graphs.

## The Free Line Tool

The Free Line Tool allows you to draw lines in a free form fashion. .

► **To draw a free-form line** 

1. From the Draw menu, select **Arbitrary line** or click on the **Free line** icon on the draw tool bar. The Properties dialog appears.
2. Configure any *line properties* including color and line width.
3. Move your cursor over EasyBuilder's design space. The cursor will turn into a crosshair. When the crosshair is placed where you would like to begin your free line, left click your mouse.
4. Move your mouse along the path that you would like to draw your free line. This can include any path - including a squiggly line, crossing over itself, etc. When you have completed your line, right click your mouse button.
5. You may repeat steps 1-4 to continue drawing free lines, or you may click the **Select** icon on the Draw toolbar, and then double click on your free lines to change any of the objects attributes.

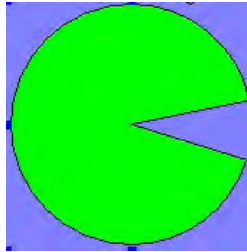
## The Pie Tool

The pie tool allows you to draw a section of a circle -from 1 -360° .

► **To use the pie tool** 

1. Select **Pie** from the Draw menu or select the **Pie** icon from the draw tool bar. The Properties dialog appears.
2. Configure any properties for your shape, including line and fill.

3. Move your cursor over the EasyBuilder work area. A crosshair will appear. Move your crosshair to where you would like your shape to appear.
4. Click and hold your mouse cursor while dragging to form a circle. When the circle is the size you wish it to be, release the left mouse button.
5. Now, left click again, and a line will appear. This is the starting point of your pie. Move along the arc of the circle until you get the shape of the pie that you want and left click again.



6. You may continue creating pie shapes, or click on the select icon and select the pie to resize it or configure its attributes.

## The Linked Line Tool

The linked line tool allows you to draw lines that are linked end to end.

### ► To use the linked line tool

1. Select **Link line** from the Draw menu or select the **Linked line** tool from the draw toolbar. The Properties dialog appears.
2. Configure any properties for your line.
3. Move your cursor over the EasyBuilder work area. A crosshair will appear. Move your crosshair to where you would like to begin your lines.
4. Click your mouse to begin your line. Move your mouse to the point where you would like your first line to end and your next line to begin and click your mouse again. Repeat this step until you are done drawing lines. Right click your mouse to stop drawing lines.
5. You may continue creating linked lines or click on the select icon and select the linked lines to move them or configure their attributes.

## Using Text

EasyBuilder uses Windows TrueType Fonts. This means that you can use any fonts that you have available to you in your Microsoft Windows, including Unicode characters for foreign fonts.

Text sizing is available from 8 to 144 pixels, and text can be blinking, underlined or *italic*. Text can be used in object labels or placed onscreen in text boxes.

► To create a text box 

1. From the Draw menu, click **Text**. Or click the **Text** icon in the Draw toolbar. The New Text Object dialog box appears.



2. Select the *font* from the Font: drop down menu.
3. Select *color* and *size* from their respective drop down menus.
4. Select the type of *justification* in Align: box. If you'd like the text to blink, select it in the Blink: drop-down menu. You may also check either *Italic* or *Underline* for style if you wish.
5. If you'd like the message to be scrolling, select the *direction* in which you'd like the movement to occur, and then adjust the movement *speed*. Check the continuous checkbox if you'd like the message to scroll continuously, as opposed to just once.
6. In the Content: box, enter the *text* that you would like to appear in the text box.
7. Click **OK**. The main screen of EasyBuilder reappears with the outline of the text box tagged to the cursor in the upper left corner of the work area.
8. Move the text box to the area you wish to place it on the window and click.
9. The text box appears with small square boxes around the perimeter. Click on another area of the window to end the edit session or double-click to reedit the text box parameters.

► To change the font size 

1. By clicking onto a text box, you can easily change the font size by using the **Font:** box or the **Enlarge** or **Reduce** buttons on the Manager toolbar.
2. Click on the pull-down box of the **Font:** box and select one of the font sizes; or,
3. Click the **Enlarge** button to increase the size of the text box by one font size.
4. Click the **Reduce** button to decrease the size of the text box by one font size.

5. Note that if the text box overlaps one of the borders of the window when enlarged, then the command to enlarge the text box is ignored.

## Predefined Shapes and Pictures

In addition to using the drawing tools to create your own graphics, EasyBuilder provides several types of predefined graphics objects: shapes and pictures (bitmaps, JPGs and GIFs including animated GIFs). These graphics objects can be stored in libraries that come with EasyBuilder, (more on libraries in the next section). All graphics can be used for the same purpose -- to display a predefined complex graphics object on the OIT screen. The difference between the types of graphics is how they are composed and stored.

Shapes are vector-based graphics which are stored in files that contain dimensional information about the shape. Shapes are actually composed of several simpler graphics like the ones created by the drawing tools. Because of this, they can usually be more easily modified than bitmaps.

Pictures are pixel-based graphics which are stored in files that contain information about each pixel to compose a graphic. The pixel is the smallest possible detail that you can change on a screen. Because of this, although these graphics are more difficult to modify, they offer the most flexibility in customizing the graphic object.


In most cases, you won't need to know the difference between a shape graphic and a pixel-based graphic. You will base your selection on what looks good. However, keep the following comparisons in mind.

	Shapes	Pictures
Ease of Creation	Can be created using EasyBuilder	Must be created using a desktop graphics application such as Microsoft Paint™
Memory Requirements	Low	High
Quality of picture after changing size	Good	Fair to Bad
Time required to display on OIT screen	Quick if not too complex	Quick if not too large

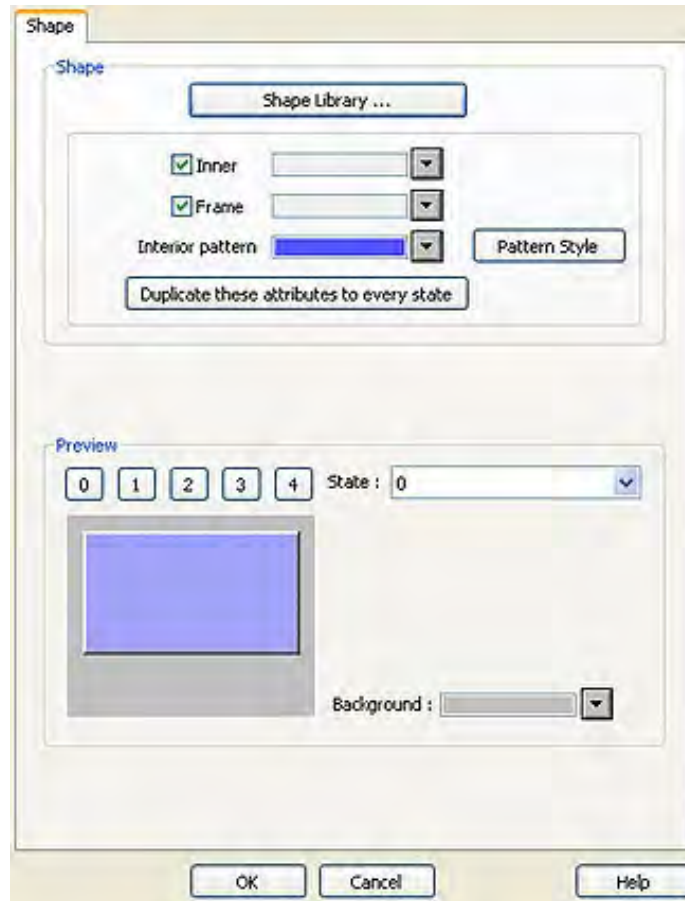
Shapes and pictures are used in both passive and active graphics objects. Each shape or picture can be resized or moved just like any drawing object. The following examples show how to place a pre-defined shape or a picture on the OIT screen as a passive object. Using shapes and picture with active graphics objects is covered in Chapter 8 "Representing Data with Graphics Objects". Information on creating new shapes and bitmaps and storing them in libraries will be covered in a later section of this chapter.

### *Using a Predefined Shape*

You can select from many shapes that are included in the EasyBuilder configuration software. Shapes can be used for active or passive objects. The following procedure describes using a passive shape object

► To use a shape 

1. From the Draw menu, click **Shape**. Or click the **Shape** icon in the Draw toolbar. The New Shape Object dialog box appears.





- Click the **Shape library...** button. The Shape Library dialog box appears.



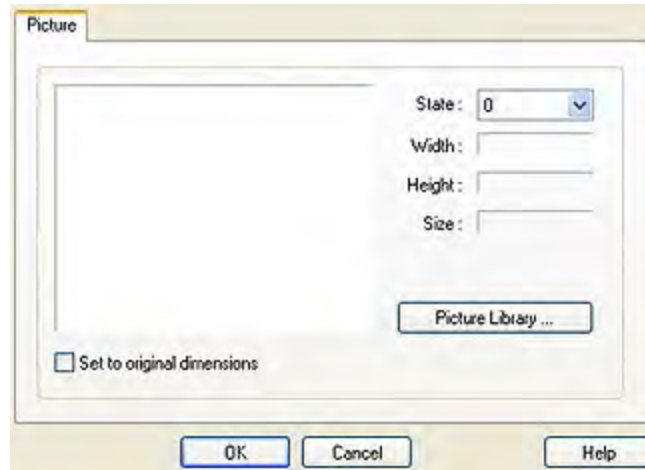
- Choose one of the available *libraries* in the Library: section.
- Use the scroll bar to view available shapes. Click on the shape that you want.
- Click **OK**. The New Shape Object dialog box reappears. The shape that you selected will be displayed in this dialog box.
- Click **OK**. The main screen of EasyBuilder reappears with an outline of the shape object in the upper left corner of the work area.
- Move the shape to the location desired and click. The shape appears with small square boxes around the perimeter of the shape.
- The shape can be resized or you can edit the shape's parameters by double-clicking to enter the **Shape Object's Properties** dialog box, or by dragging on the resizing handles on the corners of the object.

### **Using a Predefined Picture**

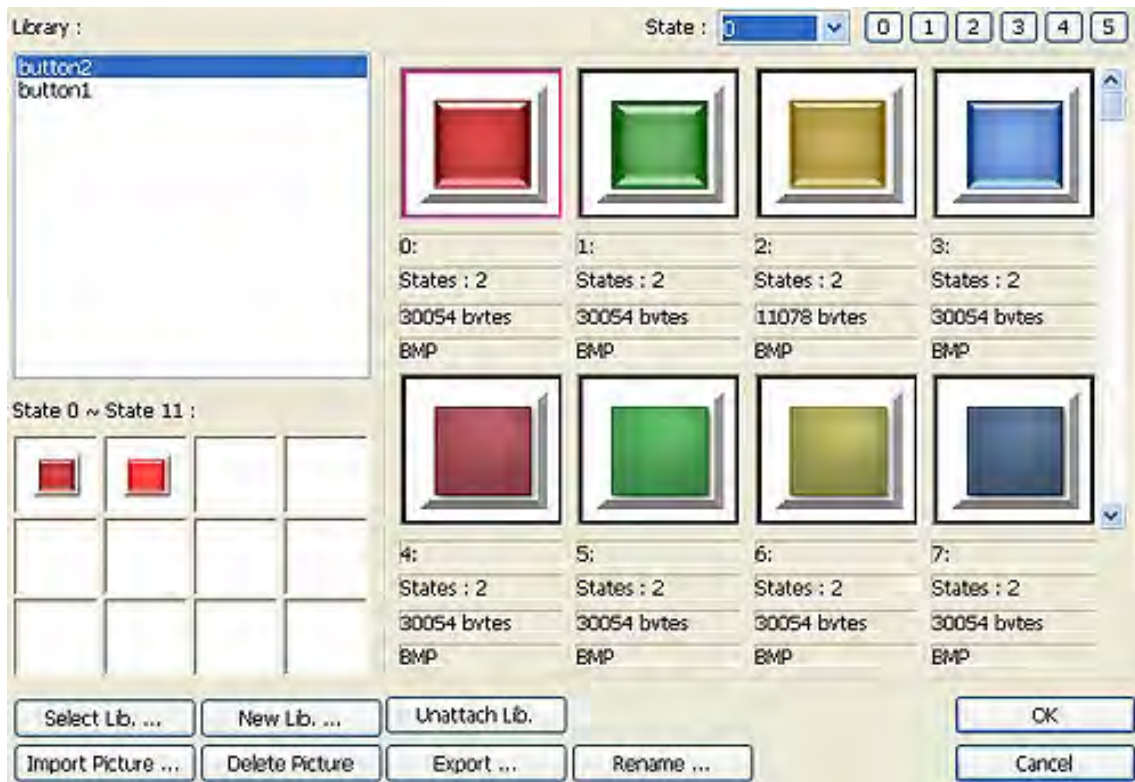
You can select from many pictures that are included in the EasyBuilder configuration software. Pictures can be used for active or passive objects. The following procedure describes using a passive picture object.

► To use a picture 

1. From the Draw menu, click **Picture**. Or click the **Picture** icon in the Draw toolbar. The New Picture Object dialog box appears.



2. Click the **Picture library...** button. The Picture Library dialog box appears.



3. Choose a *library* from the Library: section by double clicking.
4. Use the scroll bar to view available bitmaps. Click on the bitmap that you want.
5. Click **OK**. The New Picture Object dialog box reappears. The picture that you selected will be displayed in this dialog box.
6. Click **OK**. The main screen of EasyBuilder reappears with an outline of the bitmap object in the upper left corner of the work area.

7. Move the picture to the location desired and click. The picture appears with small square boxes around the perimeter of the bitmap.
8. The bitmap can be resized or you can edit the bitmap's parameters by double-clicking to enter the **Bitmap Object's Properties** dialog box.

## Graphics Libraries

Graphics Libraries are files that come with EasyBuilder which contain graphics shapes and pictures. Many predefined shapes and pictures are included with EasyBuilder. In addition, you can create new pictures or shapes and add them to the libraries. You can even create new libraries as you need them. Each library contains a maximum of 48 shapes or pictures. Each shape or picture has a maximum of 255 'states' or pictures tied to it.

EasyBuilder allows a maximum of ten bitmap libraries and ten shape libraries to be open per project file. Finally, a group library can be used to store several graphics objects (such as a keypad) to be easily used when needed. For example, you may create a bar graph that could be used in several different projects. The bar graph can be stored into the group library for easy retrieval.

All libraries are accessed from the **Library** menu or by clicking the appropriate icon in the **Standard** toolbar.

### What are 'states'?

Many of the graphics objects (ex. Set Word Object or Moving Shape Object) use states to determine which picture to display. Shapes and pictures can be configured with up to 255 different states.

Each shape can be configured to show a different picture depending upon a 'state'. States are used when the shape is tied to an active object that is monitoring a value in a PLC. The state corresponds to the actual value in the PLC register. For instance, you would use a two-state shape to represent a Bit Lamp object. You can also use a different picture of a shape for a passive object. Click the **State:** pull-down box in the Shape or Bitmap library and select a different number. Most of the predefined shapes in the libraries have at least two states.

### Using Shape Libraries

When you start a new project, two libraries (button1 and button2) are automatically open.

► **To display the Shape Library dialog box** 

1. From the Library menu, click **Shape**, then click **Call up library**. Or click the **Call up Shape Library** icon from the Standard toolbar. The Shape Library dialog box appears.



The Shape Library provides several options which are discussed in the following sections.

## Accessing and Creating Shape Libraries

You can open, close, or even create new shape libraries easily using the Shape Library dialog box.

### ► To open other shape libraries

1. With the Shape Library dialog box displayed, click **Select Lib...** The Open dialog box appears.
2. Select which library you would like to open, then click **Open**. The Shape Library dialog box reappears with the selected library displayed.

If the project has ten libraries open, you may want to close a library so that another one may be opened.

### ► To close or ‘unattach’ an open shape library

1. With the Shape Library dialog box displayed, choose from one of the open libraries of shapes in the Shape: section.
2. Select which library you would like to close, then click **Unattach Lib**. A dialog box appears asking if you really want to unattach the selected library. Click **Yes**.
3. The Shape Library dialog box reappears and the library is removed from the **Shape library:** list.

You have the option of creating a new library. This can be very useful when organizing shapes.

### ► To create a new shape library

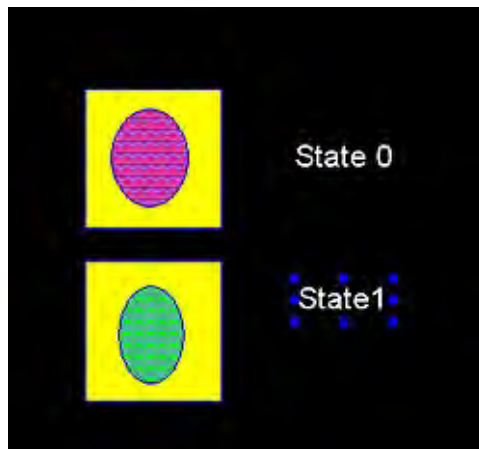
1. With the Shape Library dialog box displayed, click **New Lib...** The New Library dialog box appears.
2. Enter a name for the new library. Click **OK**.
3. The Shape Library dialog box appears with the new library loaded. Click **Close**.
4. The new library is now available to this project but it is not stored into the **Library** subdirectory of EasyBuilder until the project file has been saved.

## Creating, deleting and placing shapes

EasyBuilder provides drawing tools that can be used to create shapes. For more information on how to use the drawing tools, consult the first part of this chapter. The shapes that you create can be permanently stored into the shape libraries for access by any project.

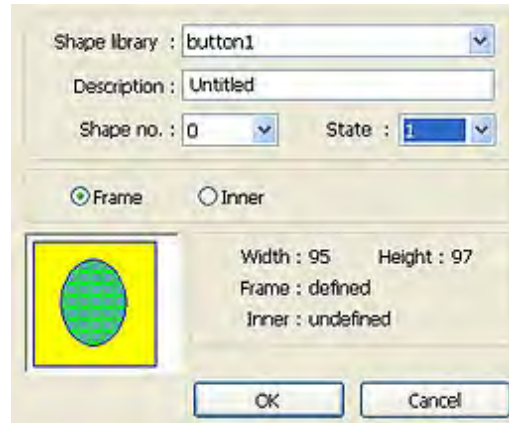
### ► To add a new shape to a library

1. This example shows how to add a two-state shape to a library. You must first create the shapes that you plan on storing using the drawing tools supplied with EasyBuilder. The shapes can be any size. You can use any combination of drawing tools to create the shape.

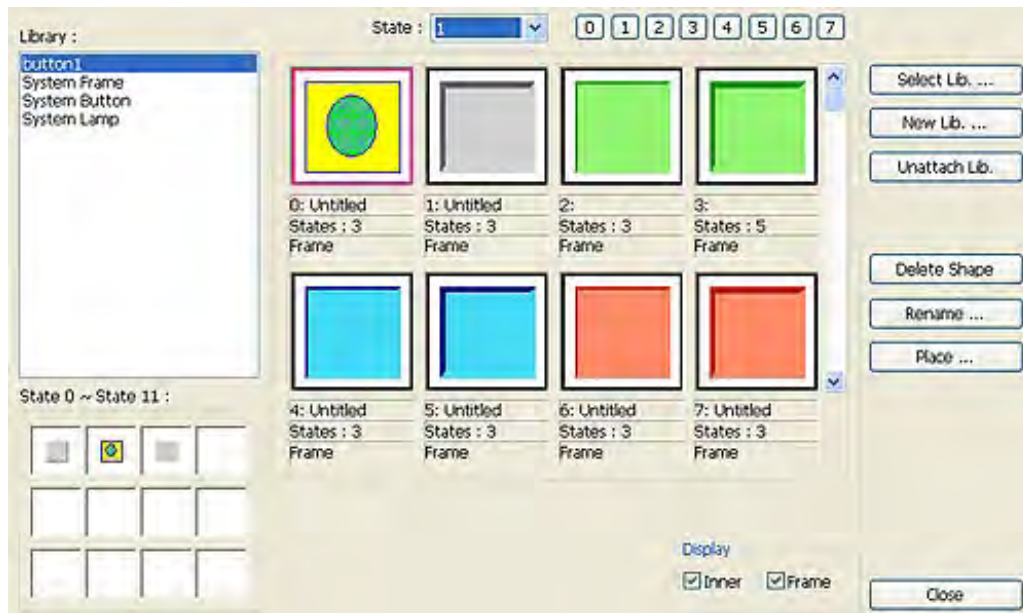


2. Highlight the shape by clicking the mouse and dragging a rectangular outline so that it covers the entire shape. When you are done, you should see small black squares around the perimeter of the shape.

- From the Library menu click **Shape**, then **Save to library**. Or click the **Save Objects to Shape Library** icon in the Standard toolbar. The Save to Shape Library dialog box appears.



- Click the **Shape library:** box and select the library you wish to store the shape in
- Click the **Shape No.:** pull-down box to select the location (0) to store the shape. Note that the **State No.:** will automatically change to 0, first unused state.
- Enter a title for the new shape in the **Description:** box.
- Select the **Frame** option button to have the graphic represent the border of the new shape; or the **Inner** option to have the graphic represent the interior
- Click **OK**. The Shape Library dialog box appears with the shape stored in the library.



- Click **Close** to return to the main screen of EasyBuilder.
- From the **File** menu, click **Save** to save the shape to the library. Important- you must save before attempting to store the next shape.
- Highlight the other shape by clicking the mouse and dragging a rectangular outline so that it covers the entire shape.
- Click the **Save Objects to Shape Library** icon in the Standard toolbar. The Save to Library dialog box appears.
- Click the **Shape library:** box and select the library you wish to store the shape in.
- Click the **Shape No.:** pull-down box to select the location (0) to store the shape. Note that the **State No.:** will automatically change to state 1, the next available state.

15. Click **OK**. The Shape Library dialog box appears with the shape stored in the library under **State 1**.
16. Click **Close** to return to the main screen of EasyBuilder.
17. From the File menu, click **Save** to save the shape to the library.



*When adding new shapes to existing libraries that come with the EZware-5000 software, remember that any future upgrade that you receive from Maple Systems may overwrite existing libraries during installation. This will delete any new shapes that you have entered. Therefore, we recommend that you create new libraries to store these shapes or make backup files of the libraries before installing new upgrades.*

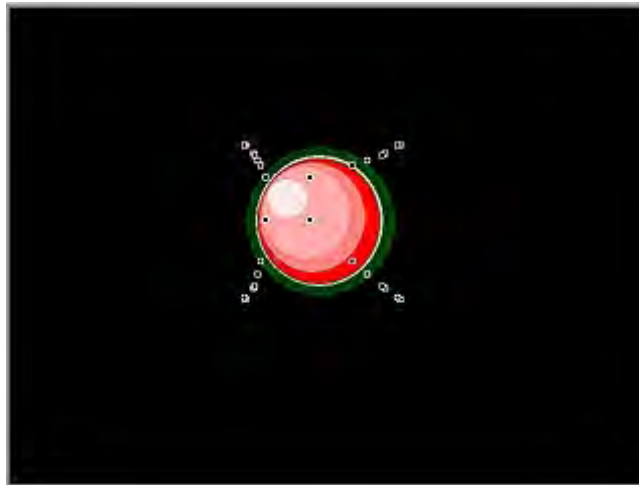
► **To delete a shape from a library**

1. From the Library menu, click **Shape**, then **Call up library**.
2. Select the library that the shape is located in.
3. Click on the shape that you wish to delete from the library.
4. Click **Delete**. The shape is removed from the library.
5. Click **Close** to go back to the main screen of EasyBuilder.
6. Save the project file to permanently save changes made to the library. Caution – once project is saved, the shape is permanently deleted from the library and cannot be replaced.

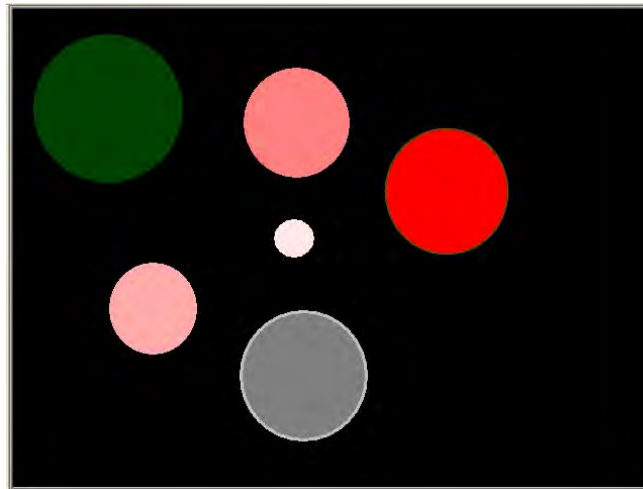
Finally, you may want to use all or parts of an existing shape to create a new one. To do this, you must first ‘place’ the shape onto the work area of EasyBuilder for editing.

► **To place a shape on the work area**

1. From the Library menu, click **Shape**, then **Call up library**.
2. Select the library in which the shape is located. For this example, select **button1**.
3. Click on the shape that you want to place on the work area. For this example, select button number 12.
4. Click **Place...** The main screen of EasyBuilder reappears with the shape in the upper-left corner of the work area.
5. For this example, move the shape to the middle of the work area, as shown.



- Highlight the shape and separate each separate component.



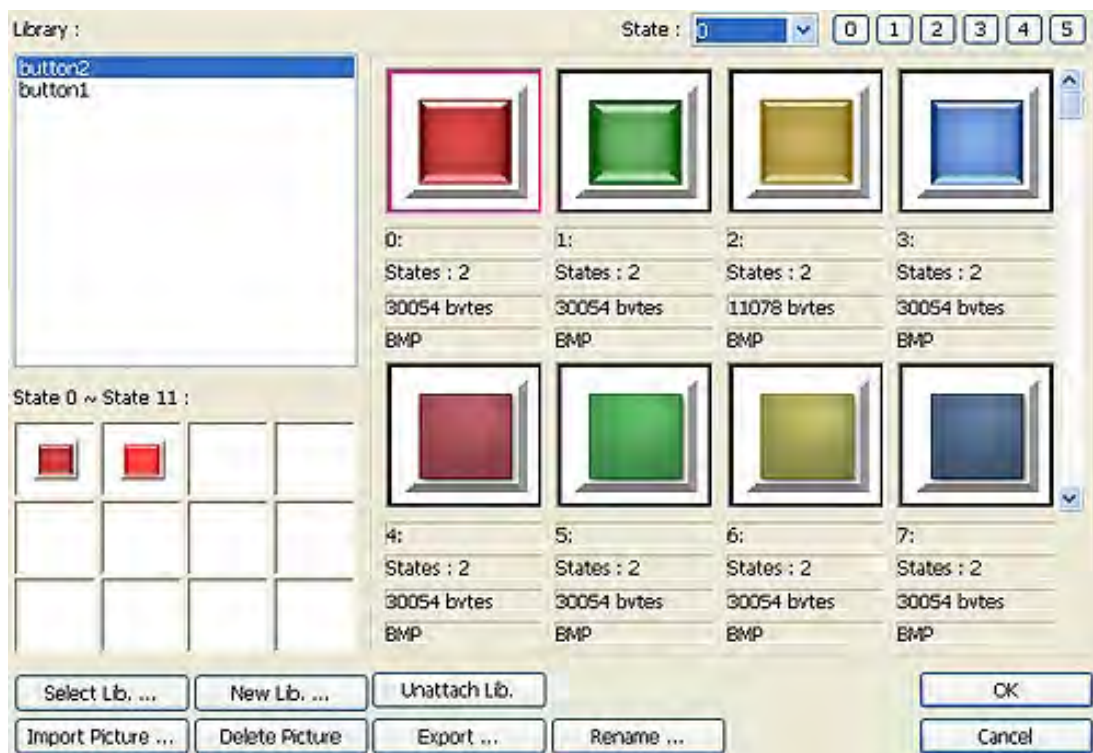
- As you can see, this particular shape is composed of six objects. You can edit these objects to change the shape or create a new shape from these objects. Please note that you are not actually changing the shape stored in the library. To change the shape, you must delete the shape and store any changes you make to a new shape.

### Using Picture Libraries

EasyBuilder includes picture libraries from which you can select pictures.

- To display the Picture Library dialog box 

- From the Library menu, click **Picture**, then click **Call up library**. Or click the **Call up Picture Library** icon from the Standard toolbar. The Picture Library dialog box appears.



The Picture Library provides several options which are discussed in the following sections.

## Accessing and Creating Picture Libraries

You can open, close, or even create new picture libraries easily using the Picture Library dialog box.

### ► To open other picture libraries

1. With the Picture Library dialog box displayed, click **Select Lib...** The Open dialog box appears.
2. Select which library you would like to open, then click **Open**. The Picture Library dialog box reappears with the selected library displayed.

If the project has ten libraries open, you may want to close a library so that another one may be opened.

### ► To close or 'unattach' an open picture library

1. With the Picture Library dialog box displayed, choose from one of the open libraries of pictures in the library: section.
2. Select which library you would like to close, then click **Unattach Lib**. A dialog box appears asking if you really want to unattach the selected library. Click **Yes**.
3. The Picture Library dialog box reappears and the library is removed from the Library: list.

You have the option of creating a new library. This can be very useful when organizing bitmaps.

### ► To create a new picture library

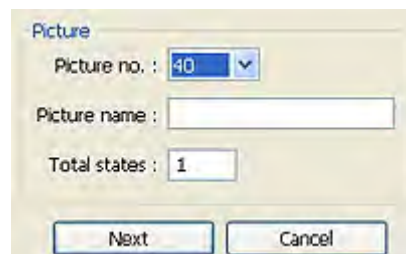
1. With the Picture Library dialog box displayed, click **New Lib...** The New Library dialog box appears.
2. Enter a name for the new library. Click **OK**.
3. The Picture Library dialog box appears with the new library loaded. Click **Close**.
4. The new library is now available to this project but it is not stored into the **Library** subdirectory of EasyBuilder until the project file has been saved.

## Creating and exporting pictures

Although EasyBuilder does not provide the ability to create bitmaps, the software can import bitmaps created from some other application program. This allows the ability to use a wide selection of pictures - including bitmaps, JPGs, GIFs and animated GIFs that can be permanently added to your picture libraries. EasyBuilder accepts pictures that are up to 65K color resolution.

### ► To add a new picture to a library

1. Display the Picture Library dialog box.
2. Select one of the picture libraries from the Library: section.
3. Click on the box location where you would like to enter the new picture.
4. Click **Import picture...** The Import Picture dialog box displays.



5. The **Picture No.:** is the location of the picture in the picture file. This should reflect the box location that you selected in step 4.
6. Enter a title for the new picture in the **Picture name:** box.
7. Enter the total number of states (max. 255) for the bitmap in the **Total states:** box.




- Click **Next**. The Get Picture dialog displays.



- Click **Browse...** to locate the picture to be imported. Once the file is located, it will appear in the square on the right. Click **Finished** to import the picture.
- Click **Next**. The Get Bitmap Graphics dialog box appears..
- Click **OK** again to return to the main screen of EasyBuilder. Place the picture somewhere in the work area.
- From the **File** menu, click **Save** to permanently save the picture to the library.

---

 *When adding new pictures to existing libraries that come with the EZware-5000 software, remember that any future upgrade that you receive from Maple Systems may overwrite existing libraries during installation. This will delete any new bitmaps that you have entered; therefore, we recommend that you create new libraries to store these pictures or make backup files of the libraries before installing new upgrades.*

---

If you find that some of the pictures included in the libraries are not useful, you have the option of deleting them so that they may be replaced by another bitmap.

► **To delete a picture from a library**

- From the Library menu, click **Picture**, then **Call up library**. The Picture Library dialog box appears.
- Select the library that the picture is located in.
- Click on the picture that you wish to delete from the library.
- Click **Delete picture**. The bitmap is removed from the library.
- Click **Close** to go back to the main screen of EasyBuilder.
- Save the project file to permanently save changes made to the library. Caution – once the project is saved, the picture is permanently deleted from the library.

Finally, you may want to export a picture located in one of the picture libraries so that it can be modified or used in other programs.

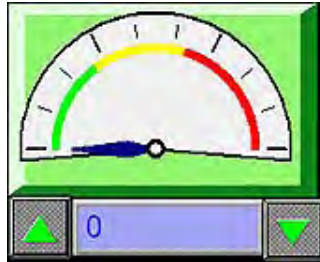
► **To export a picture**

- From the Library menu, click **Picture**, then **Call up library**. The Picture Library dialog box appears.
- Select the library that the picture is located in.
- Click on the picture that you want to export.
- Click **Export...** The Save As dialog box appears.
- In the **File name:** box, enter a file name to which you wish to store the picture.
- Click **Save**. The picture is saved and the Picture Library reappears.

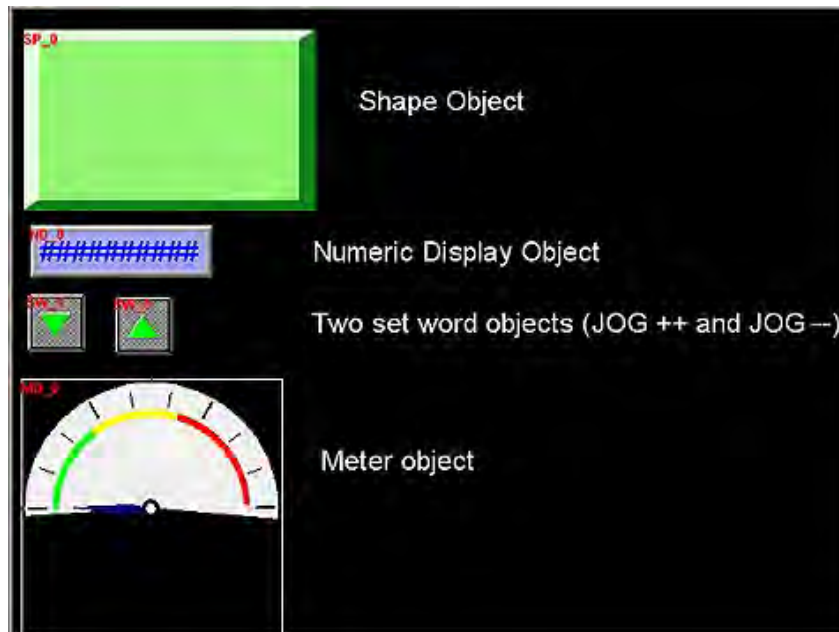
- Click **Close** to exit the Picture Library.

### Using Group Libraries

Any graphics object or objects displayed on the work area of EasyBuilder can be grouped together and stored into a group library. Graphics objects can be passive or active. The groups that you create can be permanently stored into the group libraries for access by any project. Creating group objects often saves a lot of time when creating new projects because you can use the same objects repeatedly. For example, suppose you regularly use a scale meter similar to the following:



This one scale meter is actually composed of five objects.

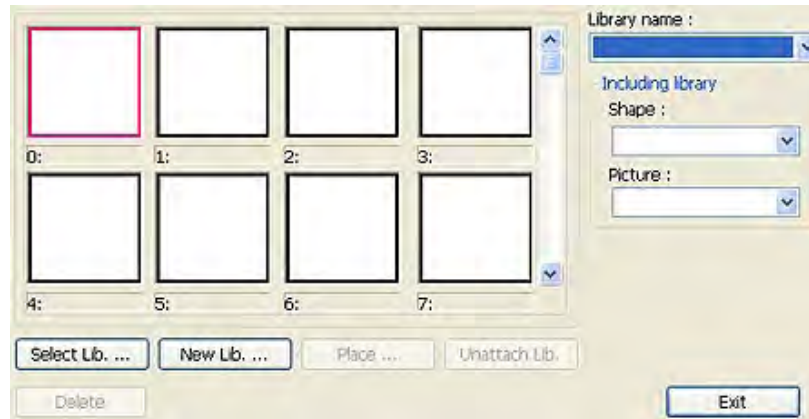


Rather than waste time recreating this scale meter for each new application, it would be much easier to store the meter into a group library.

► **To display the Group Library dialog box**



1. From the Library menu, click **Group**, then click **Call up library**. Or click the **Call up Group Library** icon from the Standard toolbar. The Group Library dialog box appears.



The Group Library provides several options which are discussed in the following sections.

### Accessing and Creating Group Libraries

Group libraries can be opened and closed easily using the Group Library dialog box. It is even possible to create new group libraries.

► **To open other group libraries**

1. With the Group Library dialog box displayed, click **Select Lib...** The Open dialog box appears.
2. Select which library you would like to open, then click **Open**. The Group Library dialog box reappears with the selected library displayed.

If the project has ten libraries open, you may want to close a library so that another one may be opened.

► **To close or 'unattach' an open group library**

1. With the Group Library dialog box displayed, click the pull-down box of **Library Name:** and choose from one of the open libraries of groups.
2. Select which library you would like to close, then click **Unattach Lib.** A dialog box appears asking if you really want to unattach the selected library. Click **Yes**.
3. The Group Library dialog box reappears and the library is removed from the **Group library:** list.

The option of creating a new library is also available. This can be very useful when organizing groups.

► **To create a new group library**

1. With the Group Library dialog box displayed, click **New Lib...** The New Library dialog box appears.
2. Enter a *name* for the new library. Click **OK**.
3. The Group Library dialog box appears with the new library loaded. Click **Close**.
4. The new library is now available to this project but it is not stored into the **Library** subdirectory of EasyBuilder until the project file has been saved.

### Creating, deleting and placing groups

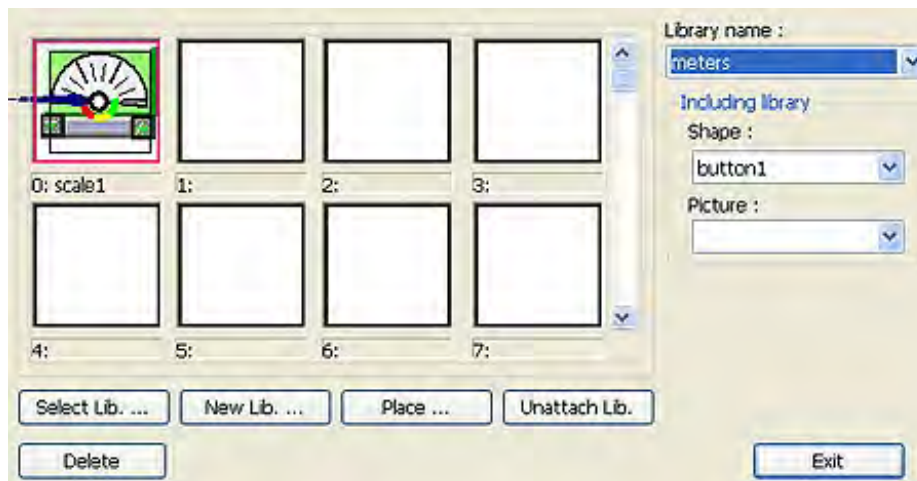
Once a complex group object has been created, it must be stored permanently into a group library where it can be retrieved and used in multiple projects. At any time, it is possible to place a group object onto the work area of EasyBuilder for usage or editing.

► To add a new group to a library 

1. Highlight the group object by clicking the mouse and dragging a rectangular outline so that it covers the entire group. When you are done, you should see small black squares around the perimeter of the group. For this example, the scale meter shown at the beginning of this section is highlighted.
2. From the **Library** menu click **Group**, then **Save to library**. Or click the **Save to Group Library** icon in the Standard toolbar. The Save to Library dialog box appears.



3. Click the **Group library:** box and select the library you wish to store the group in, (for this example **meters**).
4. Click the **Group No.:** pull-down box to select the location to store the group. Note that the **Group No.:** will automatically change to **0**, (first group not used).
5. Enter a title for the new group in the **Description:** box.
6. Click **OK**. The Group Library dialog box appears with the group stored in the library.



7. Notice that any shape or bitmap libraries used to create the group object are listed for your reference in the **Including library** box.
8. Click **Close** to return to the main screen of EasyBuilder.
9. From the File menu, click **Save** to permanently save the group to the library.



*When adding new group objects to existing group libraries that come with the EZware-5000 software, remember that any future upgrade that you receive from Maple Systems may overwrite existing libraries during installation. This will delete any new group objects that you have entered. Therefore, we recommend that you create new libraries to store these group objects or make backup files of the libraries before installing new upgrades.*

If you find that some of the groups included in the libraries are not useful, you have the option of deleting them so that they may be replaced by another group.

► **To delete a group from a library**

1. From the Library menu, click **Group**, then **Call up library**.
2. Select the library that the group is located in.
3. Click on the group that you wish to delete from the library.
4. Click **Delete**. The group is removed from the library.
5. Click **Close** to go back to the main screen of EasyBuilder.
6. Save the project file to permanently save changes made to the library. Caution – once a project is saved, the group is permanently deleted from the library and cannot be replaced.

Finally, to place a group object onto the work area for use or editing, do the following.

► **To place a group on the work area**

1. From the Library menu, click **Group**, then **Call up library**.
2. Select the library in which the group is located. For this example, select **keypad**.
3. Click on the group that you want to place on the work area. For this example, select item **0**.
4. Click **Place...** The main screen of EasyBuilder reappears with the group in the upper-left corner of the work area.
5. Move the group object to the preferred location on the window.

This completes our discussion on creating shapes, bitmaps, and group objects with EasyBuilder. This chapter also provided some examples on how to use the drawing tools. The next several chapters describe the many parts or active graphics objects that can be placed onto windows.

There is one other type of library that we will discuss here. Although it is not a graphic, sound libraries work in much the same way that all of the other libraries do, and therefore fit into our discussion of libraries.

## ***Using Sound Libraries***

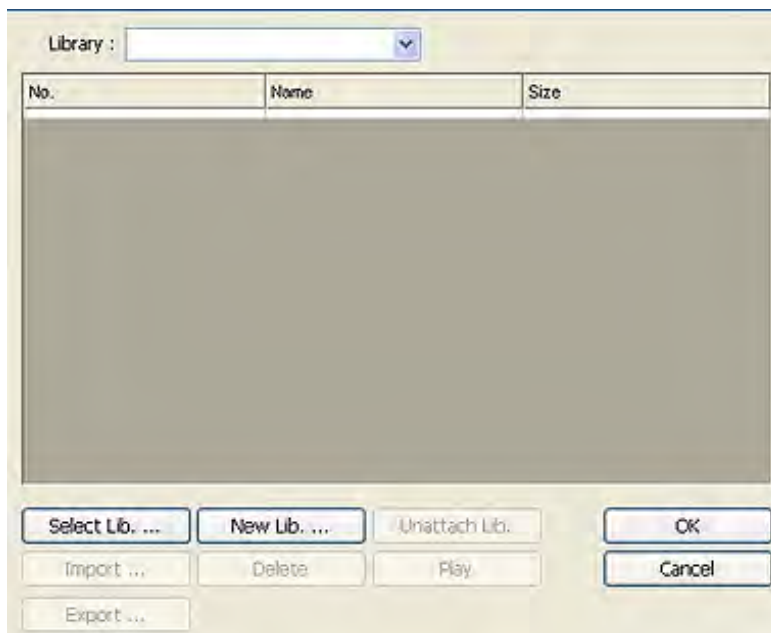
The Silver Series can import and play WAV files to signal alarm conditions or to be linked to a PLC control object. These sounds can be found and stored in sound libraries. Objects that can have sounds linked can be configured in the individual object.

## **Accessing and Creating Sound Libraries**

Sound libraries can be easily accessed and opened using the Sound Library dialog.

► **To open access the sound libraries:**

1. Select **Library - Sound**. The Sound Library dialog appears.



2. Click **Select Lib. ...** and select a library. Click okay. Now the library dialog will show with the list populated.

► **To preview the sounds in the library:**

1. Select **Library-Sound**. The Sound library dialog appears.
2. Highlight the *sound* that you would like to preview.
3. Click **Play**.

► **To create a sound library:**

1. Select **Library-Sound**. The Sound Library dialog appears.
2. Click **New Lib...** the New library dialog appears.
3. Type a *name* for your library.
4. Click **OK**.

## Importing and Exporting WAV files

► **To import a WAV file for use in sound libraries**

1. Select **Library-Sound** to display the Sound Library dialog.
2. Click **Import**. The Open file dialog appears.
3. Browse to the sound that you want to import and click **Open**.

► **To export a WAV file for use in other applications**

1. Select **Library-Sound** to display the Sound Library dialog.
2. Highlight the WAV to export from the library table and click **Export...** The Save in.. dialog appears.
3. Browse to the *file location* where you will save the sound and type a *name* for the sound. Click **Save**.

## Deleting and Unattaching Sounds and Sound Libraries

► **To delete a sound from the sound library:**

1. Select **Library-Sound** to display the Sound Library dialog.
2. Highlight the sound that you would like to delete.
3. Click **Delete...**

4. Select **File...Save**. You must save the project in order for the delete to be permanent.

► **To unattach a sound library:**

1. Select **Library-Sound**. The Sound Library dialog appears.
2. Click **Unattach Lib**.
3. A dialog pops up asking if you really want to unattach the library. Click **Yes**.

**Please Use This Page For Your Notes:**



## Chapter 9 - Creating and Using Databases and Languages

EasyBuilder includes the ability to create and use both a Tag Database Library and a Label Database Library. The Tag Library is a database of PLC register addresses. Once created, the Tag Database allows individual object address assigning through Tag selection from the Tag Library. The Label Library is a database of text labels. Once created, the Label Database allows individual object text labeling through Label selection from the Label Library. The Label text may also be represented in up to eight languages, each selectable for display.

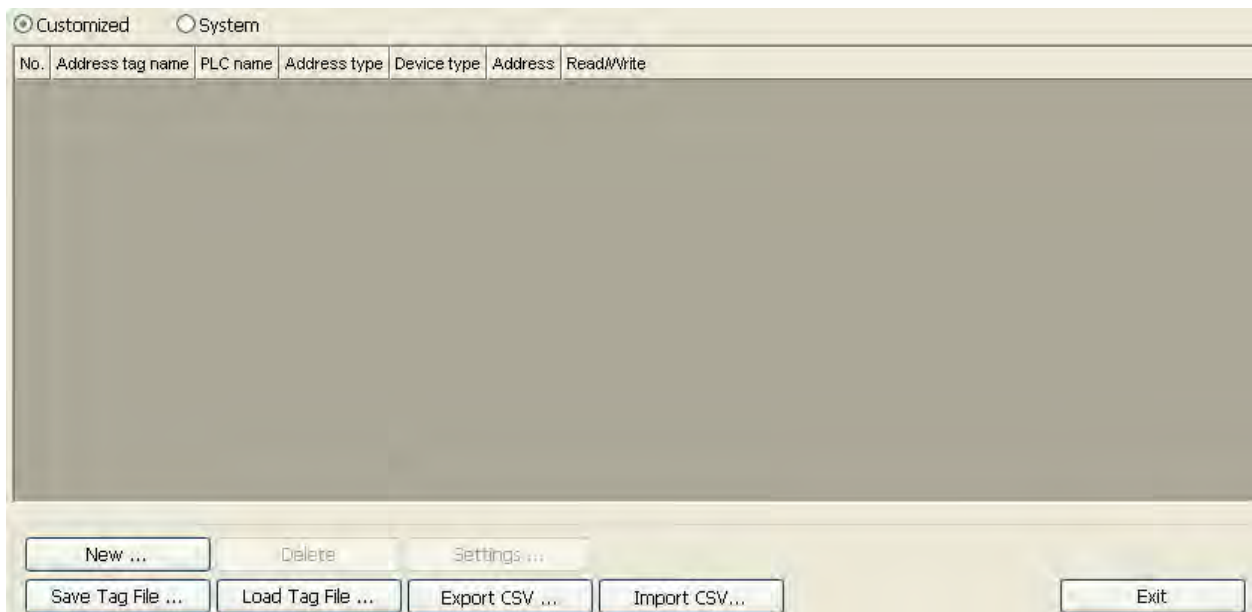
### Creating and Using the Tag Library

The Tag Library is a database of register addresses, with each tag representing a single address. EasyBuilder gives you the option of using either Customized or System tags. Customized tags are tags that you build yourself. System tags are predefined tags that allow access to certain built-in functions and settings. You can select whether you will be using customized or system tags in the Address Tag Library dialog. The Tag listing does not contain display information, as display attributes are set when the individual object using the Tag listing is created.

#### ► To create a Tag Database Library



1. From the Library menu, click **Tag...**, or click the Address Tag Library Manager button from the Standard toolbar. The Address Tag Library dialog appears.



- To create a new Tag listing, click the **New...** button to display the Tag dialog.

- Enter the new *Tag name*. Tag names may be up to 100 characters, any character.
- Select the *PLC name*: as configured in the Device Table from the drop down menu.
- Select the *Address type* (Bit or Word) from the drop down menu..
- Select the *Device type* from the drop down menu.
- Enter the *Register Address*.
- Click **OK**.
- Make additional entries to the database by repeating the procedure.

► **To delete a tag:**

- Open the Tag Library as directed above.
- Select the tag to delete.
- Click the **Delete** button.

► **To modify a tag:**

- Open the Tag Library as directed above.
- Select the tag to edit.
- Click the **Settings..** button.
- Edit as applicable and then click **OK**.

## Importing and Exporting the Tag Library

This feature allows tags from the tag library to be saved into a .tgl file format. Once saved, the file can be loaded into another project.

► **To save the tag library:**

- Click on **Save Tag File...** The Open dialog box appears.
- Enter the *name* of the file where you wish to save the data.
- Click **Open**.

► **To load the Tag Library from an existing .tgl file:**

- Click on the **Load Tag File...**The Open dialog box appears.
- Browse for the .tgl file that contains the tags.
- Click **Open**.

---

 *The .tgl file format can not be edited.*

---

The tag library can also be imported and exported using a comma-separated variable (.csv) file format. This format is very useful if you need to make several changes to the database; or, if creating a new project, it is easier and faster to create and edit the tags in a .csv file.

► **To export the tag library to a .csv file:**

- Click **Export CSV**. The Open dialog box appears.
- Enter the *name of the file* where you wish to save the data.
- Click **Open**.

► **To import the tag library from a .csv file:**

1. Click **Import CSV...** The open dialog box appears.
2. Brows for the CSV file containing the tags.
3. Click **Open**.

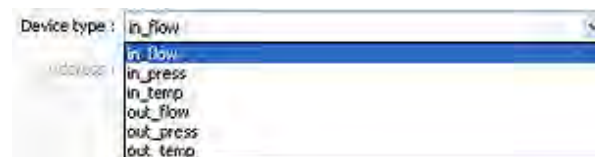
## Using the Tag Library

Once the Tag Database has been created, individual objects can use the Library to assign a Tag to the object, referenced to the PLC register address.

1. On addressable objects, once the Tag Database has been created, a User-defined tag checkbox is available. This box is not available if no tags have been configured in the Database Library.



2. Check **User-defined tag** checkbox. The **Device Type** list will be filled with a list of available tags. Objects addressable as Register-type objects will display Tags configured as Word; Objects addressable as Bit-type objects will display Tags configured as Bit.



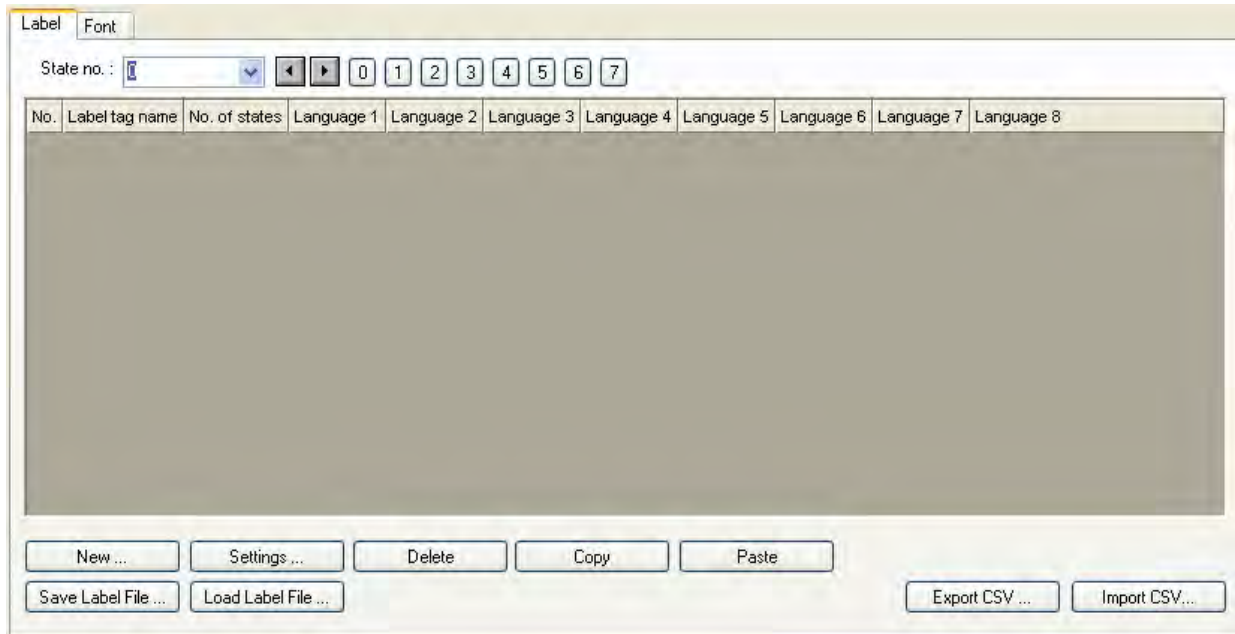
3. Select the desired **Tag** from the Device Type list to address the Object to the referenced Register Address.

## Creating the Label Library

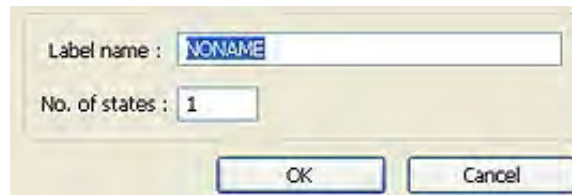
The Label library is a database of text Labels, for use with objects utilizing text labels descriptive of their status or condition. Each label can consist of up to 256 unique text strings (representing the maximum number of states available per object), displayable in up to eight separate languages.

## To create a label text database library

1. From the Library menu, click **Label...**, or click the call-up label library manager icon from the standard toolbar. The Label Library dialog box appears.



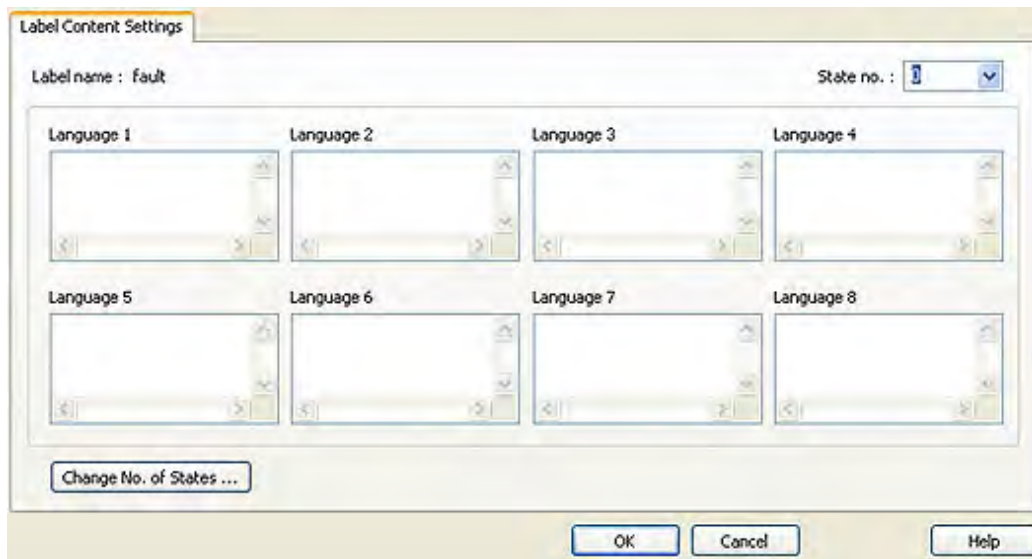
2. Create a label by opening the Label Library as detailed above and Clicking the **New...** button to display the Label dialog.



3. Enter the *Title (Name)* for the label, and how many states the new label requires. For example, a label Titled “Start” requiring 2 states, to represent separate conditions, where condition one status is “START”, condition two is “STOP”.
4. Click the **OK** button, and the new label is added to the database. The Name will appear in a selectable listing of available Labels for use with objects that support text field labeling.


No.	Name
1	Fault
2	lang
3	on
4	run
5	start

- The new label is still selected. Click the **Settings...** button to display the *Label Content Setting* dialog. In the *State No.* list pull-down box, select the state that the text is representing.



- In each **Language** box, as needed, enter the text representing that state in each language.
- If further text labeling is required to represent conditions or status in additional states, while still in the *Settings...* mode for Label Content Setting, select the state requiring text labeling. Enter the text in the Language boxes as above.

---

 *There is no inter-language conversion or translation capability. Entries must be made using the correct wording for the language designated, to display as entered.*

---

- Continue to add state descriptive text labeling as required.
- When all text has been entered for all states desired, Click **OK**. The Label is now contained in the Label Library and displayed when the Library is opened. View text labels for the various states of each label in the Library by selecting the State to display in the pull down box.
- Once a label text has been created and you wish to change the number of states, you may do so by double clicking on the desired label text. On the Label Content Setting dialog box, click the **Change No of States...**button, and then enter the *No. of states* desired for the label.

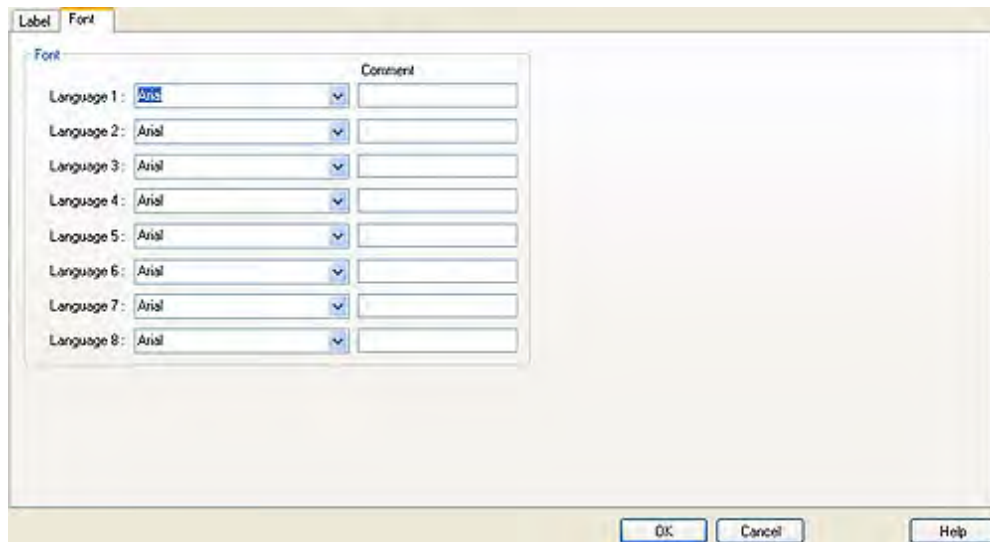
## Setting Different Fonts for Different Languages

If you would like to use different fonts for different languages, for instances, if the language you are using requires special characters or unicode fonts, you can select a different font for each language using the label library.

### ► To set a different font for each language

- Select **Library-Label** or select the **Label Library Manager** icon from the standard toolbar.

- The Label Library dialog appears. Select the **Font** tab.



- Select the *font* for each language from the drop down menu. Note that the fonts must be available to you in your copy of Windows.
- Click **OK**.

### Importing and Exporting the Label Library

This feature allows you to save the labels from the label library into a .lbl file format. Once saved, the file can be loaded into another project.

#### ► To save the label library:

- Click on **Save Label File...** The open dialog box appears.
- Enter the *name of the file* where you wish to save the data.
- Click **Open**.

#### ► To load the label library from an existing .lbl file:

- Click **Load Label File...** The Open dialog box appears.
- Browse for the lbl file that contains the labels.
- Click **Open**.

---

 *The .lbl file format can not be edited.*

---

The label library can also be imported and exported using a comma-separated variable (.csv) file format. This format is very useful if you need to make several changes to the database. When creating a new project, it is easier and faster to create and edit tags as .csv files.

#### ► To export the label library:

- Click **Export CSV File...** The Open dialog box appears.
- Enter the *name of the file* where you wish to save the data.
- Click **Open**.

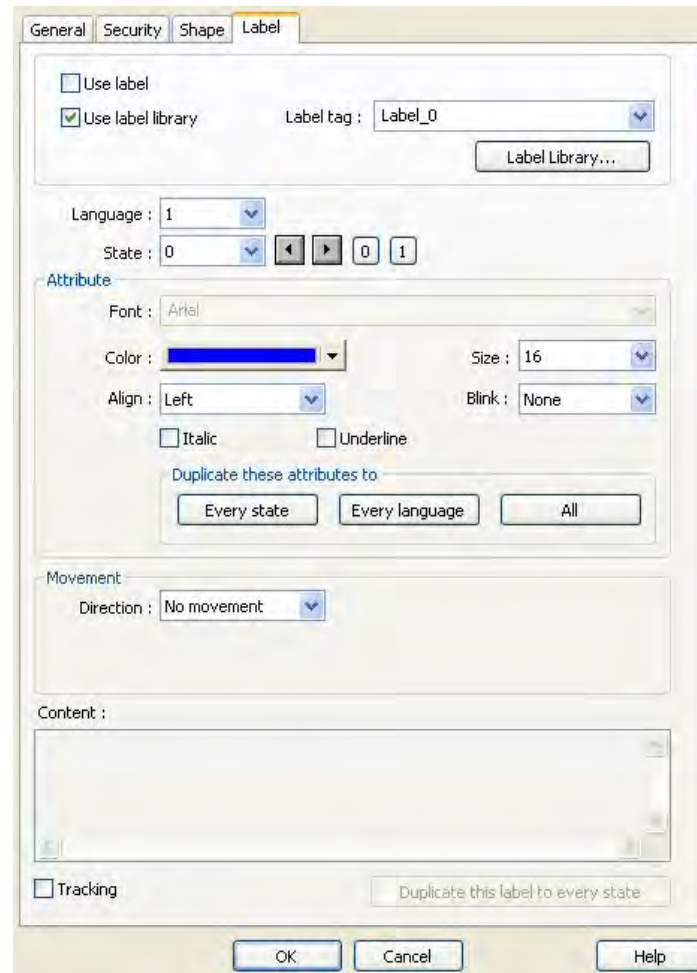
#### ► To import the label library:

- Click **Import CSV File...** The Open dialog box appears.
- Browse for the CSV file that contains the labels.
- Click **Open**.

## Using the Label Library

Once entries have been made to the Label Library, they can be used with any object that supports text field labeling.

1. Select an object that supports text field labeling as required. On the **Label** tab for the object's **Attributes**, check the **Use Label Library** checkbox. The Label Index list will be enabled, containing all of the labels in the database.



2. Select the applicable *label* from the label tag drop down menu. The label will display on the object, representative for states and languages as entered in the **Label Library** dialog.
3. Once created with Labels assigned, objects can be viewed displaying the various text labeling in each of the states detailed, in each of the languages utilized. To view, on EasyBuilder's Standard Toolbar, select the state and language for objects displayed on the window. The object's text labeling display will change according to the state and language selected for viewing.
4. To use the Multi-Language features of the Label Library to display object labels in languages other than the default language (Language 0), the function must be enabled. Enable the Multi-Language capability as detailed below.



### Using Languages with the Label Library

The Silver Series supports up to eight languages for use with the Label Library text labeling. In order to use the Multi-Language features, a label configured in multiple languages must be created and stored in the Label Library, as detailed above. The language displayed is selected through Internal Local Word Control, LW9134.

1. Text Labels in any of up to eight languages can be displayed, as configured in the Label Library. In LW9134, a Value of 0 enables Language 1 as configured in the Label Library, a Value of 1 enables Language 2, a Value of 2 enables Language 3, and a Value of 3 enables Language 4, etc.

- The appropriate value must be written to LW9134 to enable a designated language to display. For example, a series of set word objects writing value to LW9134 could be placed on a setup screen. Each set word might be labeled with the Language to be enabled, and the appropriate value addressed to LW9134. If Set Word #0 enables English and English is the configured Language 0 in the Label Library, pressing the English Set Word object writing a 0 to LW9134 will enable label text display in English. If Set Word #1 enables French and French is the configured Language 1 in the Label Library, pressing the French Set Word object writing a 1 to LW9134 will enable text display in French. Continue for all Languages configured in the Label Library.



*LW9134 is the System Tag Language Mode.*



## Chapter 10 - Representing Data with Graphics Objects

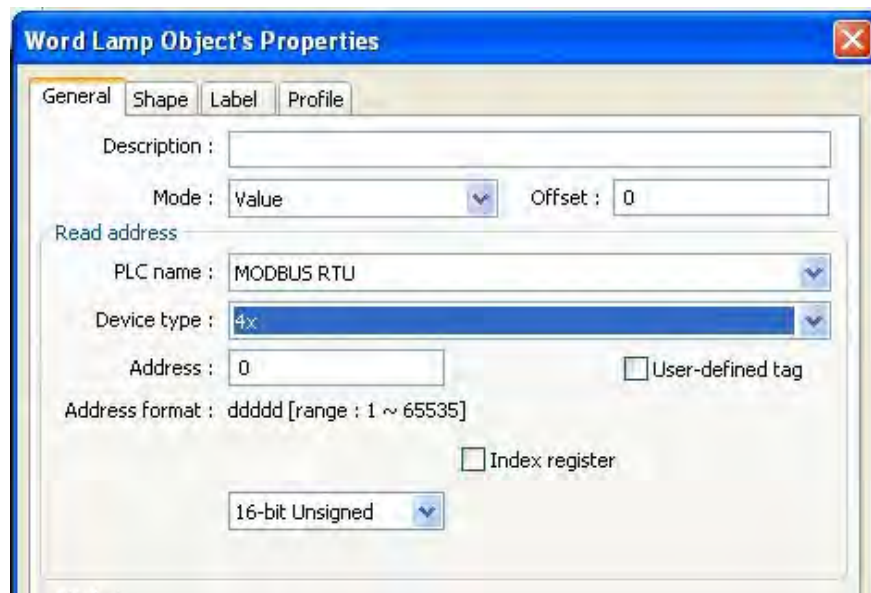
EasyBuilder includes several active graphics objects or 'parts' which are used to represent data that is stored in the PLC or internal memory of the OIT. The data represented can be single bit coils, 16-bit, 32-bit, or 64-bit registers. The data can be represented as numbers, ASCII characters, or as graphic shapes or bitmaps. This chapter focuses on only parts that perform relatively simple functions. More complex parts used for alarms, trending, etc. will be reserved for later chapters.

### General Properties of Objects

Many of the objects in EZware-5000 share several common elements that are described in more detail in this section.

When you create a new object, a popup dialog box always appears that is used to assign the particular attributes to that object.

For example, below is the attribute box that appears when a new Word Lamp object is created:



*Most of the objects have four tabs: General, Shape, Label, and Profile.*

### Under the General Tab

#### Description:

This is a text box that allows the programmer to add more details about this particular object. Please note that information put into this box is not downloaded into the OIT.

**Read Address or Write Address frame:**

This is used to assign a PLC or internal HMI memory address to the object. Included in this category are the following bulleted items:

**PLC name:** select the target communications driver (port) Note: In EasyBuilder-5000, go to Edit...System Parameters...Device tab to assign a communications driver to a physical serial or Ethernet port on the OIT.

**Device type:** select the target PLC memory area. Note: For more information on options and ranges, consult the appropriate Controller Information Sheet documentation that is included with the EZware-5000 software and available for download from the Maple Systems [website](#).

**Address:** enter the specific PLC memory address.

**System tag:** this appears if you have selected Local HMI for PLC name. System tags are internal OIT memory addresses that are reserved for specific function. Please read the next section in this chapter for more information.

**User-defined tag:** this appears if you have created any tags in the Tag Library and you have selected an alternate communications driver (not Local HMI) for PLC name. After you have checked User-defined tags, the tags appear in the Device Type drop-down box. Note: for more information on the Tag Library feature, see Chapter 9: Creating and Using Databases and Languages.

**Index Register:** this optional feature creates a pointer value using internal OIT Index Registers. This is useful if you wish to dynamically (during OIT runtime) change the target PLC address without modifying the graphics object and downloading a new project into the OIT. Note: for more information on the Indexing feature, see Chapter 10: Using Index Registers.

**Data format:** this field determines how the OIT will interpret the data it reads from the PLC register. Options are 16-bit BCD, 32-bit BCD, 16-bit Unsigned, 16-bit Signed, 32-bit Unsigned, 32-bit Signed, and 32-bit Float.

**Macro frame:**

Available on the Set Bit, Toggle Switch or PLC Control object, this feature activates a pre-defined Macro. For more information on macros, see Chapter 15 – Macros.

**Execute macro:** check to enable activation of a macro when object is pressed.

**Macro:** Select the predefined macro from the drop-down list box. Note: to create a new macro, click Tools...Macro in the standard toolbar of EasyBuilder-5000.

**Notification frame:**

Available on the Set Word, Slider, Numeric Input, ASCII Input, and Function Key objects, this feature can be used to 'notify' the PLC or other device (such as another OIT) that the object has been pressed by the operator. This can be used to simultaneously start other functions or routines in the PLC or to simply 'warn' the PLC or OIT that an action is about to occur.

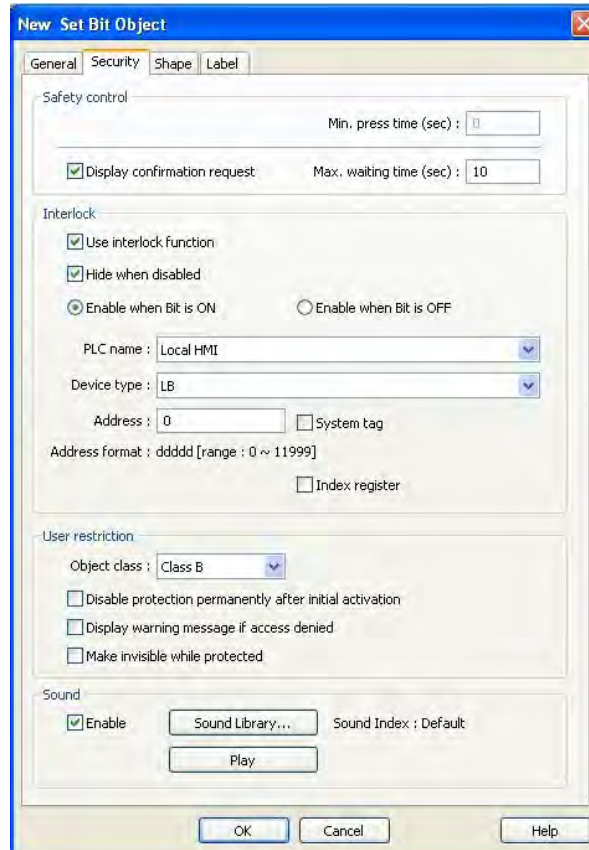
**Enable** – check to enable this feature

- Set ON or Set OFF - select which condition is performed by the object on the Notification bit.
- Before writing or After writing – select when notification occurs before predefined action of the object is processed or after.

## The Security tab

Many objects that allow the operator to write values to the PLC (i.e. Set Bit, Set Word, Toggle Switch, Option List Switch, Slider, Multi-State Switch, Numeric Input, ASCII Input, Function Key, Data Transfer (Trigger-Based), and Backup) have additional security features that can be used to ensure that the operator is allowed to write to the PLC only when specific conditions have been met or a password entered.

These features are enabled on the Security tab:



### Safety Control frame:

Use this feature to ensure that the object was not accidentally pressed by the user.

**Min. press time (sec):** Range is 0 to 100 seconds. If used, the operator must press and hold down the target object (i.e. button) for the specified time period before the action is processed. For example, if the object is a Set Bit configured to set (turn ON) a PLC coil when pressed and Minimum press time is set to 3 seconds, the operator must press and hold down the Set Bit button for three seconds before the OIT sends a command to set the PLC bit.

**Display confirmation request:** if enabled, this feature will display a popup dialog box on the OIT screen:



When the dialog box pops up, the operator presses the 'Yes' button to process the predefined action.

**Interlock frame:**

This feature allows an ‘interlock’ to be used. An interlock is typically a bit in the PLC that must be set or ‘activated’ before the object can perform its predefined action.

**Use interlock function checkbox:** check to enable this feature

**Hide when disabled checkbox:** use this option if you want the object to be invisible when the interlock is disabled.

**Enable when Bit is ON, Enable when Bit is OFF:** select which condition of the interlock bit activates the object.

**User restriction frame:**

This feature is used in conjunction with the Security tab under Edit...System Parameters to require that a password be entered by the operator before the predefined action of the object is processed. For more information on using the Password Security feature, see Chapter 6: Using EZware-5000 General Settings...Security section.

**Object class:** there are six ‘classes’ or security levels- Class A thru Class F. Select the class level that the HMI must be set to before allowing the object to be processed.

**Disable protection permanently after initial activation:** if enabled, the OIT will no longer check the current security level after the initial activation.

**Display warning message if access denied:** displays a warning message, alerting the operator that object is password-restricted. If not enabled, the action is simply not processed.

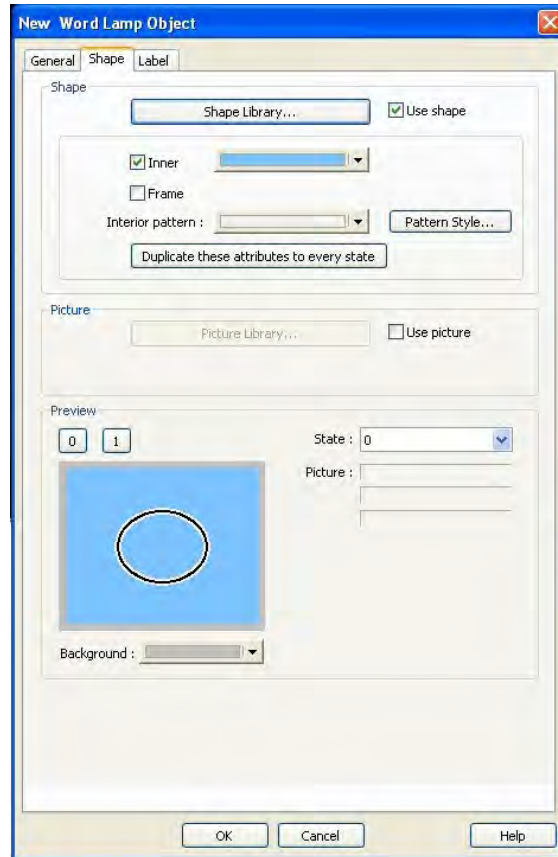
**Make invisible while protected:** if enabled, the object remains invisible until the proper security level is activated.

**Sound frame:**

This feature is available only for OIT models that support an audio out jack. The specified sound is played through the amplified speaker attached to the OIT whenever the object is pressed.

### Shape tab:

This is used to assign a graphic picture(s) to the object that represents the data from the PLC. The picture can be a simple button key, lamp, or other object. The pictures are stored in graphics libraries of which there are two types: Shape libraries are vector-based graphics pictures and Picture libraries are pixel-based (BMP, JPG, or GIF) graphics. EZware-5000 includes many pre-defined libraries or you can create additional libraries of objects (see Chapter 8: Creating Graphics Objects for more information). Below is an example of the shape tab for a Word Lamp object:



### Shape frame:

The Shape frame is used to assign a Shape Library graphic to the object. Click Use shape to select. The following features are found in the shape frame:

**Shape Library... button** – Click on this button to select a particular shape from one of the Shape Libraries.

**Inner checkbox:** this appears when using the System Frame, System Button, or System Lamp libraries. When checked, it allows you to change the inner color of the shape selected. Use the adjacent drop down box to select the color.

**Frame checkbox:** same as the Inner checkbox, except that this controls the color of the outside frame of the shape selected.

**Interior pattern:** As with the Inner and Frame checkboxes, the Interior pattern is only available when the Shape Library selected is System Frame, System Button, or System Lamp. Click the Pattern Style button to select a pattern for the inner color of the shape selected. Use the color drop-down box to select the secondary color used for the pattern.

**Duplicate these attributes to every state:** Click this button to copy the colors selected to every state of the selected System shape.

**Picture frame:**

Used to assign a Picture Library graphic to the object. Click Use picture to select

**Picture Library... button** – Click on this button to select a particular picture from one of the Picture Libraries.

**Preview frame:**

Used to quickly check how the selected graphic looks for each state of the object.

0 thru 4 buttons: these are ‘quick buttons’ that, when pressed, show the graphic that is configured to represent that particular state of the object.

State: If more than five states are available for this object, use this drop-down box to view the graphic that represents each state.

Picture: displays the name of the particular picture selected from one of the Picture Libraries.

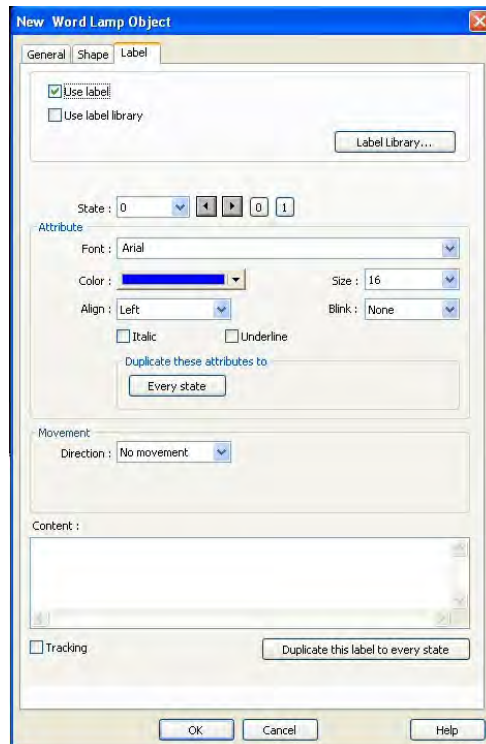
Type: displays type of picture (BMP, GIF, or JPG) used

Size: displays size of picture in pixels

Background: if the background color matches the color of the picture, you can use this drop-down box to change the background color, making it easier to view the graphic used.

**Label tab:**

The label tab can be used to assign a text label to each state of the selected object. For example, a push on/off button might have ‘OFF’ for the label assigned to State 0 and ‘ON’ assigned to State 1. You can assign labels directly to the object or you can create a library of labels for multiple objects using the Label Library feature (for more information on Label Libraries, see Chapter 9: Creating and Using Databases and Languages). Below is an example of the label tab for a Word Lamp object:



**Top frame:**

Use label: click to enable the label feature

Use label library: clicking this checkbox allows you to select a label tag from the label library. Note: this feature is not available until you first create label tags in the Label Library.

- Label tag: Select from the drop-down box which label tag you wish to use.

- Label Library... button: click to go to the Label Library to add/edit tags.

- Language: this appears if you are using the Label Library. Allows you to see the text for that particular language and change the font size but you cannot edit this from this tab- you must go to the Label Library. Note: for more information on using Languages, see Chapter 9: Creating and Using Databases and Languages

State: use this drop-down box or the buttons adjacent to see and adjust the parameters for the text for that particular state.

**Attribute frame:**

**Font:** select type of font

**Color:** select color for text

**Size:** select font size

**Align:** this is how the text will align in relation to the size of the object. Options are left, center, or right text alignment.

**Blink:** blinks the text. Options are none, 1 second, and 0.5 seconds.

**Italic:** to show text in *italics*. Note: Although selectable, this option is not available for every font.

**Underline:** to show text in underline. Note: Although selectable, this option is not available for every font.

**Duplicate** these attributes to

- Every state button: When clicked, EZware-5000 will automatically copy the attributes to every state.
- Every language button: copies attributes to every language. Note: this button is only available when a label tag is used.
- All: copies for all states and all languages. Note: only available when label tag is used

**Movement frame:**

**Direction:** a special effect that allows you to create a label that will scroll across the object in the direction selected: Left, Right, Up, or Down.

- Speed: determines the relative scrolling speed (slowest setting on the left)
- Continuous: if checked, the text will repeat itself continuously

**Content frame:**

**Text box:** this is the area in which the text is entered. If the Label Library feature is used, the text for the label used is displayed here but cannot be edited (go into the Label Library to edit the text).

**Tracking:** After you have placed the object on screen, you can see how the text is positioned on the object, as in this example:



If you click twice on the object, you can highlight the text label and position it anywhere on the object that you see fit:



The Tracking feature, if checked, will automatically move the text labels assigned to the other states to the same position as the current state text label you have selected:



Duplicate this label to every state button: When clicked, the EZware-5000 software will copy the text in the current state to every other state. Note: Not available when using Label Library feature.



**Profile tab:**

The profile tab only appears after the object has been created. It can be used to change the location of the object on the screen or the size.

**Position frame:**

use these settings to adjust the location of the object on the OIT screen (with reference to the upper left corner of the object). Note that you can also simply click and drag the object to a new location as well.

**Pinned:** when checked, this feature prevents the programmer from accidentally moving the object on the OIT screen using the 'click and drag' feature of the mouse.

**X:** the horizontal coordinate of the object

**Y:** the vertical coordinate of the object

**Size frame:**

use to adjust the height and width of the object (in pixels). Note that you can also resize the object by clicking once on the object. This will display eight perimeter squares which you can then click and drag to resize the object.

**Width:** total width of the object

**Height:** total height of the object.

**Please Use This Page For Your Notes:**

## Using Index Registers

The Index Register feature is an option available when configuring most graphics objects (i.e. Bit Lamp, Bar Graph, Numeric Input objects). An index register allows you to dynamically (during OIT runtime) change the target PLC address that is read or written to by the graphics object. You don't have to reedit the OIT project and download a new project into the unit. This feature is useful in situations where multiple data registers must be read/written to by the OIT and you want to minimize the number of graphic objects displayed on the OIT screen. With an index register, one graphics object can be used to read or write to multiple PLC registers.

The EZware-5000 software has 32 index registers available. Each index register uses an internal memory address of the OIT:

Index Name	Assigned Internal Memory Address
INDEX 0	LW9200 (16-bit)
INDEX 1	LW9201 (16-bit)
...	...
INDEX 15	LW9215 (16-bit)
INDEX 16	LW9216 (32-bit)
...	...
INDEX 31	LW9260 (32-bit)

*Note: the first 16 index registers (INDEX 0-15) use 16-bit registers with an upper limit range of 65535, the second 16 index registers (INDEX 16-31) use 32-bit registers with an upper limit range of 4,294,967,295.*

## How to configure a graphics object using an index register

In this example, let's configure a Numeric Input object using the Index Register feature:



The target PLC address is MODBUS RTU 4x100 (40001). The index register selected is INDEX 0 (LW9200, 16-bit). If the index register feature were not used, then the OIT would simply read the contents of PLC register 40001 and display the value.

Since the index register feature is enabled, the OIT will first read the value located in INDEX 0 (LW9200) and then add this value to the PLC register 40001. For example, if the value read in INDEX 0 is 5, then the OIT will read PLC register 40006 {40001 [PLC Address] + 5 [INDEX0] = 40006}. You can then create another Numeric Input object (Set Word, Multi-State Switch, or Option List) that is used to change the value in INDEX0.

## Using Internal Data Memory of OIT

The Silver Series contains internal data memory, which can be used to store information that either must be sent to or received from the PLC. The OIT has two types of Internal Data Registers; Local and Recipe.

When using local registers, Local Word and Local Bit, the OIT retains the data stored for as long as the OIT is operating. Once power is removed, all data stored inside the OIT is lost.

When using Recipe registers, Recipe Word and Recipe Bit, the data is stored in RAM, but is written to non-volatile EEPROM every five minutes .

**The following chart shows the memory available in the OIT**

### Reserved Local Bits /System Tags:

Address	Function	Notes
LB-9000	initialized as ON	Bits are set ON at startup, but can be reset and used as needed.
LB-9001	initialized as ON	Bits are set ON at startup, but can be reset and used as needed.
LB-9002	initialized as ON	
LB-9003	initialized as ON	
LB-9004	initialized as ON	
LB-9005	initialized as ON	
LB-9006	initialized as ON	
LB-9007	initialized as ON	
LB-9008	initialized as ON	
LB-9009	initialized as ON	
LB-9010	recipe download indicator	ON during recipe download
LB-9011	recipe upload indicator	ON during recipe upload
LB-9012	recipe download/upload indicator	ON during recipe download or upload
LB-9013	FS window control[hide(ON)/show(OFF)]	Controls ONLY the Fast Select Window
LB-9014	FS button control[hide(ON)/show(OFF)]	Controls ONLY the Fast Select button
LB-9015	FS window/button control[hide(ON)/show(OFF)]	Controls BOTH
LB-9017	disable write-back in PLC control's [change window]	ON to disable screen-number writeback
LB-9018	disable/enable mouse cursor	ON=cursor disabled
LB-9019	disable/enable buzzer	ON=buzzer enabled
LB-9021	reset current event log (set ON)	
LB-9022	delete the oldest event log file (set ON)	
LB-9023	delete all event log files (set ON)	
LB-9024	refresh event log information (set ON)	
LB-9025	delete the oldest data sampling file (set ON)	

LB-9026	delete all data sampling files (set ON)	
LB-9027	refresh data sampling information (set ON)	
LB-9028	reset all recipe data (set ON)	
LB-9029	Save all recipe data to machine (set ON)	
LB-9030	Update COM1 comm. parameters	ON=update
LB-9031	Update COM2 comm. parameters	ON=update
LB-9032	Update COM3 comm. parameters	ON=update
LB-9033	Not used	
LB-9034	Save Event/Data log to HMI	ON=save
LB-9035	HMI memory below limit	ON=alarm
LB-9036	CF memory below limit	ON=alarm
LB-9037	USB1 memory below limit	ON=alarm
LB-9038	USB2 memory below limit	ON=alarm
LB-9039	backup in progress	ON
LB-9040	backlight up (set ON)	Backlight brighter
LB-9041	backlight down (set ON)	Backlight dimmer
LB-9042	acknowledge all events	Set ON
LB-9043	unacknowledged events exist	Set by the OIT
LB-9050	change security to level 0 (set ON)	
LB-9051	change security to level 1 (set ON)	
LB-9052	change security to level 2 (set ON)	
LB-9053	change security to level 3 (set ON)	
LB-9054	change security to level 4 (set ON)	
LB-9055	change security to level 5 (set ON)	
LB-9060	password error	ON when an invalid password has been entered
LB-9061	update password (set ON)	
LB-9070	Toshiba Timer/Counter write control	set ON before writing to a time or counter in a Toshiba PLC
LB-9100	remote HMI 1 status (set on to retry connection)	OFF when remote OIT is offline
LB-9101	remote HMI 2 status (set on to retry connection)	
LB-9102	remote HMI 3 status (set on to retry connection)	
LB-9103	remote HMI 4 status (set on to retry connection)	
LB-9104	remote HMI 5 status (set on to retry connection)	
LB-9105	remote HMI 6 status (set on to retry connection)	
LB-9106	remote HMI 7 status (set on to retry connection)	
LB-9107	remote HMI 8 status (set on to retry connection)	
LB-9150	Auto connection for PLC 1 (COM1) (when ON)	ON to automatically restart communication
LB-9151	Auto connection for PLC 2 (COM2) (when ON)	
LB-9152	Auto connection for PLC 3 (COM3) (when ON)	

LB-9153	Auto connection for PLC 4 (ethernet) (when ON)	
LB-9154	Auto connection for PLC 5 (ethernet) (when ON)	
LB-9155	Auto connection for PLC 6 (ethernet) (when ON)	
LB-9156	Auto connection for PLC 7 (ethernet) (when ON)	
LB-9157	Auto connection for PLC 8 (ethernet) (when ON)	
LB-9158	Auto connection for PLC 9 (ethernet) (when ON)	
LB-9200	PLC 1 status (SN0, COM1), set on to retry connection	Automatically resets when comm fails
LB-9201	PLC 1 status (SN1, COM1), set on to retry connection	
LB-9202	PLC 1 status (SN2, COM1), set on to retry connection	
LB-9203	PLC 1 status (SN3, COM1), set on to retry connection	
LB-9204	PLC 1 status (SN4, COM1), set on to retry connection	
LB-9205	PLC 1 status (SN5, COM1), set on to retry connection	
LB-9206	PLC 1 status (SN6, COM1), set on to retry connection	
LB-9207	PLC 1 status (SN7, COM1), set on to retry connection	
LB-9500	PLC 2 status (SN0, COM2), set on to retry connection	
LB-9501	PLC 2 status (SN1, COM2), set on to retry connection	
LB-9502	PLC 2 status (SN2, COM2), set on to retry connection	
LB-9503	PLC 2 status (SN3, COM2), set on to retry connection	
LB-9504	PLC 2 status (SN4, COM2), set on to retry connection	
LB-9505	PLC 2 status (SN5, COM2), set on to retry connection	
LB-9506	PLC 2 status (SN6, COM2), set on to retry connection	
LB-9507	PLC 2 status (SN7, COM2), set on to retry connection	
LB-9800	PLC 3 status (SN0, COM3), set on to retry connection	
LB-9801	PLC 3 status (SN1, COM3), set on to retry connection	
LB-9802	PLC 3 status (SN2, COM3), set on to retry connection	
LB-9803	PLC 3 status (SN3, COM3), set on to retry connection	
LB-9804	PLC 3 status (SN4, COM3), set on to retry connection	
LB-9805	PLC 3 status (SN5, COM3), set on to retry connection	

LB-9806	PLC 3 status (SN6, COM3), set on to retry connection	
LB-9807	PLC 3 status (SN7, COM3), set on to retry connection	
LB-10100	PLC 4 status (ethernet), set on to retry connection	
LB-10400	PLC 5 status (ethernet), set on to retry connection	
LB-10700	PLC 6 status (ethernet), set on to retry connection	
LB-11100	PLC 7 status (ethernet), set on to retry connection	
LB-11300	PLC 8 status (ethernet), set on to retry connection	
LB-11600	PLC 9 status (ethernet), set on to retry connection	

### Reserved Local Word (LW) / System Tags Table

Address	Function	Notes
LW-9002 (32bit-float)	input high limit	Format is float regardless.
LW-9004 (32bit-float)	input low limit	This bit is set when a recipe is downloading to the PLC
LW-9006 (16bit-uns)	number of remote HMIs connected	
LW-9010 (16bit-BCD)	local second	OIT real-time clock data in BCD format
LW-9011 (16bit-BCD)	local minute	
LW-9012 (16bit-BCD)	local hour	
LW-9013 (16bit-BCD)	local day	
LW-9014 (16bit-BCD)	local month	
LW-9015 (16bit-BCD)	local year	
LW-9016 (16bit-BCD)	local week	
LW-9017 (16bit)	local second	OIT real-time clock data in int format
LW-9018 (16bit)	local minute	
LW-9019 (16bit)	local hour	
LW-9020 (16bit)	local day	
LW-9021 (16bit)	local month	
LW-9022 (16bit)	local year	
LW-9023 (16bit)	local week	
LW-9025 (16bit)	CPU loading %	0-100
LW-9030 (32bit)	system time (unit 0.1 second)	
LW-9040 (16bit)	backlight index	Backlight brightness level
LW-9050 (16bit)	current base window ID	
LW-9060 (16bit)	no. of event log files	
LW-9061 (32bit)	size of event log files	
LW-9063 (16bit)	no. of data sampling files	
LW-9064 (32bit)	size of data sampling files	
LW-9070 (16bit)	number of KBytes of free space before warning is triggered	
LW-9071 (16bit)	reserved number of KBytes of free space	
LW-9072 (32bit)	current free HMI memory (KBytes)	
LW-9074 (32bit)	current free CF memory (KBytes)	



LW-9076 (32bit)	current free USB1 memory (KBytes)	
LW-9078 (32bit)	current free USB2 memory (KBytes)	
LW-9100 (16bit)	project name (16 words)	
LW-9116 (32bit)	project size in bytes	
LW-9118 (32bit)	project size in K bytes	
LW-9120 (32bit)	compiler version	
LW-9122 (16bit)	project compiled date [year]	
LW-9123 (16bit)	project compiled date [month]	
LW-9124 (16bit)	project compiled date [day]	
LW-9125 (16bit)	ethernet gateway 0 (machine used only)	
LW-9126 (16bit)	ethernet gateway 1 (machine used only)	
LW-9127 (16bit)	ethernet gateway 2 (machine used only)	
LW-9128 (16bit)	ethernet gateway 3 (machine used only)	
LW-9129 (16bit)	ethernet IP 0 (machine used only)	
LW-1930 (16bit)	ethernet IP 1 (machine used only)	
LW-1931 (16bit)	ethernet IP 2 (machine used only)	
LW-1932 (16bit)	ethernet IP 3 (machine used only)	
LW-9133 (16bit)	ethernet port no.	
LW-9134 (16bit)	language mode	
LW-9150 (32 words)	keyboard's input data (ASCII)	
LW-9200 (16bit)	address index 0	
LW-9201 (16bit)	address index 1	
LW-9202 (16bit)	address index 2	
LW-9203 (16bit)	address index 3	
LW-9204 (16bit)	address index 4	
LW-9205 (16bit)	address index 5	
LW-9206 (16bit)	address index 6	
LW-9207 (16bit)	address index 7	
LW-9208 (16bit)	address index 8	
LW-9209 (16bit)	address index 9	
LW-9210 (16bit)	address index 10	
LW-9211 (16bit)	address index 11	
LW-9212 (16bit)	address index 12	
LW-9213 (16bit)	address index 13	
LW-9214 (16bit)	address index 14	
LW-9215 (16bit)	address index 15	
LW-9219 (16bit)	user no. (user 1 ~ user 3)	
LW-9220 (32bit)	password	
LW-9222 (16bit)	classes available to current user	Bit0=class A; Bit5=class F
LW-9230 (32bit)	address index 16	
LW-9232 (32bit)	address index 17	
LW-9234 (32bit)	address index 18	
LW-9236 (32bit)	address index 19	
LW-9238 (32bit)	address index 20	
LW-9240 (32bit)	address index 21	
LW-9242 (32bit)	address index 22	

LW-9244 (32bit)	address index 23	
LW-9246 (32bit)	address index 24	
LW-9248 (32bit)	address index 25	
LW-9250 (32bit)	address index 26	
LW-9252 (32bit)	address index 27	
LW-9254 (32bit)	address index 28	
LW-9256 (32bit)	address index 29	
LW-9258 (32bit)	address index 30	
LW-9260 (32bit)	address index 31	
LW-9300 (16bit)	driver ID for local PLC 1	
LW-9301 (16bit)	driver ID for local PLC 2	
LW-9302 (16bit)	driver ID for local PLC 3	
LW-9303 (16bit)	driver ID for local PLC 4	
LW-9350 (16bit)	pending command no. in local HMI	
LW-9351 (16bit)	pending command no. in PLC 1 (COM 1)	
LW-9352 (16bit)	pending command no. in PLC 2 (COM 2)	
LW-9353 (16bit)	pending command no. in PLC 3 (COM 3)	
LW-9354 (16bit)	pending command no. in PLC 4 (ethernet)	
LW-9355 (16bit)	pending command no. in PLC 5 (ethernet)	
LW-9356 (16bit)	pending command no. in PLC 6 (ethernet)	
LW-9357 (16bit)	pending command no. in PLC 7 (ethernet)	
LW-9400 (16bit)	error code for PLC 0	
LW-9401 (16bit)	error code for PLC 1	
LW-9402 (16bit)	error code for PLC 2	
LW-9403 (16bit)	error code for PLC 3	
LW-9404 (16bit)	error code for PLC 4	
LW-9405 (16bit)	error code for PLC 5	
LW-9406 (16bit)	error code for PLC 6	
LW-9407 (16bit)	error code for PLC 7	
LW-9500 (32bit)	user 1's password (level 1)	
LW-9502 (32bit)	user 1's password (level 2)	
LW-9504 (32bit)	user 1's password (level 3)	
LW-9506 (32bit)	user 1's password (level 4)	
LW-9508 (32bit)	user 1's password (level 5)	
LW-9510 (32bit)	user 1's password (level 6)	
LW-9512 (32bit)	user 2's password (level 1)	
LW-9514 (32bit)	user 2's password (level 2)	
LW-9516 (32bit)	user 2's password (level 3)	
LW-9518 (32bit)	user 2's password (level 4)	
LW-9520 (32bit)	user 2's password (level 5)	
LW-9522 (32bit)	user 2's password (level 6)	
LW-9540 (16bit)	reserved for caps lock	
LW-9541 (16bit)	MODBUS slave station number for COM1	
LW-9542 (16bit)	MODBUS slave station number for COM2	
LW-9543 (16bit)	MODBUS slave station number for COM3	

LW-9544 (16bit)	MODBUS slave station number for Ethernet	
LW-9550 (16bit)	COM1 Mode	0=RS232; 1=RS485 2W, 2=RS485 4W
LW-9551 (16bit)	COM1 Baud	0=4800; 1=9600; 2=19200; 3=38400; 4=57600; 5=115200
LW-9552 (16bit)	COM1 Data bits	7=7 data bits; 8=8 data bits
LW-9553 (16bit)	COM1 Parity	0=none; 1=even; 2=odd
LW-9554 (16bit)	COM1 Stop bits	1=1 stop bit; 2=2 stop bits
LW-9555	Not used	
LW-9556 (16bit)	COM2 Baud	
LW-9557 (16bit)	COM2 Data bits	
LW-9558 (16bit)	COM2 Parity	
LW-9559 (16bit)	COM2 Stop bits	
LW-9560 (16bit)	COM3 mode	
LW-9561 (16bit)	COM3 Baud	
LW-9562 (16bit)	COM3 Data bits	
LW-9563 (16bit)	COM3 Parity	
LW-9564 (16bit)	COM3 Stop bits	
LW-9600 (16bit)	PLC4, IP0	Format IP0.IP1.IP2.IP3
LW-9601 (16bit)	PLC4, IP1	
LW-9602 (16bit)	PLC4, IP2	
LW-9603 (16bit)	PLC4, IP3	
LW-9604 (16bit)	PLC4, IP Port	
LW-9605 (16bit)	PLC5, IP0	
LW-9606 (16bit)	PLC5, IP1	
LW-9607 (16bit)	PLC5, IP2	
LW-9608 (16bit)	PLC5, IP3	
LW-9609 (16bit)	PLC5, IP Port	
LW-9610 (16bit)	PLC6, IP0	
LW-9611 (16bit)	PLC6, IP1	
LW-9612 (16bit)	PLC6, IP2	
LW-9613 (16bit)	PLC6, IP3	
LW-9614 (16bit)	PLC6, IP Port	
LW-9615 (16bit)	PLC7, IP0	
LW-9616 (16bit)	PLC7, IP1	
LW-9617 (16bit)	PLC7, IP2	
LW-9618 (16bit)	PLC7, IP3	
LW-9619 (16bit)	PLC7, IP Port	
LW-9620 (16bit)	PLC8, IP0	
LW-9621 (16bit)	PLC8, IP1	
LW-9622 (16bit)	PLC8, IP2	
LW-9623 (16bit)	PLC8, IP3	
LW-9624 (16bit)	PLC8, IP Port	
LW-9625 (16bit)	PLC9, IP0	
LW-9626 (16bit)	PLC9, IP1	
LW-9627 (16bit)	PLC9, IP2	
LW-9628 (16bit)	PLC9, IP3	

LW-9629 (16bit)	PLC9, IP Port	
LW-9770 (16bit)	remote printer server IP0	format: IP0.IP1.IP2.IP3
LW-9771 (16bit)	remote printer server IP1	
LW-9772 (16bit)	remote printer server IP2	
LW-9773 (16bit)	remote printer server IP3	
LW-9774 (6 words)	remote printer server password	
LW-9800 (16bit)	remote HMI #1 IP0	format: IP0.IP1.IP2.IP3
LW-9801 (16bit)	remote HMI #1 IP1	
LW-9802 (16bit)	remote HMI#1 IP2	
LW-9803 (16bit)	remote HMI#1 IP3	
LW-9804 (16bit)	remote HMI #1 Port	
LW-9805 (16bit)	remote HMI#2 IP 0	
LW-9806 (16bit)	remote HMI#2 IP 1	
LW-9807 (16bit)	remote HMI #2 IP 2	
LW-9808 (16bit)	remote HMI#2 IP3	
LW-9809 (16bit)	remote HMI #2 Port	
LW-9810 (16bit)	remote HMI #3 IP 0	
LW-9811 (16bit)	remote HMI #3 IP 1	
LW-9812 (16bit)	remote HMI #3 IP 2	
LW-9813 (16 bit)	remote HMI #3 IP 3	
LW-9814 (16bit)	remote HMI #3 Port	
LW-9815 (16bit)	remote HMI#4 IP0	
LW-9816 (16bit)	remote HMI #4 IP1	
LW-9817 (16bit)	remote HMI#4 IP2	
LW-9818 (16bit)	remote HMI#4 IP3	
LW-9819 (16bit)	remote HMI#4 Port	
LW-9820 (16bit)	remote HMI#5 IP 0	
LW-9821 (16bit)	remote HMI #5 IP1	
LW-9822 (16bit)	remote HMI#5 IP2	
LW-9823 (16bit)	remote HMI#5 IP 3	
LW-9824 (16bit)	remote HMI #5 Port	
LW-9825 (16bit)	remote HMI #6 IP 0	
LW-9826 (16bit)	remote HMI#6 IP1	
LW-9827 (16bit)	remote HMI#6 IP2	
LW-9828 (16bit)	remote HMI #6 IP3	
LW-9829 (16bit)	remote HMI #6 Port	
LW-9830 (16bit)	remote HMI #7 IP 0	
LW-9831 (16bit)	remote HMI #7 IP 1	
LW-9832 (16bit)	remote HMI #7 IP2	
LW-9833 (16bit)	remote HMI #7 IP3	
LW-9834 (16bit)	remote HMI #7 Port	
LW-9835 (16bit)	remote HMI #8 IP 0	
LW-9836 (16bit)	remote HMI #8 IP1	
LW-9837 (16bit)	remote HMI #8 IP2	
LW-9838 (16bit)	remote HMI #8 IP3	
LW-9839 (16bit)	remote HMI #8 Port	

LW-9900	Reserved	
LW-9901 (16bit)	Pass-through source comm. port	1-3
LW-9902 (16bit)	Pass-through destination comm. port	1-3

## Representing PLC Coil Registers

PLC coils (or binary registers) and internal coils of the OIT are represented using three parts: the Bit Lamp Object, the Set Bit Object, and the Toggle Switch Object. Each has a particular function that makes them unique but they are all constructed and used in essentially the same manner.

### *The Bit Lamp Object*

The Bit Lamp Object is used to represent the value of a PLC coil. The object continuously reads the PLC coil and displays the corresponding shape or bitmap that is tagged to the On or Off state of the coil.



► To create a Bit Lamp Object 

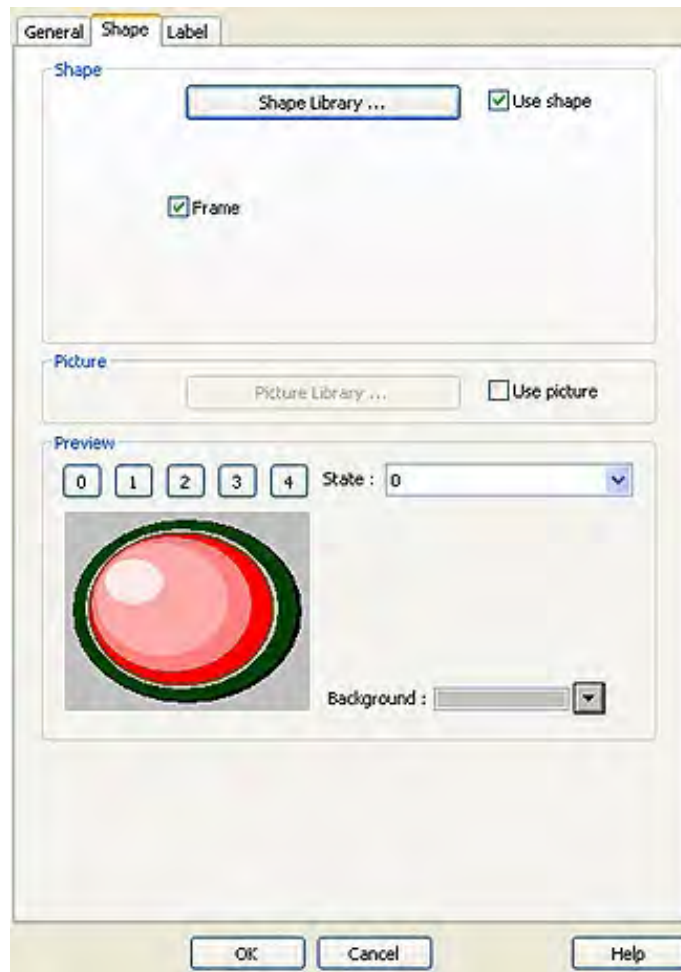
1. From the Objects menu, click **Lamp - Bit Lamp**. Or click the **Bit Lamp** icon in the Part1 toolbar. The New Bit Lamp Object dialog box appears.



2. Use the Description: box to enter a *title* for the Bit Lamp part. A description is not necessary but does help you identify the purpose of the part.
3. In the Read address frame, select the *PLC Name* and *Device type* from the drop down menu, and specify the *PLC coil* or *HMI internal memory address*.
4. In the Blinking frame, select how you want the Bit Lamp object to operate:

Mode	Description
None	Displays State 0 object when clear; displays State 1 object when set.
Alternating Image on State 0	Alternates between state 0 and 1 object when clear; displays the state 1 object when set.
Alternating Image on State 1	Displays the state 1 object when clear; alternates between state 0 and 1 when set.
Blinking on State 0	Displays state 0 object when clear; blinks state 0 object when set
Blinking on State 1	Displays state 0 object when clear; blinks state 1 object when set

5. Click the **Shape** tab to display the Shape form



6. Select either a shape or bitmap to represent the Bit Lamp object. If you need more information on how to do this, consult Chapter 6 “Creating Graphics Objects”.

- Click the **Label** tab to display the Label form.

**New Bit Lamp Object**

General Shape **Label**

Use label  
 Use label library

State : 0 [Left Arrow] [Right Arrow] 0 1

**Attribute**

Font : Arial [Dropdown]  
Color : [Blue] [Dropdown] Size : 16 [Dropdown]  
Align : Left [Dropdown] Blink : None [Dropdown]  
 Italic  Underline  
Duplicate these attributes to every state

**Movement**

Direction : No movement [Dropdown]

**Content**

[Text Area]  
 Tracking Duplicate this label to every state

OK Cancel Help

- Click the **Use label** checkbox if you want to use a label.
- In the Attribute box, select the desire font, font size, alignment and color. Click the **Duplicate this label to every state** button if you want to use the same label for all states.
- You can have a different label for each state of the Bit Lamp. In the **State:** box, enter the state that you wish to edit, then enter the label designations for that state in the **Content:** box. Do the same for other states.
- In the Movement box, select the *direction* you'd like the label to move from the drop down menu if you desire movement.
- In the *Content* box, type the content of the label. If you'd like the label to duplicate to every state, click the **Duplicate this label to every state** button.



13. Click the **Tracking** checkbox if you want the labels for all states to follow or ‘track’ with the movement of one label when it is moved after the Bit Lamp object is placed onto the window.
14. Click **OK**. The New Bit Lamp Object form closes and the main screen of EasyBuilder appears with the cursor tied to a rectangular outline of the part you just created at the upper left corner of the screen. Move the part to the desired location of the window.
15. Once the part is placed onto the window, you can adjust the location of the label inside the part by clicking once on the label. This will highlight the entire object. Now click on the label again. Now only the label is highlighted, allowing you to move it without moving the part. Note: if you double-click (click twice rapidly) then you will not highlight the label but rather enter the Bit Lamp Object’s Attribute dialog box.

## The Set Bit Object

The Set Bit Object is used to write a value to a PLC coil. The touchscreen object displays the State 0 shape until pressed. When pressed, it displays the State 1 shape and executes the function that was assigned to it.

### ► To create a Set Bit Object

1. From the Objects menu, click **Button-Set Bit**. Or click the **Set Bit** icon in the Part1 toolbar. The New Set Bit Object dialog box appears.

The screenshot shows the 'New Set Bit Object' dialog box with the 'General' tab selected. The fields are as follows:

- Description: (empty text box)
- PLC name: Local HMI (dropdown menu)
- Write address:
  - Device type: LB (dropdown menu)
  - Address: 0 (text box)
  - System tag:
  - Index: INDEX 0 (16-bit) (dropdown menu)
  - Index register:
  - Write after button is released:
- Attribute:
  - Set style: Set ON (dropdown menu)
- Macro:
  - Execute macro:

2. Use the Description: box to enter a *title* for the Set Bit part. A description is not necessary but does help you identify the purpose of the part.
3. In the **Write address** frame, select the *Device type* from the drop down menu and specify the PLC coil or HMI internal memory address in the Address: field.
4. In the Attribute frame, use the drop down menu to select how you want the Set Bit object to operate:

Set ON	When touched, sets the Bit, and leaves it set.
Set OFF	When touched, clears the Bit, and leaves it cleared.

Toggle	When touched, toggles the current state of the Bit.
Momentary	Sets the Bit when touched, clears the Bit when released.
Periodic Toggle	Toggles the Bit ON and OFF at the specified time interval.
Set ON When Window Opens	Sets the Bit when the Window containing the Object is displayed.
Set OFF When Window Opens	Clears the Bit when the Window containing the Object is displayed.
Set ON When Window Closes	Sets the Bit when the Window containing the Object is closed.
Set OFF When Window Closes	Clears the Bit when the Window containing the Object is closed.
Set ON When Backlight On	Sets the Bit when the Backlight turns on.
Set OFF When Backlight On	Clears the Bit when the Backlight turns on.
Set ON At Backlight Off	Sets the Bit when the Backlight turns off.
Set OFF At Backlight Off	Clears the Bit when the Backlight turns off.

5. Refer to the Bit Lamp Object section for information on using the Shape tab. Note that it is not required to use a shape or a bitmap. In some modes such as **Set ON at window open** or **Periodical toggle** you may prefer to place the part on the window without a shape or bitmap tied to it.
6. Refer to the Bit Lamp Object section for information on using the Label tab.
7. Click **OK**. The Create Set Bit Object form closes and the main screen of EasyBuilder appears with the cursor tied to a rectangular outline of the part you just created at the upper left corner of the screen. Move the part to the location on the window that you want it.
8. Once the part is placed onto the window, you can adjust the location of the label inside the part by clicking once on the label. This will highlight the entire object. Now click on the label again. Now only the label is highlighted, allowing you to move it without moving the part. Note: if you double-click (click twice rapidly) then you will not highlight the label but rather enter the Set Bit Object's Attribute dialog box.

### **The Toggle Switch Object**

The Toggle Switch Object is used to represent the value of one PLC coil while able to write a value to the same or another PLC coil. The object continuously reads one PLC coil and displays the corresponding shape or bitmap that is tagged to the On or Off state of the coil. Also the touchscreen object when pressed, executes the function that was assigned to it.

► To create a Toggle Switch Object 

1. From the Objects menu, click **Button-Toggle Switch**. Or click the **Toggle Switch** icon in the Part1 toolbar. The New Toggle Switch Object dialog box appears.

2. Use the Description: box to enter a *title* for the Toggle Switch part. A description is not necessary but does help you identify the purpose of the part.
3. In the **Read address** frame, select the PLC coil or HMI internal memory address that the part continuously reads.
4. In the **Write address** frame, specify the *Device type*: from the drop down menu and then select the *PLC coil or HMI internal memory address* that the part writes to when pressed.
5. In the Attribute frame, select how you want the Toggle Switch object to operate:

Switch Style	Description
Set ON	Sets a PLC coil when pressed.
Set OFF	Clears a PLC coil when pressed.
Toggle	Alternates between setting and clearing a PLC coil.

Momentary	Sets a PLC coil when pressed; clears a PLC coil when released
-----------	---

6. Refer to the Bit Lamp Object section for information on using the Shape tab
7. Refer to the Bit Lamp Object section for information on using the Label tab.
8. Click **OK**. The Create Toggle Switch Object form closes and the main screen of EasyBuilder appears with the cursor tied to a rectangular outline of the part you just created at the upper left corner of the screen. Move the part to the desired location in the window.
9. Once the part is placed onto the window, you can adjust the location of the label inside the part by clicking once on the label. This will highlight the entire object. Now click on the label again. Now only the label is highlighted, allowing you to move it without moving the part. Note: if you double-click (click twice rapidly) then you will not highlight the label but rather enter the Toggle Switch Object's Attribute dialog box.

## Representing PLC Data Registers

PLC registers and internal data registers of the OIT are represented using nine parts:

- Word Lamp Object
- Set Word Object
- Multistate Switch Object
- Numeric Display Object
- Numeric Input Extend Object
- ASCII Data Object
- ASCII Input Extend Object
- Moving Shape Object
- Animation Object

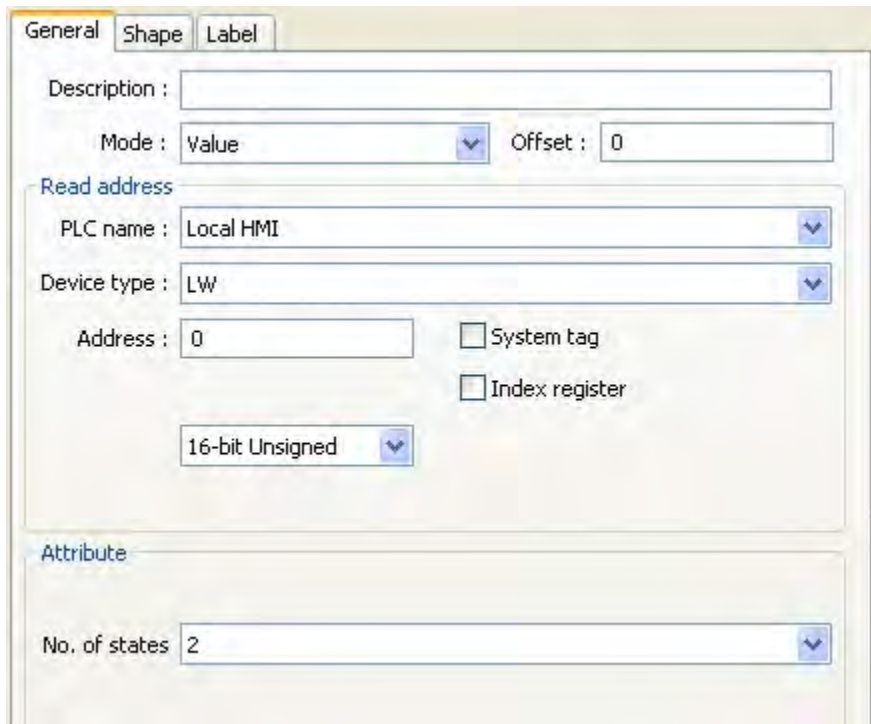
They are constructed in much the same manner as the parts described in the previous section. With these parts, however, you can represent data from 16-bit, 32-bit, or even 64-bit data registers.

### ***The Word Lamp Object***

The Word Lamp Object is used to represent the value of a PLC register. The object continuously reads the PLC register and displays the corresponding shape or bitmap that is tagged to the number of states defined.

► To create a Word Lamp Object 

1. From the Objects menu, click **Lamp-Word Lamp**. Or click the **Word Lamp** icon in the Part1 toolbar. The Create Word Lamp Object dialog box appears.

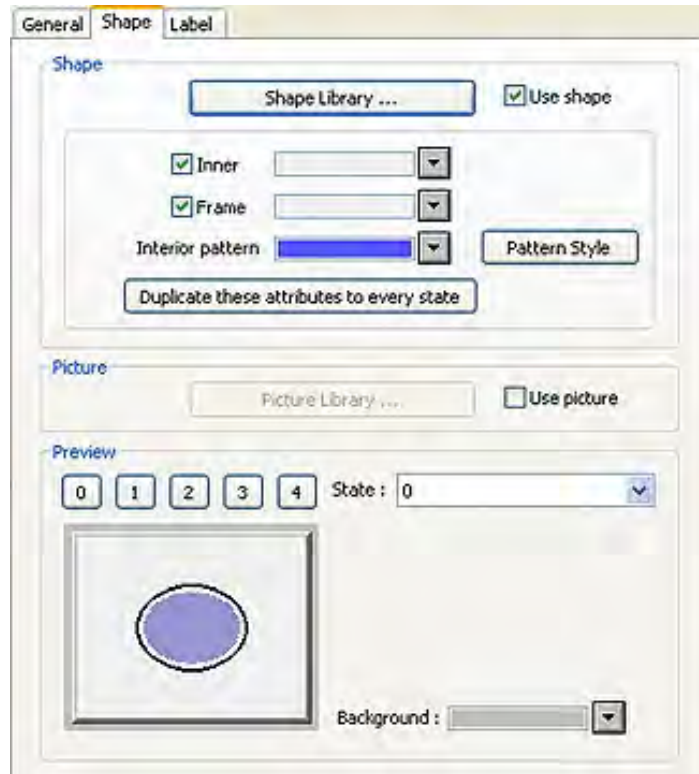


2. Use the Description: box to enter a *title* for the Word Lamp part. A description is not necessary but does help you identify the purpose of the part.
3. Mode: Sets how the object determines the state.

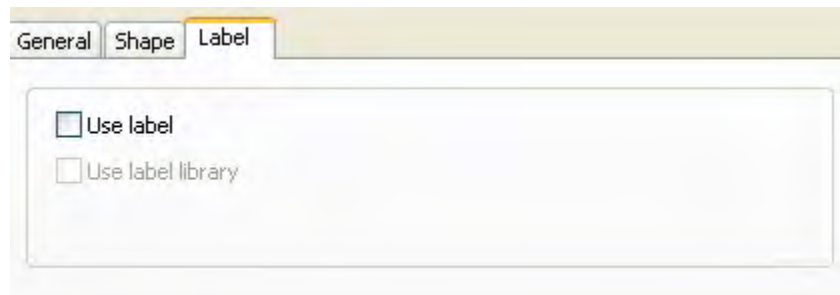
Value	The state of the object is the value in the specified Read Address														
LSB: Note: This mode will not work correctly when data is 32-bit float	<p>The state of the object is determined by the first bit that is set in the value specified by the Read Address:</p> <table border="0"> <tr><td>00000000</td><td>state 0</td></tr> <tr><td>00000001</td><td>state 1</td></tr> <tr><td>00000010</td><td>state 2</td></tr> <tr><td>00000100</td><td>state 3</td></tr> <tr><td>10000000</td><td>state 8</td></tr> </table> <p>If more than one bit is set, the lowest-order bit is used:</p> <table border="0"> <tr><td>00000011</td><td>state 1</td></tr> <tr><td>10001000</td><td>state 4</td></tr> </table>	00000000	state 0	00000001	state 1	00000010	state 2	00000100	state 3	10000000	state 8	00000011	state 1	10001000	state 4
00000000	state 0														
00000001	state 1														
00000010	state 2														
00000100	state 3														
10000000	state 8														
00000011	state 1														
10001000	state 4														
Change State by Time	The state of the object increments automatically at the interval specified by <i>Time Interval</i> . After the last state, the word lamp reverts to state 0.														

4. In the Read address frame, select the *Device type* from the drop down menu, and then select the PLC register address.
5. In the **No. of states:** box, select from 1 to 256 states. If the actual value in the PLC register is out-of-range for the number of states specified, then no shape or bitmap is displayed.

- Click the **Shape** tab to display the Shape form.



- Select either a shape or bitmap to represent the Word Lamp object. If you need more information on how to do this, consult Chapter 6 “Creating Graphics Objects”.
- Click the **Label** tab to display the Label form..



- Click the **Use Label** checkbox if you want to use a label. Click the **Duplicate this label to other states** button if you want to use the same label for all states.
- You can have a different label for each state of the Word Lamp. In the **State:** box, enter the state that you wish to edit, then enter the label for that state in the **Content:** box. Do the same for the other states.
- Select the *font* that you want from the Font: drop down menu, and then select the *font color* and *font size* from the appropriate drop down menus. You can select a different font, color and size for each label state.
- Enter how you want the label to be positioned in the Align: box. You can select a different position for each label state.
- Click the **Tracking** checkbox if you want the labels for all states to follow or ‘track’ with the movement of one label when it is moved after the Word Lamp object is placed onto the window.

14. Click **OK**. The Create Word Lamp Object form closes and the main screen of EasyBuilder appears with the cursor tied to a rectangular outline of the part you just created at the upper left corner of the screen. Move the part to the desired location on the window.
15. Once the part is placed onto the window, you can adjust the location of the label inside the part by clicking once on the label. This will highlight the entire object. Now click on the label again. Now only the label is highlighted, allowing you to move it without moving the part. Note: if you double-click (click twice rapidly) then you will not highlight the label but rather enter the Word Lamp Object's Attribute dialog box.

## The Set Word Object

The Set Word Object is used to write a value to a PLC register. The touchscreen object displays the State 0 shape until pressed. When pressed, it displays the State 1 shape and executes the function that was assigned to it.

### ► To create a Set Word Object



1. From the Objects menu, click **Button-Set Word**. Or click the **Set Word** icon in the Part1 toolbar. The New Set Word Object dialog box appears.

The dialog box is titled 'New Set Word Object' and has four tabs: General, Security, Shape, and Label. The 'General' tab is selected. It contains the following sections:

- Description:** A text input field.
- Write address:**
  - PLC name: Local HMI (dropdown)
  - Device type: LW (dropdown)
  - Address: 0 (text input)
  - 16-bit Unsigned (dropdown)
  - System tag:
  - Index register:
  - Write after button is released:
- Notification:**
  - Enable:
  - Set ON:  Set OFF:
  - Before writing:  After writing:
  - PLC name: Local HMI (dropdown)
  - Device type: LB (dropdown)
  - Address: ch.19 (text input)
  - System tag:
  - Index register:
- Attribute:**
  - Set Style: Write constant value (dropdown)
  - Set value: 0 (text input)

2. Use the Description: box to enter a *title* for the Set Word part. A description is not necessary but does help you identify the purpose of the part.
3. In the Write address frame, select *Device type* from the drop down menu, and then specify the *PLC coil or HMI internal memory address*.

4. Select the *data format*.
5. In the Attribute frame, select how you want the Set Word object to operate:

Write Constant Value	When touched, sets the Write Address to the specified value.
Increment Value (JOG+)	Each time it is touched, increments the value in the Write Address by the value specified by Inc. Value, up to the value specified by Upper Limit.
Decrement Value (JOG-)	Each time it is touched, decrements the value in the Write Address by the value specified by Dec. Value, down to the value specified by Bottom Limit.
Press and Hold Increment (JOG++)	When touched, increments the value in the Write Address by the value specified by Inc. Value, up to the value specified by Upper Limit. When held for the time specified by Jog Delay, continues to increment the value in the Write Address, at the rate specified by Jog Speed.
Press and Hold Decrement (JOG—)	When touched, decrements the value in the Write Address by the value specified by Dec. Value, down to the value specified by Bottom Limit. When held for the time specified by Jog Delay, continues to decrement the value in the Write Address, at the rate specified by Jog Speed.
Periodic JOG++	Increments the value in the Write Address by the value specified by Inc. Value, up to the value specified by Upper Limit, at the rate indicated by Time Interval.
Automatic JOG—	Decrement the value in the Write Address by the value specified by Dec. Value, down to the value specified by Bottom Limit, at the rate indicated by Time Interval.
Periodic Bounce	Similar to Periodic Jog++, except that the value begins decrementing when it reaches the Upper Limit. Then, the value begins incrementing again.
Periodic Step Up	Starting with the value in the Low Limit box, increments the value in the specified register by 1, until the value in the High Limit box is reached, at the rate specified by the Time Interval. When the High Limit is reached, the register value is reset to the Low Limit, and the incrementing repeats.
Periodic Step Down	Starting with the value in the High Limit box, decrements the value in the specified register by 1, until the value in the Low Limit box is reached, at the rate specified by the Time Interval. When the Low Limit is reached, the register value is reset to the High Limit, and the decrementing repeats.
Set When Window Opens	Sets the Write Address to the value specified by Set Value when the Window containing the Object is displayed.
Set When Window Closes	Sets the Write Address to the value specified by Set Value, when the Window containing the Object is closed.
Set When Backlight On	Writes the specified value to the Write Address when the Backlight turns on.
Set When Backlight Off	Writes the specified value to the Write Address when the Backlight turns off.

6. Refer to the Bit Lamp Object section for information on using the **Shape** tab. Note that it is not required to use a shape or a bitmap. In some modes such as **Set When Window Opens** or **Periodic JOG++** you may prefer to place the part on the window without a shape or bitmap tied to it.
7. Refer to the Bit Lamp Object section for information on using the Label tab.
8. Click **OK**. The New Set Word Object dialog closes and the main screen of EasyBuilder appears with the cursor tied to a rectangular outline of the part you just created at the upper left corner of the screen. Move the part to the desired location on the window.
9. Once the part is placed onto the window, you can adjust the location of the label inside the part by clicking once on the label. This will highlight the entire object. Click on the label again. Now only the label is highlighted, allowing you to move it without moving the part. Note: if you double-click (click twice rapidly) then you will not highlight the label but rather enter the Set Word Object's Attribute dialog box.

## The Slider Object

The slider object is used to write data to a PLC Register. It also reflects the value in the PLC register. Data is written to the register when the slider is released. The following data formats are supported:

- 16-bit BCD
- 32-bit BCD
- 16-bit Unsigned
- 16-bit Signed
- 32-bit Unsigned



- 32-bit Signed
- 32-bit Float

► **To create a slider object:**

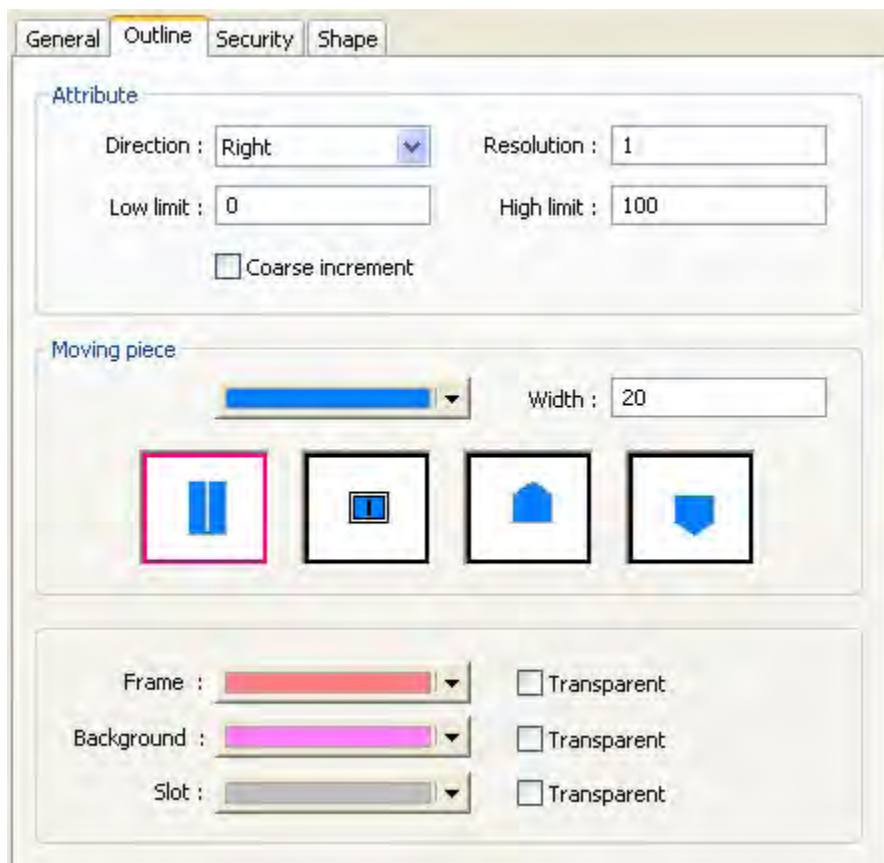
1. From the Objects menu, select **Button-Slider**. Or, click the slider icon from the object toolbar. The New Slider Object Dialog appears.

The image shows the 'New Slider Object Dialog' with the 'General' tab selected. The dialog is divided into several sections:

- Description:** A text input field.
- Write address:** A section containing:
  - PLC name:** A dropdown menu set to 'Local HMI'.
  - Device type:** A dropdown menu set to 'LW'.
  - Address:** A text input field set to '0'.
  - 16-bit Unsigned:** A dropdown menu.
  - System tag:** An unchecked checkbox.
  - Index register:** An unchecked checkbox.
- Notification:** A section containing:
  - Enable:** A checked checkbox.
  - Set ON / Set OFF:** Radio buttons, with 'Set OFF' selected.
  - Before writing / After writing:** Checkboxes, with 'Before writing' checked.
  - PLC name:** A dropdown menu set to 'Local HMI'.
  - Device type:** A dropdown menu set to 'LB'.
  - Address:** A text input field set to 'ch.19'.
  - System tag:** An unchecked checkbox.
  - Index register:** An unchecked checkbox.

2. Use the Description: box to enter a title for the slider part. A description is not necessary but does help you identify the purpose of the part.
3. In the **Write address** frame, specify the *Device type*: from the drop down menu and then select the *PLC coil* or *HMI internal memory address* that the part writes to when pressed.

- Click the Outline tab:



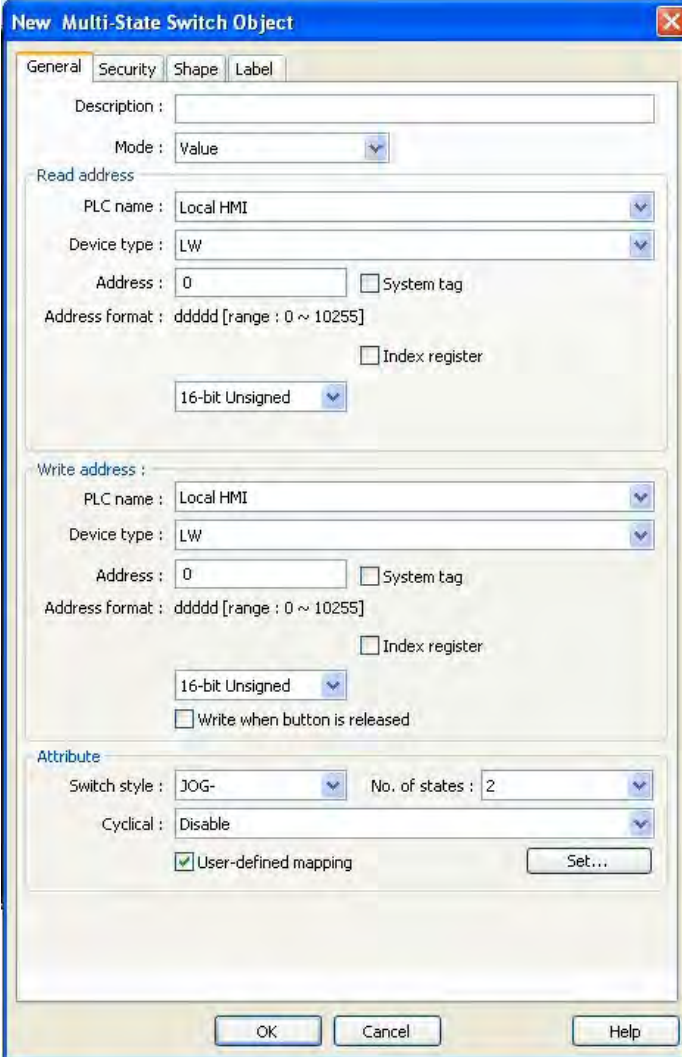
- In the Attribute section, select the *direction* that you want the slider to move. Next, set the *resolution* as the amount by which the register value changes as the slider is moved. Finally, set the *high limit* and *low limit* as the minimum and maximum values to be written to the register. Check the **coarse increment** check box to allow the slider to change register by a value larger than that specified by the Resolution setting and then select the increment by which the register will be changed.
- In the Moving piece section, select the *color*, *size*, and *appearance* of the slider's pointer.
- In the colors section, select the *color* and *transparency* for the *frame*, *background* and *slot* of the slider.

### **The MultiState Switch Object**

The MultiState Switch Object is used to represent the value of one 16-bit, 32-bit or 64-bit PLC register while writing a 16-bit, 32-bit or 64-bit value to the same or another PLC register each time the object is pressed. The object continuously reads one PLC register and displays the corresponding shape or bitmap that is tagged to the defined states of the register. For example, if the value read in the PLC is 0, then the MultiState Switch displays the graphic picture or text label that is assigned to State 0. The touchscreen object when pressed, executes the function that was assigned to it.

► To create a MultiState Switch Object 

1. From the Objects menu, click **Button-MultiState Switch**. Or click the **MultiState Switch** icon in the Part1 toolbar. The New MultiState Switch Object dialog box appears.



**New Multi-State Switch Object**

General Security Shape Label

Description :

Mode : Value

Read address

PLC name : Local HMI

Device type : LW

Address : 0  System tag

Address format : dddd [range : 0 ~ 10255]

Index register

16-bit Unsigned

Write address :

PLC name : Local HMI

Device type : LW

Address : 0  System tag

Address format : dddd [range : 0 ~ 10255]

Index register

16-bit Unsigned

Write when button is released

Attribute

Switch style : JOG-  No. of states : 2

Cyclical : Disable

User-defined mapping

OK Cancel Help

2. **Use the Description box:** to enter a *title* for the MultiState Switch part. A description is not necessary but does help you identify the purpose of the part.
3. **Value:** The state of the object is the value in the specified Read Address. Offset option, if used, is subtracted from the value read to determine the State that is displayed. For example, if offset is 2, Read Address is set to LW200 and the value read is 5, then the state displayed is 3 ( 5[LW200] – 2 [Offset] = 3).

Value	The state of the object is the value in the specified Read Address.
LSB: Note: This Mode will not work correctly when the data format is 32-bit float	<p>The state of the object is determined by the first bit (from right to left) that is set in the value specified by the Read Address:</p> <p>00000000    state 0  00000001    state 1  00000010    state 2  00000100    state 3  10000000    state 8</p> <p>If more than one bit is set, the lowest-order bit is used:</p> <p>00000011    state 1  10001000    state 4</p>

4. **Read Address frame:** this is the target PLC address that is read by the OIT to determine what state to display. For more information, see General Properties section at the beginning of this chapter.
5. **Write Address frame:** this is the target PLC address that is written to by the OIT when the Multi-State switch button is pressed. For more information, see General Properties section at the beginning of this chapter.
  - a. Write when button is release checkbox: check this box to write to the PLC after the Multi-State Switch has been released.
6. **Attribute frame:** User-defined mapping: this feature allows you to create a table that matches each State to a unique constant number. To use this feature, check the User-defined mapping checkbox, then click on the Set.. button:

**Mapping**

State	Value
0	0
1	20
2	40
3	60
4	80
5 (error)	

Set default

OK

Cancel

Input illegal

Remain current state     Jump to error state

Notification

Enable     Set ON     Set OFF

PLC name : Local HMI

Device type : LB

Address : 0     System tag

Address format : ddddd [range : 0 ~ 11999]

Index register

The number of states shown depends upon the number of states selected in the Attribute frame. In this example, there are a total of five states. Click on the entry in the Value column for each State to enter a new constant value:

State	Value
0	0
▶ 1	20
2	40
...	...

*Note: You cannot enter a value for the last (error) state. The error state is always represented by any value that is not represented by one of the other states.*

*Note: Negative values are not supported. Floating point values supported.*

*Also note that the values entered for each state are unique- not related to other values entered for other states.*

**For example:** entering a value of 14 for State 2 in the example above is acceptable.

**Input illegal frame:** this setting determines how the Multi-State Switch will ‘react’ if the value read from the PLC register does not exactly match any of the predefined values assigned to each state.

**Remain current state:** select this option if you want the Multi-State Switch to simply remain in the current state if an invalid value is read.

**Jump to error state:** select this option if you want the Multi-State Switch to display the ‘error state’ which is simply a blank (no picture or label is displayed). The Multi-State Switch will remain blank and inoperative until a valid state value is read from the PLC.

**Notification frame:** this optional feature can be used to set (or clear) a target PLC bit when the Multi-State Switch object reads a value in the PLC register that is invalid. Note: the PLC must clear this bit after notification from the OIT.

**Set ON or Set OFF:** select which action the HMI takes when an error occurs.

Switch Style	Option	Description
JOG+		Increments the PLC register by 1 each time the key is pressed. When the highest state is reached, the key resets to State 0.
	No. of states:	Range: 1 to 255
JOG-		Decrements the PLC register by 1 each time the key is pressed. When the State 0 is reached, the key resets to the highest state.
	No. of states:	Range: 1 to 255

Cyclical	Option	Description
	Disable	When the value in the specified register reaches its maximum, the next touch will have no effect on the value.
	Enable	When the value in the specified register reaches its maximum, the next touch will cause the value to reset to 0.

- Click **OK**. The New MultiState Switch Object form closes and the main screen of EasyBuilder appears with the cursor tied to a rectangular outline of the part you just created at the upper left corner of the screen. Move the part to the desired location on the window.
- Once the part is placed onto the window, you can adjust the location of the label inside the part by clicking once on the label. This will highlight the entire object. Now click on the label again. Now only the label is highlighted, allowing you to move it without moving the part. Note: if you double-click (click twice rapidly) then you will not highlight the label but rather enter the MultiState Switch Object's Properties dialog box.

### The Numeric Display Object

The Numeric Display Object is used to display the numeric value of a PLC register. The object continuously reads the PLC register and displays the corresponding numeric data in the format specified. The PLC register can be a 16-bit, 32-bit, or a 64-bit word.

#### ► To create a Numeric Display Object



- From the Objects menu, click **Numeric/ASCII-Numeric Display**. Or click the **Numeric Display** icon in the Part1 toolbar. The New Numeric Data Object dialog box appears.

- Use the Description: box to enter a *title* for the *numeric display* part. A description is not necessary but does help you identify the purpose of the part.
- In the Read address frame, select the *device type* from the drop down menu and then specify the *PLC register address* that the part continuously reads.
- Click the **Numeric Format** tab to display the Numeric format form.
- In the Display attribute box, select the format type you wish to use. The options are:

Format	Range	Description
16-bit BCD	0 to 32,767 BCD	16-bit Binary-Coded Decimal
32-bit BCD	0 to 65,536 BCD	32-bit Binary-Coded Decimal
16-bit Hex	0 to FFFF	16-bit Hexadecimal
32-bit Hex	0 to FFFFFFFF	32-bit Hexadecimal

16-bit Binary	00... to 11... (16 bits)	16 Binary Digits
32-bit Binary	00... to 11... (32 bits)	32 Binary Digits
16-bit Unsigned	0 to 65,537	16-bit Unsigned Integer
16-bit Signed	-32,768 to +32,768	16-bit Signed Integer
32-bit Unsigned	0 to 4,294,967,295	32-bit Unsigned Integer
32-bit Signed	-2,147,483,647 to +2,147,483,647	32-bit Signed Integer
32-bit Float	-9999999999.999999999 to 9999999999.999999999	IEEE754 Single Precision

6. Check the *Mask* box to hide the value with asterisks.
7. In the Number of Digits section, select how the data is to be represented.

Format	Options	Range	Description
16-bit BCD	Left of Dec. Pt.	1 to 11	Digits to display to the left of the Decimal Point
	Right of Dec. Pt.	0 to 11	Digits to display to the right of the Decimal Point
32-bit BCD	Left of Dec. Pt.	1 to 11	Digits to display to the left of the Decimal Point
	Right of Dec. Pt.	0 to 11	Digits to display to the right of the Decimal Point
16-bit Hex	Left of Dec. Pt.	N/A	Displays 4 hex digits to the left of the Decimal Point
	Right of Dec. Pt.	N/A	No digits are displayed to the right of the Decimal Point
32-bit Hex	Left of Dec. Pt.	N/A	Displays 8 hex digits to the left of the Decimal Point
	Right of Dec. Pt.	N/A	No digits are displayed to the right of the Decimal Point
16-bit Binary	Left of Dec. Pt.	N/A	Displays 16 binary digits to the left of the Decimal Point
	Right of Dec. Pt.	N/A	No digits are displayed to the right of the Decimal Point
32-bit Binary	Left of Dec. Pt.	N/A	Displays 32 binary digits to the left of the Decimal Point
	Right of Dec. Pt.	N/A	No digits are displayed to the right of the Decimal Point
16-bit Unsigned	Left of Dec. Pt.	1 to 11	Digits to display to the right of the Decimal Point
	Right of Dec. Pt.	0 to 11	Digits to display to the left of the Decimal Point
16-bit Signed	Left of Dec. Pt.	1 to 11	Digits to display to the right of the Decimal Point
	Right of Dec. Pt.	0-11	Digits to display to the left of the Decimal Point
32-bit Unsigned	Left of Dec. Pt.	1 to 11	Digits to display to the right of the Decimal Point
	Right of Dec. Pt.	0-11	Digits to display to the left of the Decimal Point
32-bit Signed	Left of Dec. Pt.	1 to 11	Digits to display to the right of the Decimal Point
	Right of Dec. Pt.	0-11	Digits to display to the left of the Decimal Point
32-bit Float	Left of Dec. Pt.	1 to 11	Digits to display to the right of the Decimal Point
	Right of Dec. Pt.	0-11	Digits to display to the left of the Decimal Point

8. The Unsigned, Signed, and Float formats have the option of scaling the data. To display a scaled value, check the *Do Conversion* box. In the *Engineering Low* field, enter the *lowest value* that the OIT is to display. In the *Engineering High* field, enter the *highest value* that the OIT is to display. The conversion will also use the Low and High limits specified in the Limits section.
9. In the *Limits* section, first select if the Input Low and High values will be entered directly, or read from a series of PLC registers. If read from a register, the Low Limit is read from the specified register, and the High Limit is read from the next consecutive register. In either case, the valid ranges are listed below.

Format	Options	Range
16-bit BCD	Low Limit	0
	High Limit	65,535
32-bit BCD	Low Limit	0
	High Limit	4,294,967,295
16-bit Hex	Low Limit	0
	High Limit	65,535
32-bit Hex	Low Limit	0
	High Limit	4,294,967,295
16-bit Binary	Low Limit	0
	High Limit	65,535
32-bit Binary	Low Limit	0
	High Limit	4,294,967,295
16-bit Unsigned	Low Limit	0
	High Limit	65535
16-bit Signed	Low Limit	-32,768
	High Limit	32,768
32-bit Unsigned	Low Limit	0
	High Limit	4,294,967,295
32-bit Signed	Low Limit	-2,147,483,647
	High Limit	+2,147,483,647
32-bit Float	Low Limit	-9999999999.9999999999
	High Limit	+9999999999.9999999999

10. Click the **Font** tab to configure the font.
11. Select your *Font* from the drop down menu. Enter the *color* that you want for the digits displayed in the color: box and then select the *font size*.
12. Enter how you want the digits to be positioned in the Align: box. The choices are:
  - **Right Adjust** - aligns the number to the right margin of the part.
  - **Left Adjust** - aligns the number to the left margin of the part.
  - **Leading Zero** - inserts leading zeroes
13. Click **OK**. The Numeric Data Object form closes and the main screen of EasyBuilder appears with the cursor tied to a rectangular outline of the part you just created at the upper left corner of the screen. Move the part to the desired location on the window.
14. Once the part is placed onto the window, you can modify the attributes by double-clicking on the part to display the Numeric Data Object's Attribute dialog box.

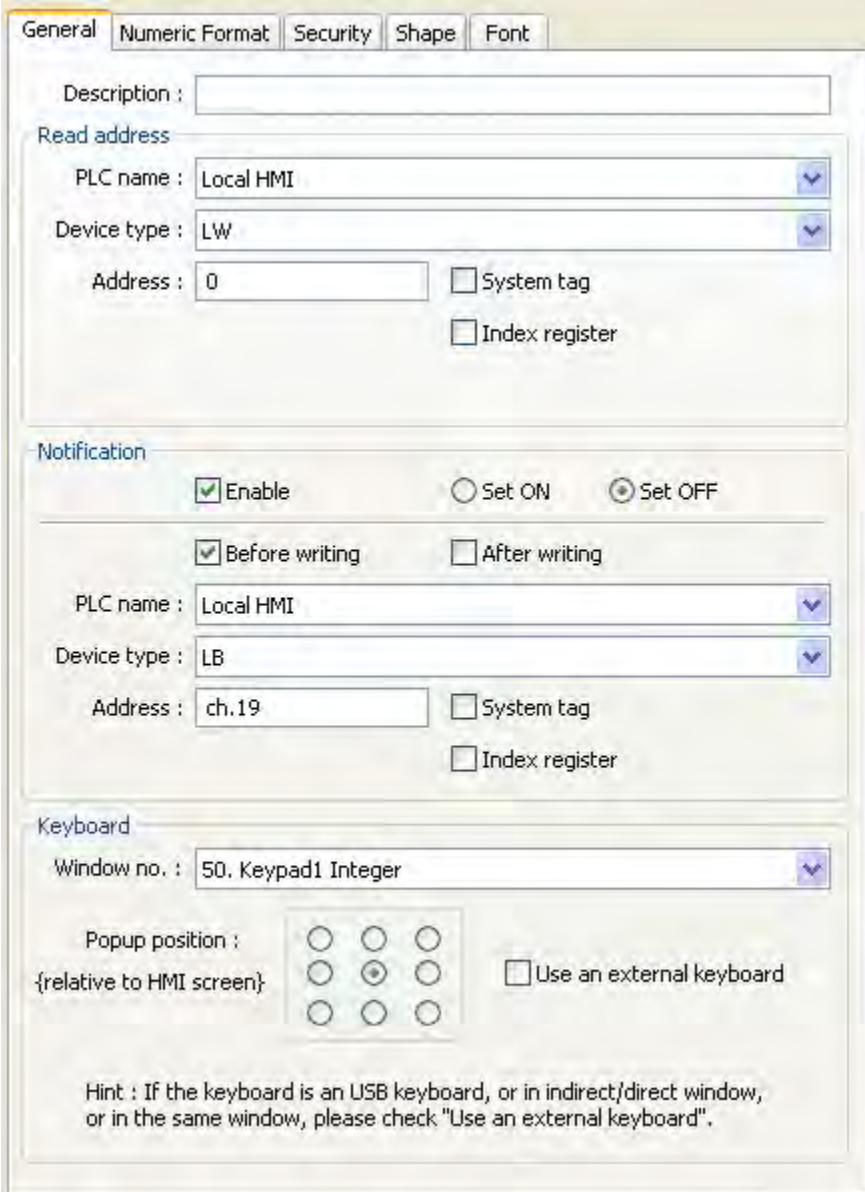
### ***The Numeric Input Object***

The Numeric Input Object is used to read and write numeric values to a PLC register. The object continuously reads the PLC register and displays the corresponding value in the format specified. The PLC register can be a 16-bit or 32-bit word. By using the keypad, the operator can enter a new value.



► To create a Numeric Input Object 

1. From the Objects menu, click **Numeric/ASCII-Numeric Input**. Or click the **Numeric Input** icon in the Part1 toolbar. The New Numeric Input Object dialog box appears.



The dialog box is titled "New Numeric Input Object" and has four tabs: General, Numeric Format, Security, Shape, and Font. The General tab is selected.

**Read address**

Description :

Read address

PLC name : Local HMI

Device type : LW

Address : 0  System tag  Index register

**Notification**

Enable  Set ON  Set OFF

Before writing  After writing

PLC name : Local HMI

Device type : LB

Address : ch.19  System tag  Index register

**Keyboard**

Window no. : 50. Keypad1 Integer

Popup position :

Use an external keyboard

Hint : If the keyboard is an USB keyboard, or in indirect/direct window, or in the same window, please check "Use an external keyboard".

2. Use the Description: box to enter a *title* for the Numeric Input part. A description is not necessary but does help you identify the purpose of the part.
3. In the **Read address** frame, select the *device type* from the drop down menu and then specify the *PLC register address* that the part continuously reads and writes to. Select the *data format*.
4. In the Keyboard frame, select *window that contains the keypad* you will be using for data entry from the drop-down menu, and then select the spot on the nine-dot matrix where you'd like the keypad to popup relative to the OIT display.

5. Click the **Numeric Format** tab..

6. In the **Display** attribute box, select the format type you wish to use. The options are:

Format	Range	Description
16-bit BCD	0 to 32,767 BCD	16-bit Binary-Coded Decimal
32-bit BCD	0 to 65,536 BCD	32-bit Binary-Coded Decimal
16-bit Hex	0 to FFFF	16-bit Hexadecimal
32-bit Hex	0 to FFFFFFFF	32-bit Hexadecimal
16-bit Binary	00... to 11... (16 bits)	16 Binary Digits
32-bit Binary	00... to 11... (32 bits)	32 Binary Digits
16-bit Unsigned	0 to 65,537	16-bit Unsigned Integer
16-bit Signed	-32,768 to +32,768	16-bit Signed Integer
32-bit Unsigned	0 to 4,294,967,295	32-bit Unsigned Integer
32-bit Signed	-2,147,483,647 to +2,147,483,647	32-bit Signed Integer
32-bit Float	-9999999999.9999999999 to 9999999999.9999999999	IEEE754 Single Precision

6. Check the *Mask* box to hide the value with asterisks.

7. In the Number of Digits section, select how the data is to be represented.

Format	Options	Range	Description
16-bit BCD	Left of Dec. Pt.	1 to 11	Digits to display to the left of the Decimal Point
	Right of Dec. Pt.	0 to 11	Digits to display to the right of the Decimal Point

32-bit BCD	Left of Dec. Pt.	1 to 11	Digits to display to the left of the Decimal Point
	Right of Dec. Pt.	0 to 11	Digits to display to the right of the Decimal Point
16-bit Hex	Left of Dec. Pt.	N/A	Displays 4 hex digits to the left of the Decimal Point
	Right of Dec. Pt.	N/A	No digits are displayed to the right of the Decimal Point
32-bit Hex	Left of Dec. Pt.	N/A	Displays 8 hex digits to the left of the Decimal Point
	Right of Dec. Pt.	N/A	No digits are displayed to the right of the Decimal Point
16-bit Binary	Left of Dec. Pt.	N/A	Displays 16 binary digits to the left of the Decimal Point
	Right of Dec. Pt.	N/A	No digits are displayed to the right of the Decimal Point
32-bit Binary	Left of Dec. Pt.	N/A	Displays 32 binary digits to the left of the Decimal Point
	Right of Dec. Pt.	N/A	No digits are displayed to the right of the Decimal Point
16-bit Unsigned	Left of Dec. Pt.	1 to 11	Digits to display to the right of the Decimal Point
	Right of Dec. Pt.	0 to 11	Digits to display to the left of the Decimal Point
16-bit Signed	Left of Dec. Pt.	1 to 11	Digits to display to the right of the Decimal Point
	Right of Dec. Pt.	0-11	Digits to display to the left of the Decimal Point
32-bit Unsigned	Left of Dec. Pt.	1 to 11	Digits to display to the right of the Decimal Point
	Right of Dec. Pt.	0-11	Digits to display to the left of the Decimal Point
32-bit Signed	Left of Dec. Pt.	1 to 11	Digits to display to the right of the Decimal Point
	Right of Dec. Pt.	0-11	Digits to display to the left of the Decimal Point
32-bit Float	Left of Dec. Pt.	1 to 11	Digits to display to the right of the Decimal Point
	Right of Dec. Pt.	0-11	Digits to display to the left of the Decimal Point

8. The Unsigned, Signed, and Float formats have the option of scaling the data. To display a scaled value, check the *Do Conversion* box. In the *Engineering Low* field, enter the *lowest value* that the OIT is to display. In the *Engineering High* field, enter the *highest value* that the OIT is to display. The conversion will also use the Low and High limits specified in the Limits section.
9. In the *Limits* section, first select if the Input Low and High values will be entered directly, or read from a series of PLC registers. If read from a register, the Low Limit is read from the specified register, and the High Limit is read from the next consecutive register. In either case, the valid ranges are listed below.

Format	Options	Range
16-bit BCD	Low Limit	0
	High Limit	65,535
32-bit BCD	Low Limit	0
	High Limit	4,294,967,295
16-bit Hex	Low Limit	0
	High Limit	65,535
32-bit Hex	Low Limit	0
	High Limit	4,294,967,295
16-bit Binary	Low Limit	0
	High Limit	65,535
32-bit Binary	Low Limit	0
	High Limit	4,294,967,295
16-bit Unsigned	Low Limit	0
	High Limit	65535
16-bit Signed	Low Limit	-32,768
	High Limit	32,768

32-bit Unsigned	Low Limit	0
	High Limit	4,294,967,295
32-bit Signed	Low Limit	-2,147,483,647
	High Limit	+2,147,483,647
32-bit Float	Low Limit	-9999999999.999999999
	High Limit	+9999999999.999999999

10. Click the Security tab to enable security for the numeric input object, if needed. Interlock frame, check the Enable box if the PLC is to enable or disable the object. The object will be enabled when the specified bit is set or clear, depending on whether the "Enable when Bit is ON" or "Enable when Bit is OFF" option is selected. To make the object invisible when disabled, check the Hide When Disabled checkbox. In the remaining options, select the PLC / Controller that controls the bit, and select the Device Type and the address. If security is enabled, you have the option of selecting up to level 6 of security and selecting to make the object invisible until the password has been entered.

**Interlock**

Use interlock function

Hide when disabled

Enable when Bit is ON       Enable when Bit is OFF

PLC name : Local HMI

Device type : LB

Address : 0       System tag

Index register

**User restriction**

Object class : None

**Sound**

Enable      Sound Library ...      Beep

Play

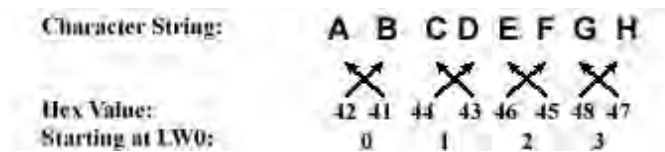
11. Click the Shape tab to configure a shape or bitmap that is displayed behind the number.
12. Click on the Font tab to select a font.
13. Select the *font* you want from the Font: drop down menu. Select the desired font size and color.
14. Enter the desired text positioning in the Align: box. The choices are:
- **Right Adjust** - aligns the number to the right margin of the part.

- **Left Adjust** - aligns the number to the left margin of the part.
  - **Leading Zero** - inserts leading zeroes
15. Click **OK**. The Numeric Input Object form closes and the main screen of EasyBuilder appears with the cursor tied to a rectangular outline of the part you just created at the upper left corner of the screen. Move the part to the desired location on the window.
  16. Once the part is placed onto the window, you can modify the attributes by double-clicking on the part to display the Numeric Input Object's Attribute dialog box. You may also resize the object by single clicking on the object and pulling the border of the object to the appropriate size. The object's associated text may be repositioned by clicking on the object, and then by dragging the text to its new position.

## The ASCII Display Object

The ASCII Display Object is used to display the character values of a series of PLC registers. The object continuously reads the PLC registers and displays the corresponding ASCII Data in the format specified. The ASCII Display Object can read up to 32 PLC registers to display a maximum of 32 characters.

Each 16 bit register stores two ASCII characters. The first ASCII character is stored in the eight least significant bits of the register. The 16-bit register that you select is the starting point for the ASCII character string. The example below illustrates how an eight character string is arranged:



### ► To create an ASCII Display Object

1. From the Objects menu, click **Numeric/ASCII-ASCII Display**. Or click the **ASCII Display** icon in the Part1 toolbar. The Create ASCII Display Object dialog box appears.

The screenshot shows the 'Create ASCII Display Object' dialog box with the 'General' tab selected. The 'Description' field is empty. Under the 'Read address' section, 'PLC name' is set to 'Local HMI', 'Device type' is 'LW', and 'Address' is '0'. There are checkboxes for 'System tag' and 'Index register', both of which are unchecked. The 'No. of words' is set to '1'.

2. Use the Description: box to enter a *title* for the ASCII Display part. A description is not necessary but does help you identify the purpose of the part.
3. In the Read address frame, select the *device type* from the drop down menu and then specify the *PLC register address* that the part continuously reads. Select *No. of words*: Remember, 1 word = 2 characters.
4. Click the **Font** tab to configure the font.
5. Click **OK**. The ASCII Data Object form closes and the main screen of EasyBuilder appears with the cursor tied to a rectangular outline of the part you just created at the upper left corner of the screen. Move the part to the desired location on the window.

- Once the part is placed onto the window, you can modify the attributes by double-clicking on the part to display the ASCII Data Object's Attribute dialog box.

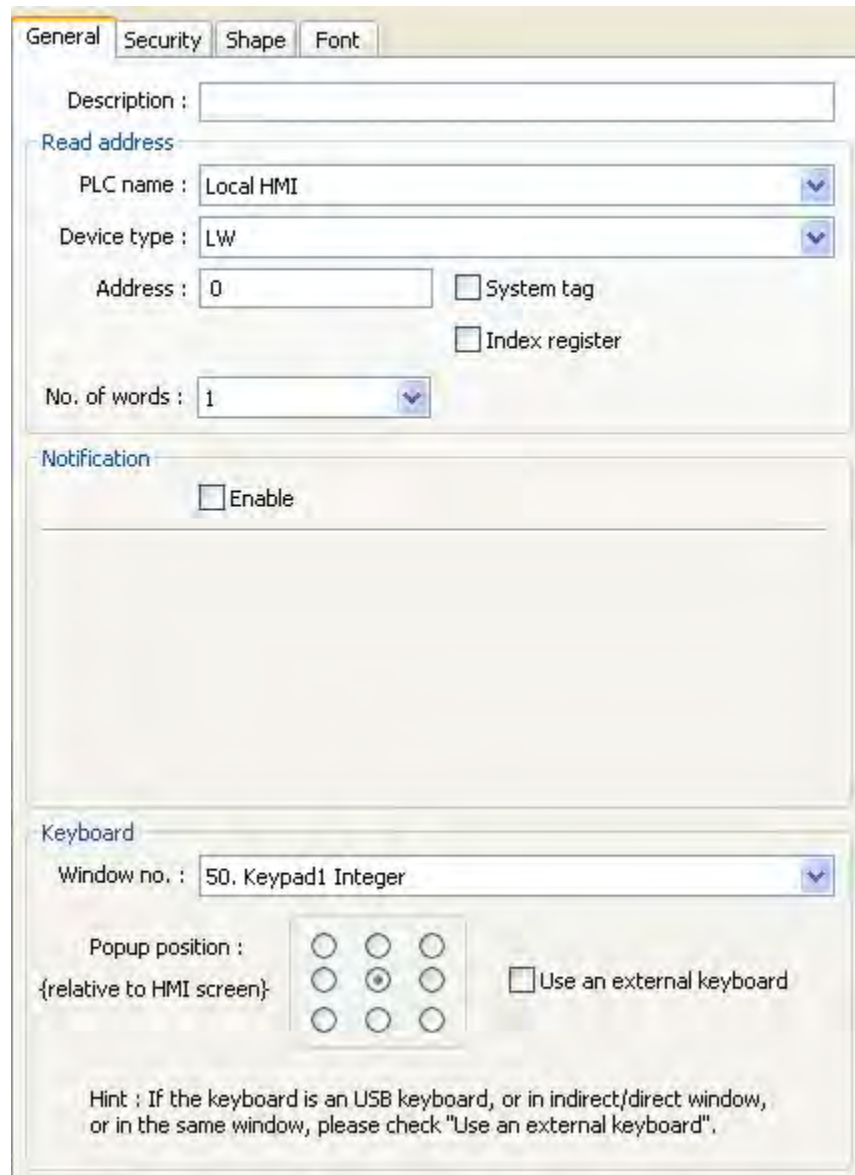
### The ASCII Input Object

The ASCII Input Object is used to enter characters into a series of PLC registers. The object continuously reads the PLC registers and displays the corresponding ASCII characters in the format specified. The ASCII Input Extend Object can write a maximum of 32 characters to 16 contiguous PLC registers.

See the ASCII Data section for information on how the ASCII characters are actually stored in the PLC registers.

#### ► To create a ASCII Input Object

- From the Objects menu, click **Numeric ASCII-ASCII Input..** Or click the **ASCII Input** icon in the Part1 toolbar. The New ASCII Input Object dialog box appears.



The dialog box is titled "New ASCII Input Object" and has four tabs: "General", "Security", "Shape", and "Font". The "General" tab is selected. The "Description" field is empty. The "Read address" section includes "PLC name" (Local HMI), "Device type" (LW), "Address" (0), and "No. of words" (1). There are checkboxes for "System tag" and "Index register", both of which are unchecked. The "Notification" section has an "Enable" checkbox, which is unchecked. The "Keyboard" section includes "Window no." (50. Keypad1 Integer), a "Popup position" grid (3x3 grid with the center circle selected), and a "Use an external keyboard" checkbox, which is unchecked. A hint is provided at the bottom: "Hint : If the keyboard is a USB keyboard, or in indirect/direct window, or in the same window, please check 'Use an external keyboard'".

- Use the **Description:** box to enter a title for the ASCII Input part. A description is not necessary but does help you identify the purpose of the part.

3. In the Read address frame, select the *device type* from the drop down menu and then specify the *PLC register address* that the part continuously reads. Select *No. of words*: Remember, 1 word = 2 characters.
4. In the Notification frame, check the **Enable** box if the OIT is to notify the PLC when the object has been activated. Configure whether the specified bit is to be turned on or off, whether the bit is to be written before or after the data is written, and finally specify the PLC Coil that the OIT is to write to.
5. In the Keyboard section, select your *keypad* from the drop-down menu, and then select the *popup position* relative to the OIT screen using the nine-dot matrix indicated (for instance, clicking the middle dot will pop the keypad up in the middle of the display).
6. Click the **Security** tab to display the security form.

The screenshot shows a configuration window titled "Interlock". It contains three main sections:

- Interlock:**
  - Checked:  Use interlock function
  - Checked:  Hide when disabled
  - Selected:  Enable when Bit is ON
  - Unselected:  Enable when Bit is OFF
  - PLC name: Local HMI (dropdown)
  - Device type: LB (dropdown)
  - Address: 0 (text field)
  - System tag:  (checkbox)
  - Index register:  (checkbox)
- User restriction:**
  - Object class: None (dropdown)
- Sound:**
  - Checked:  Enable
  - Buttons: Sound Library ... (button), Beep (checkbox)
  - Button: Play (button)

7. To enable security for the object, click the **Enable** checkbox and configure PLC settings from the drop-down menu.
8. Select the *level of user restriction* from the Object class drop-down menu.
9. You may select to make the object invisible while protected. You may also wish to enable a sound that plays when the protection keypad pops up. Sounds can be selected from sound libraries.
10. Click the **Shape** tab to configure a shape or bitmap that is displayed behind the number.
11. Click the **Font** tab to configure the font.

### The Moving Shape Object

The Moving Shape Object performs three functions depending upon the values in three consecutive PLC registers:

- Similar to the Word Lamp Object, the Moving Shape object continuously reads a PLC register and displays the corresponding shape or bitmap that is tagged to the number of states defined.

- the Moving Shape object continuously reads a second PLC register to determine where along an x-axis to display the shape or bitmap.
- the Moving Shape object continuously reads a third PLC register to determine where along a y-axis to display the shape or bitmap.

This feature allows you to create a shape or bitmap on a screen that can change the way it looks as well as where it is located on the OIT screen depending upon the control of the PLC. This can be used to simulate a control process such as some products moving on a conveyor system to different parts of the plant.

### ► To create a Moving Shape Object



1. From the Objects menu, click **Moving Shape**. Or click the **Moving Shape** icon in the Part2 toolbar. The Create Moving Shape Object dialog box appears.

2. Use the Description: box to enter a *title* for the Moving Shape part. A description is not necessary but does help you identify the purpose of the part.
3. In the **Read address** frame, select the *device type* from the drop down menu, and then specify the *starting PLC coil* address that the part reads from. Select the *data type*. The number of registers read depends upon the Style used.
4. In the **Attribute** frame, select how you want the Moving Shape object to operate:



Style	Option	Description
X axis only		Reads two 16-bit registers. Register #1 determines the state of the shape displayed. Register #2 determines the position of the shape along the X axis. The OIT places the shape relative to the starting location. For example, if the shape is located on the window at X=50, and Register #2 has a value of 10, the shape is placed at X=60. If Register #2 has a value of -10, the shape is placed at X=40.
	No. of states:	Range = 0 to 255
	Input low:	Not used
	Input high:	Not used
Y axis only		Reads two 16-bit registers. Register #1 determines the state of the shape displayed. Register #2 determines the position of the shape along the Y axis. The OIT places the shape relative to the starting location. For example, if the shape is located on the window at Y=50, and Register #2 has a value of 10, the shape is placed at Y=60. If Register #2 has a value of -10, the shape is placed at Y=40.
	No. of states:	Range = 0 to 255
	Input low:	Not used
	Input high:	Not used
X & Y axis		Reads three 16-bit registers. Register #1 determines the state of the shape displayed. Register #2 determines the position of the shape along the X axis. Register #3 determines the position of the shape along the Y axis. The OIT places the shape relative to the starting location. For example, if the shape is located on the window at X=50, Y=10, and Register #2 has a value of 10 and Register #3 has a value of 5, the shape is placed at X=60, Y=15. If Register #2= -10 and Register #3= -5, the shape is placed at X=40, Y=5.
	No. of states:	Range = 0 to 255
	Input low:	Not used
	Input high:	Not used
X axis w/scaling		Reads two 16-bit registers. Register #1 determines the state of the shape displayed. Register #2 determines the position of the shape along the X axis. The OIT places the shape relative to the starting location using scaling to calculate position. For example, if Input low=0, Input high=10, Scaling low=0, and Scaling high=100, then when the value in the PLC register increases by 1, the shape moves <i>right</i> along the X axis by 10 pixels.
	No. of states:	Range = 0 to 255
	Input low:	This is the lowest raw data value that is in the PLC data register
	Input high:	This is the highest raw data value that is in the PLC data register
	Scaling low:	This is the low scaled value.
	Scaling high:	This is the high scaled value.
Y axis w/scaling		Reads two 16-bit registers. Register #1 determines the state of the shape displayed. Register #2 determines the position of the shape along the Y axis. The OIT places the shape relative to the starting location using scaling to calculate position. For example, if Input low=0, Input high=10, Scaling low=0, and Scaling high=100, then when the value in the PLC register increases by 1, the shape moves <i>down</i> along the Y axis by 10 pixels.
	No. of states:	Range = 0 to 255
	Input low:	This is the lowest raw data value that is in the PLC data register
	Input high:	This is the highest raw data value that is in the PLC data register
	Scaling low:	This is the low scaled value.
	Scaling high:	This is the high scaled value.

X axis w/reverse scaling		Reads two 16-bit registers. Register #1 determines the state of the shape displayed. Register #2 determines the position of the shape along the X axis. The OIT places the shape relative to the starting location using scaling to calculate position. For example, if Input low=0, Input high=10, Scaling low=0, and Scaling high=100, then when the value in the PLC register increases by 1, the shape moves <i>left</i> along the X axis by 10 pixels.
	No. of states:	Range = 0 to 255
	Input low:	This is the lowest raw data value that is in the PLC data register
	Input high:	This is the highest raw data value that is in the PLC data register
	Scaling low:	This is the low scaled value.
	Scaling high:	This is the high scaled value.
Y axis w/reverse scaling		Reads two 16-bit registers. Register #1 determines the state of the shape displayed. Register #2 determines the position of the shape along the Y axis. The OIT places the shape relative to the starting location using scaling to calculate position. For example, if Input low=0, Input high=10, Scaling low=0, and Scaling high=100, then when the value in the PLC register increases by 1, the shape moves <i>up</i> along the Y axis by 10 pixels.
	No. of states:	Range = 0 to 255
	Input low:	This is the lowest raw data value that is in the PLC data register
	Input high:	This is the highest raw data value that is in the PLC data register
	Scaling low:	This is the low scaled value.
	Scaling high:	This is the high scaled value.

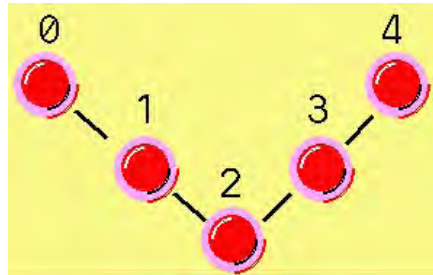
5. Refer to the Bit Lamp Object section for information on using the Shape tab.
6. Refer to the Bit Lamp Object section for information on using the Label tab.
7. Click **OK**. The Create Moving Shape Object form closes and the main screen of EasyBuilder appears with the cursor tied to a rectangular outline of the part you just created at the upper left corner of the screen. Move the part to the desired location on the window.
8. Once the part is placed onto the window, you can adjust the location of the label inside the part by clicking once on the label. This will highlight the entire object. Click on the label again. Now only the label is highlighted, allowing you to move it without moving the part. Note: if you double-click (click twice rapidly) then you will not highlight the label but rather enter the Moving Shape Object's Attribute dialog box.

## The Animation Object

The Animation Object is very similar to the Moving Shape Object. It also allows you to create a shape or bitmap on a screen that can change the way it looks as well as where it is located on the OIT screen depending upon the control of the PLC. However, the Animation Object moves along a predefined path that is determined when you create the part. Because of this, the Animation Object requires only two registers for changing state and X/Y movement:

- the Animation object continuously reads a PLC register and displays the corresponding shape or bitmap that is tagged to the number of states defined.
- the Animation object continuously reads a second PLC register to determine which position along the predefined path it should be.

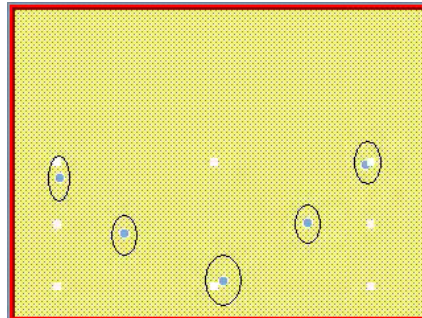
As shown in the following example, the part moves along a predefined track when the value in the second PLC register is incremented:



### To create an Animation Object



1. From the Objects menu, click **Animation**. Or click the **Animation** icon in the Part2 toolbar. The mouse cursor changes from a pointer to a circle when placed over the work area of EasyBuilder. Use the cursor to set the path that the Animation object is to take. At each location along the path, click the mouse to place another position marker. A maximum of 255 position markers may be placed onto the window screen.
2. When you have finished placing the position markers, right click the mouse to display the part at the first position.



- Double click the part. The Animation Object's Properties dialog box appears:.



- Use the Description: box to enter a *title* for the Animation part. A description is not necessary but does help you identify the purpose of the part.
- In the Read address frame, select the *device type* from the drop down menu and then indicate the *starting PLC coil* address that the part reads from. Select the *data format*.
- In the **No. of states:** box, select the total *number of states* to be displayed.
- Refer to the Bit Lamp Object section for information on using the **Shape** tab.
- Refer to the Bit Lamp Object section for information on using the **Label** tab.
- Click on the **Profile** tab. The Profile dialog box appears.



- The Position settings refer to the overall location of the Animation Object on the window screen.
- The Size settings refer to the overall size of the Animation Object.
- The Shape rectangle size settings refer to the size of the shape or picture that you are using.
- The Trajectory setting allows you to manipulate the position of each point along the path of movement. Click on the points pull-down box to select the point you wish to change, then edit the X and Y positions.
- Click **OK**. The Animation Object's Properties dialog form closes and the main screen of EasyBuilder appears with the Animation part you just created on the screen. The shape you selected is located at the first position on the path.
- Once the part is placed onto the window, you can adjust the attributes by double-clicking on the part to enter the Animation Object's Properties dialog box.

## The XY Plot Object

An XY Plot Object plots one series of values against another series of values.

- To create an XY Plot Object:



1. Select **Objects - XY Plot** from the EZBuilder menu or select the XY plot icon from the parts toolbar. The New XY Plot Object dialog appears.

2. Select the General tab.
  - Description: Enter the text description or name for the object.
  - Direction: Specify the *direction of the plot*:
    - Right - the origin is at the lower left. The X-axis moves to the right, the Y-axis moves up.
    - Left - the origin is at the lower right. The X-axis moves to the left, the Y-axis moves up.
    - Up - the origin is at the lower left. The X-axis moves up, the Y-axis moves to the right.
    - Down - the origin is at the upper left. The X-axis moves down, the Y-axis moves to the right.
  - No of channels: Specify the *number of datasets to display*.
  - In the Control Address section, set the **PLC name** of the PLC to read/write. Set the **Device Type** and the **Control Address**. The Control Address field determines how the XY Plot operates:

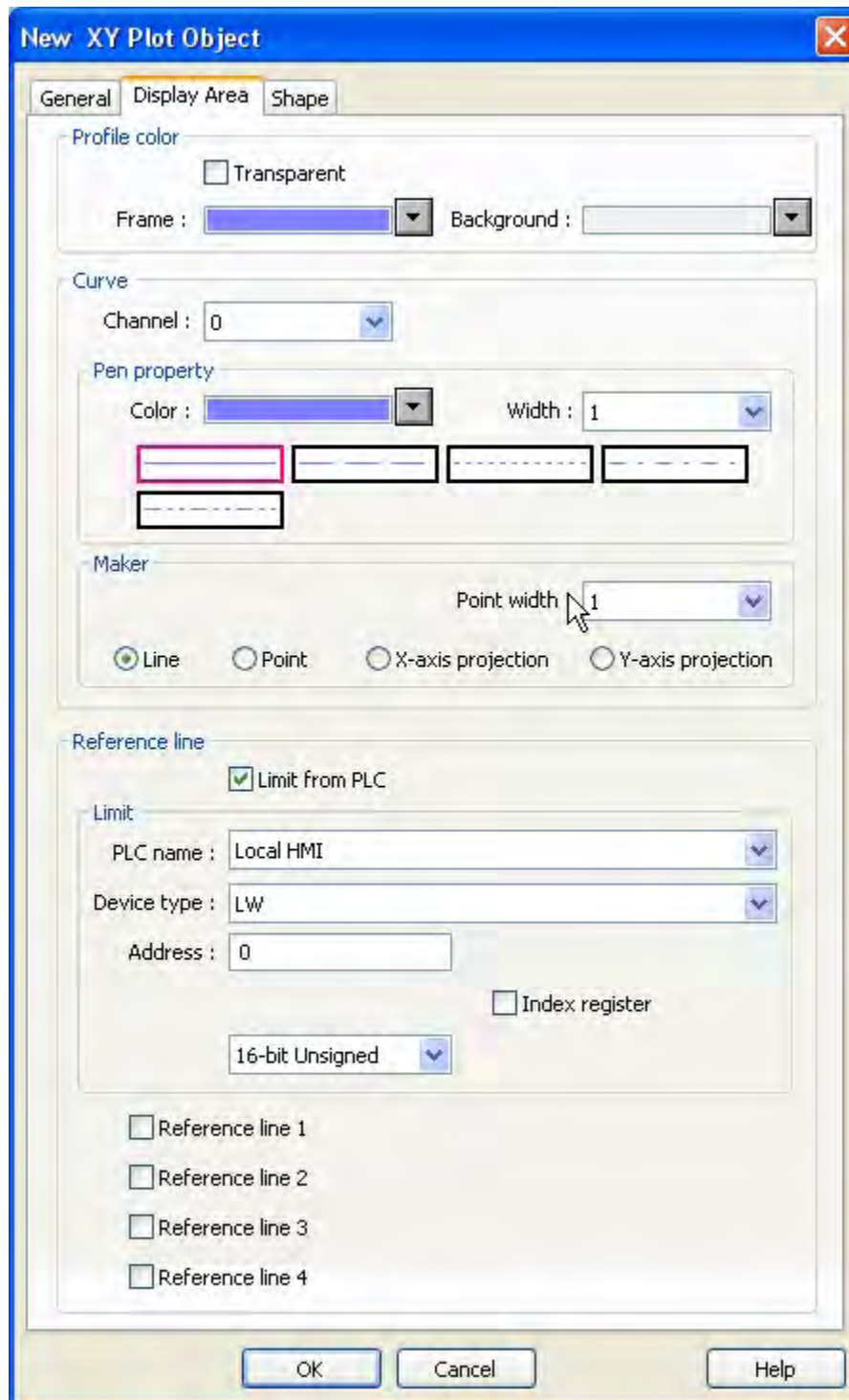
---

Trigger Value	Effect
1	The current data is plotted, any existing plot data remains.
2	All plotted data is cleared.
3	The plot is cleared, then the current data is plotted.

Check the **User defined tag checkbox** if you wish to fill the Device list with Tag Names instead of using the PLC's Device types. Set **No. of Data Address** to indicate the address (offset from the Control Address) that contains the number of data points to plot .

- Select the **channel** to configure.
- In the Read Address section, select the **PLC Name**. Check the **Separate Addresses for X and Y Data** checkbox if you wish the XY Plot to use separate sets of addresses for X and Y data. If unchecked, all data is read from the same set of addresses. The current Read Address settings are displayed in the Read Address box. Click **Settings...** to reconfigure.
- In the Limits section, check the **Dynamic Limits** checkbox if you wish the high and low limits to be read from the controller register. Otherwise, specify the limits by completing the *X-axis high limit, X-axis low limit, Y-axis high limit and Y-axis low limit*.

3. Select the Display Area tab:



- In the Profile Color section, check the **Transparent** checkbox if you wish the frame and background of the XY plot to be invisible. Otherwise, select the *frame* and *background* colors.
- In the Curve section, select the *channel number* to configure and then set *pen properties* and how the plot will display in the *Maker* section.
- In the Reference line section, check the **Limit from PLC** checkbox if checked, the low and high limits are read from controller register. Then set *PLC name*, *Device type* and *Address*.

- If you'd like reference lines on the plot, check the **Reference line** checkboxes and specify the position.
4. Select the Shape tab:



- In the Shape section, , check the **Use shape** checkbox and then click the **Shape library...** button to select a background shape from the shape library. Check the **Inner** checkbox to use the interior portion of the selected shape and select the color. Check the **Frame** checkbox to use the frame of the shape and select the color.
- In the Picture section, select the **Use picture** checkbox and then click the **Picture library...** button to select a background picture. Click the **Set to original dimensions** checkbox to always set the picture to its original size.
- Use the preview section to preview a part's appearance in its selected state.

5. Click **OK** to save settings.



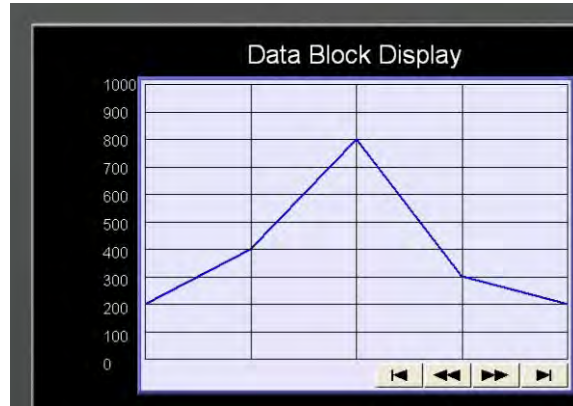
**Please Use This Page For Your Notes:**

**Please Use This Page For Your Notes:**

## Data Block Display

The Data Block Display reads data points read from consecutive PLC registers based upon a triggered event then plots them as a line graph. It is very similar to the Trend Graph object, except that data points are plotted based upon a triggering device rather than time-based.

Each line represents a "channel" or collection of data points read simultaneously from consecutive PLC registers. Below is an example of a simple data block display with a single channel:



In this example, data is captured from five consecutive PLC registers (LW3-7):

LW3(200)

LW4(400)

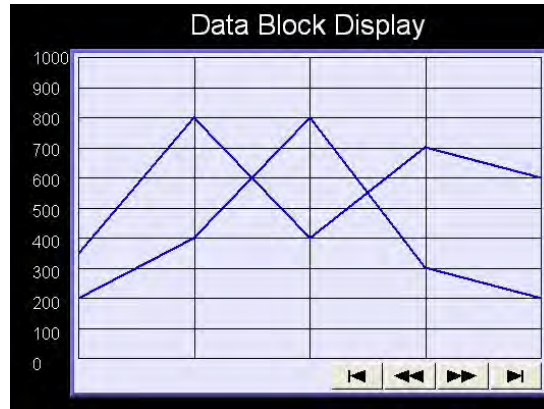
LW5(800)

LW6 (300)

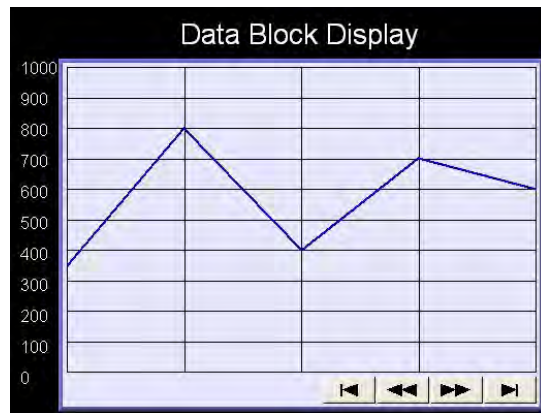
LW7(200)

This data was read from the PLC immediately after a trigger called the 'Control Word Address' is set to 1 (in this example, we used LW0). The data points are plotted evenly across the X axis according to predefined settings.

*NOTE: If data is captured again, it can be overlaid on the prior reading or you can erase the current plot and replace it with a new graph:*

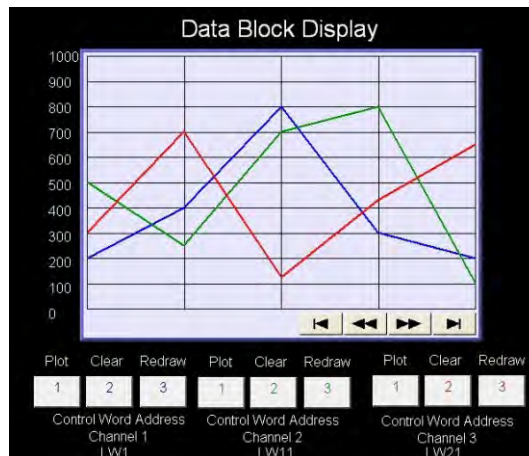


In the above graph the Control Word Address is set to 1 to overlay the plots.



In the above graph the Control Word Address is set to 3 to erase existing plots and trigger a new plot.

A maximum of twelve channels can be displayed. In this example, three channels are displayed:



Each channel is controlled by a unique Control Word Address.

### To create a Data Block Display Object:

1. From the Objects menu, click **Data Block Display**. Or click the Data Block Display icon in the Object2 toolbar. The New Data Block Display dialog box appears:

2. **Description:** see section “General Properties” at the beginning of this chapter.

3. **No. of channel:** Enter the total number of channels (collection of data points depicted as a line graph) that are displayed. Range is 1 to 12.

4. **Cursor line frame:** this option is used to display a vertical ‘watch’ line on the line graph. During operation, when the user touches the Data Block display object, a cursor line appears. The OIT then reads all of the data points along this vertical line and writes the values to predefined registers. Use the Numeric Display object to display these registers.

- a. Enable: check to enable the cursor ‘watch’ line feature.
- b. Color: select the color required for the vertical cursor line
- c. PLC Name, etc: use to select the target PLC register. For more information on each field, see “General Properties” section at beginning of this chapter.

5. **Channel:** select to define the Control Address parameters for each channel.

6. **Control Address frame:** use to select the target PLC register.

a. **PLC Name:** see “General Properties” section at beginning of this chapter.

b. **Device type:** same as above

c. **Control word address:** this is the specific PLC address from the Device Type memory area. The Control Word Address is the ‘trigger’ that is used during operation to capture the data and plot a line graph. To do this, the operator writes the following command values to this register:

0 = No action (default) occurs.

*Note: the OIT will automatically write the value of 0 after performing the specified command*

1 = Plot the data points Note: use this command to overlap consecutive plots

2 = Clear the graph

3 = Clear the graph and plot new points

d. **No. of data address:** This is the register that the OIT uses to determine how many consecutive data points to capture. By default, the address is Control word address+1. The maximum number of data points is 1024.

e. **Data storage start:** This is the starting register address that is read by the OIT to capture the data points that are displayed as a line graph. By default, it is the Control Word Address+2. The total number of addresses read is determined by ‘No. of data address’ setting above. If the ‘Offset to start address’ is checked, this register is read by the OIT as an offset. The value is added to the Control Word Address to determine the first register to be read for data points.

f. **Offset to start address:** enable this to change the Data Storage Start address to an Offset.

g. **Index Register:** see “General Properties” section at beginning of this chapter.

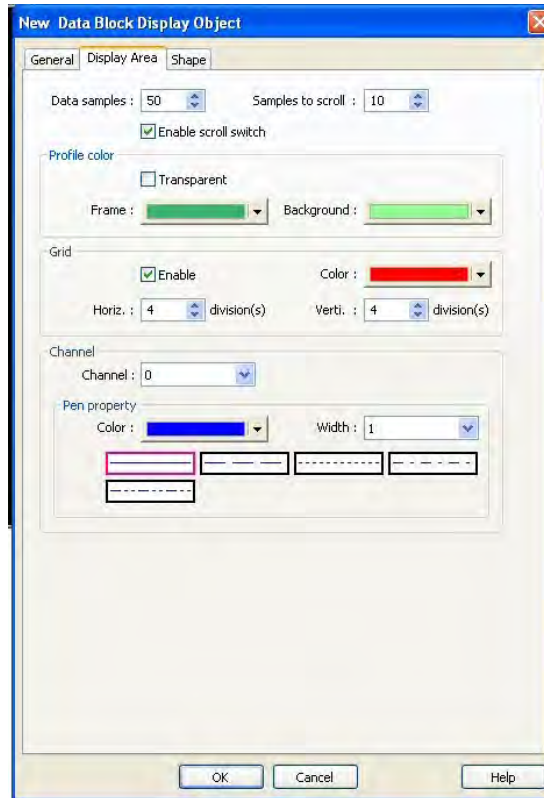
h. **Format:** Determines how the OIT will interpret the data values.Limit:

i. **Limit:**

i. Min: This is the minimum range of the line graph.

ii. Max: This is the maximum range to the line graph

**Display Area tab: used to configure how the data appears and sampling rate:**



1. **Data samples:** Determines how many data points or samples are displayed on a single screen. Note: if this number is smaller than the actual number of data points collected, the scroll button feature should be enabled.

2. **Samples to scroll:** Determines how much the line graph moves **forward** or **backward** when using the scrolling buttons. Units are based upon number of data points.

3. **Enable scroll switch:** displays scroll buttons on the line graph if enabled.

4. **Profile color frame:** this determines the colors used for the background of the line graph

a. **Transparent:** If enabled, only the channels are displayed. This feature is useful when overlapping on some other objects.

b. **Frame:** Determines the color of a thin line that encircles or frames the line graph. Note: if a shape is used, it will overwrite this setting.

c. **Background:** Determines the background color.

5. **Grid frame:** Used to display a grid

a. **Enable:** Check to enable a grid

b. **Color:** Select the color for the grid lines

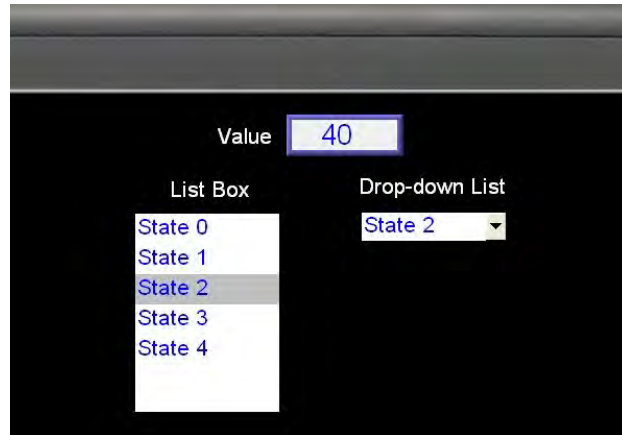
c. **Horiz:** the number of divisions that divide the X-axis.

- d. **Verti:** the number of divisions that divide the Y-axis.
6. **Channel frame:** these settings are used to configure the color of each line.
- a. **Channel:** select the particular channel that you wish to change color properties.
  - b. **Pen property frame:**
    - i. **Color:** Select the particular color for the channel
    - ii. **Width:** Select the width of the line. Note: if a value other than 1 is selected for the width, the style option is not available.
    - iii. **Style:** determines the style of the line displayed when width is set to 1.
7. **Shape tab:** used to configure the text labels and constant values assigned to each State.



## Option List

The Option List Object is used to write predefined values to a target PLC register based upon which state is selected in a list. Up to 255 states are supported. The Option List Object can be displayed as either a simple list box or a drop-down menu. Unique labels can be created for each state.



### To create an Option List Object

1. From the Objects menu, click **Button-Option List**.

Or click the **Option List icon** in the Object 1 toolbar. The New Option List Object dialog box appears:

The screenshot shows the 'New Option List Object' dialog box with the following settings:

- Description:** [Empty text box]
- Attribute:**
  - Mode: List box
  - Background: [Color selection box]
  - No. of state: 5
  - Selection: [Color selection box]
- Control address:**
  - PLC name: Local HMI
  - Device type: LW
  - Address: 0
  - System tag
  - Address format: dddd [range: 0 ~ 10255]
  - Index register
  - 16-bit Signed
  - Write when button is released

2. **Description:** see section “General Properties” at the beginning of this chapter.

3. **Attribute frame:** this is used to determine which type of list box is used, color, and total number of states.

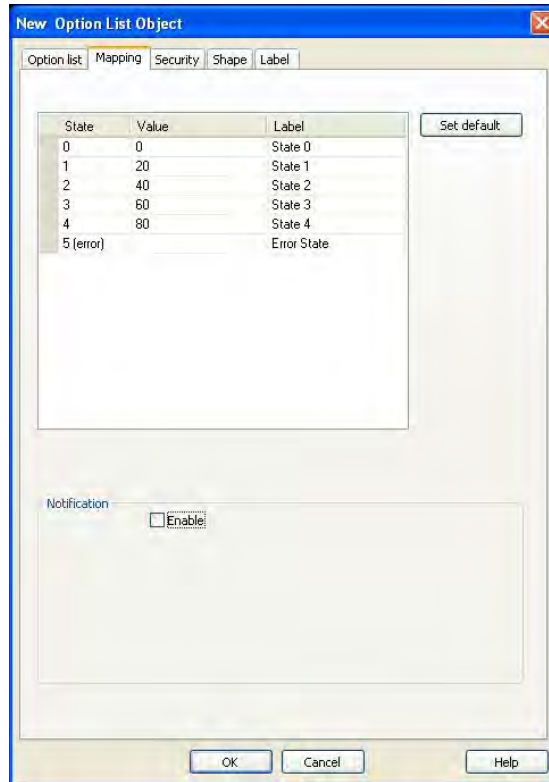
- Mode:** options are List box and Drop-down List. List box is a simple list in which all of the states are shown. Drop-down List is a menu that opens when you click on it.
- No. of state:** range is 1 to 255. Note: that if the range is larger than the size of the object, a scroll down box is used.
- Background:** color of the background of the list box.
- Selection:** color of the ‘selected’ or active state.

4. **Control address frame:** use to select the target PLC register. For more information on each field, see General Properties section at the beginning of this chapter.

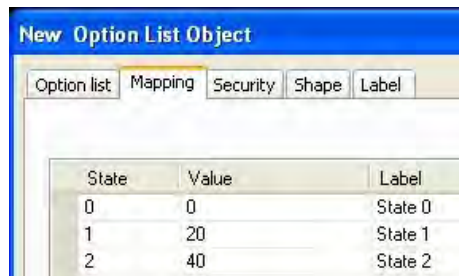
- Write when button is released: if unchecked, the predefined constant value is written to the target PLC register immediately after the state has been selected. If checked, the value is not written until the operator removes his finger from the touchscreen.

**Mapping tab:** used to configure the text labels and constant values assigned to each State.

1. Mapping table:



The number of states shown depends upon the number of states selected in the Attribute frame. In this example, there are a total of five states. Click on the entry in the Value column for each State to enter a new constant value:



*Note: You cannot enter a value for the last (error) state. The error state is always represented by any value that is not represented by one of the other states. Note: Negative values are not supported. Floating point values are supported.*

*Also note that the values entered for each state are unique- not related to other values entered for other states. For example, entering a value of 14 for State 2 in the example above is acceptable.*

**Notification frame:** this optional feature can be used to set (or clear) a target PLC bit when the Multi-State Switch object reads a value in the PLC register that is invalid. Note: the PLC must clear this bit after notification from the OIT.

Set ON or Set OFF- select which action the HMI takes when an error occurs.

**Security frame:** see section “General Properties” at beginning of this chapter.

**Shape frame:** Enabling a shape will overwrite any settings for the background color under the Option list tab.

**Label frame:** Determines the attributes of the text entered for each state on the Mapping tab.

**Profile:** see section “General Properties” at beginning of this chapter.

## Chapter 11 - Using and Creating Keypads

The OIT operator must have a keypad available to enter new data when using the Numeric Input Object or the ASCII Input Object. EasyBuilder includes group libraries that contain sample keypads to be placed onto a window screen. You can also create your own custom keypads. This chapter focuses on how to create keypads, display them on the window screen, and use them for entering data into PLC registers.

### How to Create a Keypad

Any keypad that you create is actually composed of keys using the Function Key Object. The Function Key Object can create alphanumeric keys as well as control keys, (i.e. Delete, Enter, Clear, etc.)

► To create a Function Key 

1. From the Objects menu, click **Button-Function Key**. Or click the **Function Key** icon in the Part1 toolbar. The New Function Key Object dialog box appears.

2. Use the Description: box to enter a title for the Function Key part. A description is not necessary but does help you identify the purpose of the part.
3. In the ASCII mode section, there are four control key options. Control keys are used during data entry to perform a specific function.

Function	Description
[ENTER]	Configures the Function Key as an Enter key. When pressed, it will write the alphanumeric characters entered into a Numeric Input Object or ASCII Input Object to the target PLC register.

[BACK SPACE]	Configures the Function Key as a Delete key. When pressed, it will delete the last alphanumeric character entered.
[CLEAR]	Configures the Function Key as a Clear key. When pressed, it will clear the data displayed in the Numeric Input Object or ASCII Input Object. Note: this key does not clear the actual target PLC register until the Enter key is pressed.
[ESC]	Configures the Function Key as an Escape key. When pressed, it will exit the editing mode.

4. Just below the four control key options, is the **ASCII** checkbox. Select this when you want to configure the Function Key to enter an alphanumeric character. When this checkbox is selected, the **Character:** box is activated. Use this box to enter the ASCII character you want to use.
5. Click the **Shape** tab to display the Shape form.
6. Select either a shape or bitmap to represent the Function Key object. If you need more information on how to do this, consult Chapter 6 “Creating Graphics Objects”.
7. Click the **Label** tab to display the Label form.
8. Click the **Use Label** checkbox to use a label.
9. You can have a different label for each state of the Bit Lamp. In the **State:** box, enter the state that you wish to edit, then enter the label for that state in the **Content:** box. Do the same for the other state.
10. Enter the color that you want for the label in the **Color:** box. You can select a different color for each label state.
11. Enter how you want the label to be positioned in the **Align:** box. You can select a different position for each label state.
12. Finally, select the font style and size from their respective drop down menus.. You can select a different font for each label state.
13. Click the **Tracking** checkbox if you want the labels for all states to follow or ‘track’ with the movement of one label when it is moved after the Function Key object is placed onto the window. Click **Duplicate this label to other states** if you want to use the same text on all states.
14. Click the **Security** tab if you would like to enable security for the function key and configure security settings as described in previous sections.
15. Click **OK**. The New Function Key Object form closes and the main screen of EasyBuilder appears with the cursor tied to a rectangular outline of the part you just created at the upper left corner of the screen. Move the part to the desired location on the window.
16. Once the part is placed onto the window, you can adjust the location of the label inside the part by clicking once on the label. This will highlight the entire object. Now click on the label again. Now only the label is highlighted, allowing you to move it without moving the part. Note: if you double-click (click twice rapidly) then you will not highlight the label but rather enter the Function Key Object’s Attribute dialog box.

To create a custom keypad, you create several of these Function Key Objects and group them together on a window to form the keypad. Following the procedure outlined in Chapter 6, you can then save the keypad to a group library for later use in other projects. Several predefined keypads are included with EasyBuilder. To use these keypads, use the **Call Group Library** function and access the **keypad** library.

## Displaying and Using a Keypad

Let’s go through a sample project that uses one of the predefined keypads included in EasyBuilder to edit a Numeric Input Object and an ASCII Input Object. This example will show you how to place these objects on a window screen, and then use the keypad to edit the data registers.

### ► To create a sample keypad entry screen

1. Start a new project file, with the initial screen configured as the startup screen. Note: this is the default setting. Display the initial screen in the work area of EasyBuilder.
2. From the Objects menu, select **Numeric Input**. The New Numeric Object dialog box appears.
3. Create a Numeric Input Object with the following parameters:

Tab	Frame	Parameter	Value
General	Description		Numeric Input Object
	Read Address	PLC Name:	Local HMI
		Device Type:	LW
		Address:	0
	Keyboard	Window No	54:ASCII Middle
Popup position		Select middle option in the right-hand row	
Numeric Format	Display	Data Format	16-bit signed
	Number of Digits	Left of Decimal Pt	5



For more information on how to use the Numeric Input Object, consult Chapter 8 “Representing Data with Graphics Objects”.

4. Create an ASCII Input Object with the following parameters:

Tab	Frame	Parameter	Value
General	Description		ASCII Input Object
	Read Address	PLC Name:	Local HMI
		Device Type	LW
		Address:	1
		No. of words	12
	Keyboard	Window No.	54. ASCII Middle
Popup position		Select middle option in the right-hand row	

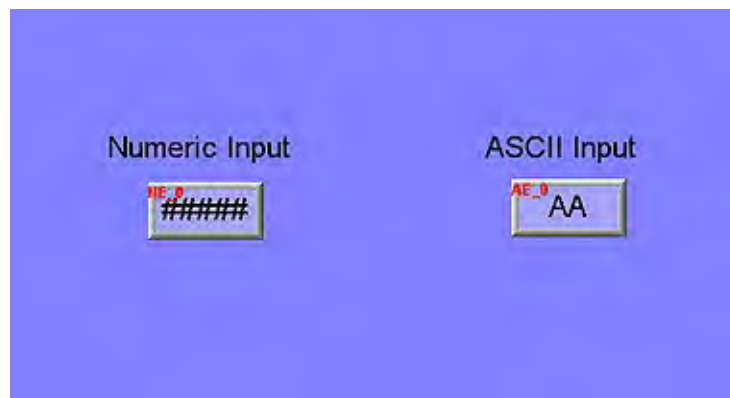


For more information on how to use the ASCII Input Extend Object, consult Chapter 8 “Representing Data with Graphics Objects”.

5. Create two text boxes; one to identify the Numeric Input register, and one to label the ASCII Input register.

#### ► Editing data registers using a keypad

1. Save and compile the new project, then test the operation of the keypad by either running in Simulation mode or by downloading the project to the OIT.
2. The following screen will appear:



3. **Change the value of the Numeric Input register.** Touch the Numeric Input Object to activate the edit mode. The keypad will pop up and the 0 digit will start to blink.



4. Using the keypad, enter digits 1, 2, 3, 4, and 5. As you enter the digits, each one should appear in the Numeric Input Object box.
5. Press the **Backspace** key to delete digit **5**.
6. Try pressing any of the alpha characters (A-Z). Notice that alpha characters are not accepted into a Numeric Input Object register.
7. Press the **Clear** key. Notice that the entire register is cleared to **0**.
8. Enter digits 5, 4, 3, 2, 1, and then press the **Enter** key. The keypad closes but the value was not accepted. This indicates that the value entered is out-of-range. Touch the numeric input object again. Press the **Clear** key and enter 32767. Press the **Enter** key. Notice that this value is accepted and blinking stops.
9. Touch the Numeric Input Object register again to activate editing mode. Enter five more digits then press the **Esc** key. Notice that the entry is cancelled and the previous number 32767 is restored.
10. **Change the value of the ASCII Input register.** Touch the ASCII Input Object to activate the edit mode.
11. Using the keypad, enter the character string "HELLO WORLD". As you enter the characters, each one should appear in the ASCII Input Object box. Press the **Enter** key to send the characters to the OIT internal data register.

**Please Use This Page For Your Notes:**

## Chapter 12 - Bar Graphs, Meters and Trends

This chapter focuses on three special graphic objects, which can be used to display PLC data registers. You read in Chapter 6 how to use shapes and bitmaps to represent the data in PLC registers as states. You also read how to use alphanumeric data fields to display the contents of PLC registers as either numbers or ASCII characters. We now introduce three more options to display the data in PLC registers:

Bar Graph Objects



Trend Display Objects



Meter Display Objects



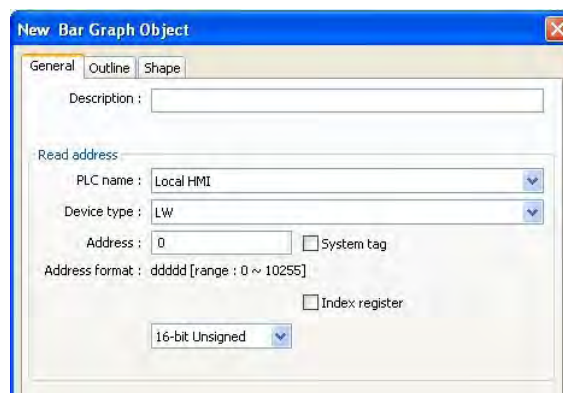
### Creating Bar Graphs

The Bar Graph Object is used to represent the data in a 16-bit or 32-bit PLC register as a bar graph. You can configure the bar graph to move up, down, right, or left. The bar graph can be configured with any base number that represents 0 level and any span range. You can create the bar graph with an alarm low and high setting to indicate to the OIT operator that an underflow/overflow alarm condition exists. In addition, you can set the alarm low and high limits to be controlled by two additional PLC registers so that the low and high limits are variable. Bar graphs can even be constructed with a shape or bitmap overlaying the bar graph to create flow tanks, temperature gauges, etc.

#### ► To create a Bar Graph Object



1. From the Objects menu, click **Bar Graph**. Or click the **Bar Graph** icon in the Part2 toolbar. The New Bar Graph Object dialog box appears.



2. Use the Description: box to enter a *title* for the Bar Graph part. A description is not necessary but does help you identify the purpose of the part.
3. In the Read address frame, select the *device type* from the drop down menu, and then indicate the *PLC register* address. Select the data format.

4. Click on the **Outline** tab to display the outline form.

5. In the attribute section, select how the Bar Graph is to operate.

Function	Option	Description
Type	Normal	The bar starts at one end of the object, and moves to the other end.
	Offset	The bar starts at some non-zero point defined by the <i>Origin</i> , and moves through the origin as the value changes.
Zero		The PLC Register values corresponding to an unfilled bargraph.
Origin		The starting point of the bar when the <i>Type</i> is set to <i>Offset</i> .
Direction	Up	The bargraph starts at the bottom of the object, and moves up as it fills.
	Down	The bargraph starts at the top of the object, and moves down as it fills.
	Right	The bargraph starts at the left of the object, and moves right as it fills.
	Left	The bargraph starts at the right of the object, and moves left as it fills.
Span		The PLC Register values corresponding to a completely filled bargraph.
Bar Width Ratio (%)		The percent of the object's width that the bar occupies.

## 6. The Bar Style/Color determines the appearance of the Bar Graph

Function	Description
Frame	Set the color of the outside of the bar graph.
Bar	Set the color of the filled part of the bar.
Background	Set the color of the unfilled part of the bar.
Bar Style	Set the pattern of the filled part of the bar.

## 7. Set up the Target Indicator. This selects the color of the filled part of the bar when it reaches a preset value.

Function	Description
Enable	When checked, the Target Indicator is active.
Color	Set the color to change the bar when the value is equal to the Target Value.
Target Value	The value at which to change the color of the bar.
Tolerance	Set a tolerance for the Target Value. When the value is less than or greater than the Target Value by this value, the bar will remain in the selected color.

## 8. Select the Alarm Colors.

Function	Description
Low Limit	The value at which the bar will change to the Low Color.
Low Color	The color to change the bar when the value reaches the Low Limit.
High Limit	The value at which the bar will change to the High Color.
High Color	The color to change the bar when the value reaches the High Limit.

## 9. Set up Dynamic Limits. When checked, the Target/Alarm Dynamic Address option allows certain alarm limits to be read from the PLC / Controller. The Target Value, and Low and High Limit fields are disabled. Data is read from the PLC as follows:

Specified Address	Low Limit
Specified Address + 1	High Limit
Specified Address + 2	Target Value

## 10. Once the part is placed onto the window, you can adjust the attributes of the Bar Graph by double-clicking on the part.

## Creating Meter Displays

The Meter Display Object is used to represent the data in a 16-bit or 32-bit PLC register as a scaled meter. The meter can be easily customized to the look that you want.

The Display Meter can be configured with any base number that represents 0 level and any span range.

### ► To create a Meter Display Object



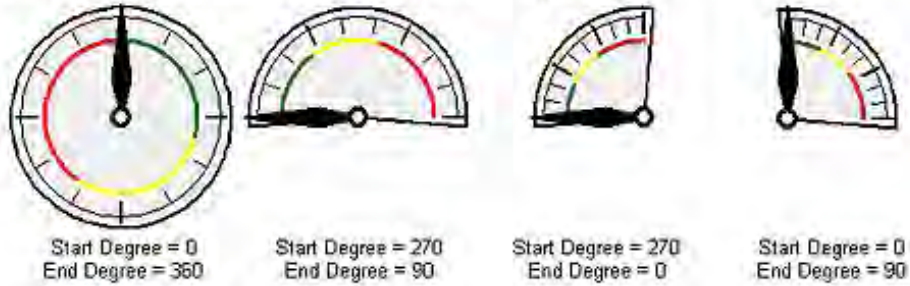
1. From the Objects menu, click **Meter Display**. Or click the **Meter Display** icon in the Part2 toolbar. The New Display Meter Object dialog box appears.

2. Use the Description: box to enter a title for the Meter Display part. A description is not necessary but does help you identify the purpose of the part.
3. In the Read address frame, select the *Device type* from the drop down menu, and then specify the *PLC register address*. Select the data format.
4. Click the **Outline** tab to display the Outline form. This is where you will configure all of the attributes of your meter.

- In the Degree section, you can select the range of your meter, where the range is 0 to 360. The start degree, 0 is at 12:00. Below are four examples of start and end degrees for different meters:

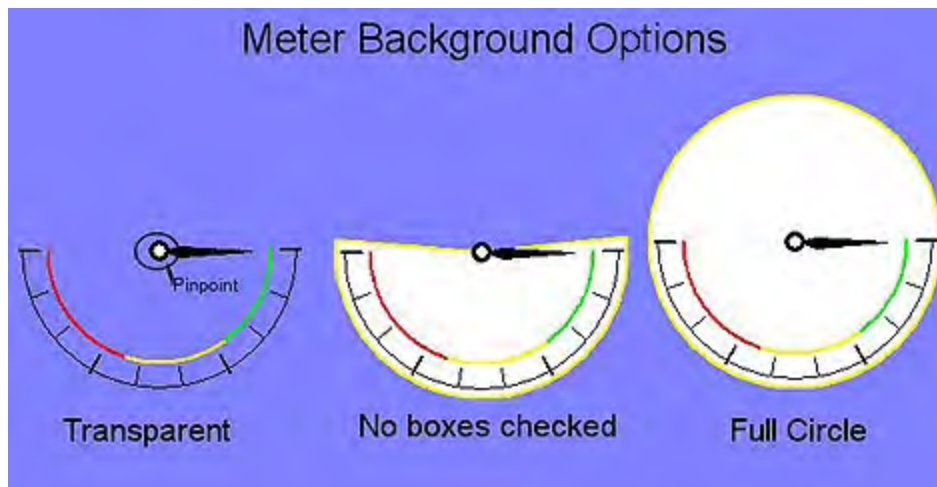
**Start Degree**

The position in a circle where the meter display starts. Range is 0-360.



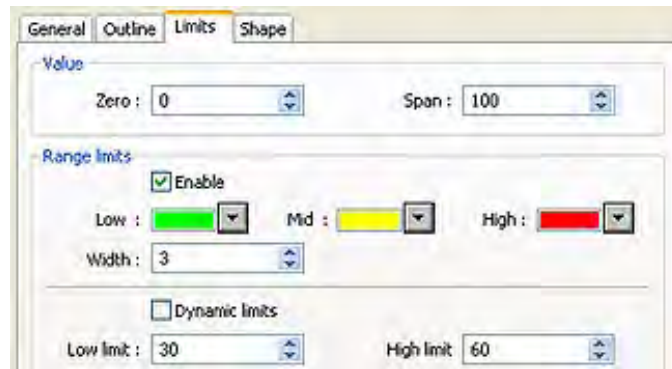
Note that meters can have any start and end degree between 0 and 360.

- The Background section allows you to select background options for the meter. The following illustrations illustrates how your meter will appear with different options checked:



- The Tick Marks section allows you to configure the length, scale and sub-scale of the tick marks.
- In the Pointer section, select *arm style*, *color*, *length*, and *width*.
- The Pinpoint section allows you to adjust the style and size of the pinpoint. See the illustration above to see the pinpoint.

10. Select the **Limits** tab to display the Limits form.



11. In this section, you can configure a scale that goes on the face of the meter that shows by a color indication when high and low limits have been exceeded. If you look at the example on the previous page, you will see this is represented on the face of the meter as a red, yellow and green strip. In the Value section, select the *zero* and *span*.
12. In the Range limits section, if you'd like to enable the limits, click the **enable** check box and then select the *colors* for low, mid and high. Select the *width*, in pixels, of the color band. The values from 'zero' to low limit will be on the 'low' color band; values from low limit to high limit will be on the 'mid' color band; values greater than high limit will be on the 'high' color band.
13. Select the *low limit* and *high limit*. If the limits are dynamic, click the **Dynamic limits** check box and then configure the PLC addressing that pops up.
14. If you'd like a background shape to appear with your meter, click the Shape tag and select a shape.
15. Click **OK**. The meter appears on the work area of EasyBuilder. Click to drag and place your meter on your work area. If you need to reconfigure any of the properties, you may double click on the object and the properties dialog box will reappear.

## Creating Trend Displays & Data Sampling Objects

Data Sampling objects can be used to log historical data. Once data is historically logged, it can be saved to the OIT's internal memory, or to an external memory source such as CompactFlash, USB or a PC.

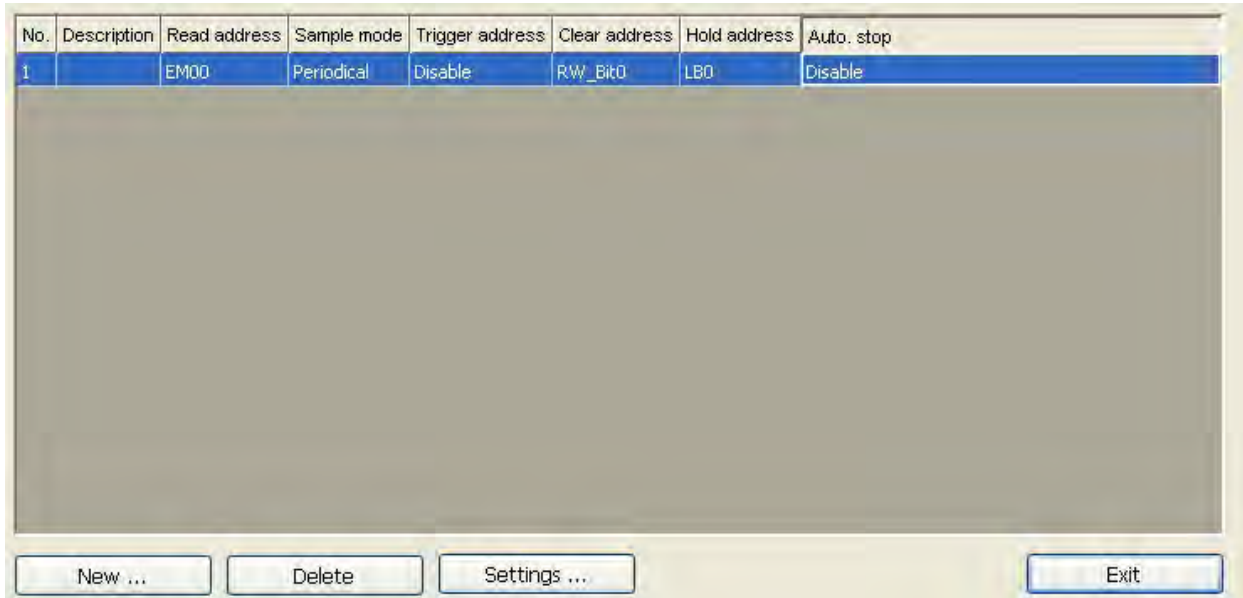
The Trend Display Object is used to sample the data in a single or multiple 16-bit PLC registers and plot this data on a time graph. You can configure the Trend Display as a single page or with multiple pages that can be viewed by scrolling through the pages. The sampling rate and number of plots per page can be customized per trend graph. You can display a maximum of 16 channels with varying colors to distinguish the channels. Finally, a hold feature allows you to 'freeze' the graph and/or clear the graph.

In order to create a trend object, you must first create a number of data sampling objects from which to sample the trends.



## To create a data sampling object

1. From the Objects menu, select **Data Sampling**. The Data Sampling Object Dialog appears.



2. Click **New...** to create a new data sampling object.

The image shows a configuration dialog box for creating a new data sampling object. It is divided into several sections:

- Description:** A text input field.
- Sampling mode:** Two radio buttons:  Time-based and  Trigger-based.
- Sampling time interval:** A dropdown menu set to "1 second(s)".
- Read address:**
  - PLC name:** A dropdown menu set to "Local HMI".
  - Device type:** A dropdown menu set to "LW".
  - Address:** A text input field set to "0".
  - System tag
  - Index register
- Data Record:**
  - Max. data records:** A text input field set to "1000".
  - Auto. stop
  - Data Format ...** button
  - Data length:** A text input field set to "0 word(s)".

PLC name : Local HMI

Clear address

Enable

Device type : LB

Address : 0  System tag

Index register

Hold address

Enable

Device type : LB

Address : 0  System tag

Index register

History files

Save to HMI memory  Save to CF card

Save to USB 1  Save to USB 2

3. Description: Enter a *name* for the object. This isn't necessary, but will help in later identification of the object.
4. Max data records: Set the *maximum number of records* to log.
5. Select the *Device type* from the drop down menu, and then specify the *PLC name* and *address*. Click **Data Format...** to set configure the format of each logged item. The Data Format dialog appears. Click **New...** to call the configuration dialog.

Description : 16-bit Unsigned

Read Address

Data type : 16-bit Unsigned


Save to default OK Cancel

6. Enter a *Description* for the data format and then select the *Data type*. Click **OK** to return to the Data configuration dialog and then **Exit** to return to the Data Sampling dialog.
7. Click the **enable** check box in the Clear address section if you would like the data sample to be cleared when the bit is momentarily set. If enabled, specify the *PLC address* and *device type*.

8. Click the **enable** check box in the Hold section if you would like the data sample to be paused as long as the specified bit is set. If enabled, specify the *PLC address* and *device type*.
9. Select either *time based* or *trigger based* sampling of data. Time based samples data at pre-set intervals of time. Trigger-based samples data when a specified bit undergoes a specified change. If time-based, specify the *sampling time interval* (in seconds). If trigger-based, specify *Device type*, *PLC address* and *PLC name*.
10. Check the **auto stop** check box if you would like the data sampling to automatically stop when the number of records in Max data records has been reached. Otherwise, the existing logged data will be deleted and new data will be recorded.
11. Select the *type of memory* to which you would like to save the data files, and then specify the *filename* under which the data will be saved.
12. Click **OK**.
13. Follow steps 1-12 to create additional data sampling objects.
14. Click **Exit** to exit the data sampling object.

► **To create a Trend Display Object** 

1. From the Objects menu, click **Trend Display**. Or click the **Trend Display** icon in the Part2 toolbar. The New Trend Display Object dialog box appears.



General Trend Shape

Description :

Data Sampling Object index : 0

Trend type : Real-time No. of channels : 1

Note : if no. of channels is changed, you must reset HMI's data logs !!

Distance between data samples :  Pixel  Time

Distance : 100 pixel(s)

PLC name : Local HMI

Hold control

Enable

Device type : LB

Address : 0  System tag  Index register

Watch line

Enable

Device type : LW

Address : ch.19  System tag  Index register

2. In the Data Log Object Index, select the *data sampling object* whose data is to be trended from the drop down menu.
3. For Display Mode, select whether the data showing will be historical data or data in real time.

4. In channel no., select the number of pens tracking data. This number should be less than or equal to the number of items configured in the selected data log index.
5. Select the name of the *PLC* to read/write to from the drop down menu.
6. Check the **Enable** checkbox in the Hold Control section if you'd like the trend to pause when a specified bit is set, and then configure the *PLC address* and *device type*.
7. Click the **Enable** checkbox in the Watch line section if you'd like to place a watchline on the trend that places a line at a specific register value, and then configure the *PLC address* and *device type*.
8. Configure the History control section if the Display mode is set to history.
9. Click on the **Trend** tab to display the Trend form.

The screenshot shows the 'Trend' configuration dialog box with the following settings:

- General** tab: Frame (dropdown), Background (dropdown),  Show scroll controls.
- Grid**:  Enable.
- Channel**: Channel: 0 (dropdown).
- Pen property**: Color: Black (dropdown), Width: 2 (dropdown). A preview shows a black line on a white background.
- Limits**:  Dynamic limits, Zero: 1 (text box), Span: 2 (text box).
- Time/Date**:  Time (radio buttons: HH:MM:SS, HH:MM),  Date (radio buttons: MM/DD/YY, DD/MM/YY, DD.MM.YY), Color: Red (dropdown).

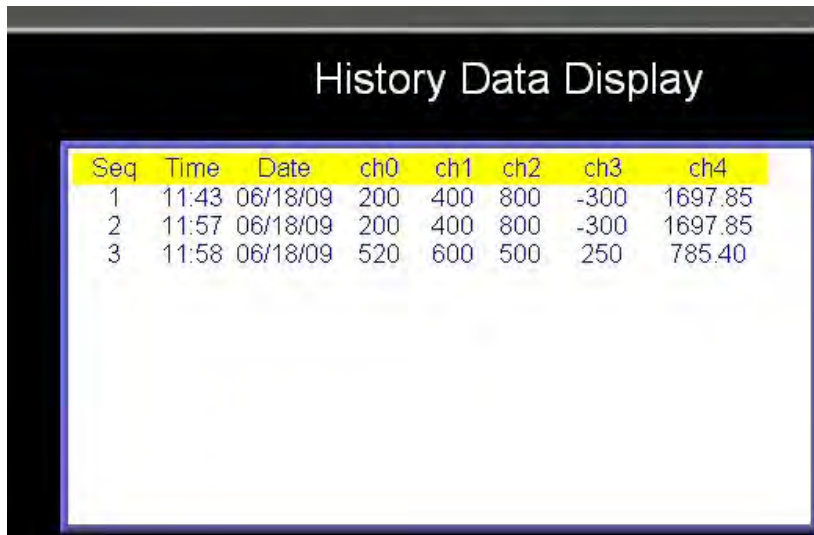
10. Enter a *description* for the trend. This is not necessary, but will help in identification later.
11. Select the *frame color* and the *background color* for the trend from the drop down menus.

12. Check the **Enable** checkbox in the grid section if you would like the trend to display a grid, and then configure the X and Y axis of the grid from the scroll boxes.
13. In the channel section, select the channel to configure from the drop-down menu, and then select the pen properties for that channel. Next, select the *value* at which the pen will be at the bottom of the trend (zero) and the span of the trend. Finally, check the **Enable** checkbox if you would like the time and date to be displayed on the OIT then select the format in which you'd like the time and date displayed.
14. Click the **Shape** tab to configure a background shape, if any.
15. Click the **Profile** tab if you'd like to position your trend using the properties dialog using the X,Y position instead of dragging and placing it in the work area.
16. Click **OK**. The Trend appears on EasyBuilder's work area. Move the trend to place it on the display. If you'd like to reconfigure any of its properties, double click on the trend to display the Trend Display Object's Properties Dialog.

## History Data Display

The History Data Display shows data captured by the Data Sampling object (for more information on using the Data Sampling object, see Chapter 12: Bar Graphs, Meters and Trends, Section 'Creating Trend Displays & Data Sampling Objects').

Data is displayed in tabular format with option to show the time and/or date of when data was captured. A control register is used to determine which historical data record is displayed (i.e. most recent, yesterday, etc.) based upon date of capture:



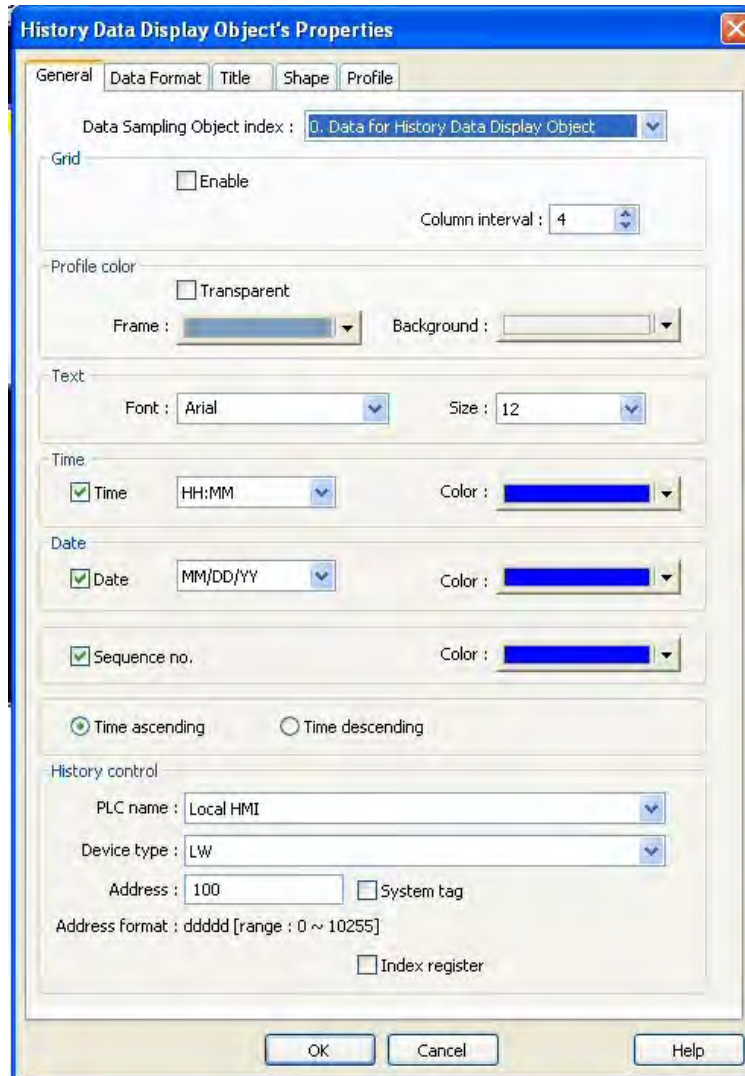
The screenshot shows a window titled "History Data Display" with a table of data. The table has seven columns: Seq, Time, Date, ch0, ch1, ch2, ch3, and ch4. The data is as follows:

Seq	Time	Date	ch0	ch1	ch2	ch3	ch4
1	11:43	06/18/09	200	400	800	-300	1697.85
2	11:57	06/18/09	200	400	800	-300	1697.85
3	11:58	06/18/09	520	600	500	250	785.40

The OIT reads from the Sampling Object datalog file (\*.dtl) immediately after the screen that contains the History Data Display object is displayed. Note: the data is only read and updated whenever the screen is first displayed. Data recorded to the datalog file (\*.dtl) after the History Data Display object is shown will not appear on the History Data Display object until the screen is refreshed (closed, then opened again). A maximum of 20 channels can be displayed.

## To create a History Data Display Object

1. From the Objects menu, click **History Data Display**. Or click the History Data Display icon in the Object 2 toolbar. The New History Data Display dialog box appears:



2. **Data Sampling Object index:** select from drop-down list which data file to use. Data files are created using the Data Sampling object.
3. **Grid:** Displays grid lines on the data table.
  - a. Enable: check to enable grid lines
  - b. Color: select color used for grid lines
  - c. Column Interval: enter the number of pixels that separate each column of data.
4. **Profile color:** determines the color of the background of the object.
  - a. Transparent: enable if you do not want a background color or frame to appear.
  - b. Frame: color for the perimeter frame
  - c. Background: color for the background of the object.



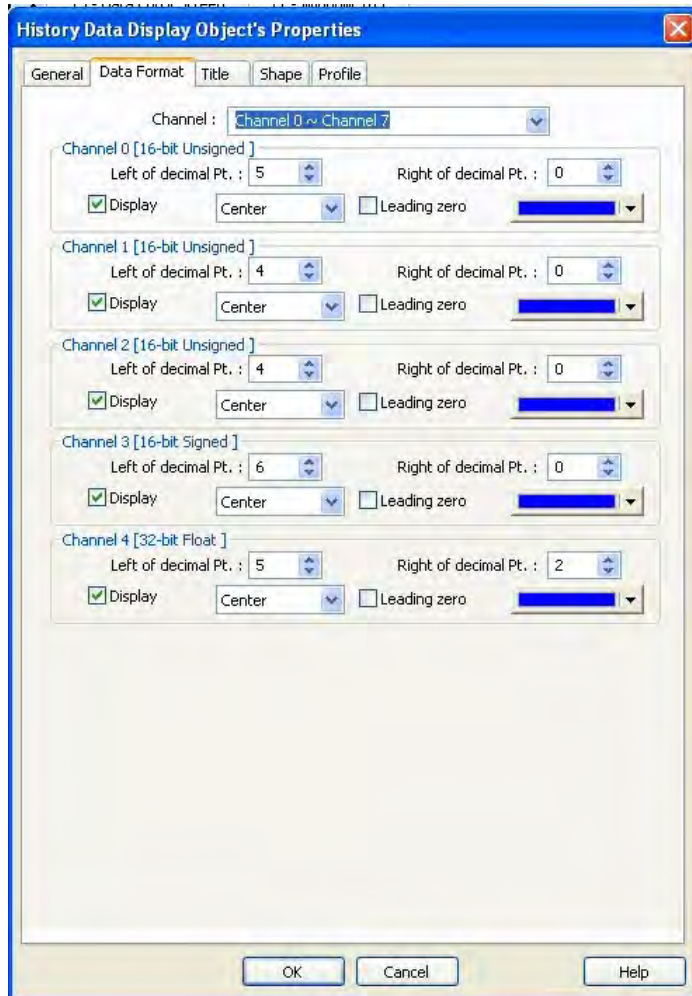
5. **Text:** select the font and size for all text that appears
  - a. **Font:** select the truetype font for the text
  - b. **Size:** select the size of the font
6. **Time:** enable to display the time when the data was captured by the Data Sampling object.
  - a. **Style:** select from HH:MM:SS, HH:MM, or DD:HH:MM
  - b. **Color:** determines the color used to display the time
7. **Date:** enable to display the date when the data was captured by the Data Sampling object.
  - a. **Style:** select from MM/DD/YY, DD/MM/YY, DD.MM.YY or YY/MM/DD
  - b. **Color:** determines the color used to display the date
8. **Sequence no:** enable to display a sequence number of when the data was captured by the Data Sampling object.
  - a. **Color:** determines the color used to display the sequence number
9. **Time ascending or Time descending:** determines the order in which the data is displayed.
10. **History Control frame:** use to select the target PLC or internal OIT register that is used to determine which data log file is displayed.

*For example, suppose we have data log files that were recorded during a four day period- 20090615.dtl (June 15<sup>th</sup>), 20090616.dtl (June 16<sup>th</sup>), 20090617.dtl (June 17<sup>th</sup>), and 20090618.dtl (June 18<sup>th</sup>).*

*We choose to use internal OIT memory address LW100 for the History Control address. If the value in LW100 = 0, then the History Data Display object will display the most recent data log record- in this example, 20090618.dtl (June 18<sup>th</sup>). If the value in LW100 = 1, then 20090617.dtl (June 17<sup>th</sup>) is displayed, and so on.*

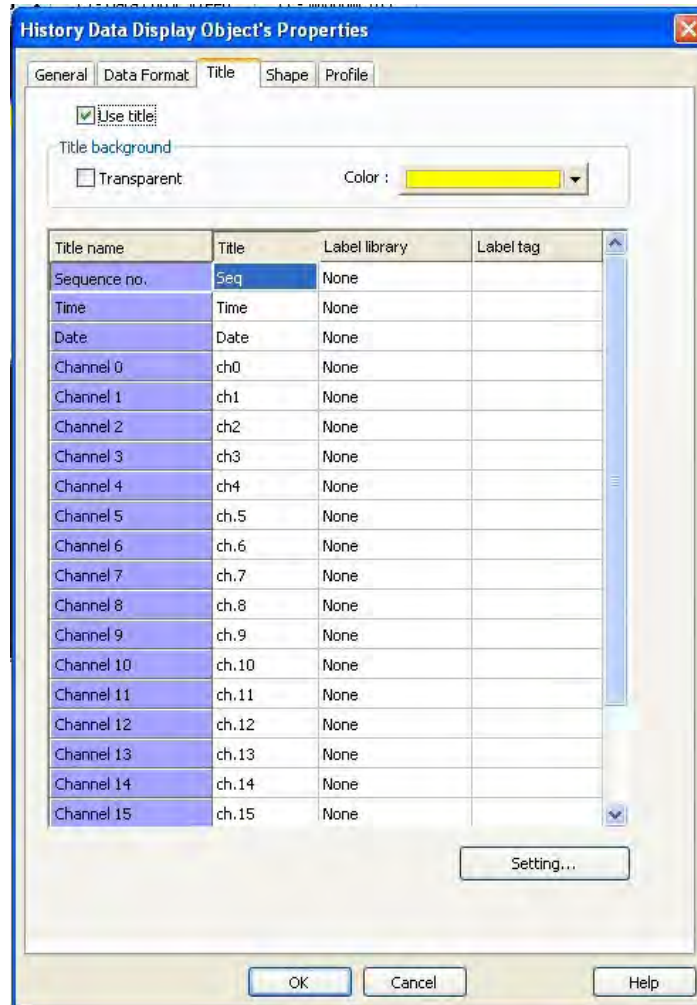
- a. **PLC Name:** see “General Properties” section at beginning of this chapter.
- b. **Device type:** same as above
- c. **History Control address:** this is the specific PLC address from the Device Type memory area. The History Control Address is used to determine which data log file is displayed. A value of 0 displays the most recent file, etc.
- d. **Index Register:** see “General Properties” section at beginning of this chapter.

**Data Format tab:** used to configure how the data appears for each column:



1. Channel: Channels are configured in groups of eight (Note: maximum of 20 channels). Use this pull-down box to select each group.
2. Channel #0-19 [16-bit Unsigned] frame: Configuration information for each channel. Description in parentheses tells you what format is used for that data variable.
3. Display checkbox: check to enable viewing that data variable.
4. Left of decimal pt: determines the number of digits that can be displayed to the left of the decimal point.
5. Right of decimal pt: determines the number of digits that can be displayed to the right of the decimal point.  
*Note: if 0, then no decimal point is displayed.*
6. Alignment: options are center, right, and left justified.
7. Leading zero: check to display leading zeroes
8. Color: select the color used to display the data.

**Title tab:** used to configure a title bar or legend at the top of the data columns:



1. **Use title:** check to use the Title bar option.
2. **Title background frame:** determines color of background on title bar
  - a. Transparent: check to remove background color
  - b. Color: select from drop-down color box
3. **Title name:** default name of each column
4. **Title:** name assigned to appear at the top of each column
5. **Label Library:** None = not used, Used = Label Library used instead of direct text entry. To assign a Label Tag, highlight the channel you wish to modify and click on the Setting... button. Note: for more information on using Label Libraries, see Chapter 9 – Creating and Using Databases and Languages, section ‘Using the Label Library’
6. **Label tag:** blank = not used, or the particular name of the label tag used.

Shape tab: refer to beginning of this section.

---

## Chapter 13 - Capturing Alarms and Events

This chapter looks at how the Silver Series uses alarms and events.

### Using Alarms

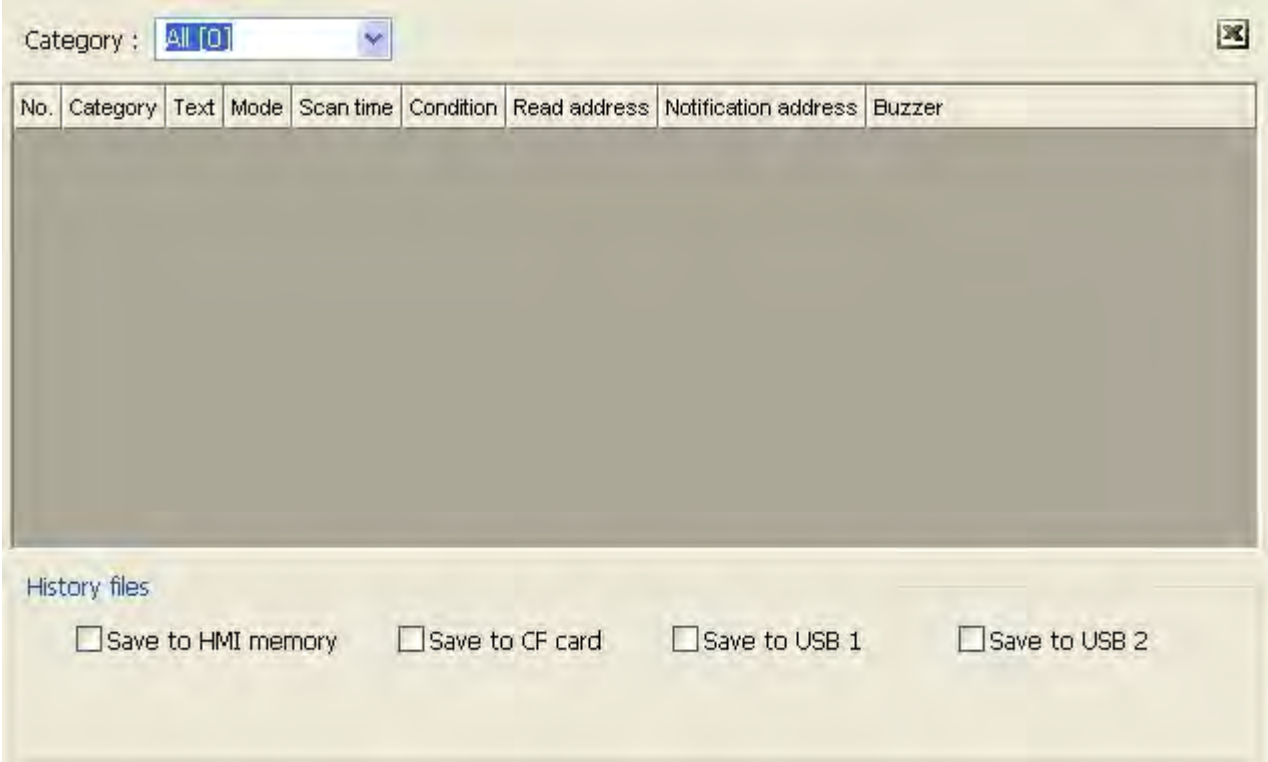
EasyBuilder has three parts that are used to perform alarm functions: the Alarm Scan object, the Alarm Display object, and the Alarm Bar object. The Alarm/Event Log object monitors alarm conditions and alerts the Alarm Display object and the Alarm Bar object when an alarm condition occurs. Use the Alarm Display object if you want to display a scrollable list of all alarms that are active. Use the Alarm Bar object to display all active alarms on a single horizontally scrolling line (like a marquee).

### Monitoring Alarms with the Alarm/Event Log

The Alarm/Event Log continuously monitors PLC coils to determine if an alarm condition has occurred. Once an alarm is active, the Alarm Scan Object directs a string of characters associated with that alarm to the Alarm Display Object and the Alarm Bar Object for display on the OIT screen.

► To create an Alarm/Event Log 

1. From the Objects menu, click **Alarm-Alarm (Event) Log**. Or click the **Alarm (Event) Log** icon in the Draw toolbar. The Alarm(Event) Log dialog box appears.



Category :

No.	Category	Text	Mode	Scan time	Condition	Read address	Notification address	Buzzer
-----	----------	------	------	-----------	-----------	--------------	----------------------	--------

History files

Save to HMI memory     Save to CF card     Save to USB 1     Save to USB 2

2. Click the **New...** button. The Alarm Event Log Properties dialog appears.

The screenshot shows the 'Alarm Event Log Properties' dialog box with the 'General' tab selected. The dialog is divided into several sections:

- General:** Category: 0, Priority level: Low (dropdown).
- Address type:** Word (dropdown), Scan time: 500 ms (dropdown), Delay time when power on: 1 second(s) (dropdown).
- Read address:** PLC name: Local HMI (dropdown), Device type: LW (dropdown), Address: 0, System tag (checkbox), Index register (checkbox), 16-bit Unsigned (dropdown).
- Notification:** Enable (checkbox).
- Condition:** Trigger if value is: < (dropdown), 0 (input field).

Buttons at the bottom: OK, Cancel, Help.

3. Select a *priority level* from the drop down menu.
4. Select either *bit* or *word* from the address type drop down menu, and then select the scan time (in millisecc) from the drop-down menu.
5. Select the *PLC name*, *device type* and *read address*, as well as the *format* in the Read Address section.
6. Click the **Enable** checkbox in the Notification section to have a specific bit set or cleared when the alarm condition is reached, and then specify the *condition*.

- Click the **Message** tab to display the Message form.

- Enter the *message* to display when the alarm or event is triggered. If you would like to configure the message in different languages, click **Label** and follow the instructions in *Chapter 7 - Creating Databases and Languages*.
- In the Write value window, enter the *window number* to display once the alarm or event is acknowledged.
- Click the **Enable** checkbox in the Sound section to have the OIT play a sound when the alarm is triggered. Click **Sound library...** to select a sound from the sound libraries. If no sound is selected, the default sound is a beep. Click **Play** to preview the sound.
- Click **OK** to return to the Alarm (Event) Log dialog. You may select a location to save History files by selecting one of the check boxes. You may also configure additional Alarm/Event log items here. Click **Close** when you are finished.

## Displaying Alarms using the Alarm Display Object

Though the Alarm (Event) Log Object continuously monitors the PLC for alarms, it cannot display the alarms without the Alarm Display Object or the Alarm Bar Object. The Alarm Display Object takes the alarm strings sent to it from the Alarm (Event) Log Object and displays them on a window screen as a list. The list contains all of the active alarms occurring with the most recently activated alarm at the top of the list. You do not have to display the entire list of alarms on a window screen. You can limit the viewable alarms to a 'scrollable window' that displays a given number of lines. You can then create a register that allows you to scroll through the list of alarms using the scrollable window.

► To create an Alarm Display Object 

1. From the Objects menu, click **Alarm-Alarm Display**. Or click the **Alarm Display** icon in the Part2 toolbar. The New Alarm Display Object dialog box appears.

2. Include categories: to include all categories (there can be up to four categories), set this at 0 to 0.
3. Select *frame* and *background colors* from the drop down menus.
4. Sort the *order* in which the alarms will appear - either ascending or descending.
5. Check the **Event trigger time** checkbox if you'd like the time the event was triggered to appear in the alarm listing, and then select the *format* you'd like the time listing to appear.
6. Click the **Shape** tab to configure a background shape, if you'd like one.
7. Click the **Font** tab to configure the font.
8. Click **OK**. The Alarm display appears on the work area of EasyBuilder. Drag it to place it where you'd like on the display.

### Displaying Alarms using the Alarm Bar Object

The Alarm Bar Object displays alarms scrolling horizontally along a single line. The alarm bar continuously scrolls each alarm until the alarm is no longer active. If more than one alarm is active, the Alarm Bar Object appends each alarm to the string of characters scrolled. This part can be used on window screens where space is very limited. It also allows you to display long alarm text strings that are too long to be shown in the Alarm Display Object.



► To create an Alarm Bar Object 

1. From the Objects menu, click **Alarm-Alarm Bar**. Or click the **Alarm Bar** icon in the Part2 toolbar. The Create Alarm Bar Object dialog box appears.

2. Include categories: to include all categories (there can be up to four categories), set this at 0 to 0.
3. Select scroll speed from the drop down menu to adjust the speed at which the message scrolls across the alarm bar.
4. Select *frame* and *background colors* from the drop down menus.
5. Sort the *order* in which the alarms will appear - either ascending or descending.
6. Check the **Event trigger time** checkbox if you'd like the time the event was triggered to appear in the alarm listing, and then select the *format* you'd like the time listing to appear.
7. Click the **Shape** tab to configure a background shape, if you'd like one.
8. Click the **Font** tab to configure the font.
9. Click **OK**. The Alarm bar appears on the work area of EasyBuilder. Drag it to place it where you'd like on the display.

The Alarm Bar Object scrolls through each character of an alarm string at a predefined rate. You can change the rate at which the characters scroll by doing the following.

## Using Events

EasyBuilder has two parts that are used to perform Event functions: the Event Log object and the Event Display object. The Event Log object monitors events and alerts the Event Display when an event is triggered. The Silver Series monitors a maximum of 200 events but you can program the OIT to monitor up to an additional 1000 events.

### Monitoring Events With the Event Log Object

The Event Log Object continuously monitors PLC coils and registers to determine if an event has occurred. Once an event has triggered, the Event Log Object directs a string of characters associated with that event to the Event Display Object for display on the HMI screen. The OIT operator can acknowledge each event by touching the string of characters that have been displayed. Once acknowledged, the Event Log Object has the option of calling a popup window.

#### ► To create an Event Log Object

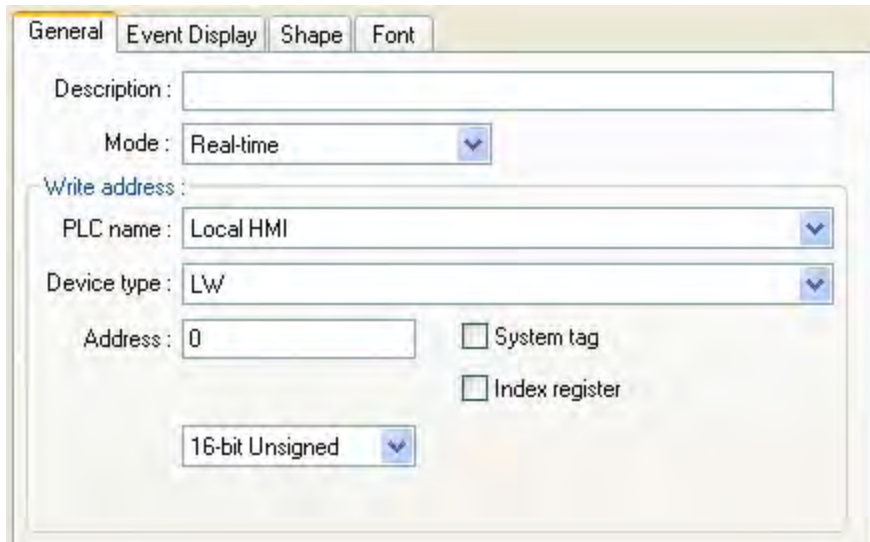
See the section on the Alarm (Event) Log.

### Displaying Events Using the Event Display Object

Though the Event Log Object continuously monitors the PLC for triggered events, it cannot display the events without the Event Display Object. The Event Display Object takes the event strings sent to it from the Event Log Object and displays them on a window screen as a list. The list contains all of the events that have occurred with the most recently triggered event at the top of the list. You do not have to display the entire list of events on a window screen. You can limit the viewable events to a 'scrollable window' that displays a given number of lines. You can then create a register that allows you to scroll through the list of events using the scrollable window. An acknowledge feature allows the OIT operator to touch an event recorded in the event list to display a popup window. The popup window can then be used to display further instructions to the OIT operator or to perform some operation.

#### ► To create an Event Display Object

1. From the Objects menu, click **Alarm- Event Display**. Or click the **Event Display** icon in the Part2 toolbar. The New Event Display Object dialog box appears.



2. Use the **Description:** box to enter a *title* for the Event Display part. A description is not necessary but does help identify the purpose of the part.
3. **Specify the PLC name** from the drop-down menu, and then select either real-time or history as the *mode* for the events to be displayed.
4. In the Write Address section, **select the Device type, address and type.**

- Click on the **Event Display** tab to display the Event Display form.

The screenshot shows the 'Event Display' configuration dialog box with the following settings:

- Include categories:** 0 thru 0 (see Alarm (Event) Log object)
- Acknowledge style:** Click
- Max. event no.:** 200
- Color:**
  - Frame: Black
  - Background: White
  - Acknowledge: Yellow
  - Return to normal: Green
  - Select box: Blue
- Format:**  Sequence no.
- Sort:**  Time ascending,  Time descending
- Time:**
  - Event trigger time,  Acknowledge time,  Return to normal time
  - HH:MM:SS,  HH:MM,  DD:HH:MM
- Date:**
  - Event trigger date
  - MM/DD/YY,  DD/MM/YY,  DD.MM.YY

- Include categories:** to include all categories (there can be up to four categories), set this at 0 to 0.
- Select click or double click** to acknowledge event with a single or double touch in the acknowledge style drop-down menu. Then select the maximum number of events before the OIT will begin overwriting old events. The valid range is 10-1000.
- Select colors** for background, frame, select box, acknowledge and return to normal.
- Check the **Sequence no** check box to add a unique serial number to each event.
- Select the **order** in which events will be sorted.
- Check the **Event trigger date** check box to have the time the event is triggered added to each event, and then choose your preference for the time format.
- Click the **Shape** tab to add a background shape, if desired.
- Click the **Font** tab to configure the font.
- Click **OK**. The object appears on the work area of EasyBuilder. Place it in the desired location.

## Chapter 14 - Data and Recipe Transfer Objects

EasyBuilder includes a Data Transfer object and a Recipe Transfer object, which enable data to be copied from one or more registers located in the PLC or the OIT to other registers. The Data Transfer object is used to continuously transfer data at a predefined rate. The Recipe Transfer object is a touchscreen object that transfers data when activated by the OIT operator.

### Using the Data Transfer Object

In some projects you create, you may need to transfer large blocks of data from PLC/OIT registers to other PLC/OIT registers. The Data Transfer Object is used to copy data from one or more PLC registers or OIT internal memory to other registers continuously on a timed interval. There is no limit to the number of Data Transfer Objects that you can create other than the OIT's available memory. Each Object created can transfer from 1-16 continuous registers or 1-255 coils.

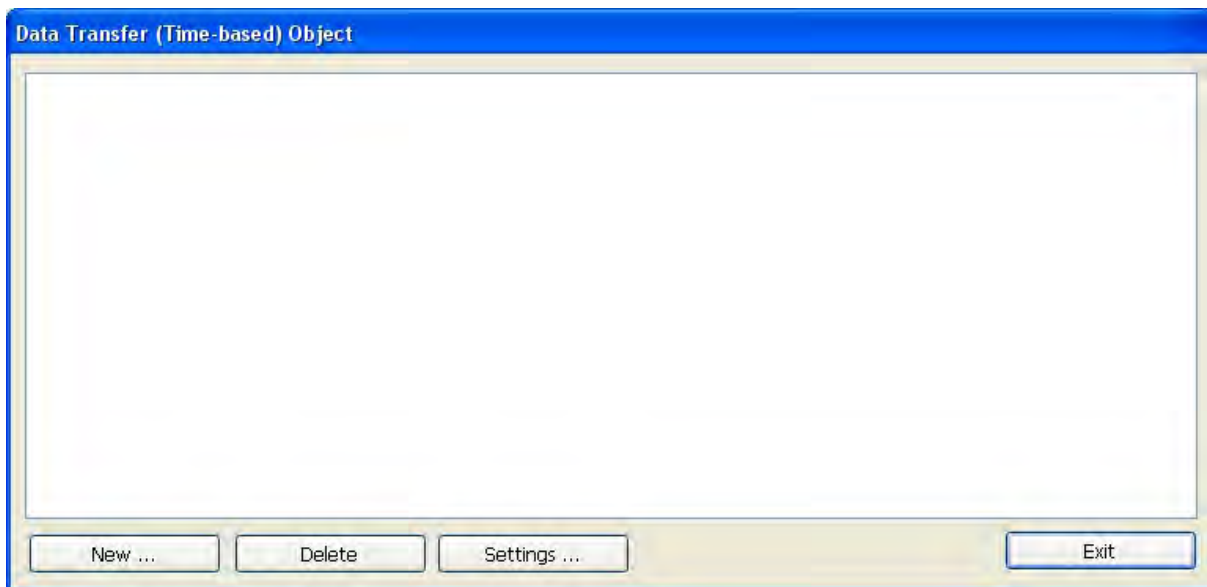
There are two types of data transfer objects available, time-based and trigger based. Time-based data transfer object periodically transfers data from one set of PLC registers to another at a predetermined interval. The interval is adjustable from 0 (the data is transferred continuously) to 25.5 seconds, in 0.1 second steps.

The trigger-based data transfer object transfers word-based data from a specified source address to another specified source address when triggered by the controller. This can be used to transfer recipe data, or any other data, from one device to another.

#### ► To create a Time-based Data Transfer Object



1. From the Objects menu, click **Data Transfer**. Or click the **Data Transfer** icon in the Draw toolbar. The Data Transfer Object dialog box appears.



- Click the **New...** button. The Data Transfer (Time-based) Object dialog box appears.

The screenshot shows the 'Data Transfer (Time-based) Object' dialog box. It has a light yellow background and is organized into four main sections:

- Description:** A text input field at the top.
- Attribute:** Contains three fields: 'Address type' (a dropdown menu set to 'Bit'), 'Interval' (a dropdown menu set to '3.0 second(s)'), and 'No. of bits' (a text input field set to '1').
- Source address:** Contains three fields: 'PLC name' (a dropdown menu set to 'Local HMI'), 'Device type' (a dropdown menu set to 'LB'), and 'Address' (a text input field set to '0'). Below these are two checkboxes: 'System tag' and 'Index register', both of which are unchecked.
- Destination address:** Contains three fields: 'PLC name' (a dropdown menu set to 'Local HMI'), 'Device type' (a dropdown menu set to 'LB'), and 'Address' (a text input field set to 'ch.19'). Below these are two checkboxes: 'System tag' and 'Index register', both of which are unchecked.

- Use the Description: box to enter a *title* for the Data Transfer part. A description is not necessary but does help you identify the purpose of the part.
- In the Attribute box, configure the Address type to transfer whether by bit or word, the interval of transfer and the number of bits or words to transfer.
- In the Source address frame, select the *PLC name* and *device type* from their drop down menus and then configure the *address*.
- In the **Destination address** frame, select the PLC coils or registers that receive the data.
- Click **OK** to close the Data Transfer Object dialog. The Data Transfer Object dialog box reappears with the new entry listed. You can edit the attributes of an existing Data Transfer object by clicking on the **Settings...** button. Any entries can be deleted by clicking on the **Delete** button.
- Click **Exit** to close the Data Transfer Object dialog box. The Data Transfer Object is active no matter what windows are currently displayed on the OIT screen; therefore, you do not have to place the object onto a window screen.

## Using the Recipe Transfer (Data Transfer, Trigger Based) Object

The Data Transfer, Trigger-Based (Recipe Transfer) Object is used to download or save data from one or more PLC registers to the OIT's internal registers that are reserved for recipes. This occurs when the object is activated by a press from the OIT operator. Each Object created can transfer from 1-32767 continuous registers.

### ► To create a Recipe Transfer (Data Transfer, Trigger Based) Object

- From the Objects menu, click **Data Transfer (Trigger-based)**. Or click the **Data Transfer**

(**trigger-based**) icon in the object 2 toolbar. The New Data Transfer (Trigger-based) Object dialog box appears.

2. Use the Description: box to enter a *title* for the Recipe Transfer part. A description is not necessary but does help you identify the purpose of the part.
3. In the Source Address frame, select the PLC registers that are to be saved or downloaded. The No. of words box is used to specify the total *number of registers* used. Select the *PLC name* and *device type* from their respective drop-down menus. Check the **Index register** checkbox if you'd like a preset offset to be added to the specified register, and then select the *index* from the drop-down box.
4. In the Destination Address frame, select the PLC registers that are to be saved or downloaded to. Destination address information is configured in the same way as Source Address information in #3 above.
5. In the Attribute frame, select either touch trigger or external trigger. If selecting external trigger, configure trigger mode and trigger address information that pops up.
6. Click the **Security** tab and configure security settings, if needed.
7. Click the **Shape** tab to configure background shape, if needed.
8. Click the **Label** tab to apply a text label, if needed.
9. Click **OK**.

## Creating a Recipe

A recipe consists of a number of registers using the RW or RB devices. These devices are in a non-volatile area of the OITs' internal memory area. A recipe is created by using the OIT's built-in Parts to store data in the RW/RB devices, and then transferring that data to the selected PLC registers.

**Recipe Devices** There are 4 types of Recipe devices: Recipe Word (RW) , Indexed address Recipe words (RWI), Recipe Bit (RB), and Indexed address Recipe Bit (RBI). The usage of RW and RB devices is the

same. Therefore, in the following discussion, *Rn* refers to both RW and RB devices, and *RnI* refers to both RWI and RBI devices.

**Rn Device**

Refers to a direct *Rn* address. That is, the value placed in the Address field of an *Rn* device refers directly to the *Rn* address. For example, *Rn100* refers directly to the data at address *Rn100*.

**RnI Device**

An address with an *RnI* device is an *Indexed* address. The address referred to by an *RnI* device is actually the specified address plus an indexed offset. The offset is the value in address LW9000. For example, suppose address LW9000 contains the value 50, and the Address field of an *RnI* device is set to 100. This *Indexed* address actually refers to the data at address *Rn150*.



*RB devices use the same memory area as the RW devices. The RB device allows Bit-level access to recipe data. The bit address is specified by using the RW address, followed by the bit number 0-F (hex). For example, Bit RB0 is Bit 0 in RW0; Bit RW2047F is Bit 16 in RW2047.*

---

**Rn Devices Vs RnI Devices****RnI devices**

Indexed address devices are most useful when it is necessary to ‘page’ through a number of recipes. *RnI* devices allow a single window to manage all recipe items. Conversely, the use of *Rn* devices requires a separate window for each recipe, with each object on each window having a separate address. This not only uses more of the OIT’s resources, but could also lead to configuration errors as objects are copied, pasted, and edited.

**Rn devices**

Use when it is desired to show unique information from a recipe. For example, it might be necessary to show, one window, the names of all of the recipes.



*For an example of how to create recipes, please consult the EZware-5000 online Help file.*

---

**Backup Part**

The Backup part is used to copy Recipe, Event, or Historical data to one of the USB memory devices attached to the OIT. The copy can be triggered by a touch or by a bit transition.

## To create a backup part:

1. From the **Object** menu, select **Backup**, or click the **Backup** icon on the Object2 toolbar. The New Backup dialog appears.

2. Use the **Description:** box to enter a title for the MultiState Switch part. A description is not necessary but does help you identify the purpose of the part.

Source	Select which dataset to back up. <i>Data Log</i> is available only if a Data Sample has been set up.
Data Sampling Object Index	Select the Data Sample set to back up. Available only when the Source is set to <i>Data Log</i> .
Backup Position	Select which USB device to back the data up to. Note: USB1 is the device that was installed first; USB2 is the device that was installed last.
Range Start	Select whether to start with <i>Today's</i> data or <i>Yesterday's</i> data.
Range Within	Select the number of days' worth of data to back up.
Attribute Mode	<i>Touch Trigger</i> causes the backup to execute when the object is touched. <i>External Trigger</i> causes the backup to execute when the specified bit transitions.
Attribute Trigger Mode	Specify how the bit must change to trigger the backup. Available only when Attribute Mode is set to <i>External Trigger</i> .
Trigger Address	Specify the <i>bit</i> address with which to trigger the backup. Available only when Attribute Mode is set to <i>External Trigger</i> .

3. Click the **Security** tab and configure security settings, if needed.
4. Click the **Shape** tab to configure background shape, if needed.
5. Click the **Label** tab to apply a text label, if needed.
6. Click **OK**.



**Please Use This Page For Your Notes:**

## Chapter 15 - Macros

### Using Macros

The “macro” language in EasyBuilder allows the programmer to perform simple transfers of data and/or simple to complex conditional logic and transfers.

Macros are high-level language programs that can run inside the OIT to perform various data transfers and can perform conditional logic. They are useful for doing tasks that are not supported in a standard object or feature of the OIT. The macro language combines syntax elements of C, C++ and a little of its own style.

Previous experience in programming with C or C++ would be helpful when developing macros. It is not the scope of this document to explain concepts of high-level language programming and the basics such as: variable data types, arrays, conditional logic, and variable passing. However, it is the scope of this document to discuss the commands and command structures that are required in building macros.

### Macro Sample and Implementation

Macros can be used for customizing an application or when a special operation is needed that the normal OIT objects (set bit, data transfer, etc) do not provide. A few possibilities are listed:

- Unusual or non-linear scaling of data
- Displaying the result of arithmetic operations on 2 or more PLC register values
- Indirect addressing (where a PLC register holds the address of the desired data)
- Lookup tables
- Transferring Data once, instead of periodically
- Writing data to the PLC in a specific sequence
- Timed splash screens

A macro is written using the “macro” utility found in the ‘Tools’ menu list of Easy Builder. The ‘Add’ button will create a new worksheet to program a new macro. The programmer first gives the macro a name in the top right corner. Then programming can be added. After the code is finished, it needs compiled. If the code compiles correctly there will be no errors and the macro name will be added to the top list. If the macro does not compile correctly it will display error codes. If the programmer should cancel out from creating the macro before a successful compile, it will put the macro into the bottom panel of the macro list and it will not be available as a usable macro, but the text will remain to be edited later.

The macros are triggered by a setting a bit or a function key. The bit is specified in the ‘PLC Control Object’ list. The bit may be a local bit ‘LB’ or it may be a PLC bit. Each macro created will have a unique name and bit. When a specified bit is set high so it executes a specific macro, the macro will be entered into and its variables initialized. The macro will run its sequence of events and will exit when finished. The bit that called the macro into execution should be cleared by the macro before it exits, otherwise the macro will immediately be entered again upon exit. If this immediate re-entry should happen, this could form an endless loop and will certainly cause a taxing of the CPU time and cause slow screen refreshing and communications.

A sample macro is listed below which reads two local word registers and adds them together and stores them into another register location:

---

```

macro_command main()

short num1
short num2
short result
bool AllDone = 0

// get two registers to add together
// (LW0 & LW1)
GetData(num1, "Local HMI", LW_BIN, 0, 1)
GetData(num2, "Local HMI", LW_BIN, 1, 1)

// add the values

```

```

result = num1 + num2

// store the result
SetData(result, "Local HMI", LW_BIN, 2, 1)

// clear the macro trigger bit
SetData(AllDone, "Local HMI", LB_BIN, 1, 1)

end macro_command

```

The macro was named "AddTwo" and the bit that executed this macro was LB:1. After the macro was written, a 'PLC Control Object' was added where the 'Type of control' was selected first to be 'Execute macro program'. Then the device type showed what bit types were available. A 'LB' was selected and the address '1' was entered. Finally, the macro name was selected. Below shows the PLC control object attribute window.

The image shows a dialog box titled "PLC Object's Attributes". It has several sections:

- Description:** A text box containing "Executes the "AddTwo" Macro when set".
- Read address:** A section with a "Device type" dropdown menu set to "LB" and a "Device Address" text box containing "1". Below this is an unchecked checkbox labeled "Aux.".
- Attribute:** A section with a "Type of control" dropdown menu set to "Execute macro program" and a "Macro name" dropdown menu set to "AddTwo".
- At the bottom, there are "OK" and "Cancel" buttons.

PLC Object's Attributes



Project Example of Add 2

**Toggle Switch Objects Attributes Dialog**

In the example above, a toggle switch object was placed on the screen as a '+' button. When this button is pushed, it forces LB:1 to ON and the 'AddTwo' macro executes, storing the result in LW:2 and turning LB:1 to OFF and the macro stops.

## Variables, Declarations and Memory Usage

Variables, constants & functions are named by the programmer. Variables and constants are called *operands*. Variable names can be any character or number (a-z, A-Z, 0-9, or '\_') and can be as many characters as you need to describe it (the compiler has been tested to variable name lengths of 32 characters, but it is best to keep the variable names small for ease of reading the code). Variable names are not case sensitive, so 'result' is the same as 'ReSult' or 'RESULT'. Variable names must always start with a letter (i.e. variables cannot begin with a number or the underscore '\_' character).

### Memory Usage

Macros use a minimum of about 2.5K, plus the memory required for each variable type:


- float: 4 bytes (32-bits)
- int: 4 bytes (32-bits)
- short: 2 bytes (16-bits)
- char: 1 byte (8-bits)

- bool: 1 bit
- arrays: the corresponding number of bytes for each element, plus another 2 bytes for the index

Macros use memory from the background task (defined as *Window 0*) area. There is a default memory size of 320K available for macros (only 220K is allotted if the 'Fast Selection' task bar is enabled) and there are other background task objects that also use memory in this area, such as:

- Trends
- Data Transfers
- Alarms
- Events
- PLC Controls
- Printer functions

---

 *If the application exceeds the 320k limit, both the simulator and the OIT will display a System Severe Error message.*

---

## Variable Declarations


Each variable that will be used in the macro needs to be declared as a specific 'type' of register. The declarations are listed as the first part of the macro. Any declarations contained within the macro function is considered 'Local' and any variable outside the macro function is considered 'Global'. Local variables are only seen within the function they are declared in. Global variables retain their values and can be globally used by all functions. Below, are listed the various data types that can be declared:

### TypeDescription

---

Float	Single-Precision Floating point variable (32-bit signed, IEEE-754 format)
Int	Integer variable (32-bit signed)
Short	Short integer variable (16-bit signed)
Char	Character variable (8-bit unsigned)
Bool	Boolean variable (1-bit)

---

 *Variables inside macros are initialized to all '1's as a default (i.e. 0xFFFF), so don't assume they are zero values when you enter the macro. It is good programming practice to initialize variables during declaration, or use assignment statements before they are used.*

---

## Variable Initialization

Initialize a value of variable in the declaration statement directly. (e.g: `int RPM = 75`) Use the assignment operator (=) to initialize a value to the variable. Variables can be declared and initialized in the following manner:

Stacked Example: Inline Example (separate like-types with a comma):

---

```
short a = 0
short b = 0
short c = 0
short a=0, b=0, c=0
```

---



*Variables contain an unknown, random value when declared. Variables must be initialized before they are used.*

## Array Initialization

### Formats

```
int MyArray[10] = {1,2,3,4,5,6,7,8,9,10}
char LetterArray[6] = 'MYWORD'
```

The initial values are written within the brackets {} and are divided by comma (.). These values are assigned in order from left to right starting from array index=0.

### Constants

A constant is a numeric or boolean value that does not change. Constants may be written as any of the following:

	Written As	Examples
Decimal constant	1234	short MyVal = 1234
Hexadecimal constant	0xFA20	short MyVal = 0xFA20
ASCII code (character constant)	'ABCD'	char String[4] = 'ABCD'
Boolean: True (not zero), False (zero)	True, 1, False, 0	bool Done = 0, or, bool Done = False

## Reserved Words

The following symbols and names are keywords that are reserved for use by macros. They cannot be used (as a complete name) in any function name, array name, or variable name. However, the reserved words may be contained within a variable name such as: my\_int, TheEnd, etc.

And	Else	next	Then
bool	End	not	To
Break	False	Or	True
BIN	float	return	void
BCD	For	select	wend
char	GetData	SetData	While
Continue	If	short	xor
down	int	Step	

## Operator

Below is a list of the recognized operators.

Group	Name	Sym
	Assignment:	=
Assignment operator:	Addition:	+
	Subtraction:	-
	Multiplication:	*
	Division:	/
	Modulo:	%
Comparison operators:	Less than:	<
	Less than or equal to:	<=
	Greater than:	>

	Greater than or equal to:	>=
	Is equal to:	==
	Not equal to:	<>
Logic operators:	Conditional 'AND':	AND
	Conditional 'OR':	OR
	Exclusive 'OR':	XOR
	Boolean 'NOT':	NOT
Bitwise and shift operators:	Left shift:	<<
	Right shift:	>>
	Bitwise 'AND':	&
	Bitwise 'OR':	
	Bitwise 'XOR':	^
	Bitwise complement:	~

### Order of Precedence

The process order of many operators within an expression is called the 'order of precedence'. Priority of the same kind of operator (From left to right, from top to bottom)

Arithmetic operator: ( \* , / ) ( % ) ( + , - )

Bitwise & Shift operators: From left to right within the expression

Comparison operator: From left to right within the expression

Logic operator: Not And Or Xor,

- Arithmetic operator is higher priority than a Bitwise operator
- Bitwise operator is higher priority than a Comparison operator
- Logic operator is higher priority than an Assignment operator

### Main Functions and Sub-functions

The macro must have one and only one "Macro\_Command main()" function which is the execution start point of any macro file. Any sub-functions must be pre-defined and written before the main() function.

The format is:

---

```
Macro_Command main( )
// The main macro code goes here
End Macro_Command
```

---

Any other sub functions must be declared before the main() function can use it, for example:

---

```
Sub SQR(int MyVar )
// sub routine code goes here
End Sub
Macro_Command main( )
// The main macro code goes here
Result = SQR(MyNum) //The sub function "SQR()" is
called
End macro command
```

---

### Local and Global Variables

Local variables exist only within the function in which they are defined.

Global variables exist throughout the entire macro, and are available for any function in the macro. NOTE: If a local variable has the same name as a global variable, the compiler will use the local variable in the function instead.



*Global variables are global only within the macro file in which they exist. It is not possible to share variables between different macro files. To share data between macro files, read/write the data to share into the HMI's Local Word (LW) and/or Local Bit (LB) addresses using the macro's GetData() and SetData()UM\_PLC functions.*

---

## Creating Variable Arrays

Arrays are fixed-depth and 1-dimensional only. Arrays can contain up to 65,535 elements. The **type** is the declaration of the array of elements, such as an array of *ints*, or *shorts*, or *bools*, etc. The **array\_size** is the number of elements to contain in the array, such as an array of [10] would contain 10 elements of the declared **type**. The first register in an array always begins with the 0<sup>th</sup> element and the last register is the declared number of elements minus 1.

The format is: `[type] Array_Name[Array_Size]`

For example:

---

```
char table[100]
```

---

The above declaration defines an array of 100 type 'char' registers named 'table'. The first element in table is address 0 and the last is 99, so it would be used as:

---

```
char FirstElement
char LastElement

FirstElement=table[0]
LastElement=table[99]
```

---

## Using Macros Within Recipes

The macro utility can easily use the recipe memory area. There are 65535 recipe word registers that are available to the user for general use (addresses RW:0 to RW:65535).

Register types: RW, RWI, and the pointer LW:9000 can be used to read or write the recipe memory area. 'RW' registers are used to directly address a recipe register. 'RWI' registers are used to indirectly address a recipe register by using the pointer in local word memory LW:9000. If indirect addressing is used, then the direct recipe address is determined by adding the value contained in LW:9000 to the address of the RWI register. For instance, to indirectly point to the direct recipe address RW:20, an example would be that the macro would store the value 15 into the LW:9000 pointer and then use an RWI address 5 (RWI:5) to GetData() or SetData() {LW:9000 = 15 + RWI:5 → RW:20}.



*The pointer has a maximum limitation of 32767, so LW:9000 can indirectly address from an offset of 0 to 32767. To indirectly address recipe registers above 32767, you will need to have the RWI address set to a higher address.*

---

The macro code on the next page will clear all general-use recipe memory area. Because of the pointer's limitation (max of LW:9000 = 32767), it breaks up the incrementing of memory into two sections, the lower half (0 to 29999) and upper half (30000 to 59999).

---

```
macro_command main()

    int Address = 0
    int Initial = 0
    short Done = 0
    short Section = 1

    // LW2 is used as an indicator
```



```

SetData(Section, "Local HMI", LW_BIN, 2, 1)

for Address = 0 to 29999
  // increment the address
  SetData(Address, "Local HMI", LW_BIN, 9000, 1)

  // write a 0 to the current address
  SetData(Initial, "Local HMI", RWI_BIN, 0, 1)
next address

// do the second half
Section = 2
SetData(Section, "Local HMI", LW_BIN, 2, 1)

// adjust actual address by using LW9000
for Address = 0 to 29999

  // increment address
  SetData(Address, "Local HMI", LW_BIN, 9000, 1)

  // write a 0 to the current address
  SetData(Initial, "Local HMI", RWI_BIN, 30000, 1)
next address

// turn off macro trigger, LB2
SetData(Initial, "Local HMI", LB_BIN, 2, 1)

end macro_command

```

## Statements, Conditions & Expressions

An *expression* combines constants, variables, arrays, functions, and operators under certain rules. Expressions can be logical or mathematical equations.

Below are listed some of the different styles of statements:

- A *declaration* (definition) statement is constructed like the following:

Format	Examples	Explanation
<i>type name</i>	short MyVar	This defines the <u>name</u> of the variable register and its <i>type</i>
<i>type name</i> [constant]	Short MyVars[10]	This defines an <u>array name</u> and the [constant] is the number of ' <i>type</i> elements' in the array

- An *assignment* statement will transfer (assign) an expression to a variable. The form is:

Format	Examples	Explanation
<i>variable = expression</i>	MyVar = 10 MyVar = x + y	This assigns the result of the expression to a variable

- *logic* or *conditional* statement processes a logic expression and branches depending on the result. This assigns the result of the expression to a variable. The ending statement '**End If**' is written as two words, not one word. Note that there is a difference between the conditional equality noted by two equal signs '=' (read as: 'is equal to') and the assignment equality noted by only one equal sign '='.

Format	Examples
<p><b>If Condition Then</b>                      [ Statements ]  <b>Else</b>                      [else statements]  <b>End If</b></p>	<pre>If x == 1 Then     y = y * 5     z = z - 12 Else     y = y * 2     z=y-1 End If</pre>
	<pre>If (x == 0 OR x == 1) AND y == 0 Then     z = 10 Else     z=5 End If</pre>

A **looping control** statement is used for looping a pre-determined number of times. For counting up, use the keyword **To**. For counting down, use the keyword **Down**. Counter Necessary. The counter of looping control. It can be integer o character.

Start

Necessary. The initial value of Counter.

End

Necessary. The end value of Counter.

Step

Options. The increment decrement step of Counter. It can be integer an can be omitted when value is 1.

Statements

- Optional. Statement block between For and Next which will be executed.

Format	Examples
	<pre>For n = 100 Down 0 Step 3     n = n * 2 Next n</pre>
<p>For counter = start To/Down end [Step steps ]                      [Statements]                      Next [counter]</p>	<pre>For loop = 0 To 20 Step 2     loop = (loop + 1) *2 Next loop</pre>

### While – Wend statement

Loop an unknown number of times. Looping is controlled by the Condition. When Condition is TRUE, the statements will be executed repetitively until the condition turns to FALSE.

Condition

Necessary. An expression which evaluates to TRUE or FALSE.

Statements

Optional. Statement block.

**break**

Used in looping control or select statement. It skips immediately to the end of the statement.

**continue**

Used in looping control statement. It quits the current iteration of a loop and starts the next one.

**return**

Returns to the function or module that called the current function.

Format	Examples
While [ <i>Condition</i> ] [Statements] Wend	<pre> While b [0] a[i] = 20 + i + * 10 If a[i ]== 120 Then c = 200 Break End If i = i + 1 Wend </pre>

## Boolean expressions

The value of Boolean expression is zero means FALSE.

The value of Boolean expression is not zero means TRUE.

## Function Calls and Passing Parameters

A sub-function must be defined before its execution and is therefore placed before the main() macro function. If there is an entry attempt into a function before it is first defined in the macro program will create a compiler error 'Function not defined'.

```

Sub int SQR(int x) //This is the sub-function that squares a
number
x = x * x
return x
End Sub
Macro_Command main( ) // This main function calls the SQR()
function.
int I = 5
I = SQR(I)
// Write the result to a LW
End Macro_Command

```

The above example code defines a sub-function called SQR(). This sub-function allows one variable to be passed into it. The function then takes the value passed to it, multiplies its value by itself and then stores the result back into the same variable. The result is returned back to the calling function.

## Reading & Writing External Registers in a Macro

A macro program can communicate with external PLC data registers and I/O. The function GetData() receives data from a PLC in the device table (or the OIT's local LW and LB memory). The function SetData() sends data to the PLC (or the OIT's local LW and LB memory).

The GetData() or the SetData() can be hand typed according to the format listed below, or the "PLC API" button located at the bottom-left of the macro editor dialog box allows the user to generate the GetData() or SetData() function code by selecting the appropriate responses within the PLC API dialog box. Be sure to declare all variables before using the 'PLC API' option. To set up the following functions, place the cursor after the declarations, and

then press the PLC API button. This way, all variables will be available in a drop-down menu list within the 'PLC API' dialog.

### GetData() Function

This function reads data from a register location in the OIT or PLC memory, and stores it into a variable.

The format for the GetData() function is:

**GetData**(DestVariable, PLC Name, Addr\_Type, Address, NumElements)

Parameter	Description	Example(s)
DestVariable	The name of the variable to store the data that is captured	MyVar, InputArray[0]
PLC Name	The name assigned to the controller in the device table	"Local HMI", "Modbus RTU"
Addr_Type	The device type and encoding method (binary or BCD) of the PLC data.	LW_Binplc, 4x_Binplc, 0x_Binaux, RW_Bcdplc
Address	The address offset in the PLC (must be written as a constant whole number – no decimals – no variables)	0, 100, (an address that has a decimal point is written here without the '.' So 1.03 is written as 103
NumElements	Number of elements to read/write (must be at least 1 and it must be a constant number) If the number of elements is more than 1, the 'DestVariable' parameter must be declared as an array.	1, 16, 100

### SetData() Function

**SetData**( DestData ,Addr\_Type , Address , NumElements )

Description: Writes data from a variable to a location in the PLC memory.

Parameter	Description	Example(S)
DestData	The name of the variable to retrieve the data from	
PLC Name	The Name assigned to the controller in the device table	"Local HMI", "Modbus RTU"
Addr_Type	The device type and encoding method (binary or BCD) of the PLC data	
Address	The address offset in the PLC	
NumElements	Number of words	

The number of registers actually read from or written to depends on both the *type* of the DestData variable and the *value* of Number of Elements. The Number of Elements refers to the number of elements of the DestData type, not necessarily the number of registers.

Type of DestData	Number of Elements	Actual Number of 16-bit Registers Read/Written
Float (32-bit)	1	2
Float (32-bit)	2	4
Int (32-bit)	1	2
Int (32-bit)	2	4
Short (16-bit)	1	1
Short (16-bit)	2	2

Char (8-bit)	1	1
Char (8-bit)	1	1
Bool (8-bit)	2	1
Bool (8-bit)	1	1

When a GetData() or SetData() function is executed using a 32-bit data type (Int or Float), the function will automatically convert the data. For example:

```
// get a floating point value from a modbus device
// note that two registers will actually be read,
// since a floating point value occupies 2 registers
macro_command main()
  float f
  GetData(f, "MODBUS", 4x_BIN, 101, 1) // f will contain a floating point
  value
end macro_command

// get (5) 32-bit integer values from the HMIs internal storage
// note that 10 registers will actually be read,
// since each integer value occupies 2 words
macro_command main()
  int Values[5]
  GetData(Values[0], "Local HMI", LW, 200, 5) // each element of
  Values[] will contain a 32-bit integer value

end macro_command
```

### Precautions, Tips & Tricks when Using Macros

- The size of a macro in an `xob` file is limited by the memory of the OIT.
- The maximum storage space of local variables in a macro is 4K bytes.
- A maximum of 256 macros are allowed in an EasyBuilder project.
- A macro may cause the OIT to lock up. Possible causes are:
  - A macro contains an infinite loop with no PLC communication.
  - The size of an array exceeds the storage space in a macro.
  - PLC communication time may cause the macro to execute slower than expected.
- Random macro compiler errors can often be corrected by recompiling again, or canceling out of the macro editor and then reopen and compile again.
- Some macros don't work the way they should due to the declaration of some variables and how they get interpreted. Some Boolean arrays work better when defined as an array of 'short's or 'int's. When getting data from an external register, the variable to receive it is best declared as a 'short' or an array of 'short's.

## Compiler Errors & Error Codes

When there are compile errors, the error description can be referenced by the compiler error message number.

Error message format: Macro\_name(: Error\_message\_number ) Error\_Message.

- (1): "Syntax error", "identifier" - There are many possibilities for the cause of this compiler error. Simply stated, the compiler found a problem with the syntax of the statement
- (2): Used without having been initialized. - Must define the size of an array during declaration. The array size declaration must be a constant whole number, not a variable.
- (3): "Redefinition error: " - The name of variable and function within its scope must be unique.

```
Macro_Command main( )
int g[10] , g //<- illegal - redefinition of 'g'
For g[2] = 0 To 2
```

- ```

    g[3] = 4
    Next g[2]
    End Macro_Command

```
- (4): "Function name error:" **'identifier'** - Reserved keywords and constant can not be the name of a function
 

```

Macro_Command If( ) //<- illegal - used 'If' in the
name

```
  - (5): "Statement missing"; Statement missing "(" or ")" - Some part of the Statement is missing, typically it is a missing parentheses
 

```

Macro_Command main ) //<- illegal - missing '('

```
  - (6): "Missing expression in 'If' statement"
  - (7): "Missing 'Then' in If statement"
  - (8): "Missing 'EndIf'";
  - (9): Unfinished "If" statement before "End If";
  - (10): "Illegal Else statement";
 

```

The format of "If" statement is:
If [logic expression]Then
[ Else [If [logic expression] Then ] ]
EndIf
Any format other than this format will cause compile
error.

```
  - (17): "For" statement error: missing "For" before "Next";
  - (18): "Should be integer of char variable";
  - (19): "Missing assign statement";
  - (20): "Missing keyword 'To'";
  - (21): "Missing 'Next' statement";
 

```

The format of "For" statement is:
For [variable] = [initial value] To [end value] [Step]
Next [variable]
Any format other than this format will cause compile error.

```
  - (22): "While" statement error: missing "While" before "Wend";
  - (23): "Missing 'Wend' statement";
 

```

The format of "While" statement is:
While [logic expression]
Wend
Any format other than this format will cause compile error.

```
  - (24): "Illegal 'Break' statement";
 

```

"Break" statement can only be used in "For", "While", or "Select
Case" statement
"Break" statement takes one line of Macro.

```
  - (25): "Illegal 'Continue' statement";
 

```

"Continue" statement can only be used in "For" statement, or "While"
statement
"Continue" statement takes one line of Macro.

```
  - (26): "Expression error";
  - (27): "Illegal operation object";
 

```

The mismatch of operation object in expression cause compile error.
For example :
Macro_Command main( )
int g[10]
For g[2] = 0 To 2
g[3] = 4 + XYZ <- illegal - XYZ is undefined
Next g[2]
End Macro_Command

```
  - (28): "Missing 'Sub'";
  - (29): "Missing 'Macro\_Command'";
 

```

The format of function declaration is: Sub(Macro_Command) [data type]
function_name( ... )

```

End Sub(Macro\_Command)

Any format other than this format will cause compile error.

- (30:) "Mismatch of the number of parameters " ;
- (31:) "Mismatch of data type of parameter" ;
- (32:) "Parameter error";
 

The parameters of a function must be equivalent to the arguments passing to a function to avoid compile error.
- (33:) "Undefined function" ;
- (34:) Illegal member of array
- (35:) Illegal definition of array";
- (36:) Illegal index of array
- (37:) "Undefined symbol" ;
 

Any variable or function should be declared before use.
- (38:) "Un-supported PLC data address" ;
 

The parameter of GetData( ... ) , SetData( ... ) should be legal PLC address.
- (39:) Should be integer, character 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 compile error.
- (40:) "Illegal Macro statement before declaration statement "

For example :

```
Macro_Command main( )
int g[10]
For g[2] = 0 To 2
g[3] = 4 + g[9]
int h , k <- illegal - definitions must occur
before any statements or expressions
Next g[2]
End Macro_Command
```

- (41:) "Floating point can not bitwise shift"
- (42:) "Missing function return value " ;
- (43:) "Function can not return a value"
- (44:) "Illegal Float data type in expression" ;
- (45:) "Error PLC address" ;
- (46:) "Stack can not exceed 4k bytes" ;
- (47:) "Only one main entrance in the Macro is allowed" ;
- (48:) "Too many main entrance: " **'identifier'**

The only one main entrance of Macro is:

```
Macro_Command function_name( )
End Macro_Command
```

## Chapter 16 Miscellaneous Objects

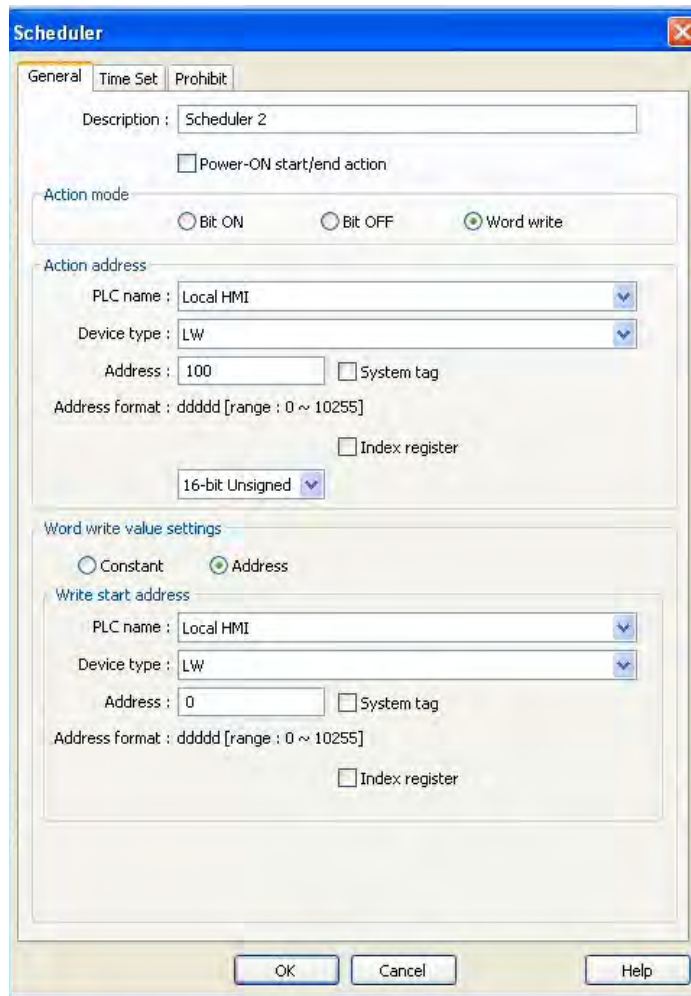
### Scheduler Object

The Scheduler Object takes advantage of the internal clock of the OIT (Note: the internal clock is not available on the HMI5043N). When used, this object continuously reads from the internal OIT clock. When a predefined time is reached, the OIT sets a bit or writes a constant value to a pre-defined OIT or PLC register. This feature can be very useful to start/stop processes that are supposed to occur on a time-scheduled basis.

#### To create a Scheduler Object:



1. From the Objects menu, click **Scheduler**. Or click the Scheduler icon in the Object 1 toolbar. The New Scheduler dialog box appears. Click the New button to add a new schedule:



The screenshot shows the 'Scheduler' dialog box with the 'General' tab selected. The 'Description' field contains 'Scheduler 2'. The 'Power-ON start/end action' checkbox is unchecked. Under 'Action mode', the 'Word write' radio button is selected. The 'Action address' section includes 'PLC name' (Local HMI), 'Device type' (LW), and 'Address' (100). The 'Address format' is 'dddd [range : 0 ~ 10255]'. The 'Word write value settings' section has 'Address' selected. The 'Write start address' section includes 'PLC name' (Local HMI), 'Device type' (LW), and 'Address' (0). The 'Address format' is 'dddd [range : 0 ~ 10255]'. The dialog box has 'OK', 'Cancel', and 'Help' buttons at the bottom.



2. **Power-ON start/end action:** enable this feature to write to the defined bit or register whenever the HMI starts or shuts down.

3. **Action mode frame:** Select type of action that is conducted when time trigger is due:

- a. Bit ON: set the selected bit.
- b. Bit OFF: turn off or clear the selected bit.
- c. Word write: write a value to the selected register.

4. **Action address:** this is the target OIT or PLC address that is written to when the selected time has passed.

Note: if End Action on the Time Set tab is enabled, the OIT will automatically use the next consecutive address to write the 'End' value.

a. PLC Name, etc.: see "General Properties" section at beginning of this chapter.

5. **Word write value settings frame:** If 'Word write' is selected in the Action mode frame, use this section to either select a constant value to be written to the target register or a register that will be read and the value copied into the target register.

a. Constant: check to write a constant value to the Action address:

Word write value settings

Constant  Address

Write start value : 70

Write end value : 30

b. Address: check to read the value from a OIT or PLC address, which is then used to write to target address:

Word write value settings

Constant  Address

Write start address

PLC name : Local HMI

Device type : LW

Address : 0  System tag

Address format : dddd [range : 0 ~ 10255]

Index register

Write end address

Address : 0 + 1

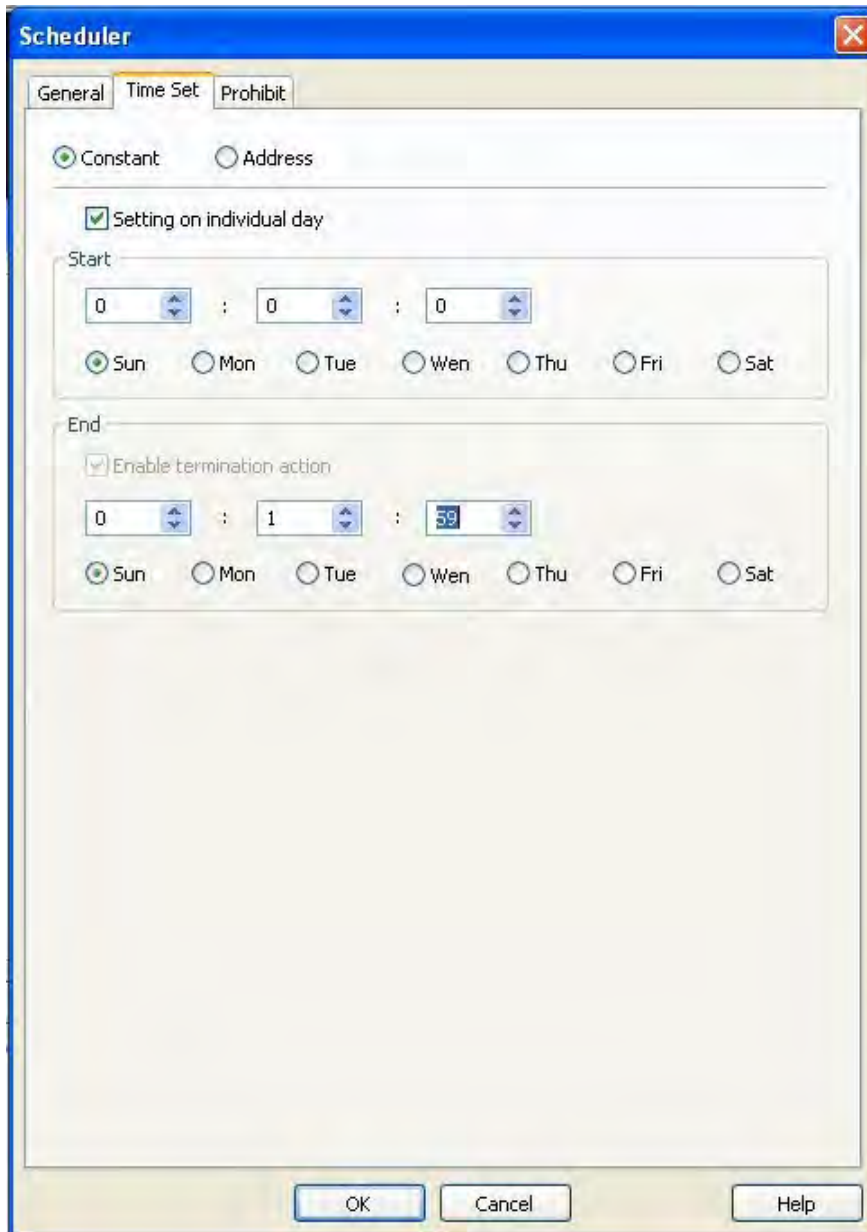
c. PLC Name, etc.: see "General Properties" section at beginning of this chapter.

*Please Note: Write end address [Address: 0+1] is the address used to read the 'End' value that is then sent to the second consecutive register that is listed under Action address frame.*

## The Time Set Tab

**Time Set tab:** determines the time trigger for the 'Start' and 'End' actions.

1. Constant or Address: used to determine if time trigger is based upon a predefined set time or is dynamically determined from reading a series of control registers.
- 2.. If Constant is selected, these are the options:



3. **Setting on individual day:** enable this option if you wish to configure a unique start time and/or end time for a particular day of the week. Disable this option, if the start time and/or end time is to be activated for multiple days of the week.

4. **Start frame:** Used to configure the time and days used for the 'start' time

- a. Hours setting: the first box represents the hour of the day. Range is 0 to 23.
- b. Minutes setting: the second box represents the minutes setting. Range is 0 to 59.
- c. Seconds setting: the third box represents the seconds setting. Range is 0 to 59.
- d. Days of the week: If 'Setting on individual day' is disabled, click on which days of the week you wish to perform the action based upon the Start time configured. Please note that any combination is allowed, (ex. Monday thru Friday; Monday, Wednesday, Friday; etc). If 'Setting on individual day' is enabled, then click the particular day of the week desired.

5. **End frame:** Used to configure the time and days used for the 'end' time:

- a. Enable termination action: If disabled, the 'end' time option is not used. If enabled, use the Hours, Minutes, and Seconds setting boxes to input a time at which the OIT will reset the target bit or write the secondary value configured on the General tab.

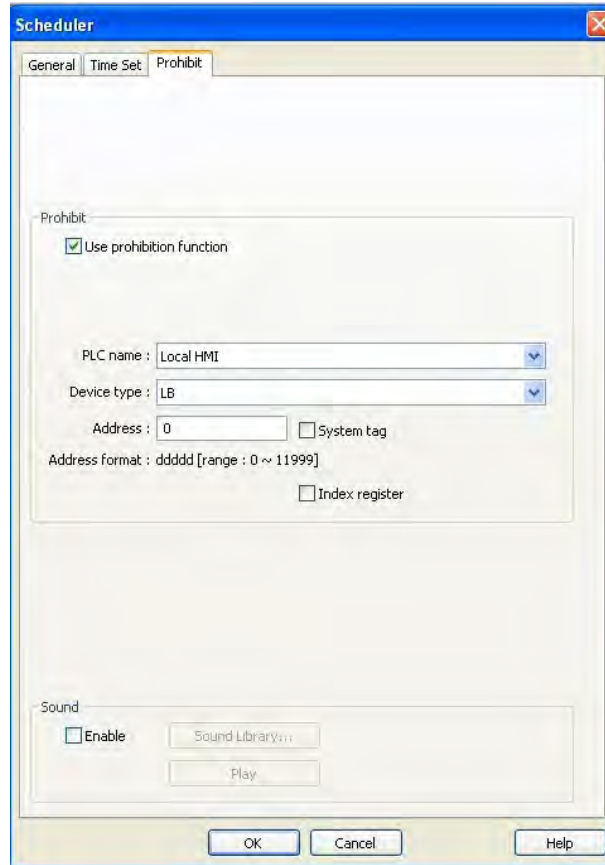
*Please Note: this option is not available, if "Setting on individual day" is enabled.*

- b. Hours setting: the first box represents the hour of the day. Range is 0 to 23.
- c. Minutes setting: the second box represents the minutes setting. Range is 0 to 59.
- d. Seconds setting: the third box represents the seconds setting. Range is 0 to 59.
- e. Days of the week: If 'Setting on individual day' is enabled, click on which days of the week you wish to perform the action based upon the End time configured.

*Please note that any combination is allowed, (ex. Start time is on Monday, End time is on Friday etc).*

## The Prohibit Tab

**Prohibit tab:** The prohibit tab is used to enable the ‘Prohibit’ feature or to enable an output sound if external speakers are used:



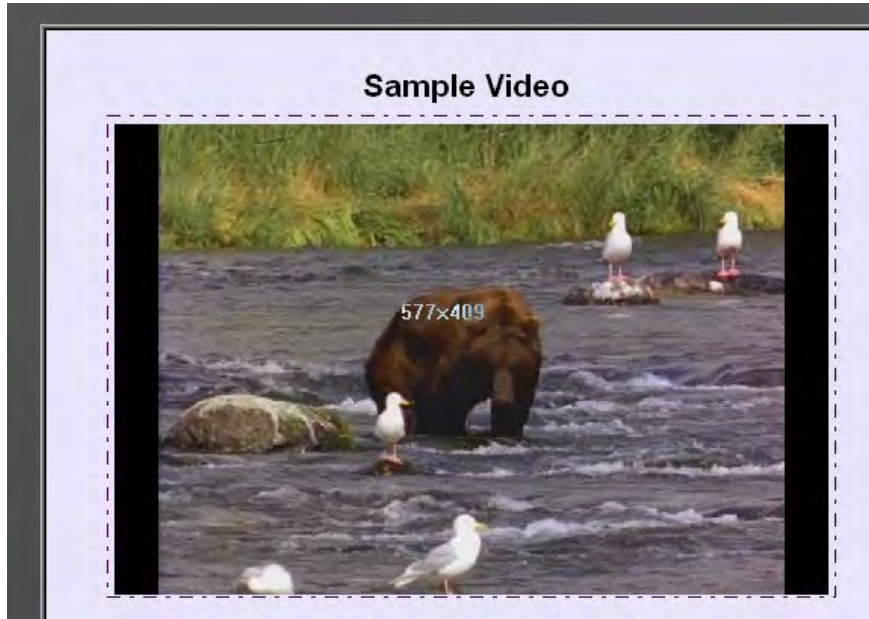
1. **Prohibit frame:** used to enable the ‘Prohibit’ feature. This feature checks to see if an assigned bit is set or ON before performing the scheduler function at the preset time. If the bit is ON, the scheduler function is not performed.

- a. Use prohibition function: check to enable the Prohibit feature.
- b. PLC name, etc: see “General Properties” section at beginning of this chapter.

2. **Sound frame:** If external speakers are connected, you can enable this feature to output an audio sound wave whenever the scheduler function is performed by the OIT.

## Media Player

The Media Player object is used to view video files on the OIT (this feature is available on the 'X' models only). Video files are most commonly used in applications for instructional or tutorial information or as a means of displaying visual content for purposes of advertising, etc.:



The OIT reads from the video file immediately after the screen that contains the Media Player object is displayed.

*Please Note: When using the Media Player object for the first time, you must download the project to the OIT via the Ethernet port. By doing this, EZWare-5000 will install the necessary Media Player drivers during the download.*

Besides simply displaying video files, the Media Player object has several controls which allow the operator to perform many operations on the file that is viewed including:

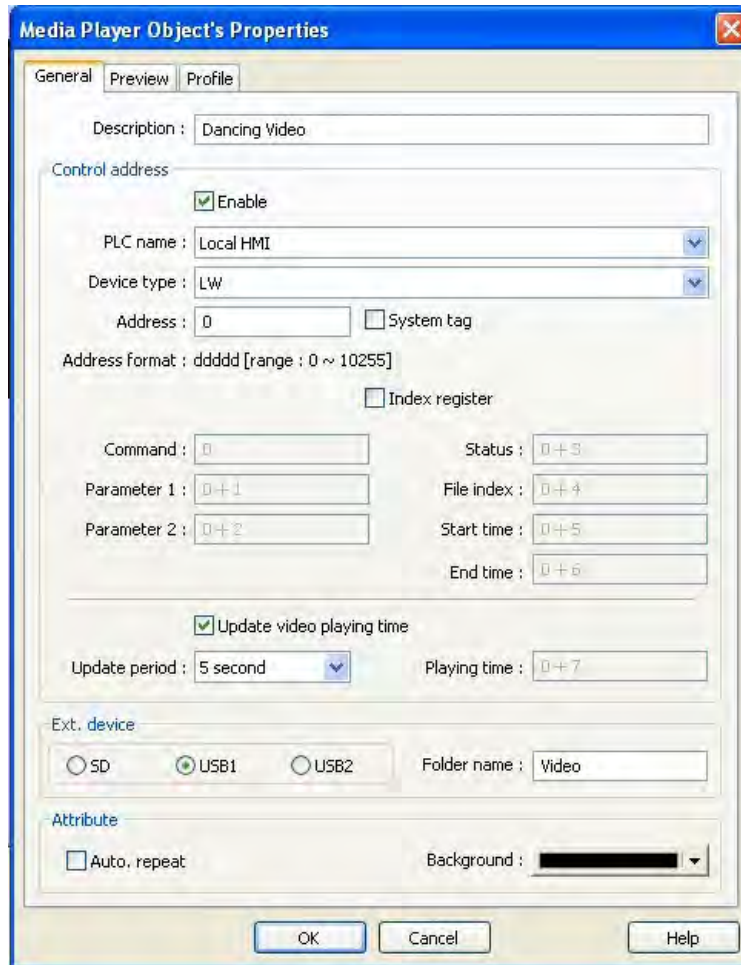
Seeking files

Zoom feature

Volume adjustment

## To create a Media Player Object:

1. From the Objects menu, click Media Player. Or click the Media Player icon in the Object 2 toolbar. The New Media Player dialog box appears:



2. Description: provides the OIT programmer with a description about this particular object for future reference.
3. Control Address frame: this feature is used to provide some control features to the media player.
  - a. Enable: check to enable the Control Address feature. This allows manual control of the Media Player object using the addresses listed below. If disabled, the Media Player will automatically play the first video in the designated folder when the screen is displayed.
  - b. PLC name, Device type, Address: see “General Properties” section at beginning of this chapter.
  - c. Command: 0 – register used to control the operation of the Media Player. In this example, LW0 is the command register. The following values will result in the following:
    - i. Value =0: **No operation.** Default value.

- 
- ii. Value =1: **Play the index file.** In this mode, use Parameter 1 to select the file index. Parameter 2 is not used. Files are sorted based upon the filename in alphabetical order. Ex. If File Index =0, then the first video file is played.
  - iii. Value =2: **Play Previous file.** In this mode, Parameter 1 and Parameter 2 are not used, so set to 0. If the File Index is set to 0, then the Media Player replays the same video from the start.
  - iv. Value =3: **Play Next file.** In this mode, Parameter 1 and Parameter 2 are not used, so set to 0. If the last video file is played, the Media Player will play the first (Index 0) file.
  - v. Value =4: **Pauses the video.** Enter 4 again to restart the video.
  - vi. Value =5: **Stops the video** that is currently playing and closes the file.
  - vii. Value =6: **Starts playing the video file at a designated target location** in the file. In this mode, use Parameter 1 to select the target location (units in seconds). Note: If the value in Parameter 1 is larger than the end time of the video file, then the video will play the final second of the file.
  - viii. Value =7: **Fast Forward** file. In this mode, use Parameter 1 to select the number of seconds to advance from the current location.
  - ix. Value =8: **Rewind** file. In this mode, use Parameter 1 to select the number of seconds to rewind from the current location.
  - x. Value =9: **Adjust volume.** Use Parameter 1 to determine volume level. Range is 0 – 128. Note: Default volume level is 128.
  - xi. Value =10: **Set Video Display Size.** Use Parameter 1 to determine the relative display size. Magnification has sixteen levels:
    - 1. Level 1 is 25% of normal
    - 2. Level 2 is 50% of normal
    - 3. Level 3 is 75% of normal
    - 4. Level 4 is 100% of normal
    - 5. Level 5 is 125% of normal
    - 6. Level 6 is 150% of normal
    - 7. Level 7 is 175% of normal
    - 8. Level 8 is 200% of normal
    - 9. Level 9 is 225% of normal
    - 10. Level 10 is 250% of normal
    - 11. Level 11 is 275% of normal
    - 12. Level 12 is 300% of normal
    - 13. Level 13 is 325% of normal
    - 14. Level 14 is 350% of normal
    - 15. Level 15 is 375% of normal
    - 16. Level 16 is 400% of normal  
Note Default size is 4 (100%).
- d. **Parameter 1:** 0+1 – depending upon the value in the Command register, this register may be used to provide additional parameters. In this example, LW1 is the parameter 1 register.

- e. **Parameter 2:** 0+2 – depending upon the value in the Command register, this register may be used to provide additional parameters. In this example, LW2 is the parameter 2 register.
- f. **Status:** 0+3 – this Read Only register is used to provide status on the video that is playing. In this example, LW3 is the status register.
  - i. Bit 0 – File Open (0=closed, 1=open)
  - ii. Bit 1 – File Playing (0=closed, 1=playing)
  - iii. Bit 8 – Command Error (0=correct, 1= command/parameter error) For example, this bit is set if the media player object is unable to locate a playable file in the designated folder.
  - iv. Bit 9 – File Error (0=file format and load correct, 1=file format or loading error)
- g. **File index:** 0+4 – this Read Only register provides the file index on the video that is playing. In this example, LW4 is the file index register. File index is used to indicate the order of the video files being displayed.
- h. **Start time:** 0+5 – this Read Only register is used to provide the start time (always 0) on the video that is currently playing. Units are in seconds. In this example, LW5 is the start time register.
- i. **End time:** 0+6 – this Read Only register is used to provide the total time length of the video that is currently playing. Units are in seconds. In this example, LW6 is the end time register.

4. **Update video playing time:** enable this feature to display the current running (elapsed) time of the video that is playing.

- a. **Update period:** select how often the OIT updates the running time. Range is 1-60 seconds.
- b. **Playing time:** 0+7 – this indicates which register is used to display the video playing time based upon an offset from the Command register. The register used is set to the Command address + 7. In this example, LW7 is the update video play time.

5. **Ex. Device frame:** Determines the location of the video files that are played.

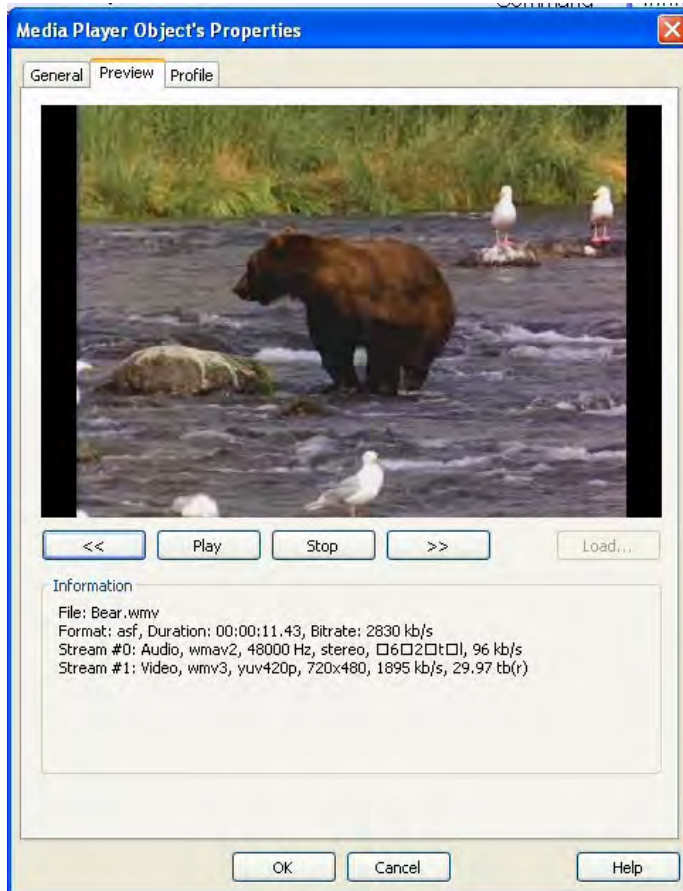
- a. **SD:** the video files are located on a Secure Digital (SD) card
- b. **USB1:** located on the USB1 port. Note: USB1 is not a specific location on the OIT. Rather, this is the first USB port on the OIT that is used.
- c. **USB2:** located on the USB2 port.
- d. **Folder name:** This is the folder on the SD or USB Flash Drive. Note: You must create a folder to store the video files- do not use the root directory of the target solid state drive. The folder name must contain characters only (character symbols such as \*, ?, <, >, are not allowed).

6. **Attribute frame:** determines additional attributes of the Media Player object.

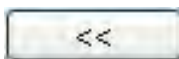
- a. **Auto. repeat:** enable if you want the Media Player object to automatically play the next video file. Note: after the last video file, the Media Player will start over with the first video file.
- b. **Background:** color for the background of the object.



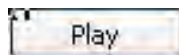
**Preview tab: used to preview a video file:**



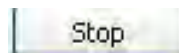
1. Rewind button: click to rewind the video by 1 minute.



2. Play button: click to play the video. Note: if the video does not start playing, then the OIT does not support that format.



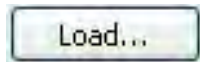
3. Stop button: click to stop playing the video.



4. Fast Forward button: click to fast-forward the video by 1 minute.



5. Load button: click to select which video to play.



6. Information screen: this provides detailed information about the particular video file being previewed.

- a. **File:** name of video file
- b. **Format:** type of video file, duration and bit rate.

**Important Note:** File formats supported are **AVI, WMV, MOV, MPEG4, FLV, and XVID.**

- c. **Stream #0:** Audio file information and type
- d. **Stream #1:** Video file information and type

### Regarding File Size:

There is no limitation to the size of the video file except for the size of the storage device (SD or USB flash drive).

On some video files, you may notice that the file will not play or may have a lag time due to the limitations of the OIT's processor speed and memory. Therefore, we recommend a file resolution no greater than 640x480 pixels and a bit rate that is less than 700 KB per second.

### To play a video

When displaying an OIT screen that has a Media Player object on it, perform the following steps to begin playing the video:

Set the value in the Parameter 1 register to select the correct video. Parameter 1 is used to select the File Index number. The video files are organized and assigned a file index number according to alphabetical order. The first file is assigned File Index 0, second file is File Index 1, etc.

Set the value in the Command Register to 1. The OIT reads the video files and starts playing the specified file.

The OIT will reset the Command Register to 0 and play the video until the end of the file. Note: during 'play' do not change the values in the Command register or Parameter 1/Parameter 2 register as this may cause erratic results.

### To stop a video

To stop playing a video that is currently running on a Media Player object:

Set the value in the Command Register to 5. The OIT stops reading the video file and displaying the movie. The OIT closes the video file.

After closing the video file, the OIT will reset the Command Register to 0. Note: during this sequence do not change the values in the Command register or Parameter 1/Parameter 2 register as this may cause erratic results.

**Please Use This Page For Your Notes:**

# Index

## A

- Adjust volume . . . . . 298
- Alarm Bar Object, using . . . . . 267
- Alarm Display Object, using . . . . . 266
- Alarm/Event Log . . . . . 263
- Alarms, creating . . . . . 263
- align bottom . . . . . 95
- align horizontal center . . . . . 95
- align left . . . . . 93
- align right . . . . . 94
- align top . . . . . 94
- align vertical center . . . . . 96
- Alignment tools . . . . . 93
- Always ON bits. . . . . 187
- Animation Object. . . . . 222
- Animation Object, using . . . . . 222
- Arc tool, using . . . . . 142
- ASCII Data Object, using . . . . . 216
- ASCII Display Object . . . . . 216
- ASCII Input Object. . . . . 217
- ASCII Input Object, using . . . . . 217
- Attributes, changing . . . . . 89
- Automatic toggling feature (periodical toggle). . . 196

## B

- background pattern . . . . . 123
- backlight settings . . . . . 21
- Backup Part. . . . . 274
- Bar Graphs . . . . . 246
- Bar Graphs, creating. . . . . 246,258
- Base windows . . . . . 123
- Bit Lamp Object . . . . . 193
- Bit Lamp Object, using. . . . . 192
- Bitmap Libraries, deleting . . . . . 156
- Bitmap Libraries, using. . . . . 154
- Bitmap Object, using . . . . . 149
- Bitmaps, exporting . . . . . 156

- Blinking feature . . . . . 193
- Boolean expressions . . . . . 286

## C

- Cable Routing and Noise Immunity . . . . . 10
- calibration . . . . . 17
- CE Compliance . . . . . 7
- change state . . . . . 86
- Circle tool, using . . . . . 141
- Closing projects . . . . . 81
- Closing windows or screens . . . . . 117
- COM ports . . . . . 25
- Common windows, above/below the base screen feature 130
- Common windows, changing. . . . . 132
- Common windows, using. . . . . 129
- Communications Settings . . . . . 62
- CompactFlash settings . . . . . 22
- CompactFlash, transferring projects . . . . . 23
- Compiling . . . . . 56
- compress . . . . . 81
- compress feature . . . . . 81
- Configuration Wiring . . . . . 14
- Connecting HMI to Computer . . . . . 35
- Control Address frame . . . . . 233
- Control panel grounding . . . . . 9
- Control panel, requirements. . . . . 8
- Control word address. . . . . 233
- Control Word Address . . . . . 232
- Copy command . . . . . 87
- Create a Keypad . . . . . 240
- create a new project . . . . . 78
- create an Animation Object. . . . . 222
- Creating a new window . . . . . 117
- CSV. . . . . 165

## D

- Data Block Display . . . . . 230,232

|                                                           |     |                                                   |     |
|-----------------------------------------------------------|-----|---------------------------------------------------|-----|
| Data Block Display Object . . . . .                       | 232 | Fast Selection windows, changing . . . . .        | 136 |
| Data Format tab. . . . .                                  | 261 | Fast Selection windows, using . . . . .           | 133 |
| data memory . . . . .                                     | 184 | firmware version. . . . .                         | 23  |
| Data Memory of OIT . . . . .                              | 184 | Fix Objects. . . . .                              | 85  |
| Data Sampling Objects . . . . .                           | 251 | Flipping tools, using . . . . .                   | 99  |
| Data storage start . . . . .                              | 233 | font size, changing . . . . .                     | 145 |
| data transfer object . . . . .                            | 271 | free line tool . . . . .                          | 143 |
| Data Transfer Object, using . . . . .                     | 271 | Full screen windows, displaying . . . . .         | 124 |
| date settings . . . . .                                   | 19  | function key . . . . .                            | 51  |
| decrement key . . . . .                                   | 46  | Function Key Object, creating keypads . . . . .   | 241 |
| Deleting a part or object . . . . .                       | 89  | Function Key Object, displaying windows. . . . .  | 124 |
| DIP switches. . . . .                                     | 17  | <b>G</b>                                          |     |
| Direct Window Object . . . . .                            | 126 | General Settings . . . . .                        | 106 |
| Display Area tab . . . . .                                | 234 | GIF . . . . .                                     | 146 |
| Display Meter Object, using . . . . .                     | 248 | Graphics Object . . . . .                         | 3   |
| Display Options . . . . .                                 | 83  | Grid frame . . . . .                              | 234 |
| Drawing tools, using . . . . .                            | 140 | Grid function. . . . .                            | 84  |
| <b>E</b>                                                  |     | grid size . . . . .                               | 84  |
| EasyBuilder Application. . . . .                          | 76  | Group Libraries, definition . . . . .             | 157 |
| Editing Commands. . . . .                                 | 86  | Group Libraries, deleting a group member. . . . . | 160 |
| Electrostatic Discharge . . . . .                         | 7   | Grouping objects. . . . .                         | 90  |
| Ellipse/Circle tool, using . . . . .                      | 141 | <b>H</b>                                          |     |
| End frame. . . . .                                        | 294 | history . . . . .                                 | 20  |
| Event Display. . . . .                                    | 269 | History Data Display . . . . .                    | 258 |
| Event Display Object . . . . .                            | 269 | <b>I</b>                                          |     |
| Event Display Object, using . . . . .                     | 269 | increment key . . . . .                           | 44  |
| Event Log Object, using . . . . .                         | 269 | Indirect Window Object . . . . .                  | 126 |
| Events, definition. . . . .                               | 269 | Input illegal frame . . . . .                     | 208 |
| exit EasyBuilder . . . . .                                | 82  | Installing the OIT . . . . .                      | 10  |
| Exiting EasyBuilder . . . . .                             | 82  | internal data memory . . . . .                    | 184 |
| extend memory . . . . .                                   | 110 | Internal OIT Memory. . . . .                      | 184 |
| extract a compressed project file. . . . .                | 82  | <b>J</b>                                          |     |
| EZware-500 configuration software, installation . . . . . | 35  | Jog feature . . . . .                             | 203 |
| EZware-500 software . . . . .                             | 61  | JPG . . . . .                                     | 146 |
| <b>F</b>                                                  |     | <b>K</b>                                          |     |
| factory configuration. . . . .                            | 17  | keypad entry screen. . . . .                      | 242 |
| Fast Forward . . . . .                                    | 298 | Keypad, creating . . . . .                        | 240 |
| Fast Selection key . . . . .                              | 135 |                                                   |     |

- Keypad, displaying and using . . . . . 242
- Keypads . . . . . 240
- L**
- Labeling an object . . . . . 195
- languages . . . . . 170
- Languages . . . . . 164
- Layering tools . . . . . 91
- LED Indicators . . . . . 4
- Libraries, Graphics using . . . . . 150
- Line tools, using . . . . . 140
- linked line tool . . . . . 144
- local setup . . . . . 17
- Lock . . . . . 100
- M**
- Macro frame . . . . . 173
- Macros . . . . . 277
- Macros, Array initialization . . . . . 281
- Macros, Memory Usage . . . . . 279
- Macros, using . . . . . 277,291
- Macros, using with recipes . . . . . 283
- Macros, Variable Declarations . . . . . 280
- Macros, Variable Initialization . . . . . 280
- Mapping tab . . . . . 237
- Mapping table . . . . . 238
- Media Player . . . . . 296
- meter . . . . . 42
- meter display. . . . . 42
- Meter Displays . . . . . 248
- Meters . . . . . 246
- Miscellaneous Objects . . . . . 292
- Monitoring Events . . . . . 269
- monopoly . . . . . 120
- Monopoly feature. . . . . 120
- Moving a part or object . . . . . 89
- Moving Shape Object . . . . . 218,219
- Moving Shape Object, scaling feature . . . . . 222
- Moving Shape Object, using . . . . . 218
- Multi-copy command, using . . . . . 87
- Multiple copies of object on screen, making . . . . . 87
- MultiState Switch Object . . . . . 205
- N**
- NEMA rating . . . . . 7
- NEMA Rating. . . . . 7
- network information . . . . . 22
- network settings . . . . . 18
- Noise Immunity . . . . . 10
- Notification frame . . . . . 173,239
- Nudge tools, using . . . . . 93
- Numeric Data Object, using . . . . . 209
- Numeric Display Object . . . . . 209
- Numeric Input Object . . . . . 211
- Numeric Input Object, using . . . . . 211
- Numeric Input register . . . . . 243
- numeric register . . . . . 43
- O**
- Object . . . . . 3
- Object Attributes, viewing. . . . . 89
- object ID tags . . . . . 85
- Object ID tags . . . . . 85
- Objects . . . . . 172
- Objects, creating & editing . . . . . 82
- off-line simulation mode. . . . . 59
- Offset to start address. . . . . 233
- OIT (Operator Interface Terminal) . . . . . 2
- on-line simulation mode . . . . . 60
- Opening projects. . . . . 81
- Opening windows or screens . . . . . 116
- Operating Modes. . . . . 63
- operating temperature . . . . . 7
- Option List . . . . . 236
- Option List icon . . . . . 237
- Option List Object . . . . . 236
- P**
- Panel Preparation . . . . . 12
- Pass-through mode. . . . . 76
- Pauses the video. . . . . 298

|                                               |       |                                     |     |
|-----------------------------------------------|-------|-------------------------------------|-----|
| picture library . . . . .                     | 154   | RnI Device . . . . .                | 275 |
| pictures . . . . .                            | 146   | Rotation tools, using . . . . .     | 100 |
| pie tool . . . . .                            | 143   | <b>S</b>                            |     |
| Play Next file . . . . .                      | 298   | Safety Precautions. . . . .         | 8   |
| Play Previous file. . . . .                   | 298   | Same Size tools . . . . .           | 97  |
| Play the index file . . . . .                 | 298   | Sample Project . . . . .            | 36  |
| PLC block pack feature. . . . .               | 111   | Saving . . . . .                    | 56  |
| PLC Coil Registers . . . . .                  | 192   | Saving Projects . . . . .           | 81  |
| PLC Control Object . . . . .                  | 125   | Scale lines tool, using . . . . .   | 142 |
| PLC, displaying windows . . . . .             | 124   | Security tab . . . . .              | 174 |
| Polygon tool, using . . . . .                 | 142   | select all objects . . . . .        | 87  |
| Popup windows, displaying . . . . .           | 124   | select next object. . . . .         | 86  |
| Port wiring . . . . .                         | 11    | Selecting graphic objects . . . . . | 86  |
| Power supply requirements . . . . .           | 10    | serial ports. . . . .               | 4   |
| Power-ON start/end action . . . . .           | 292   | Set Bit Object. . . . .             | 196 |
| Power-ON start/end action:. . . . .           | 292   | Set Bit Object, using . . . . .     | 196 |
| Printer server . . . . .                      | 111   | Set Video Display Size. . . . .     | 298 |
| Programming the OIT, wiring . . . . .         | 14    | Set Word Object . . . . .           | 201 |
| Prohibit frame . . . . .                      | 294   | Set Word Object, using. . . . .     | 201 |
| Prohibit tab . . . . .                        | 294   | Shape libraries, using. . . . .     | 151 |
| Prohibit Tab . . . . .                        | 295   | Shape Object, using . . . . .       | 147 |
| Project. . . . .                              | 3,6   | Shape tab . . . . .                 | 234 |
| Project, creating . . . . .                   | 35    | shapes. . . . .                     | 146 |
| ProjectManager. . . . .                       | 35,61 | Simulation Mode. . . . .            | 59  |
| Projects, Editing . . . . .                   | 78    | Simulation Screen . . . . .         | 59  |
| Projects, opening. . . . .                    | 78    | simulation settings . . . . .       | 75  |
| Pulsing a bit, using Set Bit Object . . . . . | 196   | Simulation Settings . . . . .       | 75  |
| <b>R</b>                                      |       | Slider . . . . .                    | 4   |
| Recipe . . . . .                              | 273   | Slider Object . . . . .             | 203 |
| Recipe Devices . . . . .                      | 273   | Snap Option . . . . .               | 85  |
| Recipe/Extended Memory Editor . . . . .       | 71    | Sound frame . . . . .               | 295 |
| Rectangle tool, using . . . . .               | 140   | sound library . . . . .             | 160 |
| Redo command . . . . .                        | 87    | Start frame . . . . .               | 295 |
| Reserved local bits . . . . .                 | 187   | Starting EZware500 . . . . .        | 35  |
| reset button. . . . .                         | 17    | Startup Window, creating . . . . .  | 48  |
| Resizing a part or object . . . . .           | 89    | States, definition of . . . . .     | 150 |
| Rewind . . . . .                              | 298   | Stops the video . . . . .           | 297 |
| Rn Device . . . . .                           | 275   | system information. . . . .         | 22  |

- system message window . . . . . 138
- System Parameters . . . . . 36
- system requirements . . . . . 35
- system settings, changing . . . . . 18
- T**
- Tag Databases . . . . . 164
- tag databases, label library . . . . . 169
- Tag databases, label library. . . . . 166
- Tag Databases, tag library . . . . . 164
- Tag databases, using the tag library . . . . . 166
- tag library. . . . . 164
- Tag Library, Import and export. . . . . 165
- Text Box Object, using . . . . . 145
- Time descending . . . . . 260
- Time Set Tab . . . . . 293
- time settings . . . . . 19
- Toggle Switch Object . . . . . 197
- Toggle Switch Object, using . . . . . 197
- Transfer Settings . . . . . 75
- Trend Display Object. . . . . 254
- Trend Display Object, using . . . . . 254
- Trend Displays . . . . . 251
- Trends . . . . . 246
- U**
- Underlay Windows
- assigning 121
  - assigning to a base window 122
  - how to display 122
  - rules that apply to 122
- Undo command . . . . . 87
- USB jump drive . . . . . 63
- USB, transferring projects . . . . . 23
- W**
- WAV files . . . . . 160
- Window Copy . . . . . 88
- Window frames, creating. . . . . 122
- Window Name . . . . . 119
- Window No. Treebar, using to navigate. . . . . 83
- Window, definition of. . . . . 3
- Window, deleting. . . . . 123
- Window, fundamental settings . . . . . 119
- Window, selecting a new background color . . . . 122
- Windows, how to display. . . . . 123
- Word Lamp Object . . . . . 200
- Word write . . . . . 292
- X**
- XY Plot Object . . . . . 224
- Z**
- zoom in . . . . . 85



**Please Use This Page For Your Notes:**

**Please Use This Page For Your Notes:**

**Please Use This Page For Your Notes:**