Installation & Operations Manual
MMI-320/640-C

Easy Builder Version 2.7.0

ORMEC Systems Corp.
19 Linden Park
Rochester, NY 14625
(585) 385-3520
June 23, 2006

About this Manual

This document is based on information available at the time of its publication. While efforts have been made to be accurate, the information in this document may not cover all details or variations in hardware or software, nor does it provide for every possibility concerning installation, operation, or maintenance. Features may be described herein which are not present in all hardware. ORMEC Systems Corp assumes no obligation of notice to holders of this document with respect to subsequent changes.

Information in this document is subject to change without notice and does not represent a commitment on the part of ORMEC Systems Corp.

Text Construction

Construction	Comment
Bold Characters	Keywords that are menu or system items or text that is part of EasyBuilder. Example: **BitLamp**
Italic Characters	Italics are used to draw emphasis to a portion of text. Example: "***Note:***"
[characters]	Characters within the [ ] are to be typed in exactly as they are printed. Example: "[c:\eb500\drivers]"
< Characters>	Indicates user information is required. A description of the information is between the < > characters. Example: "<project name>"
Key1+Key2	Press and hold Key1 then press Key2 (then release both keys). Example: "**Ctrl** + **N**" Hold Control down while pressing the N. (starts a new project)
Click	Tap the primary mouse button once on the indicated object. Example: "**Click** the **OK** button."
Double Click	Tap the primary mouse button twice on the indicated object. Example: "**Double Click** to call up the Part Attributes."
Right Click	Tap the secondary mouse button once. Example: "**Right Click** on the highlighted object to open the shortcut menu."
CAPITAL Characters	All capitals are used for Directory names, file names, and acronyms. Example: "EB500"
Menu|Menu item	This construction is used to specify menu commands. The main menu is on the left with submenus after it separated by the "**|**" character. Example: "**Edit|Align|Left**"

The abbreviations **MMI** and **HMI** are used interchangeably in this manual both refer to the touchscreen interface units.

Safety Instructions

Overview

This section states the safety instructions which must be followed when installing, operating and servicing the MMI. If neglected, physical injury and death may follow, or damage may occur to controller and related equipment. The material in this chapter must be studied before attempting any work on, or with, the unit.

Warnings and Notes

This manual distinguishes safety instructions. Warnings are used to inform of conditions, which can, lead if proper steps are not taken, to a serious fault condition, physical injury or death. Notes are used when the reader is required to pay special attention or when there is additional information available on the subject. Notes are less crucial than warnings, but should not be disregarded. Readers are notified of the need for special attention or additional in formation available on the subject with the following symbols:

Warnings

Readers are informed of situations that can result in serious physical injury and/or serious damage to equipment with the symbol shown to the left. A Warning symbol indicates that the reader should pay special attention to the accompanying text. Take precautionary steps to insure that the installation complies with warnings before continuing. Warnings include hazardous conditions that could cause personal injury or equipment damage if care is not taken. The text next to this symbol describes ways to avoid the danger.

Warnings

Dangerous Voltage Warnings: Warns of situations in which high voltage can cause physical injury and or damage equipment.
General warning: Warns of situations, which can cause physical injury and or damage equipment by means other than electrical.
Electrostatic Discharge Warning: Warns of situations in which an electrostatic discharge can damage equipment.

CAUTION!

Caution: Aims to draw special attention to the text. Be sure to understand the implications of the text before proceeding.

Note:

Note: gives additional information or points out more information available on the subject.

APPLICATIONS ASSISTANCE

This manual is designed to provide the necessary information for trouble-free installation and operation of HMI Touchscreens. Additional help is available when accessing the help functions in the EasyBuilder software. If further assistance is needed, please call ORMEC at 1-585-385-3520 or visit our web site at www.ormec.com.

IMPORTANT NOTE!

The abbreviation " **PLC** " (Programmable Logic Controller) will be used in this manual to indicate ether an ORMEC
ORION Motion Controller or
ServoWire Motion & Logic Controller (SMLC).

Backup project files (*.epj) as needed to insure the ability to modify a project in the future!

PROPRIETARY NOTICE

The information contained in this publication is derived in part from proprietary and patent data. This information has been prepared for the expressed purpose of assisting operating and maintenance personnel in the efficient use of the instrument described herein. Publication of this information does not convey any rights to use or reproduce or to use it for any purpose other than in connection with the installation, operation and maintenance of the equipment described herein.

We hope you will be pleased with our product. If you have any questions concerning our warranty, repair, modification or returned goods process, please contact your local distributor.

WARRANTY

This product is warrantied against defects in materials and workmanship for a period of twelve months from the date of shipment to Buyer.

The Warranty is limited to repair or replacement of the defective unit at the option of the manufacturer. This warranty is void if the product has been altered, misused, dismantled, or otherwise abused.

ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, ARE EXCLUDED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Section 1: Installation and Startup Guide

1.0 Getting Started

1.1 What is the MMI-320/640 Touchscreen Series?

MMI-320/640 Series

These are small touch screen interfaces for controllers. They display pictorial information, data and messages that are preloaded into them with a Personal Computer. Touch screen areas can be programmed to perform various functions.

Equipment Checklist

An MMI-320/640 system should include:

- An MMI-320/640 Series base unit
- An interface cable to the controller
- A personal computer with EasyBuilder programming software. A PC to MMI-320/640 cable is required.
- A 24VDC power supply (user provided)

Designed For Use

Provides a convenient way for a machine operator to:

- View machine status and parameters.
- Change machine status or applicable operating parameters of the machine.
- Maintain the running of the machine.

It also gives enhanced capabilities to the machine through:

- Printer output
- Direct touch screen interface
- Visually displayed prompts

2.0 Installation Instructions
2.1 Mounting Instructions

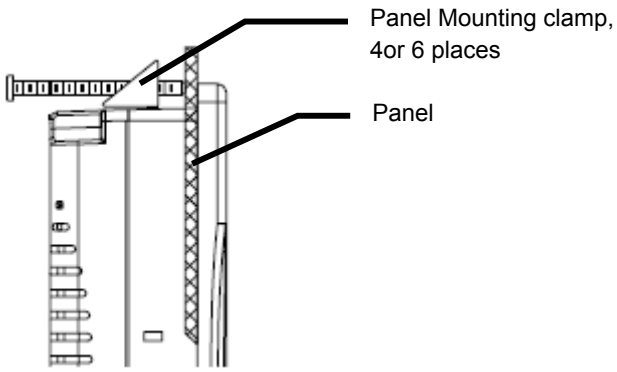
2.1.1 Location Considerations
Care should be taken when locating equipment behind the unit to ensure that AC power wiring, PLC output modules, contactors, starters and relays, and any other source of electrical interference are located away from the back of the unit.

Particular note should be taken to the position of variable speed drives and switching power supplies. Their input and load cables should be screened to a central star earth point.

2.1.2 Making a NEMA-4 Mounting

Panel Details The unit can be mounted into panels with a depth of 4”(105mm). It is recommended that the unit be mounted on the front panel of a steel enclosure, through an appropriate opening*. Allow a clearance of 1”(25mm) around the sides of the unit for mounting hardware. Allow clearance for cable connections to the back of the unit. Unit depth may vary according to cable type used. Typically, plan a depth to accommodate at least 3”(105mm) behind the panel.
Note: Deburr and clean cutout before beginning installation.

NEMA-4 Mounting Put the unit through the panel cut out. Slide the clamps into the 4 or 6 holes provided around the case. Tighten the clamping screws in an even pattern until the unit is secured in the panel.



Caution! Do not over tighten mounting clamps!

Note: **Specifications** To seal to NEMA-4 specifications, all supplied mounting clamps must be used and panel must not flex more than 0.010”.

2.1.3 Environmental Considerations
See Specifications for environmental constraints.

Location The MMI Series is designed for use in a factory environment. It is designed to operate from 32 to 113 °F (0 to 45 °C) temperatures, as found in most industrial environments. It may not be suitable for use in certain outdoor applications. Please consult the factory for advised usage in outdoor applications.

NEMA Rating The MMI Series front bezel is NEMA 4 rated. When installed properly in a NEMA 4 panel, the NEMA 4 rating of the panel is not compromised. This means that fluids do not enter the panel through the MMI series panel during wash downs.

Do not operate the unit in areas subject to explosion hazards due to flammable gases, vapors or dusts.

The unit should not be installed where fast temperature variations and/or high humidity are present. This causes condensation of water in the device.

Avoid installing units in environments where severe mechanical vibration or shocks are present. Vibration endurance: 10 to 25 Hz (X,Y,Z direction 2G 30 minutes)

2.2 Power Connections

Make sure that all local and national electrical standards are met when the installing the unit. Contact your local authorities to determine which codes apply.

2.2.1 Power Requirements

Power

The HMI can be powered by DC power only. The specified voltage range is +22 to 25 Volts DC. This insures compatibility with most controller DC systems.
The power conditioning circuitry inside the unit is accomplished by a switching power supply. The peak starting current can be as high as 700mA.

Fusing Requirements

It is recommended that all input power lines be protected from incorrect wiring or product failure by a 2 Amp fuse or a breaker.
If the display does not come on within 2 seconds of power up, remove power. An internal fuse prevents damage if the polarity of the DC power is incorrect. Check wiring to insure proper connections and try to power up again.

Warning! High Voltage

Connecting high voltages or AC power mains to the DC input makes the unit unusable and may create a hazard to personnel. Such a failure could result in serious personal injury, loss of life and or equipment damage.
DC voltage sources should provide proper isolation from main AC power and similar hazards.

Supply Voltage Condition

Do not power the HMI and inductive DC loads, or input circuitry to the controller, with the same power supply.
Note: The 24 VDC output from some controllers may not have enough current to power the HMI.

Wire Routing

Wire lengths should be minimized.

Wires should be run in pairs with a neutral or common paired with a hot or signal line.

Always use shielded cable to prevent unwanted electrical interference.

If wiring is to be exposed to lightning or surges, use appropriate surge suppression devices.

Keep AC, high energy, and rapidly switching DC wiring separate from signal wires by at least 8 inches. If signal wires must cross AC power, cross at right angles.

Equip ungrounded DC supplies with a resistor and capacitor in parallel to earth ground. This provides a path for static and high frequency dissipation. Typical values to use are 1MOhm and 4700pF.

Electrical Environment

The MMI Series has been tested to conform to European CE requirements. This means that the circuitry is designed to resist the effects of electrical noise. This does not guarantee noise immunity in severe cases. Proper wire routing and grounding insures proper operation. The MMI Series is also UL certified.

Warning! Emergency Stop

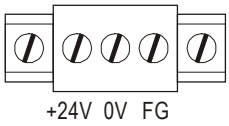
A Hard-wired EMERGENCY STOP should be fitted in any system using an HMI to comply with ICS Safety Recommendations.

Connection

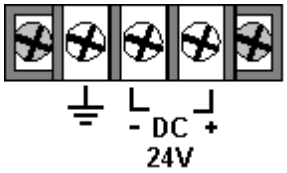
+24 VDC Wiring Diagrams

Use 18 AWG wire to connect positive DC line to the '+24V' (DC+) terminal and the DC ground to the '0V' (-DC) terminal. See text below about FG (⏚ Chassis Ground).

Terminal Plug: (MMI-720, 750 Models) To make a connection, strip about 3/8" of insulation off the end of the wire, and turn the connector screw counterclockwise until the gap is wide open. Insert the wire all the way in, and turn the screw clockwise until it's tight.



Terminal Block: (MMI-850, 1500 Models) To make a connection, strip about 3/8" of insulation off the end of the wire, turn the connector screw counterclockwise until the hold down plate is open wide enough to insert the wire. Insert the stripped portion of the wire under the plate and turn the screw clockwise until it's tight.



2.2.2 Grounding Requirements

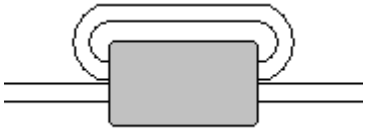
⏚ Chassis ground must be used. DC ground is not directly coupled to Earth ground internally. It is preferable not to ground DC negative return to chassis ground. Poor site earths can introduce noise into a system. If necessary, an earth connection should be made from the power supply return point to the central star earth point.

Ground conductors should be as short and as large in diameter as possible. The conductors must always be large enough to carry the maximum short circuit current of the path being considered. Ground conductors should be connected to a tree from a central star earth ground point. This ensures that no ground conductor carries current from any other branch.

2.2.3 CE Requirements

To make an HMI comply with EMC directives, and to reduce susceptibility to electrical interference, a separate #14 AWG ground wire should be taken to the chassis ground terminal of the power connector. This ground connection should be run directly to the central star earth connection point (as recommended in most Installation Instructions).

Use a ferrite core on the power wiring to reduce radiated emissions from the DC power lines. It is recommended to use a 140Ohm@100MHz ferrite core with the DC power lines looped through the core once. Position the ferrite core less than 1" away from the DC power connection points on the back of the HMI.



140 Ohm @ 100MHz ferrite core

2.2.4 Safety Guidelines

This section presents recommended installation practices, and procedures. Since no two applications are identical, these recommendations should be considered as guidelines.

Hardware Considerations

WARNING!

The system designer should be aware that devices in Controller systems could fail and thereby create an unsafe condition. Furthermore, electrical interference in an operator interface, such as an HMI, can lead to equipment start-up, which could result in property damage and/or physical injury to the equipment operator.

If you, or your company, use any programmable control systems that require an operator or attendant, you should be aware that this potential safety hazard exists and take appropriate precautions. Although the specific design steps depend on your particular application, the following precautions generally apply to installation of solid-state programmable control devices. In addition, these precautions conform to the guidelines for installation of Controllers as recommended in the NEMA ICS 3-304 Control Standards.

Programming Considerations

To conform with ICS Safety Recommendations, checks should be placed in the controller to ensure that all writable registers that control critical parts of plant or machinery have limit checks built into the program, with an out-of-limit safe shut down procedure to ensure safety of personnel.

ICS 3-304.81 Safety Recommendations:

Consideration should be given to the use of an emergency stop function, which is independent of the programmable controller.

Where the operator is exposed to the machinery, such as in loading or unloading a machine tool, or where the machine cycles automatically, consideration should be given to the use of an electromechanical override or other redundant means, independent of the programmable controller, for starting and interrupting the cycle.

If provision is required for changing programs while the equipment is in operation, consideration should be given to the use of locks or other means of assuring that such changes can be made only by authorized personnel.

These recommendations are intended as safeguards against the failure of critical components and the effects of such failures or the inadvertent errors that might be introduced if programs are changed while the equipment is in operation. *

* The ICS 3-304.81 Safety Recommendations are reproduced by permission of the National Electrical Manufacturers Association from NEMA ICS 3-304

2.3 CE Requirements

2.3.1 EU directives that apply to MMI Series

- EMC Directive (89/336/EEC, 92/31/EEC, 93/68/EEC) electromagnetic emissions and immunity
- Machinery Directive (89/392/EEC, 91/368/EEC, 93/44/EEC, 93/ 68/EEC) machine safety

MMI products are CE-marked to indicate compliance with the EMC Directive. Declarations of Conformity that specify the directive(s) and the catalog numbers of the products covered are available from ORMEC Systems Corp.

The MMI Series has been designed to operate satisfactorily in electromagnetic noise (immunity) and without emitting high levels of electrical noise into the environment (emission). The units are designed to meet European Community standards when installed per the wiring instructions in this manual.

Compatibility Standards

The MMI has been designed to meet electromagnetic compatibility for industrial environments.

Standard	Description
CISPR (EN 55011) Group 1, Class A	Radiated Emissions levels
EN50081-2	Generic emission standard, industrial environment (Also US FCC Class A)
EN50082-2	Generic immunity standard, industrial environment

2.3.2 Guide Lines for EU Installations

• Be aware that wiring leaving the cabinet where the unit is installed may be exposed to interference sources.

• The installation practices in the individual product installation manuals of other components in the system must also be followed.

• Locally applicable grounding safety regulations and machinery directives should be followed for providing a protective ground to earth. The EMC ground must be a low impedance, low inductance path to the machine chassis ground.

• Power supply to the unit must be through an IEC-rated isolation transformer.

• The Power supply to the controller must be controlled to ensure that it does not exceed over voltage category II per EN60204-1 (IEC 240).

• Other requirements of the Machinery Directive involving displays, languages, instructions, Emergency Stop functions, machine operation, protective guards and interlocks are the responsibility of the machine manufacturer.

• Use a ferrite core on the power wiring to reduce radiated emissions from the DC power lines. It is recommended to use a 140Ohm@100MHz ferrite core with the DC power lines looped through the core once. Position the ferrite core less than 1” away from the DC power connection points on the back of the UNIT.

2.3.3 Safety Guide Lines for EU Installations

• Only qualified personnel should be allowed to specify, apply, install, operate, maintain or perform any other function related to HMI products. Qualified persons are defined as follows:

System application and design engineers who are familiar with the safety concepts of automation equipment.
Installation, start-up, and service personnel who are trained to install and maintain automation equipment.
Operating personnel trained to operate automation equipment and trained on the specific safety issues and requirements of the particular equipment.

• Make sure that the voltage range for the equipment is correct before switching on the equipment.

• Emergency-tripping devices in accordance with EN60204/IEC204 must be effective in all operating modes of the automation equipment. Resetting the emergency off device must not result in any uncontrolled or undefined restart of the equipment.

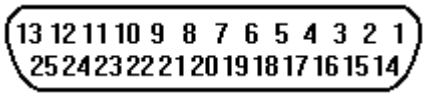
• Automation equipment and its operating elements must be installed so that unintentional operation is prevented.

• Make sure that operating sequences, interrupted by a voltage dip or power supply failure, resume proper operation when the power supply is restored. If necessary, the equipment must be forced into the “emergency off” state.

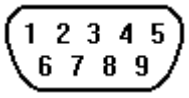
• Install the power supply and signal cables so that inductive and capacitive interference voltages do not affect automation functions.

2.4 Communications Connections

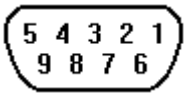
The ports as you look at the back of the case, are the ports for connecting to a printer, PLC or some external device (Controller Connectors).



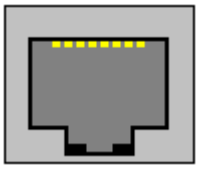
Printer Port
(25p D-Female)
(Not Available on MMI-720, 750)



PLC [RS-232] Port
(9 pin D-Female)



PC [RS-232] &
PLC [RS-485] Port
(9 pin D-Male)



Ethernet Port
(RJ-45 Male)
(Not Available on
monochrome units)

2.4.1 Connection to an External Device

Cable Requirements

Different cables are required for communications between an ORION vs an SMLC. These cables can be obtained from ORMEC.

Warning

Communications problems cause the display to show **PLC no response...** until communications can be established. During this time, the controller cannot be affected by the HMI. The COM light on the front of the HMI turns on with each communication and should appear as if always on or slightly flickering when communications are good.

Restrict cable length to avoid communications problems due to weak signals.

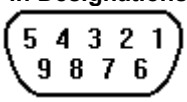
Recommended distances:

RS232:	less than 50' (15m)
RS485/422:	less than 500' (150m)
Ethernet:	less than 328' (100m)

Shielded cable must be used for long lengths or cables run in an electrically noisy environment. Use twisted pair cables for all Ethernet connections.

Do not run cables next to AC power lines or near sources of electrical noise.
Be sure that the cable ends have been inserted all of the way into mating connectors and are secure.

Pin Designations



PLC [RS-232]

Pin assignment of the 9 Pin, Female, D-SUB, PLC [RS-232] Port. This port is used for connecting the HMI to a controller or Master HMI unit. The Auxiliary (AUX) RS232 port is also accessed through this connector. Both PLC and AUX ports share the common ground.

Note: This port is <u>not</u> used for programming the HMI or for printing functions.
<u>Do not</u> plug the **MT5_PC**'s PLC cable end into this port.

Pin #	Symbol	PLC[RS232]	AUX[RS232]
1	AUX TxD		Transmitted Data
2	PLC TxD	Transmitted Data	
3	PLC RxD	Received Data	
4	Not used		Received Data
5	GND	Signal Ground	Signal Ground
6	AUX RxD		
7	PLC CTS	Clear to send input	
8	PLC RTS	Ready to send output	
9	Not used		

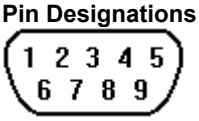
2.4.2 Connection to a Personal Computer

The 9 Pin, Female, D-SUB, PC [RS-232] & PLC [RS-485] Port on the back of the unit is the programming port (PC Connector) and RS485/422 communications port for connecting to a controller. The Auxiliary (AUX) RS485 port is also accessed through this connector.

Connection This port can be attached to a Computer via a special DB9 Female to DB9 Female cable provided with the unit.

Port Activation This port is activated automatically by the PC during: On line simulation, Download and Upload activities.
The Programmer Port cannot simulate, download or upload to the unit while it is on line with the controller at the same time. The unit must be put into “RDS” mode with the EasyManager applet first.

Pin Designations Pin assignment of the 9 Pin, Male, D-SUB PC [RS-232] & PLC [RS-485] Port

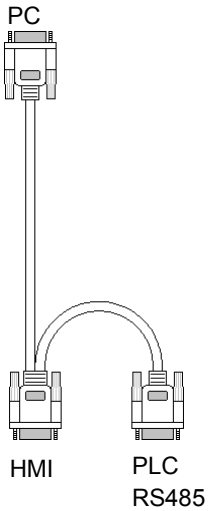


PC [RS-232] & PLC [RS-485]

Pin #	Symbol	PLC[RS485] 2 Wire	PLC[RS485] 4 Wire	AUX[RS485] 2 Wire	PC[RS232]
1	PLC RxD-	Transmit/Receive -	RS485 Receive		
2	PLC RxD+	Transmit/Receive +	RS485 Receive		
3	PLC TxD-		RS485 Transmit		
4	PLC TxD+		RS485 Transmit		
5	GND	Signal Ground	Signal Ground	Signal Ground	Signal Ground
6	AUX Data +			Transmit/Receive +	
7	PC TxD				RS232 Transmit
8	PC RxD				RS232 Receive
9	AUX Data -			Transmit/Receive -	

HMI to PC, MT5_PC Cable Configuration

Connect to Personal Computer (PC) RS232 Serial Port DB9 Female	Connect to HMI (HMI) RS232/485 [PLC] DB9 Female	Connect to Controller RS485 Port DB9 Male
1 Not used	1 RX- →	1 RX-
7 Not used	2 RX+ →	2 RX+
8 Not used	3 TX- →	3 TX-
4 Not used	4 TX+ →	4 TX+
5 GND →	5 GND →	5 GND
6 Not used	6 AUX Data +	6 AUX RS485+
2 TxD →	7 TxD	7 Not used
3 RxD →	8 RxD	8 Not used
9 Not used	9 AUX Data -	9 AUX RS485-

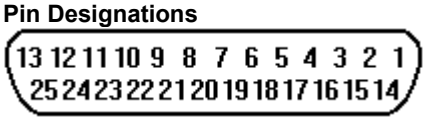


2.4.3 Connection to a Printer

The printer port on the back of the unit is a Parallel printer port and is compatible with most printers that accept parallel connectors.

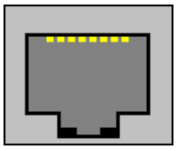
Print Out The MMI-640 models have printout capabilities. The printer port transmits data when a printable object is activated.

Pin Designations Pin assignment of the 25 Pin, D-SUB, Parallel Printer Port.



Pin #	Symbol	Function
1	STB	Output
2	DATA0	Output
3	DATA1	Output
4	DATA2	Output
5	DATA3	Output
6	DATA4	Output
7	DATA5	Output
8	DATA6	Output
9	DATA7	Output
11	BUSY	Input
15	ERROR	Input
16	INIT	Output
17-25	GND	Signal Ground

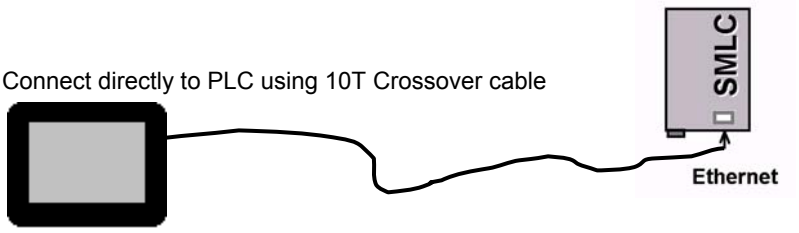
2.4.4 Ethernet Connections



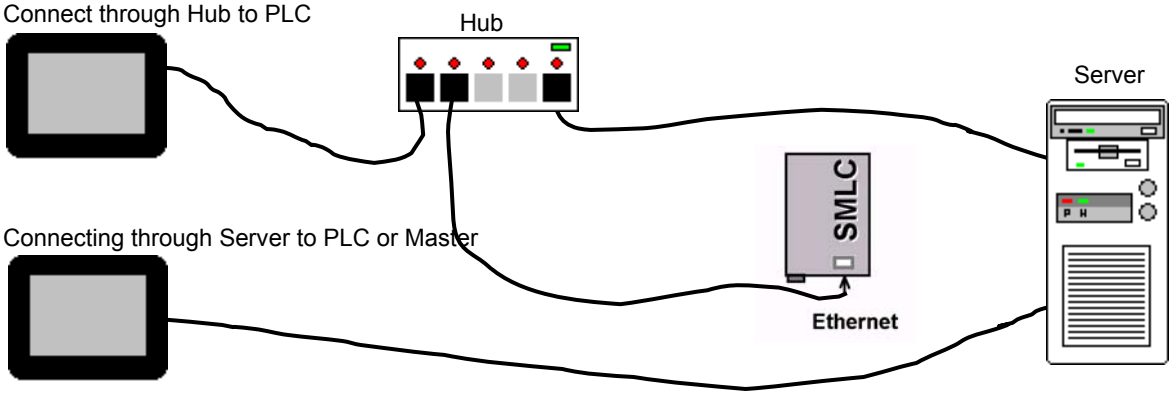
Units equipped with the Ethernet port can be connected to Ethernet devices using standard 10T CAT5 Ethernet cables. Routing may pass through Hubs and Servers as needed to connect to the PLC.

The Ethernet port can also be used for HMI Master/Slave communications using the same type of configurations.

Connect directly to PLC using 10T Crossover cable



Connect through Hub to PLC



Connecting through Server to PLC or Master

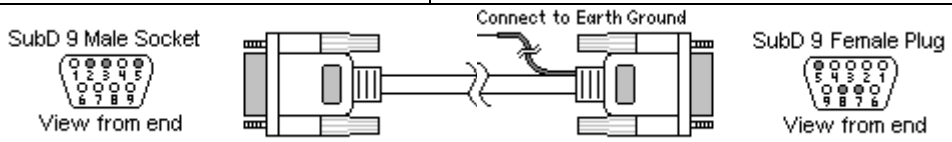
2.4.5 HMI to HMI Connections

The HMI supports a master - slave communications. One HMI is connected directly to the PLC and configured as the Master. All the other HMIs are connected in series to it and are configured as Slaves. In theory there is no limitation to the number of HMIs on a chain, however response time gradually decreases when more than three HMIs are linked together. The HMI are configured with the EasyBuilder software to be the Master or a Slave.

Wiring between two HMIs

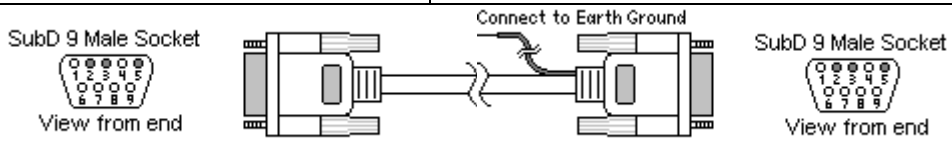
Case 1: Slave to Master, connecting PLC[RS232] directly to PC[RS-232]/PLC[RS485] combination port.

Connect to SLAVE HMI PLC[RS-232] port Cable has D-SUB Male end	**Connect to MASTER HMI PC[RS-232] port Cable has D-SUB Female end**
2 TxD →	8 RxD
3 RxD →	7 TxD
5 GND →	5 GND
	Shield → Earth Ground



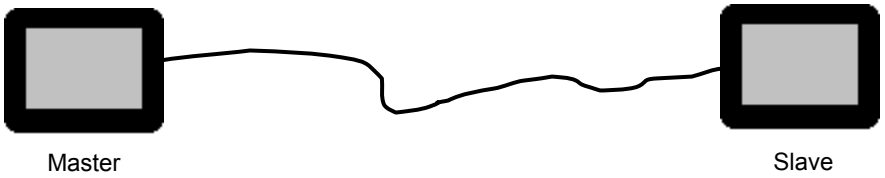
Case 2: Slave to Master, connecting PLC[RS232] to PC side of split download cable (MT5_PC) with use of Male to Male Null Modem cable.

Connect to SLAVE HMI PLC[RS-232] port Cable has D-SUB Male end	**Connect to MT5_PC to MASTER HMI PC[RS-232] port Cable has D-SUB Male end**
2 TxD →	3 RxD
3 RxD →	2 TxD
5 GND →	5 GND
	Shield → Earth Ground

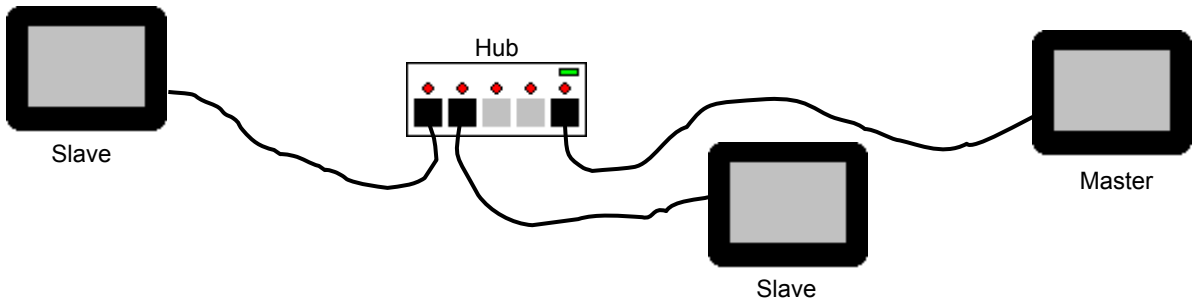


Case 3: Ethernet Connections: Units equipped with the Ethernet port can be connected via Ethernet to a Master and Slaves using standard 10T Ethernet cables. Slave to Master, connect the Ethernet ports with a standard (Category 5) Ethernet 10T crossover cable (RJ-45 to RJ45). On the other hand, use standard cables with routing passing through Hubs and Servers as needed.

Connecting a Master directly to a Slave using 10T Crossover cable



Connecting a Master through a Hub to Slaves



2.5 Dip Switch Settings

All dipswitches should normally be in the OFF (down) position. Dipswitches 1 and 2 are used to perform testing and recovery functions. Dip Switch 3 and 4 are not used and should be left in the off position.

	SW1	SW2	Mode
	OFF	OFF	Application mode (On line operations, use EasyManager or EasyBuilder to change modes)
	ON	OFF	Force to Touch Adjust mode (Used for touchscreen calibration)
	OFF	ON	Force to RDS mode (Remote Debug and Simulation, used to recover from invalid states due to corrupt downloads. Refer to section on troubleshooting.)
	ON	ON	Force to Touch Screen Test mode (Used to check accuracy of Touch Screen)

Note: It is normally not necessary to change dipswitches to put the HMI into programming (RDS) mode.

2.6 HMI Indicator Lights

PWR – Indicates if power has been applied to the unit. If this indicator fails to light, check power, check polarity of wiring and check fusing. If all conditions are correct, contact the factory for help.

CPU – Indicates that the CPU is operating properly. If this indicator fails to light, please contact factory.

COM – Lights when a serial transmission is sent or received. This indicator appears to flicker when trying to establish communications.

2.7 Other Hardware Considerations

2.7.1 Touchscreen Calibration

Normally the touchscreen is calibrated at the factory for proper functionality. If the touchscreen cannot be calibrated, please contact the factory for assistance.

Run the **EasyManager** Utility to get to the Touch screen calibration (Touch Adjust) mode.

Click **Jump to Touch Adjust** to put the unit in this mode. This is used to calibrate the touch screen.

On the screen of the HMI unit:

Touch the crosshairs as displayed in sequence on the screen.

Touch anywhere outside to the rectangles to move the crosshairs to that location.

Touch inside the left rectangle to repeat the calibration process.

Touch inside the right rectangle to accept and end the calibration procedure.

Touch the screen again to jump to application mode.

2.7.2 CCFL and Battery Replacement

It is recommended that the factory replacement of these components in case of their failure.

If the product must be returned, be sure to call ORMEC and get a Return Material Authorization (RMA) number first.

Units should be returned in their original packaging container, otherwise, any suitable rigid container can be used as a substitute. Use appropriate packing material. Damage due to shipment is not covered by the warranty. Be sure to include a description of the problem and contact details for our repair department.

All returns are evaluated for proper operation. During evaluation, customer projects are not retained in the units. If you need your project retained, please indicate this in the documentation included with the unit.

3.0 Specifications

3.1 General Specifications

Item	Specification
Input power	21-25 VDC, 500 mA @ 24VDC
CE	Complies with EN50081-2 and EN50082-2 standards
EMI	Complies with FCC Class A (Ferrite core required if using unshielded power supply wires)
Isolation resistance	Exceeds 50 MΩ at 500VDC
Vibration endurance	10 to 25 Hz (X,Y,Z direction 2G 30 minutes)
Protection structure	NEMA 4 / IP65 front panel (when mounted with gasket seal)
Operating Temperature	32 to 113 °F (0 to 45 °C)
Operation humidity	10 to 90% RH Non Condensing
Enclosure	Plastic: Polybutylene Terephthalate (PBT) and Polycarbonate(PC)

3.2 Hardware Specification 640 Models

Item	**640/10T**	**640/10M**	**640/8T**	**640/8C**
LCD Display	10.4" TFT, 256 color	10.4" STN, 4 color gray	7.7" TFT, 256 color	7.7" STN, 256 color
Display area (mm)	215(W) x 162(H)		162(W) x 123(H)	
Contrast Ratio	150:1	18:1	250:1	30:1
Brightness cd/m2	300	100	400	150
Back light	CCFLx2(MTBF 25,000hr)	CCFLx2(MTBF 15,000 hr)	CCFLx2 (MTBF 50,000 hr)	
Resolution pixels	640(W) x 480(H)			
Pixel size	0.33(W) x 0.33 (H) mm			
Touch panel	4 wire resistive type			
Touch granularity	2mm grid			
Touch Feedback	Beeper and or Graphic Indicator			
Surface hardness	4H			
Processor	32 bit RISC CPU 200 MHz			
Flash ROM Memory	2 MB Standard, 4MB with Ethernet option			
System Memory	4MB DRAM			
Battery Held Memory	128kB with Y2K compliant Real Time Clock/Calendar			
Compact Flash Slot	Used for Project transfers only	Not Available	Used for Project transfers only	
Serial ports	1 RS-232 (controller port) and 1 RS-232 / RS-485 (PC & controller port)			
Parallel port	Standard parallel printer port			
Ethernet Port	RJ-45 8 wire (10 BaseT) TCP/IP	Not Available	RJ-45 8 wire (10 BaseT) TCP/IP	
System diagnostic	Watch dog timer, power failure detection			
Dimensions Bezel H x W x D inches (H x W x D) mm	**Bezel:** 9.37 x 12.40 x 2.44 (238 x 315 x 62)		**Bezel:** 6.93 x 9.09 x 2.16 (176 x 231 x 55)	
Cutout H x W inches (H x W) mm	**Cutout:** 8.86 x 11.89 (225 x 302)		**Cutout:** 6.57 x 8.75 (167 x 222)	
Weight	Approx. 2.0 kg		Approx. 1.2 kg	

3.3 Hardware Specification 320 Models

Item	320/3T	320/6T	320/6C	320/6M
LCD Display	5.7" TFT 256 color	5.7" TFT 256 color	5.7" STN 256 color	5.7" STN blue mode
Contrast Ratio	150:1	60:1	30:1	15:1
Brightness cd/m2	500	300	150	60
Back light CCFLx1	MTBF 30,000 hr	MTBF 40,000 hr	MTBF 25,000 hr	MTBF 15,000 hr
Resolution pixels	320(W) x 234(H)	320(W) x 240(H)		
Pixel size	0.33(W) x 0.33 (H) mm			
Display area (mm)	120(W) x 90(H)			
Touch panel	4 wire resistive type			
Touch granularity	2mm grid			
Touch Feedback	Beeper and or Graphic Indicator			
Surface hardness	4H			
Processor	32 bit RISC CPU 200 MHz			
Flash Memory	1 MB Standard, 2MB with Ethernet Option			
System Memory	4MB DRAM			
Battery Held Memory	128kB with Y2K compliant Real Time Clock/Calendar			
Compact Flash Slot	Used for Project transfers only			Not Available
Serial ports	1 RS-232 (controller port) and 1 RS-232 / RS-485 (PC & controller port)			
Parallel Printer port	No printer port available			
Ethernet Port	RJ-45 8 wire (10 BaseT) TCP/IP			Not Available
System diagnostic	Watch dog timer, power failure detection			
Dimensions H x W x D inches (H x W x D) mm	**Bezel:** 5.90 x 8.00 x 2.95 (150 x 204 x 75) **Cutout:** 5.43 x 7.56 (138 x 192)			
Weight	Approx. 0.8 kg			

3.3 Functional Specification

Screen editor	EasyBuilder Version 2.6.2 or later (to be run under Windows 98 / NT / 2000 / ME / XP*)
No. of window	1 ~ 1999, limited by memory
No. of object	Up to 500 per window, limited by memory
Text strings	limited only by memory
Bitmap graphics	256 KB per graphic, limited by memory
Support PLC	Most popular PLC's
Support Printer	EPSON ESC/P2, HP PCL/(Simple page mode) or compatible
Macro scripts	Up to 256 Macro scripts per project, limited by memory

Computer requirements include at least a Pentium 90Mhz PC, 16MB RAM, 10MB available hard disk space, minimum 800x600 resolution VGA, and one available RS-232 serial port.

4.0 Trouble Shooting

4.1 Power Problems

Problems on power up: Unit does not light or unit lights but does not display any windows.

1. Check wiring for proper polarity.
2. Check power Supply for proper Voltage and Current capacity.
3. Check fuse.

Problems during operation: Faulty unit operation may be due to problems with power quality. The HMI has been designed to work in environments where electrical noise is present. However, extreme electrical noise still causes problems. Make sure that the system is properly earth grounded.

4.2 Communications Problems

Sometimes communications fail. When communications fail, the unit automatically tries to establish the communications link again. During the time the unit is establishing communications, the touchscreen of the unit does not respond. Function key operations are interrupted. The implication is that the unit should not be used for Emergency Stop applications. A loss of communications can happen at any time. Using the function keys on the unit for critical operations can lead to a potential disaster. It is good programming practice to allow for safe operation in case of interface failure.

There are various reasons why this happens;

Improper programming: If the MMI window is programmed to access data from an invalid register or bit address for the PLC, the unit receives an error message from the PLC. The unit interprets this as a loss in communications. Be sure that all data points being displayed are valid for the PLC that is connected.

Loose or incorrect cables: Make sure that all cables are secured and configured properly for the PLC.

Time outs: Make sure that the PLC is responding to requests from the HMI in a timely manner.

Power loss: Make sure the PLC has power and is running properly.

Electrical noise: Faulty unit operation may be due to problems with power quality. The HMI has been designed to work in environments where electrical noise is present. However, extreme electrical noise still causes problems. Make sure that the system is properly earth grounded.
The use of proper grounding techniques insures reliable communications. Make sure the controller and the HMI are connected to good earth ground sites. This allows EMI (Electro-Magnetic Interference, commonly called electrical noise) to be channeled to ground where it can no longer disrupt electrical operations. Be sure to route communications cables in separate bundles and locations from AC power and control wiring. Do not run communications cables near solenoid and relay coils or AC and DC drive controllers. Care should also be taken to locate the HMI itself away from sources of EMI.

4.3 Commonly Asked Questions

Q. Can I have multiple MMIs connected to one Controller?
A. The ability to connect more than one HMI to a controller is accomplished through an HMI to HMI link. This is not done through any PLC protocol. It is done through the master-slave protocol of the MMI.

Q. How do I call up windows with my PLC?
A. Use the PLC Control part configured as “Change window” to call up windows by word value. Additionally, Direct and Indirect Window parts can be used to bring up windows.

Q. Do I need to change any jumpers to go from one Controller type to another?
A. No, the driver that is downloaded into the unit at programming time determines the Controller type. The dip switches on the back of the unit should all be in the OFF position.

Q. Is there any way to completely erase the HMI user memory?
A. No, the HMI memory is initialized automatically before every download cycle.

Q. How Do I change the Battery?
A. Battery replacement requires disassembly of the unit in an ESD controlled environment. Battery life expectancy is greater than 5 years.

The MMI-7XX takes one coin type of CR2032 lithium battery to backup the recipe data and keep the RTC running. Battery specification: CR2032 3V lithium battery.

Steps for battery replacement:

1. Use EasyManager to backup the retentive memory data.
2. Turn off the HMI and remove its rear cover.
3. Remove the battery from the socket.
4. Insert a new battery into the socket.
5. Put on the rear cover.
6. Reset the RTC time and download the retentive memory data.

The MMI-850 and 1500x products have the battery soldered to the main board. We recommend that these units be sent back to the factory for battery replacement.

Q. Can I change the Backlighting bulbs in the field?
A. We recommend that the unit be sent back to the factory for bulb replacement. Backlight bulb life expectancy is greater than 5 years even if running 24 hours a day. Use the Backlight auto shutdown to conserve backlight bulb life.

Q. Why am I getting slow updates on my windows?
A. PLC communication speed controls the update speed. Try using a higher baudrate and adjusting the Block pack settings. If an overloaded window is causing a slow update, we suggest changing the window design.
NOTE: EasyWindow has a tool called “SystemResource”. It displays object queue item numbers. Use this to help detect communication problems. This problem might not show up in Offline Simulation because the PC may have more CPU speed, more caches, and more VGA speed.

4.4 Hardware Problems

4.4.1 Black Screen after download

Symptom: After project downloaded to a new touchscreen a blank black screen is displayed.
Cause: Using the older versions of EasyBuilder to download to newer model touchscreens.

Any version prior to version 2.0.2 downloading to Hardware version 3
Any version prior to version 2.5.1 downloading to Hardware version 4
Any version prior to version 2.6.0 downloading to Hardware version 4.5

Fix: Install new version of EasyBuilder on your computer.
Set DIP Switch 2 ON in the HMI and reset Unit.
Load your project using new EasyBuilder.
Set DIP Switch 2 OFF and reset Unit.
Project should now operate normally.

4.5 Repairs and Returns

An MMI is designed to provide years of trouble free service. An MMI under goes a full functional test before shipment.

The MMI warranty is for one year under normal use.
The MMI does not require any “Routine Maintenance” by the user. If a problem should occur, and all troubleshooting procedures have been exhausted, contact ORMEC at (585) 385-3520.

If the product must be returned for any reason, be sure to call ORMEC and get a Return Material Authorization (RMA) number first.

Units should be returned in their original packaging container, otherwise, any suitable rigid container can be used as a substitute. Use appropriate packing material. Damage due to shipment is not covered by the warranty. Be sure to include a description of the problem and contact details for our repair department.
All returns are evaluated for proper operation. During evaluation, customer projects are not retained in the units. If you need your project retained, please indicate this in the documentation included with the unit.
Products passing normal QC tests are returned to the customer and an evaluation charge is incurred.
If the problem is verified and the unit is in warranty, ORMEC will repair or replace the unit.

5.0 Quick Startup Guide

5.1 Connections

Set up the HMI with PC and PLC as described below.

a) Connect PC [RS-232] port of the HMI to the PC. Since the PC [RS-232] and PLC [RS-485] share the same D-SUB connector, we recommend using the HMI to PC cable provided (MT5_PC). This splits the port into two separate connectors to ease the program and test process.

b) Connect either the PLC [RS-485] or PLC [RS-232] port of the HMI to the PLC using the proper cable. (Check the PLC signal type and cable listing in the back of this manual to assure proper port connections.)

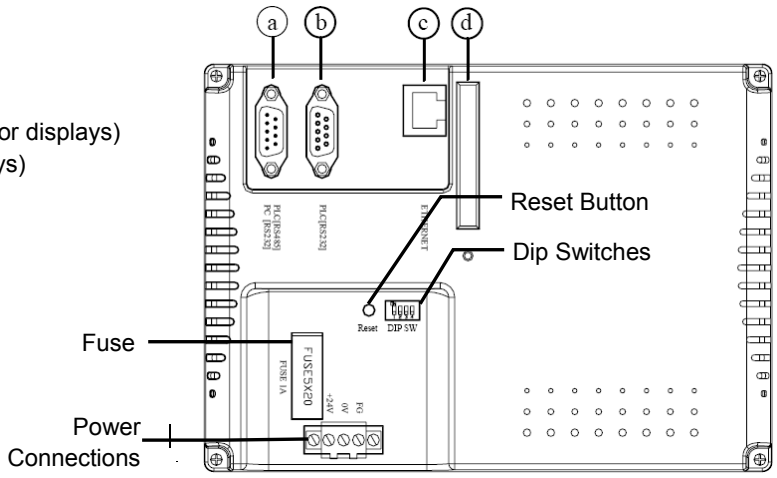
c) Connect DC 24V power to the power connector. See Section 2.2.1 for power connection specifications.

Note: Check that all Dip switches are set to the “OFF” position.

d) Apply power and, if necessary, calibrate contrast adjust to the best viewing performance. (Not applicable for MMI-640/10T) Set new V4.5 hardware versions in RDS mode where contrast can be adjusted electronically.

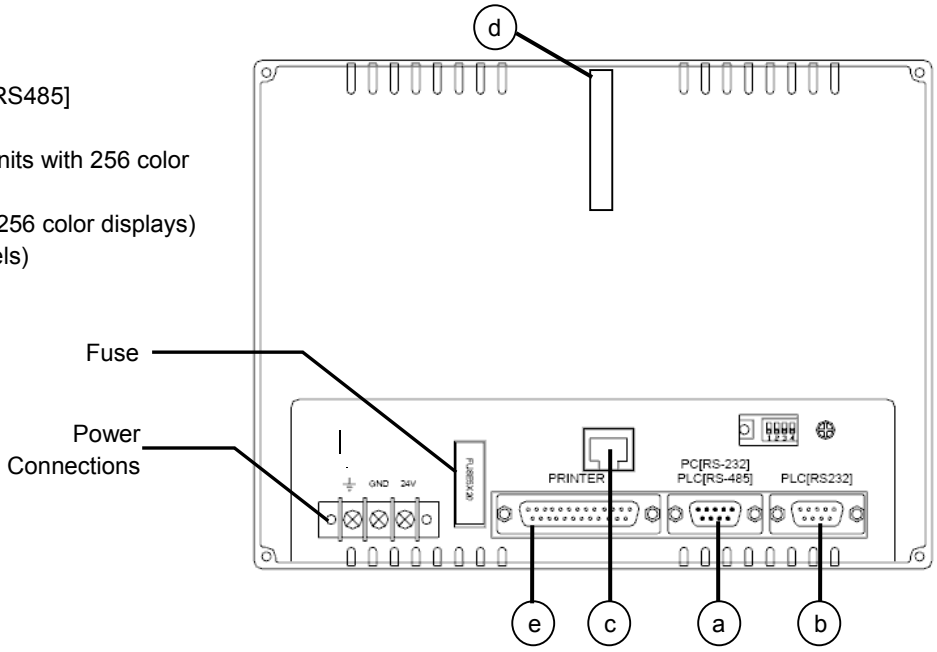
MMI-320 Connection points

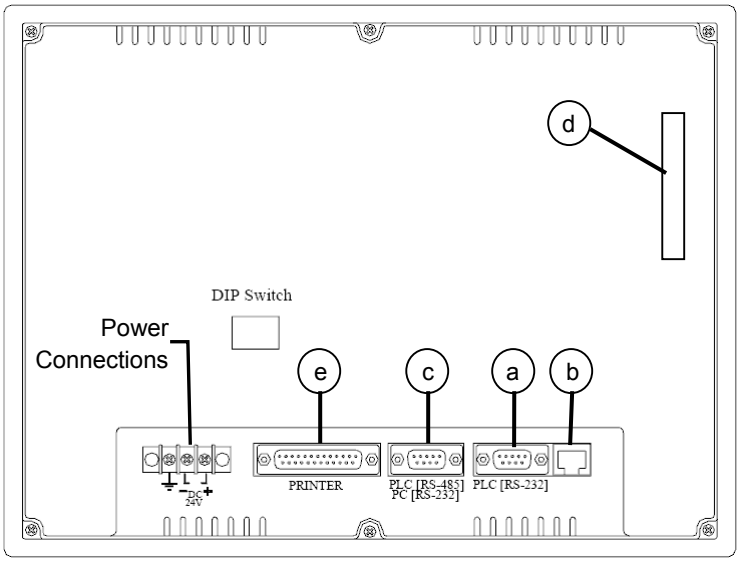
a. PLC[RS485]/PC[RS232]/AUX [RS485]
b. PLC[RS232]/AUX [RS232]
c. Ethernet port (RJ-45) (only on units with 256 color displays)
d. CF card slot (only on units with 256 color displays)



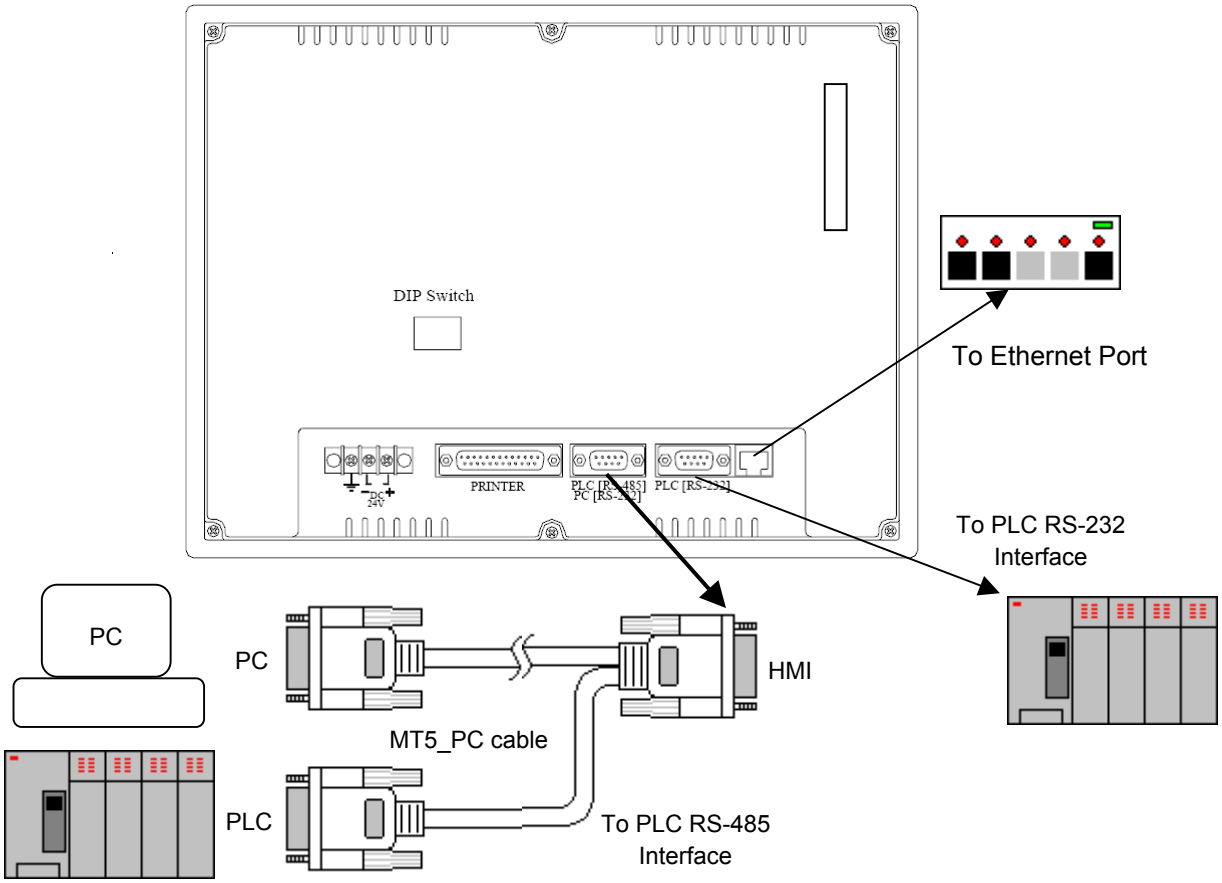
MMI-640 Connection Points

a. PLC[RS485]/PC[RS232]/AUX [RS485]
b. PLC[RS232]/AUX [RS232]
c. Ethernet port (RJ-45) (only on units with 256 color displays)
d. CF card slot (only on units with 256 color displays)
e. Printer port (only on some models)





Typical connection



5.2 Installing EasyBuilder

Install EasyBuilder 500 on your PC. Software must be installed on a PC running Windows 98®, Windows 2000® or Windows XP® software. PC screen resolution must be set to 800x600 or greater. Also, at least a Pentium 90Mhz CPU with 16MB RAM, a CDROM drive, 15MB available hard disk space, VGA video controller, and one available RS-232 serial port is required.

Put the EasyBuilder Installation CD into the CD drive. The autorun should bring up a screen showing an area to click to begin the EasyBuilder Installation.

If the Autorun sequence does not start, browse the CD with Windows® Explorer and start the installation from there. Once the Installation process is done, the Start Menu has selections for starting EasyBuilder and EasyManager.
There is no need to restart your computer after installation, although this is recommended.

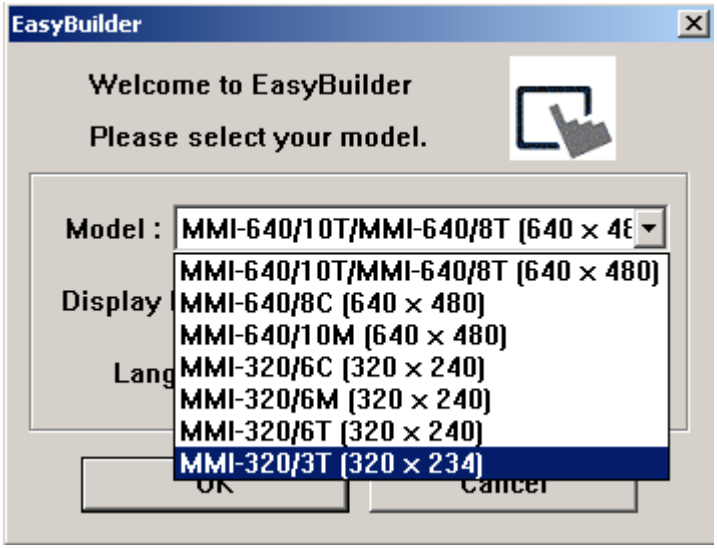
5.3 Initial Start Up

Use **EasyManager** to set the following:

1. **COM Select Dropdown:** Select the number of the RS232 Serial COM port for communications to the MMI. Ports COM1 through COM10 are available.

2. Click on **EasyBuilder** to start the screen editor for the MMI.

Note: See the Software Reference section for further details about the **EasyManager** Application.



After clicking the **EasyBuilder** button, the following popup dialog appears if this is the first time running EasyBuilder. Otherwise, the last open project is automatically opened for editing.

To start a new project, use the keyboard shortcut [Ctrl + N]

or click the  tool or select [**New**] from the **File** menu and a new project is created from the project template.

3. Select the appropriate **Model** you are programming….

ORMEC Part#	Display Type
MMI-640 / 10T (640 x 480)	10.4” TFT 256 Color
MMI-640 / 10C (640 x 480)	10.4” STN 256 Color
MMI-640 / 10M (640 x 480)	10.4” STN Mono - Grayscale
MMI-640 / 8T (640 x 480)	7.7” TFT 256 Color
MMI-640 / 8C (640 x 480)	7.7” STN 256 Color
MMI-320 / 3T (320 x 234)	5.7” TFT 256 Color
MMI-320 / 6T (320 x 240)	5.7” TFT 256 Color
MMI-320 / 6C (320 x 240)	5.7” STN 256 Color
MMI-320 / 6M (320 x 240)	5.7” STN Mono - Grayscale

4. Select the **Display Mode**….

Landscape Portrait

5. Select the Language Mode….
This setting determines the character set that is used in project development.

Select **Single Byte** for European languages. These fonts are in the directory as Ascfont.8, Ascfont.16 and Ascfont.24. These represent the 8, 16 and 24 point sizes. Larger sizes are generated from these base sizes (Example: Font size 32 is actually size 16 doubled).

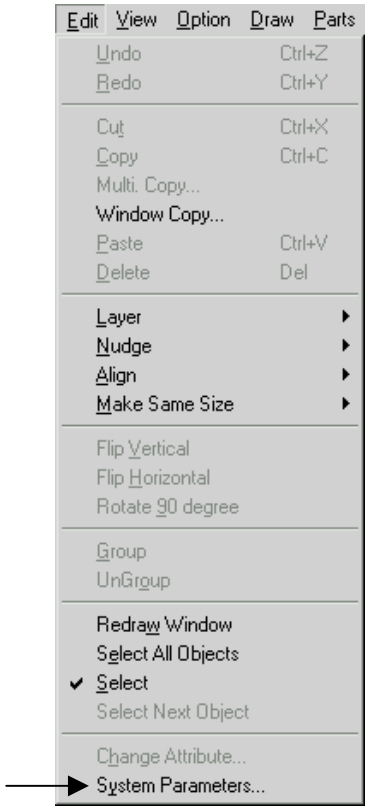
Select **Double Bytes** for Asiatic languages. Use the Font setting in the System Parameters Editor Tab to select the appropriate character set.

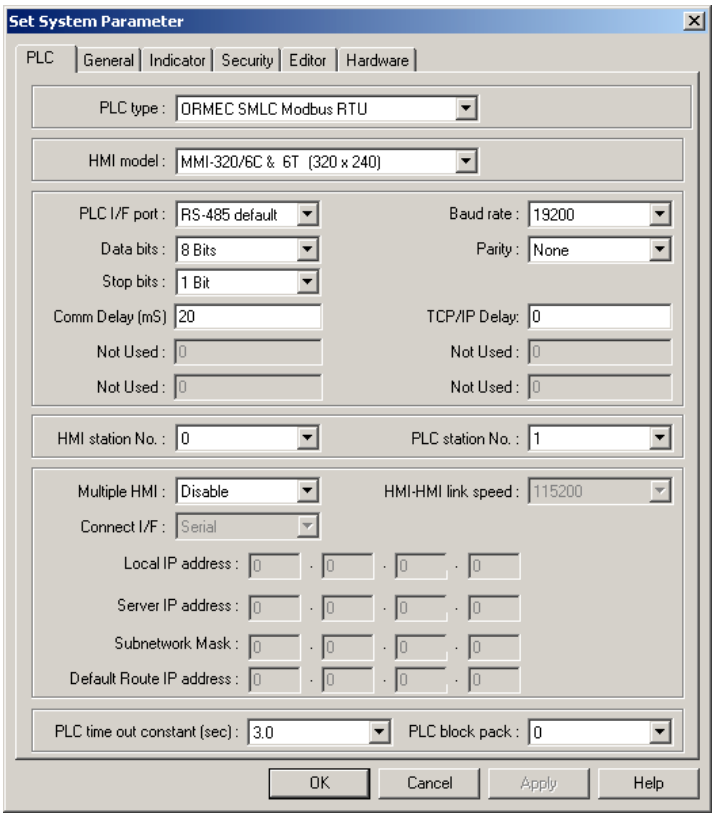
Then click, **OK**.

5.4 Creating a project

A project file (*.epj FILE) is simply a collection of all the windows and window data used by an application.

Step 1. Select **Edit|System Parameters...** and the following screen appears. Fill in the system parameters. Use the **PLC** Tab to select the set up the parameters for communicating to the PLC.





a. Select the **PLC type** from the dropdown list.

b. Confirm that the **HMI Model** is the one you are programming.

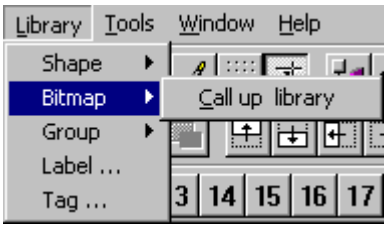
c. Check the communications settings:

Serial I/F port
Baud rate
Data bits
Parity
Stop bits
HMI station No.
PLC station No.

Set the communications parameters to match the PLC. (See the Controller Reference Guide for details.)

Note:
The **General, Indicator, Security, Editor, Hardware** and **Auxiliary Port** tabs are used for other settings. See Section 2.0 for full details. For a quick start, these settings can be left at their defaults.

Step 2. Select any additional Group, BMP and/or Shape Libraries to attach to the project. The default selection provides a good range of library objects. For a quick start, no additional libraries are needed.



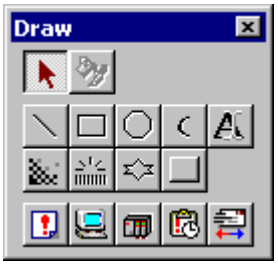
In the **Library** menu, select [**Shape, Bitmap** or **Group**] → [**Call up Library**] or use the toolbar icons  .

EasyBuilder provides three types of libraries.

A **Shape** is a collection of drawing elements, those elements, when put together, defines a graphic symbol representing a button, lamp, function key etc.
A **Bitmap** is a collection of pixel of data; each pixel can be 1, 4 or 8 bits.
A **Group** Library is a collection of shapes and bits that are frequently used and have been saved as a group.

Step 3. Design the windows

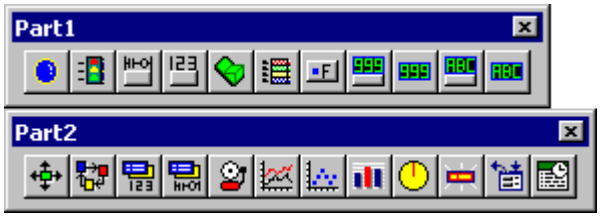
Using the parts and drawing elements, virtually any simple window display can be completed in 10 minutes. "Ease of Use" is the greatest benefit of our EasyBuilder screen editor software.

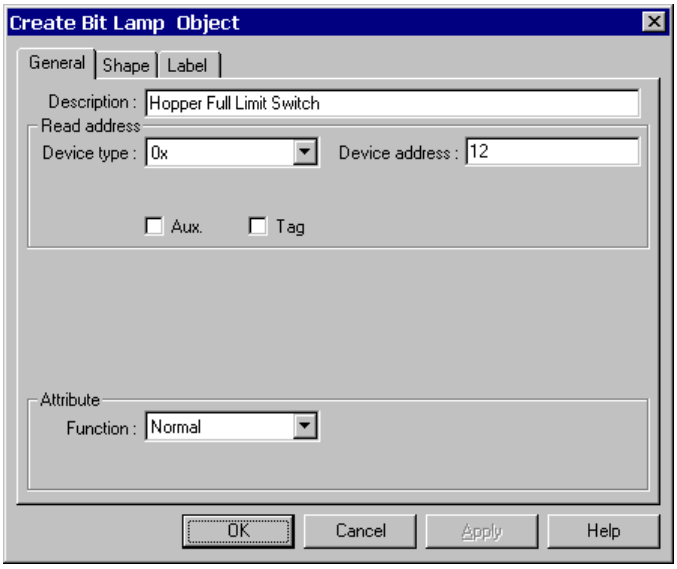


a. Use the **Draw** menu/tools to put static text and shapes on the window. The drawing tools are used in the similar fashion to most windows drawing packages.

For example: To draw a line: Select the **Line** tool. Change any attributes for the line. Click on the screen where the line is to begin, move to where the line is to end and click again to set the end point.

b. Put active parts on the screen by using the **Parts** menu/tools. The parts automatically pop up a dialog for entering information related to the part operation.





For example: Placing a Bit Lamp: Select Bit Lamp from the Parts menu. Select the type of bit to access. Enter the number of the bit. Go to the shape tab. Select Use shape and click on the Shape Library button to select a shape. Click OK to close the dialog. The Bit Lamp is placed in the upper left corner of the display. Move it to the desired position.

Step 4. Save and Compile the project

In the menu, select **File|Save** to save the project.

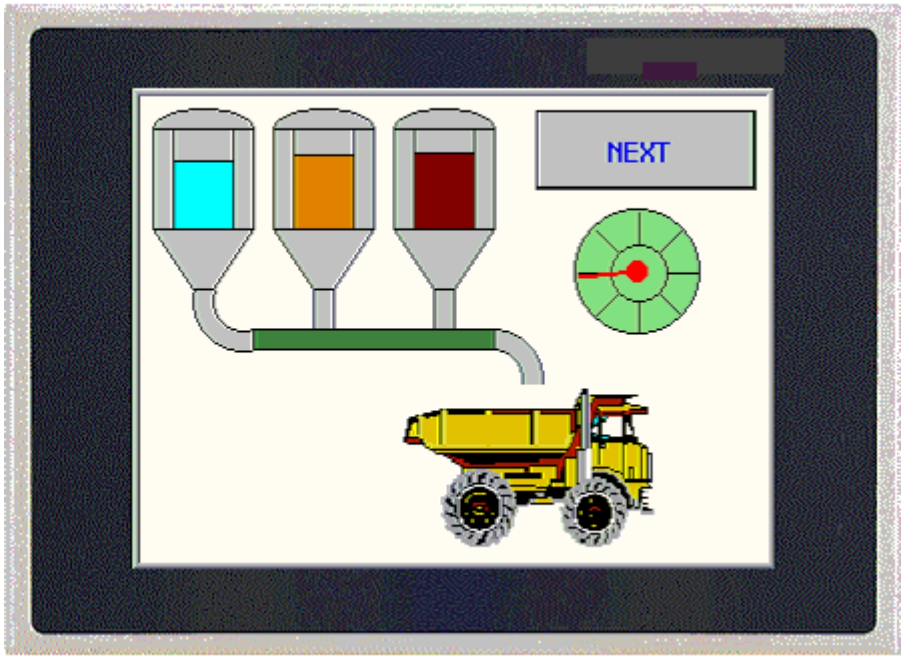
In the menu select **Tools|Compile…** to compile the project.

Step 5. Simulate the project either On-line or Off-line.

Select **Tool|On-line Simulation** operation in the menu. After the project is complied, the Simulator retrieves data from the PLC through the unit and simulates operation. Before simulation, be sure to connect the unit and the PLC properly and set the COM port in EasyManager.

Select **Tool|Off-line Simulation** operation in the menu. After the project is complied, the PC simulates the PLC and emulates operations. The unit does not have to be connected to the PC for this type of simulation. **Hint**: Use this type of simulation to demonstrate projects.

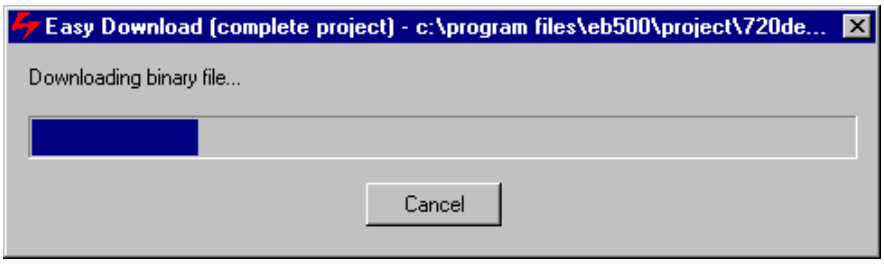
Using the Simulators before the project is completely debugged, saves time by avoiding repeated downloads.

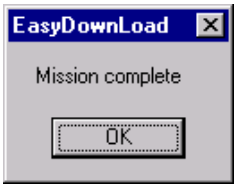


Sample Simulation Screen

Step 6. Download the project

In the menu, select **Tool|Download…** to load the image file into the MMI. Before downloading, be sure to set the COM port in the Easy Manager menu. When the download is finished, click **OK** in response to the Mission complete message. The HMI automatically switches to Application Mode.





This ends the Quick Start section. From here, it is up to you to design the spectacular window displays that make life easier for your operators.

Section 2: Software Reference Guide

6.0 EasyManager Operations

The EasyManager is a software shell for launching several utilities. Some functions are duplicated in the EasyBuilder screen-editing program. While the EasyBuilder program can be run as a stand-alone program, it is advised to run EasyManager first to set the COM port.

EasyManager has the following components:

6.1 COM Port Drop-Down Box

Select the number of the RS232 Serial COM port for communications to the MMI. Ports **COM1** through **COM10** are available for selection.

Note: Make sure that some other program is not already using the selected COM port. EasyManager and EasyBuilder will not take control of a COM port that is used by another program. Communications time out errors will be displayed when attempting EasyManager functions.

6.2 Communications Speed Drop-Down Box

This selection is used to determine the communications speed between the PC and the unit during downloads and uploads. If you are using the MT5_PC cable to download, the **115200** speed is recommended. When using extension cables, the **38400** speed is recommended.

6.3 Project or Recipe Download/Upload

There are two types of communications: Downloading - sends data to the unit and Uploading - retrieves data from a unit. The data that is sent or received can be a compiled project's object file (file with *.eob extension) or retentive memory (recipe) data (file with *.rcp extension).

Select **Project Download/Upload** to transfer only project data.
Select **Recipe Download/Upload** to transfer only retentive memory (recipe) data.

6.4 Complete or Partial Download/Upload

The Complete or Partial Download/Upload is used to determine how much retentive memory (recipe) data is transferred to or from the unit during Download or Upload.

A **Complete Download/Upload** sends/retrieves the full 65535 bytes of retentive memory (recipe) data from the MMI.
A **32K Partial Download/Upload** sends/retrieves only the first 32767 bytes of retentive memory (recipe) data.

6.5 EasyBuilder

This button launches the **EasyBuilder** screen editor. This is used for project creation, editing, compiling, simulation and downloading.

6.6 Online-Simulator

The Online simulator retrieves data from the PLC through the unit and uses it to put a realistic image of the unit on the PC screen. The image shows how a project appears after download. Use the Online-Simulator to view demonstration projects and show how the windows look in actual operation. A Touchscreen and PLC are needed to simulate a project in this mode.

Note: The project must be compiled by EasyBuilder. This creates a file with extension *.eob. Uncompiled project files with the *.epj extension cannot be simulated.

Note*:* When the **Direct Online-Simulator** option is checked the unit is not needed for Online simulation, the PC can be connected directly to the PLC with an appropriate communications cable.

6.7 Direct Online-Simulator

When this option is checked the HMI is not needed for **Online-Simulation**. The PC is connected directly to the PLC with an appropriate communications cable.

6.8 Offline-Simulator

The Offline simulator emulates the operation of a project on the PC screen. The PC is used to simulate a PLC in a limited fashion. Use the Offline-Simulator to create demonstration projects or view approximately, how the windows look in actual operation. A Touchscreen and PLC are not needed to simulate a project in this mode.

Note: The project must be compiled by EasyBuilder. This creates a file with extension *.eob. Uncompiled project files with the *.epj extension cannot be simulated.

6.9 Download

Download a project compiled by EasyBuilder to the MMI. Downloading a Project sends the object *.eob file and or recipe *.rcp file to the MMI. Downloading a Recipe sends only the recipe *.rcp file to the MMI. You are prompted to select the files as they are downloaded.

Note: The EasyManager download facility is ideal for sending out project updates. Most projects and EasyManager fit on a 1.44MB floppy disk. The user just needs to run EasyManager click on download and select the project file. The download begins automatically.

6.10 Upload

Upload the project file from HMI to an object file (*.eob). Uploading a Project retrieves the object *.eob file and or retentive memory (recipe) *.rcp file from the MMI. You are prompted to name the files as they are uploaded. Uploading Recipe retrieves the retentive memory (recipe) *.rcp file only from the MMI. You are prompted to name the file as it is uploaded.

Note: The upload file can be transferred to another MMI. This is useful for retrieving data from installed units to make a duplicate setup.

6.11 Mode Change

The EasyBuilder operates in one of three modes. Click the buttons to switch to the corresponding mode.

Note: It is not necessary to change dipswitches for a mode change.

6.11.1 **RDS** (**R**emote **D**ebug & **S**imulation) mode

The **Jump to RDS** mode is used for project development. Simulations and Up/Downloads are done in this mode. It puts the unit in a state that displays 9 to 12 lines of unit information.

Line 0: Shows the status of the unit as reported from its internal initialization routine. This should say “Initial OK”. If this line does not display the OK message, please contact the factory for assistance.

Lines 1 and 2: Show the unit’s internal firmware version information. The BootRom Version should be 1.12 with a Version Id of 0x100a or greater for proper operation with EasyBuilder V2.6.x.

Line 3: The PC communication baudrate setting. This changes automatically to match the EasyManager setting.

Line 4: Shows if the unit has a project in it or not. If no project is present, the unit does not go to Application mode.

Line 5: The download support setting. This changes automatically to match the EasyManager setting.

Line 6: This should always be Fast Mode Simulation.

Line 7: Indicates if Retentive (Recipe) memory is installed in the unit. (Old units may not have memory installed.)

Line 8: Show the unit’s driver status. The unit should indicate that it accepts ORMEC drivers. If it does not, please contact the factory for assistance.

Line 9: Shows the display resolution.

Line 10: Shows the unit’s MAC ID for Ethernet communications. (Old units do not display this information.)

Lines 11 & 12: Displays the OEM’s copyright notice.

Note: On new hardware platforms, below line 12, are two touch areas for contrast adjust and or a button for initiating Compact Flash project transfer.

6.11.2 **Application** (On line operation) mode

Clicking **Jump to Application** runs the project in memory. This is the normal operating mode of a unit once it is installed, programmed and ready for use.
Note: if there is no project in the unit the unit stays or jumps to RDS Mode.

6.11.3 **Touch Adjust** (Touch screen calibration) mode

Jump to Touch Adjust is used to calibrate the touch screen. Normally the touchscreen is calibrated at the factory for proper functionality. If the touchscreen cannot be calibrated, please contact the factory for assistance.

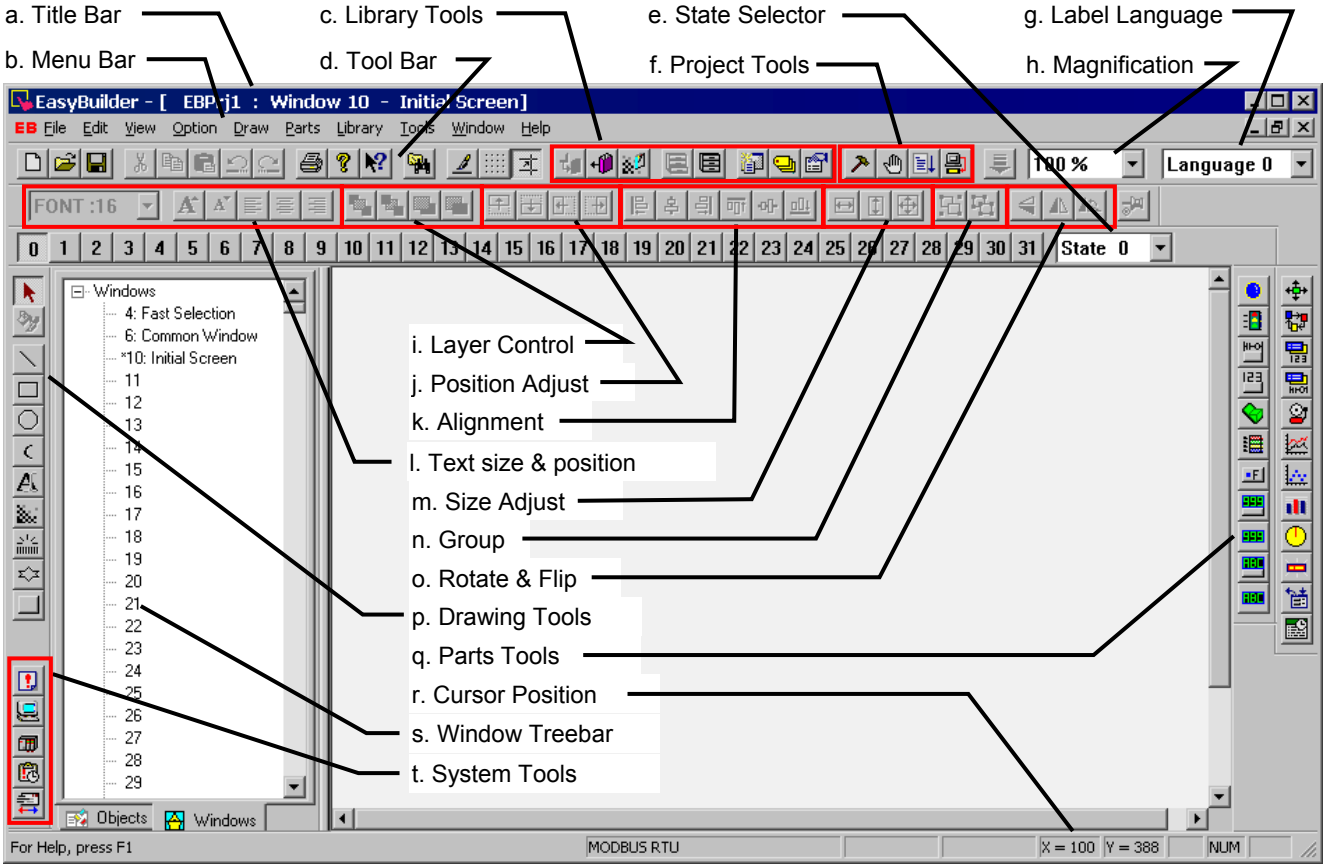
6.12 Exit

This button closes the EasyManager application.

7.0 Software Fundamentals

7.1 Screen Editor Overview

Clicking on the Easy Manager's [EasyBuilder] button causes the following window to appear. The functions of each screen area are explained below.



a. Title Bar: Displays the project's file name, active window number and title.
b. Menu Bar: Displays the menu used to select EasyBuilder commands.
c. Library Tools: Displays tools for library operations.
d. Tool Bar: Displays the icons corresponding to File, Edit, Option, Library, Tools and Help menu items.
e. State Selector: Clicking on this drop-down toggles all the parts on the window to the specified state.
f. Project tools: Used for compiling, simulating and downloading a project.
g. Language: Clicking on this drop-down toggles all the labels on the window to the specified language.
h. Magnification: Used to double screen size for fine editing and positioning.
i. Layer Control: Adjust the Layer of selected part – one layer up, one layer down, to top layer and to bottom layer.
j. Position Adjust: Adjust the position of selected parts.
k. Alignment: Make all the selected parts line up to the top, bottom, left or right.
l. Text size & position: Change the font size and alignment of the selected text.
m. Size Adjust: These tools make the dimension, width or height, of all the selected parts the same size.
n. Group: The group function makes a collection of selected parts and drawing elements as one object.
o. Rotate and Flip: Allows shapes to be flipped horizontally or vertically and rotated in 90-degree increments.
p. Drawing Tools: Icons representing each drawing tool are shown.
q. Parts Tools: Icons representing each Part type are shown. Clicking on one of these icons causes that icon's dialog box to appear. That Part's attributes can then be set and the Part can be placed on the window.
r. Cursor Position: Shows the current cursor position on the active screen relative to the upper left corner of the screen.
s. Window Treebar: Provides quick access to Window properties, objects and organization.
t. System Tools: Used to set Alarm and Event messages, PLC controls and Data Transfer.

7.1.1 Changing Screen Appearance

The screen appearance can be changed to make project editing easier. This includes what toolbars are displayed, changing grid constraint properties and magnification. These changes affect the screen but do not change a window's properties.

7.1.1.1 View Menu

The View menu is used to Hide or show Toolbars, Status bar and Treebar. The functionality of the Treebar is discussed further in the Editing Placed Objects section. Toolbars can be dragged to other sides of the editing window or dragged away from the edge to be shown as floating. Check a toolbar's name to open it. Uncheck a toolbar's name to close it.

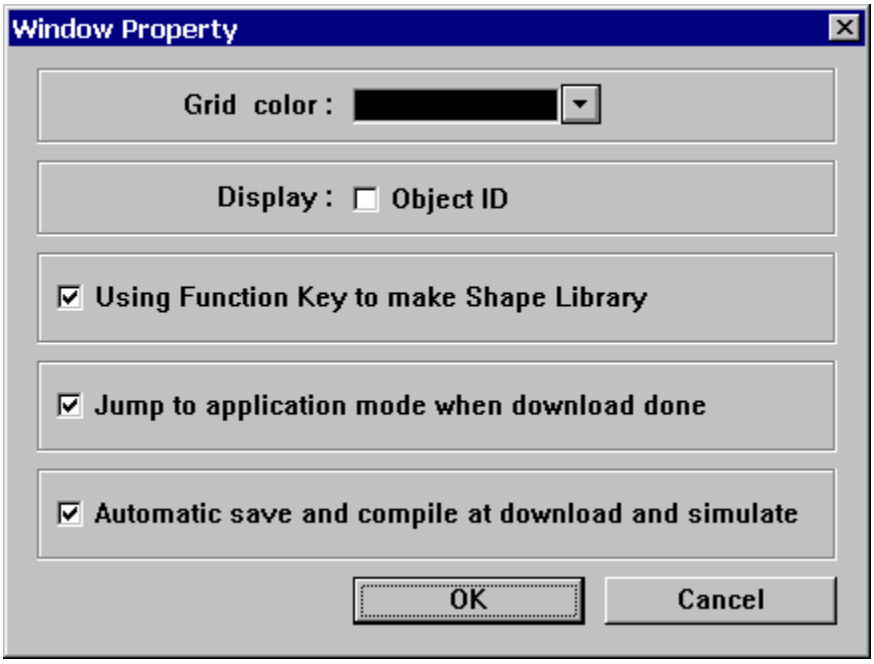
The **Size** command is used to change the edited window's magnification to 100 or 200%. Magnification makes it easier to align objects manually and to make drawing objects intersect at appropriate coordinates.

7.1.1.2 Screen Settings

Use the **Option|Grid/Snap** menu to make object alignment and drawing easy.

X, Y Select or enter the pixels from one grid point to the next in the X(Horizontal) or Y(Vertical) direction.
Display Checkbox: Enables/Disables grid visibility.
Snap Checkbox: Enables/Disables the snapping of objects to the grid.
Fix objects checkbox: Enables/Disables the anchoring of objects once they are placed. This prevents inadvertent movement of objects when selected by mouse click or rearrangement.

Use the **Option|Window Property** dialog to set the Grid color, turn the object ID display on or off and sets project save and compile automation features.



Grid color: Select the color of the grid.
Note: If the grid color is the same as the window color, it becomes invisible.

Display Object ID Checkbox: Enables/Disables the object ID number from being displayed in the upper left corner of an object.
Enable the Object ID for easy reference to objects listed in the Treebar.
Disable the Object ID to view how objects appear when displayed on the unit's screen.
Note: Object ID's may obscure the locations of object labels.

Without Object ID


With Object ID

Using Function Key to make Shape Library Checkbox: Enables/Disables the automatic creation of a shape using a function key assist. This is discussed further in the **Library Operations** Section.

Jump to application mode when download done Checkbox: Enables/Disables automatically forcing the HMI to the application mode after a download. This avoids the process of going to EasyManager and pressing the Jump to Application button or cycling power to the unit after download to put it in application mode. This is discussed further in the **EasyBuilder Operations** Section.

Automatic save and compile at download and simulate checkbox: Enables/Disables automatically saving and compiling a project before simulation or download. This shortens the usual 3-step process to one step. This is discussed further in the **EasyBuilder Operations** Section.

The **Option|Language** menu sets the program for 1 or 2 byte font character sets. The particular font set is specified in the **Edit|SystemParameters/Editor** Tab.
Note: The English version of EasyBuilder only supports single byte fonts as defined by **EasyAsciiFontMaker.exe**.

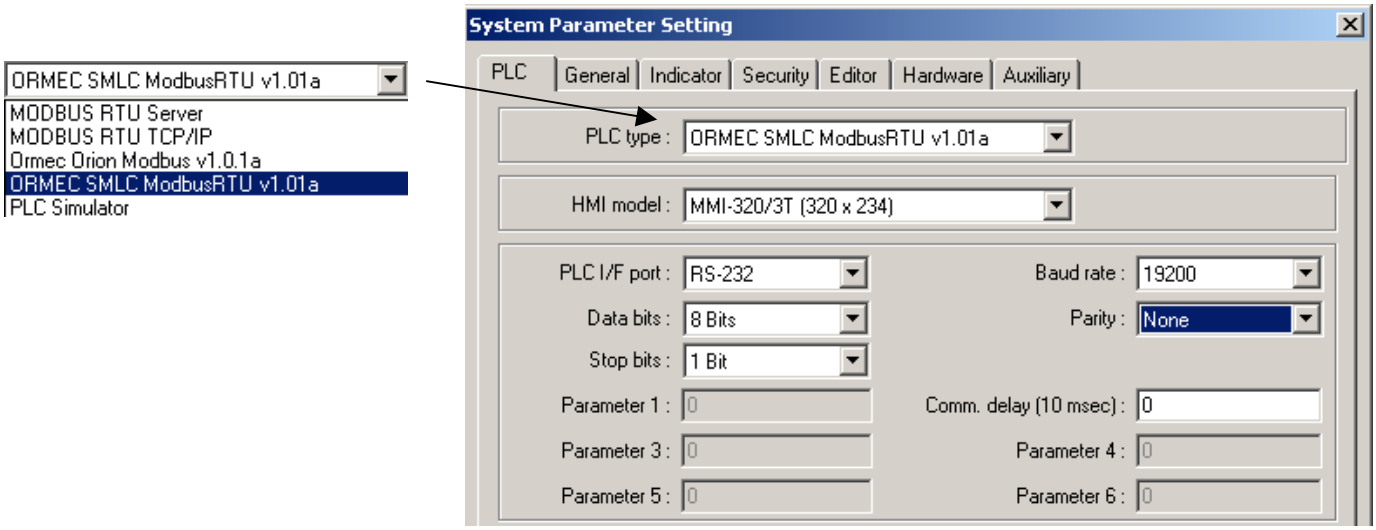
7.2 System Parameters

The first thing that should be done when starting a new project is to select the System Parameters for that project. The System Parameters determine all of the characteristics of the project as it relates to the Operating System, Window appearance and PLC.

Select the **Edit|System Parameters…** menu and the **System Parameter Setting** dialog appears. Fill in all or some of the system parameters as required.

7.2.1 The PLC Tab Parameters

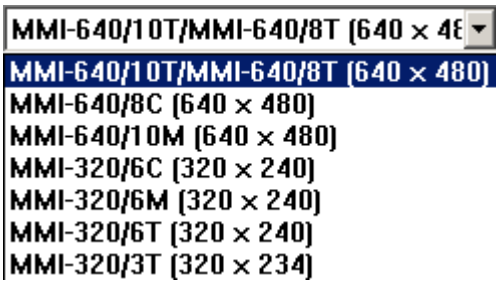
PLC type: Select the type of PLC from available PLC selection menu. The dropdown displays all of the built in PLC drivers available. Additional external "add on" drivers located in the **\Drivers** - SubDirectory are also displayed with the [PDS] suffix. Select the driver that matches your PLC's protocol.



Note: For additional information about individual drivers, see the **Controller Reference Guide** section.

MMI model: Select the unit by model number from the dropdown.

ORMEC Part#	Display Type
MMI-640 / 10T (640 x 480)	10.4" TFT 256 Color
MMI-640 / 10C (640 x 480)	10.4" STN 256 Color
MMI-640 / 10M (640 x 480)	10.4" STN Mono - Grayscale
MMI-640 / 8T (640 x 480)	7.7" TFT 256 Color
MMI-640 / 8C (640 x 480)	7.7" STN 256 Color
MMI-320 / 3T (320 x 234)	5.7" TFT 256 Color
MMI-320 / 6T (320 x 240)	5.7" TFT 256 Color
MMI-320 / 6C (320 x 240)	5.7" STN 256 Color
MMI-320 / 6M (320 x 240)	5.7" STN Mono - Grayscale



Notes:

Changing model type effects how windows are displayed.

EasyBuilder does not scale windows when display size is changed.

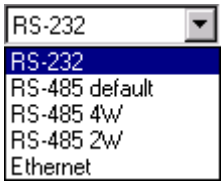
Example: Changing from a 640x480 to a 320x240 display clips the windows so that only the upper left corner is shown. Changing from a 320x240 to a 640x480 display shows the window's objects in the upper left corner.

EasyBuilder does not compensate colors so that the display is readable when changing from Color Models to Grayscale Models.

We recommend that you develop a project for its target hardware platform.

(There is no utility to automatically resize a project for a larger or smaller screen.)

Serial Port I/F: Select the PLC port's type of hardware communications. The HMI activates the corresponding port and configuration on the back of the unit.



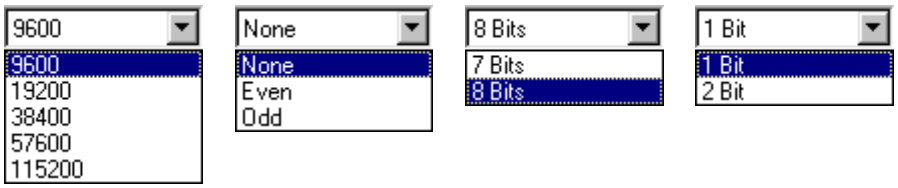
Serial Port I/F	Description
RS-232	Communications to the controller are through the **PLC[RS-232]** port. (Pins 2,3 & 5)
RS-485 default	Communications to the controller are through the **PLC[RS-485] PC[RS-232]** port. This can be 2 wire or 4 wire depending on the PLC type. When unsure select **RS-485 4W** or **RS-485 2W**. (Not supported by SMLC)
RS-485 4W	Communications to the controller are through the **PLC[RS-485] PC[RS-232]** port. (Pins 1~5) (Not supported by SMLC)
RS-485 2W	Communications to the controller are through the **PLC[RS-485] PC[RS-232]** port. (Pins 1,2 & 5) (Not supported by ORION or SMLC)
Ethernet (Option)	Communications to the controller are through the optional **Ethernet** port.

Note: RS-485 communications can be forced to 4 wire or 2 wire types. This is practical when communications adapters are used to connect several devices to one PLC. If left at the default, it uses the PLCs default configuration.

Note: For additional information about communication configurations for a selected PLC, see the **Controller Reference Guide** section that refers to that PLC.

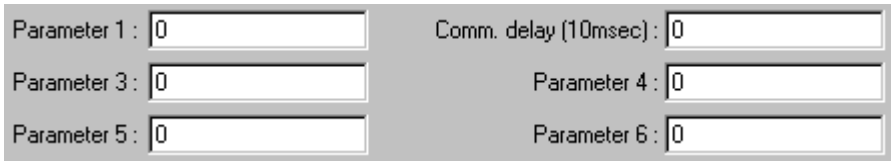
Note: The **PLC[RS-485] PC[RS-232]** port is a dual port and serves as the programming port for the MMI. This port is also used for Master/Slave communications (see below).

Baud rate, Parity, Data bits and Stop bits: Set the baud rate to match the PLC ports settings. These are disabled if Ethernet communication is selected. (Ethernet communication uses 10 BaseT and TCP/IP parameters.)



Comm. Delay and Additional Parameters:
The Comm. Delay parameter is used for setting an interval between communications requests to the controller. Enter a number from 0 to 999. The number specifies the 10-millisecond intervals between communications. For example: a setting of 100 puts a 1-second interval between communications requests. This parameter will slow data update rate on a window. It also gives slow controllers a way to meet processing demands without too many interruptions.

Set the other additional parameters as required by your driver. See the **Controller Reference Guide** section that references the PLC driver for further details.



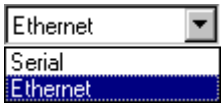
HMI station No.: This is used when PLCs require the HMI to have a node or station identifier. Set as needed or leave at 0 if not used. Station numbers are 0-255 (Use the range as appropriate for PLC type).

PLC station No.: Used when PLCs have a node or station identifier. The HMI needs the station number to initiate communications. Set as needed or leave at 0 if not used. Station numbers are 0-127 (Use the range as appropriate for PLC type). If the PLC station number does not match the PLC, the “PLC no response” error message is displayed. Station numbers above 127 (127~255) are reserved for the Auxiliary Port.

Multiple HMI: Allows more than one HMI to be connected to one PLC. Enable as a Master or a Slave, depending on connection, or Disable as needed.

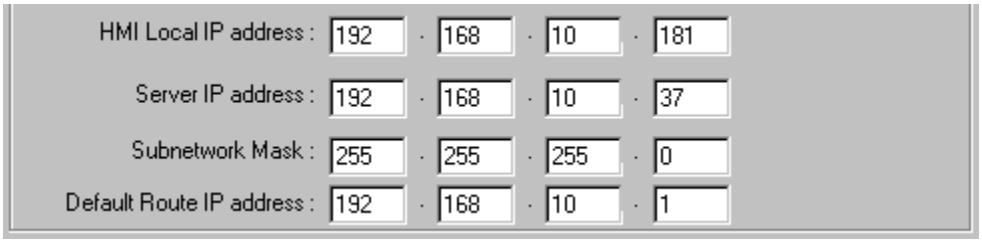
Selection	Description
Disable	Disables the chaining of multiple MMIs to one PLC.
Slave	Select this if this HMI connects to another HMI in the chain.
Master	Select this when the HMI is the unit connected directly to the PLC.

Connect I/F: Select the unit to unit connection type. **Ethernet** is available for units with the optional Ethernet port. For all others, use the **Serial** interface (I/F) selection.



HMI-HMI link speed: This is used only when a **Serial** interface is used for connecting multiple HMIs. Higher communication rates may cause faster window updates, but are diminished quicker by link distance and electrical interference.

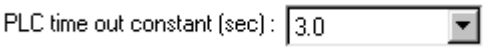
Note: All linked HMIs must be set to the same link speed. **Slave** units ignore the communication parameters associated with the PLC (Serial port I/F, Baud rate, Parity, etc.).



Ethernet IP settings: These settings are used whenever Ethernet communications is selected for Multiple HMI or HMI to PLC connections. **A working knowledge of TCP/IP networking and terminology is necessary to implement Ethernet connections.**

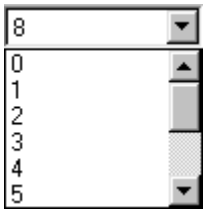
Setting	Description
HMI Local IP address	These fields are the IP address of the HMI unit.
Server IP address	The IP address of the PLC or Slave HMI.
Subnetwork Mask	The mask for the network where the HMI and PLC or Slave are located.
Default Route IP address	This is used for HMI to PLC communications. It refers to network server IP settings.

PLC time out constant (sec): This setting determines how long the HMI waits for a response from the PLC. The range is from 0.1 to 25.5 seconds. This setting is important when the PLC is normally slow to respond or several MMIs are linked to one PLC.



Note*:* When communications are broken, the red COM LED on the front Panel of the HMI flashes at this interval.

PLC block pack: Used to determine how the unit communicates to the controller. By increasing this number, larger blocks of registers can be fetched from the controller. In some cases, this speeds the update of information on the display. The range is 0 to 255.



Note: Setting the Block Pack to 0 allows the HMI to determine the block size that best fits data retrieval. This is the recommended setting.

Note: Certain PLCs have a limit as to how many data points can be uploaded at a time. If the Block pack number is set higher than this limit, it has no effect on communications.

Hint: When creating project windows, it is advised to use consecutive registers whenever possible.

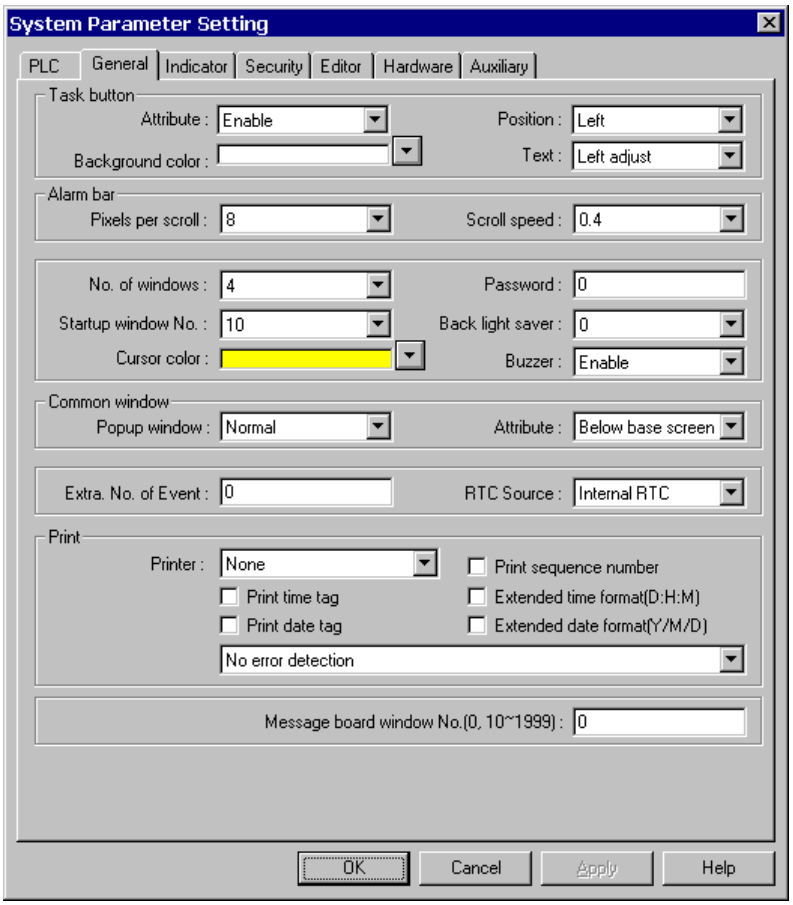
Note: Some of the project's **System Parameter/PLC Tab** settings are stored in the retentive memory of the MMI. See the **System Bit and Register Reference** section for further details.

7.2.2 The General Tab

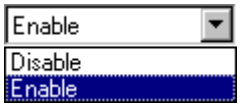
Task button

The Task button is used to pop up the **Fast Selection** window or display the **Task Bar**. Minimized window's icons are put on the **Task Bar**. This gives the project a familiar graphical user interface.

Note: Refer to the **Task Bar Operations** section for a full description of the Task Button features.

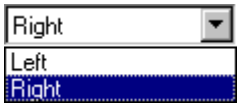


Attribute: Enables or disables this feature. If disabled, the **Fast Selection** window and **Task Bar** are not available at run time.

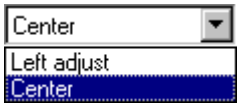


Background color: This dropdown calls the color selection dialog. Select 1 of 16 colors for the Task Bar background.

Position: The Task buttons can be located on the right or left side of the display.



Text: Determines text alignment within the Task Buttons and minimized window icons.

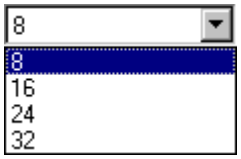


Left adjust: Shifts text to the left side of the Task Buttons and minimized window icons.
Center: Centers the text to the center of the Task Buttons and minimized window icons.

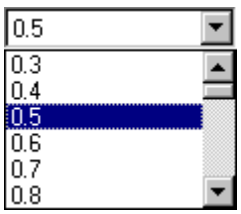
Alarm Bar

The **Alarm bar** Part displays alarm text in single line scrolling fashion. These settings affect how the text is scrolled.

Pixels per scroll: Select **8**, **16**, **24** or **32** from the drop-down menu. This specifies how many pixels are scrolled in each increment. For example: If the Pixels per scroll were set at 8 and the font size of the Alarm Bar was set to 24 (characters are 12 pixels wide), 2/3 of a character would be scrolled in with each elapse of the Scroll speed time. The larger the number the faster a given message is displayed.

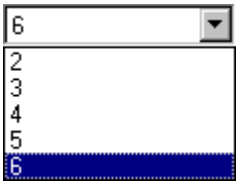


Scroll speed: This setting determines how fast each increment of the scroll is displayed (Range: 0.0 to 25.5). The setting is the time between characters. The smaller the number, the faster the alarm messages are displayed. 0.0 is the fastest with no break between characters.



Note*:* Text blurs at high speeds!
Note: Refer to the Object Reference Guide for more details about Alarm Bar operation.

No. of windows: This setting is used to specify the maximum number of windows allowed open at any one time (1 - 6). The unit ignores attempts to open more windows than specified until an existing window is closed.



When more windows are open than recommended, system resources may drop low enough to trigger a "System severe error" message". When the error occurs, the unit must be reset or power must be cycled to the unit before project execution can resume.

Note: The maximum number allowed is 6. It is recommended to reduce the maximum by one for each of the following conditions:

- The project has a Common window.
- The project has a Message board window.
- The project has print screen functions.
- The project is using bitmap compression.

If all of the above conditions apply, restrict the maximum number of windows allowed to 2.

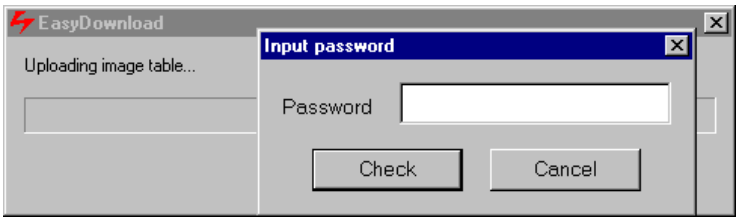
Note: The size of the minimized icon on the **Task Bar** is decreased as the minimum number of windows is increased. By selecting a lower number, more room is given for a descriptive title in the minimized icon.

Startup window No.: This is the window displayed when the HMI is powered up. Valid window numbers are 10 - 1999.
Note: If the specified window does not exist, the unit displays "System severe error (error code 10101)" when started.

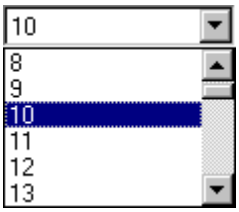
Cursor color: Determines the color of the cursor when displayed. The full range of colors is available for this parameter. This is used with Data Input Extend and ASCII Input Extend parts. The cursor (a flashing block) indicates when data entry is enabled for a selected value.

Password: Activate the password option by entering any number greater than 0. This locks the project after it is downloaded so it cannot be uploaded without first entering the password.

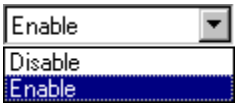
Note: Password protection does not prevent the user from Downloading. There is no protection to prevent a project from being overwritten.



Back light saver: The HMI turns off back light power if there are no touch operations within the set time (Range: 0 to 255, unit in minutes). A zero setting disables the back light “auto shutoff” function.



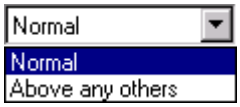
Buzzer: The buzzer sounds briefly every time the touchscreen is activated by touch. This selection allows the programmer to turn off the buzzer. The buzzer can be annoying in some applications and during development phases of a project.



Note: The buzzer cannot be assigned to alarms and is not associated with alarm functions.
Note: There are no provisions for the user to control the buzzer.

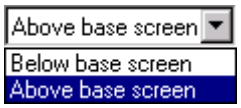
Common Window: Window 6 is the default common window. Objects placed on a common window are always displayed and active. The following settings determine how they appear.

Popup window: Determines where an indirect or direct popup window called from the common window is displayed.



Normal: The Popup window is displayed on the layer above the parent-calling window. If another window is on top of the calling window, it may hide the popup.
Above any others: The popup window is displayed on the top layer when activated.

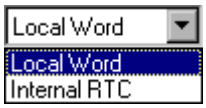
Attribute: Determines the location of the Common window.



Below base screen: The common window is always on the bottom layer of the display. All windows cover the common window.
Above base screen: The common window is always in the top layer. All windows are below the common window. Controls and objects on the Common window are always visible.

Extra No. of Event: Normally, 200 events are stored in the Event Log. If more than 200 events are needed, the additional amount is entered here. Up to 2800 additional logs can be added. For example, if 1000 events need to be logged, enter 800 in this field. See the **Event Log** part for details on the Event Logger.

RTC Source: The RTC Source determines where the Event Log part and Printer obtains the real time information for their display and printout.

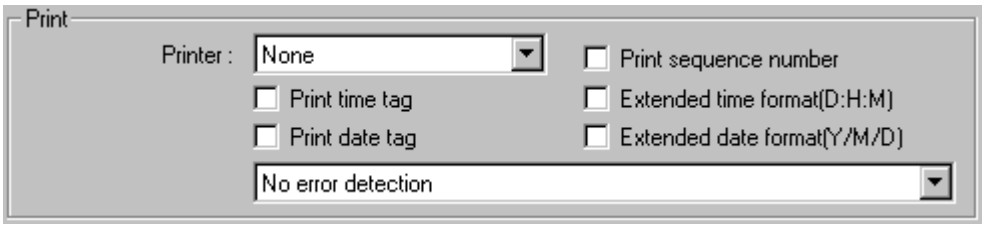


Local Word: Uses the values in **LW 9010** to **LW9016**. The RTC data from your PLC must be transferred to **LW9010** to **LW9016** before use. This is done with a **Data Transfer** part.
Note: All values must be BCD coded.

LW Address	Description	Valid Values
9010	Real Time Clock second.	0 - 59
9011	Real Time Clock minute.	0-59
9012	Real Time Clock hour.	0-23
9013	Real Time Clock day.	0-31
9014	Real Time Clock month.	0-11
9015	Real Time Clock year.	0-9999
9016	Real Time Clock day of the week.	0-6

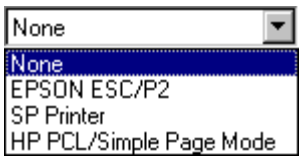
Internal RTC: Uses the RTC in the HMI (**RW60000** to **RW60006**). All values are BCD coded.

RW Address	Description	Valid Values
60000	Real Time Clock second. (read and write allowed)	0 - 59
60001	Real Time Clock minute. (read and write allowed)	0-59
60002	Real Time Clock hour. (read and write allowed)	0-23
60003	Real Time Clock day. (read and write allowed)	0-31
60004	Real Time Clock month. (read and write allowed)	0-11
60005	Real Time Clock year. (read and write allowed)	0-9999
60006	Real Time Clock day of the week. (read and write allowed)	0-6



Print: Use these settings to set the Printer protocol. This enables Function Key Screen and Event printing as well as the display of printer errors. Time format for Event printing is also set here.

Printer: Select the printer protocol.

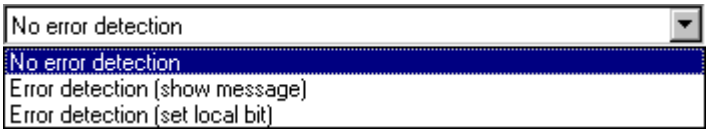


None: Printing disabled.
EPSON ESC/P2: Typically used with dot matrix line printers.
SP Printer: Simple printing used with ASCII type printers.
HP PCL/Simple Page Mode: Use with PCL compatible laser printers.

Note: Local Bit, LB 9017, is the control bit for printing. If LB9017 is set OFF/ON, printing is disabled/enabled.
Note: If **None** is selected, <u>all</u> print functions and LB9017 are disabled.

Print time tag: Print the time in Hours:Minutes:Seconds format along with the printed information.
Print date tag: Print the Date in Month:Day format along with the printed information.
Print Sequence number: Select to print the sequence number of the event along with the event occurrence.
Extended time format (D:H:/M): Change the time format in the time tag to Days/Hours/Minutes.
Extended date format (Y/M/D): Change the time format in the time tag to Year/Month/Day.

Error Detection: Select how printer errors are handled. Local Bit LB 9016 is used for internal error reporting.



Setting	Description
No error detection	Printer errors are ignored.
Error detection (show message)	A message box pops up on the HMI display when printer errors occur.
Error detection (set local bit)	LB 9016 is set to 1 when printer errors occur. LB 9016 remains high until reset by some user action. It is up to the programmer to handle this error by popping up a window or providing a way to reset the bit.

Message board window No. (0,10~1999): The message board allows the operator to draw a message within the designated window, using the touchscreen. This is especially useful when the operator needs to leave a note about machine status or the operator's whereabouts when not at the machine. The operator can simply write "MACHINE DOWN" or "Lavatory". Or, some process note or calculation can be entered as a reminder; such as "Backlash +.005". The message window retains its data until power is lost. This allows the message window to be closed and then opened later without loss of information.

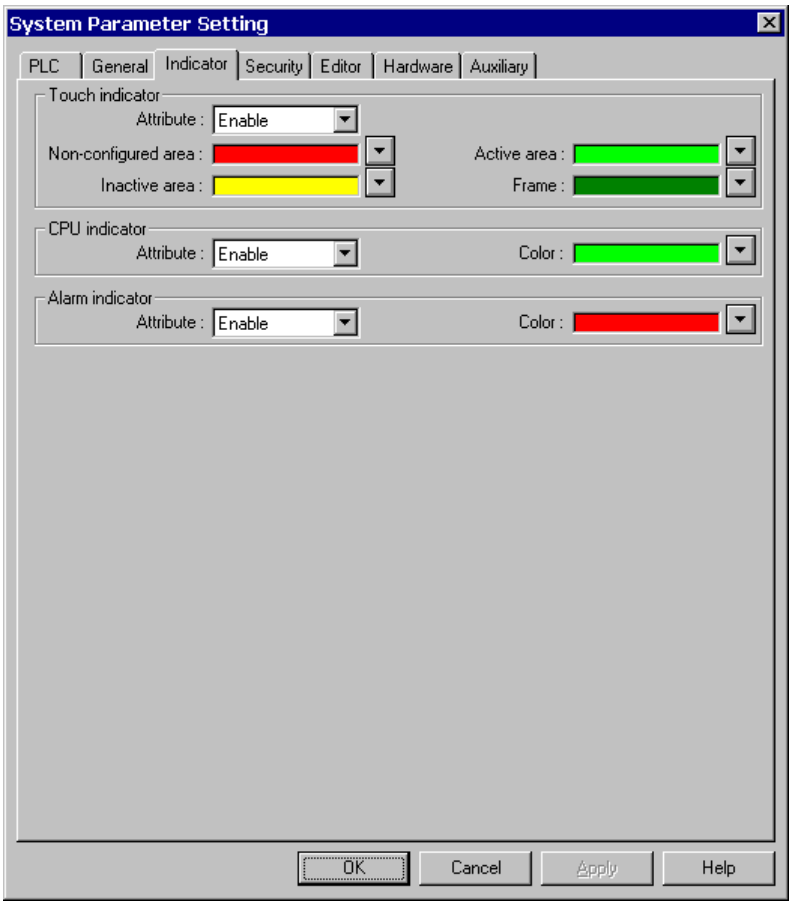
Select the window number to be used as the message board.
Note: Selecting 0 disables the Message board feature.
Note: Only one Message board window per project is allowed.
Note: See the **Function Key** part section for information about Message board controls.

Note: Some of the project's **System Parameter/General Tab** settings are stored in the retentive memory. See the **System Bit and Register Reference** section for further details.

7.2.3 The Indicator Tab

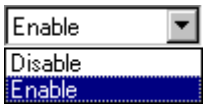
These are settings for the indicators shown on the Task Bar opposite to the Task Buttons. 

Touch indicator: The Touch indicator changes color every time a screen touch is recognized. This allows the operator to visually confirm that the touchscreen has recognized the touch and is functioning properly.



Attribute:

Enable makes the Touch indicator visible and active. **Disable** makes the indicator invisible and not active.

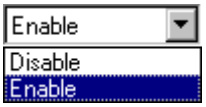


Non-configured area, Active area, and **Inactive area:** Use these to set the condition's color that is displayed in Touch indicator.

Frame: Use to set the color of the circular outline in the Touch indicator

CPU indicator: The CPU indicator is a bargraph that shows system resource usage. Typically, the bargraph increases as more windows are open. Operators can tell if a resource failure is imminent.

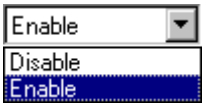
Attribute: Enables or disables this feature. If disabled, the CPU indicator is not displayed at run time.



Color: This dropdown is used to select the color of the displayed CPU indicator.

Alarm indicator: The Alarm indicator comes on when there are alarms present. This indicator is a bargraph that increases as the number of alarms increases. Operators are alerted to active alarms.

Attribute: Enables or disables this feature. If disabled, the Alarm indicator is not displayed at run time.



Color: This dropdown is used to select the color of the displayed Alarm indicator.

Sample Task Bar showing Indicators.

Note: The full range of colors is available for Indicator settings.

7.2.4 The Security Tab

The **Security** tab is used for setting security levels and access codes.

Note: Reference the **Security** section for further details on Security Control.

Security Control This check box activates the security feature. Security levels are assigned to windows in their Window Settings dialog. There are three levels of security.

Lowest - Level 0 is the lowest level of security.

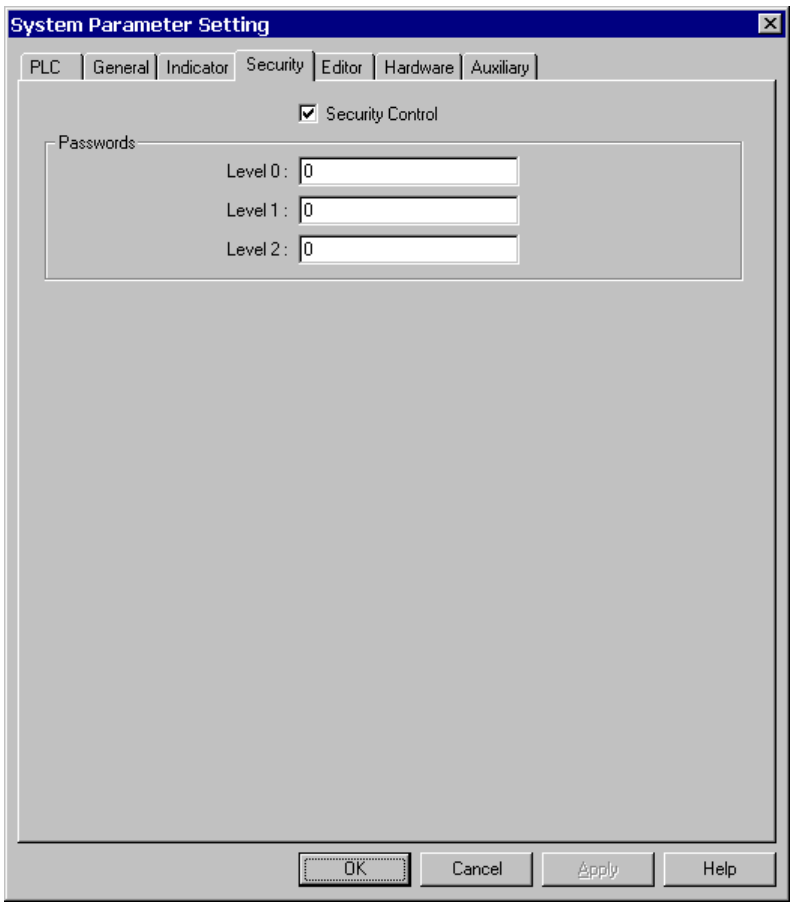
Middle - Level 1 allows access to Level 1 and Level 0 windows.

Highest - Level 2 is the highest level of security and can access all levels.

Each level's password is a double word value (0 to 99999999). Passwords are numeric digits only and cannot contain any alphabet characters.

Note: A password of 0 for a Security level disables security for that level (all users can access that level).

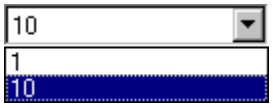
Note: The project's Security level settings are stored in the retentive memory of the MMI. See the **System Bit and Register Reference** section for further details.

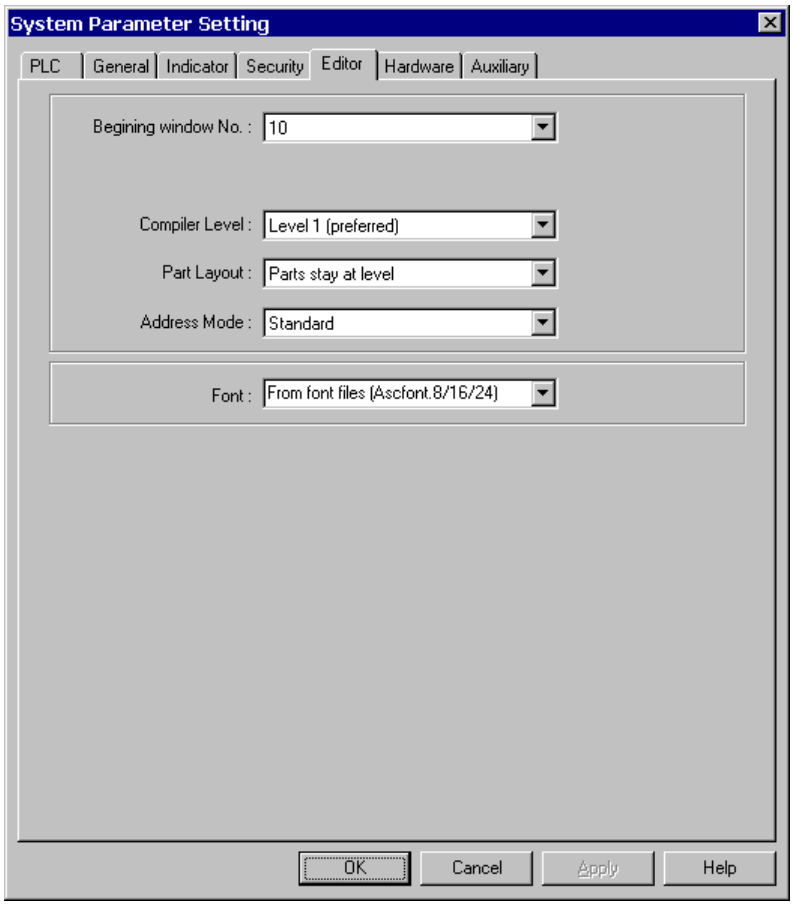


7.2.5 The Editor Tab

The **Editor** tab allows the programmer to change the project characteristics.

Beginning window No.: Changes the window numbering so it starts with 1 instead of 10.

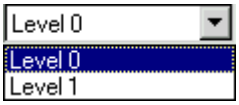




Internally, EasyBuilder maintains the initial window of 10 but displays all window numbers with an offset so they appear to start with an initial window of 1.

> ***Note:*** System windows that start with a number less than 10, such as the Fast Select window, now appear with negative numbers in the Window Treebar.
>
> ***Note:*** Changing this does not affect the window numbering for specifying the **Startup window** setting in the Set System Parameters General Tab.

Compiler level: This selection is used to determine the compile type. Older projects did not have any checks for PLC address type. Use Level 0 to recompile older projects and Level 1 for all new projects. Old projects work on new units.



Part Layout: This selection determines what happens when a Part changes state, particularly when the Part is in a lower layer, possibly covered by another window or object.
Select **Control** to have the part move to the topmost layer and become visible.
Select **Nature** to have the part stay in its original layer, still changing state, though partially or not visible.

Address Mode: This selection is used to enable Read and Write Device address fields to accept Extended notation. Extended notation is used when a touchscreen is connected to more than one controller. This is used with RS485/422 communications schemes.

> ***Note:*** The selected PLC in the Set System Parameters PLC Tab must have multiple PLC communications available. For example: PLC type Modbus (485 2 wire) The addressing method is as follows:

Device type	Address	Description
0x:	1#06	PLC 1, Data Bit 00006
4x:	2#45	PLC 2, Holding Register 40045
3x:	12#54	PLC 12, Input Register 30054

Number before '#' is the Station number: 0~255

Font: The font selection drop-down shows the selection of available fonts. Font selection is only used for Asiatic Languages. Font selection does not affect Western Languages. The fixed font files in EasyBuilder are always used.

Note: The Font selection works in conjunction with the **Option|Language** menu's selection.

Option|Language: This setting determines the character set that is used in project development.

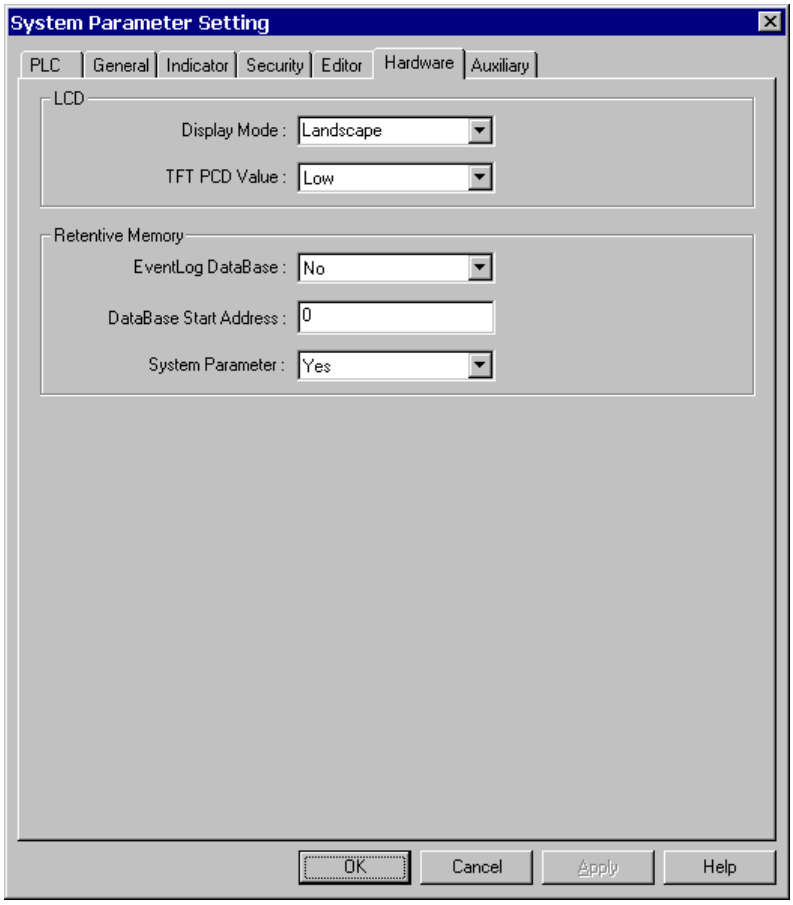
Select **Single Byte** for European languages. Single Byte uses extended fonts (ALT key sequences) to type characters used in some European countries. Single Byte uses the built in fonts. These fonts are in the directory as Ascfont.8, Ascfont.16 and Ascfont.24. These represent the 8, 16 and 24 point sizes. Larger sizes are generated from these base sizes (Example: Font size 32 is actually size 16 doubled).

Select **Double Bytes** for Asiatic languages. Use the Font setting in the System Parameters Editor Tab to select the appropriate character set.

Note: The fixed western fonts can be edited using the **EasyAsciiFontMaker** editor included with the software. Use **EasyAsciiFontMaker** to edit the Font files if a character is not found.

7.2.6 The Hardware Tab

Use the Hardware tab settings for configuring TFT display characteristics, display orientation and retentive memory use.



LCD

Display Mode: Determines the orientation of the project on the display.

Landscape is for wide aspect displays. The long edges of the display are the top and bottom of the windows. (Default)

Portrait gives a vertical orientation to the project. The long edges of the display are the left and right sides.

TFT PCD Value: This setting applies to the MMI-640/10T and MMI-320/6T only. It allows the user to change the PCD of the display. The PCD is related to the scan frequency of LCD; (the higher scan frequency, the more stable the display).

Suggested use:

Set TFT PCD as **High** if there is no message board in the project.

Set TFT PCD as **Low** if there is a message board in the project.

Retentive Memory

These settings apply to how the retentive memory is allocated in the unit.

EventLog DataBase: Enables user to keep EventLog information in retentive memory. When the HMI starts up, it checks the in retentive memory and retrieves EventLog DataBase information.

DataBase Start Address: (0~59999). Information is stored in retentive memory starting with the specified address.
Note: LW9058-9059 contains the size in words of the EventLog in retentive memory. Keep in mind that once a block of retentive memory has been reserved, your project cannot use it for other purposes. We recommend that you specify the DataBase Start Address to be after stored recipes so they are not overwritten with Event Log data.

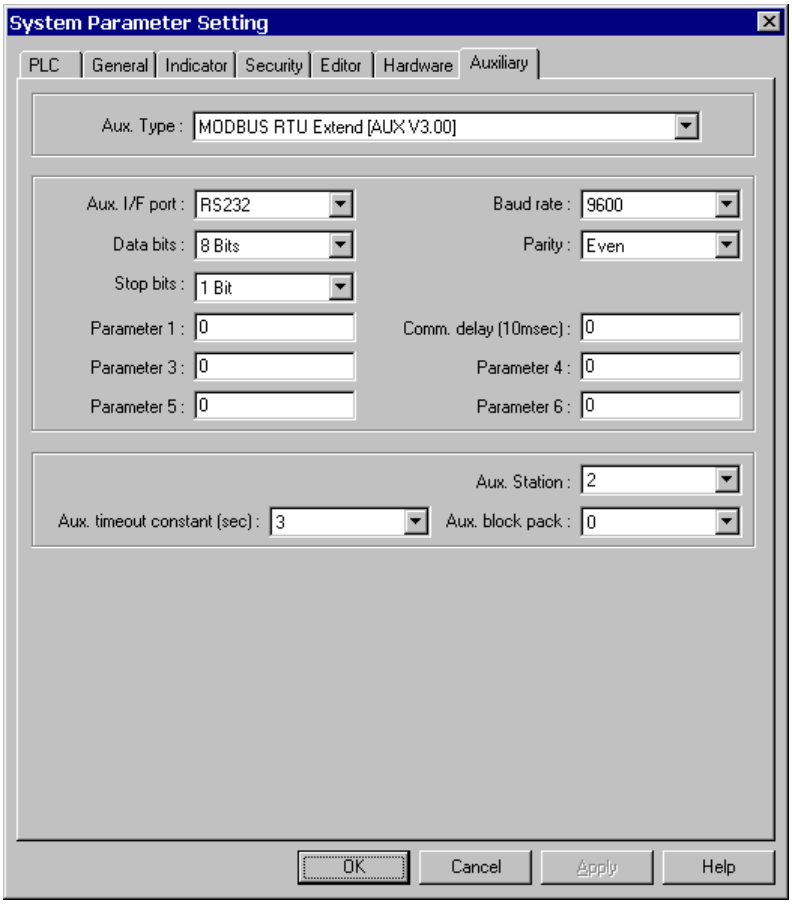
System Parameter: Enables the user to change some System Parameter Settings in the HMI. The parameters are stored in retentive memory beginning with RW60040.
For example: RW60041 represents the PLC port baudrate. If you change this value, the next time the HMI is powered up, the baudrate is set to the value in RW60041.
Note: Set LB9044 to ON to activate System Parameter changes from the Retentive Word area immediately.
Note: Caution should always be observed when changing System Parameters. There is the possibility of making the unit unable to function in the application. When this happens, the project must be downloaded again to restore the System Parameters.
Note: **System Bit and Register Reference** listing of System Reserved Words.

General Tab	**System Parameter**
RW60061	Back light saver: 0(Disable)1~255 second(Enable)
RW60064	Buzzer: 0:None 1:Yes
Security Tab	
RW60071	Security Control: 0:None 1:Yes
RW60072	Password: level 0 (two words)
RW60074	Password: level 1 (two words)
RW60076	Password: level 2 (two words)

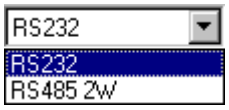
7.2.7 The Auxiliary Tab

The Auxiliary tab is used for configuring the auxiliary port. This port is configured in the same fashion as the PLC communications port. Special drivers are needed for this port. This port does not support AB DH485 or TELEMECANIQUE UniTelWay communications.

Aux. type: Select the type of PLC from available PLC selection menu. The dropdown displays all of the PLC drivers available. These drivers are located in the **\Drivers** - SubDirectory with the *.pds extension. Select the driver that matches your PLC's protocol.



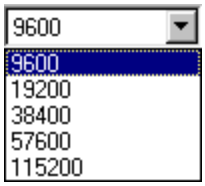
Serial Port I/F: Select the Auxiliary port's type of hardware communications. The HMI activates the corresponding port pins. Only RS232 and 2 wire RS485 configurations are available.

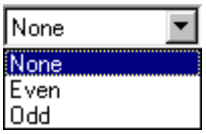


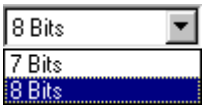
Note: For additional information about communication configurations for a selected PLC, see the **Controller Reference Guide** section that refers to that PLC.

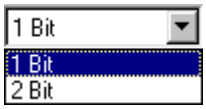
Note: The **AUX[RS-485] PC[RS-232]** port is a dual port.

Baud rate, Parity, Data bits and Stop bits: Set the baud rate to match the PLC ports settings.



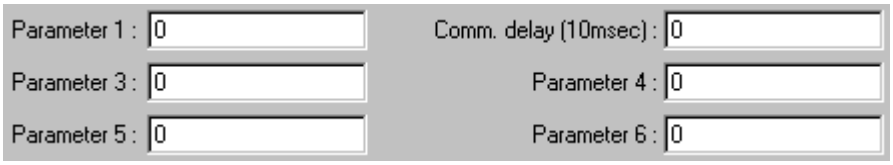




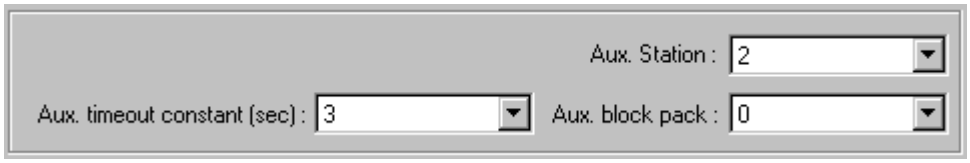


Comm. delay and Additional Parameters:
The Comm. delay parameter is used for setting an interval between communications requests to the controller. Enter a number from 0 to 999. The number specifies the 10-millisecond intervals between communications. Foe example: a setting of 100 puts a 1-second interval between communications requests. This parameter will slow data update rate on a window. It also gives slow controllers a way to meet processing demands without too many interruptions.

Set the other additional parameters as required by your driver. See the **Controller Reference Guide** section that references the PLC driver for further details.



Aux. Station: Used when PLCs have a node or station identifier. The HMI needs the station number to initiate communications. Set as needed or leave at 0 if not used. Station numbers are 0-255 (Use the range as appropriate for PLC type). If the PLC station number does not match the PLC, the “PLC no response” error message is displayed.



Aux. time out constant (sec): This setting determines how long the HMI waits for a response from the PLC. The range is from 0 to 127 seconds. This setting is important when the PLC is normally slow to respond or several MMIs are linked to one PLC.

> ***Note****:* When communications are broken, the red COM LED on the front Panel of the HMI flashes at this interval.

Aux. block pack: Used to determine how the unit communicates to the controller. By increasing this number, larger blocks of registers can be fetched from the controller. In some cases, this speeds the update of information on the display. The range is 0 to 255.

> ***Note:*** Setting the Block Pack to 0 allows the HMI to determine the block size that best fits data retrieval. This is the recommended setting.

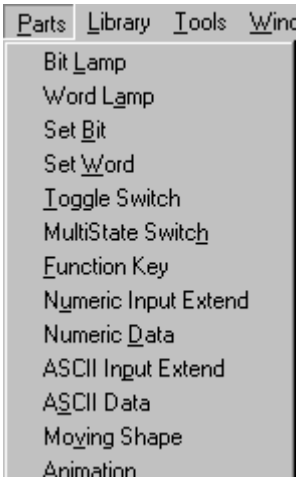
> ***Note:*** Certain PLCs have a limit as to how many data points can be uploaded at a time. If the Block pack number is set higher than this limit, it may effect communications.

> **Hint:** When creating project windows, it is advised to use consecutive registers whenever possible.

7.3 Part Placement

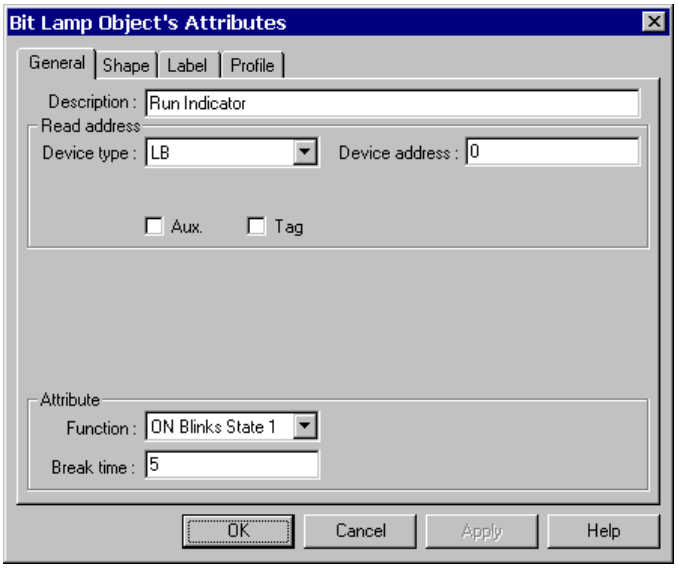
7.3.1 Part Placement Summary

A window is composed of a variety of parts such as Switches, Lamps, Numerical data and Graphics. The procedure to add a part to the window is a simple, 3 step, procedure.



1. Select a Part by either clicking on a Parts toolbar icon, or dragging over to one of the **Parts** menu's selections.

2. When the Part's dialog box appears, set the Part's attribute data, such as PLC read/write address, Shape or BMP, and Label as described in this section. Some Parts have additional tabs that require more information.



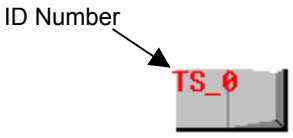
Typical Part's Dialog Box

3. After all of the Part's attributes have been set, simply click on the OK button to place it on the current window. If desired, use the Part's sizing handles to adjust its size, or click on the Part to select it and drag it anywhere on the window.

A Bit Lamp

7.3.2 Part ID Numbers

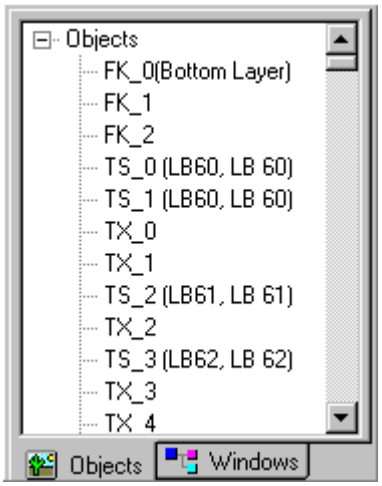
An ID number is a code number that is assigned *automatically* to a Part or Object.



The number identifies the specific part on the current window.

The Part ID is shown in the upper left corner of the part.

WL-001 The 2-letter prefix identifies what type of object it is. The suffix identifies the number of that object. Numbering begins at zero for each type on each window.



Objects are displayed by Part ID in the Window Treebar. Clicking on the part in the Treebar highlights it in the window. See **Treebar Operations** for more details.

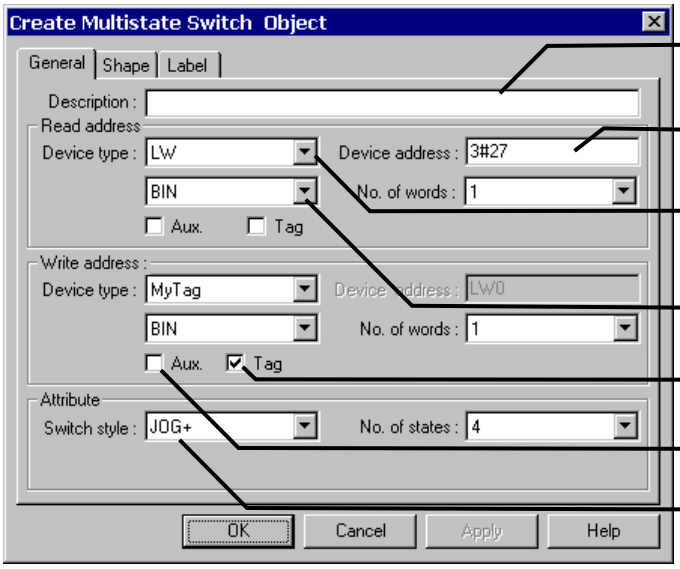
The displaying the Part ID is disabled by unchecking the **Display** setting in the **Option|Window Property** menu item.

Note: The programmer cannot change the ID number.

7.3.3 Part Dialog Features

This section gives a brief overview about part definition. Some Parts have additional tabs that require additional information. For full details about a particular part see Section 3, **Object Reference Guide**.

7.3.3.1 The General Tab



Enter your description here. A reference name (not displayed) that you assign to the Part.

Device address: Bit or register number as is applicable.

Device type: Click down arrow icon to show possible address types.

Select to receive data as Binary or BCD encoded.

Check box to enable Device selection by Tag name.

Check box to redirect request to the Auxiliary port.

Attribute: Additional information for part characteristics is entered here.

7.3.3.1.1 Attaching a Description

If desired, a comment can be attached to a Part. This is a good practice for future reference and useful when editing a project. The maximum description length is 1023 characters.

Note: Descriptions are not downloaded to the MMI. An uploaded project has blank description fields.
Note: Descriptions are not included in the Object Summary printout..

7.3.3.1.2 Entering a PLC Read or Write address

Valid Device types and Device addresses depend on the individual PLC type. If you haven't done so, select the appropriate PLC type in the **Edit|System Parameters** menu item.

Direct Device Entry

Select the **Device type** from the drop down list. Usually these prefixes are the same as those used in the PLCs documentation.

Next, enter the **Device address**. The Device address does not need the prefix repeated. The device address also does not need leading zeroes.

Example: Selecting R00023 as a read address; Select R as the Device type, enter 23 as the Device address.

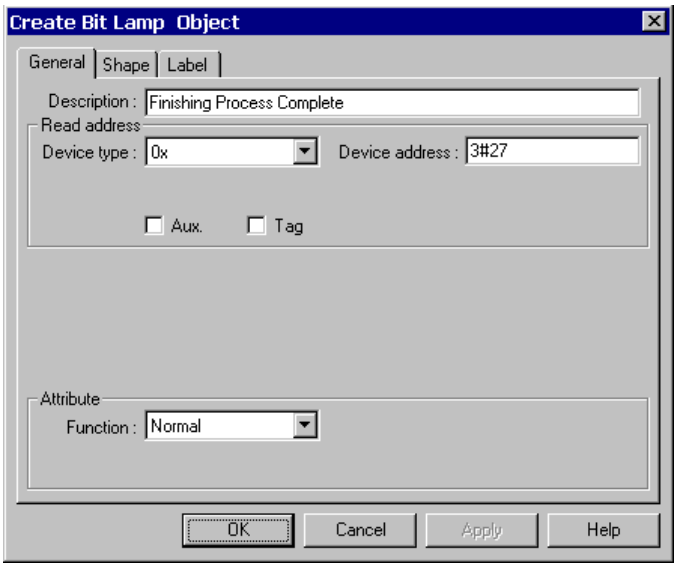
Note: Entering bit addresses.
Bits are usually entered as the bit word followed by the bit number. When entering bit numbers, delimiters such as "space", "." and "/" should not be used. For example bit 11 of Input word 3 would be entered as "311". In some cases, a hexadecimal notation is required (3A). See the Controller Reference Section, for details on bit format for your particular controller.

Note: Extended address mode

When more than one unit is connected to multiple PLCs of the same kind. (Available for some RS485 models listed with the 485 2W suffix.

The addressing method is as follows:
Device type, followed by the PLC ID number, a '#', then the word or bit number.

As shown in diagram: PLC 3, 0x, bit 27 is used for Device.



Example: Modbus RTU (485 2W)

Device type	Address	Description
0x:	1#06	PLC 1, Data Bit 00006
4x:	2#45	PLC 2, Holding Register 40045
3x:	12#54	PLC 12, Input Register 30054

Note: PLC ID number range is limited to: 0~255.

Note: HMI Internal registers and bits. (Reference the **System Bit and Register** section.)

- **LB** (Local Bit) and **LW** (Local Word) are the designations for internal bits and words of the MMI. The range for general use is 0 to 8999. Bits and words from 9000 to 9999 are reserved for system use.
- **RB** bits and **RW** words are internal retentive memory options. **RBI** and **RWI** are the indexes to retentive memory locations.
- **Ms_RB, Ms_RW, Ms_LW and Ms_LB** are bits and words located in the Master HMI when Master/Slave link ups are used.

Using a Tag to enter a Read or Write Device

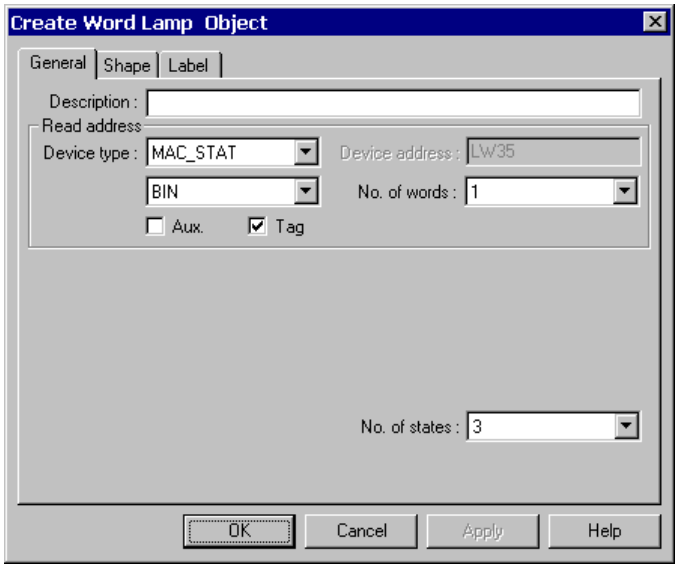
Check the Tag box to select a Tag for the Device Type and Address. Select the Tag that corresponds to the control bit or word for the Part.

Note: Tags must be predefined before use. The Tag database is located in the **Library|Tag…** menu. See the section on Tags for more details.

Retrieving Data from the Auxiliary Port

Check the Aux. box to direct the source of the Read/Write address to the Auxiliary Port.

Note: Be sure the Auxiliary Port is set in the System Parameters. If no device is present on the Auxiliary port, an error will be returned.



7.3.3.1.3 Data format types

BIN (binary format) Decimal numbers are encoded by bit weight

Bit number	15															0
Weight	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

BCD (binary coded decimal format) Decimal numbers are encoded by bit weight per 4-bit nibble

15			12	11			8	7			4	3			0
10^{3}				10^{2}				10^{1}				10^{0}			

For example: The Read address raw data is **0000 0100 0010 0101**.

The **BIN** format treats the data as **1061**

The **BCD** format treats the data as **425**

7.3.3.1.4 Attributes

Some Parts have Attribute parameters on the General Tab. Select the desired attribute from the dropdown and fill in any information required. The attributes vary from part to part. Reference Section 3, Object Reference Guide, for complete details about a Part's attributes.

7.3.3.2 The Shape Tab

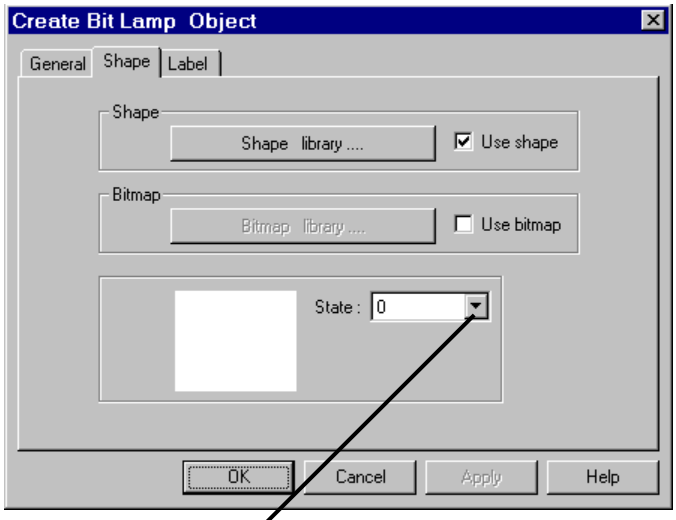
The Shape tab is used to assign a graphic background to an object. If no graphic is assigned then the object has a transparent background. A frame is shown on the project window to indicate that a part is present. The frame is not visible in simulations or the MMI.

Selecting a Shape or Bitmap

1. Click on the **Shape** tab
2. Select **Use shape** or **Use bitmap**
3. Click **Shape library** or **Bitmap library**

The shape/bitmap browser pops up. Select the desired shape/bitmap. Shapes are stored in shape libraries (*.slb files). Bitmap graphics are stored in Bitmap libraries (*.blb files).

A maximum of 10 libraries can be attached to a project. By calling up different shape or bitmap libraries, useful graphics for almost any application can be found. Users can also build up their own shape and bitmap libraries.



The **State** dropdown can be used to view the various states of the selected shape.

Note: If both shape and bitmap are selected for a part, the graphic shown in the preview box is the one displayed on the window at run time. (Usually the Shape takes precedence.)

Note: Reference the Library Operations section for full details about the Library features.

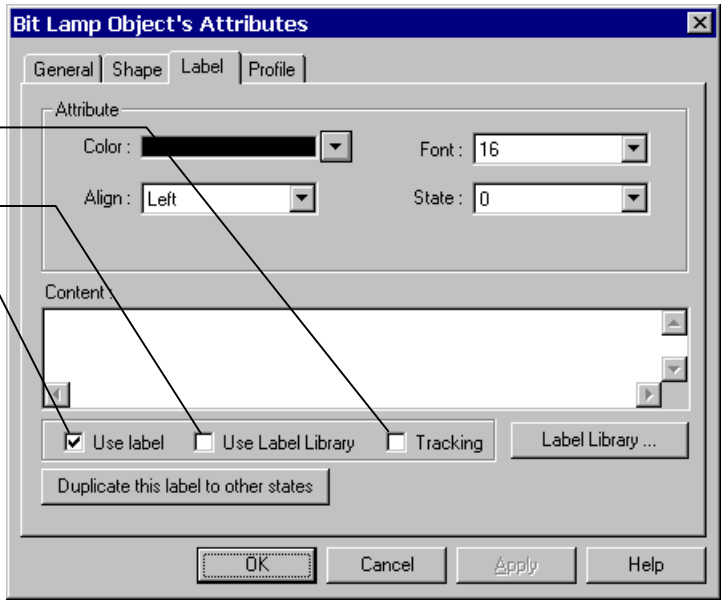
7.3.3.3 The Label Tab

Labels are text that is shown on the face of the Switch, Lamp or other object (part). Different text can be assigned to different states of a part. Example: State 0 = ”OFF”, State 1 = ”ON”.

Select **Tracking** to place and move labels of different states all at once and together.

Select **Use Label Library** to enable label selection from a predefined set of Labels.

Select **Use Label** to enable label display.



Adding a Label to a Part

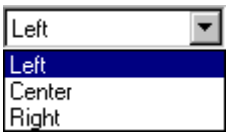
1. Click on **Label** tab
2. Select Attributes for the label
3. Select the state to label
4. Type the state’s label in the Content box

Color

Select a color for the text from the popup Color dialog. Colors can be changed from state to state. The Grayscale units have a color selection of only 4 colors. If you have a color unit, there are 256 colors available. See the section on Drawing for details about color selection.

Align (Justification)

Select the desired text alignment. Alignment can vary from state to state. When the text input for a Label becomes 2 lines or more, the alignment can be defined as **Left**, **Center** or **Right** justified.



Note: **Center** justifies the lines of text only; it does not center the text on the object.

Note: Text alignment can be changed after a part is placed by using the text alignment tools in the tool bar.

Font

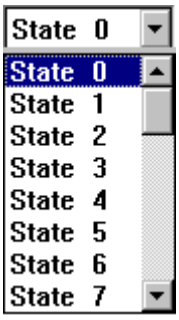
Select the desired font size. Font size can vary from state to state.
Font sizes 8, 16, 24, 32, 48, 64 and 96 are available.

Note: Font size can be changed after a part is placed by using the font tools in the tool bar.

State

States are displayed only for those objects with multiple states. The text displayed for each state can be set independently. Simply select a state in order to set its text, font, color and alignment.

Note: The State can be viewed after a part is placed by using the State dropdown in the tool bar.

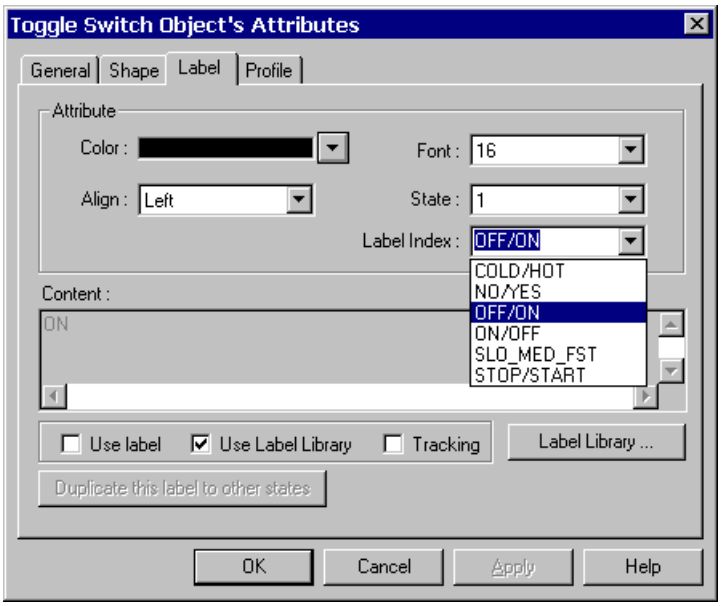


State 0

State 1

Content Box

There are two options for applying text as the Label. The programmer can:
Enter text directly in the Content box (**Use label**)
Or
Select a predefined Label (**Use Label Library**).



Use label selected…

Type in the characters to be displayed on the button in the Content field. The "ENTER" key can be used to move to the next line if desired. Text is limited by screen size. If entered text exceeds the screen size, it is automatically deleted when the part is placed. Be sure to enter text for all of the desired states.

Use Label Library selected…

Select the label to use for the state from the **Label Index** dropdown. Text entry into the Content Box is disabled. Click the **Label Library…** button to add or view predefined labels.

Note: All states must use the same method for entering the label. You cannot have a predefined label for one state and a directly entered text for another within the same part.

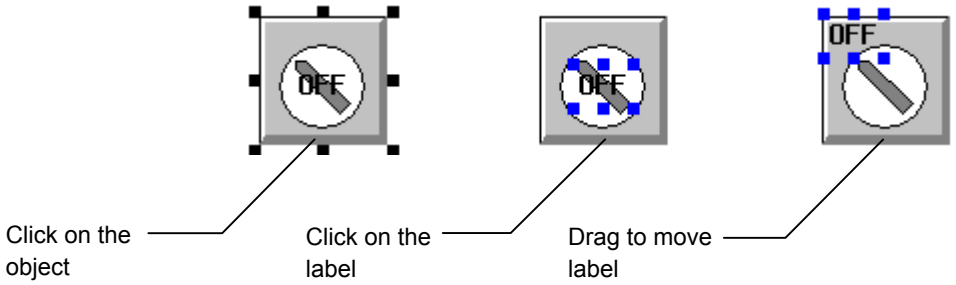
7.3.3.4 Part Placement

Once all of the Attributes have been set as desired, click the OK button to place a new part on the screen. The part is shown in the upper left corner of the editing window. The part can then be dragged into position and resized as desired.

Note: Label Position

Once the Part is placed on the window, the Label can be moved anywhere within the boundary of the object. Just click on the label and drag the mouse to move it to the desired position. Toggle through the state designator to display the predefined label in each state.

Note: If **Tracking** is enabled, the labels in all states are located where the current label is displayed.



7.4 Window Operations

Every new project begins with a default window, window number 10. Usually more than one window is needed for a project. There are three types of windows – Base, Fast and Common. Allowable window numbers are from 10 to 1999.

Window types

Base Window:

These are the normal windows displayed during HMI runtime operations. They are used as informational or control screens that allow the operator manage a process. A PLC control or function key may call them up. They are also used with Direct and Indirect Window parts to appear on the screen as needed. The possible range of Base window numbers is 10 to 1999 (1 to 1989 if starting window is set to 1).

Fast Selection:

The Fast Selection window is activated when the Screen Task Button is pressed. The size is limited to 10 to 200 pixels wide by 10 to 480 pixels high. It is designated as window number 4 (-5 if starting window is set to 1). See the System Parameters and Task Bar sections for more details about the Fast Selection window.

> ***Note:*** Base windows can be reassigned to the Fast Select function through a Function Key with the **Jog-FS** action. The Base window replacing the Fast Select window must be the same height and width as the Fast Select window 4.

Common Window:

Common windows are used as foregrounds or backgrounds to Base windows. All controls on the acting Common window are active. In addition, any objects drawn on the common window and parts with shapes/bitmaps are visible when on top of the controlling Base window. This saves memory by allowing the programmer to provide a common header or set of controls that are always active no matter what base screen is called.

The default Common window is window 6 (-5 if starting window is set to 1).

> ***Note:*** Base windows can be reassigned as the Common window through a Function Key with the **Change Common Window** action. Base window numbers 10 to 1999 can also be assigned as Common windows. When called as Common windows they behave as Common windows by replacing the default Common window (6).

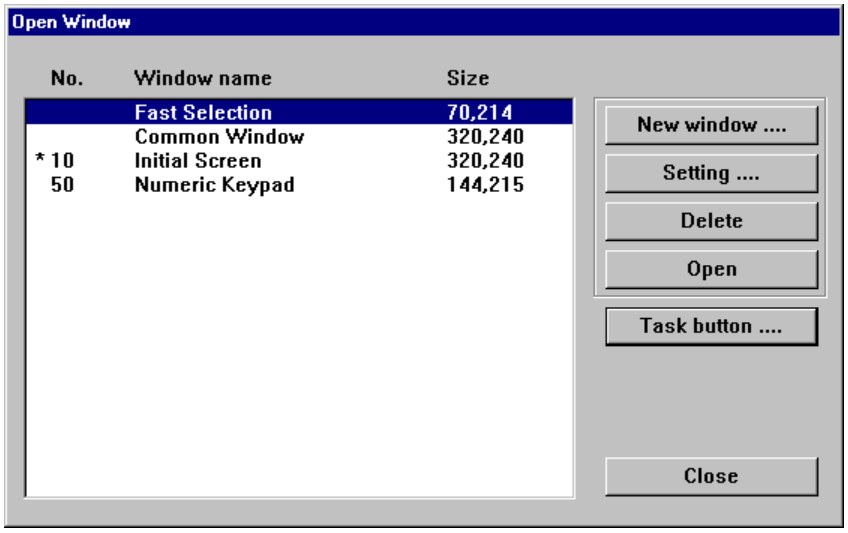
> ***Note:*** Base windows may be specified as smaller than full screen to facilitate a pop-up operation. These small windows, when popped up, covers the Common window objects.

7.4.1 Creating New Windows

7.4.1.1 Procedure for Creating a New Window Number

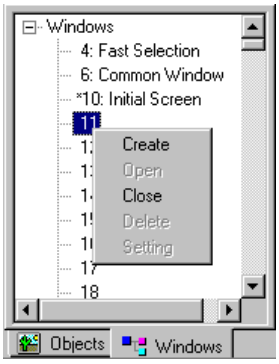
In the **Window** menu, select the **Open Window** command.

In the Open Window dialog click on **New Window….**



Select the window type; **Base Window** or **Fast Selection** or **Common Window**.

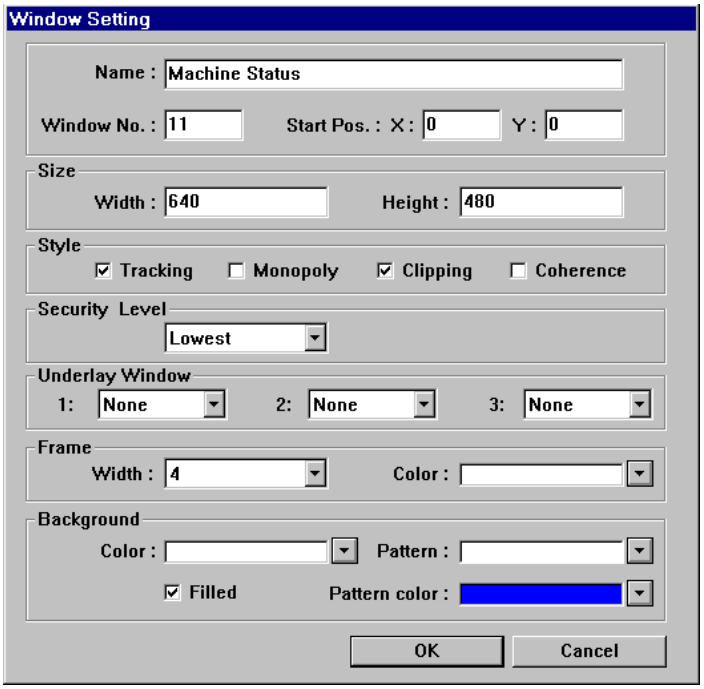
Or …



Right click on an undefined window number in the Treebar

Select **Create** from the popup menu.

Setup the window properties as desired



Name: Give each window a unique name in order to identify it. The name is shown in the Window Treebar.

> ***Note:*** Window Names are not downloaded to the MMI. Uploaded projects shows the default window name.

Window No: This value can be changed if creating the Window from the Open Window dialog. Valid window numbers are from 10 to 1999 (1 to 1989 if starting window is set to 1).

> ***Note:*** The window number cannot be changed after a window is created.

Start Pos.: Set the coordinates of the upper left-hand corner of the window when displayed on the MMI. This applies only if the window is less than a full screen.

Size: Set appropriate width and height if it is going to be a pop-up window, otherwise, set window width as full screen size (320x240 or 640x480 depending on model)

Style:

Tracking, Monopoly Clipping and Coherence

These parameters specify the relationship between a child pop-up window and it's base calling window.

Monopoly: If a child window is marked as **Monopoly,** the calling window is frozen when it is popped up.

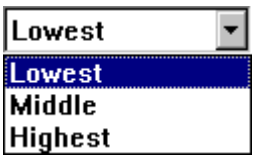
Clipping: A child window marked as **Clipping** limits its boundary to the calling window. This means the child window's display area outside of the base calling window's frame is cut off.

Tracking: A child **Tracking** window moves when the calling window is moved.

Note: Both the Clipping and Tracking must be selected or deselected.

Coherence: Coherence means that there can be no windows between the calling window and this window. One cannot be brought to the foreground without the other; both are brought forward.

Security: This prevents unauthorized access to windows when the System Parameter, Security feature is activated. Lowest is security level 0, Middle is security level 1 and Highest is security level 2. See the **Security** Section for further details.



Underlay Window: Up to three windows can be specified as underlay windows. An underlay window adds its objects into the base window being created. The Underlay window's attributes (Style, Security, Underlay, Frame, etc.) are ignored for this purpose.

LIMITATION NOTE: Trend displays cannot be on both an Underlay window and a Normal window. Projects with underlay windows cannot be opened by EasyBuilder versions prior to 1.6.6 even if System Parameter Compile level is set to 0.

Frame: A Frame is suggested for child windows. The width in pixels of the frame can be selected from the drop down (0, 4, 6, 8, 10, 12, 14, and 16). If the Frame is not set to 0, the 16 basic colors are available for the frame.

Note: Objects placed on top of the frame during project development are covered by the frame when the project is running.

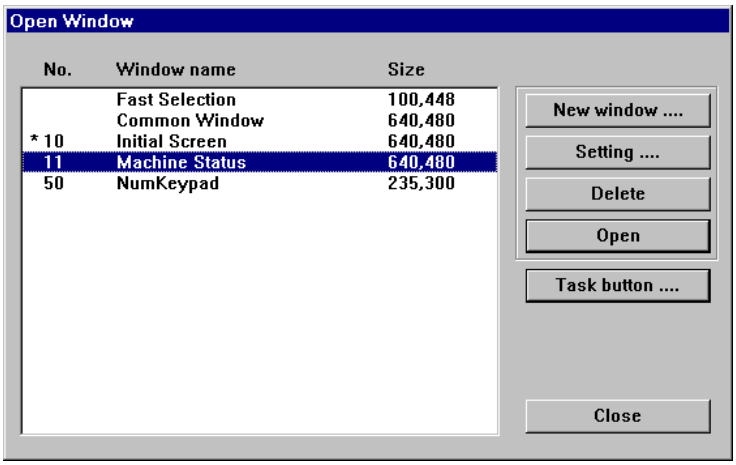
Background: Select the background Color, Pattern and Pattern Color for the window. The Filled option enables and disables the Pattern in Design and Run modes. See the Drawing section for details about available patterns.

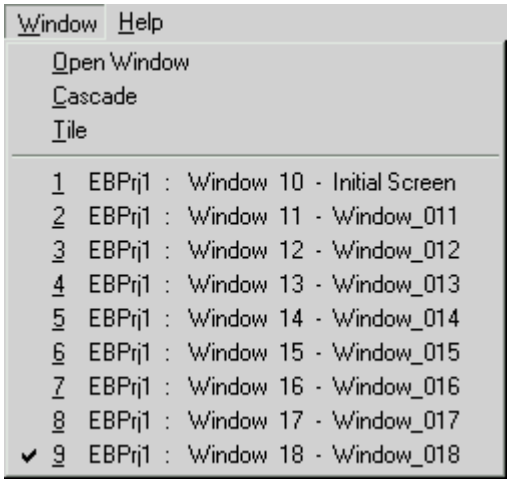
Note: If the **Filled** option is not checked the window's background color is not shown during project design but is shown during simulation and run modes.

7.4.1.2 Opening Windows for Editing

After a window is created, with the **Open Window** dialog box select it from the list and double click it or press **Open** to open the window.

Note: The windows marked with an "*" are already opened.

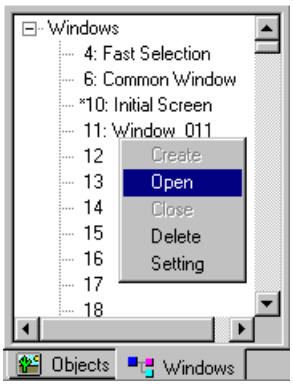




To edit a window that has already been opened, just select **Window** menu then select the window.

Alternatively, Right click the window number in the **Treebar** to popup the shortcut menu. Then select **Open.** If the window is already open, double-click the window number to bring it to the front.

Note: The windows marked with an "*" are already opened.



7.4.1.3 Modifying Window Properties

Select the **Window|Open Window** menu, Then select the window from the list and press **Setting**. The window properties can now be modified.

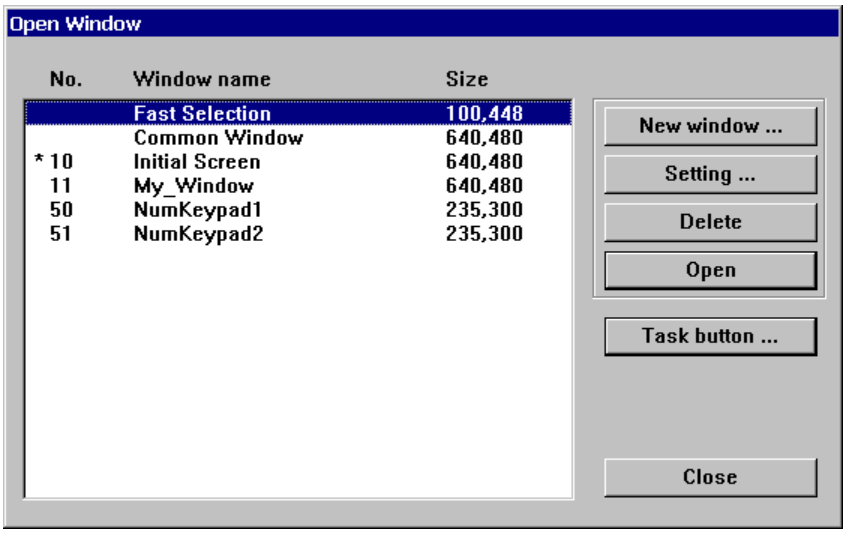
Note: The window number cannot be changed after window is created.

7.4.1.4 Deleting a Window

Select the **Window|Open Window** menu. Then select the window from the list and press **Delete**.

Alternatively, Right click the window number in the **Treebar**. Select **Setting** to edit the properties or **Delete** to delete the window.

Note: A window **Delete** cannot be restored with the **Undo** command!



7.4.2 Adding Objects to a Window

Once a window is opened and displayed, the programmer can draw shapes and add parts to the window. Use the Draw and Parts menu or toolbars to create the look and feel desired.
Graphics placed with the Drawing Tools do not respond to touch actions. Some Parts do not react to touch actions. It is advised that each part be reviewed and understood by going through the Parts Reference Section.

7.4.2.1 Drawing Graphics

Click on the drawing tool or select the item from the **Draw** Menu and drag the shape on the window. Once placed the shape can be resized using its “handles”. Handles are the highlighted squares that appear around the graphic when it is selected. An Attribute Dialog to adjust color and fill characteristics is available through **Edit|Change Attribute…** menu or by double-clicking the graphic or clicking on the **Edit** tool.

7.4.2.2 Adding Parts

Click on the Part tool or select the item from the **Part** Menu. Fill in the Part’s Attributes in the popup dialog. Then, place the Part on the window. Some Parts can be resized after placement. The Attribute Dialog can be recalled through **Edit|Change Atribute…** menu or by double-clicking the Part, right clicking and selecting **Attributes** from the shortcut menu, or clicking on the Attributes tool.

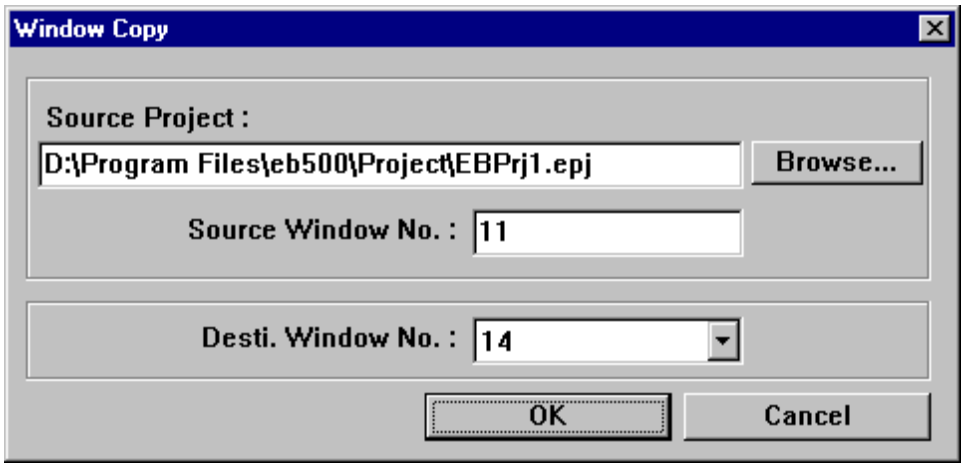
7.4.3 Copying and Importing Windows from Other Projects

Windows created in one project can be copied or imported into another project. This is done through the **Edit|Window copy…** menu command. Selecting this menu item pops up the **Window Copy** dialog box.

Browse to the *epj file that has the window you want to copy or import. Fill in the **Source Window No.** from the *.epj project. Select a window number in the open project where the window is to be placed (**Desti. Window No.**). If the Destination window already exists, the program displays a warning dialog that allows you to cancel the operation or continue. If you chose continue, it overwrites the window. Click **OK** to finish.

Note: Shapes and Bitmaps needed for the imported windows are not automatically brought into the open project. Be sure to attach the appropriate libraries before importing windows.

Note: A window copy cannot be reversed with the **Undo** command!



The above example shows Window 11 from project EBPrj1 about to be copied to window 14 of the open project.
If the open project some project other than EBPrj1, the window is imported. If EBPrj1 is the open project then the window is copied (duplicated) to the destination window number.

7.4.4 Changing and Popping Up Windows

EasyBuilder has 4 ways to change or popup windows during simulation or run operation. See Section 3, Object Reference Guide, for complete details about **PLC Controls**, **Functions**, **Indirect** and **Direct** windows, their attributes, and how to implement them.

PLC controlled action:

Use the **Change Window** PLC Control to replace all other Base windows being displayed. Common and Fast Selection windows are not affected.

Operator controlled action:

Use **Function** parts with the **Change Window** attribute to replace all other Base windows being displayed. Common and Fast Selection windows are not affected.

Use **Function Key** parts with the **Popup Window** attribute to place another Base over top of the active window. The only way to close a popup window is to put a Function button with the **Close Window** attribute in it.

Note: A popup window of this type cannot use a Function part to pop up another window.

Use **Function Key** parts with the **Change Common Window** attribute to replace the currently active common window. Base and Fast Selection windows are not affected.

Use **Function Key** parts with the **Jog FS-Window** attribute to replace the currently active Fast Selection window. Common and Base windows are not affected.

Use **Function Key** parts with the **Return to Previous** attribute to replace the current Base window with the last active Base window. Common and Fast Selection windows are not affected.

PLC or Operator controlled action:

Use **Indirect** and **Direct** window parts to overlay popup windows over active windows in a predefined area. Windows called by **Indirect** and **Direct** window parts do not replace full screen windows.

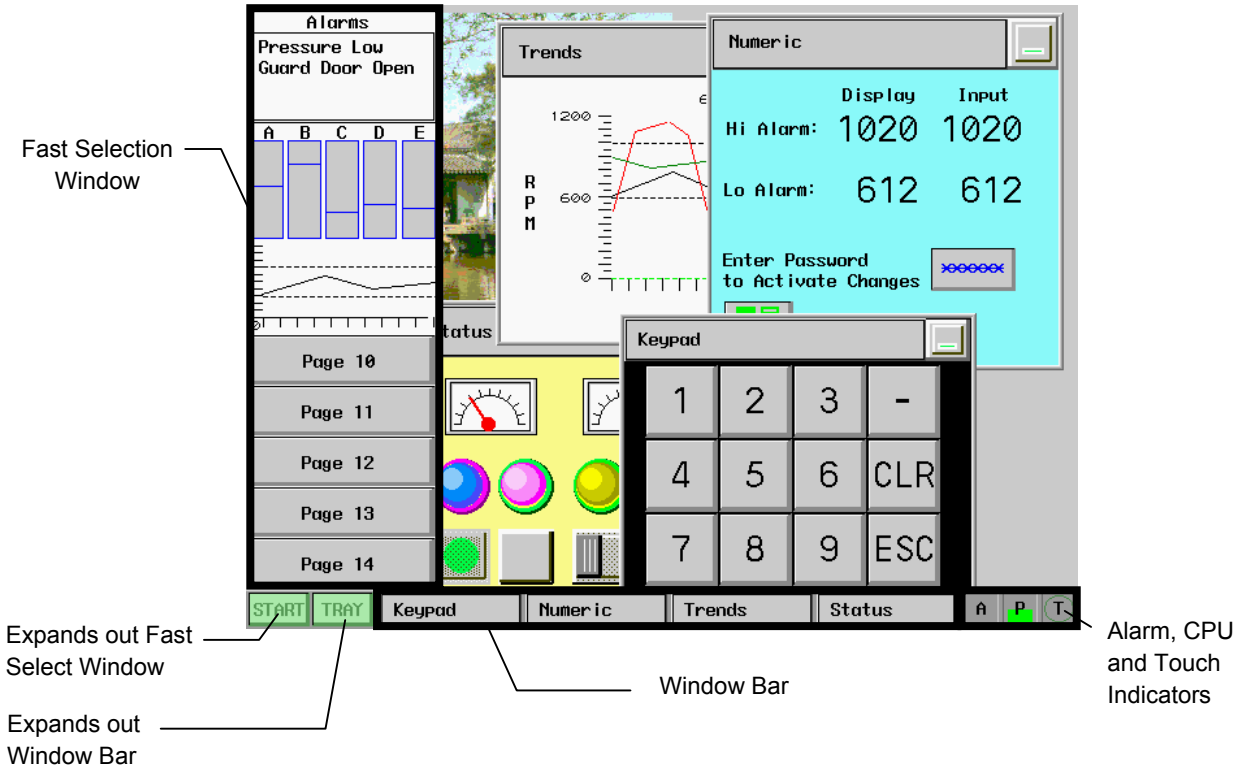
Note: **Indirect** and **Direct** windows can call up other **Indirect** and **Direct** windows.

Note: **Indirect** and **Direct** windows do not work properly when they are on the same Base window as a Function key **Popup Window** part. Additionally, do not put Function key **Popup Window** parts on **Indirect** and **Direct** windows.

7.5 Task Bar Operations

The Task Bar allows the programmer to create projects with familiar graphic features. The Task Bar holds the buttons for a pop-up menu and system tray. This allows the user to pop-up (maximize) or icon-ize (minimize) child windows and change window displays. The “Task bar” must be enabled in the System Parameters for the Task buttons to be displayed.

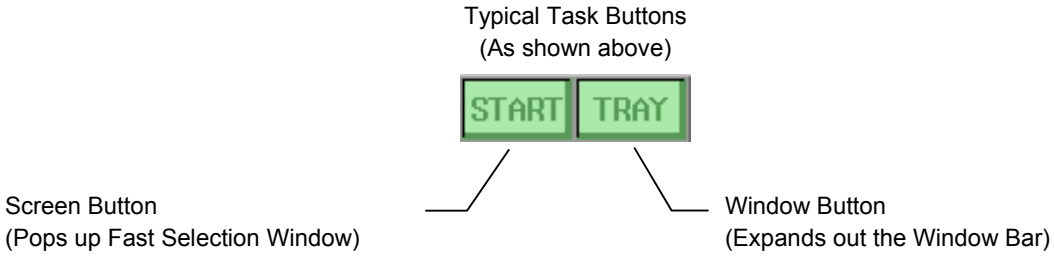
7.5.1 Task Button Overview



Press the Screen task button to pop-up the Fast Selection window.

The Fast Selection window can have several Function parts (placed on it by the user) to change to different windows. Since the Fast Selection window is global to all windows, it can be called up at any time. To change windows, press the function part that targets the destination window.

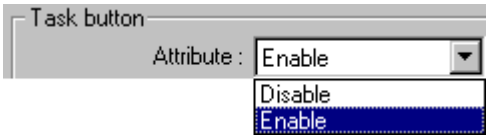
The Window Bar accommodates up to six window icons. Double-click the **Minimize** function part of a window to shrink that window to the Window Bar. Click it again to return the window to its original location and size.



7.5.2 Procedure to Setup Task Buttons

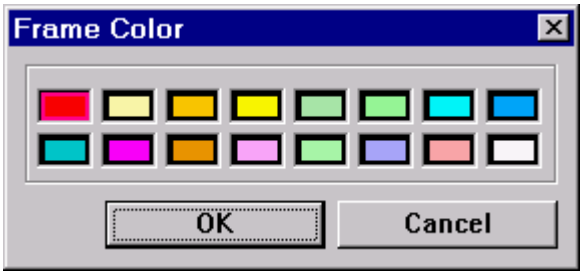
1. Select **Edit|System Parameters** menu. Click on the General Tab.

2. Select the different drop downs to determine how the task bar appears.



Select Attribute **Enable**.

Note: The task buttons are visible at run time in the lower right or left corner of the display. The Task buttons are not visible when editing a window.



Select **Background color** from pop up dialog. Extended colors are not available for the Task Bar background.

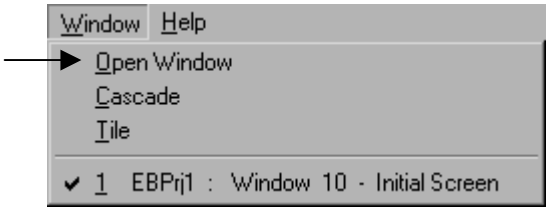
Position: This selection determines on which side the Fast Selection window and Task Buttons are displayed.

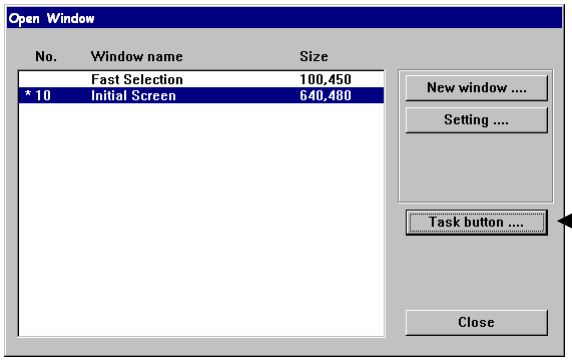
Select **Right** or **Left**.

The **Text** attribute is used to determine how text is displayed in the task buttons and Window Bar icons.

Select **Left Adjust** or **Center**.

3. Once the selections in the System Parameters dialog have been set, select the **Window** menu, **Open Window** item.





Click **Task button….**

Fill in the blanks of the Window Setting dialog.

Name: This is always fixed as Task Bar

Window No. This is always fixed at 2.

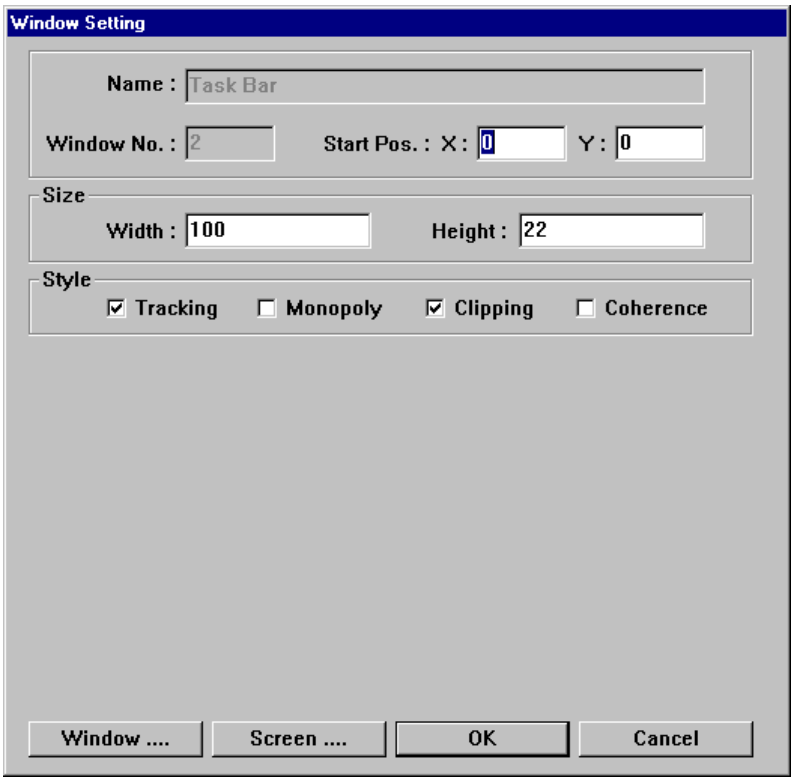
Start Pos.: Changes to X and Y have no effect. The task bar is always positioned at the bottom of the screen.

Size: Enter the size of the combined two Task Bar buttons. Each is given half of the Width specification.

Width: The range is 10 to 200
Height: The range is 10 to 100

Note: The Window Bar is the same height as the Height specification. The Indicators are displayed as square objects with the Height specification determining their size.

Change of **Style** has no effect.



4. Click the **Window....** and **Screen....** buttons to setup the Task Buttons.

The **Window Attribute** dialog has the settings for the button that controls the **Window Bar** where window icons are displayed.

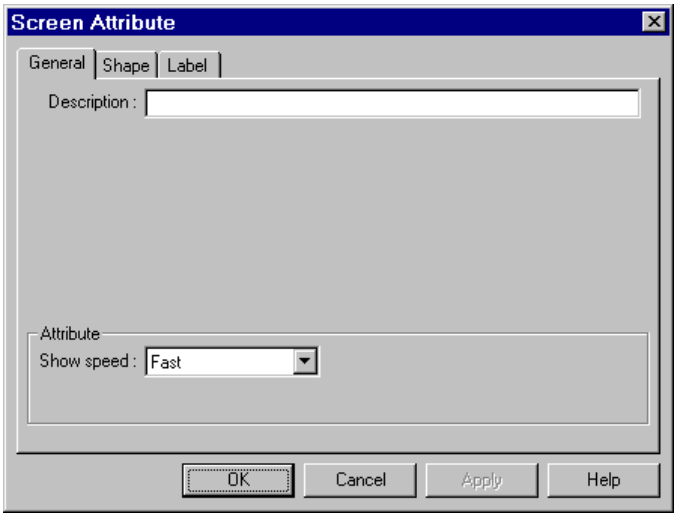
The **Screen Attribute** dialog has the settings for the button that controls the popup **Fast Selection** window.

A **Description** can be entered if desired.

Attribute: **Show speed:** Select **Fast** or **Slow**. This setting determines the pop up animation speed.

Shape: Select a shape for the Screen or Window button. Just like any other button object.

Label: Place a customized label on the Screen or Window button. Just like any other label.



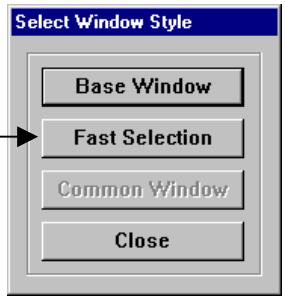
Note: Label text is limited to the button width. Use appropriate text Size and Color.

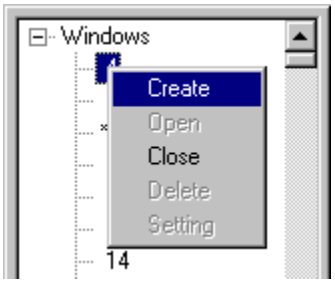
Once these 4 steps are completed, run the project in Off-line simulation to see how the Task Bar appears. Edit the Task Bar as needed to obtain the desired appearance.

7.5.3 Creating the Fast Selection Window

Every new project starts with a default Fast Selection Window. This window can be deleted and a new Fast Selection window created to fit your needs. To create a new Fast Selection window first delete the existing Fast Selection window.

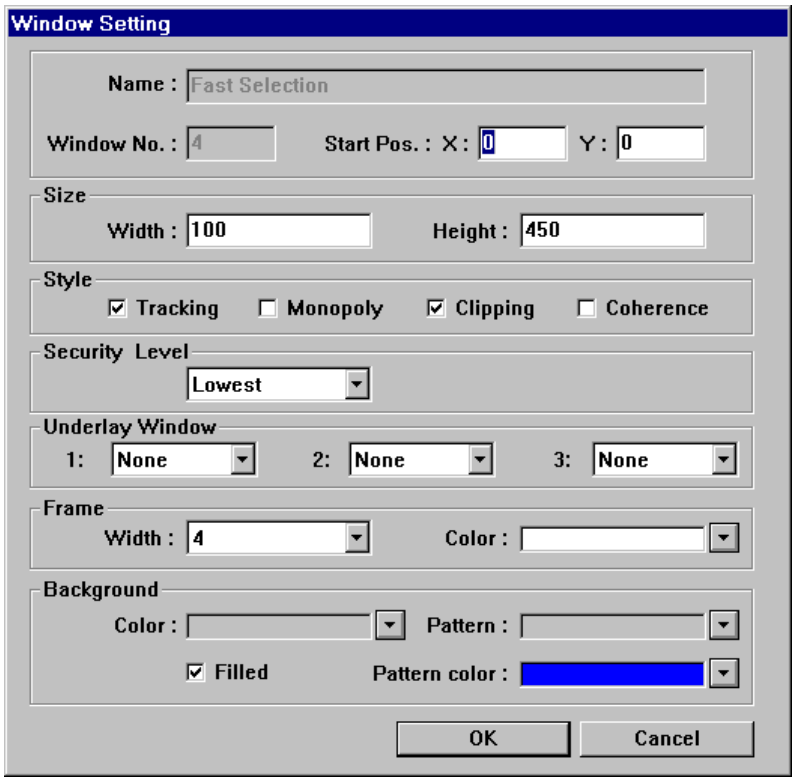
Select **Window** menu, **Open Window**
Click the **New Window….** button, then click **Fast Selection**.





Alternatively, Right click window **4** in the Window Treebar. Then select **Create** from the pop up menu.

Fill in the blanks of the Window Setting dialog.



Name: This is always fixed as Fast Selection

Window No. This is always fixed at 4.

Start Pos.: Changes to X and Y have no effect. The Fast Selection is always positioned at the bottom of the screen.

Size: Enter the size of the Fast Selection Window. The Width and Height size are limited.

For 640 x 480 pixel displays range is:
Width: 10 to 200
Height: 10 to 480

For 320 x 240 pixel displays range is:
Width: 10 to 200
Height: 10 to 240

Change of **Style** has no effect.

Security Level has no effect on Fast Selection windows. These windows are always accessible.

Underlay Windows can be assigned as desired.

Set the **Frame** and **Background** attributes as desired.

Follow guidelines in the section on “Creating New Windows” to place buttons and parts on the Fast Selection Window.

Note: Any objects that can be placed on a Base window can be placed on the Fast Selection window.

7.5.4 Using the Task Bar

The Task Bar is used to hold the window control buttons of pop-up child windows. Window control buttons only appear if a Window Bar Function part is on the child window. Minimize window parts are not necessary but helps when multiple windows are displayed to clear the screen.

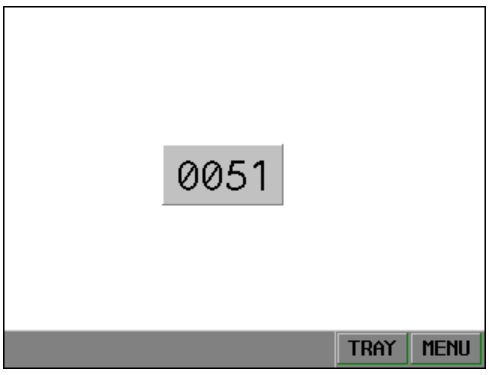
Function Parts are used to create title bars and minimize icons for child windows. See the Object Reference Guide section for more details about creating and using Function parts.

Base windows can be minimized by putting a Function part with the **Minimize window** attribute on them.

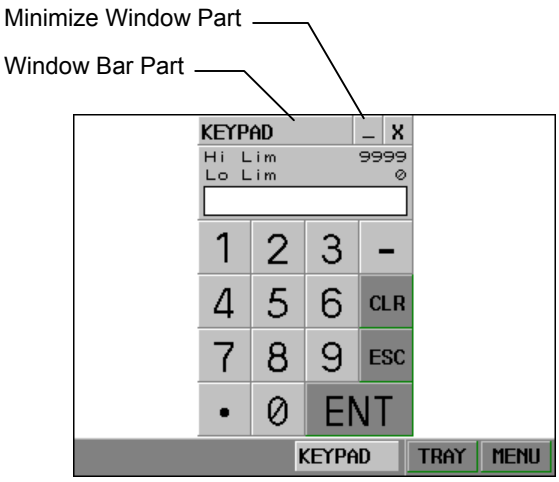
Base windows that have a Function part with the **Window Bar** attribute can be moved. The label text in the title bar is truncated and shown in the reduced icon form in the window bar.

Example:

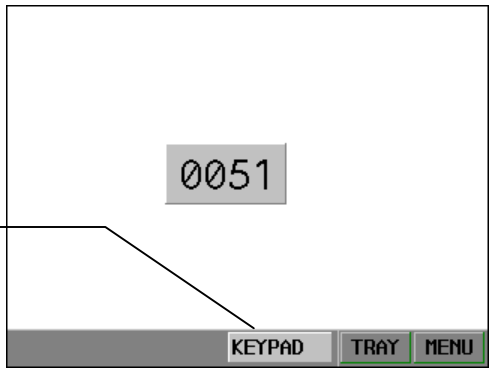
1. A Numeric Input Extend part is on a window.
2. A pop-up keypad is activated when the Numeric Input Extend part is touched. The Keypad has a Window Bar and Minimize window Function part in it.
3. The operator can minimize the keypad to view the original number if desired.
4. Touching the Minimized button on the Task Bar restores the keypad window.



Window showing Numeric Input Extend part and Task Bar
Example: Part 1



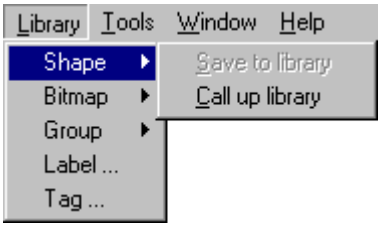
Window showing pop-up Keypad window with Control button on Task Bar
Example: Part 2 and 4



Window showing minimized Keypad window with Control button on Task Bar
Example: Part 3

7.6 Library Operations

7.6.1 Library Overview



Libraries are a way to create a collection of bitmaps, shapes and groups that apply to the project being developed. Library objects become available to a project when the Library they are in is attached to the project.. To view attached libraries, go to the **Library** menu, select the **Shape**, **Bitmap** or **Group** category and the **Call up Library** submenu item.

Default libraries are automatically assigned to a new project. Libraries can be added or deleted from a project at any time. Up to 10 libraries can be assigned to a project.

EasyBuilder provides three types of libraries:

1: **Shape Library**: A Shape is a collection of drawing elements put in by the programmer with the EasyBuilder drawing tools. Those elements, when put together, define a graphic symbol representing a button, lamp, function key etc.

2: **Bitmap Library**: A Bitmap graphics (BMP) is a collection of pixel of data. This data can come from screen captures, paint or drawing software packages that can generate *.bmp files. Bitmaps smaller than 250 KB and up to 256 colors are allowed.

3: **Group Library**: A Group is a collection of objects. (i.e. A keypad would be a group of function buttons.) Common groups can be saved to this library for use in other windows of the same project or for later use in other projects.

Comparison of Shape and Bitmap Attributes

Attribute	Bit Mapped Graphics	Shapes
Arbitrary Graphic	Good	Limited to those graphics that can be designed by basic drawing elements
Memory capacity	Very large	Small
Effect of enlarge & shrink	Fair to Bad	Good
Display Speed*	Good	Good

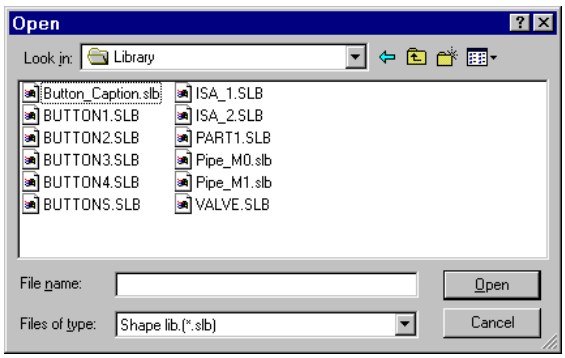
* The MMIs 32 bit processor displays both bit map graphics and shapes extremely fast, even during resize operations.

Click **Select Lib. ...** To attach an existing library to the current project. Libraries are found in the Library D-Subdirectory where EasyBuilder is installed.

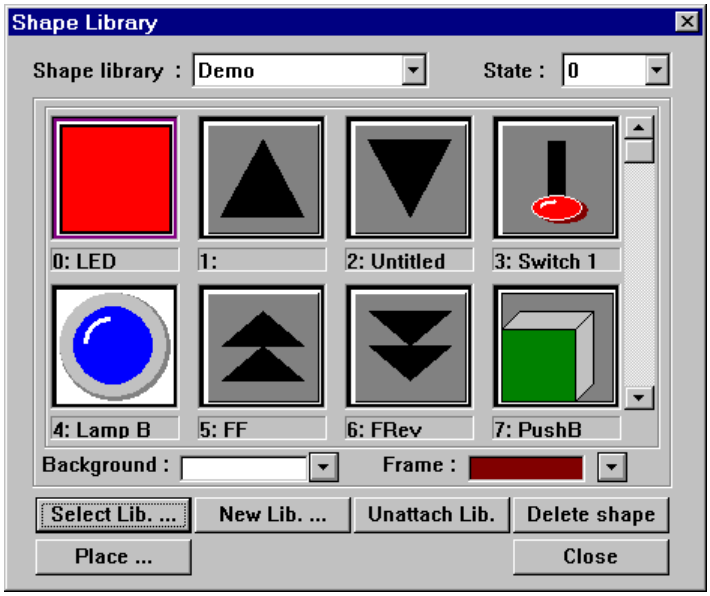
Shape libraries have the *.slb file extension.

Bitmap libraries have the *.blb file extension.

Group libraries have the *.glb file extension.



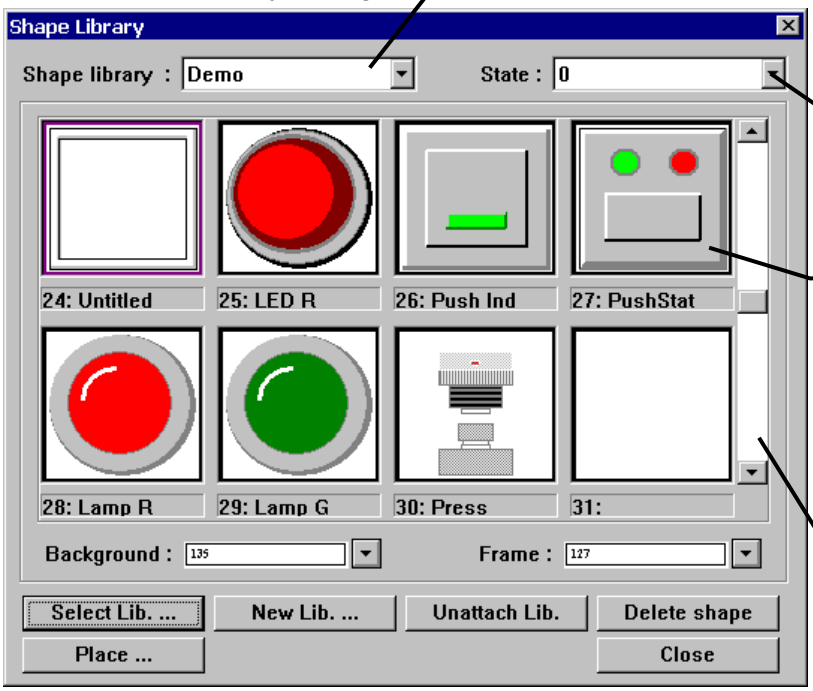
Shape Select Library Dialog



A sample library dialog

7.6.2 Shape Library Operations

7.6.2.1 Shape Library Dialog



Library Selector: Use dropdown to select any Shape Library attached to this project.

State Selector: Use dropdown to select the viewing state of the shapes displayed.

Cell: Each cell can have a maximum of 32 shapes, the first shape in the cell represents state zero (0), the second state one (1) and so on. Change the state selector to view each shape in a cell. Each cell has a number and displays an 8-character name

Scroll bar: Use the scroll bar to view above and below the displayed cells. There are 48 cells available in a library (0-47).

Shape Dialog Summary

Item	Description
Background	Allows the background field of the library objects to be changed. This is used to see how a window's background color affects the way a shape appears. The background color of the shape object in the library does not follow the shape when it is placed on the window. The color number 1~256 is displayed in box.
Frame	Used when viewing SYS_XXX libraries to uniformly change the color of all buttons in the library. This has no effect on other libraries. The color number 1~256 is displayed in box.
Select Lib.…	Attach an existing library to the current project.
New Lib.…	Attach a new (blank) library to the current project.
Unattach Lib.	Remove a library from the current project. Objects that use shapes from the deleted library are displayed as rectangle outlines. No warning is given when this happens. Be sure you want to do this operation.
Delete shape	Delete all the shapes from the selected cell. ***Note:*** This action cannot be recovered with the Undo command. ***Note:*** The result of this operation is a completely empty cell. If you want to delete only one shape out of a cell, place all of the cell's shapes on screens and delete the cell then put back into the empty cell the shapes you want to keep.
Place	Place the shape on the current window as a group of drawn objects. This is useful for editing a shape's color or using an existing shape as a base for a new shape.

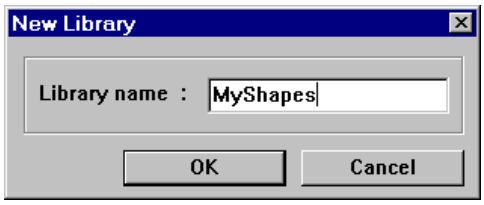
Note:

Use the **Shape Drawing Tool** to place a shape on the screen as a background. Shapes placed with the Drawing tool cannot be edited on the window.

Use the **Shape Library Place** command to put a shape on the screen broken into individual parts. Shapes placed in this manner can have each part edited.

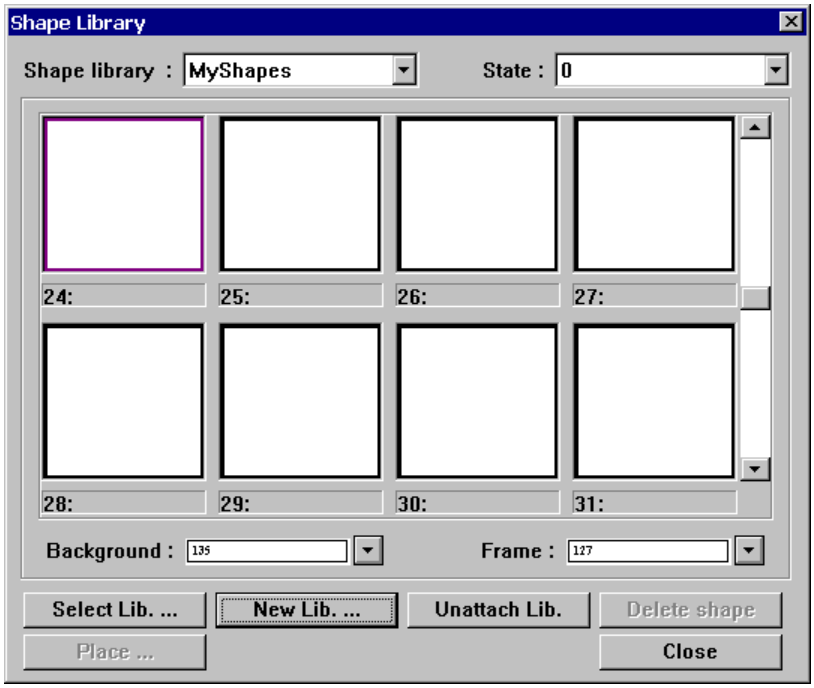
7.6.2.2 Creating a Shape Library

Click on the **New Lib.…** button. A Dialog Box for entering a file name appears.



Enter the name of the shape library to be created.

Note: Be sure the library name is not already used. Creating a new library with the same name as an old library erases the old library.
Note: All shape libraries are automatically assigned the *.slb file extension.



An empty new library dialog

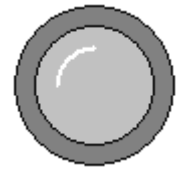
NOTE: New and modified libraries are not saved to disk until the project is saved. Closing the project without saving loses changed library data and new libraries!

7.6.2.3 Opening a Shape Library

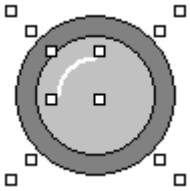
Click on the **Select Lib.…** button. A Standard Dialog Box for selecting a file appears. Select the name of the shape library to be opened.

7.6.2.4 Adding a Shape to a Library

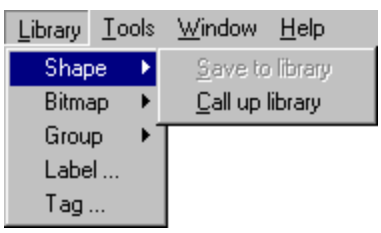
1. Open a window and Draw the graphics by using the drawing tools in the EasyBuilder program. For example, the following graphic uses the circle and arc tools.



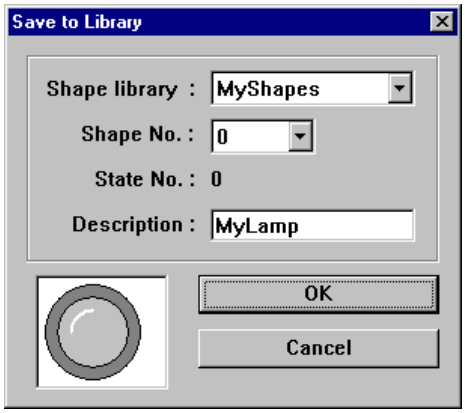
2. Select the whole graphic by using the arrow tool and dragging a rectangle around it. White handles should appear on all of the selected objects.



3. Select the **Library** menu **Shape** submenu **Save to library** item or click the  tool.

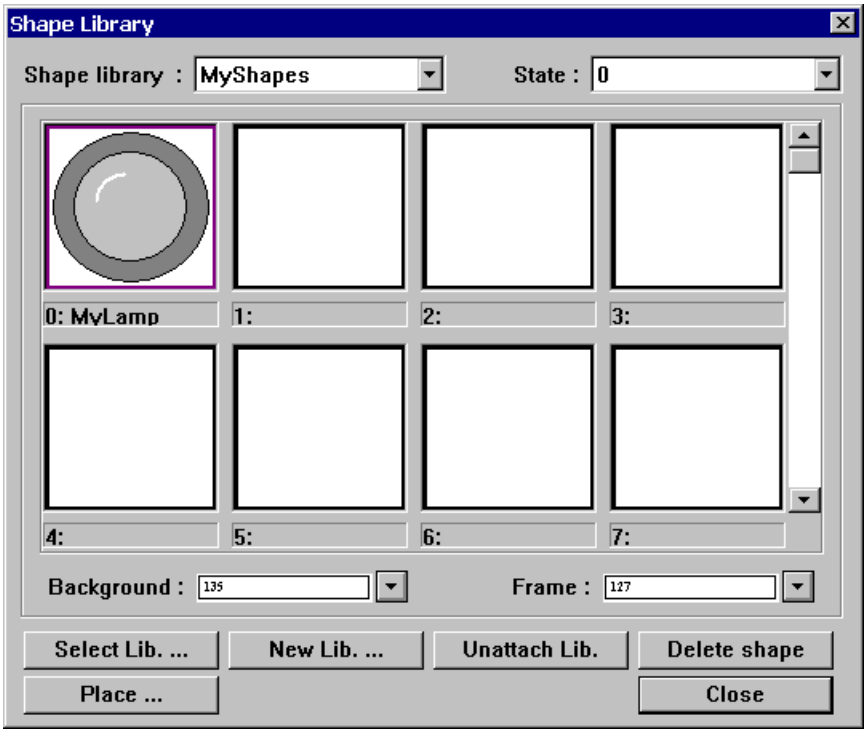


4. Fill in the Dialog information.



Item	Description
Shape library	Used to select the Shape library for adding the shape.
Shape No.	Used to select the cell no. (Shape no.) for adding the shape.
State No.	The State where the shape is to be added. This is automatically set to the first empty location in the cell.
Description	Field to enter an 8-character name for the cell. ***Note:*** Individual names cannot be assigned for each state.
OK	Save the Shape to the specified cell and close the dialog.
Cancel	Close the dialog without saving the shape to the specified cell.

5. Click **OK** to finish.



New library with example shape in cell 0

7.6.2.5 Editing an Existing Shape

First, use the **Place** button and **State** dropdown to place all of the shapes for all states of the cell to be edited on windows. The shapes are then edited using the drawing tools. The cell being edited must then be deleted from the shape library. The **Delete shape** button deletes all shapes in a cell. (The shapes cannot be restored with the Undo command.) The modified shapes are then saved back to the cell in state order sequence.

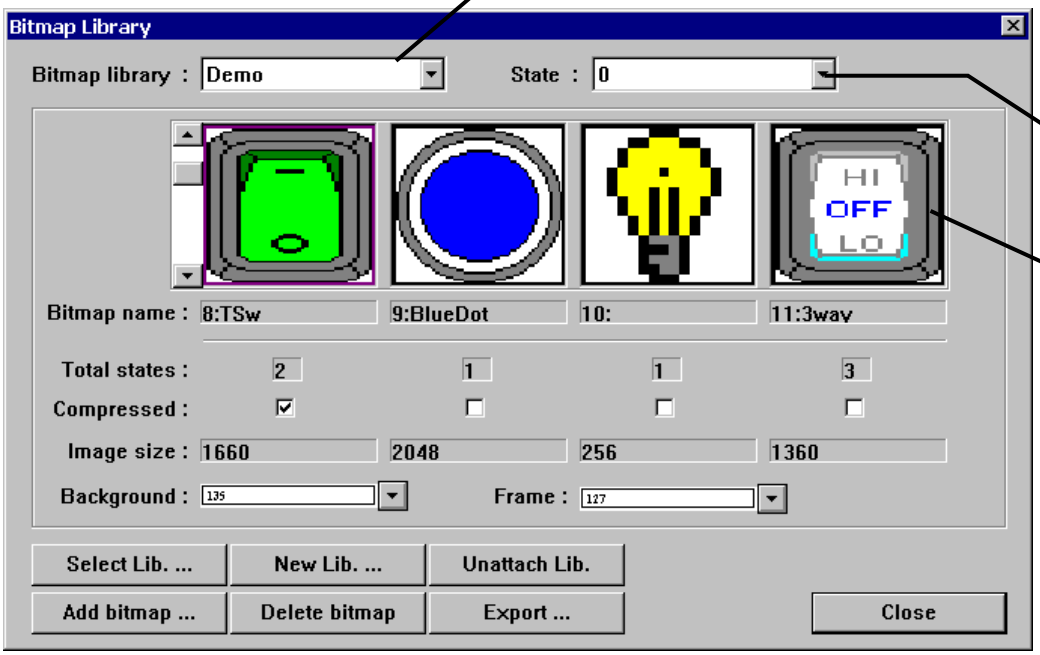
Note: There is no provision for editing or deleting only one state of a cell that has multiple states.

NOTE: New and modified libraries are not saved to disk until the project is saved. Closing the project without saving loses changed library data and new libraries!

7.6.3 Bitmap Library Operations

A Bitmap graphics (BMP) is simple pixel data. This data comes from screen captures, paint or drawing software packages that can generate *.bmp files. Bitmaps smaller than 250 KB and up to 256 colors are allowed.

7.6.3.1 Bitmap Library Dialog



Library Selector: Use dropdown to select any Bitmap library attached to this project.

State Selector: Use dropdown to select the viewing state of the cell's bitmaps.

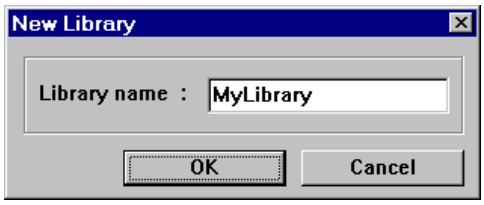
Cell: Each cell can have a maximum of 32 Bitmap graphics, the first Bitmap in the cell represents zero (0), the second represents one (1) and so on. Toggle the state selector to view each Bitmaps state in a cell.

Bitmap Library Dialog Summary

Item	Description
Bitmap name	A brief name assigned to the library cell. Names cannot exceed 8 characters.
Total states	Total number of states saved into the specified library cell. Up to 32 states can be assigned to a cell.
Compressed	These check boxes indicate if the graphic is compressed. Compressed graphics typically take up less memory. If compressed graphics are used, the No of windows setting in the System Parameters menu, General tab, can only be a maximum of 5. Be sure the graphic does not exceed the 256K graphic limit before compression.
Image size	This is the size in bytes of the graphic for download and memory purposes.
Background	Allows the background field of the library objects to be changed. This is used to see how a window's background color affects the way a bitmap appears. The background color of the bitmap object in the library does not follow the bitmap when it is placed on the window. The color number 1~256 is displayed in box.
Frame	Used when viewing SYS_XXX libraries to uniformly change the color of all buttons in the library. This has no effect on other libraries. The color number 1~256 is displayed in box.
Select Lib. …	Attach an existing library to the current project.
New Lib. …	Attach a new (blank) library to the current project.
Unattach Lib	Remove a library from the current project. Objects that use BMPs from the deleted library are displayed as rectangle outlines. No warning is given when this happens. Be sure you want to do this operation.
Add bitmap …	Add Bitmap graphics to the specified cell.
Delete bitmap	Delete Bitmap graphics from the selected cell. The result of this operation is a completely empty cell.
Export …	Save the Bitmap to a file with *.bmp format. This is useful for editing a BMPs with graphic editing programs.
Close	Closes the Bitmap Library Dialog

7.6.3.2 Creating a Bitmap Library

Click on the **New Lib. …** button. A Dialog Box for entering a file name appears.

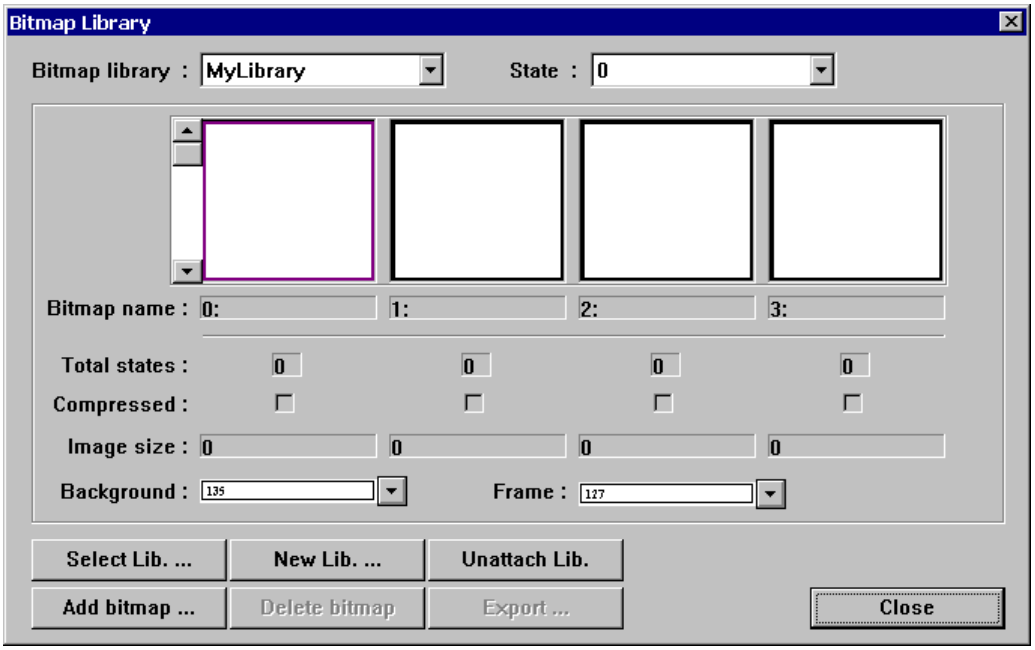


Enter the name of the bitmap library to be created.

Note: Be sure the library name is not already used. Creating a new library with the same name as an old library erases the old library.

Note: All bitmap libraries are automatically assigned the *.blb extension.

NOTE: New and modified libraries are not saved to disk until the project is saved. Closing the project without saving loses changed library data and new libraries!



An empty new library dialog

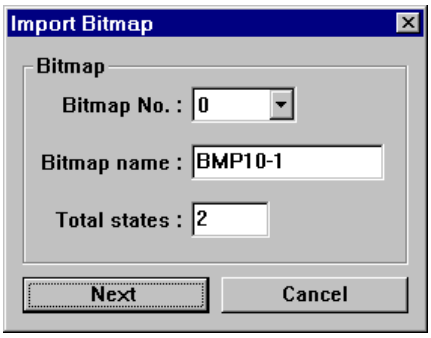
7.6.3.3 Opening a Bitmap Library

Click on the **Select Lib…** button. A Standard Dialog Box for selecting a file appears. Select the name of the bitmap library to be opened.

7.6.3.4 Adding a BMP to a library

1. Click on the **Add bitmap…** button, the following dialog box pops up. Fill in cell no., assign a name to the bitmap, and enter the number of Total states.

Item	Description
Bitmap No.	The cell where the bitmap graphics are to be imported.
Bitmap name	A brief name for the bitmap that is displayed below the cell. Up to 8 characters are displayed.
Total states	The total number of states to be in the cell. All states, up to the number specified, must be assigned a bitmap.
Next	Continue on to the Get Bitmap dialog.
Cancel	Close the dialog without saving the bitmaps to the specified cell.

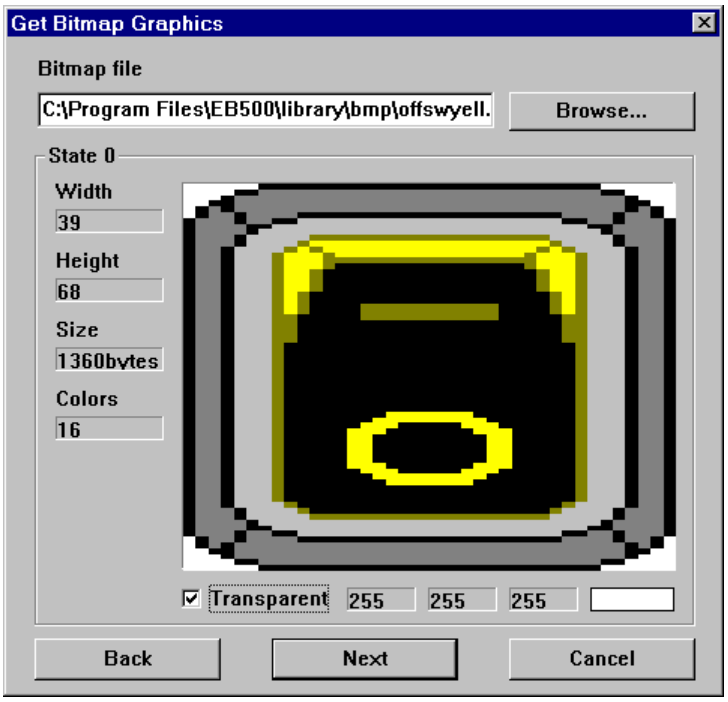


Note: Once a cell has bitmaps added, it cannot be updated later with additional bitmaps. To update a cell, the bitmaps must first be exported, the cell deleted, and the updated graphics added; along with the exported bitmaps.

When done, click on **Next** button…

The **Get Bitmap Graphics** Dialog opens.

2. Enter the path to the *.BMP in the **Bitmap file** field or click the **Browse…** button to select it with the familiar file selection dialog.



The title of the frame shows the State of the bitmap being imported. A preview of the bitmap is displayed in the frame. The preview is sized to fit in the area of the dialog. This may distort the bitmap image.

Note: When the bitmap is placed on the screen, it is sized to the proper proportions.

Information about the bitmap is displayed along the left side of the dialog.

Note: Width and Height values are in pixels.

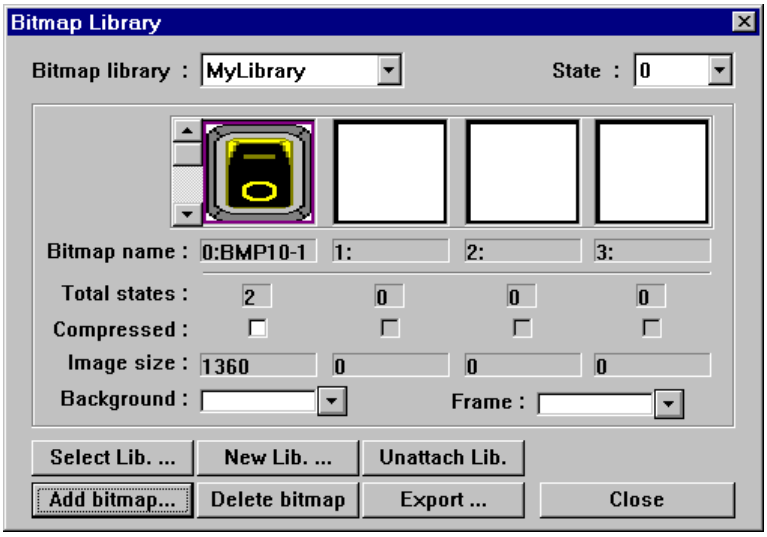
3. One color of the bitmap can be selected as transparent. This is useful for eliminating background colors.

   To make a color transparent simply select the **Transparent** check mark and then click anywhere in the area to be made transparent. In the example, white is selected as the transparent color.

   Note: The transparent fields cannot be filled in manually.

4. Click on **Next** and continue to import bitmaps until the specified State numbers are populated. The graphic importer guides you through each state to get and install BMP graphics. The Back button can be used to review the entered bitmaps.

5. Click on **Finished** when final selection is made.



New library with example bitmap in cell 0

7.6.3.5 Editing an Existing Shape

First, **Export** all of the bitmaps for all states of the cell to be edited to a directory.
Next, the bitmaps are then edited using a bitmap editing utility such as “MSPaint®”.
Then, the cell being edited must then be deleted from the bitmap library.

Note: The **Delete bitmap** operation deletes all bitmaps in a cell. There is no undo for this operation.

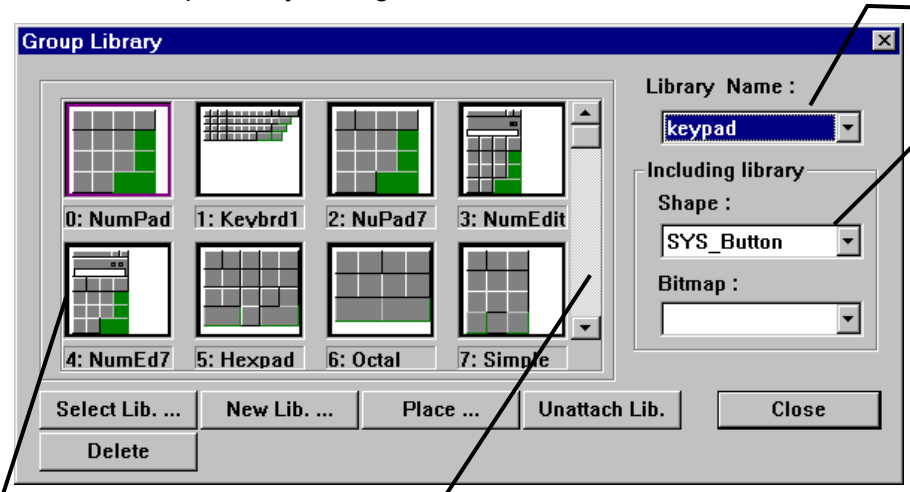
Finally, the modified bitmaps are then added back into the cell in State order sequence.

Note: There is no provision for editing or deleting only one state of a cell that has multiple states.

7.6.4 Group Library Operations

"Group Libraries" enable multiple parts and drawing objects to be combined and saved, and then called up whenever necessary. The Group objects saved in these libraries can be viewed and selected using the Browser function. Since groups of objects may include parts that refer to Shape or Bitmap graphics, the related Shape and Bitmap libraries must be attached to the project before calling up group objects.

7.6.4.1 Group Library Dialog



Library Selector: Use dropdown to select any Group Library attached to this project.

Including library: Use dropdowns to view Shape and Bitmap libraries used in creating the displayed Group library. If the related Shape and Bitmap libraries are not included with the project, the group objects are displayed as boundaries of shapes and/or bitmaps. To correct this, go to the Shape and Bitmap Library's dialog and Attach the required libraries.
Shape and Bitmap Libraries must be attached to the project before a group that uses them can be placed.

Cell: Each cell holds one group. Each cell has a number with an 8 character name.

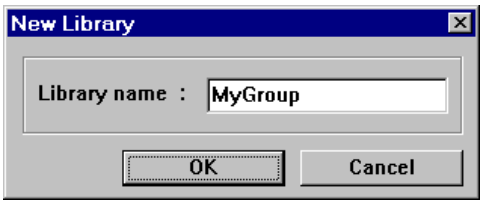
Scroll bar: Use the scroll bar to view above and below the displayed cells. There are 48 cells available in a library (0-47).

Group Dialog Summary

Item	Description
Select Lib. …	Attach an existing library to the current project.
New Lib. …	Attach a new (blank) library to the current project.
Unattach Lib.	Remove a library from the current project.
Place…	Place the group in the selected cell on the window. The group can then be moved into position as desired. Groups cannot be resized. Placed Groups can be ungrouped and edited if desired.
Delete	Delete group from the selected cell. The result of this operation is a completely empty cell.
Close	Closes the Group Library dialog.

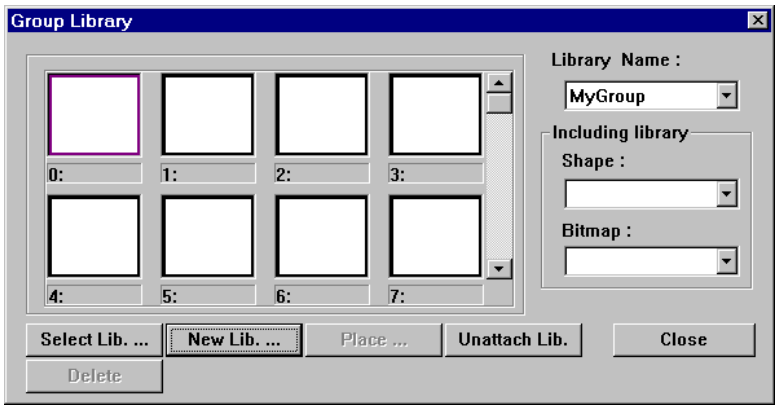
7.7.4.2 Creating a Group Library

Click on the **New Lib. …** button. A Dialog Box for entering a file name appears. Enter the name of the group library to be created.



Note: Be sure the library name is not already used. Creating a new library with the same name as an old library erases the old library.

Note: All group libraries are automatically assigned the *.glb extension.



An empty new library dialog

7.6.4.3 Opening a Group Library

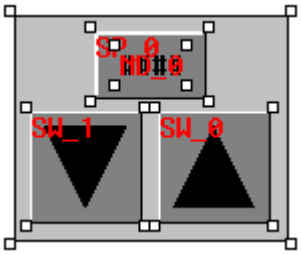
Click on the **Select Lib…** button. A Standard Dialog Box for selecting a file appears. Select the name of the Group library to be opened.

NOTE: New and modified libraries are not saved to disk until the project is saved. Closing the project without saving loses changed library data and new libraries!

7.6.4.4 Adding a Group to a Library

1. Open a window and Draw the graphics by using the drawing tools. Also, add any parts such as lamps, function keys, etc.

Example: The group shown below uses Shapes, a Numeric Data part and Rectangles.



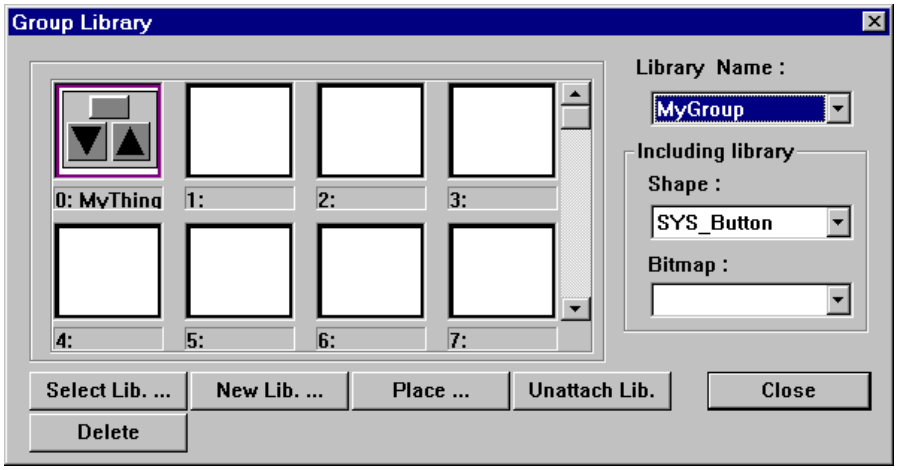
2. Use the arrow tool to select all the candidate objects on the window by dragging around them.

3. Then, in the Library menu, select the Group, Save to library submenu or click on the  tool.

4. Select the appropriate Group library. Then, fill in the Description and Group No. fields.

Item	Description
Group Library	Used to select the Group library to hold the group.
Description	A brief name for the group that is displayed below the cell. Up to 8 characters are displayed.
Group No.	The cell where the bitmap graphics are to be imported.
OK	Save the group to the specified cell and close the dialog.
Cancel	Close the dialog without saving the group to the specified cell.

5. Click **OK** to save the selected objects to the group library.



New library with example group in cell 0

7.6.4.5 Editing an Existing Group

First, place the group on a window.
Next, edit the group as desired.
Then, delete the cell that has the original group. There is no undo for this operation.
Finally, Save the edited group to the empty cell.

NOTE: New and modified libraries are not saved to disk until the project is saved. Closing the project without saving loses changed library data and new libraries!

7.6.5 System Libraries

There are 4 built in System Libraries. 2 Shape libraries and 2 bitmap libraries These Libraries are available for every project and do not count towards the 10 library limit. The libraries are not stored in the \Library sub-Folder and cannot be Unattached from a project. System shapes cannot be edited or deleted. New shapes cannot be added to these libraries.

> ***Note:*** Use System Shapes and Bitmaps to reduce the size of a project and to give it a consistent look.

7.6.6 Library Toolbar Functions

To call up an item from a library, either select one of the tool bar's icons or use the pull down menu's [Library] command.

Save selected shape or bitmap to its respective library.

Call up a Shape library.

Call up a bitmap library.

Save group of objects to a Group library.

Call up a Group library.

7.6.7 Shortcut for changing a Part's assigned Shape or Bitmap

Normally, a part is assigned a shape or bitmap when it is first edited, before placement on the window. The Shape or Bitmap can then be changed by going to the Part's attributes dialog, clicking on the Shape/Bitmap library button and then selecting the shape/bitmap. All of the dialogs are then closed.

This process can be bypassed as follows.

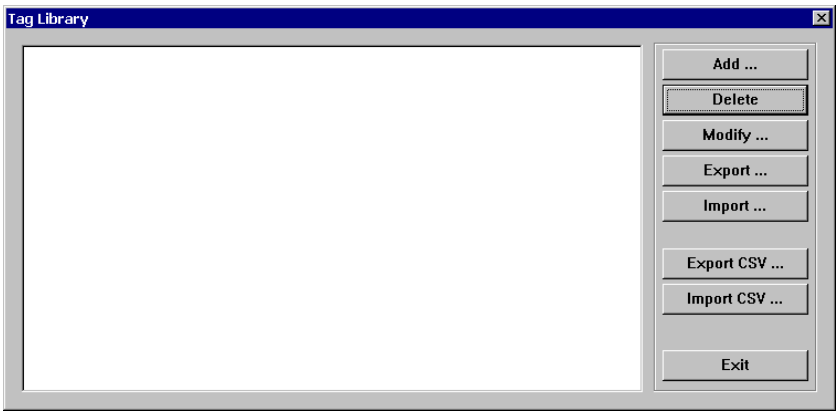
1. Simply click on the Part so its handles are showing.
2. Then click on the library tool to call up the appropriate type of library,
3. Select the Library and Cell then click OK in the library dialog.
4. The dialog closes with the selected shape/bitmap automatically assigned to the part.

7.7 Tag Definition and Use

A tag is a name that is used to reference a piece of data. Tags are assigned to any register or coil available. All of the registers and coils used in your application can be defined beforehand.

The Tag Library is an important part of a touchscreen's project. All Part definitions can be linked to this Tag Library. There are many advantages of a centralized library. You can change the tag data reference in the Tag Library, and all the instances where the tag is used are updated.

The Tag Library dialog box is used to define new tags and edit the properties of existing tags. You can access this dialog box by selecting **Tag…** from the Library menu.



Item	Description
Add …	Use to Add tags to the library.
Delete	Deletes a tag from the library. ***Note:*** Tag cannot be recovered with Undo.
Modify …	Edit an existing Tag's properties.
Export …	Exports defined tags to a file in EasyBuilder format.
Import …	Imports previously defined and exported tags from a project.
Export CSV …	Exports defined tags to a file in CSV format.
Import CSV …	Imports previously defined and CSV exported tags from a project.
Exit	Save and close the Tag Library

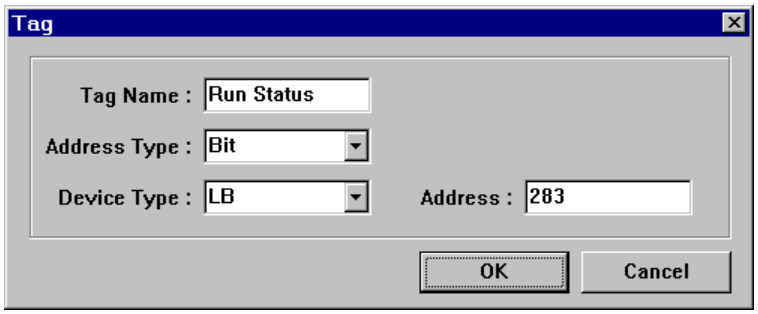
Note: Tags exported in EasyBuilder format can not be edited by other programs. Tags exported as CSV can be edited by programs that can view standard CSV files.

7.7.1 Adding a Tag to the Tag Library

1. In the Library menu, select the **Tag…** item or click on the  tool.

2. Click the **Add …** button. The Tag Dialog pops up. Fill in the Tag Name, Address Type, Device Type and Device address.

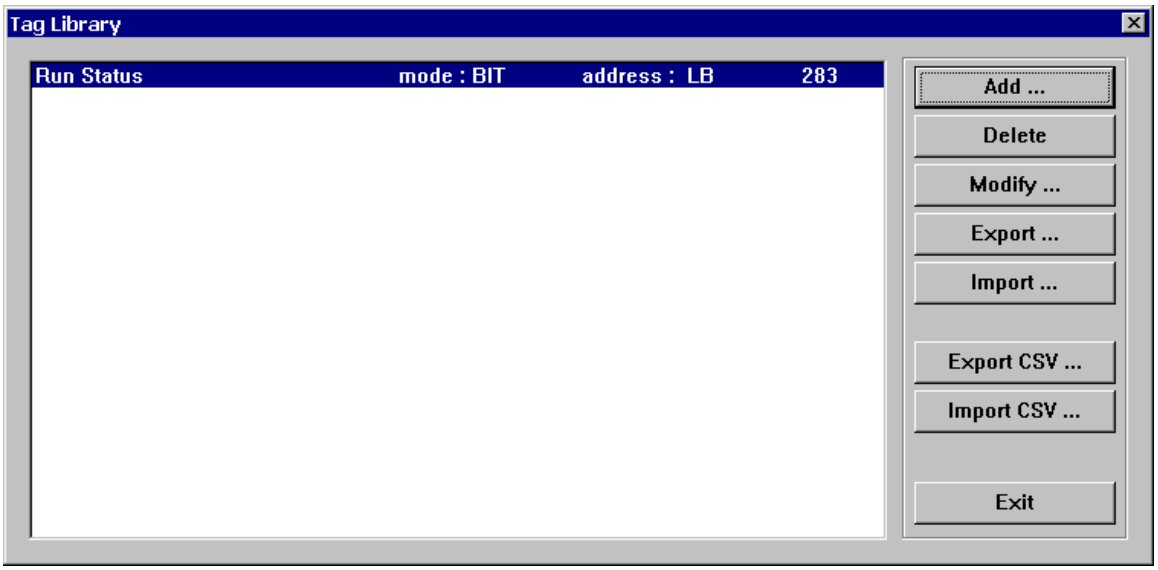
(Example shown to the right.)



Item	Description
Tag Name	A brief name for the tag. ***Note:*** Only the first 18 characters are displayed in dialogs.
Address Type	Select Bit or Word. This determines the type of Parts that have access to this tag.
Device Type	Prefix for the data type. This is dependent on PLC type.
Address	Data bit/word numerical address. ***Note:*** Extended Addressing is allowed when applicable.
OK	Saves tag to Tag Library and closes dialog.
Cancel	Close dialog without saving the entered information.

3. Click **OK** to add the Tag to the Tag Library.

Note: Any changes that you make to tags are not saved until you save your project.



Tag Library with example tag

7.7.2 Editing Tags

Call up the Tag Library, select the tag to edit and click on the **Modify …** button. Change the fields as desired. If the Address is changed, all places where the tag is used in the project are changed.

> ***Note:*** If the **Tag Name** or **Address Type** fields are changed, all places where that Tag is used are changed to the first Tag in the Tag Library. Once OK is clicked, this happens. This action cannot be recovered using the Undo command.

7.7.3 Deleting Tags

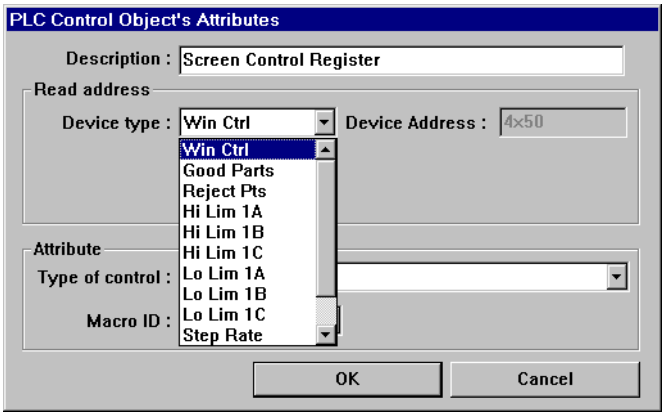
Call up the Tag Library, select the tag to delete and click on the **Delete** button.

> ***Note:*** All places where that Tag is used are changed to the first Tag in the Tag Library. Once **Delete** is clicked this happens. This action cannot be recovered using the Undo command.

7.7.4 Using Tags

Once a tag is defined it can be used anywhere a Read or Write Device is needed.

Check the **Tag** box in the address frame to select a Tag for the Device Type and Address. (Example shown above.)



The **Device type** dropdown becomes the list of defined tags. Scroll through the list and select the desired Tag.

7.7.5 Exporting and Importing Tags

Defined tags can be exported and imported to files for archiving, editing or use in other projects. Only tags exported to CSV files can be edited by other programs.

Export …: Exports tags in EasyBuilder format. This is the preferred format for archiving tags. Tags in this format can be imported into other projects but not edited with other programs.

Import …: Imports tags from a previous EasyBuilder formatted export operation.
Note: This does not import tags from a PLC's project.
Note: If the imported tags were originally exported from a project with a different PLC driver, they will be converted to the current PLC driver. Be sure to check tags for proper conversion.

Export CSV …: Exports defined tags to a file in CSV format. These tags can be viewed and edited with most text editors and spreadsheet programs.

Import CSV …: Imports tags from an exported CSV file or new file in CSV format.
Note: If the CSV file has invalid data in one of the fields, One of the following occurs:
A "?" is substituted for the invalid field

- The project compiles with an error
- The project simulates/runs with errors
- Be sure to check tags for proper format.

Note: If the imported tags were originally exported from a project with a different PLC driver, they will be converted to the current PLC driver. Be sure to check tags for proper conversion.

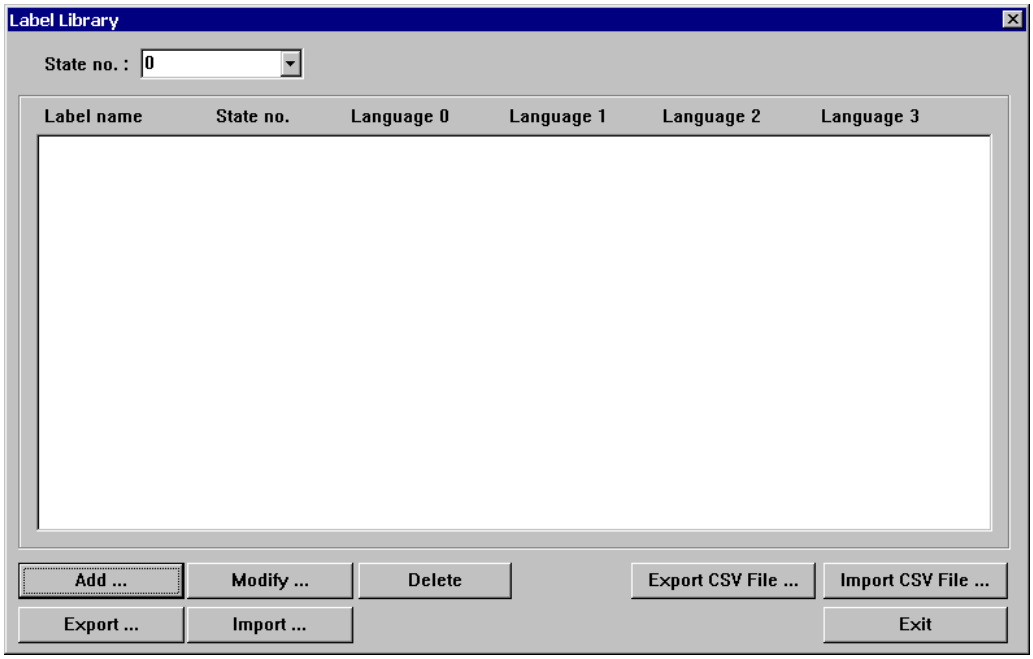
Note: CSV Format

Field 1	Field 2	Field 3
Tag Name	Device Type	Device Address

7.8 Label Definition and Use

A Label is displayed text that is used to identify data or states. Labels are assigned to any Part available. All of the possible text for several languages or conditions can be pre-defined. The Label Library allows you to change the label texts. When this is done, all instances of label use are updated.

The Label Library dialog box is used to define new labels and edit the properties of existing labels. You can access this dialog box by selecting **Label …** from the Library menu.

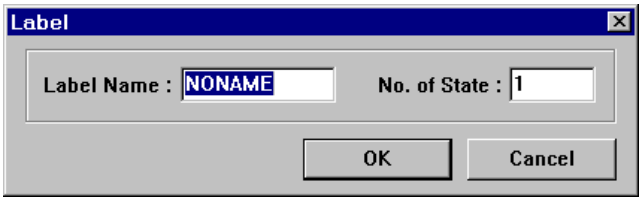


Item	Description
State No.	Select the State of the listed labels for display.
Add …	Use to Add Labels to the library.
Modify …	Edit an existing Label's properties.
Delete	Deletes a Label from the library. ***Note:*** Label cannot be recovered with Undo.
Export	Exports defined Labels to a file in EasyBuilder format.
Import	Imports previously defined and exported Labels from a project.
Export CSV File …	Exports defined Labels to a file in CSV format.
Import CSV File …	Imports previously defined and CSV exported Labels from a project.
Exit	Save and close the Label Library.

7.8.1 Adding a Label to the Label Library

1. Select in the **Library|Label …** menu or click on the  tool.

2. Click the **Add …** button. The Label Dialog pops up. Fill in the Label Name and Number of States for the Label. Once this information is entered, the Label appears in the Label Library with no text assigned. The **Modify …** button must then be clicked to edit the label texts.



Item	Description
Label Name	A brief name for the tag. ***Note:*** Only the first 18 characters are ever displayed in dialogs.
No. of State	Enter the number of States this label is to have. ***Note:*** The maximum number of States is 32
OK	Saves Label information to Label Library and closes dialog.
Cancel	Close dialog without saving the entered information.

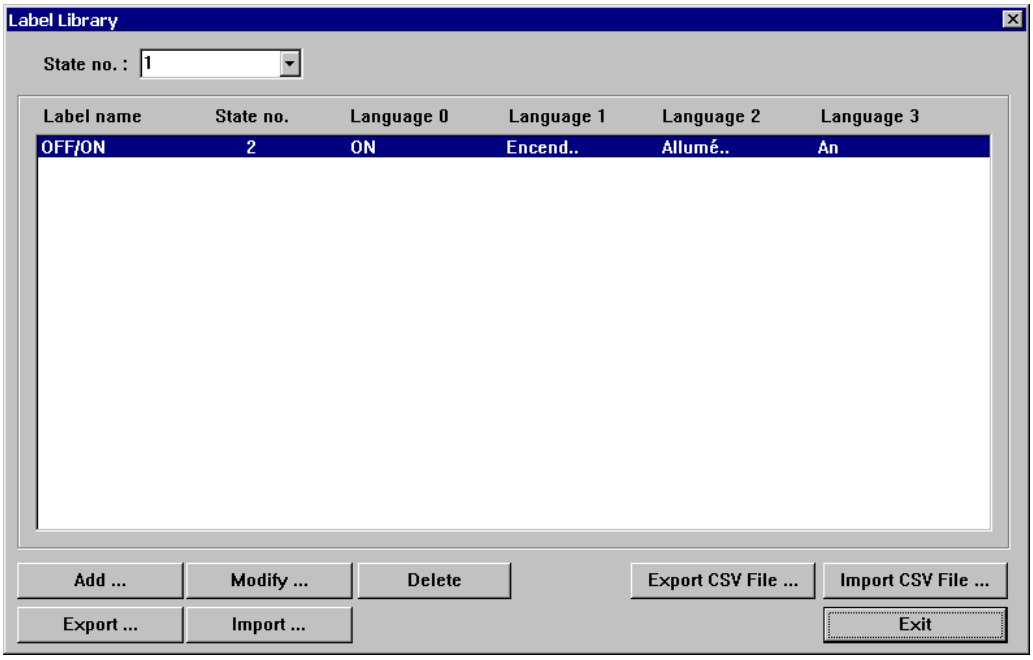
7.8.2 Editing Labels

Call up the Label Library, select the Label to edit and click on the **Modify …** button. The **Label Content Setting** dialog is displayed. Change the fields as desired. If the Language text fields are changed, all places where the Label is used in the project are changed. Use the **State No.** Dropdown to view the labels in their different states.

Note: Once **OK** is clicked, the label is saved and the dialog closed. This action cannot be recovered using the **Undo** command.

Example: Label entry for a label that displays OFF and ON depending on state. The label is assigned 2 states. The Languages are English, Spanish, French and German.
The text entry for the State 0 (“OFF”) is shown.

Use the **Change State No. …** button to edit the total number of states for the label being edited.



Label Library showing the example tag in State 1.

Note: Label text is truncated after 6 characters in the **Label Library** display. In actual use, the full label is visible.

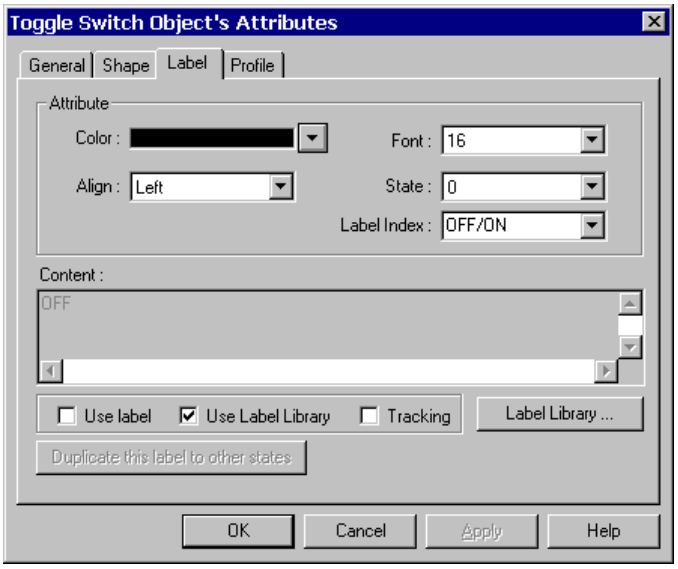
7.8.3 Deleting Labels

Call up the Label Library, select the Label to delete and click on the **Delete** button.

Note: Once **Delete** is clicked, all places where that Label is used have blank text. This action cannot be recovered using the **Undo** command.

7.8.4 Using Labels

Once a Label is defined, it can be used anywhere text is needed. This includes all Parts that have Label Tabs, Text, and Alarm Scan and Event Log objects.



Example of Toggle Switch Object's Label tab using the example Label created above.

1. Check the **Use Label Library** box to enable label use.

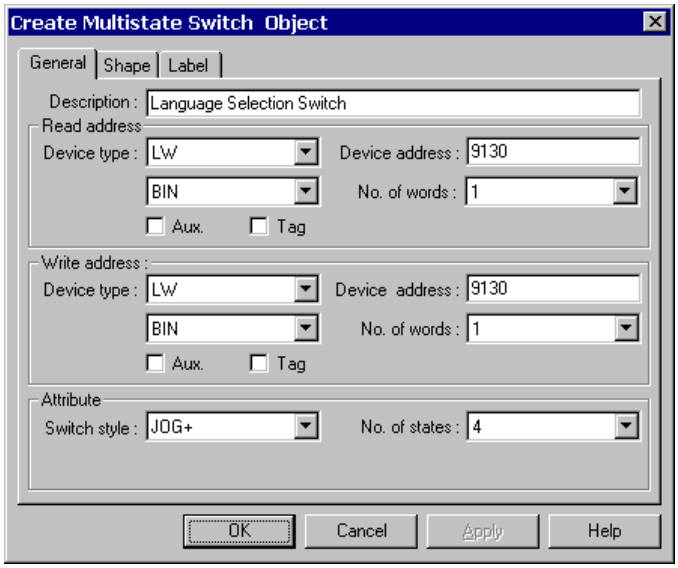
2. Select the desired Label from the **Label Index** dropdown.

3. Fill in other settings as desired.

Note: The **Label Library** button can be used to call up the Label Library for adding labels without exiting the Part or Object dialog.

After a label is assigned to a part, the Language displayed is changed during Simulations or Application mode by changing the value in System Reserved word **LW 9130**. (0 corresponds to Language State 0, 1 corresponds to Language State 1, 2 corresponds to Language State 2, and 3 corresponds to Language State 3.)

A convenient way for an operator to change language is to assign LW 9130 to a **Multistate Switch** with the **JOG+ Switch style** and **No. of states** set to 4.
Hint: Make a Label in the Label Library for the Multistate Switch so it is multi-language as well!



7.8.5 Exporting and Importing Labels

Defined Labels can be exported and imported to files for archiving, editing or use in other projects. Only Labels exported to CSV files can be edited by other programs.

Export …: Exports Labels in EasyBuilder format. This is the preferred format for archiving Labels. Labels in this format can be imported into other projects but not edited with other programs.

Import …: Imports Labels from a previous EasyBuilder formatted export operation.
Note: Labels that were originally exported from a project with a different PLC driver are imported with no detrimental effects.

Export CSV …: Exports defined Labels to a file in CSV format. These Labels can be viewed and edited with most text editors and spreadsheet programs.

Import CSV …: Imports Labels from an exported CSV file or new file in CSV format.

Note: CSV Format

Field 1	Fields 2~5	Fields 6~9	Fields 10~13	Fields … etc.
Label Name	Language 0~3 text for State 0	Language 0~3 text for State 1	Language 0~3 text for State 2	Language 0~3 text for State … etc.

7.9 Security

The HMI provides three user levels and one project level security settings. User security is activated by checking the Security Control check box in the System Parameters' dialog Security tab. Project security is activated in the System Parameters' General Tab Password setting.

7.9.1 Security Levels

7.9.1.1 Project Security: Project security has only one level.
7.9.1.2 User Security: There are 3 levels of user security. Level 0 is the lowest level of security (all have access to these windows regardless of the password), Level 1 is a middle level allowing access to level 1 and level 0 windows. Level 2 is the highest level of security and can access all levels.

7.9.2 Passwords

Passwords are Digits only (0-9); Characters cannot be part of a password. Passwords cannot be negative or numbers with decimal points.

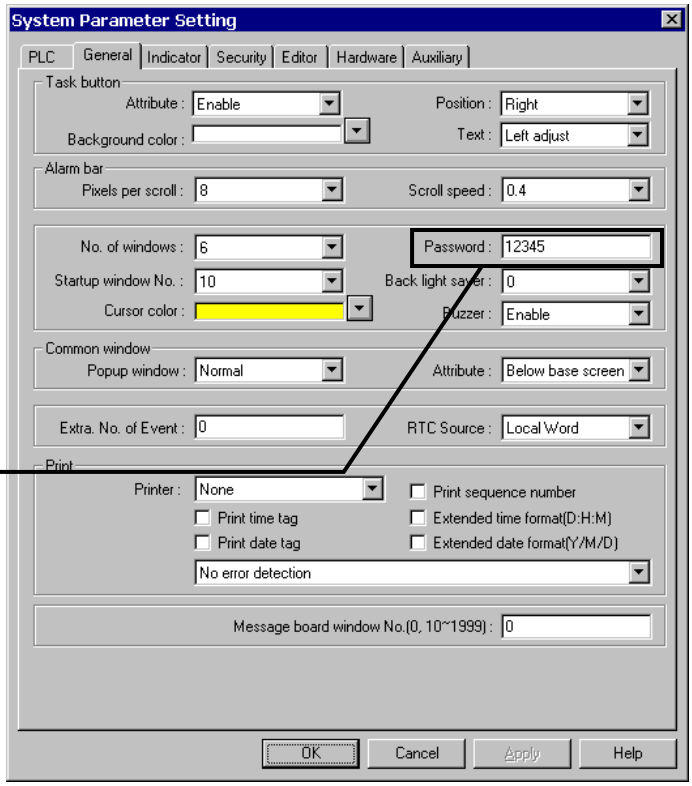
7.9.2.1 Project Security: Each levels password is a double word value 0 to 4999999999 (up to 10 digits). A password of 0 for a Project disables security for that project.
7.9.2.1 User Security: Each Level's password is a double word value (0 to 99999999). A Security level password of 0 is a valid password for locking out a level.

7.9.3 Assigning Security

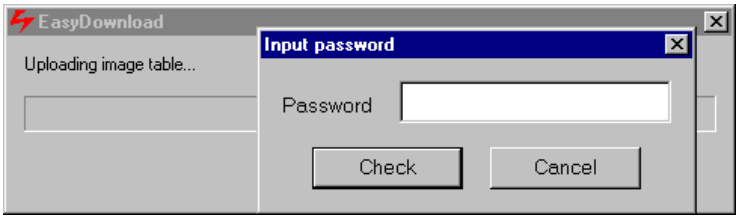
7.9.3.1 Project Security: Go to the **System Parameters**|**General** Tab, **Password** field to activate project security. Activate the password option by entering any number greater than 0. This locks the project after it is downloaded so it cannot be uploaded without first entering the password.

Note: Password protection does <u>not</u> prevent the user from Downloading. There is no protection to prevent a project from being overwritten.

Example showing a password of **12345** being assigned to a project.

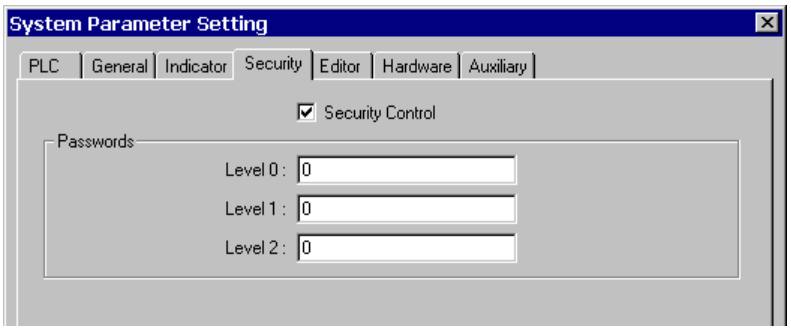


Dialog asking for Project password on upload.



7.9.3.2 User Security: Activate User security by selecting the **System Parameters|Security** Tab, and **Security Control** option.

Once user security is enabled, all base windows have a security level assigned to them. A window can only be accessed if the security level or higher for that window is active.



Note: The only exception to this is the Startup window. That window comes up when the unit is started. The security level is still at zero on startup, even if the startup window is set at a higher level. In this case, once the startup window is closed, a password level change is required to get back to it.

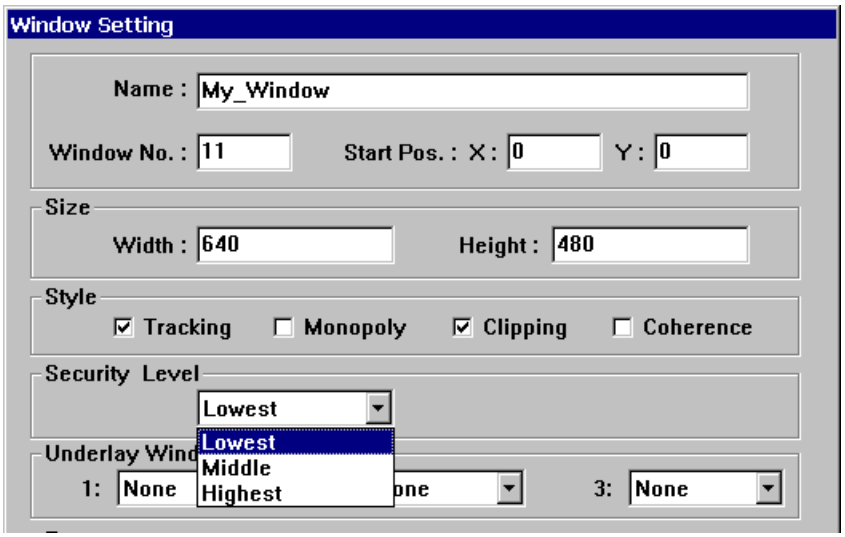
Note: The HMI only checks the security level when the user tries to change base windows with a "Change window" Function Key.

Note: Security does not inhibit Function Popup Windows, Direct Windows, Indirect Windows, and PLC Control Window Change parts.

Fill in the Password Level fields to set the passwords for each level.

Level	Window Setting	Description	Typical Use
0	Lowest	All have access to these windows regardless of the password. No Level 1 or 2 windows can be accessed from this level.	Operators and general personnel
1	Middle	Allows access to level 1 and level 0 windows. No Level 2 windows can be accessed from this level.	Setup and maintenance personnel
2	Highest	Allows access all levels	Supervisors

Window Setting Dialog showing the **Security Level** setting.



7.9.4 System Reserved Local Word Usage with Security

This applies to User Security only.

Note: The Project Password is not stored in any user accessible memory location.

Internal Local Words **LW9040~9043** are the reserved words for implementing security passwords.

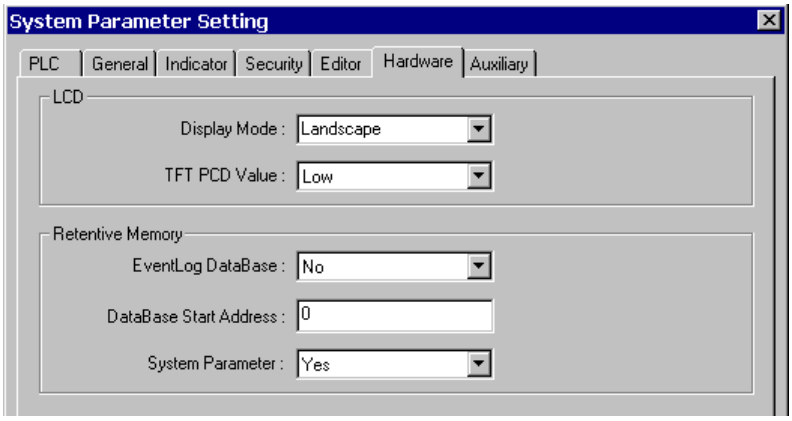
Local Word	Attribute	Description
LW9040 - LW9041	Write/Read	Double word, Different levels can be activated by changing the value in this register to a corresponding security password. Contains the last password value.
LW9042	Read only	Contains the current active security level.
LW9043	Write only	Allows the project to force the HMI to a lower security level. ***Note:*** A lower level cannot be forced to higher level.

7.9.5 System Reserved Retentive Word Usage with Security

This applies to User Security only.

The security passwords are stored in retentive memory locations when the **System Parameter|Hardware**, **System Parameter** dropdown is set to **Yes**.

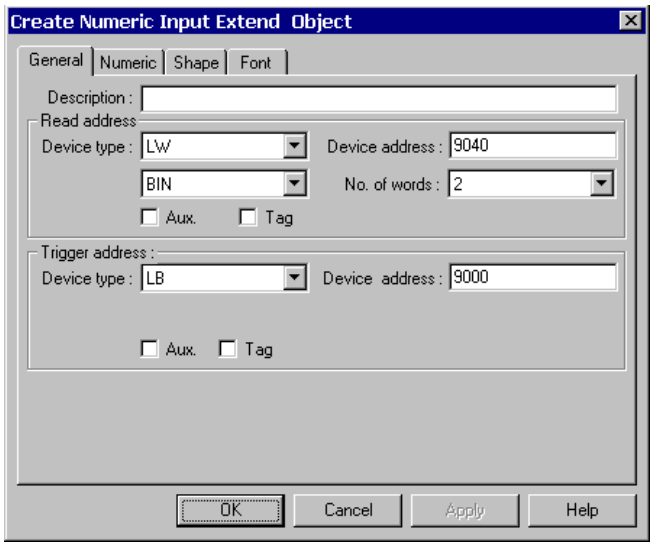
Note: This allows the security passwords to be changed if access to them is provided in the project.
They also are available for review if the project is uploaded out of the HMI and decompiled.

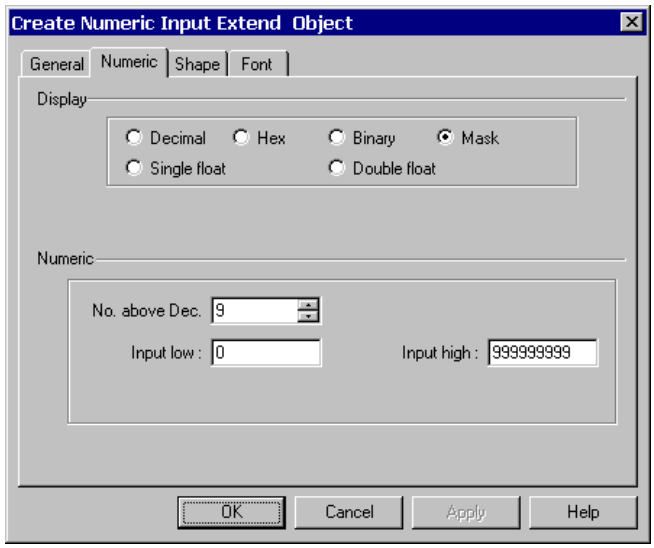


Retentive Word	Description
RW60071	Security Enable: 1: enables the security password feature, 0 disables it.
RW60072	Level 0 password (Lowest). 2 Word value
RW60074	Level 1 password (Middle). 2 Word value
RW60076	Level 2 password (Highest). 2 Word value
Local Bit	
LB9044	Set to ON to activate Security Password System Parameter changes made in the Retentive Word area. The System automatically resets this bit to 0 when done.

7.9.6 How to Implement User Security

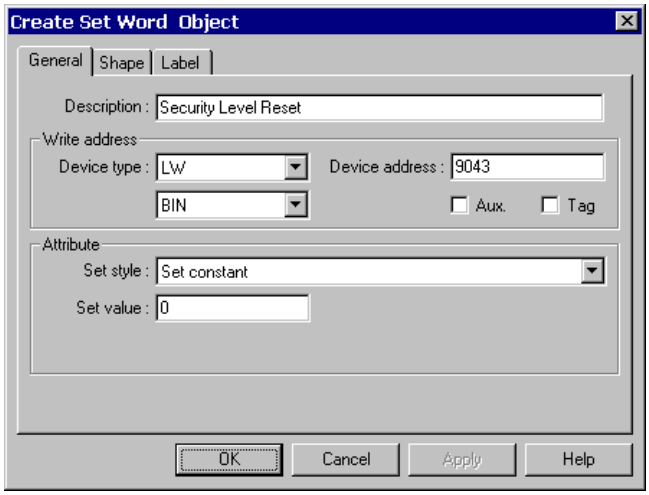
Use a data entry object that accesses **LW9040**. In the Numeric Tab select it to be displayed as "**Mask**" and as 9 digits above the decimal point with the high limit as **999999999**. Use an appropriate shape and for display. Be sure to provide a keypad in the window for data entry and a way to activate the Trigger bit. See the Numeric Input Extend section for more details about data entry objects.





Provide a way to force the level back to 0 when the secure operation is finished.

This is usually done by a Set Word part acting on **LW9043** with Set style **Set at window close** and Set value of **0**. No Shape or Label needs to be assigned to the part (this makes the part invisible). The part can be placed on any middle or high security window. An example of such a Set Word Object dialog is shown on the right.



7.9.7 Additional Security Through the PLC

Additional security can be provided by using PLC data to control **Direct** and **Indirect** windows. Program the PLC to prevent these windows from being displayed unless a password compare is positive in the PLC.

These windows are put on the Common window for universal access or put only on special Base windows to further limit their accessibility.

7.10 Print Operations

Print Operations can be used for obtaining a hardcopy of events, tracking data during a process, documenting a condition for data collection purposes and other various record keeping activities. Print functions facilitate the printing of data that is sent out as a graphic through the parallel printer port.

Note: Actual printer setup is done in the **System Parameter|General** tab. If no printer is selected these features have no effect.

Note: Not all models have print capabilities.

7.10.1 Compatible printers

Supported Printers

Protocol	Impact Printers	Inkjet Printers	Laser Printers
EPSON ESC/P2 protocol	LQ-300, LQ-300K+, LQ-570, LQ-670, LQ-1600	Stylus Color 400, 800, 850, 900, 950 Stylus Photo 750	EPL-5800, EPLN-N2010, EPL-N2750, EPL-N4000+, EPL-N2010
HP PCL/Simple Page Mode **HP PCL protocol**		HP DeskJet 450, 640C, 656C, 845C, 920C, 930C, 1125C, 6122, 5550	HP LaserJet 5L, 6L HP LaserJet 2200D
SP Printer **EPSON ESC protocol**	EPSON ESC protocol compatible 9-pin printers Point of Sale 9-pin printers		
None	Printing Disabled	Printing Disabled	Printing Disabled

7.10.2 Screen Printing with Function Buttons

Select the **Function Key** menu item or tool to define a window touch area for triggering print output. Select the **Hardcopy** option in the **Create Function Key Object** dialog. Use the **Attributes ....** button to select the desired print action.

Selection	Print action
Print Text & Meter & Trend	Prints bit and numerical data parts along with drawn objects and text placed on the display.
Print Text & Meter & Trend & All Shape but not including pattern	Prints the outline of bit and numerical data parts along with drawn objects, text and shapes placed on the display.
Print Text & Meter & Trend & Bitmap	Prints bit and numerical data parts along with drawn objects, text and bitmaps placed on the display.
Print Text & Meter & Trend & Bitmap & All Shape but not including pattern	Prints the outline of bit and numerical data parts along with drawn objects, text, bitmaps and shapes placed on the display.
Print Text & Meter & Trend & Bitmap & All Shape	Prints bit and numerical data parts along with drawn objects, text, bitmaps and shapes placed on the display.
Form Feed	Prints a Form Feed command.

Place and size the part in the window as desired. During run operations, when the function object is touched, the print action is triggered.

7.10.3 PLC Controlled Printing

There are two ways of printing by using a PLC data point. The **Screen hardcopy** control uses a bit in the PLC to trigger the printout and the **Report printout** control uses a word. If internal Local or Retentive data points are used, then the operator can trigger printing without the PLC getting involved.

7.10.3.1 Printing the currently active screen

This is used to dump the screen to the printer as a bitmap graphic. The **Screen hardcopy** PLC control monitors the specified Read address. When the trigger bit turns ON, the unit prints the active base window. Once the printout is done, the unit turns the controlling bit OFF. See the **Object Reference Guide** for more details about **PLC Controls**.

7.10.3.2 Printing a window in the background

Windows can be printed in the background while touchscreen operations are being carried out. Use the **Report printout** PLC control. This control prints the window number designated by its Read address. After the window is printed, the Read address is set to 0. For example, suppose that the Read address is D100. When the value of D100 is equal to 20, the unit prints out window 20 in the background. When the unit is finished printing, the zero is written to address D100.

Note: System Register **LW9054** specifies what is printed out in the report. Set this register before triggering printouts.

Report printout options

- **0**: All Text, Meters & Trends in the window
- **1**: All Text, Meters, Trends & Shapes but no pattern fills
- **2**: All Text, Meters, Trends & Bitmaps
- **3**: All Text, Meters, Trends, Bitmaps, & Shapes but no pattern fills
- **4**: All screen objects.

Note: System Register **LW9055** acts as an offset for the Report printout control. Set this register before triggering printouts.

Example: A PLC Control / Report printout uses D100. If (**LW9055**) = 6, and D100 = 24 then the unit prints out window 30. After printing, the unit writes back 0 to D100.

7.10.4 Printing Events

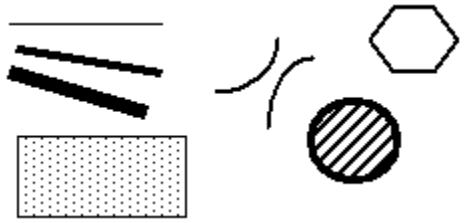
The **Event Logger** control allows events to be printed. This is specified whenever an event is being defined. The printing can occur when the Event is triggered or when the event returns to its normal state. The **System Parameters|General** Tab settings determine what is printed along with the Event message. The information selected is printed before the message.

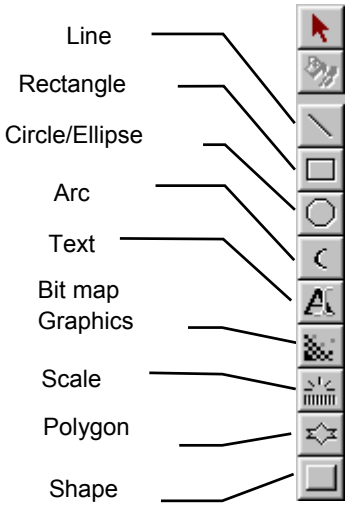
Information Item	Description
Print sequence number	The number of the event. Event numbering starts at 0.
Print time tag	The time that the print was triggered. (Hours:Minutes:Seconds)
Print date tag	The date of the printing. (Month:Day)
Extended time format (D:H:M)	Change the time format in the time tag to Days:Hours:Minutes.
Extended date format (Y:M:D)	Change the date format in the date tag to Year:Month:Day.

Note: These settings are not the same as the settings in the **Event Display** Part. Those settings determine what is displayed on the screen, not what is sent to the printer.

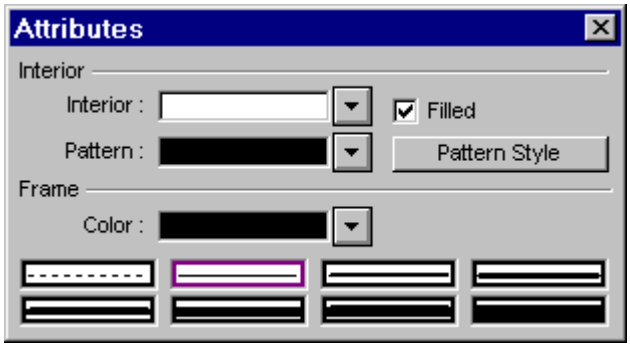
8.0 Drawing

8.1 Line/Rectangle/Ellipse/Arc/Polygon





The following **Attributes** dialog box is displayed when drawing Line, Rectangle, Ellipse, Arc and Polygon objects:



Interior: The Interior attributes are for objects that can be filled (Rectangle, Ellipse and Polygon). For objects that cannot be filled, this option is disabled. The filled attributes are not displayed unless the **Filled** check box is selected.
If the Filled option is selected, the following attributes can be set for Rectangle, Circle and Polygon.

Select Interior colors – Select a color from the 256-color (4 colors for grayscale units) selection (4 colors for grayscale units).

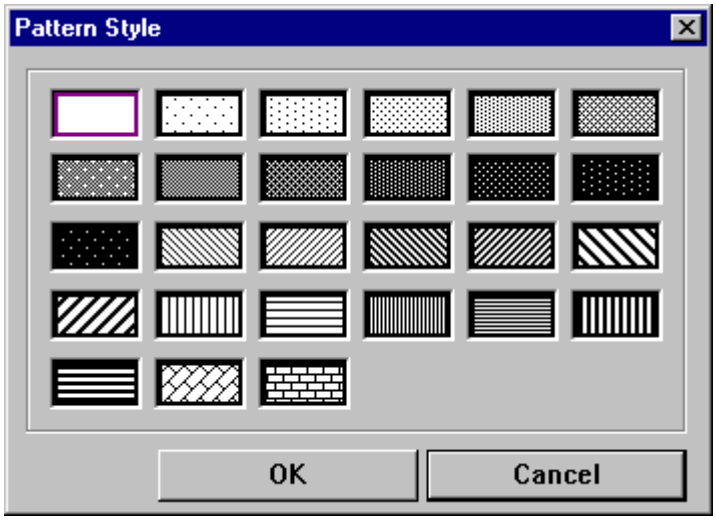
Select pattern – Choose one of the 27 different patterns available. (See below for pattern selection.)

Select pattern colors – Select a color from the 256-color (4 colors for grayscale units) selection (See below for color selection).

Frame: The Frame attributes are used to set how the lines for the object are displayed.
Select Line types – Choose one of the eight selections for the line or frame width.
Select a **Color**– Select a color from the 256-color (4 colors for grayscale units) selection (See below for color selection).

8.1.1 Note on Pattern Options

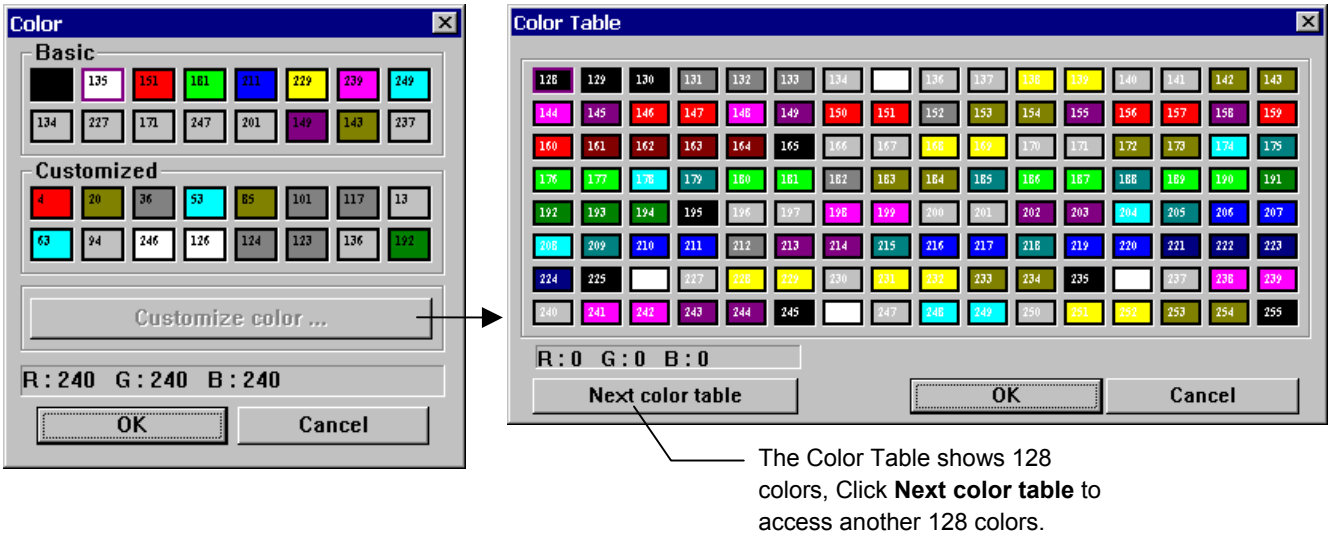
The Pattern Style dialog pops up when the Pattern Style button is clicked. Select one of the 27 standard patterns by clicking on it.



8.1.2 Note on Color Options

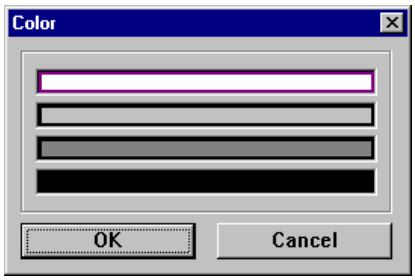
The color window pops up when a color selection dropdown is activated. One of the standard colors may be chosen or click on Customize color to access a full spectrum of color options. The color number is shown in small text inside each color's display. Additionally, the selected color's RGB numbers are displayed.

256 Color options



16 of the 256 colors are available from the "basic" palette. Additionally, 16 more colors can be put in the Customized palette. Click the Customize color button to customize the secondary palette. The "customized" palette can be changed during project development without affecting objects that are already placed. This allows the full range of 256 colors to be used.

4 color options



8.1.3 Drawing Lines, Rectangles, Ellipses, Arcs and Polygons

To draw a line, rectangle, ellipse or arc simply click and drag the mouse across the window after the appropriate drawing tool and attributes have been selected. Release the mouse and use the editing tools to resize or reposition as desired.

 ***Note:*** To draw a polygon, click and drag the mouse across the window to the first point. Then click on the additional points of the polygon. Right click the mouse to connect the last point to the first thus closing and ending the polygon. Polygon points cannot be edited, but overall size and position can be, by using the editing tools.

 ***Note:*** Use the Grid tools to help make straight lines and regular objects.

Note: Holding the **Shift** Key down while drawing has the following effects:

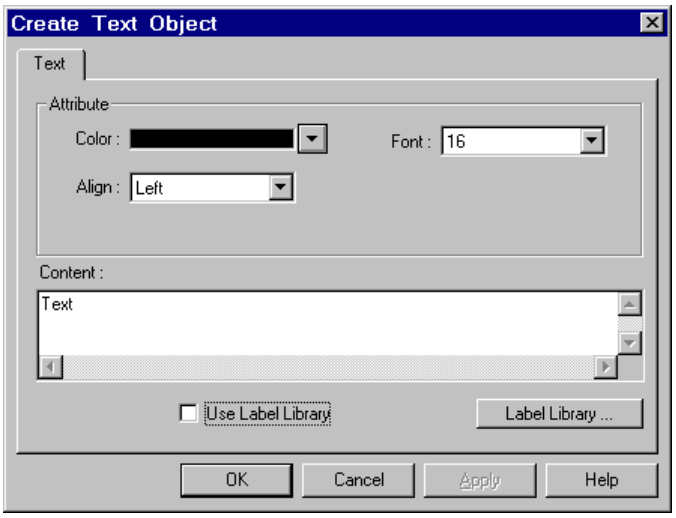
Line: Restricts to diagonals only
Rectangle: Constricts to a Square shape
Ellipse: Constricts to a Circle
Arc: Restricts to 90-degree segment
Polygon: Restricts to diagonal sides only

8.2 Text

8.2.1 Placing Text

1. Click on the Text tool  to display the Create Text Object dialog. Fill in the fields as indicated.

2. Fill in Text Tab as follows:



Color: 4 or 256 colors are available for selection, depending on model type (See above).

Font: Several font sizes are available: 8, 16, 24, 32, 48, 64 and 96.

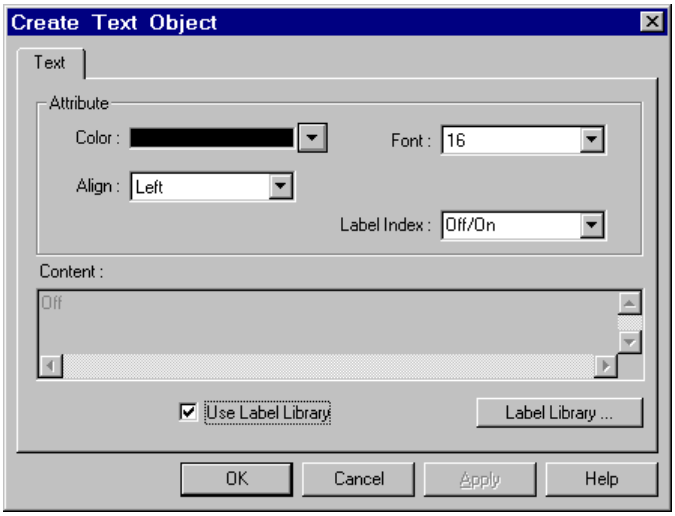
Align (Justification): When the text input for a display or Label becomes 2 lines or more, the alignment can be left, right or center justified.

Note: The Align setting does not determine where the text is placed on the window.

Content: Type in the characters to be displayed. When typing in the characters, push the “ENTER” key to move to a new line.

Optionally a Label can be used for the text.

To use a Label for the text, select the **Use Label Library** checkbox. The Contents field is grayed out and the Label Index dropdown appears.

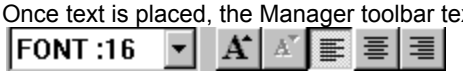


Select the desired Label from the **Label Index** dropdown.

If the dropdown does not have a suitable Label, a new Label can be added to the Library by clicking on the **Label Library …** button. Enter the Label as desired. Exit the Label Library. The new label is now available in the Label Index dropdown.

3. Click **OK** to place the text.

Note: Once text is placed, the Manager toolbar text editing tools can be used to change size and alignment.

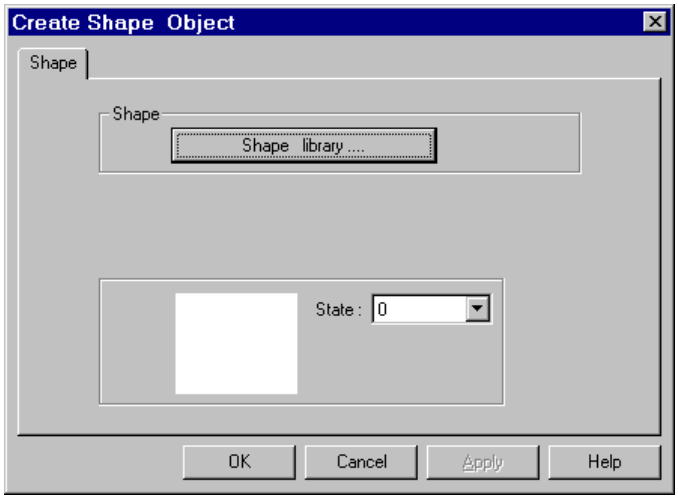


8.3 Shape

Refer to Library Operations section “Selecting a Shape”

8.3.1 Placing a Shape

1. Click on the Shape tool  to display the Create Shape Object dialog. Fill in the fields as indicated.
2. Click on the **Shape library** …. button and select a shape from a Shape library.
3. Use the **State** dropdown to view the states of the shape before placement.
4. Click OK to place the shape.



8.4 Bitmap

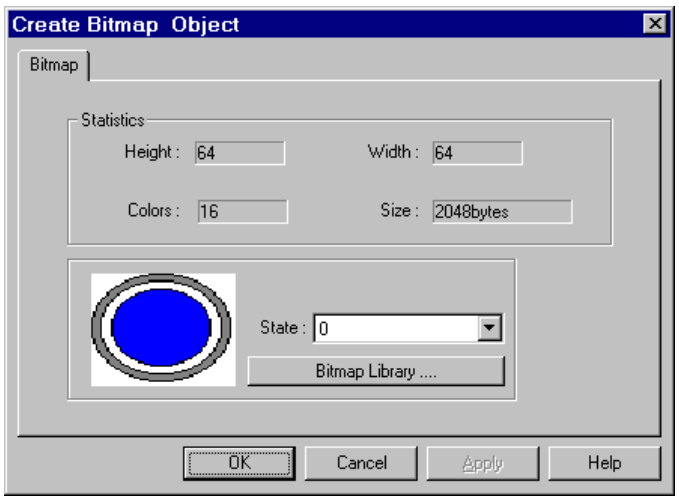
Refer to Library Operations section “Selecting a Bitmap”

8.4.1 Placing a Bitmap

1. Click on the Bitmap tool  to display the Create Bitmap dialog. Fill in the fields as indicated.
2. Click on the **Bitmap library**…. button and select a bitmap from a Bitmap library.

   Note: Once a bitmap is selected, the **Statistics** fields shows information about the bitmap. **Height** and **Width** are in pixels. **Size** is the uncompressed size of the bitmap.

   Note: The bitmap image in the display box may be distorted to fit and does not depict actual window appearance.



3. Use the **State** dropdown to view the states of the bitmap before placement.
4. Click **OK** to place the text.

8.5 Scale

The Scale is used to draw lines at regular intervals on a window. This is particularly useful with bar graph, meter or trend parts for measurement indication.

8.5.1 Drawing a Scale

1. Click Scale tool 

2. Drag on the window to the desired size. Once the drag is released, a Vertical scale is drawn.

3. Click on the Edit tool 

4. Fill in Style Tab as follows:

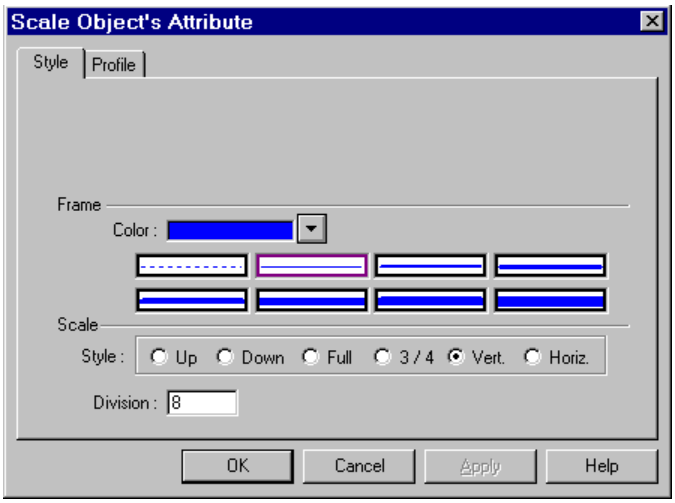
   Frame: Select the color and line style to be used for the Scale.

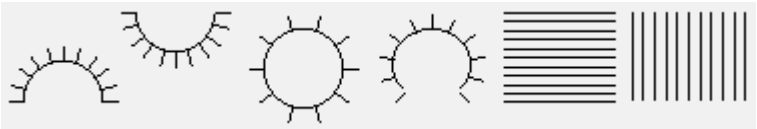
   Scale: Select the Style to be displayed: horizontal, vertical or curved (Up, Down, Full, ¾). Curved scales are used in conjunction with the Meter part.

   Division: Select the number of Divisions on the scale (Range is 1 to 255).

5. Fill in **Profile** tab to numerically adjust size if necessary.

6. Click **OK** and adjust position if necessary.





Up Down Full ¾ Vert Horiz.

9.0 Editing Placed Objects

All of the toolbar items depicted below are also available from the Edit menu. If tool bars are not visible, use the View Menu to call them up.

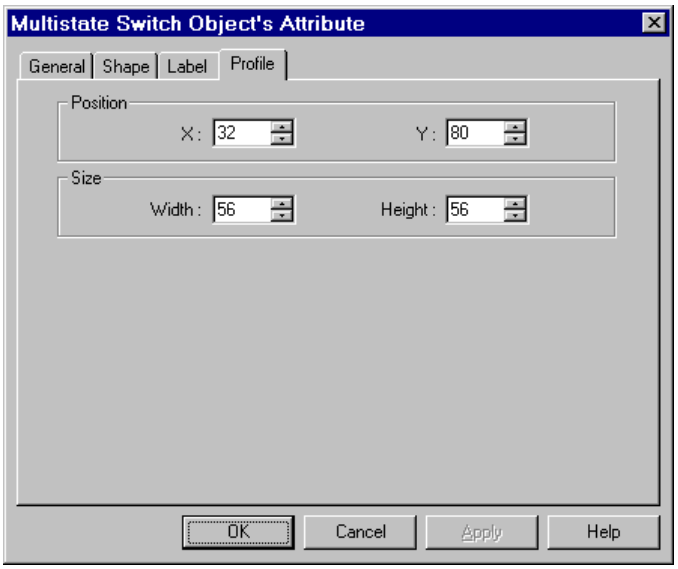
Standard Windows editing functions **Undo,** **Redo,** **Cut,** **Copy,** **Paste** and **Delete** are available for all objects.

9.1 Moving and Resizing Objects

To move and resize drawing objects: Use the pointer to select the object by clicking on it. Drag a rectangle around a group of objects to select them. Optionally, hold the **Ctrl** key down and click one at a time on the objects to move. The object or group's handles appear. The object or group can now be dragged around the window. A single object can be resized by dragging its handle.

9.1.1 The Profile Tab

Once an object is placed on the window, its position can be set to a numerical position through the profile tab of the Attributes Dialog for that part (object). Click on the Edit icon and adjust the Profile Tab settings as desired.



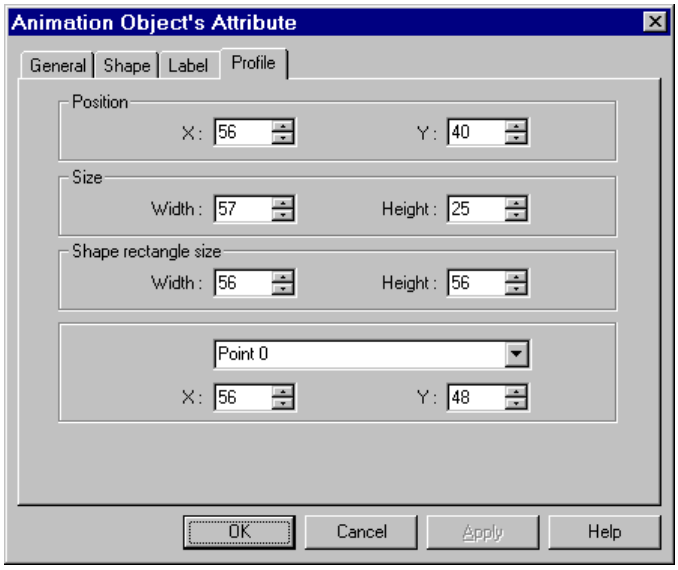
Position: X and **Y** are the location of the upper left corner of the object.

Size: Width and **Height** are the overall size of the object in pixels.

Note: The following parts do not have Profile tabs:

- **Numeric Input Extend**
- **Numeric Data**
- **ASCII Input Extend**
- **ASCII Data**
- **Alarm Display**
- **Alarm Bar**
- **Recipe Display**
- **Event Display**

The Animation Parts have additional fields as described below.



Shape rectangle size:

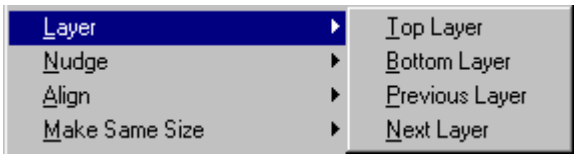
Width and **Height** are the overall size of the Shape or Bitmap assigned to the part in pixels.

Below this are fields for setting the **X** and **Y** location each **Point** along the animation path. (See Animation in the Object Reference Guide for more details.)

9.1.2 Object Order

Object order allows objects to be “layered” one on top of the other. An object may be brought to the front, back, front one layer at a time or back one layer at a time. Object order can be selected from the **Edit** menu **Layer** submenu or by clicking on the appropriate Toolbar icons.

Command	Tool	Description
Top Layer		Brings the object to the top layer (in front of all other objects).
Bottom Layer		Sends the object to the bottom layer (behind all other objects).
Previous Layer		Brings the object one layer to the forefront.
Next Layer		Sends the object one layer towards the background.

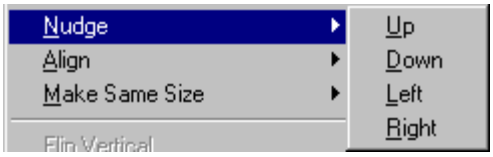


Note: Each object placed on a window resides on it’s own layer. The number of layers on a window is limited by the number of objects allowed on a window (500 maximum).

9.1.3 Nudge

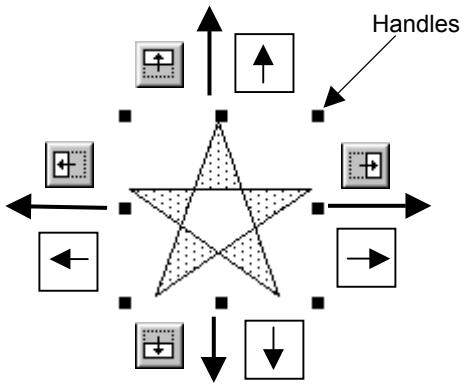
Press the cursor keys on the keyboard ( UP,  DN,  LT, and  RT), to move an object or group one pixel at a time. Object position moves can be called from the **Edit** menu **Nudge** submenu or by clicking on the appropriate Toolbar icons.

Command	Tool	Description
Up		Moves the object one pixel towards the top edge of the window.
Down		Moves the object one pixel towards the bottom edge of the window.
Left		Moves the object one pixel towards the left side of the window.
Right		Moves the object one pixel towards the right side of the window.



Note: The **Grid** features do not hinder Nudge operations.

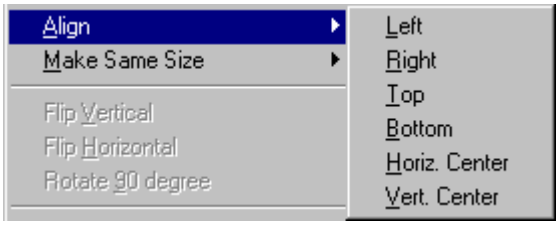
Note: When the  **Fix objects** feature is active, the nudge commands do not move an object.



9.1.4 Aligning Objects

The alignment tools are used to line up multiple objects that have been selected by dragging the cursor around them. Objects can be aligned by their left edge, vertical center, right edge, top edge, horizontal center, or bottom edge. The selected object that is in the top most layer is held in position while the other objects are aligned. Object alignment can be called from the **Edit** menu **Align** submenu or by clicking on the appropriate Toolbar icons.

Command	Tool	Description
Left		Aligns the selected objects to the left side of the object in the topmost layer.
Right		Aligns the selected objects to the right side of the object in the topmost layer.
Top		Aligns the selected objects to the top edge of the object in the topmost layer.
Bottom		Aligns the selected objects to the bottom edge of the object in the topmost layer.
Horiz. Center		Aligns the selected objects horizontally to the center of the object in the topmost layer.
Vert. Center		Aligns the selected objects vertically to the center of the object in the topmost layer.



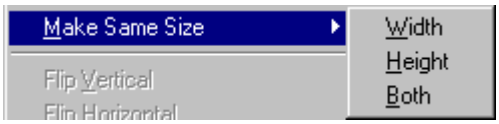
9.1.5 Resizing Objects

9.1.5.1 Resizing objects is accomplished by clicking on the object to show its "**handles**". Handles are the small rectangles that appear around the object to show it is highlighted. The handles can be dragged to change the size of the object.

9.1.5.2 The **Profile Tab** in the objects Attribute Dialog (as shown above) can also be used to change the size of an object.

9.1.5.3 When multiple objects are selected they can all be constrained to the same size by using the "Make same size" tools. Objects can be made the same **width**, **height**, or **overall size**. Object resizing can be called from the **Edit** menu **Make Same Size** submenu or by clicking on the appropriate Toolbar icons.

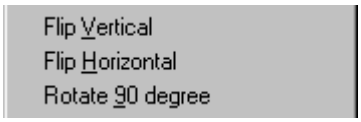
Command	Tool	Description
Width		Makes the selected objects to the same width as the object in the topmost layer.
Height		Makes the selected objects to the same height as the object in the topmost layer.
Both		Makes the selected objects to the same width and height as the object in the topmost layer.



9.1.6 Transformation Tools

Single object shapes can be flipped horizontally, vertically and rotated using the transformation tools. Object transformation can be called from the **Edit** menu **Flip Vertical, Flip Horizontal or Rotate 90 degree** items or by clicking on the appropriate Toolbar icons.

Command	Tool	Description
Flip Vertical		Flips the selected object from top to bottom.
Flip Horizontal		Flips the selected object from left to right.
Rotate 90 Degrees		Rotates the selected object counterclockwise 90 degrees.



Note: Groups, Bitmaps and Parts cannot use these functions.

9.2 Grouping Objects

When multiple objects are selected by dragging the cursor around them, they may be grouped by clicking on the group tool. If an object is not fully encompassed by the drag, it is not selected. When a group is selected, it may be broken up into its separate objects by clicking the ungroup tool. Object grouping can be called from the **Edit** menu **Group or Ungroup** items or by clicking on the appropriate Toolbar icons.

Command	Tool	Description
Group		Groups the selected objects.
Ungroup		Breaks apart the selected group into objects.

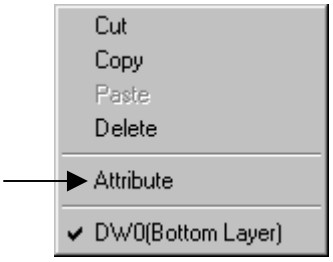
Group Operations:

Once objects are grouped, they are moved as one object.
Clicking on any object of the group selects the entire group.
A Group selected by dragging around it can be saved to Group Library.
If the group only consists of drawn objects, it can be saved to the Shape Library.

9.3 Editing Object Attributes

Object attributes can be edited by clicking on the object to show its black “handles”. There are different ways to access objects attributes.

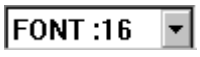
1. Click on the  Edit icon to call up its Attributes Dialog.
2. Double click on the object to call up its Attributes Dialog.
3. From the **Edit** menu, select the **Change Attribute…** command.
4. Alternately, the parts attributes can be edited by right clicking on the part and selecting **Attributes** from the pop up shortcut menu.
5. Double-click on an object ID number in the Window Treebar Object tab.



Edit the different settings as desired.

9.3.1 Text Editing tools:

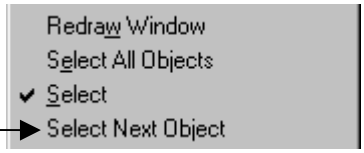
Once a Text object is placed, the toolbar text editing tools can be used to change some of the characteristics.
Note: There are no corresponding menu commands for these tools.
Note: These tools can also be used on highlighted Part Labels as well.

Tool	Description
	This tool is used to change font size. Simply select the size you want from the dropdown.
	These tools are used to increment and decrement the font size by one size. ***Note:*** You cannot use these tools to reduce the font size to 8.
	These tools are used for changing the text alignment for multiple lines of text. Options are Left, Center and Right justify commands.

9.3.2 Editing Stacked Objects

Sometimes several objects are placed on top of one another (overlapping or stacked). Selecting one of them for editing can be a problem. To get to an object in a stack, click to highlight the top object.

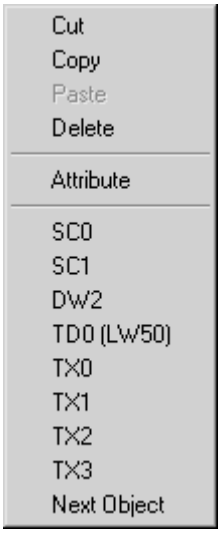
If this is not the desired object, do one of the following to get to the object:



1. From the **Edit** menu, select the **Select Next Object** command.

2. Click on the  Select tool to “tunnel” to the object that needs to be edited.

3. Right click on the object to bring up the Shortcut Menu. The bottom of the Shortcut Menu displays the **Object ID** numbers. Select the number of the object to edit. Then, right click again and select the **Attribute** menu item.



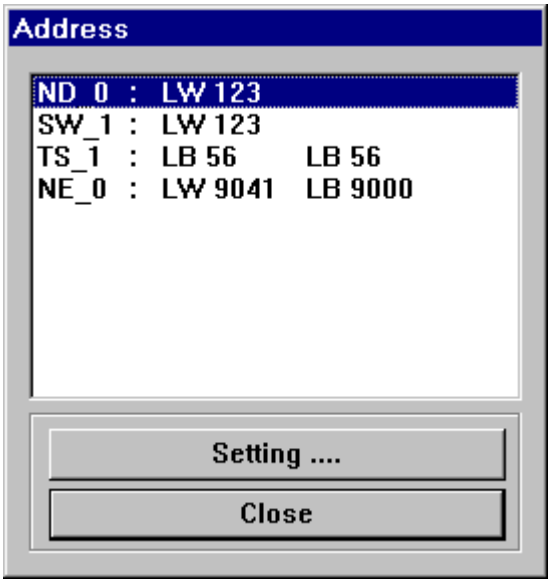
9.3.3 Editing Grouped Objects Attributes

To edit the attributes of objects that have been grouped click on the group or select multiple objects by dragging a rectangle around them.

Note: The handles for object groups are displayed with the opposite color of regular handles.

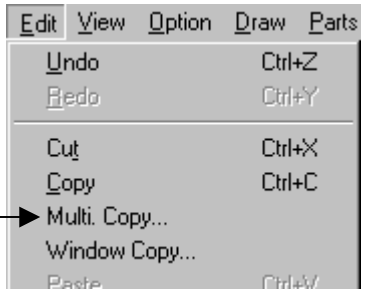
Click on the  Attribute tool. This action calls up the Address dialog that allows you to select the object to edit. Highlight the object and click on **Setting ….**

Note: Editing in this fashion only allows editing of the object’s “General” tab attributes. Drawing objects are not shown in the list and cannot be edited in this fashion.



9.4 Multi. Copy Command

The Multi. Copy command is used to duplicate a part or object multiple times. As a Part is duplicated, its Device address is incremented. This gives a convenient way to create banks of indicators and switches.

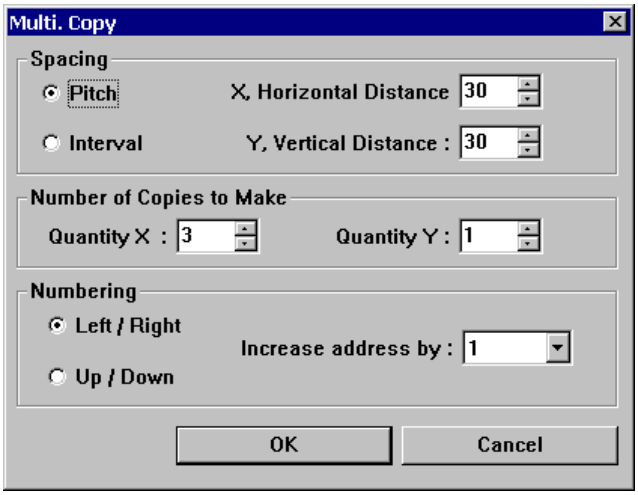


Using the Multi. Copy Command:

1. Select the Part or object to copy. (It is recommended that you find out the object's size first.)

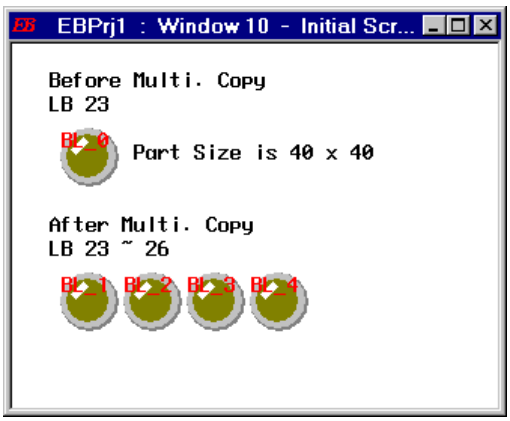
2. Select Multi. Copy from the Edit Menu.

3. Fill in the fields as described below.



Item	Description
Pitch	The space from Upper left corners of the copies are X and Y distance apart in pixels
Interval	The space between copied objects is the X and Y distance. (Right side to left side, bottom side to top side)
X, Horizontal Distance, Y, Vertical Distance	The distance between or from one copied object to the next. ***Note:*** Make these distances greater than the object's size to leave space between copies.
Quantity X, Quantity Y	The number of copies in the **X** (across) and **Y** (up and down) directions. ***Note:*** The copy command truncates if copies extend past the edge of the window. Part Device addressing is also incorrect.
Left / Right	The copied objects are duplicated from left to right one **row** at a time. Object numbering is Left to right.
Up / Down	The copied objects are duplicated from left to right one **column** at a time. Object numbering is top to bottom.
Increase address by	This value determines how much the object's Device Address is incremented for each copy.
OK	Accepts the settings and performs the Multi. Copy operation
Cancel	Cancels the Multi Copy operation and closes the **Multi. Copy** dialog

For example: If a Bit Lamp part with Device address 23 is being copied 4 times and the **Increase address by** distance is 1, after the Multi. Copy command is executed, there are 4 Bit Lamps on the display with consecutive Device addresses 23, 24, 25 and 26. If the **Increase address by** is set to 2 then the addresses would be 23, 25, 27 and 29.



Note: The **Increase address by** setting has no effect when copying graphic objects.

4. Click **OK** to perform the copy.

Note: The **Undo** command undoes a Multi. Copy action.

9.5 Finding and Replacing Device Addresses with the Find/Replace Addr… Utility

The **Find/Replace Addr…** utility is used find instance where a device address is used in project windows. This allows a user to modify a project easily when an address change happens in the controller logic program or the HMI developer wants to reconfigure an address for some other reason. This Utiltiy searches project windows and returns information about where a specific device type or address is used.

Using the **Find/Replace Addr…** Command:

1. Select **Edit|Find/Replace Addr…** from the menu or click the  tool.
2. Select to search for a **Bit Address** or a **Word Address**.
3. Fill in the **Find What** fields as described below:

   Window No.: This field holds the specific window for searching or All windows can be specified.

   Device address: The dropdown is the bit/word prefix and in the field following it the bit/word address.
4. Fill in the **Replace With Device address**. The dropdown is the bit/word prefix and in the field following it the bit/word address.
5. Click the **Find** button to see a list of instances where the **Find What** Device address is used.
6. Select the **Match whole address only** checkbox to narrow the search results if desired.

Sample Window Showing Find/Replace

7. Highlight a result and click the **Replace** button to change the objects Device address to the **Replace With** Device address.
8. Click **Replace All** to change all the found objects Device addresses to the **Replace With** Device address.

When finished select **Exit** to close the dialog.

Note: All changes are saved as they are made.

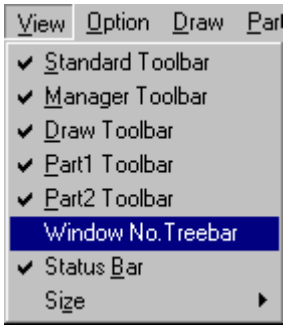
Note: The undo command cannot be used to recover changes made by the Find/Replace process.

Note: The **Find/Replace** utility does not search the Alarm Scan, PLC Control, Event Log and Data Transfer Parts. These Parts must be edited separately.

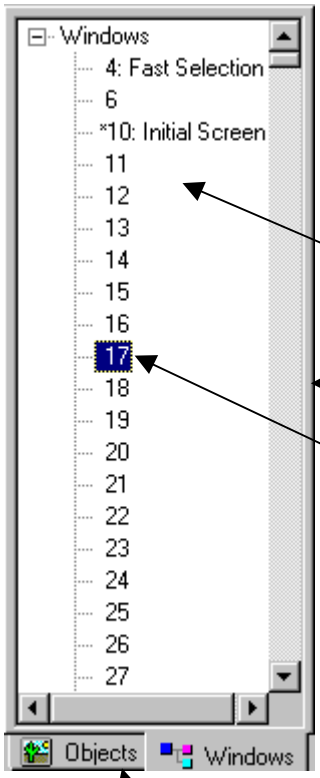
Note: The **Find/Replace** utility does not search **Macro** scripts. Scripts must be edited separately.

9.6 Using the Window No. Treebar

The Window Treebar is used during project design to call up and view Windows and Objects. To activate the Treebar select **View** menu **Window No. Treebar** item or simply press the **Tab key**.



Note: The Tab key is used to toggle the Treebar On and Off.



Initially, the Treebar appears to the left of the design window. Like the other tool bars, it can be dragged to other locations around the window or left to "float".

9.6.1 Treebar Operations

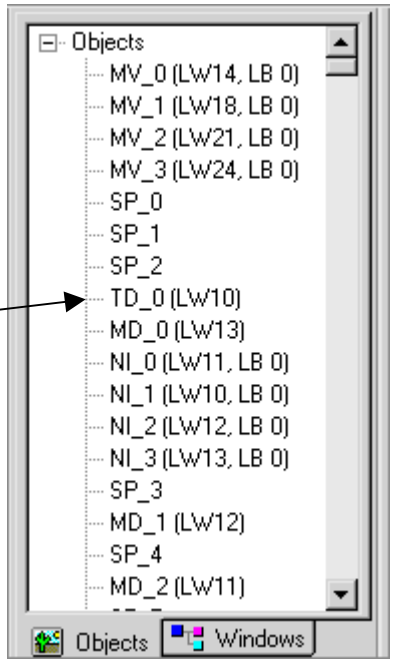
Click on any Window number to quickly bring it to the forefront in the design window.

Double Click on any closed Window number to open the Window for design.

Click and drag the Right Frame to resize the width of the Treebar.

Once a window number is highlighted, right-click in the Treebar window to popup a shortcut menu. This menu gives a quick way to create a new window, open an existing window, close an open window, delete a window from the project or get to a window's settings.

Click on the Objects Tab to view all of the objects on the Window number being displayed.



Click on an object ID number in the Treebar to highlight the object in the design window. The object can then be positioned or edited as desired.

Double-click on an object ID number to call up the attributes for that object.

10.0 System Bit and Register Reference

10.1 Local memory

Local memory resides in DRAM and every value is initialized to zero at system start up.
The HMI is equipped with following local memory ranges:

LB 0000 ~ 9999 (Bit devices) Segmented in groups of 16 bits as follows:

LB 00 ~ 15
LB 16 ~ 31
LB 32 ~ 47
… … …
LB 9968 ~ 9983
LB 9984 ~ 9999

LW 0000 ~ 9999 (Word devices)

LW0000
LW0001
LW0002
… … …
LW9998
LW9999

Each LBnnnn is a 1-bit device and each LWnnnn is a 16-bit device. These devices use separate memory areas and do not overlap (i.e. LB0000 is not the first bit of LW0000. Changing LB0000 does not affect LW0000).

Note: The Local Bit and Word addresses above 8999 are reserved for system use.

10.2 Remote memory

When using Master - Slave hardware configurations, the Slave MMIs can access the Master's Local memory.

Master local memory ranges:
Ms_LB 0000 ~ 9999 (Bit devices) and **Ms_LW** 0000 ~ 9999 (Word devices).

10.3 Reserved Local Words/Bits

Some Local Words, Local Bits and Recipe Words are reserved for special purposes. Users should not use these areas except for their specified purposes.

Local Bits: 9000~9999 are reserved
Local Words: 9000~9999 are reserved

10.3.1 Reserved Local Bits

LB Address	Description	NOTE	Version
9000~9009	Initialized as **ON**	Use these bits for objects that need an initial setting of ON. (read/write)	ver 1.2
9010	**Recipe** download indicator, it is: Set ON when downloading Set OFF when download done	Use this bit to indicate when a recipe download is in progress (read/write).	ver 1.2
9011	**Recipe** upload indicator, it is: Set ON when uploading Set OFF when upload done	Use this bit to indicate when a recipe upload is in progress. (read/write)	ver 1.2
9012	**Recipe** download/upload indicator, it is: Set ON when transferring data Set OFF when transfer done	Use this bit to indicate when any recipe transfer is in progress. (read/write)	ver 1.2
9013	Task bar **Touch Indicator** pressed bit, it is: Set ON when **Touch Indicator** is pressed	This bit does not return the state of the "touch indicator". (read/write)	Ver 1.4

LB Address	Description	NOTE	Version
9014	Task bar **CPU Indicator** pressed bit, it is: Set ON when **CPU Indicator** is pressed	This bit does not return the state of the ”CPU indicator”. (read/write)	Ver 1.4
9015	Task bar **Alarm Indicator** pressed bit, it is: Set ON when **Alarm Indicator** is pressed	This bit does not return the state of the ”alarm indicator”. (read/write)	Ver 1.4
9016	**Print Error** indicator: Changes to 1 when printing fails	Use to trigger an alarm or event to let the user know there is a problem with printing. (read only)	Ver 1.4
9017	**Printer** enable bit. The user: Sets ON to disable print functions. Sets OFF to enable print functions.	Setting in System Parameters must have a printer selected for this Bit to have an effect. (read/write)	Ver 1.4
9020	**Pen** enable bit. The user: Sets ON to enable pen functions.	Positive edge trigger. Message board use (read/write)	Ver 1.4
9021	**Brush** (Eraser) enable bit. The user: Sets ON to enable brush functions.	Positive edge trigger. Message board use (read/write)	Ver 1.4
9022	**Clipping** enable bit. The user: Sets ON to enable clip functions.	Positive edge trigger. Message board use (read/write)	Ver 1.4
9030	**Pen** width to 1 pixel enable bit. The user: Sets ON to set pen width to 1 pixel.	Positive edge trigger. Message board use (read/write)	Ver 1.4
9031	**Pen** width to 2 pixels enable bit. The user: Sets ON to set pen width to 2 pixels.	Positive edge trigger. Message board use (read/write)	Ver 1.4
9032	**Pen** width to 3 pixels enable bit. The user: Sets ON to set pen width to 3 pixels.	Positive edge trigger. Message board use (read/write)	Ver 1.4
9040	**Fast Selection** window enable bit. The user: Sets ON to hide Fast Selection window. Sets OFF to show (pop-up) Fast Selection window.	This bit overrides the System Parameter Task Bar setting. (read/write)	Ver 1.4
9041	**Task Bar** enable bit. The user: Sets ON to hide the Task Bar. Sets OFF to show (pop-up) the Task Bar.	Task Bar control (read/write)	Ver 1.4
9042	**Task Buttons** enable bit. The user: Sets ON to hide the two Task Buttons. Sets OFF to show (pop-up) the two Task Buttons.	Task Bar control (read/write)	Ver 1.4
9043	Hide/Show Task Items (**Fast Selection** screen, **Task Bar** and **Task Buttons**) The user: Sets ON to hide the Task Items. Sets OFF to show (pop-up) the Task Items.	When enabled, all items appear in their activated state. Task Bar and Fast Selection window are opened. (read/write)	Ver 1.4
9044	Enable changes made to System Parameters in Retentive memory area. The user: Sets ON to make the Security **Passwords**, **Backlight** and **Buzzer** System Parameters active. OFF has no effect.	Forcing this bit ON restores Security Passwords, Backlight and Buzzer system parameters from Reserved Retentive word area. After restoration, the system sets this bit OFF. (read/write)	Ver 2.1
9045	**Reset** MMI. The user: Sets ON to reset the HMI.	Forcing this bit ON resets the MMI. (write)	Ver 2.1
9046	**Security** level change event indicator.	Changes to 1 when going from a lower security level to a higher security level. (read only)	Ver 2.1

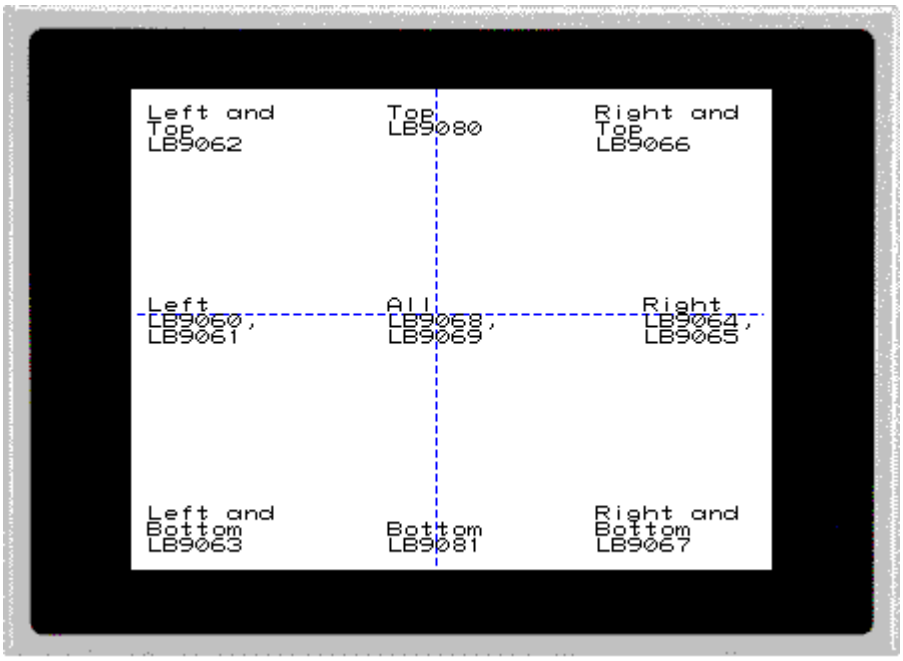
LB Address	Description	NOTE	Version
9050	Toshiba T/C write enable bit. The user: Sets ON to enable writing to T/C bits ON & OFF. Sets OFF to disable T/C writing. **T1/T1S user's manual NOTE**: When writing to Timer/Counter registers, the Timer/Counter's device data should be added to the written data. (2 bytes each, corresponding to the Timer/Counter's register)	When setting the Timer/Counter's device to ON, use '01'; for OFF use '00'. (read/write)	Ver 1.5
9051	Enable or Disable the touchscreen when the **Backlight** is turned OFF. The user: Sets ON to disable touchscreen when **Backlight** is turned OFF. Sets OFF to keep touchscreen enabled when **Backlight** is turned OFF.	OFF is the default mode. (read/write)	Ver 1.6
9052	Write back enable **PLC Control, Change window.** The user: Sets ON to disable write back to PLC. Sets OFF to enable write back to PLC.	This applies to Change Window control only. (read/write)	Ver 1.6
9055	**Disconnect action** – **PLC** communications. The user: Sets ON to continuously retry any write to PLC command. Sets OFF to stop any write to PLC commands.	When HMI is disconnected from the PLC, it acts according to the state of this local bit 9055. (read/write)	Ver 1.6
9056	**Disconnect action** –**Touchscreen** The user: Sets ON to enable touchscreen. Sets OFF to disable touchscreen.	When HMI is disconnected from the PLC, it acts according to the content of this local bit 9056. (read/write)	Ver 1.6

Bits LB9060 ~ 9069, 6080 and 6081 are used for detecting Numeric Input Extend and ASCII Input Extend presses. See the following page for Touchscreen map.

9070	**MITSUBISHI J2-S10** driver control Effects local data being added to 'R' or subtracted from 'R'. Set bit ON to disable add/subtract operations Set bit OFF to enable add/subtract operations Valid range: 0x00800000 to 0x7FFFFFFF (8388608 to 2147483647 decimal) Invalid range: 0xFF799999 to 0x80000000 (-8388608 to –2147483648 decimal).	Due to the protocol limitation of J2-S100 for 'R' registers. A value between 0x00800000 and 0xFFFFFFFF(inclusive) cannot be written to R because it creates an error. (read/write)	Ver 1.6
9071	**MITSUBISHI J2-S10** driver control Write options for 'R'. The user: Sets bit ON to write to EPROM. Sets bit OFF to write to RAM.(default)	(read/write)	Ver 1.6
9090	**Event Log Clear** The user: Sets bit ON to write to clear entries from the Event Log. This includes entries stored in RW memory, if enabled.	After the Event log is cleared, this bit auto resets to OFF. (read/write)	Ver 2.5.2
9091	**Adjusts contrast** lighter one shade. The user: Sets bit ON to activate.	After the contrast is made lighter, this bit auto resets to OFF. Hardware Version 4.5 only.(read/write)	Ver 2.6.0
9092	**Adjusts contrast** darker one shade. The user: Sets bit ON to activate.	After the contrast is made darker, this bit auto resets to OFF. Hardware Version 4.5 only.(read/write)	Ver 2.6.0

LB Address	Description	NOTE	Version
9100~9227	**PLC address/node communication** status.	These bits correspond to the PLC Station Numbers 0~127. The corresponding bit changes to 0 when communication times out. Write 1 to resume communications. (read/write)	Ver 2.6.0
9228~9355	**AUX address/node communication** status	These bits correspond to the AUX. Station Numbers 0~127. The corresponding bit changes to 0 when communication times out. Write 1 to resume communications. (read/write)	Ver 2.6.0

9060	**Keypad control** bit, left side (of window) Bit forced ON whenever a user actives an input data object (NI or AI). Bit forced OFF when entering valid data or ESC key is pressed.	User can use this bit to control a Direct window Keypad popup. Keypad window is closed if input succeeds. (read only)	Ver 1.6
9061	Keypad control bit, left side	See note for 9060. (read only)	Ver 1.6
9062	Keypad control bit, left and top side	See note for 9060. (read only)	Ver 1.6
9063	Keypad control bit, left and bottom side	See note for 9060. (read only)	Ver 1.6
9064	Keypad control bit, right side	See note for 9060. (read only)	Ver 1.6
9065	Keypad control bit, right side	See note for 9060. (read only)	Ver 1.6
9066	Keypad control bit, right and top side	See note for 9060. (read only)	Ver 1.6
9067	Keypad control bit, right and bottom side	See note for 9060. (read only)	Ver 1.6
9068	Keypad control bit, all side	See note for 9060. (read only)	Ver 1.6
9069	Keypad control bit, all side	See note for 9060. (read only)	Ver 1.6
9080	Keypad control bit, top side	See note for 9060. (read only)	Ver 2.0
9081	Keypad control bit, bottom side	See note for 9060. (read only)	Ver 2.0



10.3.2 Reserved Local Words

LW Address	Description	NOTE	Version
9000	Retentive memory **Index** base	RBI and RWI use this as an index offset when accessing retentive data. (read/write)	Ver 1.2
9002-9003	Set to **Numeric Input Maximum** value when numeric input gets the focus.	Numeric Input loads its maximum value when activated. When Numeric Input loses the focus, it is set to zero. (read only)	Ver 1.4
9004-9005	Set to **Numeric Input Minimum** value when numeric input gets the focus.	Numeric Input loads its minimum value when activated. When Numeric Input loses the focus, it is set to zero. (read only)	Ver 1.4
9006	**Message board mode** 0: pen 1: brush 2: clipping	Message board use (read)	Ver 1.4
9007	**Pen width** 0:1 pixel 1:2 pixel 2:3 pixel	Message board use (read)	Ver 1.4
9008	**Pen color** 0-255	Message board use (read/write)	Ver 1.4
9010	**Local second**	BCD code, valid values: 0 ~ 59 (read/write allow)	Ver 1.2
9011	**Local minute**	BCD code, valid values: 0 ~ 59 (read/write allow)	Ver 1.2
9012	**Local hour**	BCD code, valid values: 0 ~ 23 (read/write allow)	Ver 1.2
9013	**Local day**	BCD code, valid values: 0 ~ 31 (read/write allow)	Ver 1.2
9014	**Local month**	BCD code, valid values: 0 ~ 11 (read/write allow)	Ver 1.2
9015	**Local year**	BCD code, valid values: 0 ~ 9999 (read/write allow)	Ver 1.2
9016	**Local day of the week**	BCD code, valid values: 0 ~ 6 (read/write allow)	Ver 1.2
9020	**Object queue status** This holds the total number of objects of all the windows on the display. (Each window can hold up to 500 objects.)	If a screens object queue exceeds 1000, then the HMI screen is too complex. The HMI is in danger of reporting a Severe System Error due to low system resources.	Ver 1.4
9034-9035	**System time** (unit as 0.1 second)	Starts at 0 when project started. (read/write)	Ver 1.4
9040-9041	Window **Security password** This word must contain the password for access to secure windows.	Double word (write only)	Ver 1.6
9042	**Security level** The current active security level	(read only)	Ver 1.6
9043	**Force security level** Set to 0 (Lowest) or 1 (Middle).	A security level can only be forced to a lower level than is active. (write only)	Ver 1.6

LW Address	Description	NOTE	Version
9044	**Touch process mode** There are three operational modes to handle momentary switches. They are based on the current value of LW9044. Use a SET WORD of "Set on window open" in the Common window to configure the desired operation mode. **0:** Window popup operation is enabled during touch down and up, at touch up the BIT previously set ON is set OFF, even if a popup window hides the momentary switch. (default initial value) **1**: Window popup operation is disabled during touch down and up. **2:** Window popup operation is enabled during touch down and up. Therefore, if the momentary switch is hidden by a popup window, on touch up, the BIT is not reset.	This modification is to resolve a software constraint in older versions. When a momentary switch is pressed, (touch down) the related bit is set ON. If there is any popup window that hides the momentary switch before it is released, the bit remains ON even if released (touch up). (read/write)	Ver 1.6
9050	**Base Window Id**	Slave HMI can use this word to show the same screen as the Master.	Ver 1.2
9051	Reserved for **Base Window ID** write back operations by a slave MMI. (See LW9050)	PLC control/Change screen writes back to 9051, so it is reserved for that purpose.	Ver 1.2
9054	**Report printout option**, print out: **0**: Text, Meters and Trends **1**: Text, Meters, Trends and Shapes but not patterns **2**: Text, Meters, Trends and Bitmaps **3**: Text, Meters, Trends, Bitmaps and Shapes but not patterns **4**: All	Use this to change the attributes of a PLC Control/ Report printout assignment. (read/write)	Ver 1.5
9055	**PLC Control word Offset** There are two options in the PLC Control object that use this local word: **Change window**, and **Report printout**. The value in this word is added to the controlling data as an offset. Before writing back, the HMI subtracts this value from the data.	Use this to offset window numbers coming from the PLC. Example: A PLC Control / Change window uses D10. If (LW9055) = 10, and D10 = 4 then the HMI changes to window 14. After changing, the HMI writes back 4 to D11. (read/write)	Ver 1.6
9057	**EventLog** DataBase Item size Management information, the size of every item	Use when storing the Event Log in retentive memory. (read only)	Ver 2.1
9058-9059	**EventLog** DataBase size Management information, the size of the DataBase, the size includes management information. (total_item * item_size) + (management_info_size)	Use when storing the Event Log in retentive memory. (read only)	Ver 2.1
9060-9075	Holds **Numeric and ASCII Input**, input data. 9075 holds the least significant digits.	(read/write)	Ver 1.4

LW Address	Description	NOTE	Version
9080-9085	**Project name**	Use ASCII Data to show project name It occupies 12 bytes. (read only)	Ver 1.5
9086-9087	**Project size** in bytes	Use Numeric Data to show (In Decimal) (read only)	Ver 1.5
9088-9089	**Project size** in K bytes	Use Numeric Data to show (In Decimal) (read only)	Ver 1.5
9090-9091	**Compiler version ID**	Use Numeric Data to show (In Decimal) (read only)	Ver 1.5
9092	Project **Compile Date/Year**	Use Numeric Data to show (In Decimal) (read only)	Ver 1.5
9093	Project **Compile Date/Month**	Use Numeric Data to show (In Decimal) (read only)	Ver 1.5
9094	Project **Compile Date/Day**	Use Numeric Data to show (In Decimal) (read only)	Ver 1.5
9100	**Indirect Addressing,** For external PLC only	9100 indirect window number	Ver 1.6
9101	**Indirect Addressing,** For external PLC only	9101 indirect offset	Ver 1.6
9130	**Language control word**	Use this to change the displayed language state for all labels. The range is 0~3. (read/write)	Ver 2.5
9135	**Li-Battery Voltage in millivolts** **Note:** This is not the PLC's battery. It is the touchscreen's internal battery.	Only displayed after download. Not available in Simulation modes. Hardware Version 4.5 only. The range is 0~1228 which is scaled from 0~3V. If the LW9135 < 1126 (2.75V), change the Li-battery. (read only)	Ver 2.6.0

10.4 Retentive memory

The HMI units have 64K of battery backed RAM. This memory is accessed by using the following data types:

RB - accesses the first 2047 registers as 16 bit groups. The bit is designated as a hexadecimal number. (i.e. accessing bit 10 of word 63 would be Device type RB, Device address 63A)

RBI - accesses the first 2047 register's bits and sets the index pointer to that bit. LW9000 is used in conjunction with the RBI value to give an offset value for Recipe Transfer parts. (i.e. If RBI is set to device address 20 and LW9000 has a value of 5 in it, then bit downloads and saves would begin at RB25 = RBI20 + 5.)

RW - accesses the retentive registers as words. The full range (0 to 65535) is available but registers above 60000 are reserved for system use (See below).

RWI - Reads the register and sets the index pointer to the value in that register. LW9000 is used in conjunction with the RWI value to give an offset value for Recipe Transfer parts. See Recipe Transfer Part for an example of the RWI register is used. The valid range of RWI types is 0~ 32767.

Ms_RB and **Ms_RW** are used by Slave configured units for accessing retentive memory locations in a remote Master unit.

NOTE: All retentive data types overlap in retentive memory. (i.e. changing RB0002 to ON changes the value of RW00001. This also affects RWI00001 and RBI0002.)

Some Retentive Words are reserved for special purposes. Users should not use these areas except for their specified purposes. Retentive Words: 60000~65535 are reserved

10.4.1 Reserved Retentive Word

RW Address	Description	NOTE	Version
60000	**Real Time Clock second**	BCD code, valid values: 0 - 59 (read/write allow)	Ver 1.2
60001	**Real Time Clock minute**	BCD code, valid values: 0-59 (read/write allow)	Ver 1.2
60002	**Real Time Clock hour**	BCD code, valid values: 0-23 (read/write allow)	Ver 1.2
60003	**Real Time Clock day**	BCD code, valid values: 0-31 (read/write allow)	Ver 1.2
60004	**Real Time Clock month**	BCD code, valid values: 0-11 (read/write allow)	Ver 1.2
60005	**Real Time Clock year**	BCD code, valid values: 0-9999 (read/write allow)	Ver 1.2
60006	**Real Time Clock day of the week**	BCD code, valid values: 0-6 (read/write allow)	Ver 1.2
NOTE about RTC: When using "Objects" to display and change system time, the user must take care to enter only valid values. For example: Seconds cannot be changed to 78(BCD), if 78 (BCD) is entered, the RTC continues counting 78 79 80 ... etc. This causes unpredictable conditions to happen.			

10.4.2 System Information

System Parameter mapping to recipe card information. When a project is downloaded to a unit and run for the first time, System Parameters are stored in the System Reserved Memory area. The following is a list of the mapping relationship.
Note: Grayed out selections are not available in 2.5.x and 2.6.x versions of EasyBuilder.

Parameters from the PLC Tab

RW	System Parameter	NOTE	Version
60040	**Serial port**	0:232 1:485	Ver 2.0
60041	**Baud rate**	0:9600 1:19200 2:38400 3:57600 4:115200	Ver 2.0
60042	**Data bits**	0:7 1:8	Ver 2.0
60043	**Parity**	0:none 1:even 2:odd	Ver 2.0
60044	**Stop bits**	0:1 1:2	Ver 2.0
60045	**HMI station**		Ver 2.0
60046	**PLC station**		Ver 2.0
60047	**Multiple MMI**	0:none 1:master 2:slave	Ver 2.0
60048	**MMI-HMI link speed**	0:38400 1:115200	Ver 2.0
60049	**PLC time out constant**		Ver 2.0
60050	**PLC block pack**		Ver 2.0

Parameters from the General Tab

RW	System Parameter	NOTE	Version
60051	**Task button: Attribute**	0:Disable 1:Enable	Ver 2.0
60052	**Task button: Position**	0:Left 1:Right	Ver 2.0
60053	**Task button: background color**		Ver 2.0
60054	**Task button: Text**	0:Left adjust 1:Center	Ver 2.0
60055	**Alarm bar: Pixel per scroll**	8, 16, 24, 32	Ver 2.0
60056	**Alarm bar: Scroll speed**	units in tenths of a second, 0 to 25.5	Ver 2.0
60057	**No. of windows**	2, 3, 4, 5, 6	Ver 2.0
60060	**Startup window No.**	10~1999	Ver 2.0
60061	**Back light saver**	0 (Disable) 1~255 second (Enable)	Ver 2.1
60062	**Cursor color**		Ver 2.1
60064	**Buzzer**	0:None 1:Yes	Ver 2.1
60065	**Common window: popup window**	0:Normal 1:Above any others	Ver 2.1
60066	**Common window: Attribute**	0:Below base window 1:Above base window	Ver 2.1
60067	**Extra. No. of Event**	200~3000	Ver 2.1
60068	**RTC source**	0:Local Word 1:Internal RTC	Ver 2.1
60069	**Printer: Printer**	0:None 1:EPSON ESC/P2 2:SP printer 3.HP PCL/Simple Page Mode	Ver 2.1
60070	**Message board window No.**	0 (Disable) 10~1999 (Enable)	Ver 2.1

Parameters from the Security Tab

RW	System Parameter	NOTE	Version
60071	**Security Control:**	0:None 1:Yes	Ver 2.1
60072	**Password: level 0**	two words	Ver 2.1
60074	**Password: level 1**	two words	Ver 2.`
60076	**Password: level 2**	two words	Ver 2.`

11.0 EasyBuilder Operations

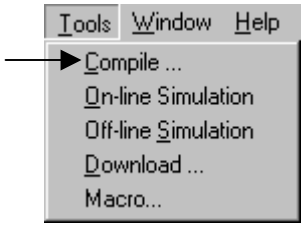
11.1 Project Operations

11.1.1 Compiling a Project

Before a project can be Simulated or downloaded it must be compiled. Compiling is the process used to convert the graphic screen information and object information into a data the units processor can understand. During this process the code is optimized and reduced. This is why large *.epj project files can fit into the HMI memory. The result is the small *.eob object file. It is this file that is downloaded to the unit or accessed for simulation.

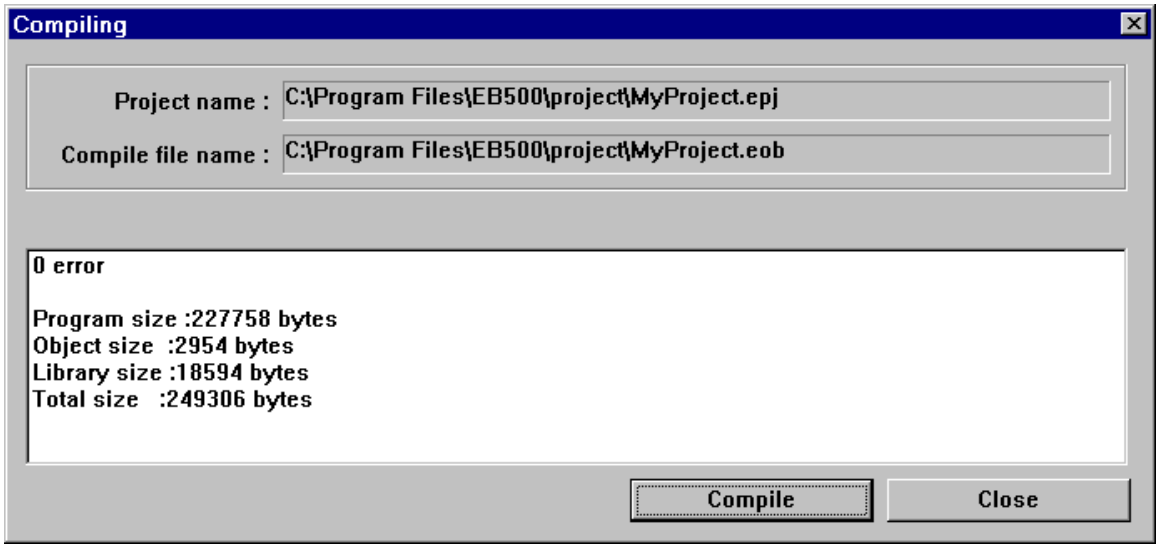
> ***Note:*** If the **Automatic save and compile at download and simulate** checkbox in the **Option**|**Window Property …** menu settings is checked, a project is automatically saved and compiled before simulation or download.


A project must be saved before compiling.



To compile a project:

1. Select the  Compile tool or select **Compile …** from the **Tools** menu.



The Project ***.epj** name and resultant compiled ***.eob** file names are shown in the **Compiling** dialog. Make sure this is the correct project to compile.

2. Click on the **Compile** button.

> ***Note:*** As the project is compiled, data or error messages are displayed in the text field. Once the compile is finished, the project file sizes are listed. If the **Total size** value is greater than the available HMI memory, the project cannot be downloaded properly. This is true even if it can be simulated without problems.

3. Click on the **Close** button to exit the Compiling dialog.

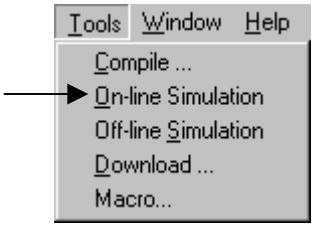
11.1.2 Simulating a Project

A compiled project can be simulated using EasyBuilder's On-line and Off-line Simulation tools.

11.1.2.1 On-line simulation

Before using the **On-line** simulator, the HMI must be connected to the PC and the PLC. Both the HMI and the PLC must be powered.

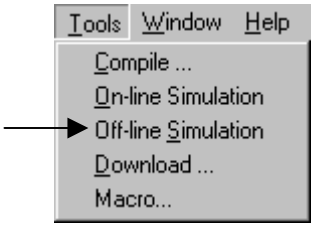
1. Select the  On Line tool or select **On-line Simulation** from the **Tools** menu.
2. When the simulation appears on the PC screen, use the PC's pointing device in the same manner as the touchscreen.
3. Right Click and select **Exit** from the pop-up menu to exit the simulation.



11.1.2.2 Off-line Simulation

Off-line simulation does not require the HMI to be connected to the PC. Off-line simulation is achieved through the PC acting as the PLC. True simulated operations may not be achieved by this method. Use Off-line simulation to see screen's appearance in actual operation.

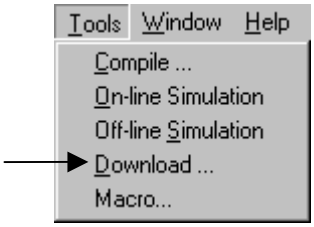
1. Select the  Off-Line tool or select **Off-line Simulation** from the **Tools** menu.
2. When the simulation appears on the PC screen, use the PC's pointing device in the same manner as the touchscreen.
3. Right Click and select **Exit** from the pop-up menu to exit the simulation.

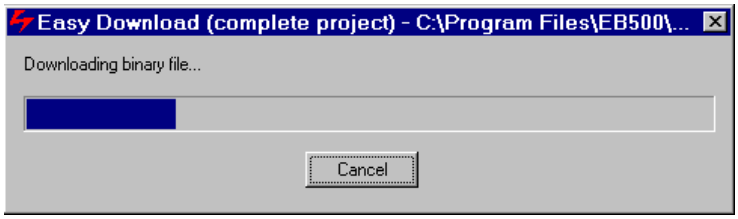


11.1.3 Downloading a Project

Before downloading a project, the HMI must be connected to the PC. The HMI must be powered. Downloading a Project from EasyBuilder sends the project being edited to the MMI.

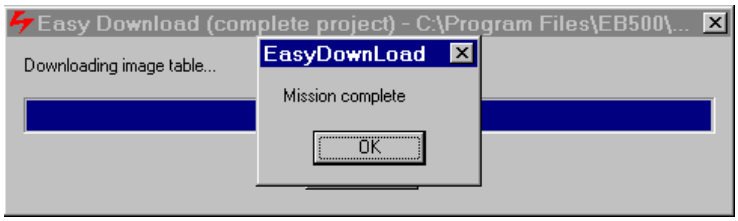
Select the  Download tool or select **Download …** from the **Tools** menu. The Downloading begins with the Easy Download dialog popping up. The HMI is then forced into the RDS mode. The HMI project memory is erased. Then the download begins. The download is tracked by the progress bar.





Pressing the **Cancel** button aborts the download. The HMI is left in the RDS mode with no project in memory.

Note: There is no Undo operation to recover the previous project.



When the download is complete, the Mission complete message is displayed.

Click **OK** to close the Easy Download dialogs.

Note: If the **Jump to application mode when download done** checkbox in the **Option**|**Window Property …** menu settings is checked, the HMI Jumps to Application automatically after download. Otherwise, force the HMI to the application mode after a download by going to **EasyManager** and **pressing Jump to Application** or Cycling power to the HMI.

11.1.4 System Error Messages

A running project (simulation or after download) displays error messages when various fault conditions occur. Below is a list of these messages.

Message	Description
Attach Fail Suggest Reset PLC	Cannot setup communication with PLC, cycle power to the PLC to clear its buffer and retry.
Attach PLC	Cannot detect the PLC. Check the cable connections and PLC port settings. Initiate communications with the PLC.
Base Window<##> Must Full Screen Size	Attempting to open a Base window number <##> that is less than full size.
BCD Transfer Overflow	Occurs when attempting to enter an illegal data value into a BCD type register (such as entering 10000 to a 16-bit BCD register).
Invalid Window Number <##>	A change screen command issued by either Function Key or PLC Control is attempting to switch to undefined screen: <##> (screen number).
MilliSecond Timer Full	Out of system resources – check factory
Not Support Operation	Attempting to do something that is not allowed for the driver.
Object Queue Full	Out of system resources – check factory
PLC Data Block Queue Full	The PLC data points on a screen cannot be divided into less than 100 read blocks. (100 is the limit of read blocks)
PLC no response *	HMI lost communication with PLC, incorrect cable configuration to PLC or incorrect System Parameter settings for PLC.
PLC responsed always delayed.	PLC reply timing was off. Try to adjust response time in the PLC if this is possible; if not, report problem to ORMEC.
PLC response error*	PLC replied with an unexpected response message, caused by a read command or write command. Check if the configured PLC device address is available.
Printer fail!	The printer is not ready or the printer buffer is full.
Printer pool overflow	The printer buffer is full. Check that the printer is on line and responding.
Project image error, compiler again	There is an error in the compiled project file (*.eob). Check the project compile level and recompile the project.
Second Timer Full	Out of system resources – check factory
System Severe Error*	System has run out of memory – check factory
System severe error (error code 10002)	System cannot uncompress a graphic because it is larger than 250KB. Do not compress bitmaps larger than 250KB. Change System Parameter/Editor Tab/Object In Compress to “No”.
System severe error (error code 10013)	Attempt to use an external driver with Hardware that is not enabled to accept it. Contact the factory for assistance.
System severe error (error code 10101)	System Parameter/General Tab/Startup window is an undefined window. Change the Startup window number to a valid window.
Text Index Out of Range: Font Style <##>	The text string contains illegal values that access characters out of the font table range. <##> denotes the illegal character code.
This PLC Only Support Read <##> Words of Word of BIT Data	A part is attempting to read more bit data from the PLC than the driver can handle. Reduce the number of bits on the screen or adjust the Block Pack.
This PLC Only Support Read <##> Words of Word Data	A part is attempting to read more data from the PLC than the driver can handle. Reduce the number of words assigned to the part or adjust the Block Pack.

This PLC Only Support Write <##> Words of Word Data	A part is attempting to write more data to the PLC than the driver can handle. Reduce the number of words assigned to that part or adjust the Block Pack.
Touch Queue Full	System has run out of memory – check factory
Width and Height Must Be same with Window 4	Attempting to change the Fast Select Window, via Function Key, with a window that is not the same size as Window 4.
White Board Support Popup Style Only	The Message Board window can only be called up as a Direct, Indirect or Popup window.
Window <##> was opened As a common Popup Window	Attempting to open a full screen Base window number <##> as a Popup window.

*Message can be customized by “System Message Part”.

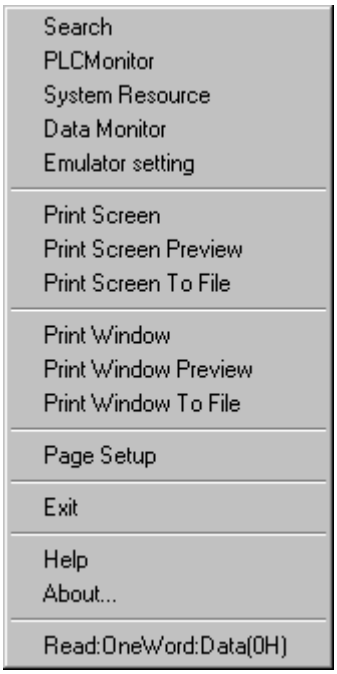
11.2 Debugging with EasyWindow

EasyBuilder projects can be debugged with the EasyWindow utility. EasyWindow is the utility that is used to run projects in Simulation mode. There are two simulation screen formats; a graphic simulation that looks like the bezel of an HMI unit with the simulation window, or a Window frame with menu and the simulation window.

For the Graphic Simulation, right click to bring up the **EasyWindow** menu.
For the Windows Frame Simulation, selections can be made right from the menu bar.

Use the **Emulator setting** menu item to change the simulation appearance.

Note: Menu items below the Emulator setting are used for project documentation. (See section on Project Management and Documentation.)



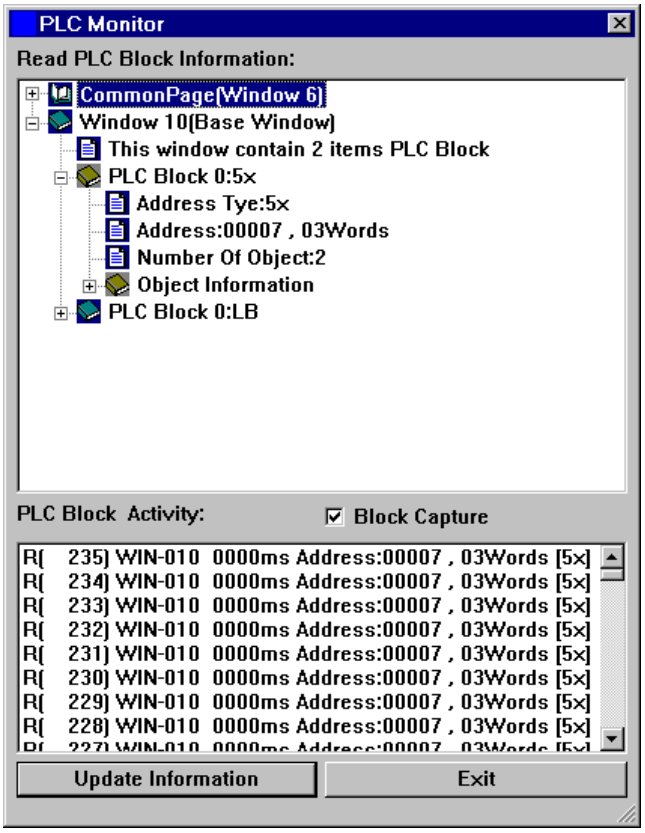
11.2.1 PLC Monitor

The PLC Monitor allows tracking of Communications and PLC Block settings.

Check the Block Capture box to view communications to the PLC. If the PLC no response message is appearing on the simulation, viewing the Block Capture reveals which communication has failed. Check that the value is valid for the PLC.

The **PLC Block Activity** is mapped as follows:

R or **W** is Read or Write operation.
(#####) is the transaction number of the communication.
WIN-### is the window number that is triggering the communication.
####ms is the time in milliseconds for the transaction to take place.
Address:##### is the Device Address accessed.
##Words is the number of words accessed in this block.
[$$$$$] is the Device Type accessed.



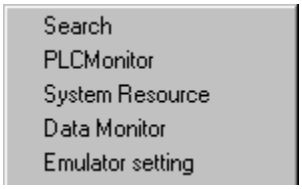
The **Read PLC Block Information** is an expandable tree that gives the layout of active project windows in the simulation as PLC Blocks. Information on a window is fetched from the PLC in Blocks.

The tree can be expanded for each window to show the number of Blocks in each window. Each **PLC Block** can be expanded to show the Block starting Address and number of Words as well as the number of Objects that use the Block information.

All of this information can be used for editing the project to optimize communications by:

- Reducing the number of Blocks required by a window
- Maximizing the number of objects that use the Block information.

11.2.2 Data Monitor



The Data Monitor tool is used to monitor data transactions for specified parts. The transaction list is updated whenever data changes are triggered. Several data Monitors can open to monitor specific part types.

The **Title bar** of the Data Monitor dialog displays the monitor number. This is useful when more than one Data Monitor is open.

Option:

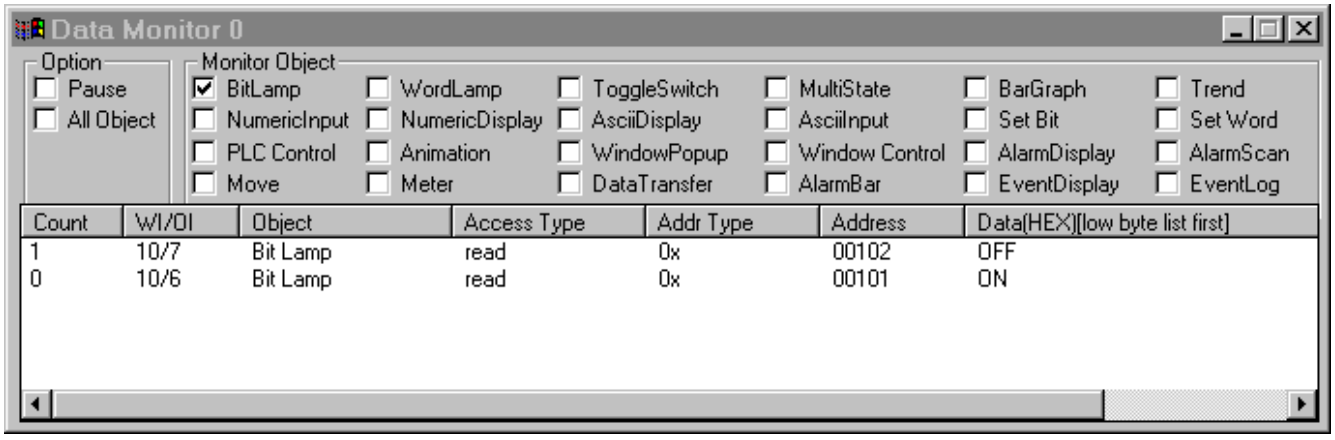
Pause Checkbox: Enables/Disables transaction updates. Use this when transactions are scrolling too fast to stop them for a snapshot of transaction history.

All Object Checkbox: Enables/Disables activating the transaction history for all objects on the active windows.

Monitor Object: Select the parts to monitor. If there are no parts of the type selected, the transaction list does not update that part type; other types on active windows are updated.

Transaction list fields

Column	Description
Count	The transaction number. The most recent transactions are on the top of the Transaction list.
WI/OI	Window number and Object ID number. Use this to tell what window the transaction is from and what Part is triggering the transaction.
Object	The Part Type that triggered the transaction.
Access Type	Indicates if the transaction was a Read or Write transaction.
Addr Type	Device type from the Part attribute.
Address	Device address from the part attribute.
Data	The state data from the device. Bits are shown as OFF or ON, words are shown as hexadecimal values.

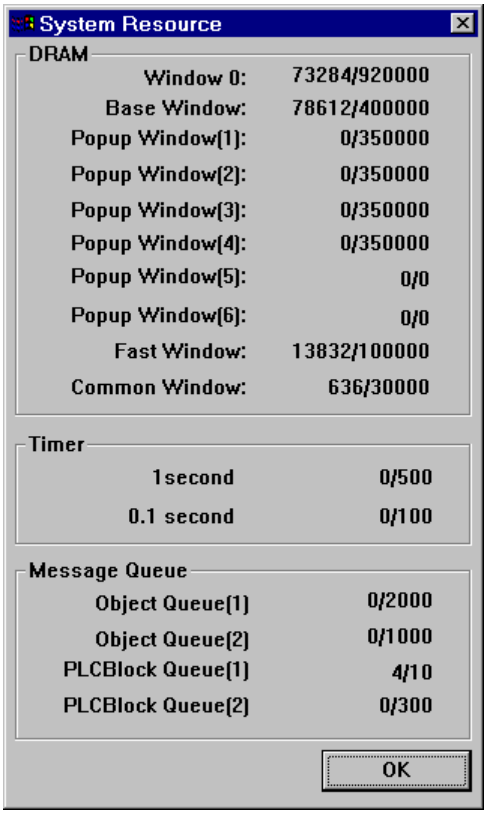


In the above example, the most current transaction shows Bit Lamp 7, on window 10 (Address 1x00102), going to the OFF State.

11.2.3 System Resource

The System Resource dialog shows the status of the HMI resources. This allows you to see the impact of your project on available memory. This is helpful in determining the cause of System errors. After each item, the amount of memory used is shown, a ”/” and the amount of resources reserved for that item.

Item	Description
Window 0	Memory used / Memory available This is the internal memory that manages Trend Displays, Event Displays, and some internal data structures. The total size of Window 0 depends on the number of resources allocated to other windows. This includes the **"No. of windows"** (setting in System Parameters) and Fast Selection window. To calculate the total memory of Window 0, start with 320kB, subtract 100kB if the Fast Selection window is enabled, add back 350kB for each window less than the full 6-window limit ((6 - #) * 350kB). In this example: (Fast Selection and 4 windows) Window 0 = 320kB - 100kB + (6-4) * 350k = 920kB
Base Window	Memory used / The amount of memory reserved for Base Windows.
Popup Window 1~6	The amount of memory used / reserved for each Popup Window. In this example, **No. of windows** is set to 4, so, Popup Windows 5 and 6 are not used so their total size is zero.
Fast Window	The amount of memory used / reserved for the Fast Selection Window.
Common Window	The amount of memory used / reserved for the Common Window.
Timer(1 second)	Number of timers / Number of reserved Timers. The number of timers depends on the number of active Trend graphs.
Timer(0.1 second)	Number of timers / Number of reserved Timers. The number of timers depends on the number of active objects with Periodical functions.
Object Queue (1) **Object Queue (2)**	Number of queued messages / Number of reserved queues.
PLC Block Queue(1) **PLC Block Queue(2)**	Number of queued block reads / Number of reserved queues.



Note: If any item exceeds the reserved limitations, a System Severe Error happens.

11.2.4 Search

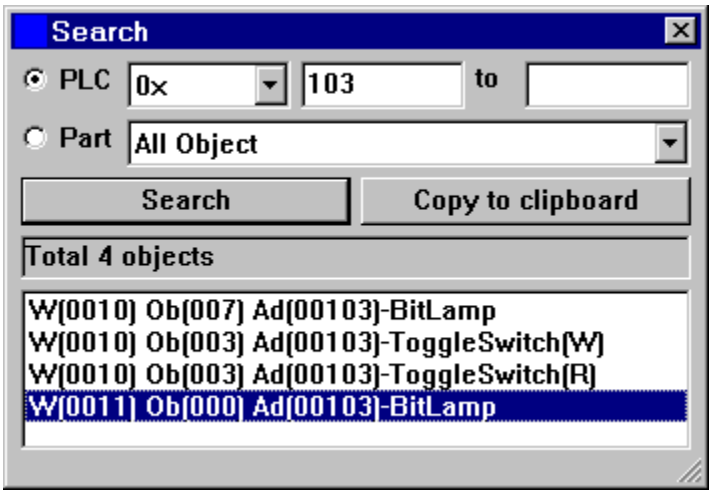
Search is used to find all instances where a PLC reference or Part type is used. This is used for documentation purposes or to eliminate repetitious Parts. The Search dialog provides two search methods.

11.2.4.1 **PLC**

This searches for a PLC address or Part type. Two search formats are provided.

A: Search from one address to another inclusive (fill in both fields, the left one less than the right).

B: Search one address only (fill in the left field only).



11.2.4.1 **Part**

This searches for a Part type in the opened project. The search result shows a list of all Part's locations of the selected type. Select Part type from the dropdown list.

Click the **Search** button to activate the search.

After the results of the search are displayed, click on a search result to jump to the window where the result is located. The selected results are highlighted with a flashing border.

Use the **Copy to clipboard** to paste the information into a text document for printing, records or other documentation function.

12.0 Project Management and Documenting a Project

The standard Windows file functions are available for EasyBuilder projects. EasyBuilder projects are saved with the *.EPJ extension. Projects that are compiled or Uploaded are saved as files with *.EOB extensions.

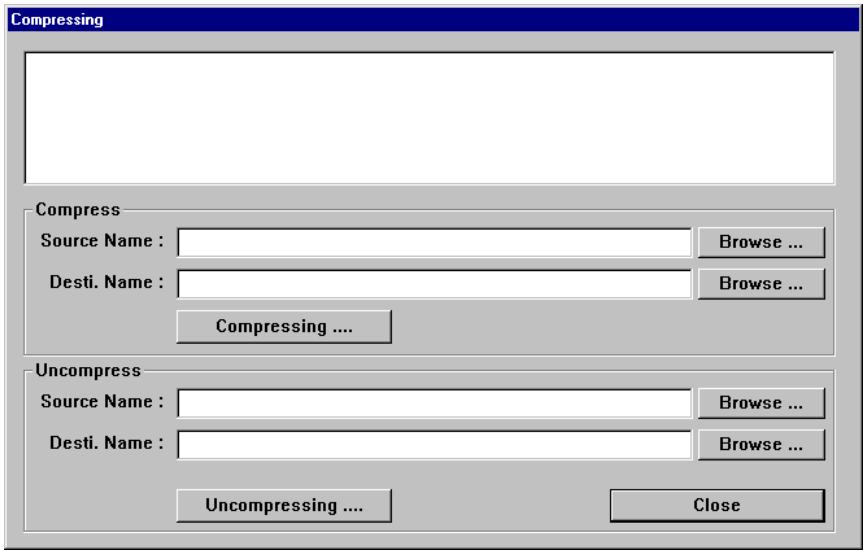
A Quick Summary of File Menu Operations:

New – Starts a new project
Open… – Opens an existing project
Close – Closes a project. (A reduced menu is displayed. See note below.)
Save – Saves a project to disk.
Save As… – Saves a project with the option of specifying a new storage location or project name.

Note: Reduced Menu when no Projects are Open
File Menu: Shows limited file functions. (**New, Open, Recent files, Exit**)
View Menu: Shows limited toolbar viewing. (**Standard toolbar, Status bar**)
Tools Menu: **Compress…** and **Decompile**
Help Menu: Access to the **Help Topics…** and **About EasyBuilder…** commands.
Note: About EasyBuilder should show version **2.6.2** for compatibility with this manual.

12.1 Compressing/Uncompressing a project

Project compression is an easy way to archive a project with its libraries. Select the **Tool|Compress …** menu item to compress a project along with all of the libraries used in that project into a single file. The extension used for compressed files is ***.CMP**. Uncompressing can be used to extract the project and its libraries later. If the original libraries or project data is lost, they can be recovered by uncompressing the archived project.



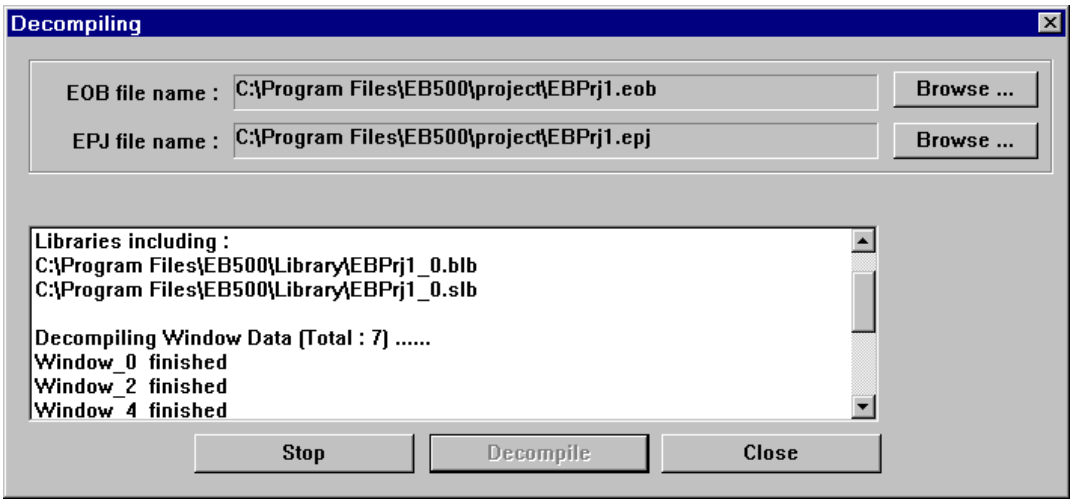
Enter the project's name in the **Source Name** field or click **Browse …** to search for the project to compress. The Destination name is automatically assigned. If you would like a different name for the compressed project, enter it in the **Desti. Name** field.

Click on the **Compressing ….** button to begin the compression process.

Note: The same procedure is used in the **Uncompressing ….** section for uncompressing a file.

12.2 Decompiling a project

The **Decompile …** tool is used to create a project ***.epj** file from an ***.eob** file. The ***.eob** file may be from a compiled project or from an upload. Use this tool to Decompile an *.eob file. All the windows and their objects are converted back into a project file. Because window and object names are not retained during the compile process, they are assigned default names in the derived project. Window Shapes and Bitmaps are saved to Library folders with the project file name as a prefix along with their associated library extension (slb, blb). Groups from the group libraries are saved as individual components (drawing elements, shapes, bitmaps and objects) and are not recreated.



> ***Note:*** Projects created with any version of EasyBuilder before version 2.0.2 cannot be decompiled. Projects uploaded from units with BootROM versions prior to V1.6, ID 0x100a cannot be decompiled. Upgrading the BootROM in an older unit does not make an upload compatible.

It is recommended that you retain all project information in an archive for future changes. See the section on compressing projects above.

Enter the **EOB file name** or click the **Browse…** button to select the *.eob file for decompile. The *.eob file can be from an existing project or upload. Browse for the project to decompile. Once an *.eob file is selected, a default name is inserted into the **EPJ file name** field. If this is not acceptable, press the **Browse…** button and select an appropriate file name for the decompiled project.

Press the **Decompile** button to begin the decompile process. As the project is decompiled, the text box displays progress and statistics about the project.

> ***Notes:*** New libraries containing all of the Shapes and Bitmaps used in the project are created with the project name as the prefix and a number as the suffix. Groups are decompiled as ungrouped objects.
> Windows do not retain their Names. Parts do not retain their descriptions.

Press the **Stop** button to stop the Decompile at any time.

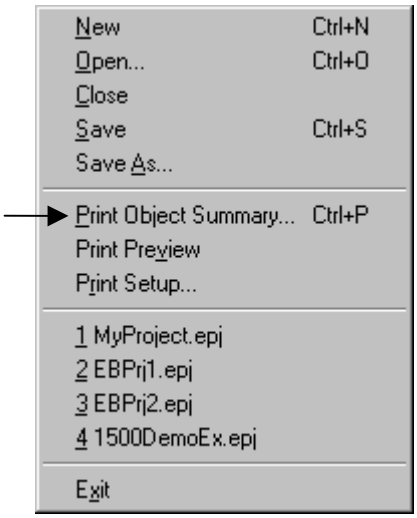
Use the **Close** button to exit the Decompiling dialog.

12.3 Documenting a project

Project Documentation is done through the File menu, Print Object Summary command or EasyWindow Utility.

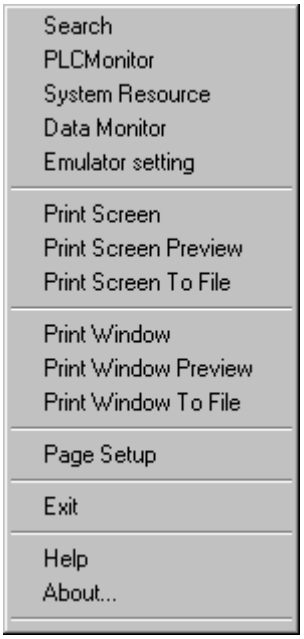
12.3.1 Print Object Summary

Use the **Print Object Summary** to send almost all project information as a printout directly to a printer or file. Every Part on every window is printed as a verbal description. For even medium sized projects, this generates hundreds of pages of documentation. It is recommended that the Summary be viewed first with the **Print Preview** command. Use the **Print Setup…** to set the printer's attributes.



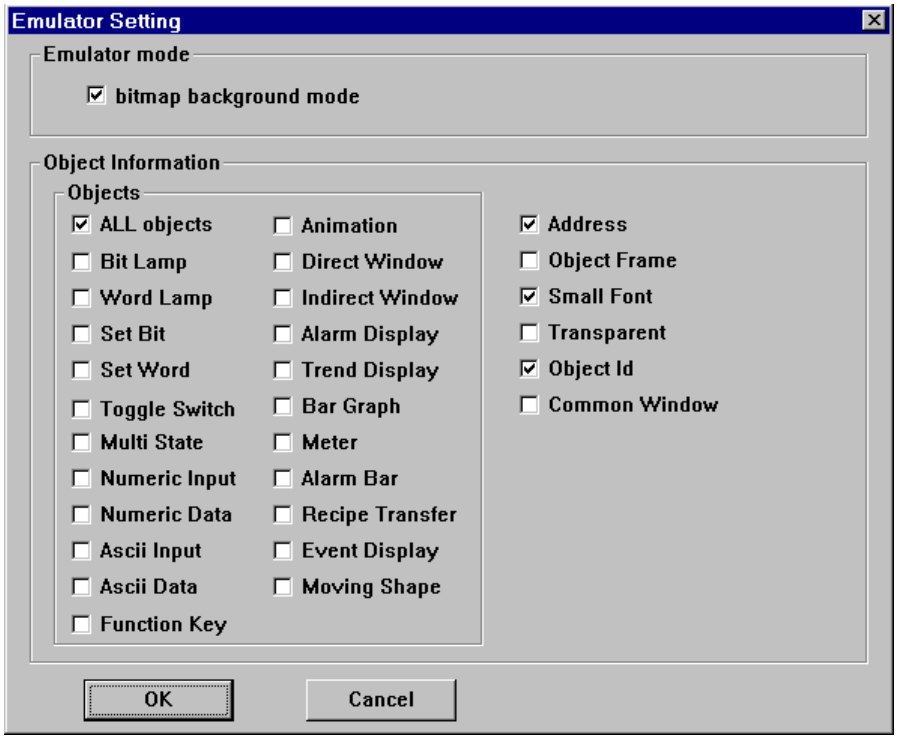
12.3.2 EasyWindow Documentation

Run the project with Offline Simulator to bring up the EasyWindow display. Right click on the display to popup the menu (in Graphic mode, else, if Frame mode, it is displayed at the top of the window).



12.3.2.1 Printing Screens with EasyWindow

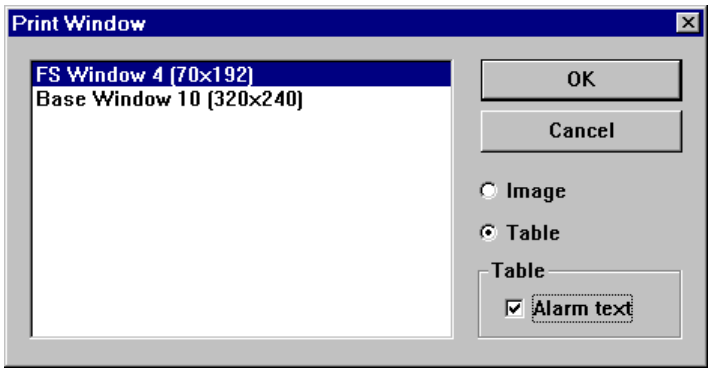
Select **Emulator setting** from the menu. Check the **ALL objects** check box. Then, click on the check boxes on the right side of the dialog to have various objects information displayed in the simulation. Click **OK**.



Next select **Print Screen, Print Screen Preview** or **Print Screen To File** to get a bitmap graphic output of the displayed screen.

12.3.2.2 Printing Object Assignments with EasyWindow

Select **Print Window, Print Window Preview** or **Print Window to File** from the menu. Select the **Image** radio button to print the screen. Select the **Table** radio button to print a table of objects on the screen. Check the **Alarm text** box if you would like active alarms printed with the Table. Click **OK**.



12.4 Using Compact Flash to transfer a project

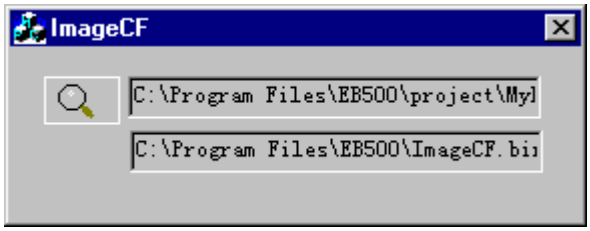
Projects can be transferred to HMI units with the "-ETH" option. These units have a compact flash slot that accepts CompactFlash™ memory cards. Projects are stored to the CompactFlash™ memory cards, transported to the unit and inserted, then transferred to the units flash memory. This is convenient for sending project updates to remote locations and customer sites. It also allows the project to be transferred to the unit without the need of a portable PC or other external equipment.

12.4.1 Transferring a project to CompactFlash™ memory

1. Compile the project as usual. This generates a project file with the EOB extension.

2. Run the **ImageCF.exe** utility.

3. Use the **ImageCF.exe** utility to select the EOB file to convert to CompactFlash™ compatible format. This is done by clicking on the magnifying glass button in the ImageCF dialog. By default, a file named ImageCF.bin is created in the folder where the **ImageCF.exe** utility is located.



4. Copy the ImageCF.bin to the CompactFlash™ memory card. A CompactFlash™ interface attached to your PC is needed to do this. CompactFlash™ interfaces are available in most stores that sell computer hardware.
   Note: The CompactFlash™ memory card may be formatted as FAT16 or FAT32, do not give the CompactFlash™ memory card a volume label.

12.4.2 Transferring a project from CompactFlash™ to an MMI

1. Remove the CompactFlash™ card from the CompactFlash™ interface and transport it to the site where the HMI is located.

2. Power down the HMI unit. Put dipswitch 2 of the unit in the up (on) position. Insert the CompactFlash™ card into the units CompactFlash™ slot.

3. Apply power to the MMI. The unit comes up in RDS mode. If the CompactFlash™ card is detected the screen displays a button with the letters "CF" on it in the lower right corner. Press this button to transfer the project from the CompactFlash™ memory card to the unit's internal flash memory.

4. Once the screen stops scrolling text and indicates that the transfer is complete, remove power from the MMI. Put dipswitch 2 back to the down (off) position, remove the CompactFlash™ memory card and apply power to the unit. The transferred project is now the running application.

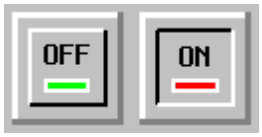
Section 3: Object Reference Guide

13.0 Objects Summary

Icon	Name	Function	ID number
	Bit Lamp	Displays ON or OFF shape to reflect current bit status in the PLC.	BL-nnn
	Word Lamp	Displays a different shape to reflect current register data in the PLC.	WL-nnn
	Set Bit	Forces a set or reset of a PLC bit address data (coil).	SB-nnn
	Set Word	Write specified data to a PLC word address (register).	SW-nnn
	Toggle Switch	Combination of bit lamp and set bit parts.	TS-nnn
	Multi-State Switch	Combination of word lamp and set word parts.	MS-nnn
	Function Key	Create a touch area to change window, input data, pop up or minimize a window.	FK-nnn
	Numeric Input	Display PLC registers data in numeric form and allow keypad input for changes.	NI-nnn
	Numeric Data	Display the PLC register data in numeric form only.	ND-nnn
	ASCII Input	Display the PLC register data in ASCII and accept keypad input.	AI-nnn
	ASCII Data	Display the PLC register data as ASCII characters.	AD-nnn
	Moving Shape	Moves a changeable object around the screen.	MV-nnn
	Animation	Moves a changeable object along predefined points on the screen.	AN-nnn
	Indirect Window	Designate to call a specific pop up window by a PLC word address.	WP-nnn
	Direct Window	Control a preset pop up window by a PLC bit address	WC-nnn
	Alarm Display	Display alarm messages as triggered by word value in a defined location.	AL-nnn
	Trend Display	Periodically retrieve a group of PLC register data and display as values verses time.	TD-nnn
	XY Plot	Periodically retrieve groups of PLC register data and display as X values verses Y values.	XY-nnn
	Bar Graph	Display the PLC register data as a bar graph (horizontal or vertical).	BG-nnn
	Meter Display	Display the PLC register data as an analog meter (semi-circle or full circle).	MD-nnn
	Alarm Bar	Display alarm messages as triggered by bit status, scrolling on a single line.	AB-nnn
	Recipe Action	Download a block of registers to the controller or upload and save a block of registers to the MMI.	RP-nnn
	Event Display	Display prioritized and formatted alarm messages as triggered by bit status in a defined location.	ED-nnn

13.1 Bit Lamp

A Bit Lamp displays the ON or OFF status of a designated bit address. If the bit status is OFF, the state 0 shape is displayed. If the bit status is ON, the state 1 shape is displayed, and so on. The corresponding label is also displayed if **Use Label** or **Use Label Library** are enabled. Bit Lamps cannot be used to change the state of a bit. Use the Set Bit or Toggle Switch parts to change Bit State.

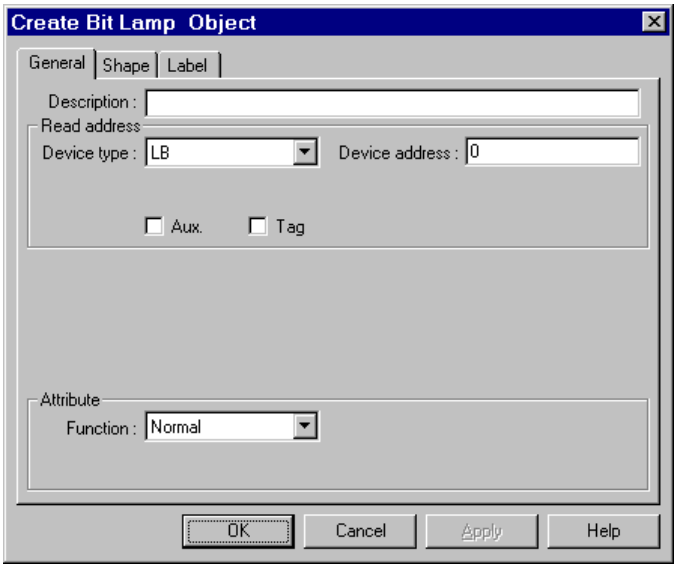


Read bit status = OFF Read bit status = ON

Procedure to place a Bit Lamp

Note: Refer to Software Reference Guide, Part Placement for further details about completing each tab item.

Step 1. Click  tool or select **Parts|Bit Lamp** from the menu.



Step 2. Fill in **General** Tab Items:

Description: A reference name (not displayed) that you assign to the Bit Lamp.

Read Address: Bit that controls the Bit Lamp state, shape and label. See section on Part Placement for more details about entering a Read address.

Device type is the bit type prefix.

Device address is the bit number.

Tag is used to select the Read address from a list of predefined Tags.

Aux. directs the Read address to be retrieved from the Auxiliary Port.

Attribute: This area is used to set the display properties.

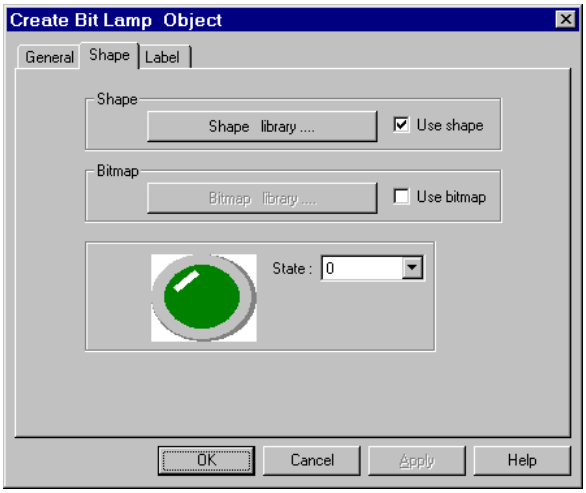
Function. There are two Function types for Bit Lamps: Normal and Blinking. If a blinking function is selected, the **Break time** needs to be set.

Function Summary:

Attribute	Description	Display at State 0	Display at State 1
Normal	Bit Lamp is displayed as the State (0 or 1) of its Shape/Bitmap and /or Label.	State 0 Shape and Label	State 1 Shape and Label
ON Blinks State 0	The State 0 Shape/Bitmap and /or Label blinks when the Read address is ON.	State 0 Shape and Label	State 0 Shape and Label blinking
ON Blinks State 1	The Shape/Bitmap and /or Label changes to State 1 and blinks when the Read address is ON.	State 0 Shape and Label	State 1 Shape and Label blinking

Break time is used to set the blink duty cycle. The Integer entered in the Break time field represents half of the 50-50 duty cycle in 1/10ths of a second. Range is 0 to 2147483647.

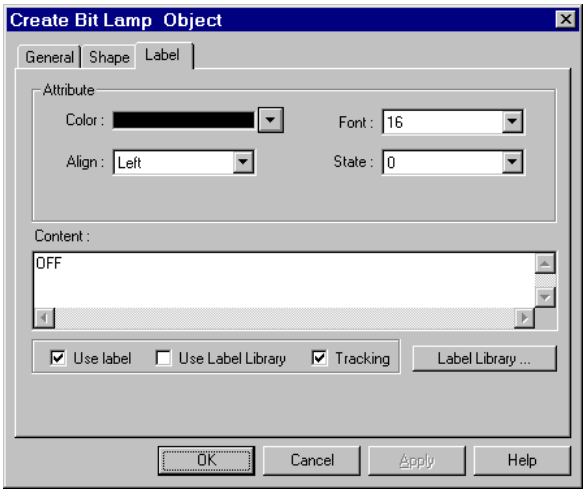
Step 3. Click the **Shape** Tab: Select **Use shape** and/or **Use bitmap**. Click **Shape library** or **Bitmap library** to bring up a library for selecting the Shape or Bitmap to display the 0(OFF) and 1(ON) states.



Note: If the Shape or Bitmap only has one state, the lamp seems to disappear when the Bit Lamp changes from that state.

Note: When both Shape and Bitmap are selected, the Shape always over writes the Bitmap graphic.

Step 4. Go to the **Label** Tab and fill in **Attribute** fields. See section on Part Placement for more details about entering Labels.



Color is the color of the text for the indicated State. Select one of 256 colors or 4 shades of bluescale.

Font is the size of the text. Text size can be different for each state.

Align is only used if multiple lines are in the label.
Note: Align justifies the text. It does not justify the text relative to the shape/bitmap. Center does not center the text in the area of the shape/bitmap.

Use the **State** dropdown to toggle between the State 0 and State 1 texts.

Content contains the text to denote State 0 and State 1.
Select **Use label** to enter a label in the **Content** field and make it active for the part.
Select **Use Label Library** to choose a predefined label from the Label Index dropdown and make it active for the part.

Select **Tracking** (if appropriate).

Tracking: Tracking is used to align all of the labels from the different states after placement. This means that if the label in State 0 is moved the labels in all other states are moved also in the same direction and position.
Note: This might be desirable for an annunciator but inappropriate for a toggle switch.

Step 5. Click **OK** to place the Bit Lamp on the window. Position the Bit Lamp and resize it if necessary. Adjust the label position as desired. Click once to select the Bit Lamp. Click the BitLamp again to adjust the Label Text. Double-click to edit the Bit Lamp.

Bit Lamp Design Tips:

1. Use the Bit Lamp to Show/Hide static shapes and backgrounds. For example: Putting a red frame around some parameters that have become out of range.

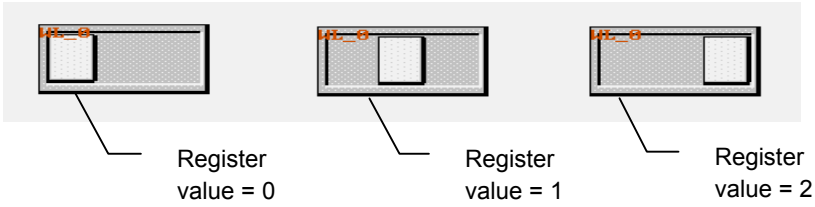
2. Assign Help text to State 1 and no text to State 0. Use the Bit Lamp to display that help when it is appropriate by forcing the Read address ON. (Alarm condition, during setup, on user request, etc.) A Shape or Bitmap is not needed in this instance.

3. Use the **Multi. Copy…** command to make a bank of indicators from a single Bit Lamp.

13.2 Word Lamp

A Word Lamp changes state according to the value in the designated word address. If the value equals 0, the first shape is displayed. If the value equals 1, the second shape is displayed, and so on. The corresponding label for the state is also displayed if “Use Label” check box is enabled.

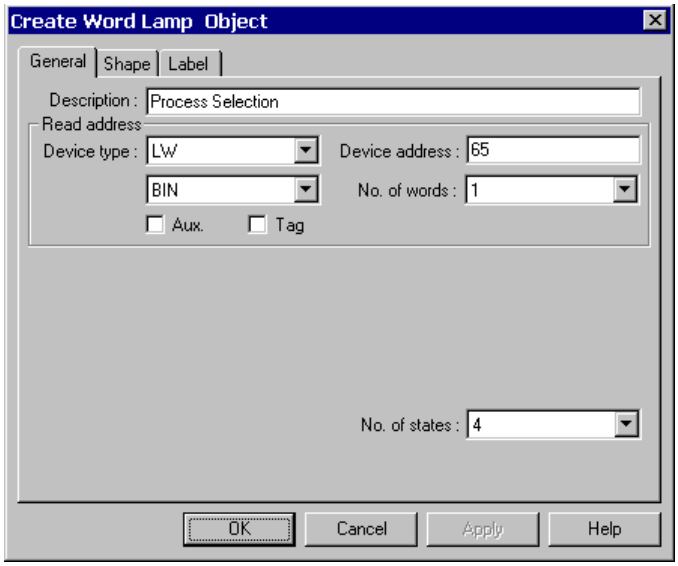
Example



Procedure to place a Word Lamp

Note: Refer to Section 2: Software Reference Guide, Part Placement for details about completing each tab item.

1. Click Word Lamp icon or select **Word Lamp** from the **Parts** menu.



2. Fill in General Tab Items:

Description: A reference name (not displayed) that you assign to the Word Lamp.

Read Address: PLC Word that controls the Word Lamp state, shape and label.

Device type is the word prefix.

Device address is the word number.

No. of words is fixed at 1 for this part.

Tag is used to select the Read address from a list of predefined Tags.

Aux. directs the Read address to be retrieved from the Auxiliary Port.

Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**)

Data format types

BIN (binary format) Decimal numbers are encoded by bit weight

Bit number	15															0
Weight	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

BCD (binary coded decimal format) Decimal numbers are encoded by bit weight per 4-bit nibble

15			12	11			8	7			4	3			0
10^{3}				10^{2}				10^{1}				10^{0}			

For example: The Read address raw data is **0000 0100 0010 0101**.

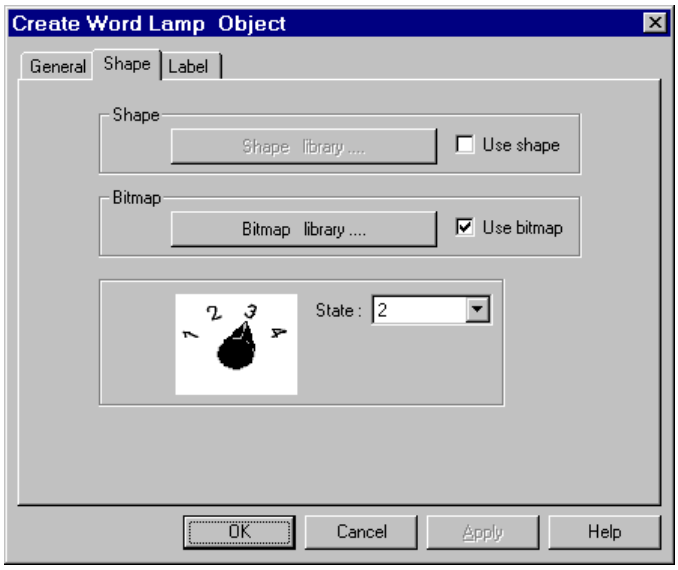
The **BIN** format treats the data as **1061**

The **BCD** format treats the data as **425**

No. of states: The value of the PLC register data minus 1. (Range: 1 to 32)

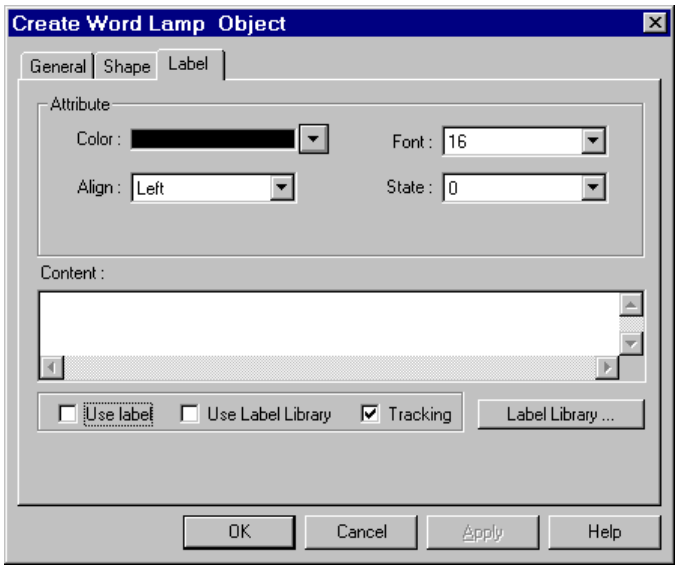
For example: A 3-position switch would have the No. of States set to 3.

3. Go to **Shape** Tab: Select **Use shape** and/or **Use bitmap**. Click **Shape library** or **Bitmap library** to display the library for choosing the Shape or Bitmap to display the word states.



> ***Note:*** If the Shape or Bitmap only has one state, the lamp seems to disappear when the Word Lamp changes from that state.
>
> ***Note:*** When both Shape and Bitmap are selected, the Shape always over writes the Bitmap graphic.

4. Go to **Label** Tab: Fill in text to denote each state.
   Note: Refer to Bit Lamp procedure for more details on filling out the Label tab.



5. Click **OK** to place the Word Lamp on the window. Position the Word Lamp and resize it if necessary. Adjust the label positions as desired.

Note: Refer to Section 2: Software Reference Guide, Part Placement for details about completing each tab item.

Word Lamp Design Tips:

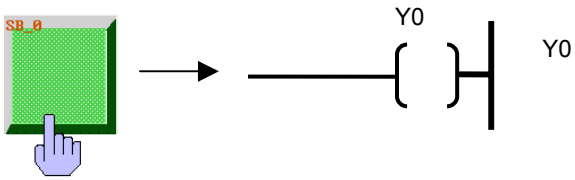
1. Use the Word Lamp as a Mini-Message Center. The Read address becomes the message triggering word. The number of messages is the number of states (1-32). Select an appropriate shape for the background. In the Label tab, enter a message for each state. To call up a message, just put its state number in the Read address word.

2. The word lamp can also be used to create traffic lights, stack pole lights and other multi-colored indicators.

3. Create animation by saving sequential bitmaps of a process to a bitmap library. Put a Periodical JOG++ Set Word part on the same window to animate the Word Lamp part.

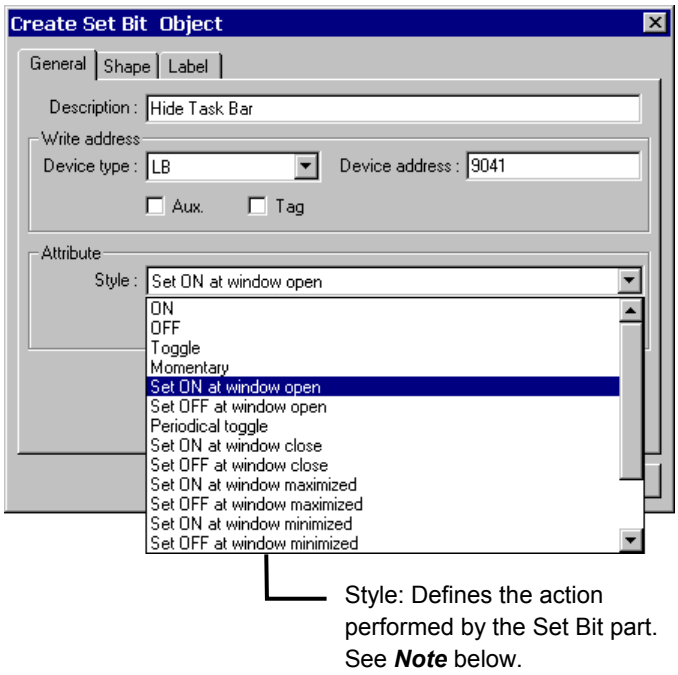
13.3 Set Bit

The Set Bit Part defines a touch area, that when activated, changes the state of a specified bit.



Procedure to place a Set Bit

1. Click Set Bit icon or select **Set Bit** from the **Parts** menu.



Style: Defines the action performed by the Set Bit part. See ***Note*** below.

2. Fill in General Tab Items:
   Description: A reference name (not displayed) that you assign to the Set Bit Part.

   Write Address: Bit t hat is acted on by the Set Bit action.
   - **Device type** is the bit prefix.
   - **Device address** is the bit number.
   - **Tag** is used to select the Write address from a list of predefined Tags.
   - **Aux.** directs the Read address to be retrieved from the Auxiliary Port.

   Attribute: Style: Defines the action performed by the Set Bit part. See ***Note*** below.

3. Go to **Shape** Tab: Select Shape or Bitmap to display graphics that represent the touch area.

   Note: Refer to Bit Lamp procedure for more details on filling out the Shape tab.

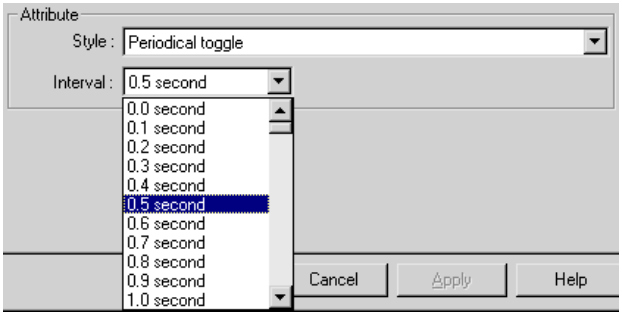
   Note: Shapes and bitmaps assigned to Set Bit parts change state only when the part is touched. They do not change state according to the Write address' state.

4. Go to **Label** Tab: Fill in fields to denote states.
   Note: Refer to Bit Lamp procedure for more details on filling out the Label tab.

5. Click **OK** to place the Set Bit part on the window. Position the Set Bit part and resize it if necessary. Adjust the label position as desired.

Note: Refer to Section 2: Software Reference Guide, Part Placement for details about completing each tab item.

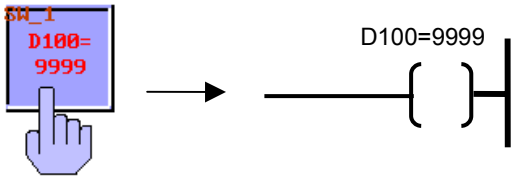
Note: Set Style Functions

Set Style	Description of action
ON	When the Set Bit Part is pressed, the designated Device address is turned ON. The state continues (i.e. remains ON) even after release.
OFF	When the Set Bit Part is pressed, the designated Device address is turned OFF. The state continues (i.e. remains OFF) even after release.
Toggle	Every time the Set Bit Part is pressed, the designated PLC bit is complemented (ON → OFF, OFF → ON)(Toggles).
Momentary	Only while the Set Bit Part is held down (pressed), is the specified Device address turned ON. Likewise, when the switch is released, the specified bit address is turned OFF. ***Note:*** This software has three operation modes to handle momentary switch operations, based on the current value of LW9044. Use an " Set ON at window open" Set Word part in the start up window to configure the desired operating mode. Operating Modes are as follows: **0:** Window popup is enabled during touch operations (down and up), when the touch is released (up) the BIT that was previously set ON is set OFF even if the popup window hides the momentary switch. (Default Mode of operation) **1**: Window popup is disabled during touch operations (down and up). This prevents popup windows from hiding momentary switches. **2:** Window popup is enabled during touch operations (down and up). If the momentary switch is hidden by a popup window at the time touch release (up), the BIT is <u>not</u> reset. ***Note:*** **DO NOT USE HMI MOMENTARY OPERATIONS FOR CRITICAL FUNCTIONS**. Communications failures between the HMI and the controller during a momentary operation may cause the HMI to ignore touch release thus causing damage to machinery and operator.
The following styles do not require touch to be activated. They are activated by the specified event or condition.	
Periodical toggle	Changes the state of the designated bit at a regular interval as selected in the Interval dropdown. Interval time in actual operation may vary from specified. Do not use HMI intervals for critical timing.  **Design Tip:** Use a Set Bit, Periodical toggle part on the common window to reset a timer in the controller. The timer can then be used as a watchdog to tell if the HMI communications link has been broken.

Set Style	Description of action
Design Tip: Use the following on window open/close Set Bit parts to initiate actions in the HMI and Controller. For example you might want to turn a bit on/off in the controller when the operator enters/exits the setup window. The Controller would then suspend operations pending the change in setup.	
Set ON at window open	When the window containing this Set Bit Part is opened, the designated Device address is turned ON.
Set OFF at window open	When the window containing this Set Bit Part is opened, the designated Device address is turned OFF.
Set ON at window close	When the window containing this Set Bit Part is closed, the designated Device address is turned ON.
Set OFF at window close	When the window containing this Set Bit Part is closed, the designated Device address is turned OFF.
Design Tip: Use the following on window maximize/minimize Set Bit Parts to initiate actions depending a window state.	
Set ON at window maximized	When the window containing this Set Bit Part is maximized, the designated Device address is turned ON.
Set OFF at window maximized	When the window containing this Set Bit Part is maximized, the designated Device address is turned OFF.
Set ON at window minimized	When the window containing this Set Bit Part is minimized, the designated Device address is turned ON.
Set OFF at window minimized	When the window containing this Set Bit Part is minimized, the designated Device address is turned OFF.
Design Tip: Use the following on backlight Set Bit Parts to initiate actions that depend on the screen being active. For example: A Set Bit part of this type might signal a process to go to “sleep mode” because no one is monitoring it.	
Set ON at backlight off	When the backlight is turned off, the designated Device address is turned ON.
Set OFF at backlight off	When the backlight is turned off, the designated Device address is turned OFF.
Design Tip: Use the following on data entry Set Bit Parts to initiate actions that depend on valid data. For example: Use a Set ON at enter success to trigger the controller to advance the HMI to the next window.	
Set ON at enter success	When a function key with the ENT attribute is pressed in response to a Numeric Input or ASCII Input part and the attempt to write the data succeeds, the designated Device address is turned ON.
Set OFF at enter success	When a function key with the ENT attribute is pressed in response to a Numeric Input or ASCII Input part and the attempt to write the data fails, the designated Device address is turned OFF.

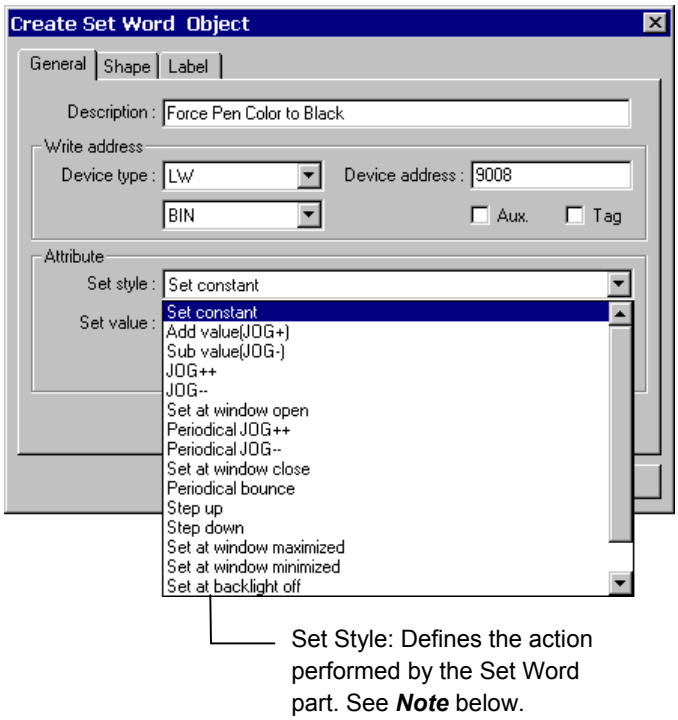
13.4 Set Word

The Set Word Part defines a touch area that when activated, writes a predefined value (constant) to the designated PLC address.



Procedure to place a Set Word

1. Click Set Word icon or select **Set Word** from the **Parts** menu.



Set Style: Defines the action performed by the Set Word part. See ***Note*** below.

2. Fill in General Tab Items:

Description: A reference name (not displayed) that you assign to the Set Word

Write Address: Word in the MMI/PLC that is affected by the Set Word control. Data in the register may be interpreted as Binary or BCD information (See note on **Word Lamp** Part).

Device type is the word prefix.
Device address is the word number.
Tag is used to select the Write address from a list of predefined Tags.
Aux. directs the Read address to be retrieved from the Auxiliary Port.

Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**). See Word Lamp 13.2 for format description.

Attribute: The action that is performed on the Write address. See ***Note*** below for an explanation of each Set style and its accompanying fields.

3. Go to **Shape** Tab: Select Shape or Bitmap to display graphics that represent the touch area.

Note: Refer to Bit Lamp procedure for more details on filling out the Shape tab.

Note: Shapes and bitmaps assigned to Set Word parts change state only when the part is touched. They do not change state according to the Write address' state.

4. Go to **Label** Tab: Fill in fields to denote states.

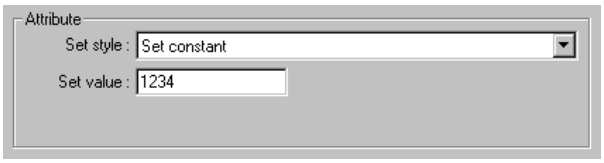
Note: Refer to Bit Lamp procedure for more details on filling out the Label tab.

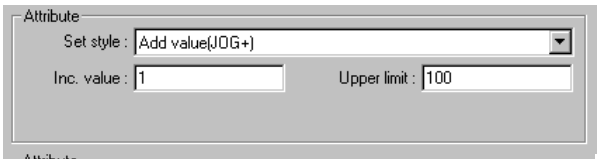
5. Click **OK** to place the Set Word part on the window. Position the Set Word part and resize it if necessary. Adjust the label position as desired.

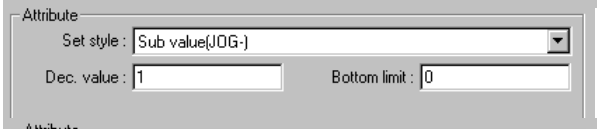
Note: Refer to Section 2: Software Reference Guide, Part Placement for details about completing each tab item.

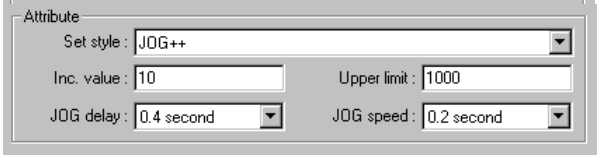
Note: Set Style Attributes

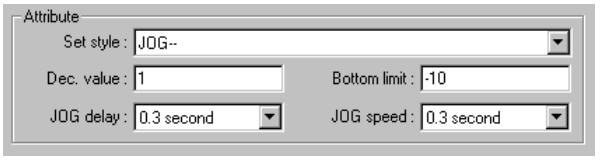
Set Style	Parameters	Description
Set constant	**Set value**	When pressed, preset **Set value** data is written to the designated word. **Design Tip:** Use to put fixed data into words. Examples: force default values; return to home positions;
Design Tip: Use the JOG attributes to create Thumbwheels, Adjust +/-, and Spin controls.		
Add value (JOG+)	**Inc. value** **Upper limit** Maximum: 32767	Every time pressed, the **Inc. value** is added to the designated word, if the result is not larger than **Upper limit**.
Sub value (JOG-)	**Dec. value** **Bottom limit** Minimum: -32768	Every time pressed, the **Dec. Value** is subtracted from the value of the designated word, if the result is not less than **Bottom Limit**.
JOG++	**Inc. value** **Upper limit** **JOG delay** **JOG speed** Maximum: 32767	When pressed, the **Inc. value** is added to the designated word at regular intervals as specified by **JOG delay*** and **JOG speed***, if the result is not larger than **Upper limit**.
JOG- -	**Dec. value** **Bottom limit** **JOG delay** **JOG speed** Minimum: -32768	When pressed, the **Dec. value** is subtracted from the value of the designated word at regular intervals as specified by **JOG delay*** and **JOG speed***, if the result is not less than **Bottom limit**.





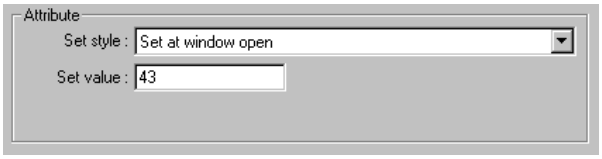


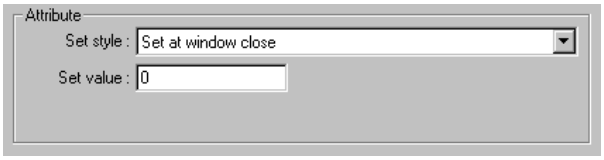




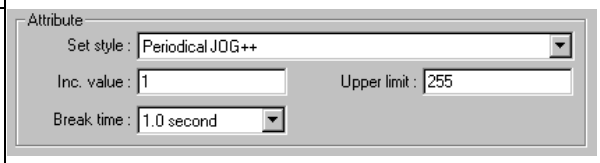
***JOG Delay**: The interval that the part must be touched before repeat action occurs.
***JOG Speed**: The interval between repeat actions once repeating begins.
Note: Interval times in actual operation may vary from specified. Do not use HMI intervals for critical timing.

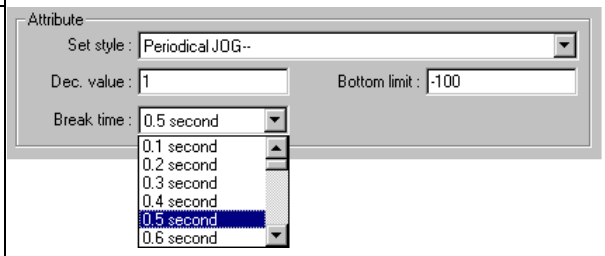
Set Style	Parameters	Description
Set at window open	**Set value**	When the window containing this part is opened, the **Set value** data is written to the designated word. **Design Tip:** Use this attribute to send the window number to the PLC so it knows what window is open.
Set at window close	**Set value**	**Set value** data is written to the designated word automatically when the window containing it is closed. **Design Tip:** Use this attribute to force the password state to 0 when a secure window is closed.

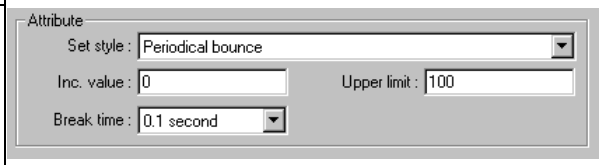




Set Style	Parameters	Description
Design Tip: Use the Periodical styles along with the Animation and Move parts to make window animations.		
Set Style	Parameters	Description
Periodical JOG++	**Inc. value** **Upper limit** **Break time** Maximum: 32767	This Attribute becomes active automatically when the window is active. The **Inc. value** is added to the designated word, if the result is not larger than **Upper limit**. Once the upper limit is reached or exceeded, the value returns to zero and incrementing begins again. The **Break time** is the time between increments.
Periodical JOG--	**Dec. value** **Bottom limit** **Break time** Minimum: -32768	This Attribute becomes active automatically when the window is active. The **Dec. value** is subtracted from the designated word, if the result is not less than **Bottom limit**. Once the bottom limit is reached or exceeded, decrementing stops. The **Break time** is the time between decrements.
Periodical bounce	**Inc. value** **Upper limit** **Break time**	This Attribute becomes active automatically when the window is active. The **Inc. value** is added to the value of the designated word at regular intervals as specified by the **Break time** until the **Upper limit** is reached. Then the **Inc value** is subtracted from the designated word until zero is reached. This cycle is repeated until the window becomes inactive.

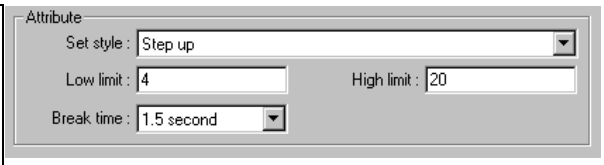


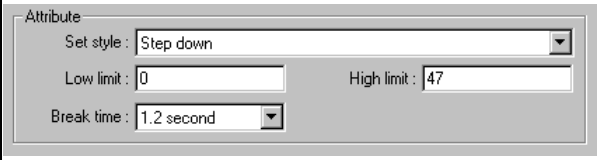
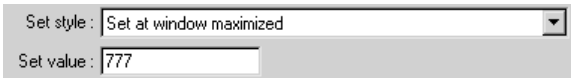




Note: Interval times in actual operation may vary from specified. Do not use HMI intervals for critical timing.

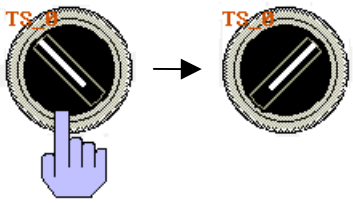
Step up	**Low limit** **High limit** **Break time**	This Attribute becomes active automatically when the window is active. The value is incremented by 1 at regular intervals as specified by the **Break time** until the **High limit** is reached. Then the value is reset to the **Low limit**. This cycle is repeated until the window becomes inactive. **Design Tip:** Use the Step styles along with the Move part to animate graphics within a fixed pixel boundary.



Set Style	Parameters	Description	
Step down	**Low limit** **High limit** **Break time**	This Attribute becomes active automatically when the window is active. The Value is decremented at regular intervals as specified by the **Break time** until the **Low limit** is reached. Then the Value is reset to the **High limit**. This cycle is repeated until the window becomes inactive.	
Set at window maximized	**Set value**	**Set value** data is written to the designated word automatically when the window containing it is maximized.	
Set at window minimized	**Set value**	**Set value** data is written to the designated word automatically when the window containing it is minimized.	
Set at backlight off	**Set value**	**Set value** data is written to the designated word automatically when the backlight is turned off.	
Design Tip: Use the following to provide error codes to the PLC after data entry.			
Set at enter success	**Set value**	When a function key with the ENT attribute is pressed in response to a Numeric Input or ASCII Input part and the attempt to write the data succeeds, the designated Device Address is loaded with the **Set value**.	
Set at enter fail	**Set value**	When a function key with the ENT attribute is pressed in response to a Numeric Input or ASCII Input part and the attempt to write the data fails, the designated Device Address is loaded with the **Set value**.	

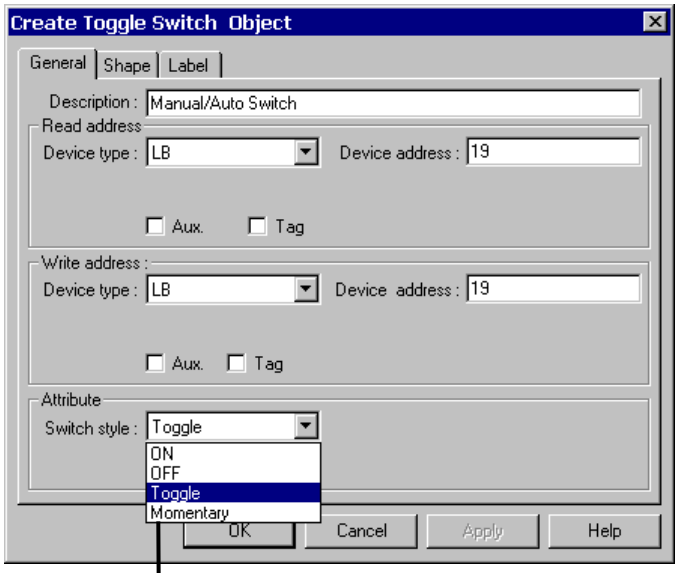
13.5 Toggle Switch

The toggle switch is a combination of Bit Lamp and Set Bit. It represents the ON/OFF status of a PLC bit address, and defines a touch area, when activated, it may turn the same or different bit ON or OFF.



Procedure to place a Toggle Switch.

1. Click Toggle Switch icon or select **Toggle Switch** from the **Parts** menu.



Switch style: Defines the action performed by the Toggle Switch part. See ***Note*** Switch Style Functions below.

2. Fill in General Tab Items:

Description: A reference name (not displayed) that you assign to the Toggle Switch.

Read address: Bit that controls the Toggle Switch state, shape and label.

Device type is the bit prefix.
Device address is the bit number.
Tag is used to select the Read address from a list of predefined Tags.
Aux. directs the Read address to be retrieved from the Auxiliary Port.

Write address: Bit that is affected by the Toggle Switch Set Style.

Device type is the bit prefix.
Device address is the bit number.
Tag is used to select the Write address from a list of predefined Tags.
Aux. directs the Write address to be retrieved from the Auxiliary Port.

Switch style: See ***Note*** Switch Style Functions below for an explanation of each Switch style.

3. Go to **Shape** Tab: Select Shape or Bitmap to display graphics that represent the touch area.

Note: Refer to Bit Lamp procedure for more details on filling out the Shape tab.

Note: Shapes and Bitmaps assigned to Toggle Switch parts change state according to the Read address' state.

4. Go to **Label** Tab: Fill in fields to denote states.
Note: Refer to Section 2: Software Reference Guide, Part Placement for more details on filling out the Label tab.

5. Click **OK** to place the Toggle Switch part on the window. Position the Toggle Switch part and resize it if necessary. Adjust the label position as desired.

Note: Refer to Section 2: Software Reference Guide, Part Placement for details about completing each tab item.

Note: Switch Style Functions

Switch Style	Description
ON	When the Set Bit Part is pressed, the Write address is turned ON. The state continues (i.e. remains ON) even after release.
OFF	When the Set Bit Part is pressed, the Write address is turned OFF. The state continues (i.e. remains OFF) even after release.
Toggle	Every time the Set Bit Part is pressed, the Write address is complemented (ON → OFF, OFF → ON)(Toggles).
Momentary*	Only while the Set Bit Part is held down (pressed), is the Write address turned ON. Likewise, when the switch is released, the Write address is turned OFF. ***Note:*** **DO NOT USE HMI MOMENTARY OPERATIONS FOR CRITICAL FUNCTIONS**. Communications failures between the HMI and the controller during a momentary operation may cause the HMI to ignore touch release thus causing damage to machinery and operator.

Note: The state of the toggle switch shape/bitmap corresponds directly to the state of the read bit.
This is different from the Set Bit part, which only changes shape/bitmap when touched.

Note: This software has three operation modes to handle momentary switch operations, based on the current value of LW9044. Use an "initial set" SET WORD in the start up window to configure the desired operating mode. Operating Modes are as follows:

0: Window popup is enabled during touch operations (down and up), when the touch is released (up) the BIT that was previously set ON is set OFF even if the popup window hides the momentary switch. (Default Mode of operation)

1: Window popup is disabled during touch operations (down and up). This prevents popup windows from hiding momentary switches.

2: Window popup is enabled during touch operations (down and up). If the momentary switch is hidden by a popup window at the time touch release (up), the BIT is <u>not</u> reset.

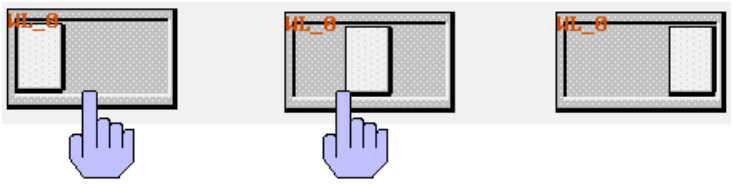
Design Tips:

1. Use the same device address for read and write to have the Toggle Switch show the state of the device it is switching.

2. Use different Read and Write addresses for processes that need confirmation of state. Example: A motor start up might need to be up to speed before operations can be performed. Use the Write address to start the motor and the Read address as an indicator of when the motor is up to speed.

3. Disable Label tab, Tracking to place Labels at different locations on the switch.

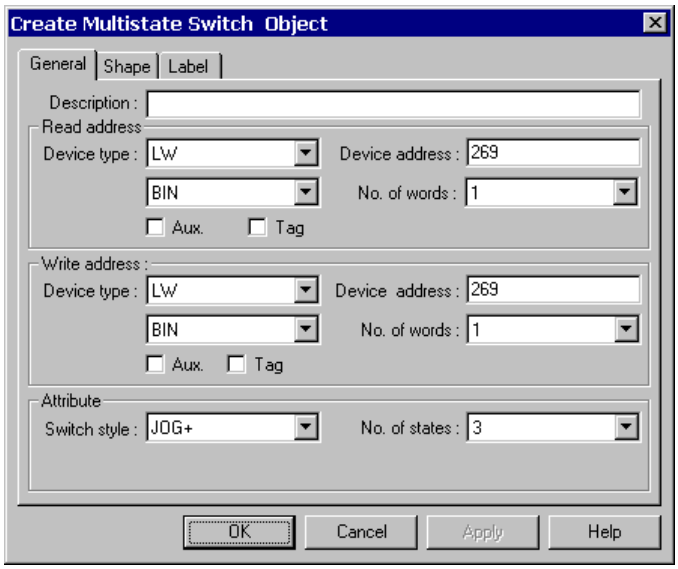
13.6 Multi-State Switch



The Multi-State Switch is a combination of Word Lamp and Set Word. It displays a different state depending on the value of the PLC monitor word address. It also defines a touch area that when activated, writes a specified data to the PLC word address, which may be the same as the Read address or a different address.

Procedure to place a Multi-State Switch

1. Click Multi-State Switch icon or select **MultiState Switch** from the **Parts** menu.



2. Fill in General Tab Items:
   Description: Reference text (not displayed) assigned to the Multi-State Switch.

   Read Address: Word in the PLC that controls the Multi-State Switch state, shape and label.
   Device type is the word prefix.
   Device address is the word number.
   Data in the register may be interpreted as Binary or BCD information.
   No. of words is restricted to 1 for this part.
   Tag is used to select the Read address from a list of predefined Tags.
   Aux. directs the Read address to be retrieved from the Auxiliary Port.

   Write Address: Word in the PLC that is controlled by the Multi-State Switch Style.
   Device type is the word prefix.
   Device address is the word number
   Data in the register may be interpreted as **BIN** (Binary) or **BCD** information.
   No. of words is restricted to 1 for this part.
   Tag is used to select the Write address from a list of predefined Tags.
   Aux. directs the Write address to be retrieved from the Auxiliary Port.

   Attribute:
   Switch Style: See table below

   No. of states: The maximum value of the register data minus 1 (The first state is when the register is at 0). The number of states is restricted to a maximum of **32** states.

3. Go to **Shape** Tab: Select Shape or Bitmap to display word states.

   Note: Refer to Bit Lamp procedure for more details on filling out the Shape tab.
   Note: If the Shape or Bitmap only has one state, the lamp seems to disappear when the Bit Lamp changes from that state.
   Note: When both Shape and Bitmap are selected, the Shape always over writes the Bitmap graphic.
   Note: Shapes and bitmaps assigned to Multi-State Switch parts do not change state when the part is touched. They only change state according to the Read address' state.

4. Go to **Label** Tab: Fill in fields to denote states.

5. Click **OK** to place the Multi-State Switch part on the window. Position the Multi-State Switch part and resize it if necessary. Adjust the label position as desired.

Note: Refer to Section 2: Software Reference Guide, Part Placement for details about completing each tab item.

Note: Switch Style Functions

Set Style	Description
JOG+	The value in the designated write address is increased by one when the Multi-State Switch is pressed.
JOG-	The value in the designated write address is decreased by one when the Multi-State Switch is pressed.

Note: Range is -32768 to 32767. Above 32767 the value cycles to -32768, below -32768 the value cycles to 32767.

Design Tips:

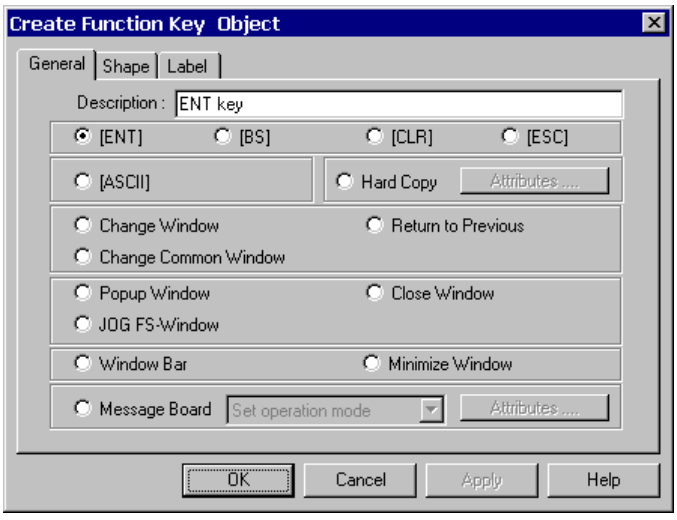
1. Use the same device address for read and write to have the Multi-State Switch show the state of the device it is changing.

2. Use the Multi-State switch to jog through text selections. For example, states 0-6 correspond to days of the week (Sunday through Saturday). Assign the labels of states 0-6 to correspond to the days of the week. You may or may not use a shape for this object. This Multi-State Switch allows the operator to select the day of the week. This is useful when setting the internal real time clock.

13.7 Function key

The Function Key is used to create an active screen area for a data entry key, inputting a digit or ASCII character, changing screen displays and minimizing or moving windows.

Procedure to place a Function Key



1. Click Function Key icon or select **Function Key** from the **Parts** menu.

2. Fill in General Tab Items:
   Description: A reference name (not displayed) that you assign to the Function Key.

Determine what function occurs when pressed, See options below.

3. Go to **Shape** Tab: Select Shape or Bitmap to display states that represent the touch area.

4. Go to **Label** Tab: Fill in fields to denote states.

5. Click OK, position the Function Key and resize it if necessary.
Click **OK** to place the Function Key part on the window. Position the Function Key part and resize it if necessary. Adjust the label position as desired.

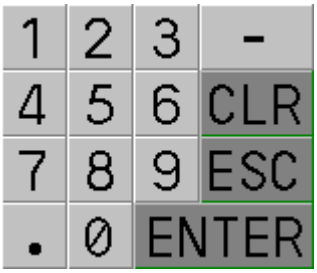
Note: Refer to Section 2: Software Reference Guide for details about completing each tab item

Function Key Topics

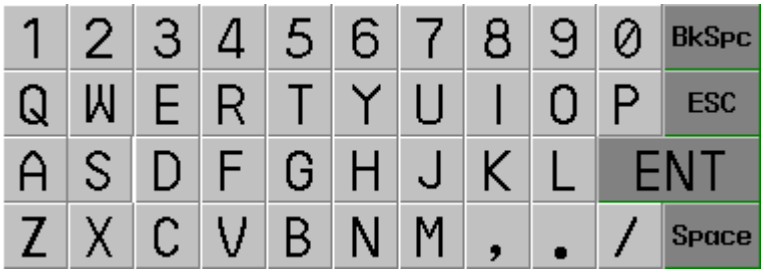
13.7.1 Character Codes and Creating a Keypad

A keypad is composed of a variety of function keys with different ASCII codes and special keys “ENT” (Enter), “BS” (Backspace), “CLR” (Clear) and ”ESC” (Escape).

By creating one function key then duplicating it, changing its character code and arranging the keys in a regular pattern, a keypad can be created. Since any character can be assigned to a function key, specialized keypads can be made for any application. Keypads can be saved in the Group Library for use wherever needed.



A typical Numeric keypad consisting of 0~9, ‘.’,’-‘, CLR, ESC and ENT



An ASCII keypad emulating a keyboard.

Keypads are used with the Numeric and ASCII Input Parts to change their values. As the first key is pressed the displayed value is cleared and new data entry begins.

Special Key Functions:

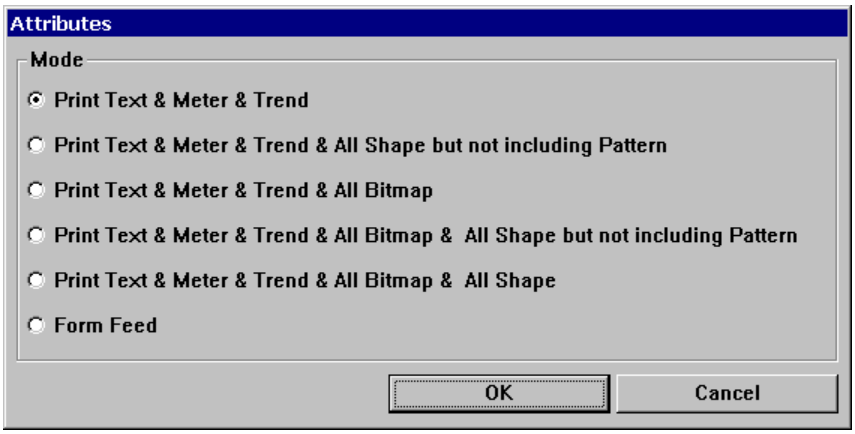
CLR key	Sets the displayed value to 0 or blank.
BS key	Erases previously entered characters from latest to earliest.
ESC key	Ends data entry without changing word values.
ENT key	Ends data entry and saves the displayed value to the word.

Note: Both the ENT and the ESC keys reset the System Reserved bits LB9060 - LB9069 to zero.

13.7.2 Hard Copy (Print Function)

Print functions facilitate the printing of screen data. Data is sent out as a graphic through the parallel printer port.

Note: not all models have print capabilities.



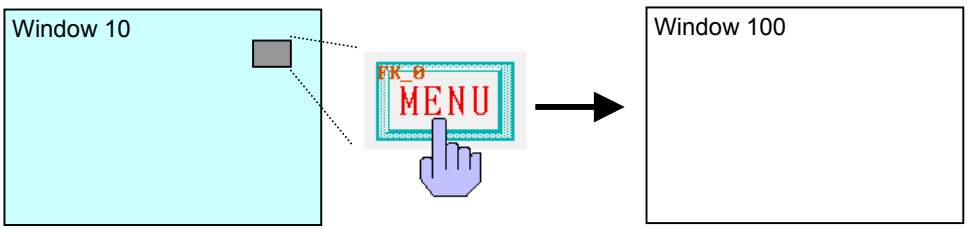
Use the **Attributes....** button to select the desired print action.

Selection	Print action
Print Text & Meter & Trend	Prints bit and numerical data parts along with drawn objects and text placed on the display.
Print Text & Meter & Trend & All Shape but not including pattern	Prints the outline of bit and numerical data parts along with drawn objects, text and shapes placed on the display.
Print Text & Meter & Trend & Bitmap	Prints bit and numerical data parts along with drawn objects, text and bitmaps placed on the display.
Print Text & Meter & Trend & Bitmap & All Shape but not including pattern	Prints the outline of bit and numerical data parts along with drawn objects, text, bitmaps and shapes placed on the display.
Print Text & Meter & Trend & Bitmap & All Shape	Prints bit and numerical data parts along with drawn objects, text, bitmaps and shapes placed on the display.
Form Feed	Prints a Form Feed command.

Note: Actual printer setup is done in the System Parameter, General tab. If no printer is selected the key has no effect.

13.7.3 Change Window

If the function key is set as “Change Window”, when the touch area is pressed (activated), the current window display terminates and changes to the window designated by “Window No.”.



Press Change Window Function key Button (in this example labeled MENU)

‘Jump to’ Window 100

Note: Be sure to select a valid window number. Project does not compile if Window No. is not defined.

13.7.4 Return to Previous

This function terminates the existing window and recalls the last active window.

For example: If window 21 was replaced by window 32, and window 32 has a Function defined as **Return to Previous**, then, when it is pressed, window 32 is terminated and window 21 is displayed.

Note: Indirect, Direct and Popup windows cannot be recalled.

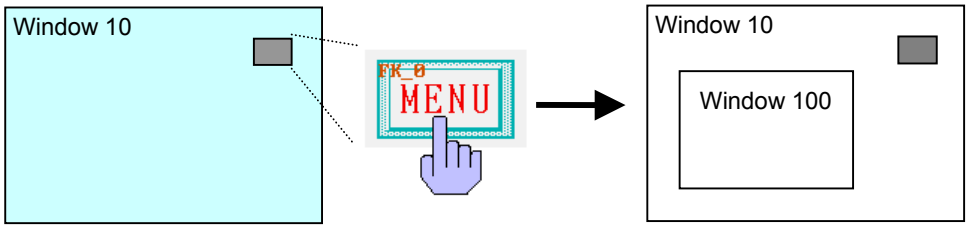
13.7.5 Change Common Window

If the function key is set as “Change Common Window”, when the touch area is pressed (activated), the current common window display terminates. The new common window designated by **Window No.** is displayed. Any full screen base window can be used as a replacement for the default Common window (Window 6).

Note: Be sure to select a valid window number. Project cannot compile if **Window No.** is not defined.

13.7.6 Popup Window

If the function key is set as “Popup Window”, when the touch area is pressed (activated), the window designated by Window No is displayed over top of the base window. The existing window(s) are not terminated.



‘Popup’ Window 100

Note: A window assigned to a function key as a Popup window and also assigned to a Direct or Indirect Window part on the same screen is not displayed when the Popup key is pressed. The Direct/Indirect Window control disables the function key. In addition, Indirect/Direct Windows disable Function key Popup keys placed in them.

13.7.7 Close Window

A Close Window function key terminates the window it is placed in when pressed. This is the method for operators to close Popup Windows.

Caution: This type of function key can also be placed on Base windows. Closing all windows may leave the operator with no way to proceed! Be sure to have a way for the operator or PLC to call up a window when this happens.

13.7.8 JOG FS-Window

This function changes the popup window that appears when the Fast Selection, Task Bar button is pressed.

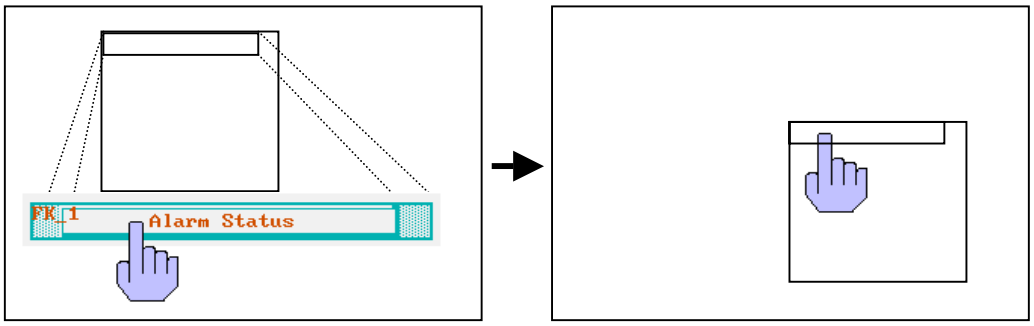
Note: The window specified must be the exact same size as the original Fast Selection window (Window 4).

Note: The Fast Selection window does not have to be showing when this button is touched in order for it to change.

Design Tips:
Use this key to change menus when going from operation to setup or help. These function keys can be placed directly on the Fast Selection window.

13.7.9 Window Bar

A function key defined with "**Window Bar**" attribute is used to move the position of a window on the screen. A Popup, Direct or Indirect window that is smaller than a full screen can make use of this function. The window with the "window bar" attributes, can be moved by touching the window bar area then touching a second position, the window is moved to that place. Typically, the Window Bar is placed across the top of the window with a label describing that window's purpose. The window bar label is displayed on the task bar icon when the window is minimized.



First touch activates the window move function

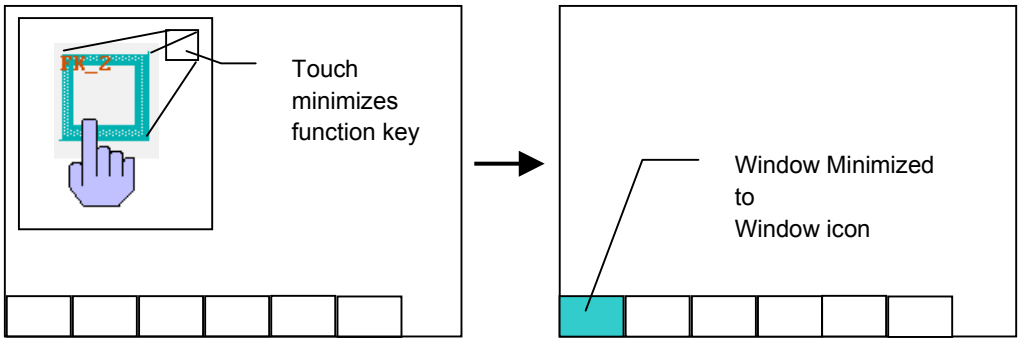
Second touch moves the window to the desired place

Note: The window bar can be used to move windows so they are partially off screen. This might appear to hide the window. However, a window can never be moved entirely off screen. If the Task Bar is showing, then the window's icon indicates that it is still open. Toggling off the Base window that called a Popup window closes the Popup window. Direct and Indirect windows can be recovered by toggling data in their control devices.

13.7.10 Minimize Window

When the function key is pressed, the window is minimized to an icon in the task bar. A touch on the window's icon maximizes the window again.

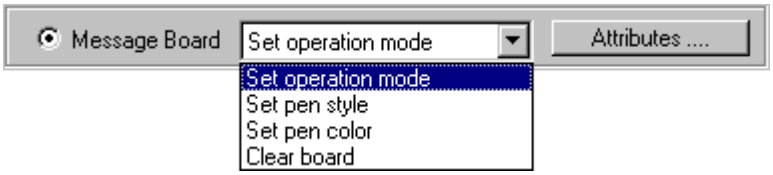
Note: The Minimize Window function works only if there is also a Window Bar on the same window.
Note: The Minimize Window function works only on Direct, Indirect and Popup windows.



13.7.11 Message Board

If the project has a message board, function buttons can be placed on the window that calls the Message Board screen to change the characteristics of the pen or erase the board.

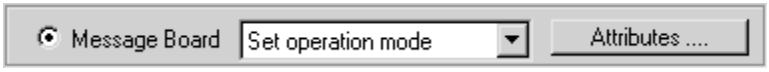
> ***Note:*** Do not put these functions directly on the Message Board. This does not cause harm but buttons may be displayed as stuck in the ON State when they are not in the OFF State.



Click on the **Message Board** radio button then select a type of action from the drop down list. Click on the **Attributes....** button to set the particular action for the function button. See below for an explanation of the actions.

13.7.11.1 Set Operation Mode

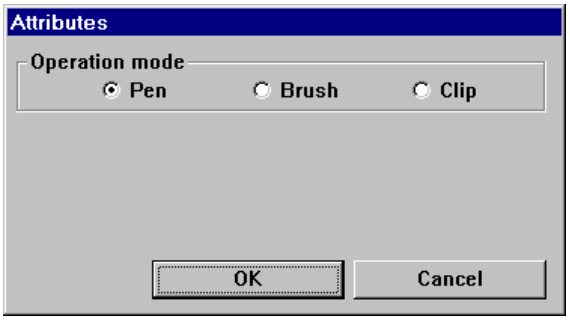
Set Operation is used to define how the touch acts on the message board.



Pen is used for drawing graphics/text on the Message Board window.

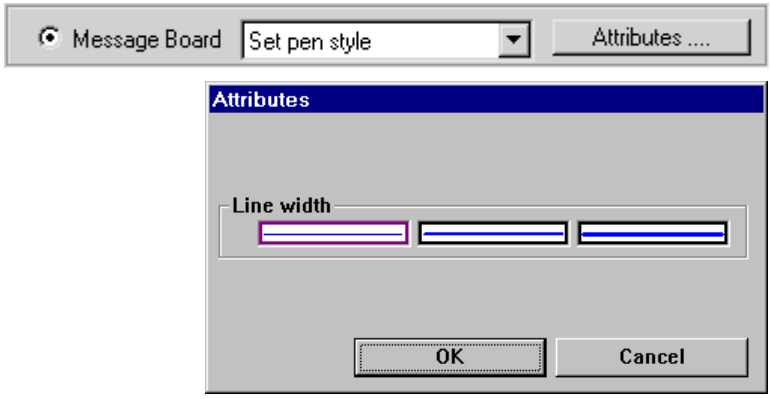
Brush is used as an eraser to erase lines drawn by the pen.

Clip allows the operator to drag a rectangle around an area. When the operator stops touching the screen the area in the rectangle is erased.



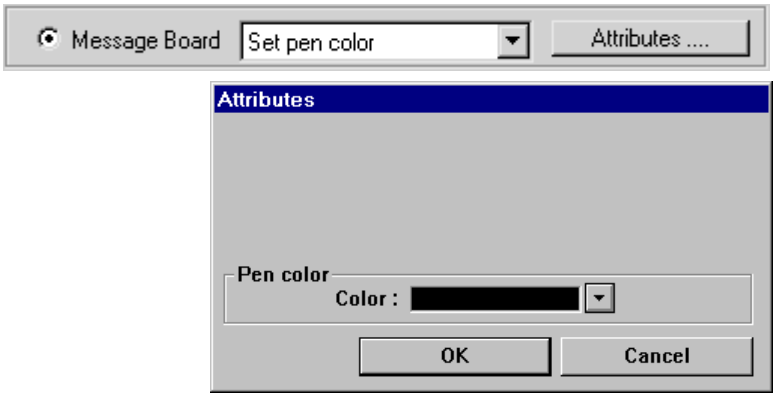
13.7.11.2 Set pen style

Set pen style is used to select the pen thickness. Click on the desired **Line width** and click **OK**.



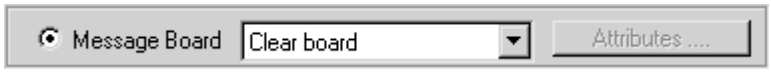
13.7.11.3 Set pen color

Set pen color is used to select the color of lines drawn on the Message window. Click on the **Color** dropdown, select the desired Color from the color chart, and click **OK**.



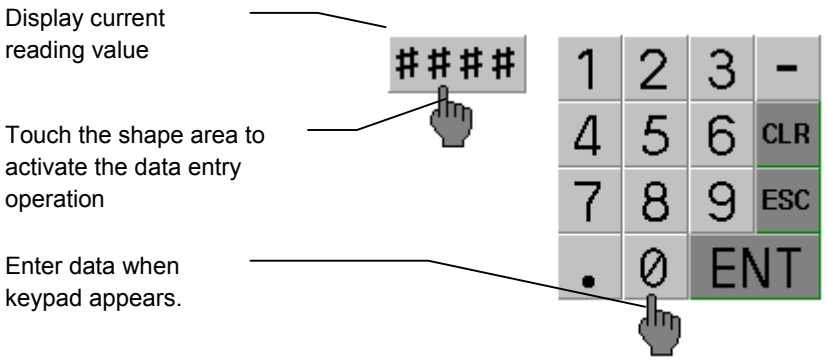
13.7.11.4 Clear board

There are no attributes for this selection. A function button with this action simply erases the entire Message window.



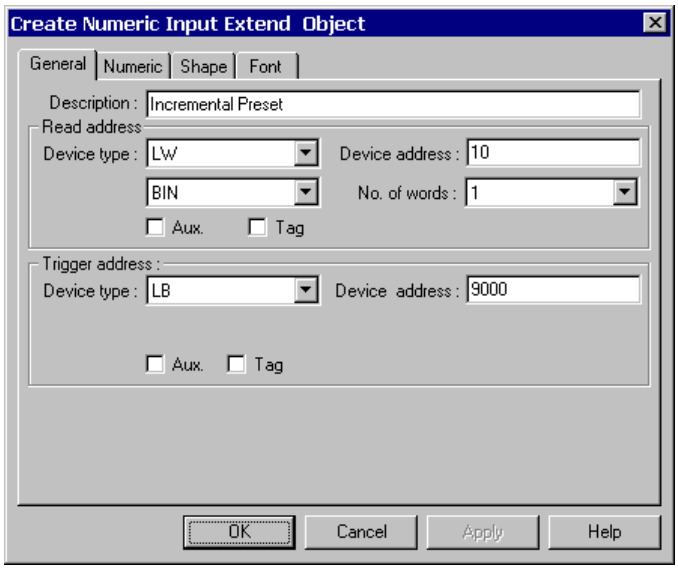
13.8 Numeric Input Extend

A Numeric Input displays the current value of a PLC register data. If the **Trigger address** bit is active, when the area of the shape is touched, a flashing cursor indicating input via keypad is activated. Use an already displayed keypad made of function keys to enter numeric data to the PLC register designated by the **Read address**.



Procedure to place a Numeric Input Extend

1. Click Numeric Input Extend icon or select **Numeric Input Extend** from the **Parts** menu.



2. Fill in General Tab Items:

Description: A reference name (not displayed) that you assign to the Numeric Input Extend.

Read Address: Word in the PLC that is displayed and modified.

Device type is the word prefix.
Device address is the word number.
Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
See Word Lamp 13.2 for format description.
No. of words:

For PLC types that handle data in words:
Select 1 Word (16 Bits) or 2 consecutive Words (32 Bits) for this part.
For PLC types that handle data in bytes:
Select 1 Word (8 Bits), 2 Words (16 Bits) or 4 Words (32 Bits) for this part.

Tag is used to select the Read address from a list of predefined Tags.
Aux. directs the Read address to be retrieved from the Auxiliary Port.

Trigger Address: Bit in the PLC that controls the ability to enter data into the Read Address. If bit is OFF, data entry is disabled.

Device type is the word prefix.
Device address is the word number.
Tag is used to select the Trigger address from a list of predefined Tags.
Aux. directs the Trigger address to be retrieved from the Auxiliary Port.

Trigger Address Design Tips:

1. LB 9000 to LB 9009 are internal bits that are set to 1 at power up. These bits can be used for Trigger addresses so that the Numeric Input parts are always active.

2. Have the PLC check a register for a password value and use a PLC bit to activate the Numeric Input for added security.

3. Fill in **Numeric** Tab items:
Display: Manipulates data so it is displayed in a useful format. See **13.8.1 Numeric Display Format** section below.

4. Go to **Shape** Tab: Select Shape or Bitmap to be used as a background button. The button is used to identify the part as an input part as opposed to just a regular data display. The button changes state when pressed.
Note: Refer to Bit Lamp procedure for more details on filling out the Shape tab.

5. Go to **Font** Tab: Fill in attributes for the displayed digits. See **13.8.2 Font Alignment** below for details.

6. Click **OK** to place the Numeric Input part on the window. Position the Numeric Input part and resize it if necessary. Adjust the number position just as you would any other label.

Note: Refer to Section 2: Software Reference Guide for details about completing each tab item

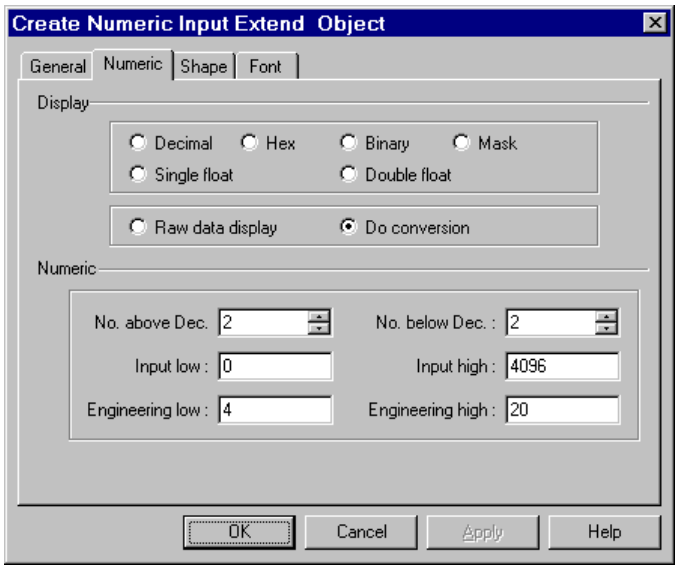
Note: **Reserved Local Bit for Keypad.**
When the user activates a Numeric or ASCII Input object, the HMI turns internal bits LB9060~LB9069 **ON**. After user input and entry, the HMI checks the input data, if the data is valid then LB9060~LB9069 are turned **OFF** otherwise the input remains active waiting for valid entry. This is used for controlling popup keypads, see below.

13.8.1 Numeric Display Format

Decimal:
For **Raw data Display**, the reading value is displayed in its original value as a decimal number in the range –32768 to 32767.

For **Do conversion,** the reading value is converted to engineering units before display.

Fill in the **Numeric** portion of the tab.
Specify number above decimal point and number below decimal point.



Scaling is performed as follows:

$$\text{Display value} = \frac{(\text{reading value} - \text{Input Low})}{(\text{Input High} - \text{Input Low})} * (\text{Engineering High} - \text{Engineering Low})$$

Note: PLCs can only accept integer values. This affects how the HMI enters and displays scaled values. It is especially evident when scaling from a smaller value to a larger value. For example: Scaling a 0-10 value to a 0-100 value only yields multiples of 10. A value of 25 can never be displayed. Entering a value of 29 returns a value of 20. The HMI does not "round". It can only truncate to the next lower value.

Note: The decimal point for **Do Conversion** shows the actual decimal portion from the scaling formula. Data entered in the Decimal portion is used for converting data back to the PLC. Make sure keypads used for data entry contain a Decimal point for these cases.
The Decimal point for **Raw data display** is for appearance only and does not indicate that scaling has been done. For example: The No. below Dec. is set to 2. The value from the PLC is 104. The value is displayed as 1.04. Entering a value of 1.55 sends 155 to the PLC.

Note: Be sure to set Input limits to the values that is sent to the PLC not displayed values. For example: Using the example in the note above, Input limits might be Input low = 0; Input high = 200; (not Input low = 0 Input high = 2).

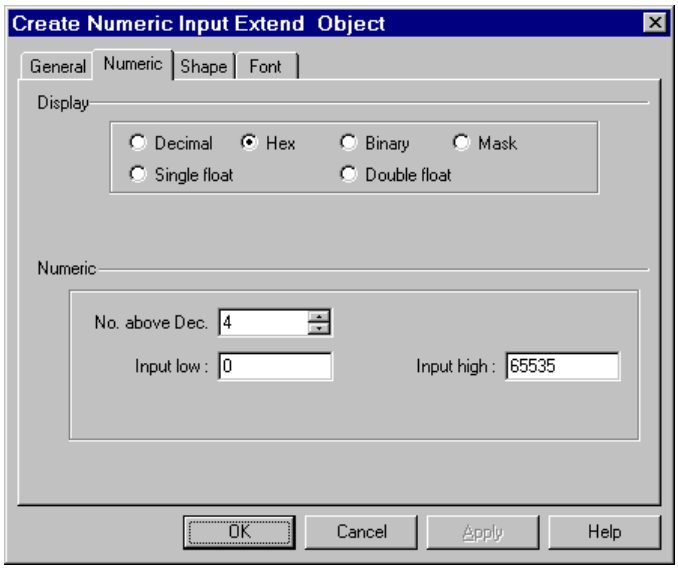
Hex: The number is displayed in hex (0~9, A~F) format. Scaling is disabled.

Binary: The number is displayed in binary (0 & 1) format. Scaling is disabled.

Mask: Displays only "****" and ignores the reading value. This is used for security code input. Scaling is disabled but decimal point is allowed.

Single float: Data in the controller is translated from a 32 bit IEEE Floating-Point format to a decimal number and displayed. Device address is changed to 2 words when this option is selected.

Double float: Data in the controller is translated from a 64 bit IEEE Floating-Point format to a decimal number and displayed. Device address is changed to 4 words when this option is selected.



Caution! Divide by 0 results in the PLC causes Floating-Point values that are invalid. These values cause the HMI to have a severe system error. Insure that floating point data is valid.

During data entry the **Input Low** and **Input High** are used to limit the range of valid input data.

Note: Input low and high are always decimal values. This true even if the displayed value is Hex or Binary.

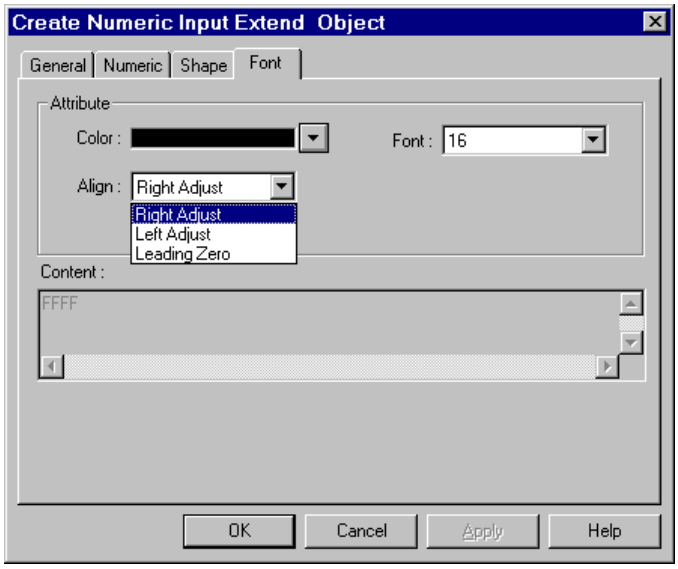
13.8.2 Font Alignment

The Font Attributes specify the size and color of display numeric digits. Available font sizes include 8, 16, 24, 32, 48, 64, 72 and 96.

Align selection is applicable for **Decimal** format only. Below is an example of how the alignment functions.
If a display format is set as:
"**No. above decimal** = 5",
"**No. below decimal** = 0" ,
With reading value 123, it is displayed in various formats as follows:

Right adjust			1	2	3
Left adjust	1	2	3		
Leading zero	0	0	1	2	3

Note: No label can be added to Numeric Input parts. Add text to the screen to provide labels for these parts.



Note: Refer to Section 2: Software Reference Guide for details about completing each tab item

Keypads and Pop-up Keypad Emulation

A default keypad is installed in Window 50 and called up in the common window of new projects. They may be deleted and your own keypads implemented. This is a brief example of how to create a pop-up keypad for use in your applications.

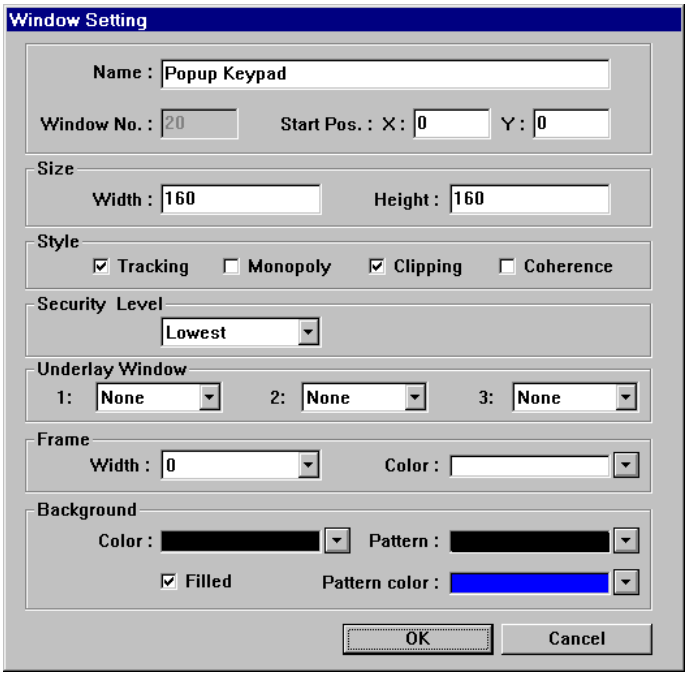
How it works when finished:

You touch the displayed numeric value.
A keypad pops up.
You edit the value.
Press the enter key.
The keypad disappears.
The new value is displayed.

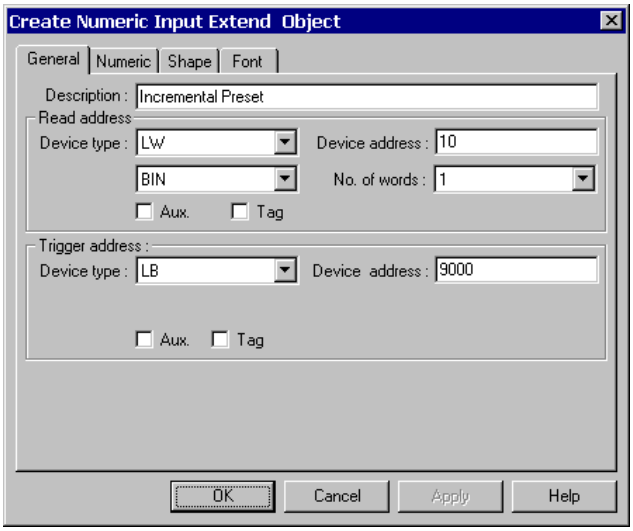
How to create it:

Step 1: Create the keypad with an Enter key and save it to a group library. Alternately you can use one of the keypads already provided in the keypads.glb library. (Located in the \library D-Subdirectory) The key is to group the keypad and find it's over all dimensions. This can be done easily by drawing a rectangle around the keypad then looking at its Attributes' Profile tab. For our example, we are using keypad 0 from the keypads.glb library. The dimensions are 160W x 160H pixels.

Step 2: Create a window with overall dimensions that match the dimensions of the keypad you are using. You may have to factor in a border if you select one for the window.
For Example, Use window 20. The dimensions are 160W x 160H pixels. If using a Frame, add extra pixels as needed. For this example, the Frame width is set to 0.

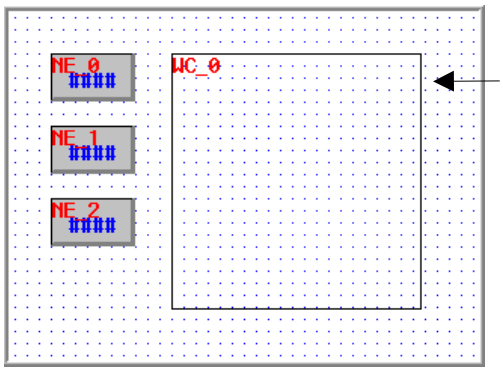


Step 3: Place keypad in this window.

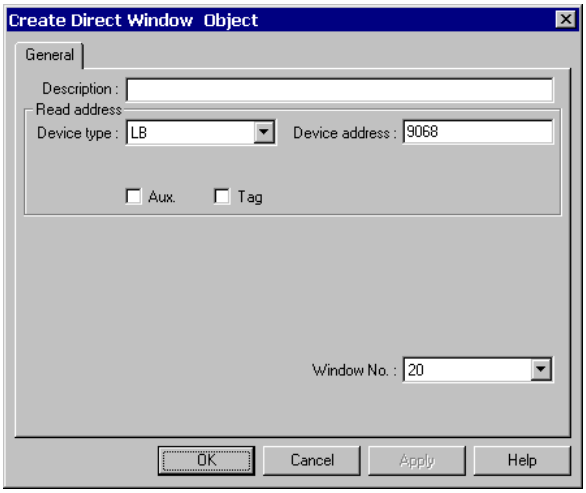


Step 4: Create Numeric Input Extend objects on the window you are developing for your project. Set the Trigger address attribute to LB9000. Be sure to set the Input low and high ranges on the numeric tab.
For our example, we are using Window 10 The input ranges are set as 0 to 1000. The objects are shown as NI_0, NI_1 and NI_2.

Step 5: Create a Direct Window object that is triggered by an appropriate bit that calls up the keypad into it. Once placed, adjust the Profile to be the same size as the dimensions of the keypad window. Do not place the Direct Window over top of the Numeric Input objects.



For our example, the Direct Window object is triggered by LB9068 (System's Input Active bit) and calls up Window 20. The Profile is set to 160W x 160H.



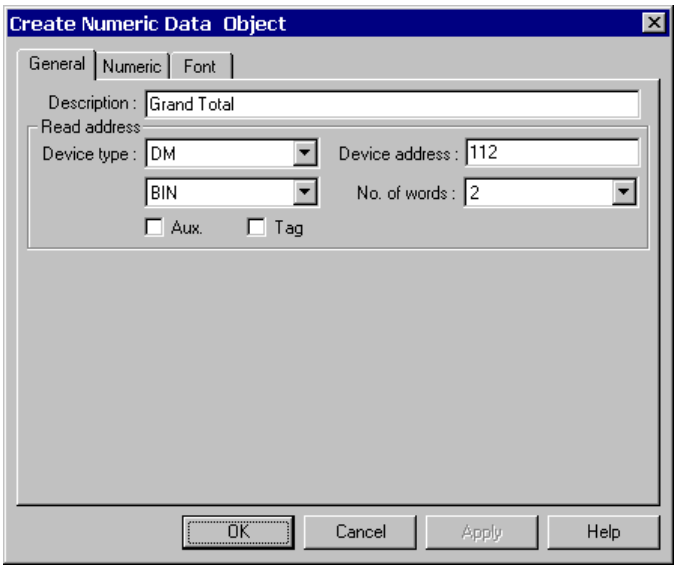
Step 6: Save, Compile and Simulate to verify proper operation.
When the Data Input object is touched the keypad should pop up and data entry should be activated. Enter a valid number, the keypad should disappear.

13.9 Numeric Data

A Numeric Data Part displays the current reading of a designated PLC register data. Data is displayed as text; no shape or bitmap can be associated with this part.

1234

Procedure to place a Numeric Data



1. Click Numeric Data icon or select **Numeric Data** from the **Parts** menu.

2. Fill in General Tab Items:
   Description: A reference name (not displayed) that you assign to the Numeric Data.

   Read Address: Word in the PLC that is displayed and modified by the Numeric Data Tab.
   Device type is the word prefix.
   Device address is the word number.
   Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
   See Word Lamp 13.2 for format description.
   No. of words:
   For PLC types that handle data in words:
   Select 1 Word (16 Bits) or 2 consecutive Words (32 Bits).
   For PLC types that handle data in consecutive bytes:
   Select 1 (8 Bits), 2 (16 Bits) or 4 (32 Bits) Words.
   Tag is used to select the Read address from a list of predefined Tags.
   Aux. directs the Read address to be retrieved from the Auxiliary Port.

3. Fill in **Numeric** Tab items: See previous section on Numeric Input Extend- 13.8.1 Numeric Display Format for details.

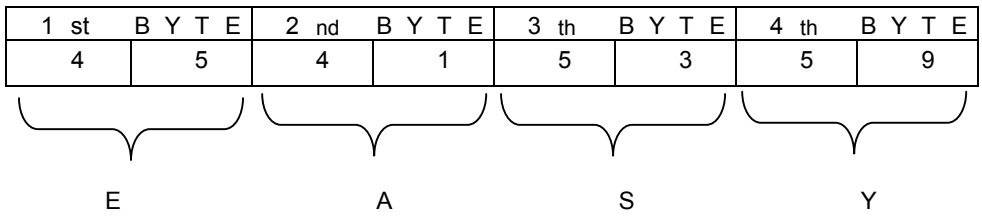
4. Go to **Font** Tab: Fill in attributes of the displayed digits. See Numeric Input Extend - 13.8.2 Font Alignment for details.

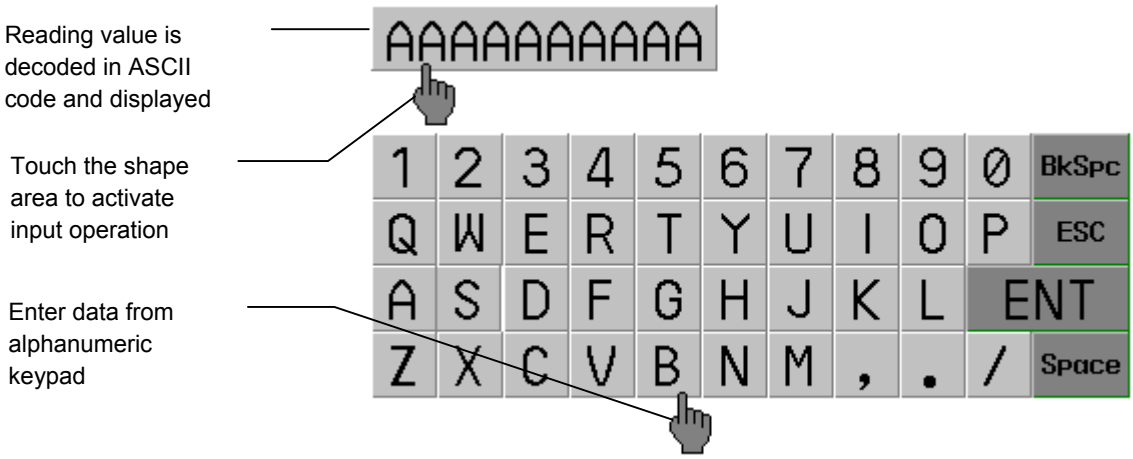
5. Click **OK** to place the Numeric Data part on the window. Position the Numeric Data part as desired.

Note: Refer to Section 2: Software Reference Guide for details about completing each tab item

13.10 ASCII Input Extend

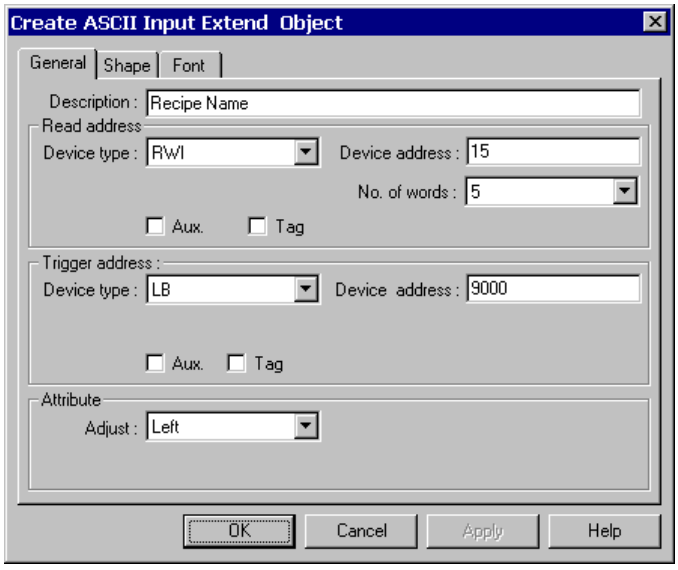
ASCII Input Extend displays current value of the PLC register(s) data as decoded by the standard ASCII character table. When the shape area is touched and the **trigger bit is active**, ASCII Input is available through any displayed alphanumeric keypad. Entered data is put into consecutive PLC registers starting with the “Read address”.





Procedure to place an ASCII Input

Click ASCII Input Extend icon or select **ASCII Input Extend** from the **Parts** menu.



1. Fill in General Tab Items:
   Description: A reference name (not displayed) that you assign to the ASCII Input.

   Read Address: Word or words (up to 16 words may be addressed in this manner, each word contains 2 ASCII characters) in the PLC that is displayed and modified by the ASCII Input.
   - **Device type** is the word prefix.
   - **Device address** is the word number.
   - **No. of words:** Up to 16 consecutive words (32 characters) can be displayed.
   - **Tag** is used to select the Read address from a list of predefined Tags.
   - **Aux.** directs the Read address to be retrieved from the Auxiliary Port.

   Trigger Address: Bit in the Controller that controls the ability to enter data into the Read Address. If bit is OFF, data entry is disabled.

Trigger Address Design Tips:

LB 9000 to LB 9009 are internal bits that are set to 1 at power up. These bits can be used for Trigger addresses so that the ASCII Input Extend parts are always active.

Have the PLC check a register for a password value and use a PLC bit to activate the ASCII Input Extend for added security.

Attribute: Select to Left or Right justify the ASCII characters as they are displayed.

2. Go to **Shape** Tab: Select Shape or Bitmap to be used as a background button. The button is used to identify the part as an input part as opposed to just a regular data display. The button changes state when pressed.

3. Go to **Font** Tab: Fill in the **Color** and **Font** attributes of the displayed data.

4. Click **OK** to place the ASCII Input Extend part on the window. Position the ASCII Input Extend part and resize it if necessary. Adjust the number position as a label.

Note: Refer to Section 2: Software Reference Guide for details about completing each tab item

ASCII Input Extend Design Tips:

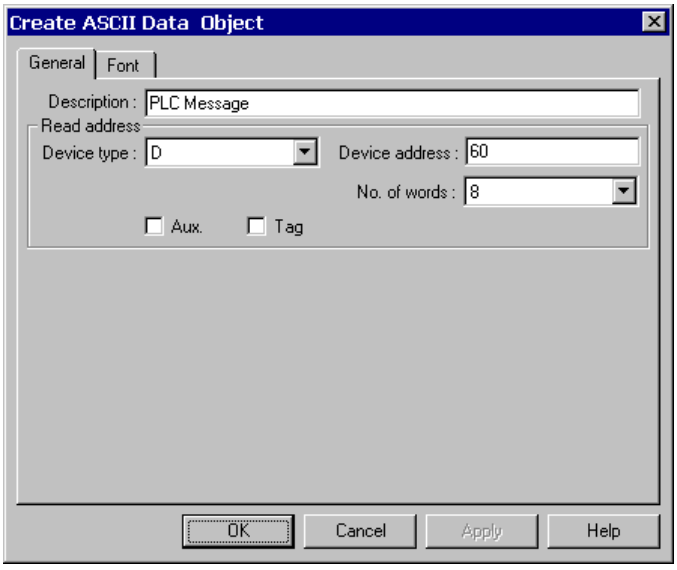
1. Use the ASCII Input Extend part to enter Names for recipes.
2. Create a message area on the screen that the operator can use to leave brief text messages about machine operation.

13.11 ASCII Data

ASCII Data displays the current value of the PLC register data. The data is decoded by standard ASCII characters table. Data changes by the operator are disabled. Reference the ASCII Input Extend section.

Reading value is decoded in ASCII code characters and displayed

Procedure to place an ASCII Data Part



1. Click ASCII Data icon or select **ASCII Data** from the **Parts** menu.

2. Fill in General Tab Items:
   Description: A reference name (not displayed) that you assign to the ASCII Input.
   Read Address: Word or words (up to 16 words may be addressed in this manner, each word contains 2 ASCII characters).
      Device type is the word prefix.
      Device address is the word number.
      No. of words: Up to 16 consecutive words (32 characters) can be displayed.
      Tag is used to select the Read address from a list of predefined Tags.
      Aux. directs the Read address to be retrieved from the Auxiliary Port.

3. Go to **Font** Tab: Fill in the **Color** and **Font** attributes of the displayed data.

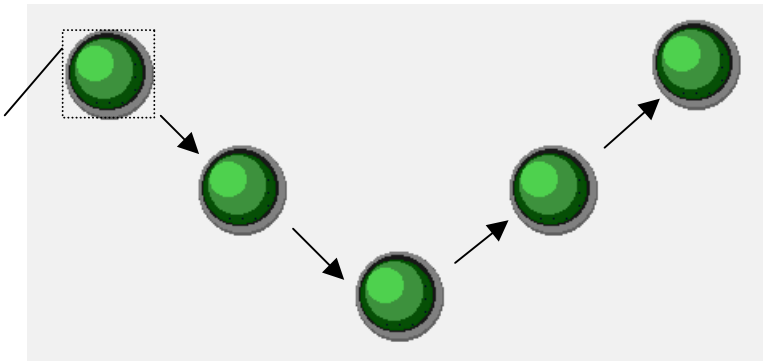
4. Click **OK** to place the Numeric Data Part on the window. Position the Numeric Data Part as desired.

ASCII Data Design Tip:
Use the ASCII Data Part to display messages, units of measure, etc. that are stored in the PLC.

13.12 Moving Shape

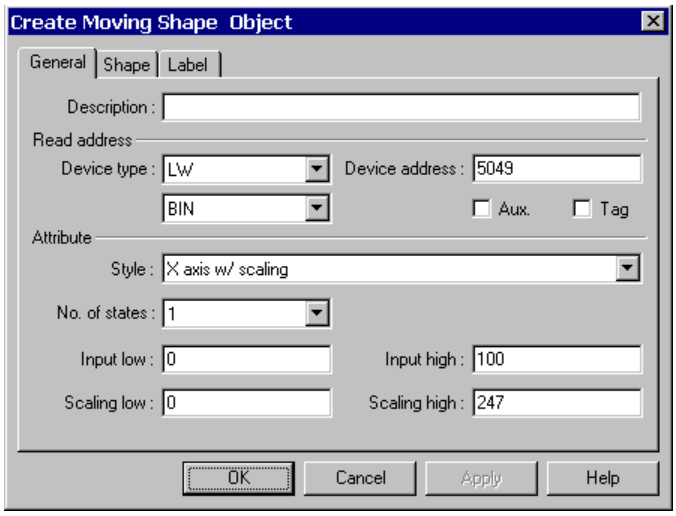
The Moving Shape tool is used to place an object in a window at a location specified by the PLC. The state and the absolute location of the shape in the window depend on the current values of three continuous PLC registers. Typically, the first register controls the state of the object the second the horizontal position (X), and the third the vertical position (Y).

X & Y position is controlled by PLC



Procedure to place a Moving Shape

1. Click Moving Shape icon or select **Moving Shape** from the **Parts** menu.



2. Fill in General Tab Items:
   Description: A reference name (not displayed) that you assign to the Moving Shape.

   Read Address: Word in the PLC that controls the Moving Shape state, position and label.
   - **Device type** is the word prefix.
   - **Device address** is the word number.
   - **Data format:** Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**). See Word Lamp 13.2 for format description.
   - **Tag** is used to select the Read address from a list of predefined Tags.
   - **Aux.** directs the Read address to be retrieved from the Auxiliary Port.

   Attribute: Style: See ***Note:*** Attribute Functions below.

3. Go to **Shape** Tab: Select Shape or Bitmap to move and display states.

4. Go to **Label** Tab: Fill in fields to denote states, if desired.

5. Click **OK** to place the Moving Shape on the window. Position the Moving Shape (during operation, initial position depends on register values) and resize it if necessary. Adjust the label position as desired.

Note: Refer to Section 2: Software Reference Guide for details about completing each tab item

Moving Shape Design Tips:

1. Use a moving shape as a pen for a trend chart. Set it so that the pen shape corresponds to the trend chart input. Change the state of the pen shape to change from green to red when a high or low limit is exceeded.

2. Use a moving shape object as a pointer to a setpoint on a bargraph. Set it so that it corresponds to the one of the variable set points. As the operator changes the setpoint, the pointer moves to a new position.

Note: Attribute Functions

No. of states: Up to 32 different states can be assigned to the Moving Shape Part. For example: If the moving shape was a needle pointer, you could have it change to different states that display it as different colors.

Style: Position is controlled by the Read Address Device as shown in the table below.

Control by PLC	X axis (all types)	Y axis (all types)	X & Y axis
Read Address	Shape or Bitmap State	Shape or Bitmap State	Shape or Bitmap State
Read Address+1	X position	Y position	X position
Read Address+2	Not Used	Not Used	Y position

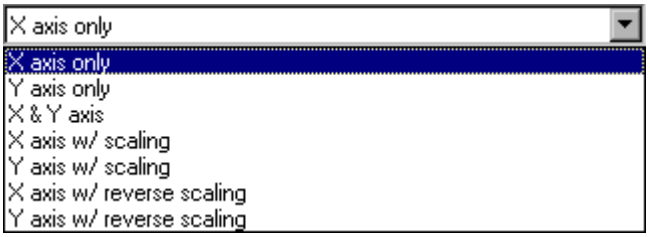
When **scaling** is used, scaling is performed as follows:

Display value = (reading value – **Input low**) * (**Scaling high** – **Scaling low**)
(**Input high** – **Input low**)

When **reverse scaling** is used, scaling is performed as follows:

Display value = (**Input high** – reading value) * (**Scaling high** – **Scaling low**)
(**Input high** – **Input low**)

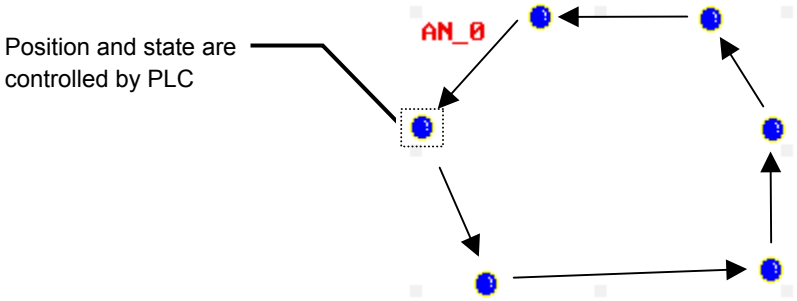
Possible selections:



Selection	Description
X axis only	Shape moves Left (0) to Right (Max. screen resolution 320 or 640)
Y axis only	Shape moves Top (0) to Bottom (Max. screen resolution 240 or 480)
X & Y axis	Shape moves Top Left (0) to Bottom Right (Max. screen resolution 240, 320 or 480, 640)
X axis w/ scaling	Shape moves Left (0) to Right (Max. scaled screen resolution 320 or 640)
Y axis w/ scaling	Shape moves Top (0) to Bottom (Max. scaled screen resolution 240 or 480)
X axis w/ reverse scaling	Shape moves Right (0) to Left (Max. scaled screen resolution 320 or 640)
Y axis w/ reverse scaling	Shape moves Bottom (0) to Top (Max. scaled screen resolution 240 or 480)

13.13 Animation

The Animation Part is used to place an object on the screen at a specified location determined by a predefined path and data in the PLC. The state and the absolute location of the shape on the screen depend on current reading value of two continuous PLC registers. Typically, the first register controls the state of the object the second controls the position along the predefined path. As the PLC position register changes value the shape or bitmap jumps to the next position along the path.

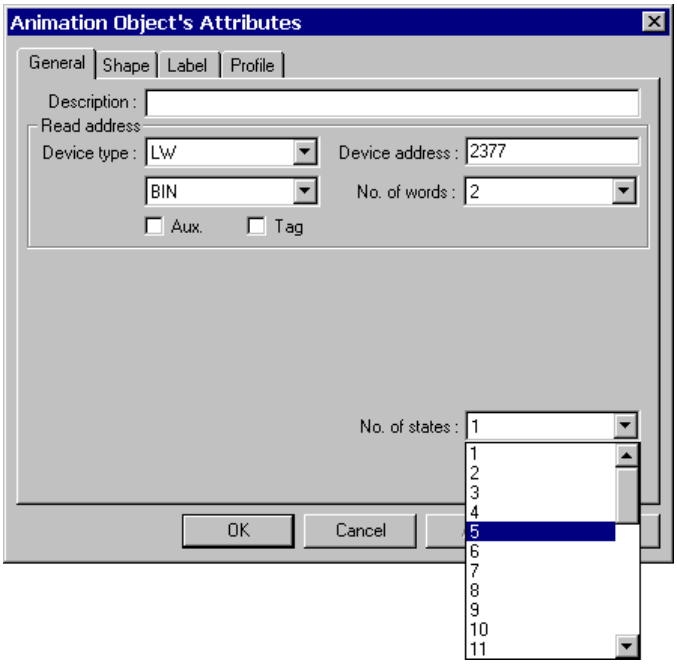


Procedure to place an animation

1. Click Animation icon or select **Moving Shape** from the **Parts** menu.

2. Click on screen to define the path (stopping points) for the object to travel. Right-click the mouse to end the path.

3. Click the Edit Attributes Tool and fill in General Tab Items:



Description: A reference name (not displayed) that you assign to the Animation.

Read Address: Word in the PLC that controls the Animation state, position and label.
- **Device type** is the word prefix.
- **Device address** is the word number.
- **Data format:** Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**). See Word Lamp 13.2 for format description.
- **No. of words** is restricted to 2 Words for this part.
- **Tag** is used to select the Read address from a list of predefined Tags.
- **Aux.** directs the Read address to be retrieved from the Auxiliary Port.

No. of states: Select the total number of states for the object. This determines what state of the shape or bitmap is displayable at stops along the path. (Range: 1 to 32)

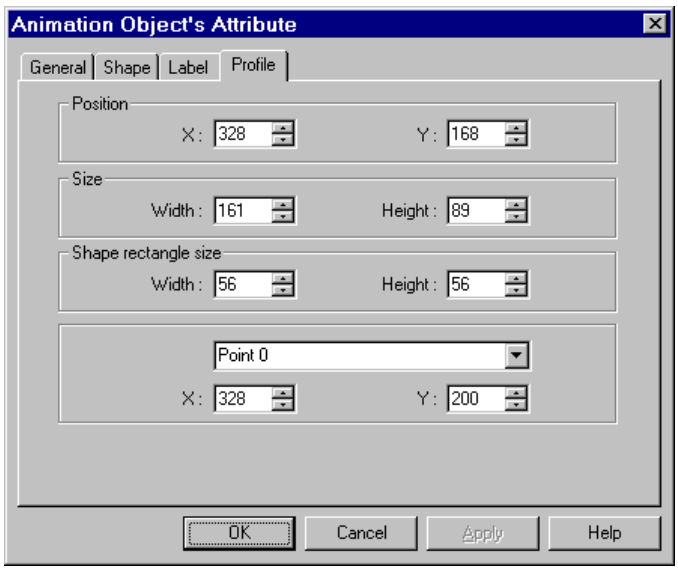
4. Go to **Shape** Tab: Select Shape or Bitmap to move and display states.

5. Go to **Label** Tab: Fill in text to denote states, if desired.

6. Click **Ok** to position the Animation part on the window. Resize the overall path area if necessary. During operation, initial position depends on the register value.

Note: Refer to Section 2: Software Reference Guide for details about completing each tab item

Note: How to edit point location



Go to the Profile tab:
The Profile tab settings are used to numerically set the following:

Position: The location of the upper left-hand corner of the Animation area.

Size: The dimensional area on the screen that the outside edge of the path occupies.

Shape Rect. Size: The area of the shape or bitmap that is used in the Animation.

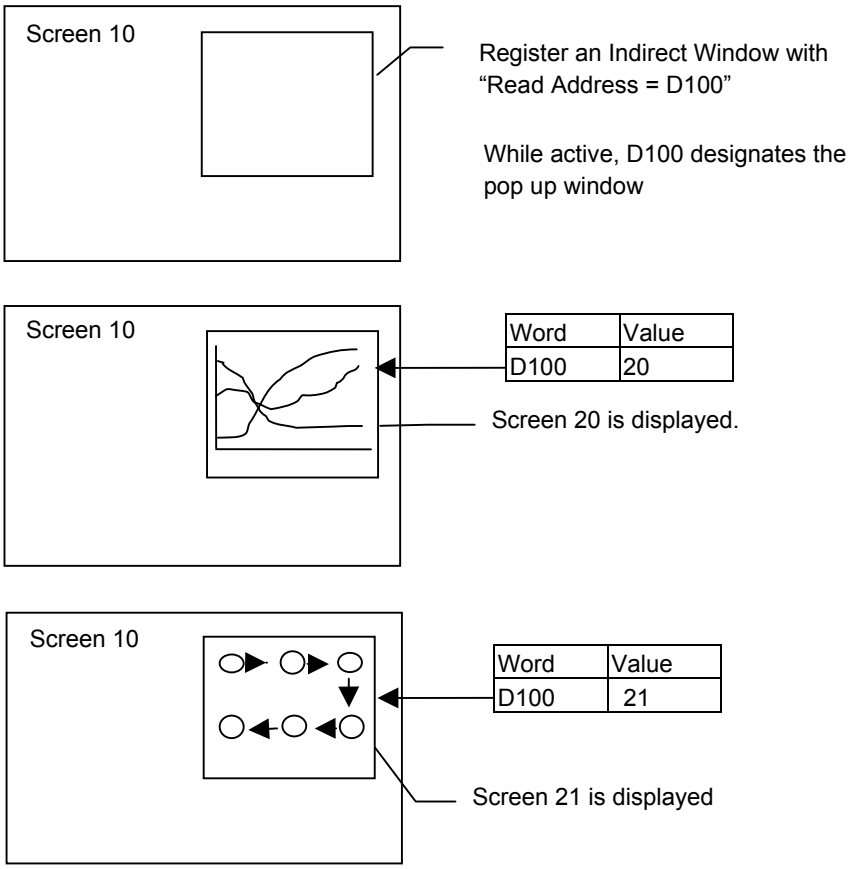
Points can be repositioned by selecting the point number for the location and adjusting the X and Y coordinates.

Moving Shape Design Tip: Use the Animation tool along with a pointing shape/bitmap to guide a user through multiple touch operations. After every touch, increment the word after the Read address.

Note: Remember, incrementing the Read address changes the state, not the position.

13.14 Indirect Window

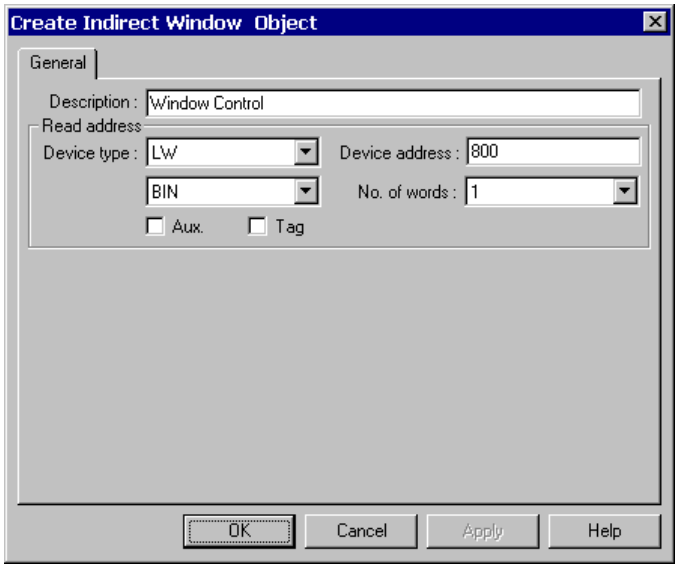
The Indirect Window places a defined child window area over the current window. The Indirect Window's screens are created the same as regular base windows. Usually their size is smaller than a full window. The Windows are then displayed in the Indirect Window frame as called by a PLC data register. The limitation to the maximum registered Indirect Windows on each screen is 500. However, at run time, only 5 windows at the most can be displayed simultaneously on each screen.



Note: Only valid window numbers are displayed. If the window number is not defined then the Indirect Window area is left transparent.

Note: Indirect Windows disable Popup function keys placed on the same screen. A popup window can only be triggered by one method on any given window.

Procedure to place an Indirect Window



1. Click the Indirect Window Tool or select **Indirect Window** from the **Parts** menu.

2. Fill in General Tab Items:

Description: A reference name (not displayed) that you assign to the Indirect Window.

Read Address: Word in the PLC that determines which window is displayed in the Indirect Window area.

Device type is the word prefix.
Device address is the word number.
Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
See Word Lamp 13.2 for format description.
No. of words is restricted to 2 Words for this part.
Tag is used to select the Read address from a list of predefined Tags.
Aux. directs the Read address to be retrieved from the Auxiliary Port.

3. Click **OK** to position the Indirect Window and resize it if necessary. Resize to fit the dimensions exactly by using the **Profile** tab settings.

Note: The size of the window limits the area of display. The window area outside of the Indirect window's is clipped.

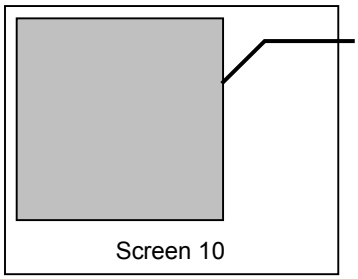
Indirect Window Design Tip:
Function Button parts "**Window Bar**" and "**Minimize Window**" can be assigned to the windows that are popped up to enable repositioning and minimizing features. Refer to "**Window Bar**" and "**Minimize Window**" in the Function Key Part section.

Use an Indirect Window to display consecutive Help screens.

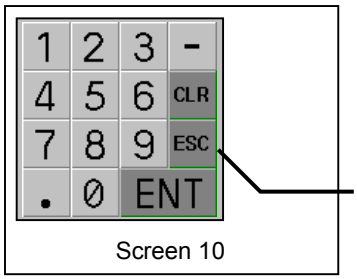
13.15 Direct Window

The Direct Window places a defined child window area over the current window. The displayed window number is preset in the Direct Window. The window for display is a regular base window.

Note: The maximum number of Direct Windows for a screen is 500 (Object limit). However, at run time, a maximum of six windows can be displayed simultaneously. This number may be reduced further depending on System Parameter settings.



Define a Direct window area with
"Read Address = M10"
and
"Window Number = 100"

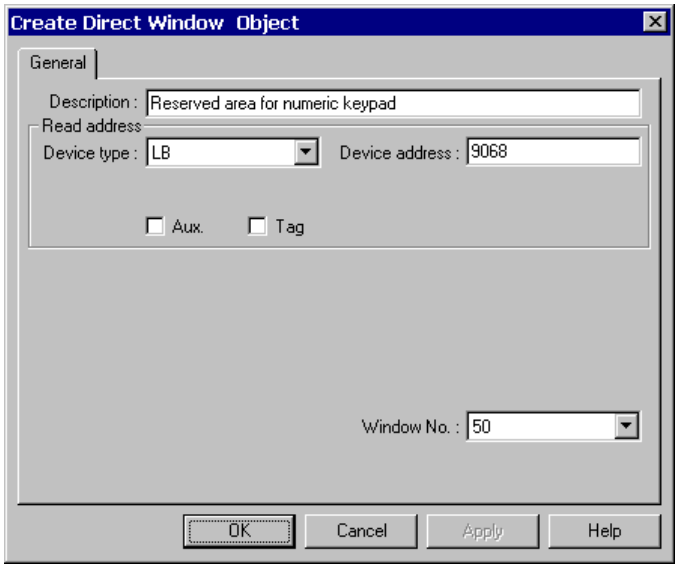


At run time when M10 = 100 the pop up window appears.

The pop up keypad is window 100.

Procedure to place a Direct Window

1. Click the Direct Window Tool or select **Direct Window** from the **Parts** menu.



2. Fill in General Tab Items:
   Description: A reference name (not displayed) that you assign to the Direct Window.

   Read Address: Bit in the PLC that calls the Direct Window.
   - **Device type** is the word prefix.
   - **Device address** is the word number.
   - **Tag** is used to select the Read address from a list of predefined Tags.
   - **Aux.** directs the Read address to be retrieved from the Auxiliary Port.

   Window No.: The base window the Direct Window displays. Only valid window numbers are shown in the drop down list.

3. Click **OK** to position the Direct Window where you want it to pop-up and resize it if necessary.

The size of the window limits the area of display. The window area outside of the window boundary is clipped. Function Button parts "**Window Bar**" and "**Minimize Window**" can be assigned to the window to enable repositioning and minimizing.

Note: Direct Windows disable Popup function keys for the same window number placed on the same screen. A window can only be triggered by one method.

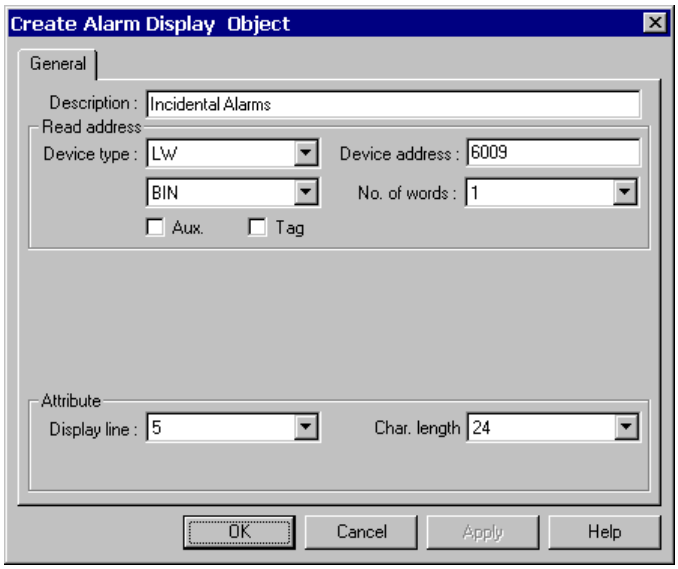
13.16 Alarm Display

The Alarm Display opens a window to display messages registered in the **Alarm Scan** list via alarm trigger conditions. All active alarm messages are sorted by time. As new alarms are displayed on the top line, older ones are scrolled down.

Note: Refer to the Alarm Scan section for more details about alarm messages.

Procedure to add Alarm Display

1. Click Alarm Display icon or select **Alarm Display** from the **Parts** menu.



2. Fill in General Tab Items:
   Description: A reference name (not displayed) that you assign to the Alarm Display.

   Read Address: The Read Address controls the scrolling, up and down, of the alarm display window. It has nothing to do with triggering alarms.
   Device type is the word prefix.
   Device address is the word number.
   Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
   See Word Lamp 13.2 for format description.
   No. of words is restricted to 2 Words for this part.
   Tag is used to select the Read address from a list of predefined Tags.
   Aux. directs the Read address to be retrieved from the Auxiliary Port.

   Attribute:
   Display line: Specifies the height of the window in Font 16 high lines. (Font size 24 takes up two lines)
   Char. length: Specifies the width of the window in Font 16 characters, extra characters past this length are truncated.

   Note: Font 16 characters are 8 pixels wide by 16 pixels high.

3. Click **OK** to place and position the Alarm Display where desired.

Note: Resize an Alarm Display part by adjusting the **Display Line** and **Char. Length** attributes.

Note: Refer to Section 2: Software Reference Guide for details about completing each tab item

Example of an Alarm Display

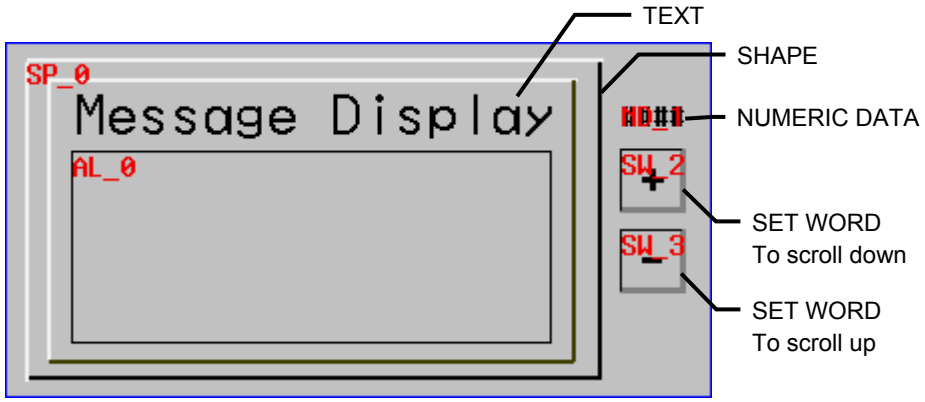
• A Shape (SP_0) is used for the background rectangle. In this example, it is Button 0 from the Button1 shape library.

• Some TEXT is placed on the Shape to identify what is being displayed. (Shown as “Message Display” in the example)

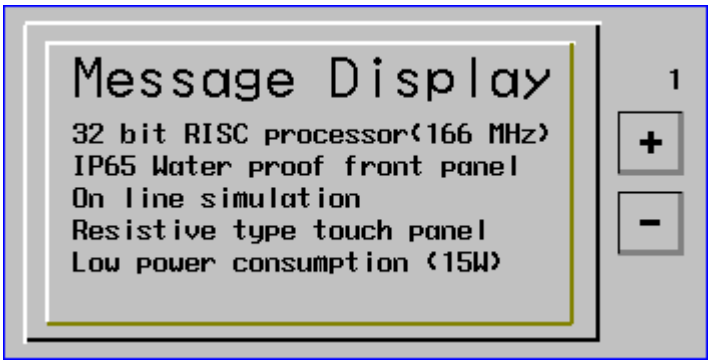
• Two Set Word parts are used to increment and decrement the alarm register. (Buttons with plus and minus signs)

• A Numeric Data part is used to display the topmost active alarm message number.

• The actual Alarm Display part is placed on top of the shape. The alarm messages appear in the Alarm Display.



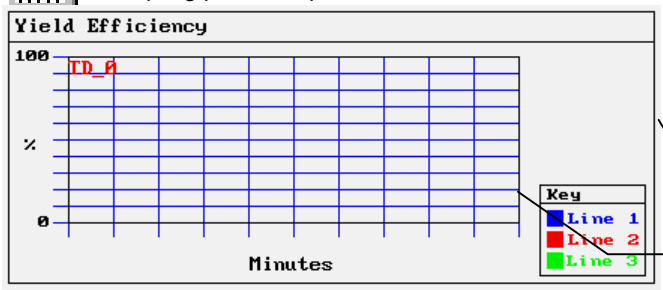
Example Alarm Display in design phase.

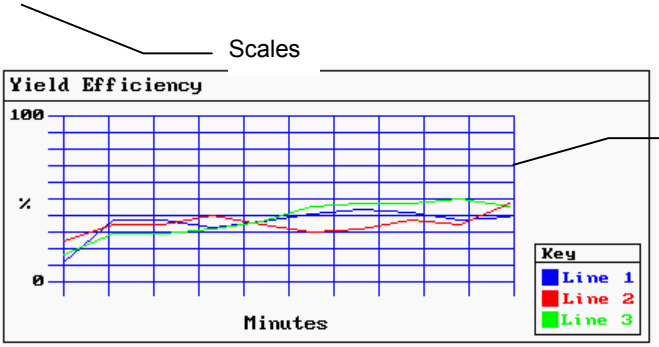


Example Alarm Display showing some Alarm Scan messages

13.17 Trend Display

The Trend Display periodically retrieves a block of PLC data and displays the trend data over time. As each sampling period elapses, the new data is read from the PLC and inserted towards the right side of the trend graph, the trend graph, when full is shifted to the left direction. The Trend Display data is retrieved on a real time basis.





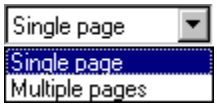
An example of a typical trend display is shown to the right. A Shape is used for the background and Scales are added to show relative information about the trend. The trend display is then placed on the Shape.

Procedure to add a Single page Trend Display

1. Click Trend Display icon or select **Trend Display** from the **Parts** menu.

2. Fill in General Tab Items:

Page type: Select Single page. Single page is a simple trend display. Data is plotted as described in the above example.

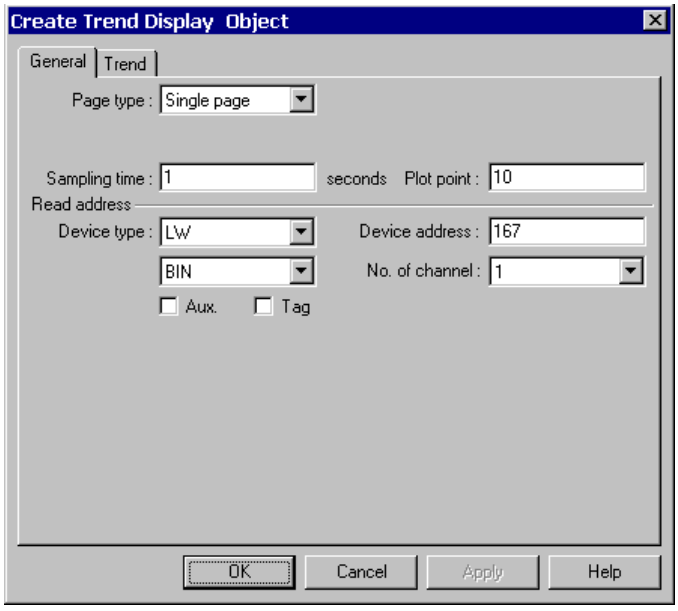


Sampling time: Time between point plots in seconds.

Plot point: the number of sample points displayed across the length of the chart.

Read address: Specify the PLC word address of the first Trend data pen, the second Trend data pen starts at read address + 1, and the third starts at read address + 2, etc.

Device type is the word prefix.
Device address is the word number.
Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
See Word Lamp 13.2 for format description.
No. of Channel: The number of PLC data words retrieved is the same as No. of Channel, one word per channel. Up to 16 channels can be specified for a trend.
Tag is used to select the Read address from a list of predefined Tags.
Aux. directs the Read address to be retrieved from the Auxiliary Port.



Trend Display Design Tip: Too many pens can make a trend cluttered and unreadable.

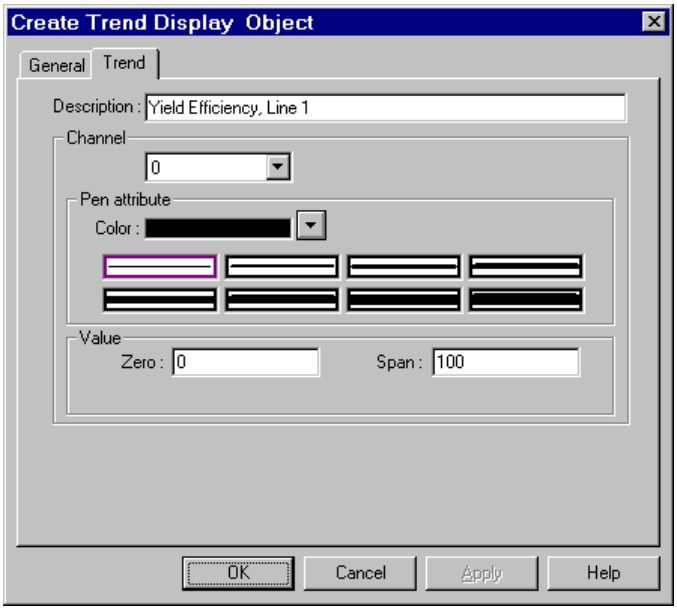
3. Fill in the **Trend** Tab: Select the “Channel” to view each channel’s settings.
4.

Description: A reference name (not displayed) that you assign to the Trend Display.

Pen attribute: Specify the Trend Color and pen thickness for the channel.

Value: Set the zero and span for each channel.

4. Click **OK** to position and resize the Trend Display

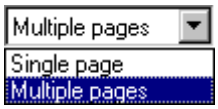


Procedure to add a Multiple pages Trend Display

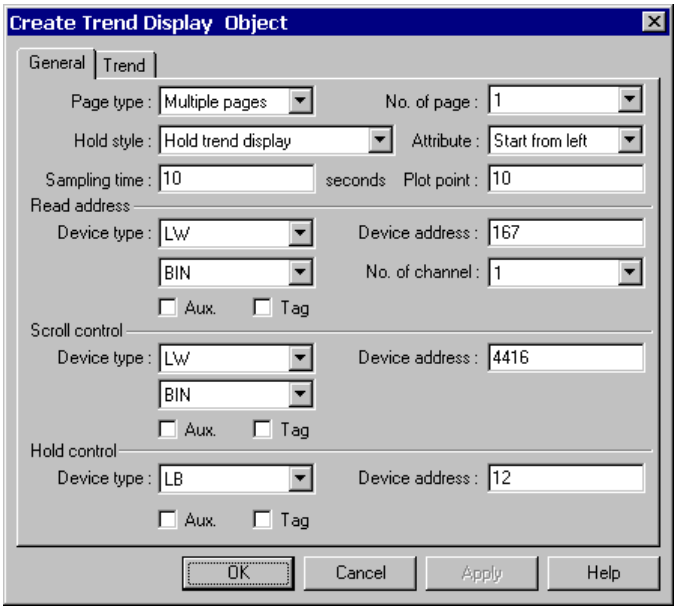
1. Click Trend Display icon or select **Trend Display** from the **Parts** menu.

2. Fill in General Tab Items:

Page type: Select **Multiple pages**. Multiple pages allows a trend display to be extended. Data is plotted as before but it is not lost after it is scrolled off the trend.



Hold style: Determines how the trend reacts when the hold bit is activated. Hold trend display simply prevents further updates until the Hold bit is turned off. Hold trend display & clear stops the trend update and clears out all pages of the trend.



No. of page: The data is held in memory for review up to the number of pages (31 maximum) specified by the dropdown. Each page is the number of sample points specified in the Plot point field. The Scroll control is used to recall plots.

Attribute: Start from left selects to have the trend pens start from the left and traverse the display before scrolling begins. Start from the right begins the pens scrolling from right to left from the first sample onwards.

Sampling time: Time between point plots in seconds.

Plot point: The number of sample points displayed across the length of the Trend Display part.

Read address: Specify the PLC word address of the first trend data pen, the second trend data pen starts at read address + 1, and the third starts at read address + 2, etc.

Device type is the word prefix.
Device address is the word number.
Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
See Word Lamp 13.2 for format description.
No. of Channel: The number of PLC data words retrieved is the same as No. of Channel, one word per channel. Up to 16 channels can be specified for a trend.
Tag is used to select the Read address from a list of predefined Tags.
Aux. directs the Read address to be retrieved from the Auxiliary Port.

Scroll control: The address of the register that determines which portion of the multiple page display is being shown on the screen. Each increment of the value in the scroll register moves the Trend display one sample plot to the left. For example: If the plot was 10 Plot points wide, the Scroll control needs to be incremented by 10 to scroll back and forth in full pages.

Device type is the word prefix.
Device address is the word number.
Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
See Word Lamp 13.2 for format description.
Tag is used to select the Read address from a list of predefined Tags.
Aux. directs the Read address to be retrieved from the Auxiliary Port.

Hold control: The bit that controls the trend update. See Hold style above.

Device type is the word prefix.
Device address is the word number.
Tag is used to select the Read address from a list of predefined Tags.
Aux. directs the Read address to be retrieved from the Auxiliary Port.

3. Fill in the **Trend** Tab: Select the “Channel” to view each channel’s settings.

Description: A reference name (not displayed) that you assign to the Trend Display.

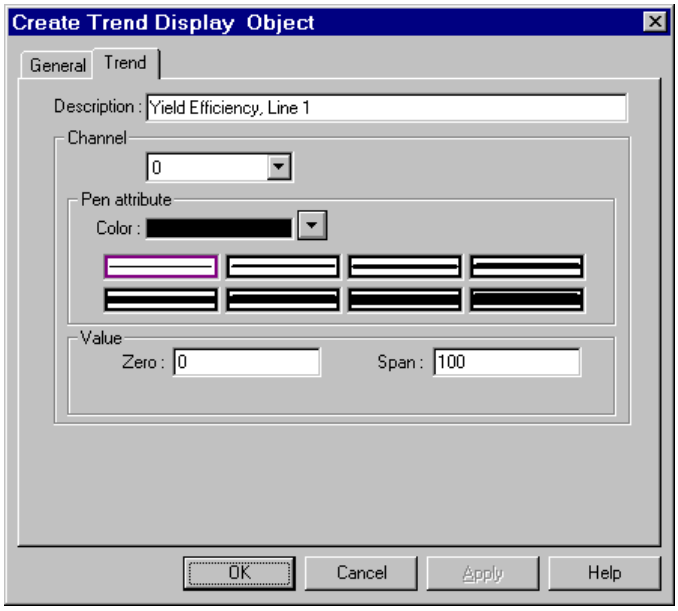
Pen attribute: Specify the Trend Color and pen thickness for the channel.

Value: Set the zero and span for each channel.

4. Click **OK** to position and resize the Trend Display.

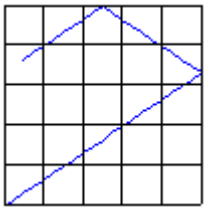
5. Add Scales, Drawing objects, and Text to give the Trend a finished look. (See example above).

Note: Refer to Section 2: Software Reference Guide for details about completing each tab item



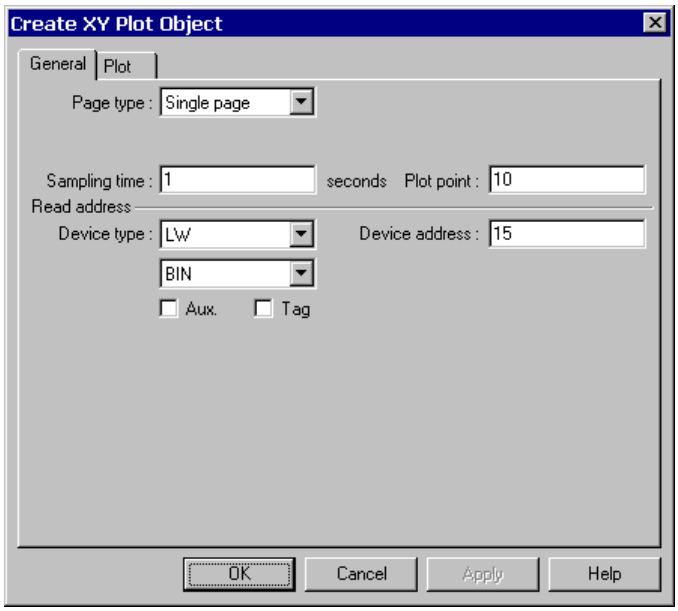
13.18 XY Plot

The XY Plot displays word data as a line plotted out on x and y coordinates The XY Plot periodically retrieves a block of PLC data and plots the data out on x and y coordinates over time. As each sampling period elapses, the new data is read from the PLC and plotted. When number of plotted points equals plot points, the first points start deleting. The XY Plot is refreshed on a real time basis.



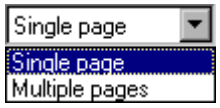
Procedure to add a Single page XY Plot

1. Click the XY Plot tool or select **XY Plot** from the **Parts** menu.



2. Fill in General Tab Items:

Page type: Select Single page. Single page is a simple XY Plot display. Data is plotted as described in above.



Sampling time: Time between point plots in seconds.

Plot point: The maximum number of sample points displayed on the plot (10 to 255).

Read address: Specify the PLC word address of the X plot data pen. The Y plot pen is Read address +1.

Device type is the word prefix.
Device address is the word number.
Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
See Word Lamp 13.2 for format description.
Tag is used to select the Read address from a list of predefined Tags.
Aux. directs the Read address to be retrieved from the Auxiliary Port.

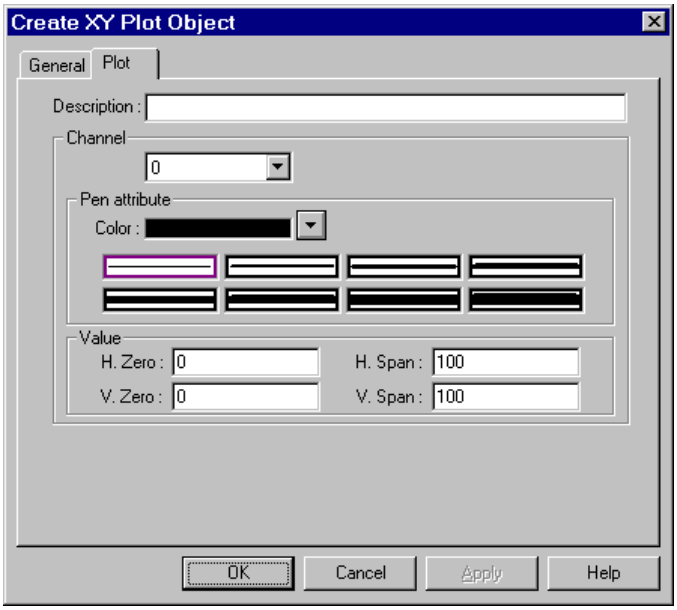
3. Fill in the **Plot** Tab: Only **Channel** 0 is available.

Description: A reference name (not displayed) that you assign to the XY Plot.

Pen attribute: Specify the pen **Color** and thickness for the channel.

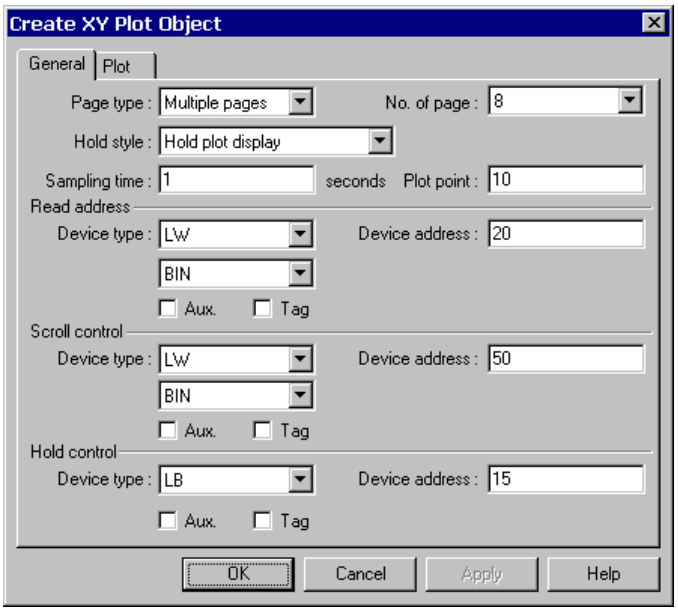
Value: Set the **Zero** and **Span** for the Horizontal (H) and Vertical (V) axis.

4. Click **OK** to position and resize the XY Plot.

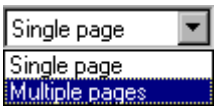


Procedure to add a Multiple pages XY Plot

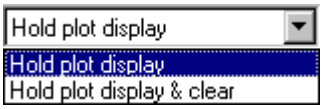
1. Click the XY Plot tool or select **XY Plot** from the **Parts** menu.



2. Fill in General Tab Items:



Page type: Select **Multiple pages**. Multiple pages allows a plot display to be extended. Data is plotted as before but it is not lost as points are erased off the plot.



Hold style: Determines how the plot reacts when the hold bit is activated. **Hold plot display** simply prevents further updates until the Hold bit is turned off. **Hold plot display & clear** stops the plot update and clears out all plots in memory.

Sampling time: Time between point plots in seconds.

No. of page: The data is held in memory for review up to the number of pages (31 maximum) specified by the dropdown. Each page is the number of sample points specified in the Plot point field. The Scroll control is used to recall plots.

Plot point: The maximum number of sample points displayed on the plot (10 to 255).

Read address: Specify the PLC word address of the X plot data pen. The Y plot pen is Read address +1.
- **Device type** is the word prefix.
- **Device address** is the word number.
- **Data format:** Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
  See Word Lamp 13.2 for format description.
- **Tag** is used to select the Read address from a list of predefined Tags.
- **Aux.** directs the Read address to be retrieved from the Auxiliary Port.

Scroll control: The address of the register that determines which portion of the multiple page display is being shown on the screen. Each increment of the value in the scroll register moves the plot display one sample plot previous.
- **Device type** is the word prefix.
- **Device address** is the word number.
- **Data format:** Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
  See Word Lamp 13.2 for format description.
- **Tag** is used to select the Read address from a list of predefined Tags.
- **Aux.** directs the Read address to be retrieved from the Auxiliary Port.

Hold control: The bit that controls the plot update. See Hold style above.
- **Device type** is the bit prefix.
- **Device address** is the bit number.
- **Tag** is used to select the Read address from a list of predefined Tags.
- **Aux.** directs the Read address to be retrieved from the Auxiliary Port.

3. Fill in the **Plot** Tab the same way as for Single Page plots. Refer to previous page.

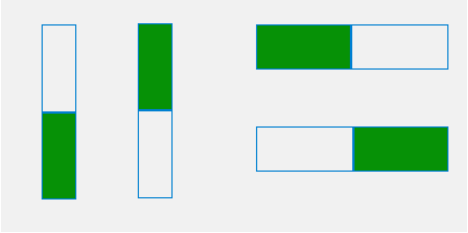
4. Click **OK** to position and resize the XY Plot.

5. Add Scales, Drawing objects, and Text to give the Plot a finished look.

Note: Refer to Section 2: Software Reference Guide for details about completing each tab item

13.19 Bar Graph

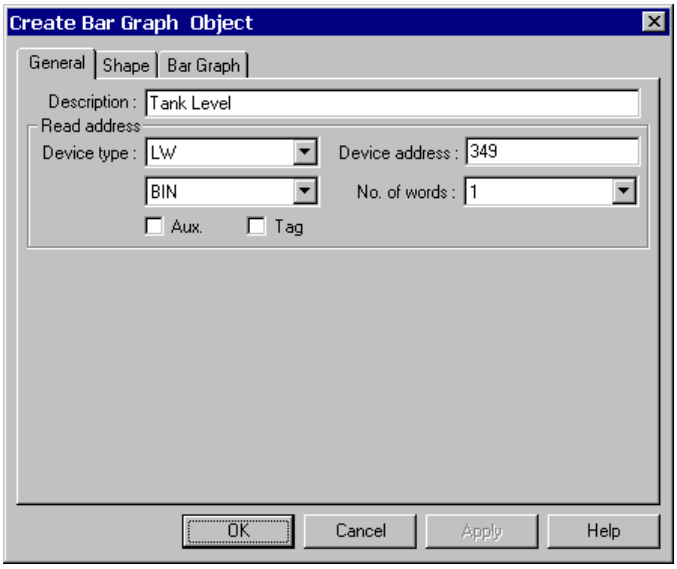
The Bar Graph displays PLC register data as a bar graph in proportion to its value as defined by the SPAN and ZERO settings.



Procedure to place a Bar Graph

1. Click the Bar Graph Tool or select **Bar Graph** from the **Parts** menu.

2. Fill in **General** Tab Items:



Description: A reference name (not displayed) that you assign to the Bar Graph.

Read Address: First Word of 3 possible that are used to control the Bar Graph display.

Device type is the word prefix.

Device address is the word number.

Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
See Word Lamp 13.2 for format description.

No. of words: Restricted to 1 or 3. Three words are needed when variable alarm is selected. The Bar Graph continuously retrieves 3 data words, one for data and one each for high and low alarm limits. One word is needed when fixed alarms are selected.

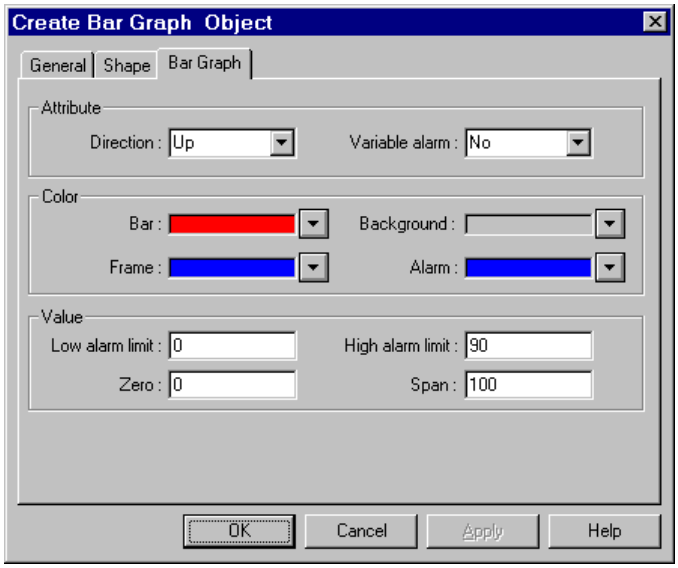
Tag is used to select the Read address from a list of predefined Tags.

Aux. directs the Read address to be retrieved from the Auxiliary Port.

3. Fill in **Shape** Tab Items. Bitmaps with transparent portions can be used to make irregular bargraphs such as tanks with cut away views.

4. Go to **Bar Graph** Tab and make settings.

Attribute:



Direction: Up, Down, Left and Right. Select the direction for the bargraph to travel.

Variable alarm:

Yes: The high and low alarm limits are retrieved from PLC data registers as described below.

No: The high and low alarm limits are set in the “Value” fields.

Variable Alarm	YES	NO
Read Address	Bar graph data	Bar graph data
Read Address+1	Low alarm	--
Read Address+2	High alarm	--

Color: Set **Bar**, **Background**, **Frame** and **Alarm** bar colors.

Value:

Low/High alarm limit: if “Variable alarm is No”, the high and low alarm limits are entered here.

Enter **Zero** and **Span**. The filled bar percentage is calculated as follows:

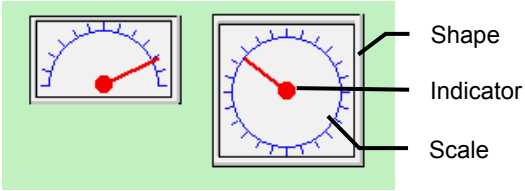
$$\% \text{ of filled bar} = \frac{(\text{Register value} - \text{Zero})}{(\text{Span} - \text{Zero})} * 100\%$$

5. Click **OK** to position the Bar Graph and resize it if necessary.

Note: Refer to Section 2: Software Reference Guide for details about completing each tab item

13.20 Meter Display

The Meter displays PLC register data as an angular indicator in proportion to its value as defined by the SPAN and ZERO setting.



A traditional meter may be composed of three types of parts

Procedure to place a Meter

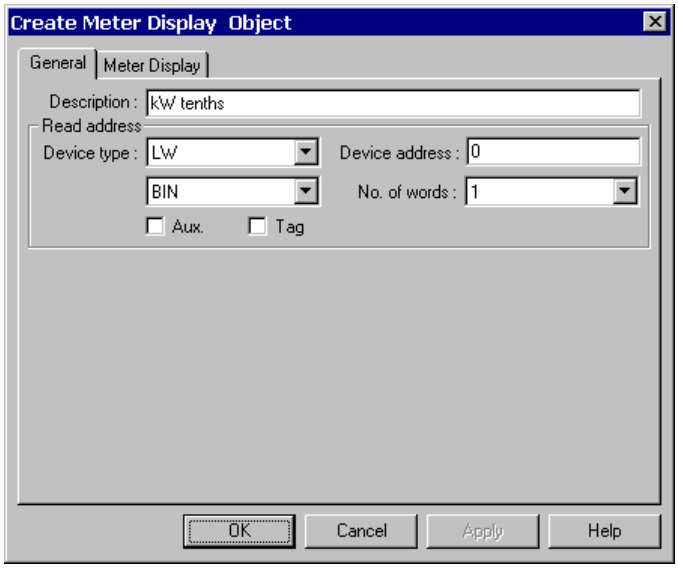
1. Click the Meter Tool or select **Meter Display** from the **Parts** menu.

2. Fill in General Tab Items:

Description: A reference name (not displayed) that you assign to the Meter

Read Address: PLC word that used to control the displayed value.

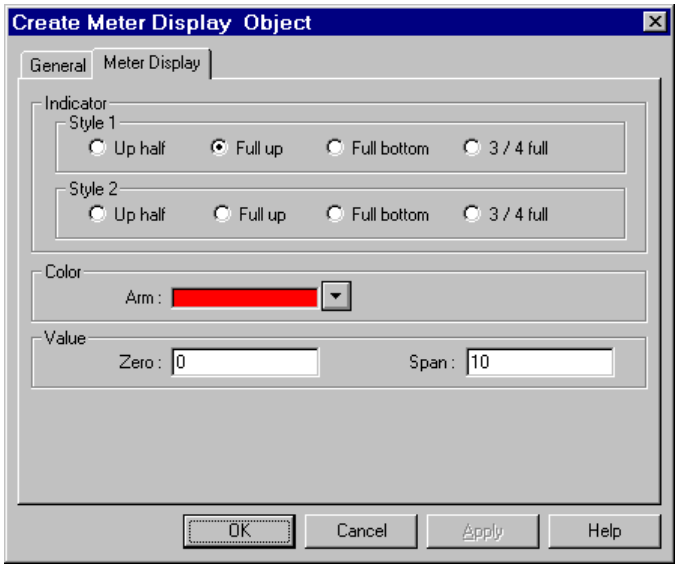
Device type is the word prefix.
Device address is the word number.
Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
See Word Lamp 13.2 for format description.
No. of words: Restricted to 1 for this part.
Tag is used to select the Read address from a list of predefined Tags.
Aux. directs the Read address to be retrieved from the Auxiliary Port.



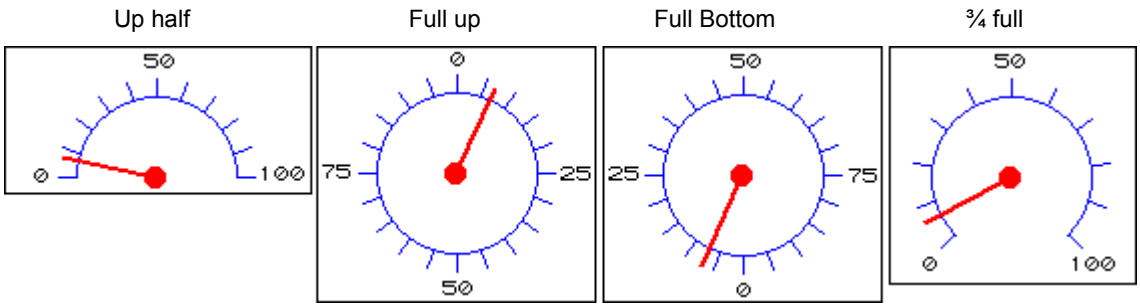
3. Go to **Meter Display** make selections and fill in fields as desired.

Indicator:
There are two Indicator Styles available; straight arm and vane pointer.

Style 1 Style 2



Attribute:



Attribute	Description
Up half	Topmost 180 degrees of semicircle, meter moves left to right clockwise
Full up	Full circle, meter moves clockwise starting from 12 o'clock position
Full bottom	Full circle, meter moves clockwise starting from 6 o'clock position
¾ full	Topmost 270 degrees of semicircle, meter moves left to right clockwise

Percentage of angle ∠ (relative to 180° for Half circle and 360° for Full circle) = $\frac{\text{(Register value – Zero)}}{\text{(Span – Zero)}} * 100\%$

Color: set indicator color.

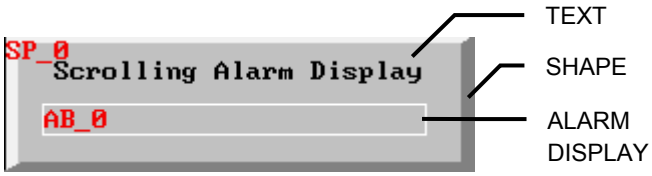
4. Click **OK** to position the Meter and resize it if necessary.

Note: Refer to Section 2: Software Reference Guide for details about completing each tab item

13.21 Alarm Bar

The Alarm Bar sets aside a window to display messages registered in the Alarm Scan list via alarm conditions. The messages are scrolled in from right to left in order of triggering. Alarms continue to scroll until they switch to the inactive state.

Example of an Alarm Bar:
• A Shape (SP_0) is used for the background rectangle.
• Some TEXT is placed on the Shape to identify what is being displayed.
• The actual Alarm Bar part is placed on top of the shape.

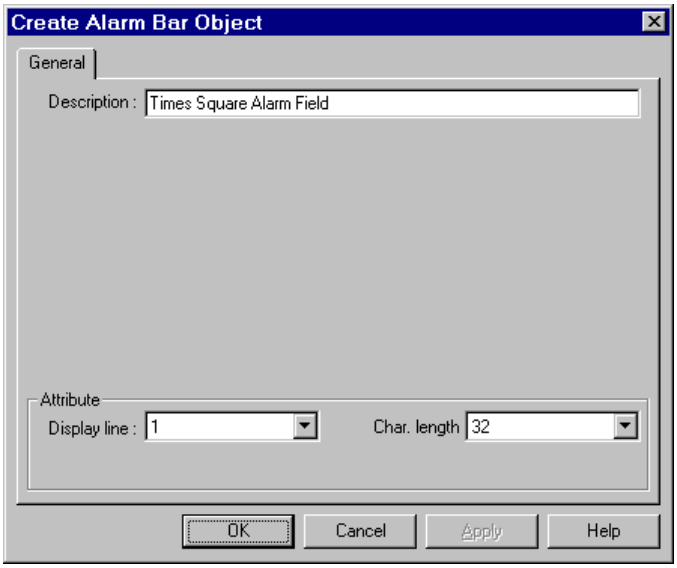


The alarm messages appear in the Alarm Bar.

Display at run time scrolling an alarm.

Procedure to add Alarm Bar

1. Click the Alarm Bar Tool or select **Alarm Bar** from the **Parts** menu.



2. Fill in General Tab Items:
   Description: A reference name (not displayed) that you assign to the Alarm Bar.

   Display line: Specifies the height of the window, in lines, at one message per line. All messages are displayed as size 16 or 24 font. Two lines are required for size 24 font.

   Char. length: Specifies the width of the window in 16 point characters. Due to scrolling, messages are never truncated as in the Alarm Display.

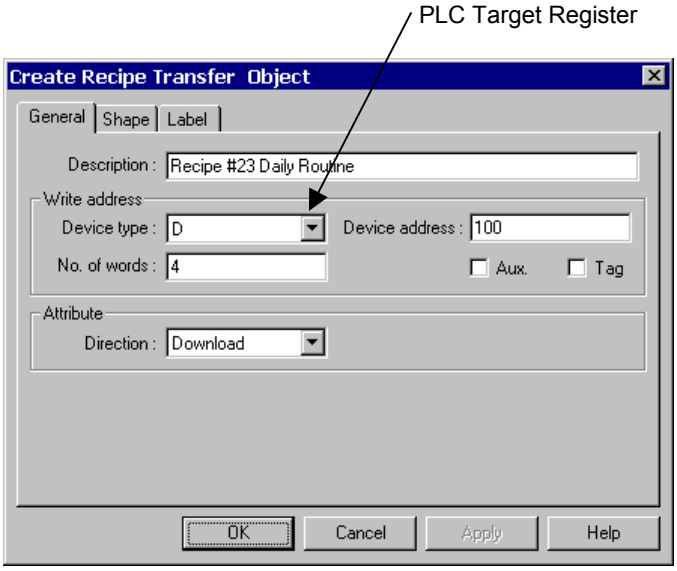
3. Click **OK** to place and position the Alarm Bar where desired.
   Note: Resize an Alarm Bar part by adjusting the **Display Line** and **Char. Length** attributes.

Design Tips:
Put an Alarm Bar object where the Task Bar is located. When the task bar is extended, it hides the bar. When it is contracted, the bar can be seen. This way, the operator can check for messages as desired and hide annoying messages. Put the alarm bar in the common window so it is always available.

13.22 Recipe Transfer

The Recipe Transfer part activates the transfer of a block of contiguous registers from the HMI to the controller or from the controller to the MMI. HMI storage address is determined by an internal word (See special note below).

Procedure to create a Recipe Transfer Part.



1. Click the Recipe Transfer Tool or select **Recipe Transfer** from the **Parts** menu.

2. Fill in General Tab Items:
   Description: A reference name (not displayed) that you assign to the Recipe Transfer.

   Write Address: Word that begins the block of registers to write or receive upload from the PLC.
   Device type is the word prefix.
   Device address is the word number.
   No. of words is how many registers are transferred.
   Tag is used to select the Read address from a list of predefined Tags.
   Aux. directs the Read address to be retrieved from the Auxiliary Port.

   Note: When using RW words for recipe transfer functions, the range is RW00000 to RW59999. The transfer starts at the RW address specified by RWIxxxx with offset LW9000 and continues up to the number of words specified.

   Attribute:
   Download moves values from the HMI retentive memory to the controller.
   Save transfers values from the controller to the HMI retentive memory area.

3. Go to **Shape** Tab: Select Shape or Bitmap of the button to activate the transfer.

4. Go to **Label** Tab: Fill in fields to denote states, if desired.

5. Click **OK** to position the part and resize it if necessary.

   Note: Refer to Section 2: Software Reference Guide for details about completing each tab item

   Note: **Reserved System Bits Used for Recipe Transfer**
   LB9010 Indicates when a download is in progress: Downloading Active (ON) or Not Downloading (OFF).
   LB9011 Indicates when an upload is in progress: Uploading Active (ON) or Not Uploading (OFF).
   LB9012 Indicates if the HMI is: Downloading (ON) or Uploading (OFF)

   Note: **System Words Used for Recipe Transfer**
   RWxxxx Retentive memory location: Downloads and uploads begin at the address specified by LW9000.
   RWIxxxx Pointer to Retentive memory: Downloads and uploads start at the xxxx address of RWIxxxx.
   LW9000 Offset Pointer to Retentive memory: Downloads and uploads begin at the xxxx address of RWIxxxx plus the offset number that is in LW9000

Note: HMI Recipe Function

The retentive memory resides in battery backed SRAM. The memory contents are preserved for at least half year after power off. The battery is recharged whenever the system is powered. The total size of retentive memory is 64K words.

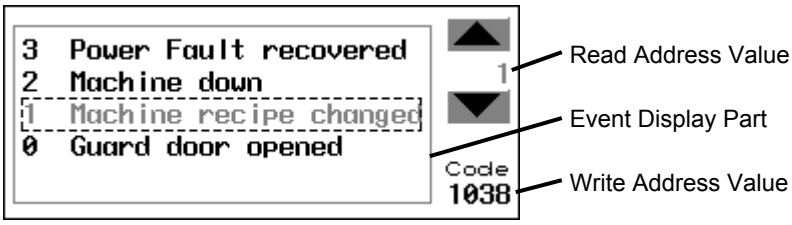
The retentive memory is accessed by an index address. The number of words you specify in LW9000 offsets an index address from its indicated address. For example if (LW9000)= 50, an RWI Device address of 100 physically accesses data at address RW 150 (100+50). This dual pointer system allows high flexibility in handing complex recipe tasks. By varying these addresses many recipes can be stored in retentive memory and downloaded or saved as needed.

For Example:
In this example, the controller has 5 registers that are parameters for machine operation (a recipe). The need to change these five registers depends on the part or process the machine is running. There are four different setups needed. These 4 setups are stored in the MMI-TS at internal register values RW200, RW210, RW220, and RW230. Download Setup 1 by writing the value 200 to internal register RWI0000. If you press a Recipe Transfer button that has the Download attribute, the Recipe at 200 is downloaded. Download Setup 2 by writing the value 210 to internal register RWI0000 and press the Recipe Transfer button that has the Download attribute (and so on).

Transfer one parameter by putting its position in LW9000. For example, as in the above example with 210 in register RWI0000, if you set the value in LW9000 to 3 and then press a 1 word Recipe Transfer button, it transfers only one register. Only parameter 3 is downloaded from Recipe 210.

13.23 Event Display

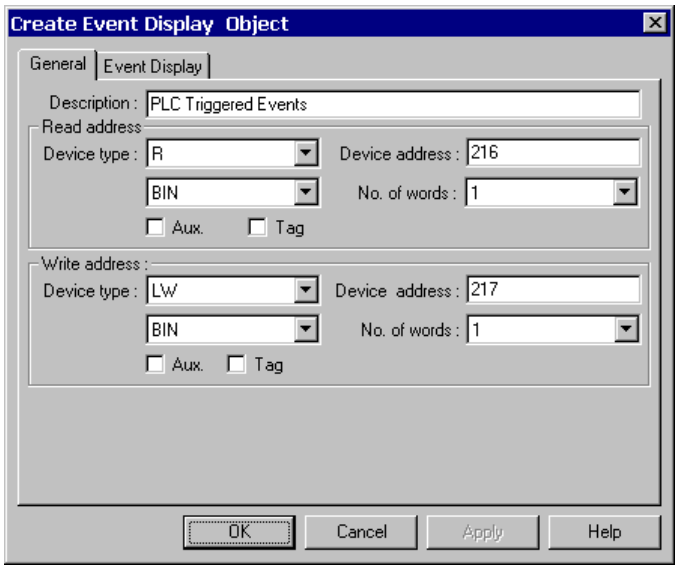
The Event Display part opens a window to display messages in prioritized order. Various formatting features allow the display of event trigger, acknowledge and return to normal times. The use of the Real Time Clock is required for proper display of the time.



Example of an Event Display showing Event #1 Selected and Acknowledged

13.22.1 Procedure to create an Event Display Part.

1. Click Event Display icon or select **Event Display** from the **Parts** menu.



2. Fill in General Tab Items:

Description: A reference name (not displayed) that you assign to the Event Display.

Read Address: Used to control the scrolling, up and down, of the Event display window. The value in the read address is the relative distance from the event in the topmost line of the display. All active messages are sorted by time, the new events are displayed on the topmost line and previous ones are below it. By varying the value in the Read address, previous Events can be viewed.

Device type is the word prefix.
Device address is the word number.
Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
See Word Lamp 13.2 for format description.
No. of words is fixed at 1 for this part.
Tag is used to select the Read address from a list of predefined Tags.
Aux. directs the Read address to be retrieved from the Auxiliary Port.

Write Address: A word that receives the write value specified in the Event Log's trigger event. The value is set when acknowledging event messages in the Event Display.

Device type is the word prefix.
Device address is the word number.
Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
See Word Lamp 13.2 for format description.
No. of words is fixed at 1 for this part.
Tag is used to select the Read address from a list of predefined Tags.
Aux. directs the Read address to be retrieved from the Auxiliary Port.

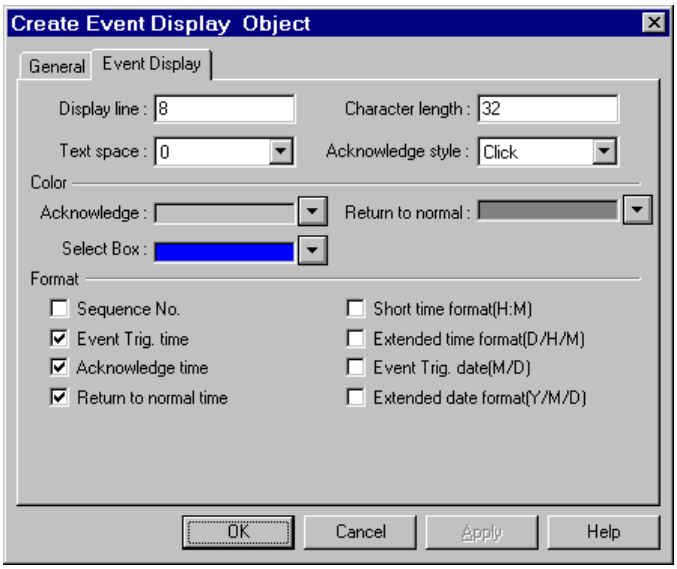
3. Fill in **Event Display** Tab Items:

Display line: Specifies the window height in lines of Font 16 text per line.

Character length: Specifies the width of the window in Font 16 characters. Extra characters past this length are truncated.

Text space: The number of pixels above and below messages.

Acknowledge style: An event is acknowledged by touching the event. Click is touch once. Double click is touch twice quickly. This puts a select box around the message and acknowledges the display.



Color: Colors of message text for different states (**Acknowledge**, **Return to normal**) of event messages. The **Select Box** is a dashed line that highlights a selected message.

Format: The information displayed before the message. Times are displayed in HH:MM:SS (Hours:Minutes:Seconds) format by default. Select the items to include in display. Some items add columns to the Event Display object. Be sure to compensate the display size to accommodate all the information.

Format Item	Description	Adds Column
Sequence No.	The number of the event. Event numbering starts at 0.	Yes
Event Trig. time	The time that the event was triggered.	Yes
Acknowledge time	The time that the event was acknowledged.	Yes
Return to normal time	The time when the event returned to a non-triggered state.	Yes
Short time format (H:M)	Change the time format in the time tag to Hours: Minutes.	No
Extended time format (D/H/M)	Change the time format in the time tag to Days/Hours: Minutes.	No
Event Trig. date(M/D)	The date when the event occurred. (Month/Day)	No
Extended date format (Y/M/D)	Change the date format in the date tag to Year/Month/Day.	No

Note: Be sure to allow sufficient character length to accommodate all information. If the Format information and message text exceeds the Character length, the message is truncated.

4. Click **OK** to position the part.

Note: Resize an Event Display part by adjusting the **Display line** and **Character length** attributes.

13.23.2 Storing Eventlog Information in Retentive Memory

Event Log data can be stored in retentive memory. Stored events are recalled at startup and displayed in Event Display parts. Use the following steps to implement this feature.

1. Go to the **System Parameters**, **Hardware Tab** and select **Set Eventlog DataBase**: No or Yes.

2. Set the **DataBase Start Address**:(RW)0~59999.

3. Set the System word LW9058 to the size the Eventlog data is allowed to occupy in retentive memory. Keep in mind that once a block of retentive memory is reserved by the Eventlog Database, the project cannot use it for other purposes.
Note: We recommended using the upper portions of retentive memory to avoid over writing recipe data.

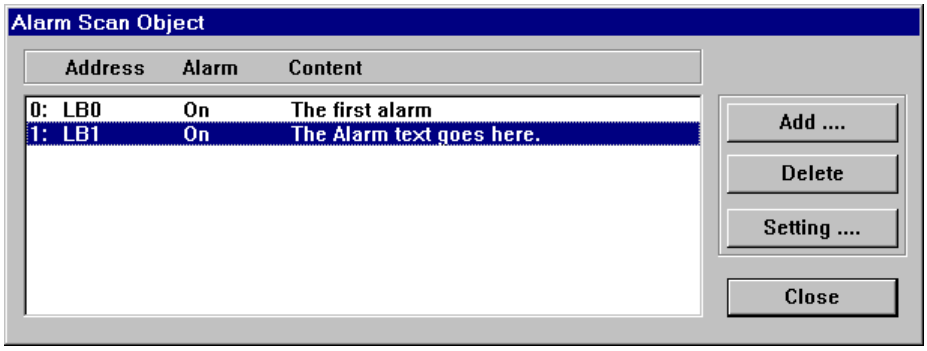
System Tools

13.24 Alarm Scan

Alarm Messages are displayed on Alarm Display and Alarm Bar parts. The message to be displayed must first be registered in the Alarm Scan list. A bit controls each message. If the bit activates the alarm (either ON or OFF), the corresponding message is displayed.

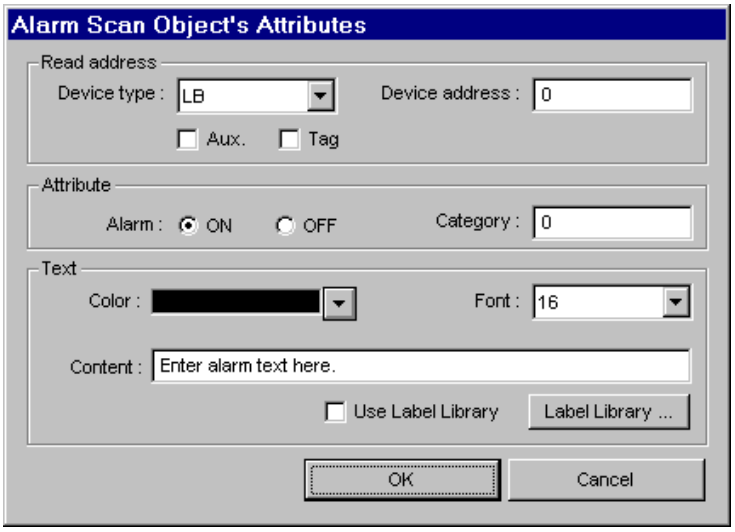
Procedure to add/modify alarm messages

1. Click Alarm Scan icon or select **Alarm Scan** from the **Parts** menu. The Alarm Scan Object message summary screen is displayed.



2. Click on the **Add** button to add a new message or click on the **Setting ….** button to edit an existing message. If you click on the **Delete** button, the current highlighted function is deleted

3. When adding or editing messages fill in the Attributes Dialog:



Read Address: specifies the PLC bit address that triggers the message.

Device type is the bit prefix.

Device address is the bit number

Tag is used to select the Read address from a list of predefined Tags.

Aux. directs the Read address to be retrieved from the Auxiliary Port.

Attribute:

Alarm

ON: displays this message when the bit is ON.

OFF: displays this message when the bit is OFF.

Category – alarm category (reserved for future use)

Text: Enter the message **Content** text, text **Color** and **Font** size.

Note: Size 24 font requires Alarm Display and Alarm Bar parts to have a minimum of two lines to be displayed properly.

4. Click **OK**, the message appears in Alarm Scan message summary box.

Note: Normally up to 200 alarm and event messages can be programmed into the Alarm Scan and Event Log objects. If more than 200 messages are needed, go to the **System Parameters**, **General Tab**, **Extra No. of Event:** and enter the additional amount. For example, if 1200 messages are needed, enter 1000. Up to 2800 extra events can be added. See Software Reference section, System Parameters for more information about this topic.

Alarm Scan Design Tip: In order to take full use of communication bandwidth, it is recommended that a block of continuous PLC bit devices be used for the Alarm Scan list. For example: Use Bits 100 to 199 to control the display of all alarm messages. In this case, one read command retrieves Bits 100 to 199 all at once instead of one bit at a time.

13.25 System Message

It is possible to customize system messages for different languages. There are three possible system messages that can be displayed.

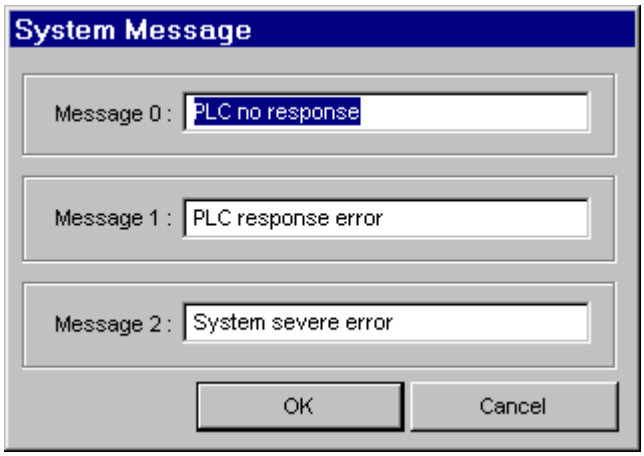
1. “PLC no response” - Displayed when the display cannot access the PLC.
2. “PLC response error” - Displayed when the reply message from PLC is different than expected.
3. “System severe error” – Displayed when the system runs out of memory or some other fatal error occurs.

Procedure to modify system messages

Click System Message icon or select **System Message** from the **Parts** menu to pop up the system message dialog box.

Fill in appropriate text for the content of each message to customize it.

Press **OK** to finish the modification.



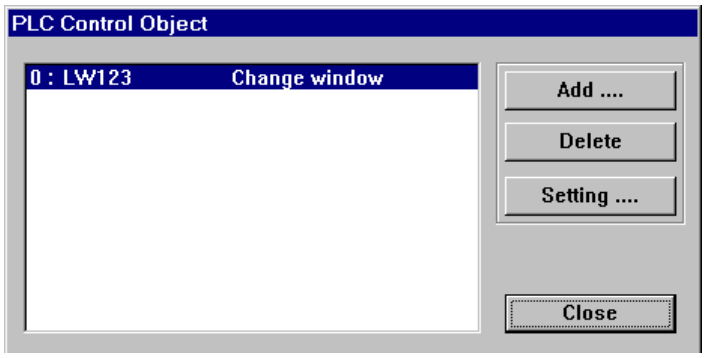
13.26 PLC Control

The PLC control provides a way for the PLC to control HMI system functions.

Procedure to create a PLC control function.

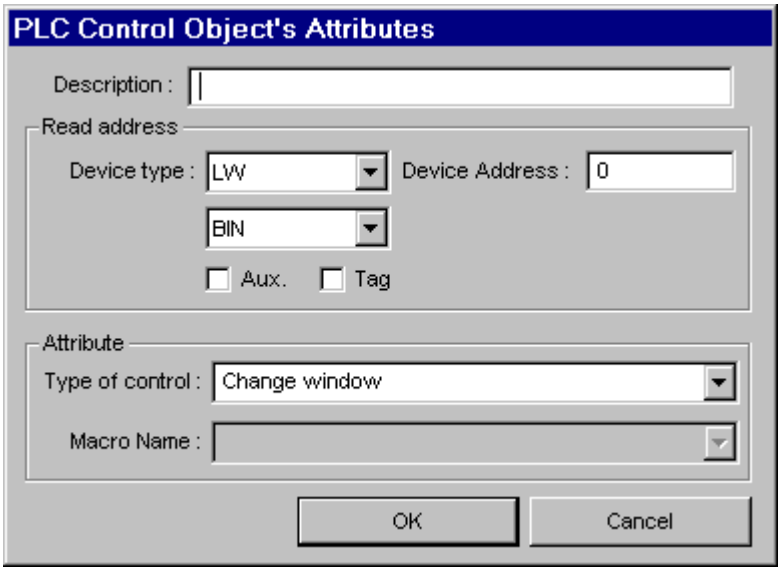
1. Click PLC control icon or select **PLC Control** from the **Parts** menu to pop up the PLC control summary screen.

2. Click on the **Add** button to create a new PLC control function. Click on the **Setting ….** button to edit an existing PLC control function. If you click on the **Delete** button, the current highlighted function is deleted.



3. Select options in the PLC Control dialog box.
   Read Address: Designates the address of the PLC control register.
   Device type is the bit or word prefix.
   Device address is the bit or word number.
   Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).
   See Word Lamp 13.2 for format description.
   Tag is used to select the Read address from a list of predefined Tags.
   Aux. directs the Read address to be retrieved from the Auxiliary Port.

   Attribute: Assign the operation activated by this function. See below.
   Type of control:
   See *Note:* Types of Controls below.



4. Click on the **OK** button to create the object and exit the dialog. The object appears in the PLC Control summary box

5. Click on the **Close** button to exit the PLC Control summary.

Note: Types of Controls

13.26.1 Change Window

This operation uses two addresses. The Read address holds the active window number. If the value stored in the Read address changes to a valid window number, that window number then replaces the currently displayed window. The new window number is moved into the Read Address + 1 register.

Read Address	Controls screen changes by number
Read Address + 1	Destination screen number is written

Note: System Register LW9055 acts as an offset for this type of control.
Example: A PLC Control / Change window uses D10. If (LW9055) = 10, and D10 = 14 then the HMI changes to window 24. After changing, the HMI writes back 14 to D11.

13.26.2 Back light control

Turns the display's backlight OFF when the Read address turns ON (positive edge triggered). When the backlight is OFF, a touch reactivates the backlight.

13.26.3 Screen hardcopy

Monitors the Read address. When the bit turns ON, the HMI prints the active base window. Once the printout is done, the HMI turns the controlling bit OFF. You must enable the printer in the System Parameter settings to use this feature.

13.26.4 Report printout

Prints the screen number designated by the Read address. After the screen is printed, the Read address is set to 0. For example, suppose that the Read address is D100. When the value of D100 is equal to 20, the HMI prints out window 20 in the background. When the HMI is finished printing, the HMI writes zero to address D100. You must enable the printer in the System Parameter settings to use this feature.

Note: System Register LW9054 specifies what is printed out in the report.
Report printout options

0: Text & Meter & Trend
1: Text & Meter & Trend & Shape but no pattern
2: Text & Meter & Trend & Bitmap
3: Text & Meter & Trend & Bitmap & Shape but not pattern
4: All

Note: System Register LW9055 acts as an offset for the Report printout control.
Example: A PLC Control / Report printout uses D100. If (LW9055) = 6, and D100 = 24 then the HMI prints out window 30. After printing, the HMI writes back 0 to D100.

13.26.5 Back light control (write back)

Turns the display's backlight OFF when the Read address turns ON (positive edge triggered). Whenever the backlight is turned OFF (through PLC control or time-out) the HMI turns the Read address bit OFF in the PLC. When the backlight is OFF, a touch reactivates the backlight.

13.26.6 Write data to PLC (base window)

The active, base window number is written to the Read address. Pop-up windows do not have their number written to the Read address.

13.26.6 General PLC Control

This control is used to trigger data transfer to and from the PLC. A Read address word is specified for the control. This object requires five or more words. The action performed is based on the value in the Read address as follows:

Read address value	Action
1	Data is transferred from the PLC to HMI RW memory locations.
2	Data is transferred from the PLC to HMI LW memory locations.
3	Data is transferred from HMI RW locations to the PLC.
4	Data is transferred from HMI LW locations to the PLC.

Note: Once the operation is finished, the Read Address is set to 0.

Control Word Assignments

Read address value	Action
Read address + 1	The number of words (**N**) to transfer (Maximum 32).
Read address + 2	The offset in the PLC of the starting address for transfers.
Read address + 3	The offset of the starting addresses for LW and RW transfers.
Read address + 4 +N	The beginning of the data for transfer and N words following

13.26.6 Execute macro program

Executes the Macro specified in the Macro ID field when the Read address turns ON (positive edge triggered). This is the only way to execute macros. Refer to the Macro Reference Section for more details about Macro creation and editing.

13.27 Event Log

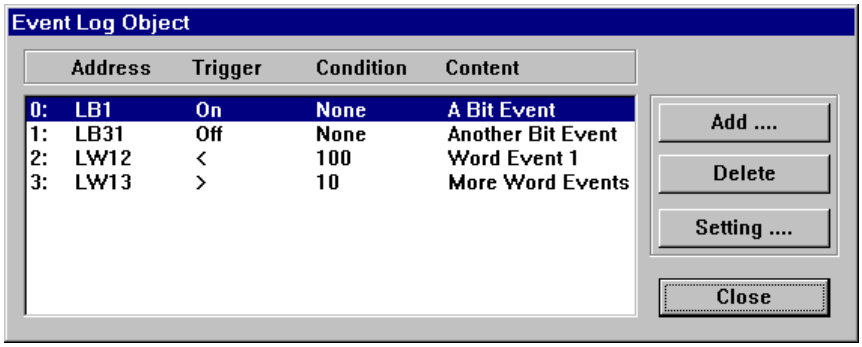
Event Log Messages are displayed on the Event Display Part. The message to be displayed on the Event Display must first be registered in the Event Log list. A bit or word device controls each message. If the bit or word device activates (either ON/OFF or value limit), the corresponding message is displayed in the Event Display. By default, events are stored in non-retentive memory and are lost when power is cycled to the unit. To retain events, activate the System Parameters, Hardware tab setting that stores events in retentive memory. Events stored there remain even when power is cycled.

Procedure to add/modify Event Log messages.

1. Click Event Log Tool or select **PLC Control** from the **Parts** menu to display the Event Log message summary screen.

2. Select the **Add** button to add a new message. Click on **Setting ….** button to edit an existing message. Use **Delete** to remove a highlighted message.

Note: There is no Undo operation for the Delete command.



3. Fill in Attributes Dialog:

Address type: Select **Bit** or **Word** address type.

Read Address: Specifies the PLC bit or word address that triggers the message.

Device type is the bit or word prefix.

Device address is the bit or word number.

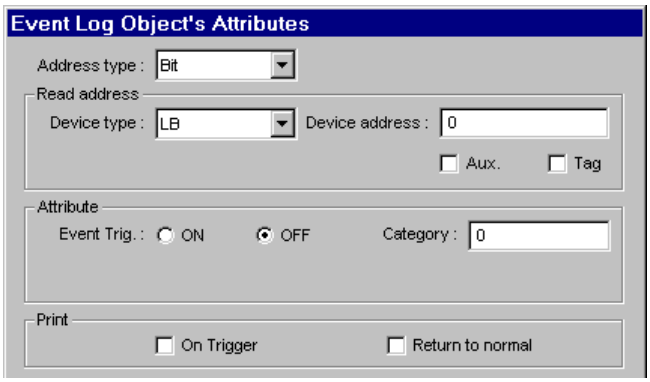
For **Word** types select how the device data is encoded.

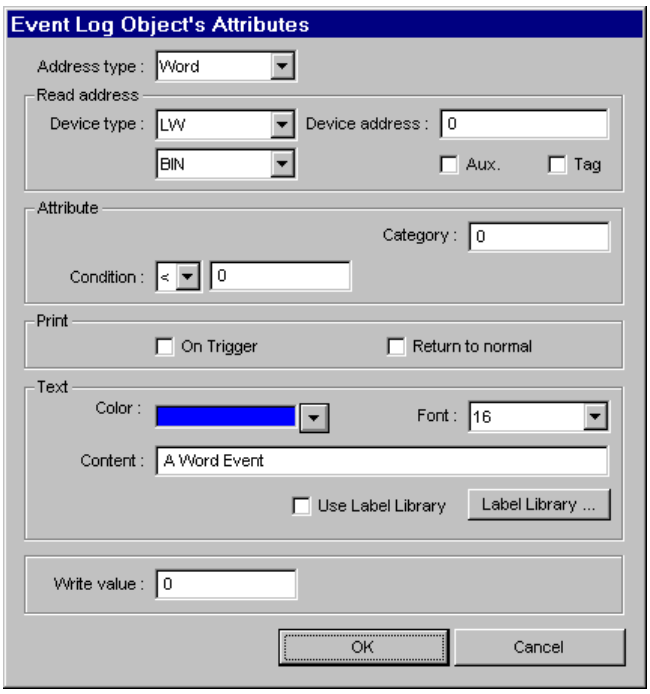
Data format: Defines data from the controller as binary (**BIN**) or binary coded decimal (**BCD**).

See Word Lamp 13.2 for format description.

Tag is used to select the Read address from a list of predefined Tags.

Aux. directs the Read address to be retrieved from the Auxiliary Port.





Attribute:

Event Trig.

Bits: **ON**: displays message when bit is ON. **OFF**: displays message when bit is OFF.

Words: Use "**<**" to trigger when value in word is less than entered value. Use "**>**" to trigger when value in word is greater than entered value.

Print: **On Trig.**: Prints out message when event is triggered.

Return to normal: Prints out message when the event returns to normal state.

Category – Event category (reserved for future use)

Text:

Fill in the fields as described below.

Item	Description
Content	Directly entered text message or Label from Label Library. See note on embedding values.
Color	Color of the text when displayed in the Event Display part.
Font	Font size, 16 or 24 point. Default is 16.
Use Label Library	Select to use a predefined Label form the Label Library instead of directly entered text.
Label Library …	Click to call up the Label Library for editing or adding Labels.
Write Value	A value to be written to the Write address of the Event Display when the event is acknowledged.

Note: Data from Local Words can be embedded in an Event message for printing only. To embed values use the **%nnd** format. For display, the message is blank where the word data is displayed.

To print out current value of PLC register data, first assign a Recipe transfer object or PLC Control/General Control to move data from the PLC to internal (LW) memory.

Where:

%	The starting delimiter
nn	An internal register (LW) number 00-99
d	The ending delimiter

For example: The Content field is set as "Current temperature value %25dF: HIGH ALARM"
If LW25 = 120 then the message are printed out as "Current temperature value 120F: HIGH ALARM".

4. Click **OK**, the message appears in the Event Log message summary box.

> ***Note:*** Normally 200 events are stored in the Event Log. Once the Event Log is full, Events are deleted from the beginning of the list as new Events occur (First-In-First-Out). If more than 200 events are needed, go to the System Parameters, General Tab, **Extra No. of Event:** and enter the additional amount is entered here. Up to 2800 additional logs can be added. For example, if 1000 events need to be logged, enter 800 in this field.
>
> ***Note:*** In order to take full use of communication bandwidth, it is recommended that a block of continuous PLC bit devices be used for the Event Log list. For example: Use Bits 100 to 199 to control the display of all event messages. In this case, one read command retrieves Bits 100 to 199 all at once instead of one bit at a time.

Storing Event Log Data to Retentive Memory

Normally, Event Log data is stored in temporary memory. Event Log data is lost when power to the unit is lost. Event Log data can be saved to retentive memory by enabling the **System Parameters - The Hardware Tab,** EventLog Database check box. This sets aside a 20 word header area for the database. Events are then stored after that area in retentive memory as they occur.

System reserved words used in conjunction with this feature:

LW9057 contains the size that is allocated for a single event. This is the size of the largest possible **EventLog** DataBase Item.

LW9058-9059 contains the size in words of the EventLog in retentive memory. Keep in mind that once a block of retentive memory has been reserved, your project cannot use it for other purposes. We recommend that you specify the DataBase Start Address to be after stored recipes so they are not overwritten with Event Log data.

LB9090 is a bit that clears the event log data in memory. This is used to reset the log once its information has been uploaded or is no longer useful.

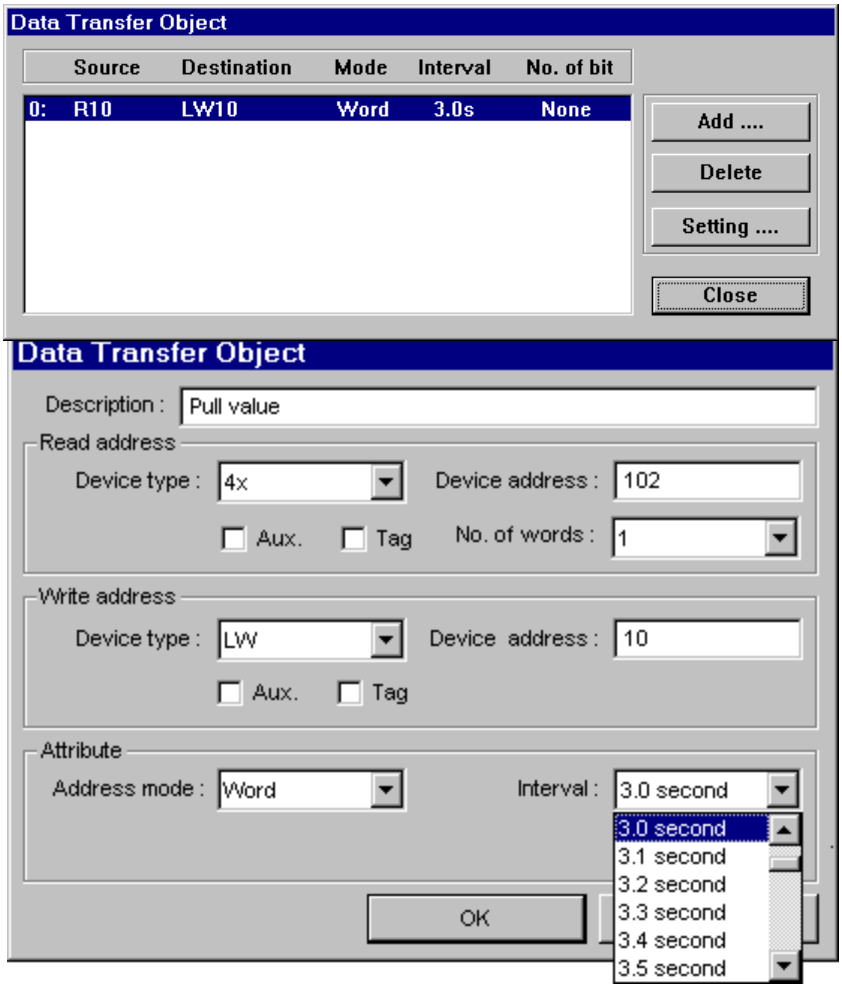
13.28 Data Transfer

This part is used to transfer data from one location in the Controller or HMI to another in the controller or MMI. Data may be single or multiple bit or word data. Data Transfer functions are always active no matter what screen is being displayed.

Procedure to create a Data Transfer function.

1. Click Data Transfer Tool or select **Data Transfer** from the **Parts** menu to pop up the Data Transfer summary screen.

2. Click on the **Add ….** button to create a new Data Transfer function or click on the **Setting ….** icon to modify an existing Data Transfer function. If you click on the **Delete** button, the current highlighted function is deleted.



3. Assign options in the Data Transfer Object dialog box.

Description: A reference name (not displayed) that you assign to the Data Transfer.

Source address: The starting address of the bit or word to transfer.

Device type is the word prefix.
Device address is the word number.
No. of Words is the number of consecutive words to transfer. (Applies to Word Transfers only)
Tag is used to select the Read address from a list of predefined Tags.
Aux. directs the Read address to be retrieved from the Auxiliary Port.

Destination Address: The starting address of the bit or word that receives the data.

Device type is the word prefix.
Device address is the word number.
Tag is used to select the Destination address from a list of predefined Tags.

Attribute:
Set **Address mode** to transfer **Bit** or **Word** data.
Number of Bits is the number of consecutive bits to transfer. (Applies to Bit Transfers only)
The **Interval** selection is the frequency rate (0.0 to 25.5 seconds) at which the block transfer is repeated. An Interval of 0.0 transfers the data once at power up.

Note: Be sure to allow for communications time lags. Specifying a time that is shorter than the time it takes to actually transfer the data locks up the unit.

4. Click **OK** when done entering values.

5. Click **Close** in the Data Transfer Object dialog when all desired Data Transfer functions have been programmed.

Section 4 Macro Reference

14.1 Overview

Macros provide the additional functionality your application may need. Macros are automated sequences of commands that are executed at run-time. Macros allow you to perform tasks such as complex scaling operations, string handling, and user interactions with your project.

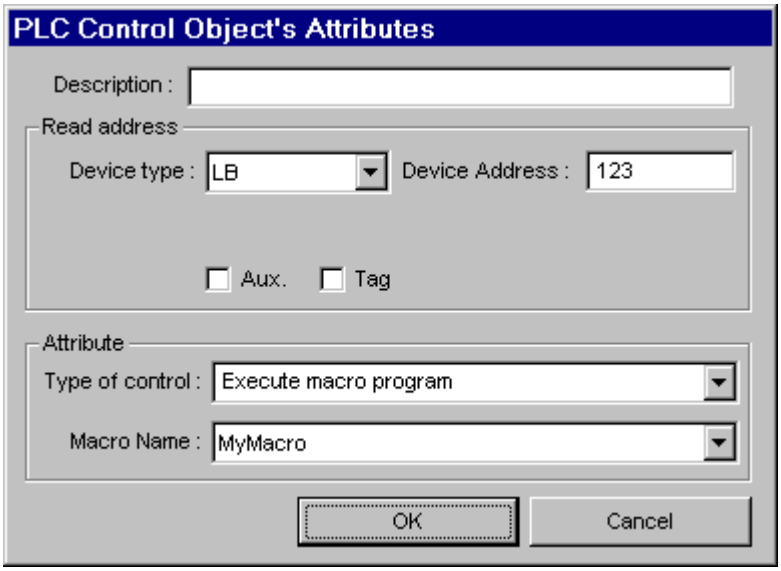
Note: Up to 256 Macros can be placed in a project.

14.1.1 Triggering a Macro

A Macro is triggered by turning ON a bit that is assigned to an **Execute macro program** PLC Control.

Note: The Macro is triggered every time the Bit's Read address is turned ON. Leaving the bit ON does not continually execute the Macro. The bit must be turned OFF then back ON to trigger the Macro another time.

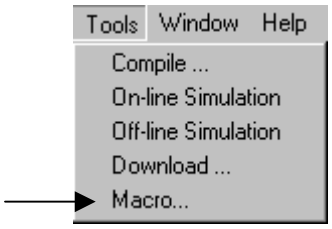
The triggering bit can be programmed to turn on with any PLC or HMI Project action.



14.2 Editing Macros

The Macro Dialog is opened from the **Tools**|**Macro…** Menu command.

Use the  tool shortcut for the **Macro…** command.

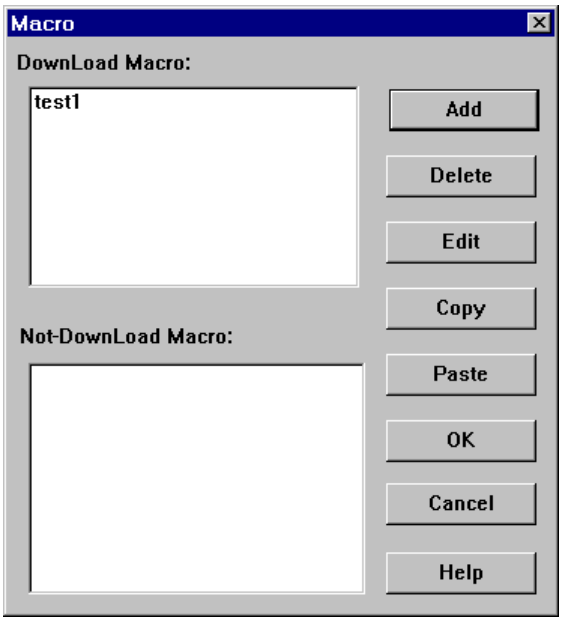


14.2.1 Macro Dialog Features

The Macro Dialog is used for creating and managing defined Macros.

DownLoad Macro: The list of Macros that are added to the project *.eob file at compile time.

Not-DownLoad Macro: The list of Macros not added to the project's compiled *.eob file. The Macros in this list are not compiled and may be used for reference. They may be in the development process or reserved for future use in the project.



Add: Starts the process for adding and editing a new Macro.

Delete: This command is used to delete the highlighted Macro from one of the lists. This command cannot be recovered with **Undo**.

Edit: Brings up the Macro editor displaying the highlighted Macro.

Copy: Copies the Highlighted Macro.

Paste: Pastes a copied Macro into the list with a digit after the name. Use this to duplicate Macros.

OK: Exits the Macro Dialog and saves changes.

Cancel: Exits the Macro Dialog without saving changes.

Help: Calls up the help topic for the Macro Dialog.

14.2.2 Workspace Macro Editor

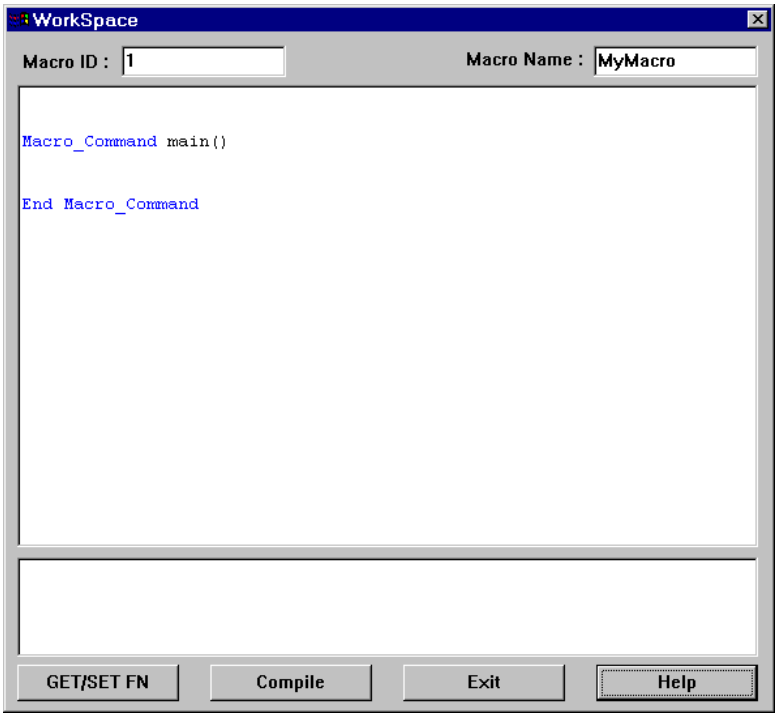
The Workspace Dialog is where the Macro editing is done. Macro commands are entered into the upper text field. Review the Macro command functions and format section before attempting to enter Macro text.

Macro ID: This number identifies the Macro for use with PLC Control, Execute macro program commands. The valid range is 1 to 65535.

Macro Name: A brief but descriptive title for the Macro. This title is displayed in the list boxes of the Macro Dialog. Only 27 characters are displayed.

The upper text box is used for editing the Macro. The lower text box is for displaying compile messages and errors.

GET/SET FN: Calls up the PLC Dialog wizard for creating commands for internal register operations. See the Macro Syntax section.



Compile: Checks the Macro for errors and displays them in the lower text box of the Workspace Dialog.
If no errors are found, the Macro is compiled and the Workspace Dialog is closed. The macro name is displayed in the *DownLoad* Macro Dialog list.

Workspace Dialog showing a compile error.

Cancel: Closes the Workspace Dialog. If the Macro has not been compiled successfully, the macro name is displayed in the *Not DownLoad* Macro Dialog list.

Help: Calls up the help topic for Macro editing.

14.3 Syntax

A Macro is made up of statements. The statements contain constants, variables and operators. The statements are put in a specific order to create the desired output.

14.3.1 Constants and Variables

14.3.1.1 Constants

Constants are fixed values and can be entered directly into statements as:

Constant Type	Note	Example
Decimal Integer		1234, -345, 0, 1138463220
Hexadecimal	Must begin with **0x**	0x2A, 0xFFFF, 0x20B54A
ASCII	Strings must be enclosed in single quotes	‘a’, ‘test3’, ‘47’
Boolean		True, False

Example statement using a constant:

C = 397

Where: C is a variable and 397 a constant.

Note: PLC and System registers and bits cannot be directly put into a Macro. Use the GetData and SetData Statements to assign their values to Macro Variables for later use in mathematical manipulation.

14.3.1.2 Variables

Variables are names that represent information. The information can change as the variable is modified by statements.

Naming rules for Variables:

1. All variable names must start with an alphabet character. Other characters in the variable can be characters or numbers.
2. Punctuation, the "_" character, spaces and non-alphanumeric characters cannot be used in variable names.
3. Variable names longer than 32 characters are not allowed.
4. Reserved words cannot be used as Variable names.

Examples of good variable names:	
Temperature	LimitSwitch41
PresetCounter	x
win	errorcode

Examples of bad variable names:	Error
3wayswitch	begins with a number
Run_Lamp	illegal character "_"
mod	reserved word
OverCurrentSensingRelayTrippingPoint	longer than 32 characters

There are 5 different variable types:

Type	Description	Range
bool	1 bit (discrete) variable	0,1
char	8 bit (byte) variable	±128
short	16 bit (word) variable	±32767
int	32 bit (double word) variable	±4294836225
float	32 bit (floating point) variable	±4294836225

Declaring Variables

Variables must be declared before being used. All variable declarations must be made before any other statements in the macro. To declare a variable, specify the type then the variable name.

Example of a simple declaration	`int n`

If there more than a single variable of a particular type is needed, they can be declared in the same statement by separating them with commas.

Example of multiple variable declaration	`short a, InputBlock1, color`

Declaring Arrays

Macros support one-dimensional arrays. To declare an array of variables, specify the type, the variable name then the number of variables in the array enclosed in brackets "[ ]". Arrays are 1 to 4096 variables in length. (Macros only support up to 4096 variables per macro)

Example of Array declaration	`int g[10]`

Variable and Array Initialization

There are two ways variables can be initialized:

1. By statement using the assignment operator (=).

Example of statement initialization	`int n, V[4]`
	`n = 399`
	`V[0] = 24`

2. During declaration

Example of initialization during declaration	`char i='z', a, b=3`

The declaration of arrays is a special case. An entire array can be initialized during declaration by enclosing comma separated values inside curly brackets "{ }"

Example of Array initialization at declaration	`int V[4]={24, 15, 5, 0}`

14.3.2 Operators

Operators are used to designate how data is to be manipulated. In each statement, the operator on the left is set to the conditions on the right. The simplest of these is the assignment operator.

Operator	Description	Example
=	Assignment Operator	preset = 77

Arithmetic operators are used for scaling and offsetting values. The priority of arithmetic operators from highest to lowest is as follows: multiplication and division (*, /) first then modulo division (mod) and finally addition and subtraction (+, -).

Arithmetic Operators	Description	Example
+	Addition	productionTotal = good + reject
-	Subtraction	weight = Ainput - tare
*****	Multiplication	scaled = flow * 0.12
/	Division	ratio = InA / InB
%	Modulo Division (returns remainder)	quarter = n % 3

Comparison operators are used for conditional statements such as If/Then or While/Wend. The priority of comparison operators is from left to right within the statement.

Comparison Operators	Description	Example
<	Less Than	If preset < b Then preset = b
<=	Less than or Equal	If pressure < 25 Then alarm = True
>	Greater Than	If count > total Then stop = True
>=	Greater Than or Equal	While x >= 0 …statements… Wend
==	Equal	If (x mod 2) == 1 Then odd = True
<>	Not Equal	If adjust <> final Then error = 84

Logic operators are used with the other operators to make boolean (True, False) results for conditional statements. The priority of logic operators from highest to lowest is as follows: Not, And, Or, Xor

Logic Operators	Description	Example
And	Conditional AND	If (x < b) And alarmFlag Then error = 26
Or	Conditional OR	If In1 OR In2 Then comp = 534
Xor	Conditional Exclusive Or	If error Xor 256 Then fault = True
Not	Conditional NOT	While x >= 0 And Not alarm …statements… Wend

Bitwise and Shift Operators are used to manipulate bits within char, short, and int variable types. The priority of these operators is from left to right within the statement.

Shift Operators	Description	Example
<<	Shift left specified number of bits	z = x << 2
>>	Shift right specified number of bits	MSB = timer >> 8
Bitwise Operators		
&	ANDs two values together	alarm = error & 128
|	ORs two values together	load = In1 and Out4
^	XORs two values together	a = a ^ b
~	Compliments a value	inverse = ~value

Note: The overall priority of all operators from highest to lowest is as follows:

Operations within parenthesis are carried out first.
Arithmetic operations
Shift and Bitwise operations
Comparison Operations
Logic Operations
Assignments

14.3.3 Reserved Keywords

The following keywords are reserved for Macro use. They cannot be used for variable, array, or function names.
+, -, *, /, %, >=, >, <, <=, <>, ==, And, Or, Xor, Not, <<, >>, =, &, |, ^, ~
If, Then, Else, End, Select, Case, For, To, Down, Step, Next, While, Wend, Break , Continue, Return, Sub, bool, char, short, int, float

14.4 Statement Construction

Statement construction depends on the statement's purpose. All statements begin on a new line.

14.4.1 Definition Statement

This covers the declaration of variables and arrays. The formal construction is as follows:

type name
type name[constant]

14.4.2 Assignment Statement

Assignment statements use the assignment operator to move data from the expression on the right side of the operator to the variable on the left side. An expression is the combination of variables, constants and operators to yield a value.

Variable = Expression

14.4.3 Logical Statements

Logical statements perform actions depending on conditions. There are two types of conditional statements.

14.4.3.1 If-Then-Else Statements

These statements perform actions depending on the condition of a Boolean expression.
The Syntax is as follows:

Single Line format:

```
If <Condition> Then [statements] Else [statements]
```

Block format:

```
If <Condition> Then
  [Statements]
Else
  [Statements]
End If
```

Syntax description:

If	Must be used to begin the statement
<Condition>	Required. This is the controlling statement. It is FALSE when the condition evaluates to 0 and TRUE when it evaluates to 1.
Then	Must precede the statements to execute if the Condition evaluates to TRUE.
[Statements]	Statements to execute. These can include other If-Then-Else statements.
Else	Must precede the statements to execute if the Condition evaluates to FALSE.
End If	Must be used to end an If-Then-Else statement when it is entered in Block format.

14.4.3.2 Select-Case Statements

The Select Case is a specialized form of the If-Then-Else statement. It allows multiple comparisons to values.
The Syntax is as follows:

```
Select Case <Expression>
[Case <evaluation>]
  [Statements]
[Case Else]
  [Statements]
End Select
```

Syntax description:

Select Case	Must be used to begin the statement
< Expression >	Required. This is the controlling expression. The result of evaluating the expression is used as a test in the Case expressions.
Case	Must precede the statements that evaluate the results of the Select Case <Expression>
< evaluation >	This is the controlling statement for each Case. It is compared to the results of the Select Case <Expression>. If it evaluates to TRUE then the statements below it are executed. If fit evaluates to FALSE then the macro jumps to the next Case or Case Else.
[Statements]	Statements to execute when the evaluation is TRUE.
Case Else	Must precede the statements to execute if none of the Cases evaluates to TRUE.
End Select	Must be used to end a Select Case.

14.4.4 Reiterative Statements

Reiterative statements control loops and repetitive tasks depending on conditions. There are two types of reiterative statements.

14.4.4.1 For-Next Statements

The For-Next construction is for stepping through a fixed number of iterations. A variable is used as a counter to track progress and test for ending conditions. Use this for fixed execution counts.
The Syntax is as follows:

```
For <Counter> = <StartValue> To <EndValue> [Step <Stepvalue> ]
or
For <Counter> = <StartValue> Down <EndValue> [Step <Stepvalue> ]
  [Statements]
Next [counter]
```

Syntax description:

For	Must be used to begin the statement
< Counter >	Required. This is the controlling variable. The result of evaluating the variable is used as a test for completion.
<StartValue>	Required. The < Counter >s initialized value.
To or Down	This determines if the Step control increments or decrements the <Counter>. “To” increments <Counter> by <Stepvalue>. “Down” decrements <Counter> by <Stepvalue>.
<EndValue>	Required. The test point. If the < Counter >is greater than this value, the macro exits the loop.
Step	Optional keyword. Specifies that a <StepValue> other than one is to be used.
<StepValue>	The value that < Counter > is changed every loop. If [Step <Stepvalue>] keywords are omitted the step value defaults to 1.
[Statements]	Statements to execute when the evaluation is TRUE. For-Next loops may be nested.
Next	Must precede the statements to execute if none of the Cases evaluates to TRUE.
[counter]	Optional < Counter > name. This is used when nesting For-Next loops.

14.4.4.2 While-Wend Statements

The While-Wend construction is for stepping through an unknown number of iterations. A variable is used to test for ending conditions. When the condition is TRUE, the statements are executed repetitively until the condition becomes FALSE.

Caution! Care must be taken to avoid infinite loops. Infinite loops lock-up the MMI.

The Syntax is as follows:

```
While <Condition>
  [Statements]
Wend
```

Syntax description:

While	Must be used to begin the statement
<Condition>	Required. This is the controlling statement. When it is TRUE the loop begins execution. When it is FALSE the loop terminates.
[Statements]	Statements to execute when the evaluation is TRUE.
Wend	Indicates the end of the While-Wend statement.

14.4.5 Optional Keywords

These optional keywords can be used in statements to control macro execution.

Syntax description:

break	Used in For-Next, While-Wend or Select-Case blocks. It causes the macro to end the block.
continue	Used in For-Next or While-Wend blocks. It ends the current iteration of the loop and starts the next one.
return [value]	Ends the macro or function. The optional value can be a fixed number or a variable used in the macro/function.

14.5 Macro Construction

A Macro is constructed in the following fashion:

Global Variable Declarations .....Optional

Sub Function Block Declarations.....Optional
Local Variable Declarations
  [Statements]
End Sub

“Macro_Command main( )” .....Required
Local Variable Declarations
  [Statements]
“End Macro_Command” .....Required

14.5.1 Local and Global Variables

Local variables are used within the main macro function or in a defined function block. Its value remains valid only within the specific block.

Global variables are declared before any function blocks and is valid for all functions in the macro. When local variables and global variables have the same declaration names, only the local variables are valid.

14.5.1 Function Blocks

Function Blocks are useful for reducing repetitive code, must be defined before use and can use any variable and statement type. A function block is called by putting its name followed by parameters, in parenthesis, in the Main Macro Function. After the function block is executed, it returns the value to the Main Function where it is used as an assignment or condition.
Note: Macro Functions cannot be nested or interleaved.

The Syntax is as follows:

```
Sub type <name> [(parameters)]
  Variable Declarations
  [Statements]
  [Return [value]]
End Sub
```

Syntax description:

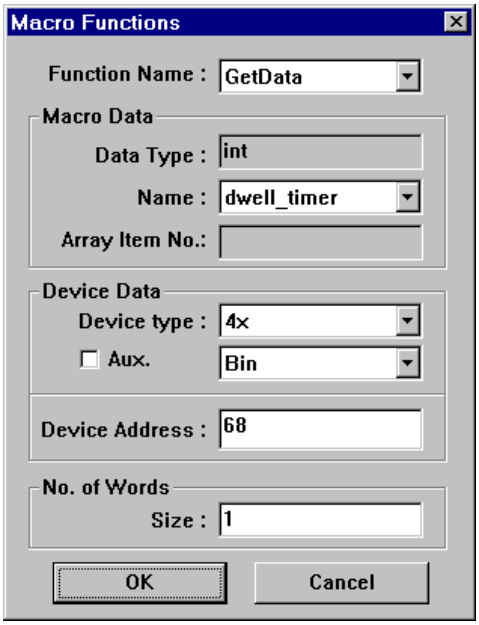
Sub	Must be used to begin the Function Block
type	Required. This is the data type of value that the function returns.
<name>	The name of the Function. It is used in the Main Macro to call the function.
(parameters)	Optional. The parameters hold values that are passed to the function by the Main Macro. The passed parameters must have their type declared in the parameter field and assigned a variable name. For example: Sub int MyFunction (int x, int y). x and y would be integers passed to the function by the Main Macro. This function is called by a statement that looks similar to this: "z = MyFunction(123, preset)" Notice that the calling statement can pass hard coded values or variables to the function. After the function is executed, an integer value is returned to "z".
Variable Declarations	Variables that are used in the Function Block must be declared first. This is in addition to passed parameters. In the above example x and y are variables that the function can use. Global variables are also available for use in Function Blocks.
[Statements]	Statements to execute.
[Return [value]]	Used to return a value to the calling statement. The value can be a constant or a variable. Return also ends Function Block execution.
End Sub	Must be used to end a Function Block.

14.5.2 Built in Function Blocks

EasyBuilder has two built in Function Blocks for retrieving and transferring data to the PLC. They are accessed by pressing the **GET/SET FN** button in the Macro Editor.

14.5.2.1 GetData Function

The GetData Function retrieves data from the PLC and puts it into a variable for the Macro to use. To access the **GetData** Function, Click on a blank line in a Statement area of the macro then click on the **GET/SET FN** button. The **Macro Functions** dialog pops up.



Select **GetData** as the **Function Name**.

Macro Data
Data Type: The variable type that receives the data. This automatically changes to match the variable selected in the **Name**.

Name: This dropdown list displays the declared variables of the Macro. Select the variable to hold the PLC data.

Array Item No.: If an array is selected for the **Name**, the particular item of the array must be designated.

Device Data
Device Type: Select the Device type from the dropdown list. The Device type depends on the PLC driver. Once this is selected, you can chose to get the data in Binary (**Bin**) or BCD (**Bcd**) format.

Aux. directs the Device Data to be retrieved from the Auxiliary Port.

Device Address: this is the address number of the Device Type. The format is same as any other EasyBuilder part.
Note: Extended Addressing mode is not allowed in this field. If Extended addressing is active, PLC data can be put into Local words/bits using the Data Transfer Part or General PLC Control, and then manipulated from there.

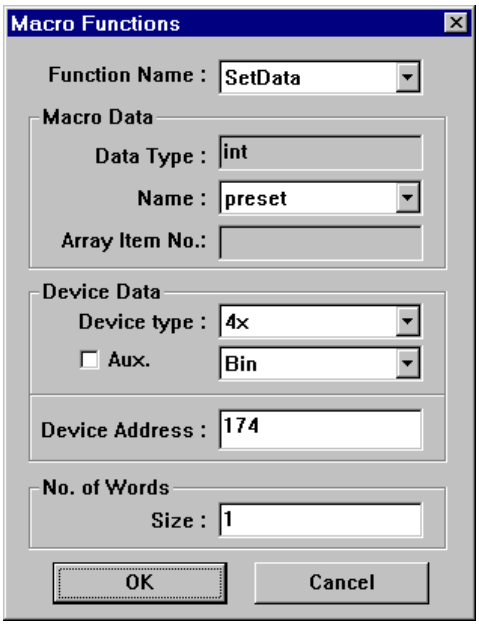
No. of Words
Size: This is the number of words to get.

Click **OK** to place the **GetData** Function into the code. Click **Cancel** to exit the function without saving any changes. Using the above graphic as an example, the code looks like this:

```
GetData(DwellTimer ,4x_Bin ,68,1)
```

14.5.2.2 SetData Function

The **SetData** Function sends data to the PLC and from a variable. To access the SetData Function, Click on a blank line in a Statement area of the macro then click on the **GET/SET FN** button. The Macro Functions dialog pops up.



Select **SetData** from the dropdown as the **Function Name**.

Macro Data
Data Type: This is the variable type for transfer into the PLC. This automatically changes to match the variable selected in the **Name**.

Name: This dropdown list displays the declared variables of the Macro. Select the variable to send to the PLC.

Array Item No.: If an array is selected for the **Name**, the particular item of the array must be designated.

Device Data
Device Type: Select the Device type from the dropdown list. The Device type depends on the PLC driver. Once this is selected, you can chose to send the data in Binary (**Bin**) or BCD (**Bcd**) format.

Aux. directs the Device Data to be sent to the Auxiliary Port.

Device Address: The address number of the Device Type. The format is same as any other EasyBuilder part.
Note: Extended Addressing mode is not allowed in this field. If Extended addressing is active, Variable data can be put into Local words/bits then transferred to the PLC by using the Data Transfer Part or General PLC Control.

No. of Words
Size: This is the number of words to send.

Click **OK** to place the SetData Function into the code. Click **Cancel** to exit the function without saving any changes. Using the above graphic as an example, the code looks like this:

```
SetData(preset ,4x_Bin ,174,1)
```

Note: Once you are familiar with the format, the **GetData** and **SetData** functions can be entered by hand. The compiler notices any errors if the format is incorrect.

14.6 Compile error messages

There are many causes for compiler errors. As the compiler detects errors it displays a warning message in the lower text box of the Workspace Editor. Macros with errors cannot be downloaded and are listed in the Not Download Macro box of the Macro Dialog.

Error message format: Macro_name(: Error_Number) Error_Message
The Error_ Number corresponds to one of the conditions listed below.

Error_Number descriptions

Number	Message Format	Sample Code	Reason for error

1	"Syntax error:" 'identifier'	`Macro_Command main( )`
		`  Char i , 3xyz`
		`  int g[4]`
		`  g[3] = 4`
		`End Macro_Command`
			An unsupported variable name ,"Error message: "Syntax error: 3x"
			Variable names must begin with alphabet characters.

2	'identifier' used without having been initialized	`Macro_Command main( )`
		`  Char i`
		`  int g[i]`
		`  g[3] = 4`
		`End Macro_Command`
			Undefined size of an array. Arrays must be defined with fixed values.

3	"Re-declaration error: " 'identifier'	`Macro_Command main( )`
		`  int g[10] , g`
		`  For g = 0 To 2`
		`  g[3] = 4`
		`  Next g`
		`End Macro_Command`
			"g" is defined twice in the same function. The name of variable or function cannot be used more than once in a function.

4	"Function name error:" 'identifier'	`Macro_Command If( )`
		`. . .`
		`End Macro_Command`
			"If" is being used as the Function name. Reserved keywords or constants cannot be the name of a function

5	"Statement missing"	`Macro_Command main )`	Function definition missing "("
6	"Missing expression in If statement"		The format for "If-Then" statement is: If [logic expression]Then [Else [If [logic expression] Then]] EndIf Formats other than this cause compile errors.
7	"Missing "Then" in If statement"		
8	"Missing "EndIf" "		
9	"Unfinished "If' statement before "End If" "		
10	"Illegal Else statement"		

11	"There should be constant behind "Case" "		The format for "Select-Case" statement is:
			Select Case [expression]
			  Case [constant]
			  Case [constant]
			  Case [constant]
			  Case Else
			End Select
			
			Formats other than this cause compile errors.

12	"Missing "Case" behind "Select" "		
13	"Missing "expression" behind "Select Case" "		
14	"Missing "End Select" statement"		
15	"Illegal "Case" statement"		
16	"Unfinished "Select" statement before "End Select" "		
17	" "For" statement error: missing "For" before "Next" "		The format for "For-Next" statement is: For [variable] = [initial value] To [end value] [Step] Next [variable] Formats other than this cause compile errors.
18	"Should be integer of char variable"		
19	"Missing assign statement"		
20	"Missing keyword "To" "		
21	"Missing "Next" statement"		
22	" "While" statement error: missing "While" before "Wend" "		The format of "While" statement is: While [logic expression] Wend Formats other than this cause compile errors.
23	"Missing "Wend" statement"		

<table>
<tr><td>24</td><td>"Illegal “Break” statement</td><td colspan="2">“Break” statement can only be used in “For-Next”, “While-Wend”, or “Select-Case” statements.
“Break” statement takes one line of Macro.</td></tr>
<tr><td>25</td><td>"Illegal “Continue” statement"</td><td colspan="2">“Continue” statement can only be used in “For-Next” or “While-Wend” statements.
“Continue” statement takes one line of Macro.</td></tr>
<tr><td>26</td><td colspan="3">"Expression error"</td></tr>
<tr><td>27</td><td>"Illegal operation object"</td><td>Macro_Command main( )
int g[10] , g
g[3] = 4 + xyz
End Macro_Command</td><td>“xyz” represents an incompatible object. The mismatch of operational objects in an expression causes compile errors.</td></tr>
<tr><td>28</td><td>"Missing “Sub” "</td><td colspan="2">The format of function declaration is:
Sub(Macro_Command) [data type] function_name( )
End Sub(Macro_Command)

Formats other than this cause compile errors.</td></tr>
<tr><td>29</td><td colspan="3">"Missing “Macro_Command” "</td></tr>
<tr><td>30</td><td>"Mismatch of the number of parameters "</td><td colspan="2" rowspan="3">The number of parameters in a function’s declaration must be equivalent to the number of arguments passed to the function.</td></tr>
<tr><td>31</td><td>"Mismatch of data type of parameter"</td></tr>
<tr><td>32</td><td>"Parameter error"</td></tr>
<tr><td>33</td><td>"Undefined function"</td><td colspan="2">Any variable or function should be declared before use.</td></tr>
<tr><td>34</td><td>"Illegal member of array</td><td colspan="2"></td></tr>
<tr><td>35</td><td>"Illegal definition of array";</td><td colspan="2"></td></tr>
<tr><td>36</td><td>"Illegal index of array"</td><td colspan="2"></td></tr>
<tr><td>37</td><td>"Undefined symbol"</td><td colspan="2"></td></tr>
<tr><td>38</td><td>"Un-supported PLC data address" ;</td><td colspan="2">The parameter of GetData(…) , SetData(…) should be legal PLC address.</td></tr>
<tr><td>39</td><td>"Should be integer, character or constant"</td><td colspan="2">The format of array is:
Declaration: array_name[constant]
(constant is the size of the array)

Usage: array_name[integer, character or constant]

Formats other than this cause compile errors.</td></tr>
<tr><td>40</td><td>"Illegal Macro statement before declaration statement "</td><td>Macro_Command main( )
int g[10]
For g[2] = 0 To 2
g[3] = 4 + g[9]
int h , k
Next g[2]
End Macro_Command</td><td>Statement position of “int h , k” declaration caused the error. All declarations must be made before operational statements.</td></tr>
<tr><td>41</td><td colspan="3">"Floating point can not bitwise shift"</td></tr>
<tr><td>42</td><td colspan="3">"Missing function return value "</td></tr>
<tr><td>43</td><td colspan="3">"Function can not return a value"</td></tr>
<tr><td>44</td><td colspan="3">"Illegal Float data type in expression"</td></tr>
<tr><td>45</td><td colspan="3">"Error PLC address"</td></tr>
<tr><td>46</td><td colspan="3">"Stack can not exceed 4k bytes"</td></tr>
<tr><td>47</td><td>"Only one main entrance in the Macro is allowed"</td><td colspan="2" rowspan="2">The only one main entrance of Macro is:
Macro_Command function_name( )
End Macro_Command

You cannot nest or interleave Macro Functions.</td></tr>
<tr><td>48</td><td>"Too many main entrance: " 'identifier'</td></tr>
</table>

14.7 Sample Macro Code

1: Various expressions
(Arithmetic, Bitwise shift, Logic and Comparison)

```
Macro_Command main( )
int b[10]

b[0]= (400 + 400 << 2) / 401
b[1]= 22 *2 - 30 % 7
b[2]= 111 >> 2
b[3]= 403 > 9 + 3 >= 9 + 3 < 4 + 3 <= 8
b[4]= Not 8 + 1 And 2 + 1 Or 0 + 1 Xor 2
b[5]= 405 And 3 And Not 0
b[6]= 8 & 4 + 4 & 4 + 8 | 4 + 8 ^ 4
b[7]= 6 - ~ 4
b[8]= 0x11
b[9]= 409

End Macro_Command
```

2: For-Next

```
Macro_Command main( )
int a[10]
int b[10]

int i For i = 0 To 4 Step 1
   If( a[0] == 400 ) Then
     GetData(a[0] ,4x_Bin ,1,9)
     SetData(b[0] ,4x_Bin ,11,10)
   End If
Next
End Macro_Command
```

3: While-Wend, if, break, continue

```
Macro_Command main( )
int b[10]
int I

b[0]= (400 + 400 << 2) - 401 + 99
b[1]= 100
i = 5
While i == 5
    SetData(b[1] ,4x_Bin ,11,1)
    If b[1] == 100 Then
        Break
        SetData(b[0] ,4x_Bin ,12,1)
        i = 4
        Continue
        SetData(b[0] ,4x_Bin ,13,1)
    End If
Wend
End Macro_Command
```

4: Global variables and function call

```
char g

Sub int sin(int j ,int k)
int y
SetData(j ,4x_Bin ,14,1)
GetData(y ,4x_Bin ,15,1)
g = y
Return y
End Sub

Macro_Command main( )
int b[5]
int i

b[0]= (400 + 400 << 2) - 401 + 99
b[1]= 100
b[2]= 111
b[3]= 403
i = 5
While i == 5 - 20 % 3
   SetData(b[1] ,4x_Bin ,11,1)
   If b[1] == 100 Then
      SetData(b[0] ,4x_Bin ,12,1)
      i = sin(b[2],b[3] )
      SetData(g ,4x_Bin ,17,1)
      SetData(i ,4x_Bin ,16,1)
   End If
Wend
End Macro_Command
```

5: Initializing and working with arrays and PLC data

```
Macro_Command main( )
int w, i[10]
char x

For w = 0 To 9
   i[w] = 0
Next
GetData(i[0] ,4x_Bin ,1,4)
w = i[0] + i[1] + i[2] + i[3]
If w > 150 Then
    For w = 0 To 3
       i[w] = 0
    Next
    SetData(i[0] ,4x_Bin ,1,4)
    x = 0
    SetData(x ,1x_Bin ,3,1)
    GetData(i[0] ,4x_Bin ,212,2)
    x = 0
    SetData(x ,0x_Bin ,4,1)
Else
    For w = 0 To 3
       i[w] = i[w] + 2*w
    Next
    SetData(i[0] ,4x_Bin ,1,4)
End If
End Macro_Command
```

6: If-Then-Else

```
Macro_Command main( )
int j, k[10]

For j = 0 To 10
    k[j] = j
Next
If k[0] == 0 Then
    SetData(k[4],4x_Bin ,1,1)
End If
If k[0] == 0 Then
    SetData(k[4],4x_Bin ,1,1)
Else
    SetData(k[5],4x_Bin ,1,1)
End If
If k[1] == 0 Then
    SetData(k[3],4x_Bin ,2,1)
Else If k[1] == 1 Then
    SetData(k[1],4x_Bin ,3,1)
End If
If k[1] == 0 Then
    SetData(20,4x_Bin ,4,1)
Else If k[1] == 2 Then
    SetData(k[1],4x_Bin ,5,1)
Else
    SetData(k[2],4x_Bin ,6,1)
End If
End Macro_Command
```

7: Select-Case

```
Macro_Command main( )
int j, k[10]

For j = 0 To 10
    k[j] = j
Next
Select Case k[1]
    Case 1
        j = 1
    Break

    Case 2
        j = 2
    Break
End Select
SetData(j,4x_Bin ,1,1)
End Macro_Command
```

```
Macro_Command main( )
int k[10]
int j
j = 90 * 1
k[4] = 8
Select Case k[4]
    Case 1
        j = 1
    Break

    Case 3
        j = 2
End Select
SetData(j,4x_Bin ,1,1)
End Macro_Command
```

8: Array

```
Macro_Command main( )
int a[25]
int b[1]
int i

b[0] = 13
For i = 0 To b[0] Step 1
    a[i] = 20 + i * 10
Next
SetData(a[0] ,4x_Bin ,1,13)
End Macro_Command
```

9: While-Wend

```
Macro_Command main( )
char Toggleflag
int a[13]
int b[14]
int c = 4848
char i = 0
b[0] = 13

While b[0]
   a[i] = 20 + i * 10
   If a[i] == 120 Then
      c =200
      Break
   End If
   i = i + 1
Wend
SetData(c ,4x_Bin ,2,1)
End Macro_Command
```

Nested While Loops

```
Macro_Command main( )
int a[250]
int b[250]
int i = 1

a[0] = 1
b[0] = 2
While a[0]
   While b[0]
      b[i ] = 10 + i * 10
      If b[i] == 210 Then
         SetData(i ,4x_Bin ,1,2)
         b[i] = 200
         Break
      End If
      i = i + 1
   Wend
   If b[i] == 200 Then
      SetData(b[0] ,4x_Bin ,1,4)
      Break
   End If
   SetData(b[0] ,4x_Bin ,1,3)
Wend
End Macro_Command
```

10: Break and Continue

(1)

```
Macro_Command main( )
int j, K[10]

For j = 0 To 10
   If j % 2 == 0 Then
      k[j] = 5
   Else
      Break
   End If
Next
SetData(k[0],4x_Bin ,1,10)
End Macro_Command
```

(2)

```
Macro_Command main( )
int a[13]
int b[14]
int c = 4848
char i = 0

b[0] = 13
While b[0]
   a[i] = 20 + i * 10
   If a[i] == 120 Then
      c =200
      i = i + 1
      Continue
   End If
   i = i + 1
   If c == 200 Then
      SetData(c ,4x_Bin ,2,1)
      Break
   End If
Wend
End Macro_Command
```

11: Infinite loop

The loop is created because b[0] is never set to zero. Caution! Infinite loops slow down screen updates or lock up the MMI. Do not attempt to run this code in simulation mode.

```
Macro_Command main( )
int b[14]
char i = 0

b[0] = 13
While b[0]
   i = i + 1
Wend
End Macro_Command
```

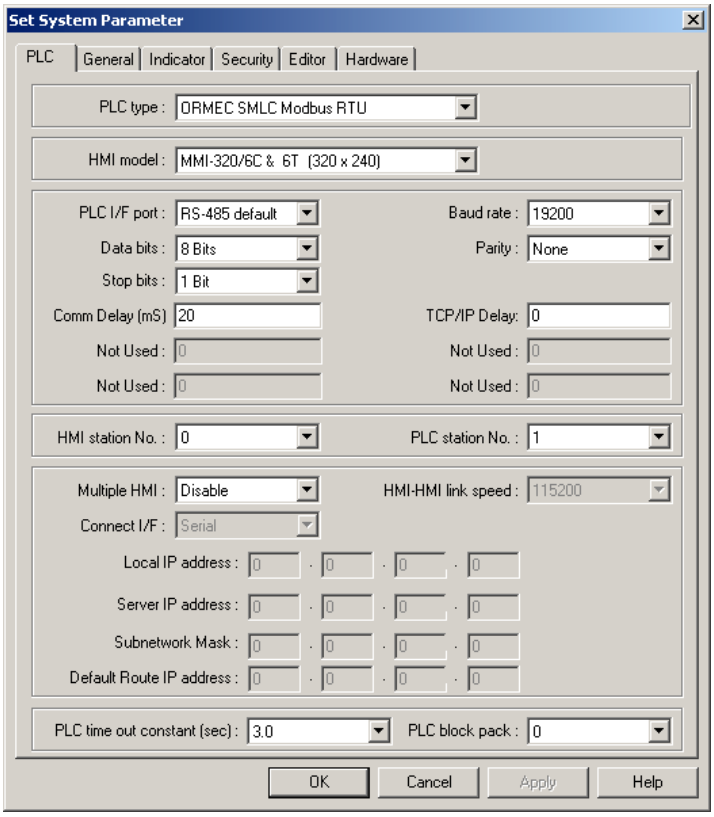
Section 5 Controller Reference

15.0 Communications Overview

A programming cable is provided with each unit. In addition to this cable, a cable is needed to connect the HMI to your Controller (PLC). This section provides pin out diagrams so that you can make your own cable. In some cases, a ready made 6' (2m) cable is available. The available cables have their part number listed with the wiring diagrams.

15.1 Communications settings

Once the cable is made and the HMI is connected to the PLC communications can begin. It is up to the programmer to make sure the settings in the PLC are correct and that they match the settings in the MMI. The HMI settings are made through the EasyBuilder Software. Run EasyBuilder, open or start a new project and go to the **Edit|System Parameters…** Menu selection.



Select the **PLC type** from the drop down menu. If the **PLC type** is not shown, consult the factory for availability. Make the appropriate selections for your particular situation.

Serial I/F port: Selects the type of communications and thus the HMI port to which the PLC is connected.
Baud rate: Communication speed, this must match the PLC setting.
Data bits, Stop bits, Parity: Communications attributes that also must match the PLC settings.
Parameter 1 ~ 6: Set additional parameters according to PLC type specification sheets.

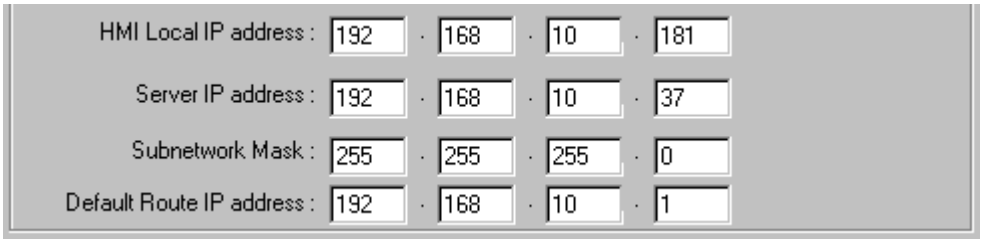
HMI station No.: The ID number of the HMI when used with PLCs that allow multiple devices connected to its port.
PLC station No.: The ID number of the PLC. Used when multiple PLCs have the capability to be on line with other devices.

Multiple HMI: Allows more than one HMI to be connected to one PLC. Enable as a Master or a Slave, depending on connection, or Disable as needed.
HMI-HMI link speed: This is used only when multiple HMIs are connected to a single PLC. Faster link speeds may increase screen updates but it is affected by distance of link and electrical interference.

Note: All linked MMI's must be set to the same speed. When an HMI is set as "slave", it ignores the communication parameters connected with the PLC (Serial port I/F, Baud rate, etc.).

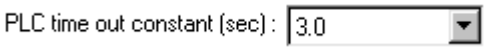
Connect I/F: Select the unit to unit connection type. **Ethernet** is available for units with the optional Ethernet port. For all others, use the **Serial** interface (I/F) selection.

Ethernet IP settings: These settings are used whenever Ethernet communications is selected for HMI to MMI or MMI to PLC connections. A working knowledge of TCP/IP networking and terminology is necessary to implement Ethernet connections.



HMI Local IP address: These fields are the IP address of the HMI unit.
Server IP address: The IP address of the PLC or Slave MMI.
Subnetwork Mask: The mask for the network where the HMI and PLC or Slave are located.
Default Route IP address: This is used for HMI to PLC communications and refers to network server IP settings.

PLC time out constant (sec): This setting determines how long the HMI waits for a response from the PLC before retry. The range is from 3.0 to 25.5 seconds. This setting is important when the PLC is normally slow to respond or if several MMI's are linked to one PLC.



PLC block pack: Used to determine how the unit communicates to the controller. By increasing this number, larger blocks of registers can be fetched from the controller. In some cases, this speeds the update of information on the display. *Hint: When creating a display, it is advised that consecutive registers be used whenever possible.*

Note: **Communication disconnection handling process**
Two conditions cause PLC-HMI communications problems.
The PLC does not respond
The PLC gives an error code response

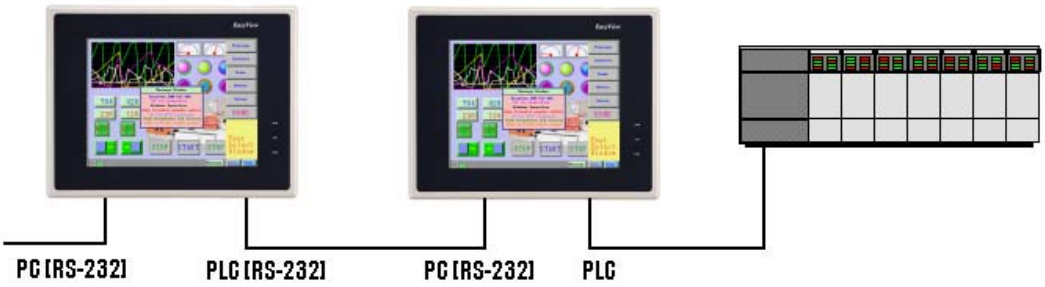
If the HMI detects communication problems, it responds according to the state of local bit **LB9055**.
Bit is OFF, **0**: Abort: Any command to write data to the PLC is deleted.
Bit is ON, **1**: Retry: Keeps on writing command to the PLC.

15.2 Master-Slave Configuration

The HMI supports a master - slave communications. One HMI is connected directly to the PLC and configured as the Master. All the other MMIs are connected in series to it and are configured as Slaves. In theory there is no limitation to the number of MMIs on a chain, however response time gradually decreases when more than three MMIs are linked together. The HMI are configured with the EasyBuilder software to be the Master or a Slave.

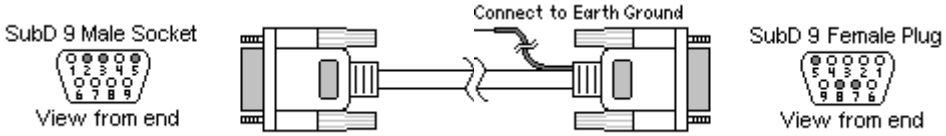
Serial Linking is accomplished by connecting the PLC [RS232] port of the slave to the RS232 side of the PLC/PC 232/485 connector of the master or successive unit. Ethernet linking is via crossover cable or through a hub with standard Ethernet 10BaseT cabling.

Serial wiring between two MMIs



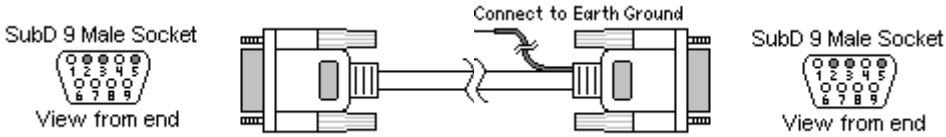
Case 1: Slave to Master, connecting PLC[RS232] directly to PC[RS-232]/PLC[RS485] combination port.

Connect to SLAVE HMI PLC[RS-232] port Cable has D-SUB Male end	**Connect to MASTER HMI PC[RS-232] port Cable has D-SUB Female end**
2 TxD →	8 RxD
3 RxD →	7 TxD
5 GND →	5 GND
	Shield → Earth Ground



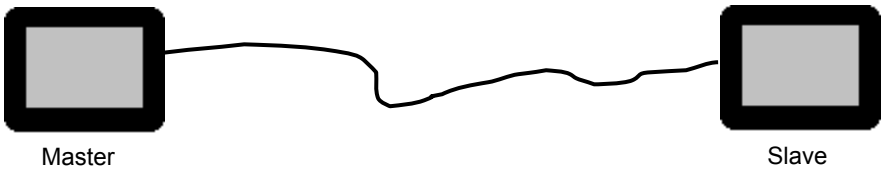
Case 2: Slave to Master, connecting PLC[RS232] to PC side of split download cable (MT5_PC) with use of Male to Male Null Modem cable.

Connect to SLAVE HMI PLC[RS-232] port Cable has D-SUB Male end	**Connect to MT5_PC to MASTER HMI PC[RS-232] port Cable has D-SUB Male end**
2 TxD →	3 RxD
3 RxD →	2 TxD
5 GND →	5 GND
	Shield → Earth Ground

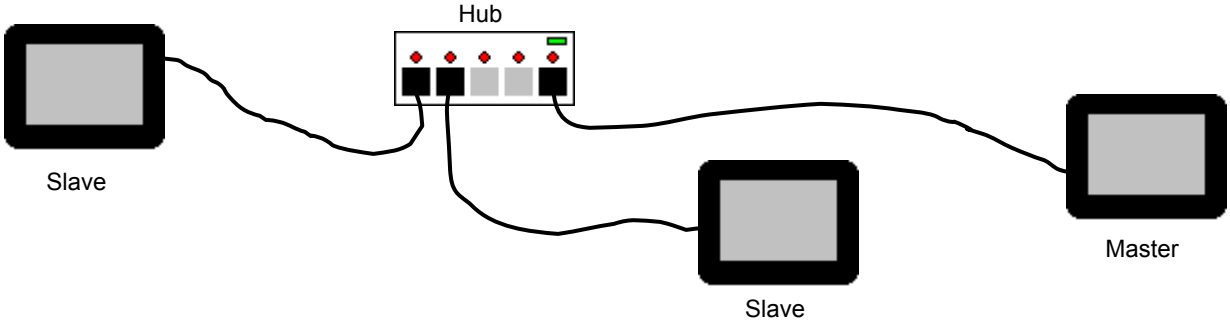


Case 3 Optional Ethernet Connections: Units equipped with the optional Ethernet port can be connected via Ethernet to a Master and Slaves using standard 10T Ethernet cables. Slave to Master, connect the Ethernet ports with a standard (Category 5) Ethernet 10T crossover cable (RJ-45 to RJ45). On the other hand, use standard cables with routing passing through Hubs and Servers as needed.

Connecting a Master directly to a Slave using 10T Crossover cable



Connecting a Master through a Hub to Slaves



15.3 Drivers

There are two types of drivers used with EasyBuilder.

15.3.1 Standard Drivers

Standard Drivers are those that are built into the EasyBuilder software package. To access the register ranges for these drivers, use the **PLCAddressView.exe** utility. The following are brief reference sheets for these drivers. It is understood that the programmer is familiar with the Programmable Controller and its operation. In most cases, the communications parameters listed in the reference are the default settings. Not all drivers listed in the System Parameters have reference sheets.

15.3.2 External Drivers

EasyBuilder version 2.0.2 and later now supports enhanced external drivers. These are drivers developed for special applications or extended support over and above the list of Standard PLC drivers.

If an external driver is not installed on your computer, it can be downloaded from the KEP website (www.kep.com) downloads area.

Steps to Activate a Driver

Create a Subdirectory under the EasyBuilder main directory named ***drivers***. If the directory already exists then there is no need to create a second directory.

Copy the driver file into the ***drivers*** directory. Driver files end with the *.pds extension.

Start EasyBuilder and go to the **System Parameters, PLC** tab. The new driver should be listed along with all of the standard drivers. Drivers are listed in alphabetical order.

16.0 Driver Specifications

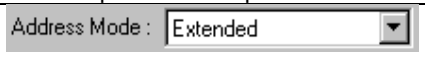
The following are brief reference sheets for the drivers. It is understood that the programmer is familiar with the Programmable Controller and its operation. In most cases, the communications parameters listed in the reference are the default settings.

16.30 MODICON MODBUS RTU / MODBUS RTU(485 2W)

16.30.1 HMI Setting

Communication parameters must be set the same as those of the PLC. The following settings are recommended.

Parameter	Recommended Setting	Notes
Serial I/F port	RS-232, RS485 2W	Options: RS-232, RS-485 default/2W/4W, Ethernet – Dependant on PLC hardware
Baud rate	9600	Options: 9600, 19200, 38400, 57600, 115200 - Must match PLC port setting
Parity	Even	Options: Odd, Even, None - Must match PLC port setting
Data bits	8	Options: 7, 8 - Must match PLC port setting
Stop bits	1	Options: 1, 2 - Must match PLC port setting
HMI station No.	0	Not used
PLC station No.	0	Options: 0-255 - As per PLC setting if not using Extended Addressing
Comm. delay (10msec)	0	Request delay: 0~999 @10 milliseconds. Use when the PLC requires a delay between requests. Not implemented in simulation mode.

MODBUS RTU(485 2W) supports Extended Address mode. 

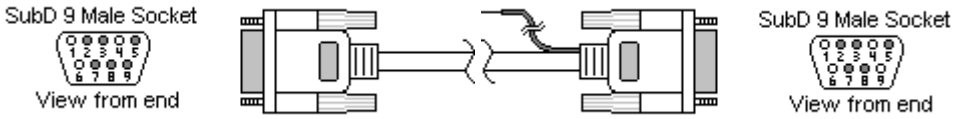
16.30.2 PLC Setting

Communication parameters must be the same as those of MMI. PLC ID number must match setting in MMI. PLC must be configured as a Modbus Slave.

16.30.3 Wiring

16.30.3.2 Wiring for cable connections for ORMEC SMLC

HMI PLC[RS-232]	MODBUS COMPATIBLE CPU RS-232 port
2 TD →	2 RD
3 RD →	3 TD
5 GND → 9 →	5 GND
7 RTS → 8 CTS	7 RTS → 8 CTS
	Shield → Earth Ground



16.30.4 Device address

Bit/Word	Type	Address Format	Range	Notes
B	1x	ddddd	ddddd: 1~9999	Input bits (Read only)
B	0x	ddddd	ddddd: 1~9999	Internal Coils
B	3x_bit	ddddbb	dddd: 1~9999 bb: 0~15	Internal Input Coils (Read only)
B	4x_bit	ddddbb	dddd: 1~9999 bb: 0~15	Internal Register Coils
W	3x	ddddd	ddddd: 1~9999	Input Registers (Read only)
W	4x	ddddd	ddddd: 1~9999	Internal Registers

Note: The Controller's memory range may vary within the range supported by the driver. Use appropriate addresses.

16.32 MODBUS RTU TCP/IP

16.32.1 HMI Setting – Ethernet port hardware option must be present

Communication parameters must be set the same as those of the PLC. The following settings are recommended.

Parameter	Recommended Setting	Notes
Serial I/F port	Ethernet	
HMI station No.	0	Not used
PLC station No.	1	Options: 0-255 - As per PLC setting
Comm. delay (10msec)	0	Request delay: 0~999 @10 milliseconds. Use when the PLC requires a delay between requests. Not implemented in simulation mode.
Multiple HMI	Master	
Connect I/F	Ethernet	
Local IP address		The IP address of the HMI unit
Server IP address		The IP address of the PLC
Subnetwork Mask		IP Mask
Default Route IP address		The IP address of the main server

16.32.2 PLC Setting

PLC ID number must match setting in MMI

16.32.3 Wiring

Use the same wiring as the MODBUS RTU driver. CAT-5 class through cable to Hub or crossover cable to Controller.

16.32.4 Device address

Bit/Word	Type	Address Format	Range	Notes
B	1x	ddddd	ddddd: 1~9999	Input bits (Read only)
B	0x	ddddd	ddddd: 1~9999	Internal Coils
B	3x_bit	dddddb	dddd: 1~9999 bb: 0~15	Internal Input Coils (Read only)
B	4x_bit	dddddb	dddd: 1~9999 bb: 0~15	Internal Register Coils
W	3x	ddddd	ddddd: 1~9999	Input Registers (Read only)
W	4x	ddddd	ddddd: 1~9999	Internal Registers
W	5x	ddddd	ddddd: 1~9999	4x type with Double Words swapped

Note: The Controller's memory range may vary within the range supported by the driver. Use appropriate addresses.

NOTE:

Address type "5x" is mapped to 4x Internal Registers. 5x register types are the same as "4x" except "5x" double word data is swapped.

For example,
If 4x has the following information

Address	1	2	3	4	5	6
Data in word	0x123	0x267	0x3	0x4	0x5	0x0
4x Data	0x2670123		0x40003		0x5	

For 5x, it becomes

Address	1	2	3	4	5	6
Data in word	0x123	0x267	0x3	0x4	0x5	0x0
5x Data	0x1230267		0x30004		0x50000	