Modicon M340

FactoryCast User Manual

(Original Document)

12/2018

www.schneider-electric.com

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

Registered Trademarks

Microsoft®, Windows®, Silverlight® and Expression Blend® are registered trademarks of Microsoft Corporation.

Table of Contents

Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

BEFORE YOU BEGIN

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

⚠ WARNING

UNGUARDED EQUIPMENT

- Do not use this software and related automation equipment on equipment which does not have point-of-operation protection.
- Do not reach into machinery during operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

START-UP AND TEST

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check be made and that enough time is allowed to perform complete and satisfactory testing.

⚠ WARNING

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

OPERATION AND ADJUSTMENTS

The following precautions are from the NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer’s instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer’s instructions and the machinery used with the electrical equipment.
- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

About the Book

At a Glance

Document Scope

This user guide introduces the FactoryCast for Modicon M340 software package, which is used to customize a Web site on the embedded server module. The site can be accessed via a browser to view and modify data from a Modicon M340 programmable logic controller (PLC). FactoryCast provides the Web pages and Java applets needed to view run-time data from a controller.

FactoryCast offers 2 levels to customize the default Web site.

- Create a Web-enabled database to view and modify the run-time values of symbols (variables) and direct addresses.
- Add Web pages to the site.

FactoryCast provides the tools needed to maintain a Web site on the embedded server, including methods for downloading, backing up, and restoring files.

Validity Note

This documentation is valid for EcoStruxure™ Control Expert 14.0 or later.

Related Documents

Title of Documentation	Reference Number
Web Designer Objects Library for Microsoft Expression Blend®	S1A75812 (English)
Modicon M340 for Ethernet Communications Modules and Processors User Manual	31007131 (English), 31007132 (French), 31007133 (German), 31007134 (Spanish), 31007493 (Chinese), 31007494 (Italian)
Web Designer for Modicon M340, Premium and QuantumUser Manual	35016149 (English), 35016150 (French)

You can download these technical publications and other technical information from our website at https://www.schneider-electric.com/en/download

Product Related Information

⚠ **WARNING**

UNINTENDED EQUIPMENT OPERATION

The application of this product requires expertise in the design and programming of control systems. Only persons with such expertise should be allowed to program, install, alter, and apply this product.

Follow all local and national safety codes and standards.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Chapter 1
Introduction to FactoryCast

Purpose

This chapter describes FactoryCast for Modicon M340, including its functions, components and system requirements.

What Is in This Chapter?

This chapter contains the following topics:

About FactoryCast

Overview

FactoryCast is a software package that you use to customize a Web site on the Embedded Web Server module. The site can be accessed via a browser to view and modify data from a Modicon M340 programmable logic controller (PLC).

FactoryCast Web Site

FactoryCast provides the Web pages and Java applets you need to view run-time data from your controller. You can use the FactoryCast default Web site simply by configuring the module and accessing it with a browser over the intranet.

Custom Web Site

FactoryCast gives you two levels to customize the default Web site embedded in the PLC module.

- Create a Web-enabled database to view and modify the run-time values of PLC symbols (variables) and direct addresses
- Add your own Web pages to the site

Site Maintenance

FactoryCast also provides the tools you need to maintain your Web site on the Embedded Server, including ways to download, backup, and restore files.

FactoryCast components

Overview

The FactoryCast components are:

- FactoryCast Server
- Control Expert
- Web Designer for Modicon M340
- Web browser

Description of the FactoryCast Components

For Modicon M340, the FactoryCast components are integrated as follows:

This Component...	Is Used To...
BMX NOE 0110	Store the Web site on its Embedded Server.
Unity Pro 4.0 and later	Program the controller.
Web Designer for Modicon M340	Configure the Web site, download data and maintain the site on the server.
Web browser (IE, Firefox...)	View the Web site.

NOTE: Unity Pro is the former name of Control Expert for version 13.1 or earlier.

NOTE: FactoryCast requires the installation of the following software:

- Sun Java Virtual Machine (JVM) version 1.6 or above. The installation software for the Sun JVM is supplied on the CD.
- Silverlight 4 plug-in. You can download the installation software for the Silverlight 4 plug-in at the www.microsoft.com web site.

FactoryCast Server

Overview

FactoryCast Server consists of HTTP and FTP servers embedded in a Modicon M340 Ethernet option module.

How it Works

The servers contain a default set of diagnostic Web pages and Java applets. The user can add custom Web pages and applets for specific applications.

Description

The FactoryCast server is accessible via Ethernet TCP/IP 10/100BaseT twist pair. You can configure the Web server using Web Designer for Modicon M340. The BMX NOE 0110 module reserves 64 Mb of its memory for customization of the website.

Available Features

The FactoryCast Server offers run-time monitoring and diagnostics via predefined Web pages. The features available include:

- diagnostic tools:
  - a Rack Viewer that displays the status and configuration of the controller, embedded server module, and other option modules and I/O modules.
  - an Alarm Viewer that displays application events, when diagnostic properties are enabled in the PLC application.
- monitoring tools:
  - a Data Editor that helps you view and modify variables (symbols and direct addresses).
  - a Graphics Editor that you can use to create and display graphical objects. Each graphical object can be linked to a variable or address in the embedded server.
  - a PLC Program Viewer that lets you monitor the PLC program during run-time.

Web Designer for Modicon M340

Overview

You can use Web Designer for Modicon M340 to configure and maintain your Web site.

You also can use it to create a Web-enabled database of variables (symbols) and direct addresses, which can be viewed and modified during run-time over the Web.

Another function of Web Designer for Modicon M340 is to provide general FTP capabilities enabling you to load Custom Web pages to the FactoryCast module.

NOTE: Web Designer for Modicon M340 configuration and use is not detailed in this documentation. Refer to the *Web Designer for Modicon M340 User Manual* for more information.

Configuring a Site

Web Designer for Modicon M340 enables you to:

- Set security, including passwords and read/write protection
- Add your own Web pages, images and Java applets to a site
- Download and upload files to the Embedded Server

Creating a Database

The configuration tool allows you to create a Web-enabled database using symbols (variables) and direct addresses from your Control Expert database. You can use this Web-enabled database to view and modify the value of symbols (variables) and direct addresses while the controller is running.

Maintaining a Site

Web Designer for Modicon M340 enables you to backup files, restore files and, if necessary, reflash files to the Embedded Server.

Opening Applications from Interim Target-Patched Versions of Web Designer

Some previous versions of Web Designer were patched with a file to target an interim exec firmware version such as BMX NOE 0110 version 4.51. Applications created with such versions of Web Designer might not open as expected in the latest version of Web Designer.

If your application was created with a version of Web Designer that was patched for an interim firmware version, and does not open as expected in a later Web Designer version, perform the following steps.

Step	Action
1	Open the folder containing the project. For instance:. `C:\Documents and Settings\user name\My Documents\Schneider Electric\Web Designer\projects\project name` where `user name` is your Windows login name and `project name` is the name of the project you want to edit.
2	Use a text editor such as Wordpad to open the file *targetversion.xml*. Locate the lines that show the target firmware version. For instance, in the following example, the target firmware version is 4.51, as indicated by the values of `target firmwareVersion` and `version`: `<?xml version="1.0" encoding="UTF-8" ?>- <Targets>      <target firmwareVersion="v4.51" name="TSX ETG 3021-Target0" version="4.51" /></Targets>`
3	Change the values of `target firmwareVersion` and `version` to reflect the latest Web Designer version number. For instance, in the following example, the latest Web Designer version is 5.0. `<?xml version="1.0" encoding="UTF-8" ?>- <Targets>      <target firmwareVersion="v5.0" name="TSX ETG 3021-Target0" version="5.0" /></Targets>`
4	Save the *targetversion.xml* file.

Variable Consistency Management

Overview

FactoryCast modules allow you to create Web-enabled pages using symbols (variables) coming from your Control Expert application. At runtime, these symbols are animated and communicate in real-time with the PLC.

This can lead to inconsistency between the variables exchanged with the PLC and the variables of the PLC application (i.e. modification of the Control Expert application). In this case, the real-time communication between the Web page and the PLC is stopped.

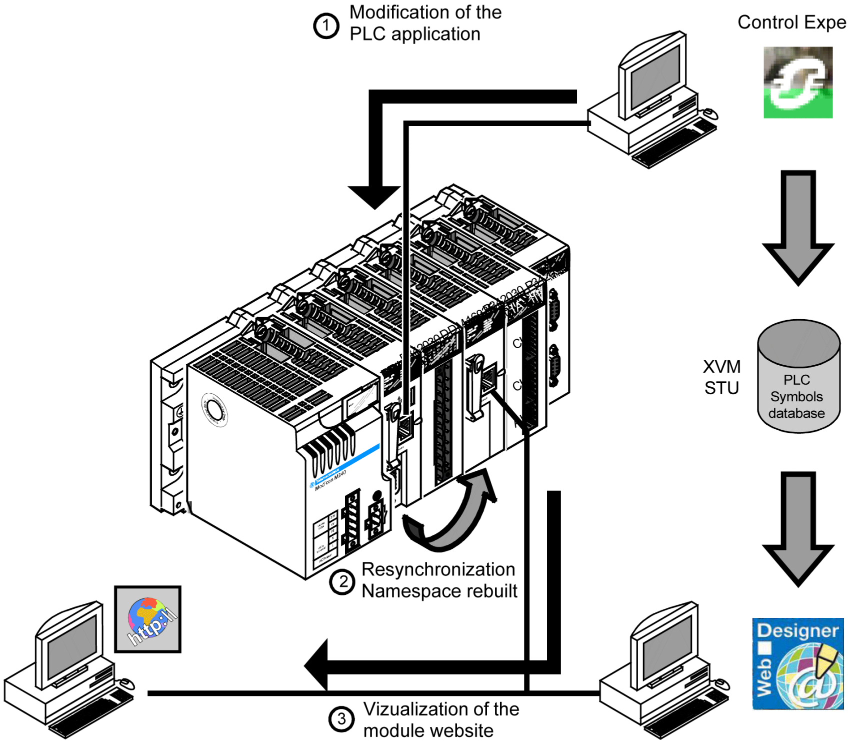
Originally, when an inconsistency was detected between the PLC and the FactoryCast applications, you had to resynchronize manually the FactoryCast application using an intermediate export file generated by Control Expert.

Thanks to the Data dictionary feature of Unity Pro 4.0 and the Variable Consistency Management of FactoryCast 3.5, the FactoryCast application is able to resynchronize with the PLC and to restart the real-time communication automatically.

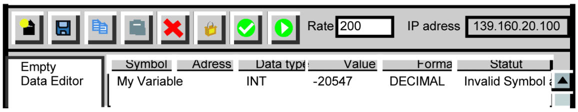
NOTE: Unity Pro is the former name of Control Expert for version 13.1 or earlier.

Synchronization Process

The following figure shows you the synchronization process:


1 Modification of the PLC application
Control Expert
XVM
STU
PLC Symbols database
2 Resynchronization Namespace rebuilt
http://
Designer
Web
3 Vizualization of the module website

DATA EDITOR



Rate 200 IP adress 139.160.20.100

Empty Data Editor

Symbol	Adress	Data type	Value	Forma	Statut
My Variable		INT	-20547	DECIMAL	Invalid Symbol a

The following procedure assumes you already have:

- transferred your Control Expert application into the PLC with the Data Dictionary feature *(see EcoStruxure™ Control Expert, Operating Modes)* enabled (refer to the following note).
- imported the symbols from your Control Expert application to Web Designer for Modicon M340.
- transferred the Web Designer for Modicon M340 project to the FactoryCast module.

NOTE: Enable the Data Dictionary feature in your Control Expert project in order to use automatic synchronization. This feature increases the size of your application. Refer to the Memory Usage page *(see EcoStruxure™ Control Expert, Operating Modes)* to know the size of your application.

Step	Software	Description
1	Unity Pro 4.0 or later	Modification of the Unity Pro application that runs on the PLC.
2	-	The FactoryCast module database synchronizes automatically with the PLC database using the data dictionary feature of Control Expert.
3	Web browser	Visualization of the Web pages.

NOTE: Unity Pro is the former name of Control Expert for version 13.1 or earlier.

NOTE:

- Variable Consistency Management operates between the PLC and the FactoryCast module. Therefore, you need to synchronize manually the Web Designer for Modicon M340 namespace after modification of the PLC application.
- In case of a modification of the PLC program that cannot be automatically synchronized (i.e. creation of new variables), you can force the synchronization of the Namespace by clicking the ‘Reload PLC symbols’ (▶) button of the Data Editor toolbar.

Chapter 2
Web Site Security

Overview

Do not use FactoryCast modules to support critical functions. Before configuring your Web site develop a security plan. Unlike default Web site data, data from a custom Web site is write enabled.

Pay particular attention to the people with access rights to the site and to the data that can be changed. This chapter describes security events and presents security mechanisms accessible to users of the Web utility.

What Is in This Chapter?

This chapter contains the following topics:

Internal Security

Overview

FactoryCast products provide two mechanisms so that only authorized users view and modify your data.

- Password entry
- Access restrictions

⚠ **WARNING**

UNINTENDED OPERATION

Keep strict control of access to the embedded server:

- Change passwords monthly.
- Do not use simple user names and passwords.
- Disable default passwords before commissioning the module.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Password Protection

Although you may add unprotected Web pages to the site, the default Web pages and any other pages to which you want to restrict access can only be viewed by users who supply the correct user name and password *(see page 59)*.

Restricting Access to Web Pages

When you create a Web site and you want to restrict access to it, place it in the folder called *secure*. The uploading of the custom Web site is subject to security conditions linked to a FTP password *(see page 61)*.

External Security

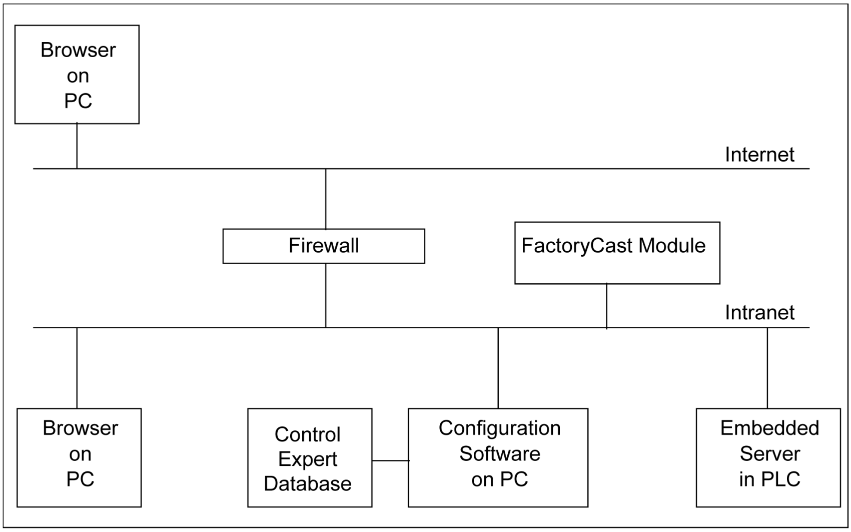
Overview

If your network has been configured to enable users to consult your Internet site, your security system is the same as that of an intranet site, only you have an additional security measure: a firewall.

Architecture of a Firewall

A firewall forms a gateway between Internet and your embedded server. You can use a firewall to restrict or forbid access to your website.

A firewall can be configured to authorize network connections to a limited range of ports, or to authorize traffic to or from certain IP addresses.



Types of Firewalls

There are two types of firewalls:

- Network firewalls
- Application firewalls

Network Firewalls

Network firewalls are often installed between the Internet and a single entry point to an intranet or internal protected network.

Application-Level Firewalls

An application firewall works for an application, for example FTP. It intercepts all traffic sent to this application, and decides whether or not to transmit this traffic to the application. Application firewalls are located on individual host computers.

Firewall Configuration

Web Designer for Modicon M340 uses HTTP, FTP and Schneider Electric Modbus application protocol (MBAP) to access embedded server pages and files. If you want viewers to be able to access your site from the Internet and your embedded server is protected by a firewall, configure the firewall to authorize HTTP, FTP and MBAP traffic.

Port	Protocol	Access to...
21	FTP	Protected embedded server files
Higher than 1024		
80	HTTP	Web pages
502	MBAP	Operational data

NOTE:

- The default FTP name and password are USER/USER.
- The FactoryCast client follows the "Firewall Friendly FTP" standard, RFC 1579. It issues an FTP PASV command to the FactoryCast server before any attempt to establish an FTP data connection.
- The online mode of the configuration tool is not operational if the module is behind a firewall. The ports in this mode are dynamically assigned.

Variable Access Security, Symbol, Direct Address

Presentation

Users who enter the data edition write password *(see page 58)* can only modify variables (symbols) and direct addresses that are write-enabled. When you create a Web-enabled database of variables and direct addresses, you can designate each element as read-only or write-enabled.

Unauthorized or incorrect modifications made to symbols and direct addresses may have undesirable effects or even change the behavior of your application in ways that may be unintended or even hazardous.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Carefully select the symbols and direct addresses you authorize to be modified online.
- Do not authorize online modifications of critical process variables.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Chapter 3
Default Web Site for Modicon M340

Overview

When you receive the PLC module, it already contains a default Web site with the Rack Viewer and the Run-Time Data Editor pages pre-loaded.

You can view these pages simply by installing the module and configuring its IP address. To access the site, type the IP address of the module in your browser and enter the default user name and password: ‘USER/USER’.

This section describes the pages in the default Web site.

What Is in This Chapter?

This chapter contains the following sections:

Section 3.1
Modicon M340 Home Page

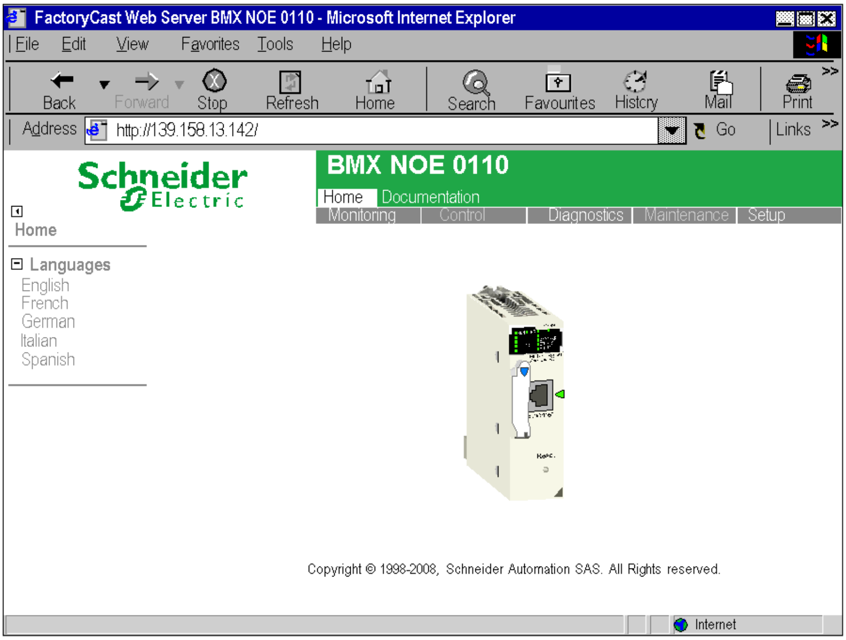
Modicon M340 Home Page

Overview

The visitor accesses the Modicon M340 home page by entering the IP address of the module in his web browser. No password is required to display this page.

Home Page

The Modicon M340 home page looks like this:



Address: http://139.158.13.142/

Copyright © 1998-2008, Schneider Automation SAS. All Rights reserved.

Links

From the Modicon M340 home page, you can access the following pages:

- Monitoring *(see page 32)*
- Diagnostics *(see page 34)*
- Setup *(see page 56)*
- Documentation
- Foreign language links exist for French, German, Italian and Spanish versions

The visitor will have to supply a user name and a password to access the services proposed in these pages.

Section 3.2
Monitoring Page

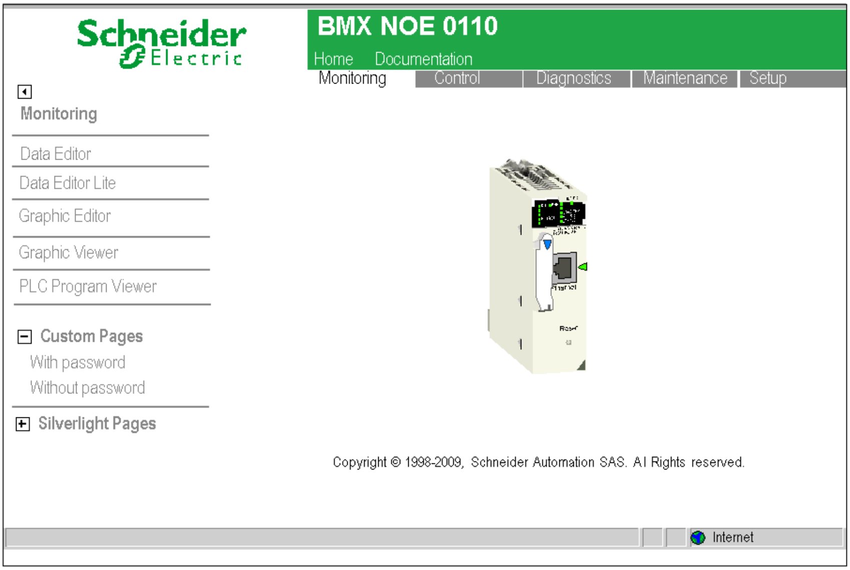
Monitoring Page

Monitoring Page

This page lists the various viewer services supported by the default web site of the module and provides links for accessing the services you require.

Illustration

The **Monitoring** page looks like this:



Copyright © 1998-2009, Schneider Automation SAS. Al Rights reserved.

To access the service you require, click on a link.

- Data editor *(see page 70)*: for creating variable data tables, so as to be able to determine their value when the table is animated.
- Data editor Lite *(see page 85)*: it is a light version of the Data Editor. It provides the same interface than the standard Data Editor but with some restrictions on accessible data type.
- Graphic editor *(see page 95)*: for creating graphics, so as to be able to determine the value of variables when the graphic is animated.

- Graphic viewer *(see page 135)*: to access the graphic pages created by the user using the graphic editor.
- PLC program viewer *(see page 137)*: Display Control Expert programs in run mode using a web browser.
- Password-protected custom pages *(see page 145)*: For viewing protected screen pages created by the user.
- Custom pages without password protection *(see page 145)*: For viewing unprotected screen pages created by the user.
- Silverlight pages *(see page 208)*: For viewing user-defined pages created with *Microsoft Expression Blend®* and added to the web site.

Section 3.3
Diagnostics Page

Overview

This section describes the different links of the **Diagnostics** page.

What Is in This Section?

This section contains the following topics:

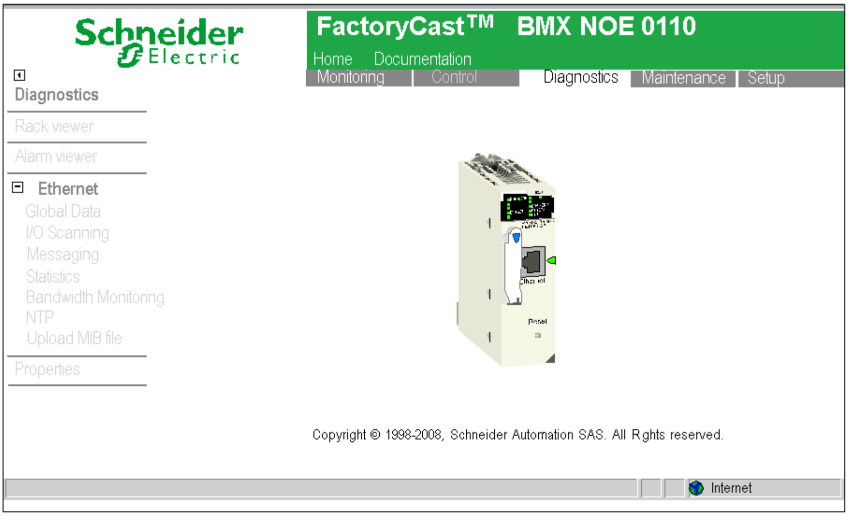
Diagnostics Page

Diagnostics Page

This page lists the various services supported by the default website of the module and provides links for accessing the services you require.

Illustration

The **Diagnostics** page looks like this:



Links

To access the service you require, click on a link:

- Rack viewer *(see page 36)*
- Alarm viewer *(see page 63)*
- Global Data *(see page 49)*
- I/O Scanning *(see page 50)*
- Messaging *(see page 51)*
- Statistics *(see page 52)*
- Bandwidth Monitoring *(see page 52)*
- NTP *(see page 53)*
- Upload MIB File *(see page 54)*
- Properties *(see page 55)*

Rack Viewer Page

Overview

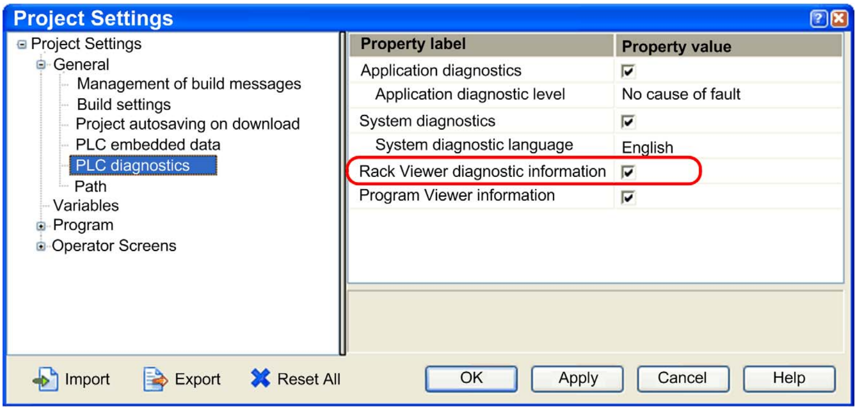
The **Rack Viewer** page displays the current configuration of your network, as either a Microsoft Silverlight (plugin) or a Java (plugin) page. The technology used to display your network depends on both:

- how you configured your Control Expert application **Project Settings**, and
- the browser-supporting software that is installed on your PC

Project Settings

To enable the display of Silverlight pages in **Rack Viewer** page, make the following project setting configuration:

Step	Action
1	In Control Expert, select **Tools** → **Project Settings...** to open the **Project Settings window**.
2	Select **Project Settings** → **General** → **PLC diagnostics** in the tree control on the left side of the window.
3	Select **Rack Viewer diagnostics information** as displayed below, then click **OK**.



Project Settings

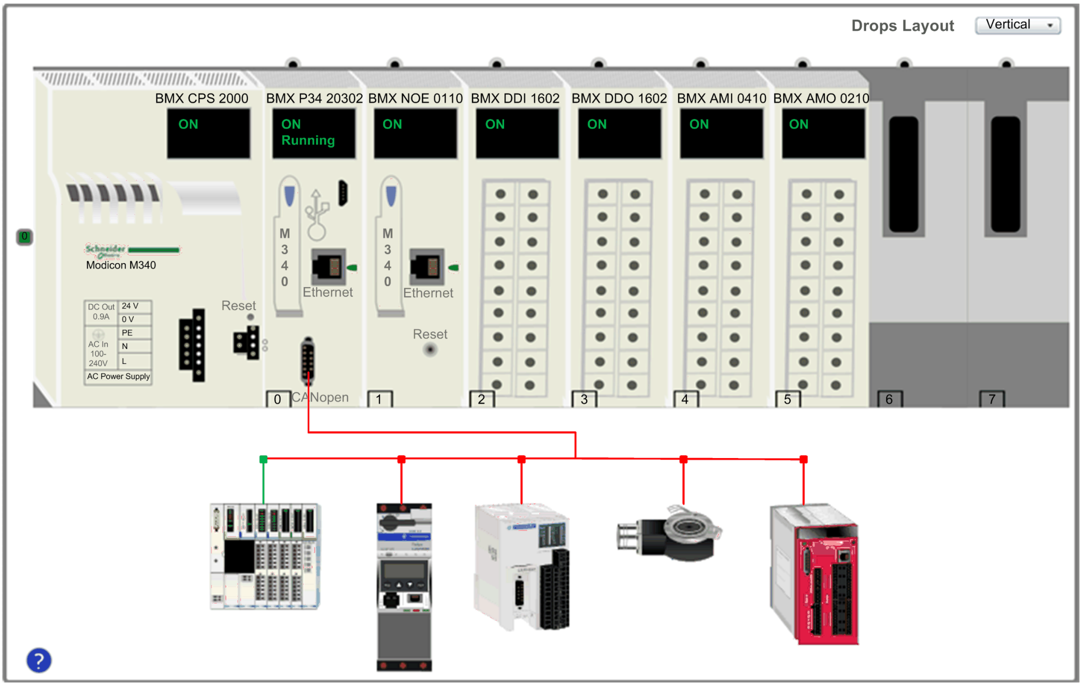
- Project Settings
  - General
    - Management of build messages
    - Build settings
    - Project autosaving on download
    - PLC embedded data
    - PLC diagnostics
    - Path
  - Variables
  - Program
  - Operator Screens

Property label	Property value
Application diagnostics	☑
Application diagnostic level	No cause of fault
System diagnostics	☑
System diagnostic language	English
Rack Viewer diagnostic information	☑
Program Viewer information	☑

NOTE: If you do not select **Rack Viewer diagnostic information**, as described above, the Java version of the **Rack Viewer** page will display.

Sample Silverlight Page

Here is an example of the **Rack Viewer** page presented in Silverlight:



Drops Layout Vertical

The Silverlight version of the **Rack Viewer** presents the configured network racks and modules in one screen. In the above example, the network consists of a local main rack connected to a CANopen network. CANopen devices that physically exist on the network are depicted with green connections; devices that **Rack Viewer** could not detect are depicted with red connections.

Use the mouse wheel to zoom in and out of the **Rack Viewer** page.

Hold down the left mouse button and move the mouse to drag the display.

Each Modicon M340 module displays the `Health Bit` status for that module:

- ON (in green) indicates that the module is functioning properly.
- OFF (in red) indicates that the module is not functioning properly

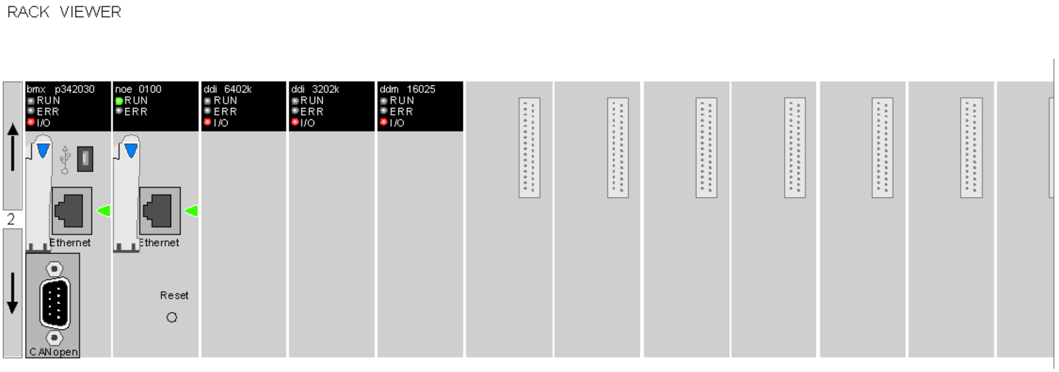
Click on a Modicon M340 module to open a pop-up page with detailed information about that module.

Click on the controller’s CANopen port to o pen a pop-up page describing CANopen network status *(see page 41)*.

Refer to the description of sample **Rack Viewer** pages for Modicon M340 modules *(see page 40)* for more information about Silverlight® pages.

Sample Java Page

Here is an example of a Rack Viewer page.



Accessing a Rack:

The left side of the Rack Viewer enables you to navigate between the racks:

- Click the up arrow to access the preceding rack
- Click the down arrow to access the next rack
- The rack number is displayed in the middle

Module Data:

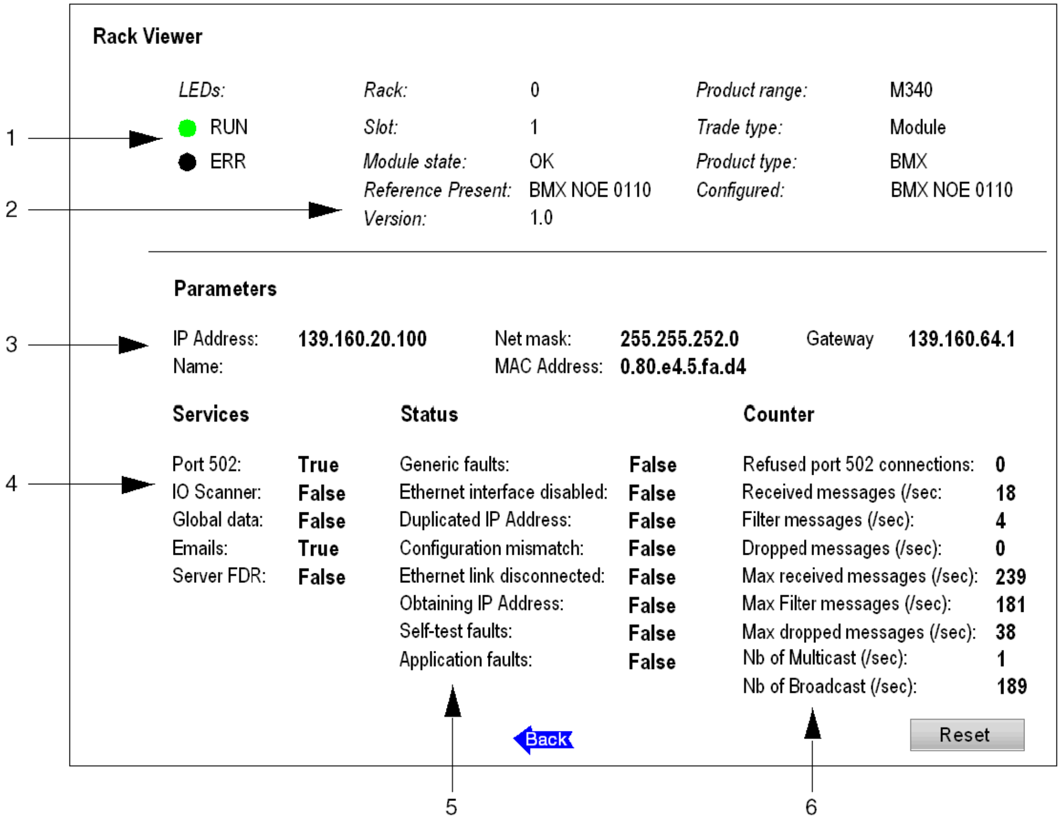
The following information is provided for each module displayed in the rack:

- A preview of the modules
- Status of the modules with the diagnostic LEDs

Module Status:

Click a module icon to obtain detailed information about it. The content of the page displayed depends on the type of module (CPU, Ethernet, TOR).

The status page for an Ethernet device looks like this:



Rack Viewer

LEDs:	*Rack:*	0	*Product range:*	M340
● RUN	*Slot:*	1	*Trade type:*	Module
● ERR	*Module state:*	OK	*Product type:*	BMX
	Reference Present:	BMX NOE 0110	*Configured:*	BMX NOE 0110
	Version:	1.0		

Parameters

IP Address:	**139.160.20.100**	Net mask:	**255.255.252.0**	Gateway	**139.160.64.1**
Name:		MAC Address:	**0.80.e4.5.fa.d4**		

Services		**Status**		**Counter**	
Port 502:	**True**	Generic faults:	**False**	Refused port 502 connections:	**0**
IO Scanner:	**False**	Ethernet interface disabled:	**False**	Received messages (/sec:	**18**
Global data:	**False**	Duplicated IP Address:	**False**	Filter messages (/sec):	**4**
Emails:	**True**	Configuration mismatch:	**False**	Dropped messages (/sec):	**0**
Server FDR:	**False**	Ethernet link disconnected:	**False**	Max received messages (/sec):	**239**
		Obtaining IP Address:	**False**	Max Filter messages (/sec):	**181**
		Self-test faults:	**False**	Max dropped messages (/sec):	**38**
		Application faults:	**False**	Nb of Multicast (/sec):	**1**
				Nb of Broadcast (/sec):	**189**

1 Status of the modules with the diagnostic LEDs
2 General information about the product and its references
3 Network parameters
4 Available services
5 Status of the bus
6 Diagnostic counters

Rack Viewer (Silverlight) pages

Introduction

Many Modicon M340 FactoryCast modules present diagnostic Silverlight® web pages, including pages for controllers, communication modules, I/O modules, motion modules, counting modules, and the ASI bus. All of these modules can be selected, and their status displayed, by clicking on a module in the **Rack Viewer**.

This topic presents examples of typical Silverlight module status pages.

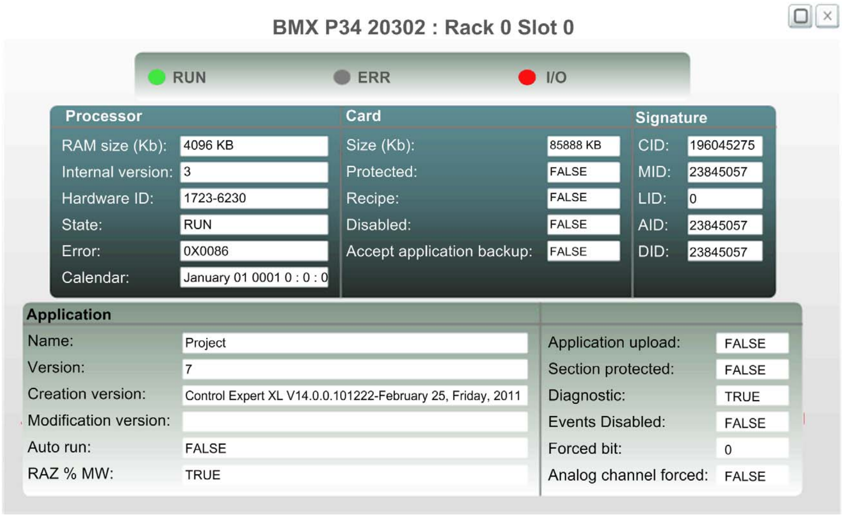
Controller Status

The controller status page is a read-only Silverlight page. Click on the controller in the **Rack Viewer** to display the following information:

- the module name and its rack and slot location
- the state of the controller’s **RUN**, **ERR** and **I/O** LEDs
- information describing the processor and network card
- information describing the application running in the controller

NOTE: For a description of a specific controller’s parameters, consult the user documentation for that module.

The controller status page looks like this



BMX P34 20302 : Rack 0 Slot 0

● RUN ● ERR ● I/O

Processor
RAM size (Kb): 4096 KB
Internal version: 3
Hardware ID: 1723-6230
State: RUN
Error: 0X0086
Calendar: January 01 0001 0 : 0 : 0

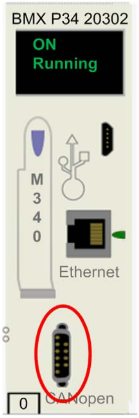
Card
Size (Kb): 85888 KB
Protected: FALSE
Recipe: FALSE
Disabled: FALSE
Accept application backup: FALSE

Signature
CID: 196045275
MID: 23845057
LID: 0
AID: 23845057
DID: 23845057

Application
Name: Project
Version: 7
Creation version: Control Expert XL V14.0.0.101222-February 25, Friday, 2011
Modification version:
Auto run: FALSE
RAZ % MW: TRUE
Application upload: FALSE
Section protected: FALSE
Diagnostic: TRUE
Events Disabled: FALSE
Forced bit: 0
Analog channel forced: FALSE

CANopen Network Status

To open the CANopen network status page, click on the controller's CANopen port (circled in red, below) in the **Rack Viewer** display:

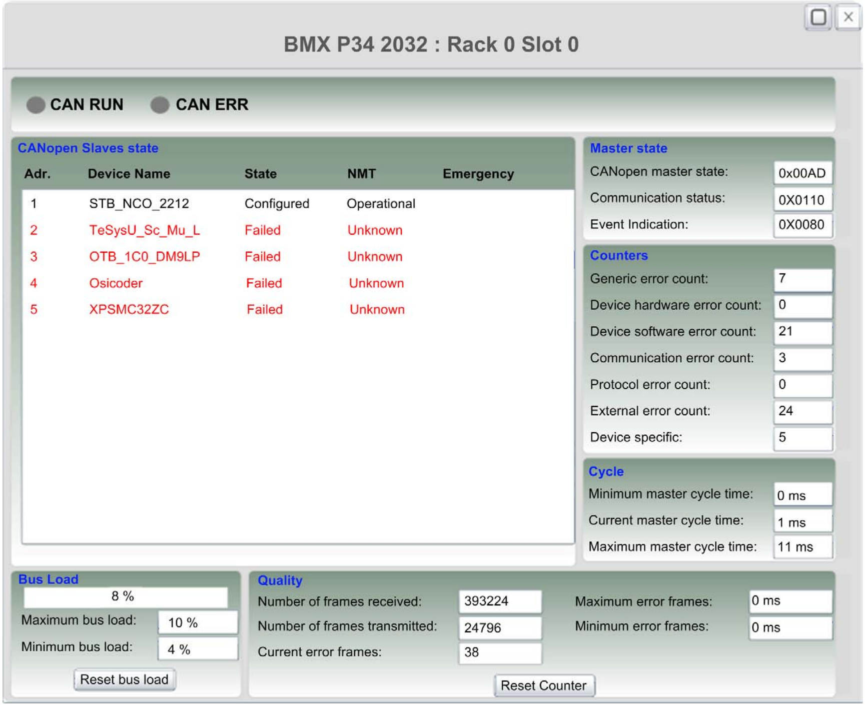


The CANopen network status page is a read-only Silverlight page that displays the following information:

- the module name and its rack and slot location
- the state of the controller's **CAN RUN** and **CAN ERR** LEDs
- information describing the CANopen master state
- a list of CANopen slave devices, containing a description of each device and its state
- CANopen network operating status, including counters, cycle, load and quality statistics

NOTE: For a description of the CANopen network parameters, consult the user documentation for the specific controller module.

The CANopen network status page looks like this:



BMX P34 2032 : Rack 0 Slot 0

● CAN RUN ● CAN ERR

CANopen Slaves state

Adr.	Device Name	State	NMT	Emergency
1	STB_NCO_2212	Configured	Operational	
2	TeSysU_Sc_Mu_L	Failed	Unknown	
3	OTB_1C0_DM9LP	Failed	Unknown	
4	Osicoder	Failed	Unknown	
5	XPSMC32ZC	Failed	Unknown	

Master state

CANopen master state:	0x00AD
Communication status:	0X0110
Event Indication:	0X0080

Counters

Generic error count:	7
Device hardware error count:	0
Device software error count:	21
Communication error count:	3
Protocol error count:	0
External error count:	24
Device specific:	5

Cycle

Minimum master cycle time:	0 ms
Current master cycle time:	1 ms
Maximum master cycle time:	11 ms

Bus Load

	8 %
Maximum bus load:	10 %
Minimum bus load:	4 %

Quality

Number of frames received:	393224	Maximum error frames:	0 ms
Number of frames transmitted:	24796	Minimum error frames:	0 ms
Current error frames:	38		

CANopen devices that are physically located on the network appear in the **CANopen Slaves status** list in black text; CANopen devices that are configured in Control Expert but not detected on the CANopen network appear in red text.

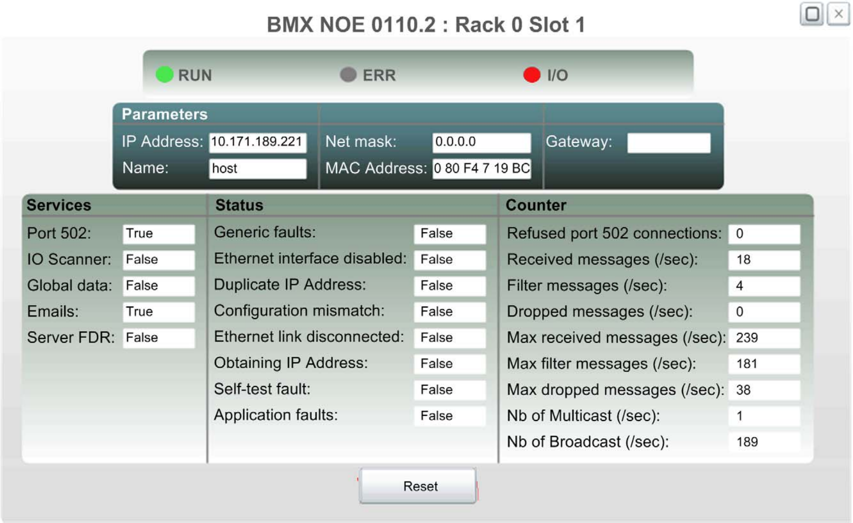
Communication Module Status

The Ethernet communication module status page is a read-only Silverlight page that displays the following module information:

- the module name and its rack and slot location
- its **RUN**, **ERR** and **I/O** LEDs
- IP address settings
- a list of services, indicating if each service is enabled (true) or disabled (false)
- module operational status
- Ethernet connection and message counting statistics

NOTE: For a description of the Ethernet communication module parameters, consult the user documentation for the specific module.

The Ethernet communication module status page looks like this:



BMX NOE 0110.2 : Rack 0 Slot 1

● RUN ● ERR ● I/O

Parameters

IP Address:	10.171.189.221	Net mask:	0.0.0.0	Gateway:	
Name:	host	MAC Address:	0 80 F4 7 19 BC		

Services		Status		Counter	
Port 502:	True	Generic faults:	False	Refused port 502 connections:	0
IO Scanner:	False	Ethernet interface disabled:	False	Received messages (/sec):	18
Global data:	False	Duplicate IP Address:	False	Filter messages (/sec):	4
Emails:	True	Configuration mismatch:	False	Dropped messages (/sec):	0
Server FDR:	False	Ethernet link disconnected:	False	Max received messages (/sec):	239
		Obtaining IP Address:	False	Max filter messages (/sec):	181
		Self-test fault:	False	Max dropped messages (/sec):	38
		Application faults:	False	Nb of Multicast (/sec):	1
				Nb of Broadcast (/sec):	189

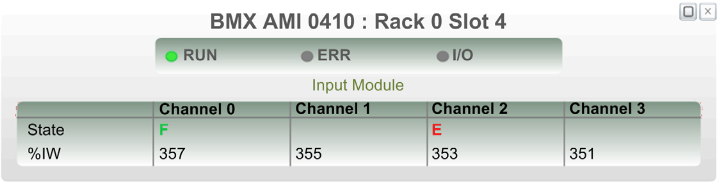
Analog Input/ Output Status

When you click on an analog input or analog output module in a local or remote rack, a Silverlight page opens displaying information about that module, including

- the analog input or output module type
- rack and slot location
- a list of words assigned to the module
- the status of the module's **RUN**, **ERR** and **I/O** LEDs
- the address assignment of each of analog channel
- the state of each analog channel, where:
  - E indicates detected error
  - F indicates forced
  - FB indicates the configured fall back position

NOTE: For a description of the I/O module parameters, consult the user documentation for the specific I/O module.

An analog input module status page looks like this:



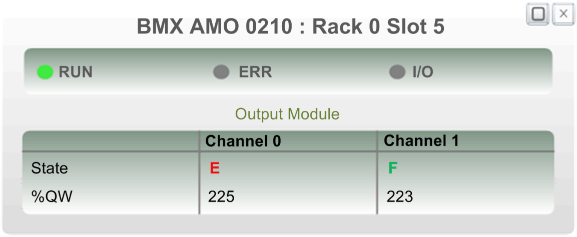
BMX AMI 0410 : Rack 0 Slot 4

● RUN ● ERR ● I/O

Input Module

	Channel 0	Channel 1	Channel 2	Channel 3
State	F		E	
%IW	357	355	353	351

An analog output module status page looks like this:



BMX AMO 0210 : Rack 0 Slot 5

● RUN ● ERR ● I/O

Output Module

	Channel 0	Channel 1
State	E	F
%QW	225	223

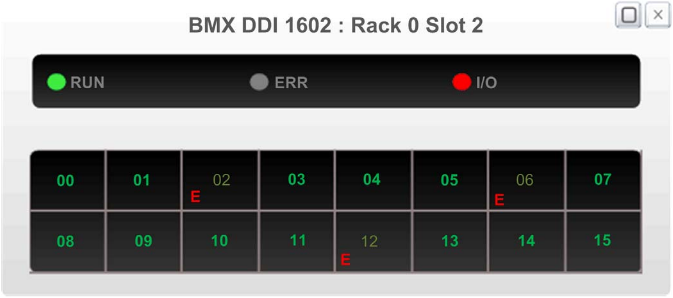
Digital Input / Output Status

When you click on a digital input or digital output module in a local or remote rack, a Silverlight page opens displaying information about that module, including

- the digital input or output module type
- rack and slot location
- the status of the module's **RUN**, **ERR** and **I/O** LEDs
- a presentation of each of the module's input or output points, where:
  - bright green = active
  - dull green = inactive
- the state of each input or output point, where:
  - for discrete outputs:
    - – "E" indicates a detected error
    - – "F" indicates forced
    - – "FB" indicates the configured fall back state
  - for discrete inputs:
    - – "E" indicates a detected error

NOTE: For a description of the I/O module parameters, consult the user documentation for the specific I/O module.

A digital input module status page looks like this:

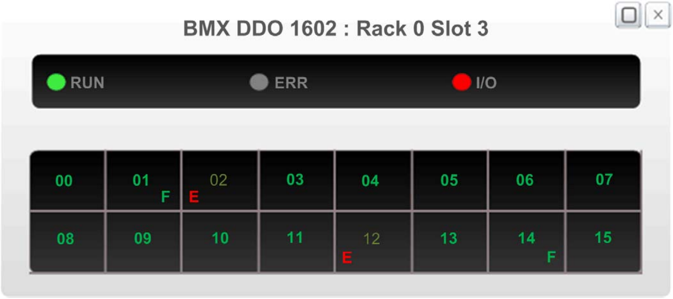


BMX DDI 1602 : Rack 0 Slot 2

● RUN ● ERR ● I/O

00	01	02 E	03	04	05	06 E	07
08	09	10	11	12 E	13	14	15

A digital output module status page looks like this:



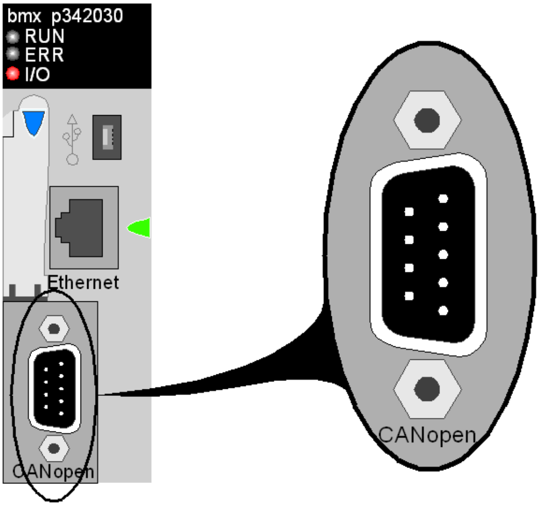
CANopen Status (Java)

Introduction

You can use the Java version of the **Rack Viewer** to explore the CANopen devices connected to the bus.

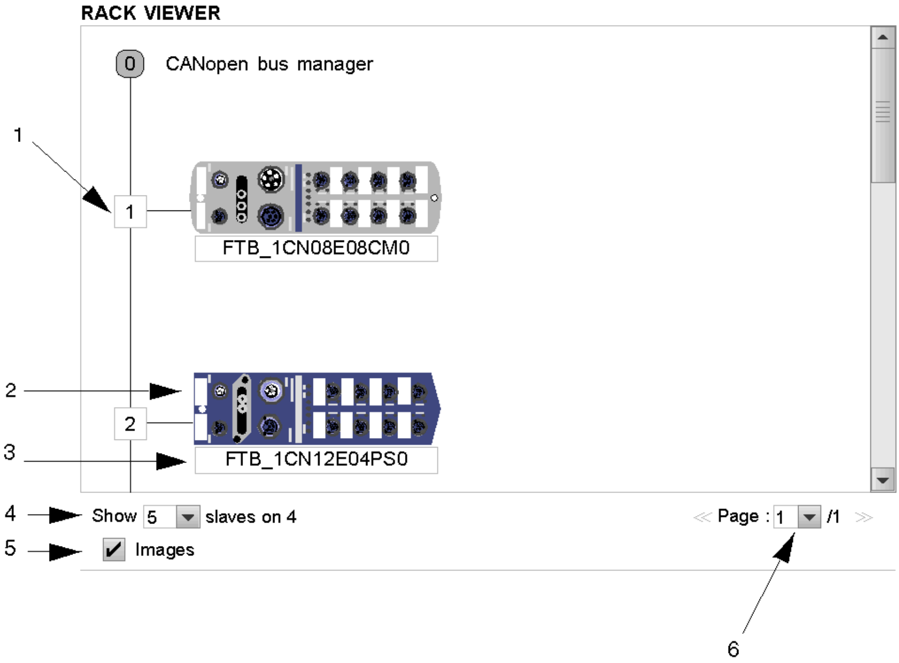
Accessing the CANopen Status page

The procedure below shows how to access the CANopen status page.

Step	Action
1	Click the **Diagnostics** link on the home page.
2	Click the **Rack Viewer** link on the diagnostics page.
3	Click the CANopen port on the front side of the CPU:  bmx p342030 RUN ERR I/O Ethernet CANopen CANopen
4	The **Rack Viewer** displays the CANopen network status page.
5	Click on the **CANopen bus explorer** button, to open the **CANopen bus editor**.

CANopen Bus Editor

The CANopen bus editor looks like this:



Elements and Functions

This table describes the different areas that make up the configuration screen:

Number	Element	Function
1	Logical address area	This area includes the addresses of the devices connected to the bus.
2	Module area	This area includes the devices that are configured to the bus.
3	Device name	Name of the device connected to the bus.
Display Properties		
4	Elements by page	Set the number of devices displayed on a page. You can display 5, 10, 25, 50 or 100 devices.
5	Images	Select this box to display bitmaps of the devices connected to the bus.
6	Page navigator	These buttons enable you to navigate through the pages displaying the bus devices.

NOTE:

- When the Rack viewer identifies the CANopen device through its catalog containing the device bitmap, the image of the device is displayed. Otherwise, it displays a generic Standard CANopen bitmap for accessible but unknown CANopen devices.
- The devices that the CANopen master cannot reach are displayed without reference in red.

Ethernet Module Diagnostic Pages

Overview

The **NOE Diagnostics** menu contains a list of links for accessing the different diagnostic pages for the Ethernet module:

- Global Data utility
- I/O scanning utility
- Messaging utility
- Ethernet module statistics
- Bandwidth monitoring utility
- Network Time Protocol time synchronization service parameters
- Upload MIB file utility

Click on a link to access the desired diagnostics page.

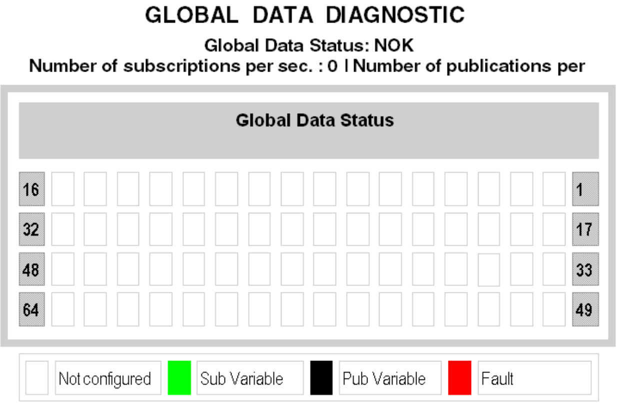
Global Data Page

Information on the general diagnostics of Global Data can be found at the top of this page:

- Status
- Number of publications per second
- Number of subscriptions per second

This page also shows a table of all published and subscribed variables in the same distribution group. Each variable is identified by its Identifier:

- Green for the subscribed variables
- Black for the published variables
- White for unconfigured variables
- Red for variables with communication interruptions



GLOBAL DATA DIAGNOSTIC

Global Data Status: NOK

Number of subscriptions per sec. : 0 I Number of publications per

Global Data Status

16																	1
32																	17
48																	33
64																	49

□ Not configured 🟩 Sub Variable ■ Pub Variable 🟥 Fault

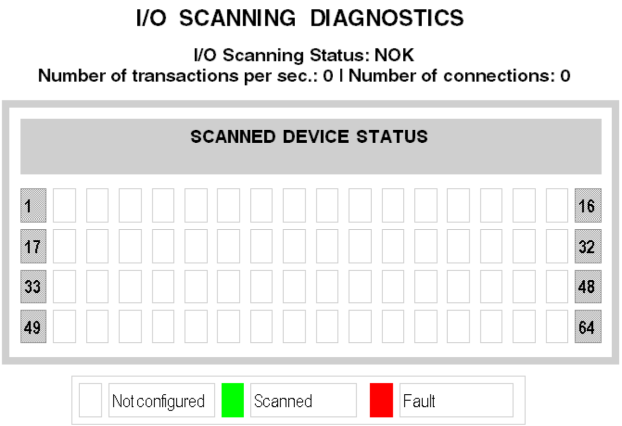
I/O Scanning Page

Information on the general diagnostics of the I/O scanning utility can be found at the top of this page:

- Status
- Number of transactions per second
- Number of connections

This page also displays a summary of the status of all modules:

- Green for the **scanned** modules
- White for the **unconfigured** modules
- Red for improperly operating modules



I/O SCANNING DIAGNOSTICS

I/O Scanning Status: NOK

Number of transactions per sec.: 0 I Number of connections: 0

SCANNED DEVICE STATUS

1															16
17															32
33															48
49															64

□ Not configured 🟩 Scanned 🟥 Fault

NOTE:
The data status (OK/NOK) indicates only the state of your local configured network:

- OK: The table reports the state of scanned (remote) devices
- NOK: Your local system is "not OK," so data in the table is inconsequential.

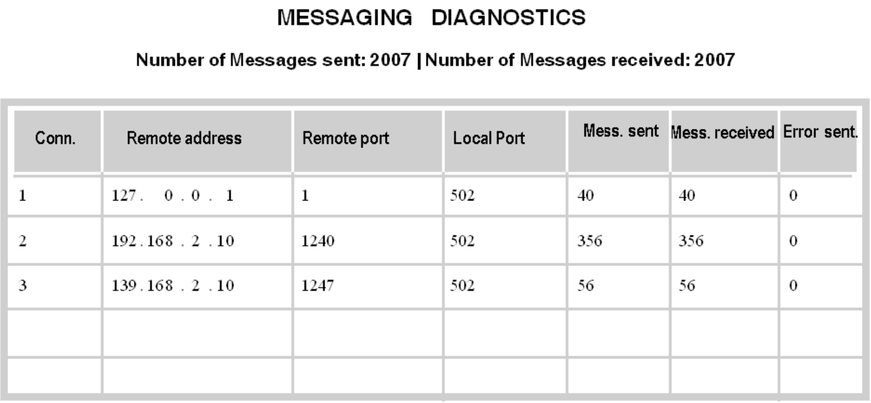
Messaging Page

This page provides current information on the open TCP connections on port 502.

The number of sent/received messages on the port can be found at the top of this page.

A table provides, for each connection (numbered from 1 to 64):

- The remote IP Address
- The remote TCP port
- The local TCP port
- The number of messages sent from this connection
- The number of messages received from this connection
- The detected error number on this connection



MESSAGING DIAGNOSTICS

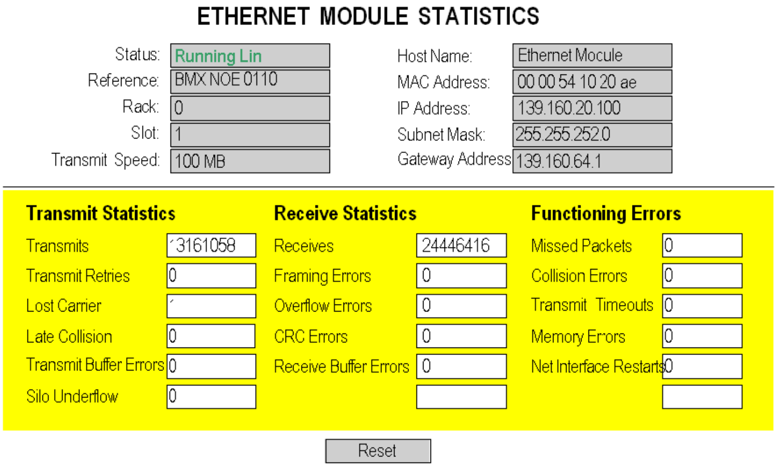
Number of Messages sent: 2007 | Number of Messages received: 2007

Conn.	Remote address	Remote port	Local Port	Mess. sent	Mess. received	Error sent.
1	127 . 0 . 0 . 1	1	502	40	40	0
2	192 . 168 . 2 . 10	1240	502	356	356	0
3	139 . 168 . 2 . 10	1247	502	56	56	0

NOTE: Following a request to close a connection, the PLC may hold the connection open in its memory for a few minutes, during which the table will reflect the open connection.
Number of Messages received is not reset after a port 502 connection is closed. Therefore, the count indicates the total number of messages that have been received since the module was started.
The remote address ‘127.0.0.1’ is used as Private System Connection For Diagnostic Feature or SOAP Communications.

Ethernet Statistics Page

The Ethernet Module Statistics page provides information about the status, transmit and receive statistics, and detected errors for the Embedded Server module. Access this page by selecting the NOE module form the local rack or use the hyperlink Statistics. The following graphic is an example **Ethernet Module Statistics** page:



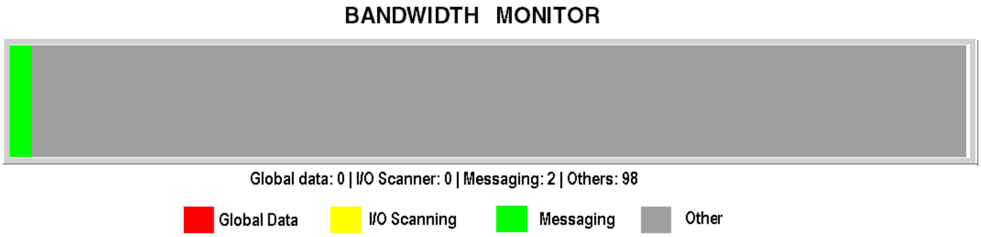
ETHERNET MODULE STATISTICS

Status:	Running Lin	Host Name:	Ethernet Mocule
Reference:	BMX NOE 0110	MAC Address:	00 00 54 10 20 ae
Rack:	0	IP Address:	139.160.20.100
Slot:	1	Subnet Mask:	255.255.252.0
Transmit Speed:	100 MB	Gateway Address	139.160.64.1

Transmit Statistics		Receive Statistics		Functioning Errors	
Transmits	`3161058	Receives	24446416	Missed Packets	0
Transmit Retries	0	Framing Errors	0	Collision Errors	0
Lost Carrier	`	Overflow Errors	0	Transmit Timeouts	0
Late Collision	0	CRC Errors	0	Memory Errors	0
Transmit Buffer Errors	0	Receive Buffer Errors	0	Net Interface Restarts	0
Silo Underflow	0				

Bandwidth Monitoring Page

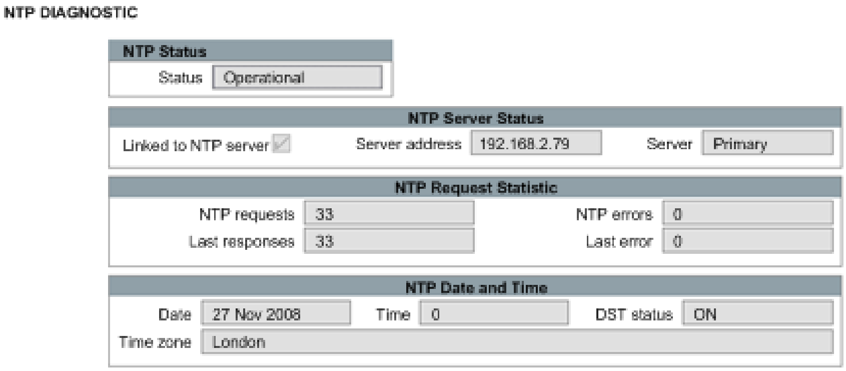
This page displays the load distribution of the Embedded Server module between the Global Data utilities, I/O Scanning, Messaging, and other utilities:



BANDWIDTH MONITOR

Global data: 0 | I/O Scanner: 0 | Messaging: 2 | Others: 98

NTP Diagnostics Page



NTP DIAGNOSTIC

NTP Status

Status: Operational

NTP Server Status

Linked to NTP server ☑ Server address: 192.168.2.79 Server: Primary

NTP Request Statistic

NTP requests: 33 NTP errors: 0

Last responses: 33 Last error: 0

NTP Date and Time

Date: 27 Nov 2008 Time: 0 DST status: ON

Time zone: London

Time synchronization service parameters:

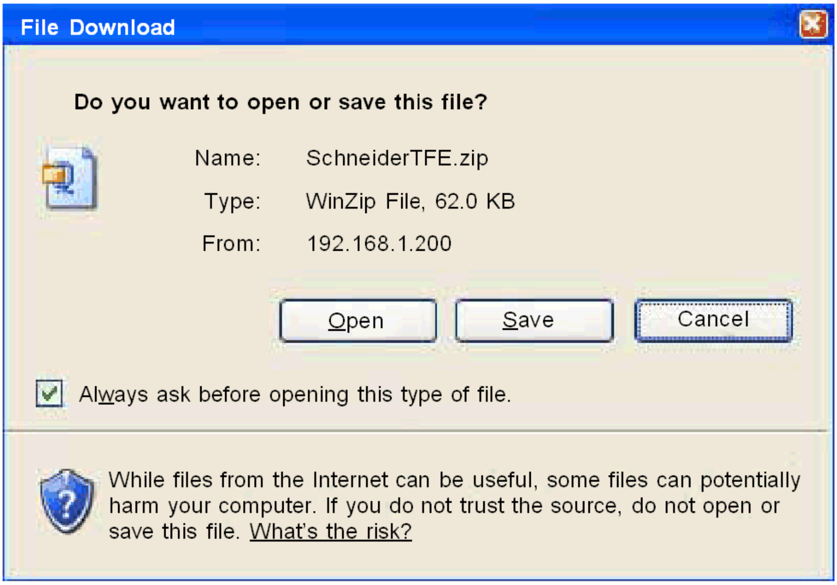
Parameter	Description
NTP status	Service is correctly configured (OK)
NTP server status	NTP client is connected to the NTP server, and if the server is Primary or Standby
NTP requests	Total number of client requests sent to the NTP server
NTP responses	Total number of server responses sent from the NTP server
Number of errors	Total number of unanswered NTP requests
Last error code	Last detected error code received from the NTP client
Date	Date in d/m/y format
Time	Time
Time zone	Time zone plus or minus Universal Time, Coordinated (UTC)
DST	Daylight saving time (DST) parameter is either 1. on (enabled) 2. off (disabled)

Last Error field displays values, which indicate the type of detected event.

Type of detected event	Value
Component OK and running	0
Excessive network traffic or server overload	1
Bad parameters in the configuration	3
Component is disabled	4
Incorrect IP	9
Time zone file absent	14
Incorrect syntax in the *customrules* file	15

Upload MIB file

When you select **Upload MIB File**, the **File Download** dialog box appears. You are asked if you want to save the MIB file or open it:



Do you want to open or save this file?

Name: SchneiderTFE.zip

Type: WinZip File, 62.0 KB

From: 192.168.1.200

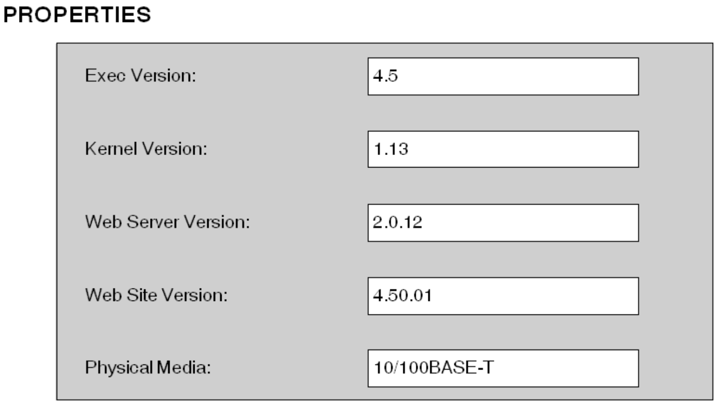
☑ Always ask before opening this type of file.

While files from the Internet can be useful, some files can potentially harm your computer. If you do not trust the source, do not open or save this file. What's the risk?

NOE Properties Page

Introduction to the NOE Properties Page

You can navigate to the **NOE Properties Page** from the Diagnostics page *(see page 35)*. The **NOE Properties Page** displays the versions of the Exec, Kernel, and Web Pages and the Physical Media:



PROPERTIES

Exec Version:	4.5
Kernel Version:	1.13
Web Server Version:	2.0.12
Web Site Version:	4.50.01
Physical Media:	10/100BASE-T

NOTE: This page only reports this information. The fields can not be changed.

Section 3.4
Setup Page

Overview

This section describes the different links of the **Setup** page.

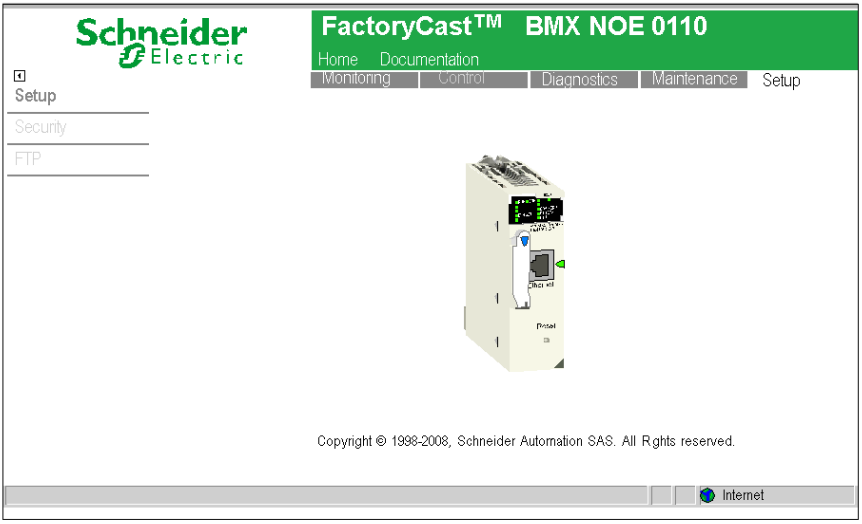
What Is in This Section?

This section contains the following topics:

Setup Page

Setup Page

Use this page to access configuration settings for the module:



Copyright © 1998-2008, Schneider Automation SAS. All Rghts reserved.

Click on these links to view a description of the configurable settings:

- Security *(see page 58)*
- FTP *(see page 61)*

Changing the HTTP and Write Passwords

Introduction

This page enables you to change:

- The user name and password for accessing the home page
- The password for writing variables to the data editor

NOTE: The maximum length of the user name and password is 16 characters (a-z, A-Z and 0-9).

NOTE:

- Access to this page is password protected. The default values for the username/password fields are **USER/USER**.
- The default username and password for FTP are **USER/USERUSER**.

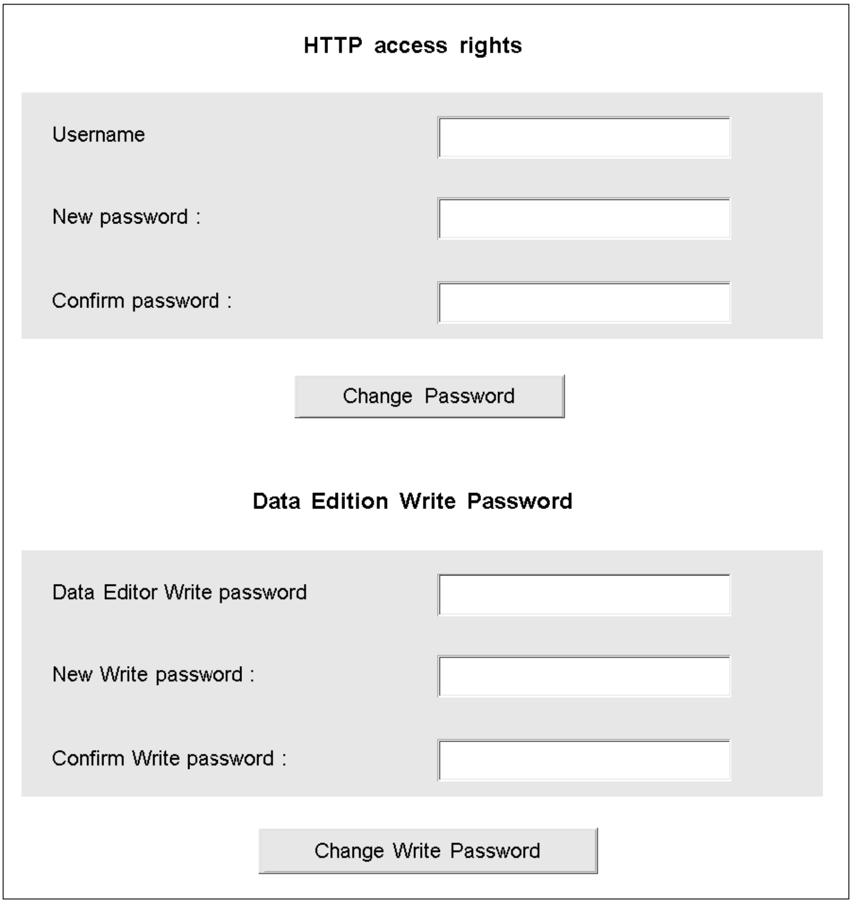
Accessing the Security Configuration Page

The following procedure shows how to access the security configuration page.

Step	Action
1	Click the **Setup** link on the **Home** page.
2	Click the **Security** link on the **Setup** page.

Viewing the Security Page

The **Security** page looks like this:



HTTP access rights

Username

New password :

Confirm password :

Change Password

Data Edition Write Password

Data Editor Write password

New Write password :

Confirm Write password :

Change Write Password

Modifying HTTP Access Rights

To modify HTTP access rights, perform the following steps:

Step	Action
1	Enter the current username.
2	Enter the new password.
3	Confirm the new password.
4	Confirm the change by clicking **Change Password**. **Result**: A confirmation window appears.

Changing the Data Editor Write Password

To change the data editor write password, perform the following steps:

Step	Action
1	Enter the current data editor write password (field is case-sensitive).
2	Enter the new write password.
3	Confirm the new write password.
4	Confirm the change by clicking **Change Write Password**. **Result**: A confirmation window appears.

Changing the FTP Password

Presentation

This page enables you to change the user name and password of the FTP service.

NOTE: The maximum length of the user name and password is 40 characters (a-z, A-Z and 0-9). The password length has to be at least 8 characters.

NOTE: This page is password protected. The default values for the Username/Password fields are **USER/USER**.

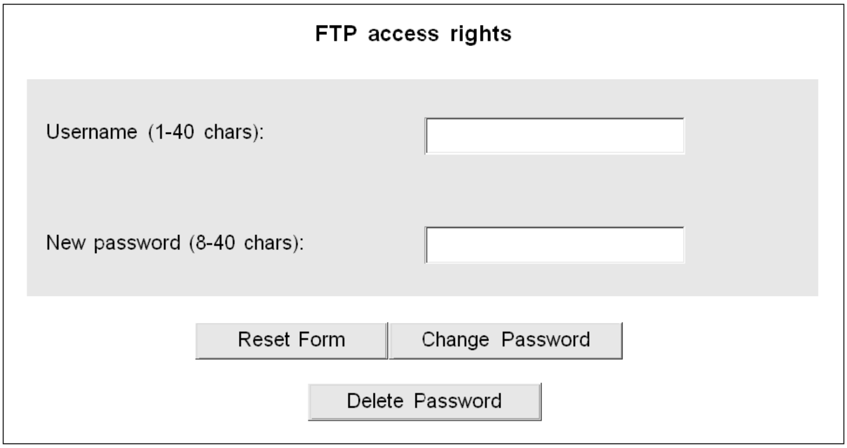
Accessing the FTP Configuration Page

The procedure below shows how to access the security configuration page.

Step	Action
1	Click the **Setup** link on the Home page
2	Click the **FTP** link on the setup page

View of the FTP Page

The FTP page looks like this:



FTP access rights

Username (1-40 chars):

New password (8-40 chars):

Modifying FTP Access Rights

The procedure is as follows:

Step	Action
1	Enter the current username.
2	Enter the new password.
3	Confirm the change by clicking **Change Password**. **Result**: a confirmation window appears.

Section 3.5
Alarm Viewer

Overview

The Alarm Viewer is a Java applet designed to monitor Modicon M340 PLCs. It is a Client Runtime Function and will run in the JVM of internet client browser (Internet Explorer, Netscape Navigator, etc.).

What Is in This Section?

This section contains the following topics:

Display

Access

Select **Diagnostics | Alarm Viewer**.

NOTE: Alarm Viewer is a plug-in. You must download it before you can use it.

The display is composed of

- Button bar (in the work area)
- List of alarms
- Status frame

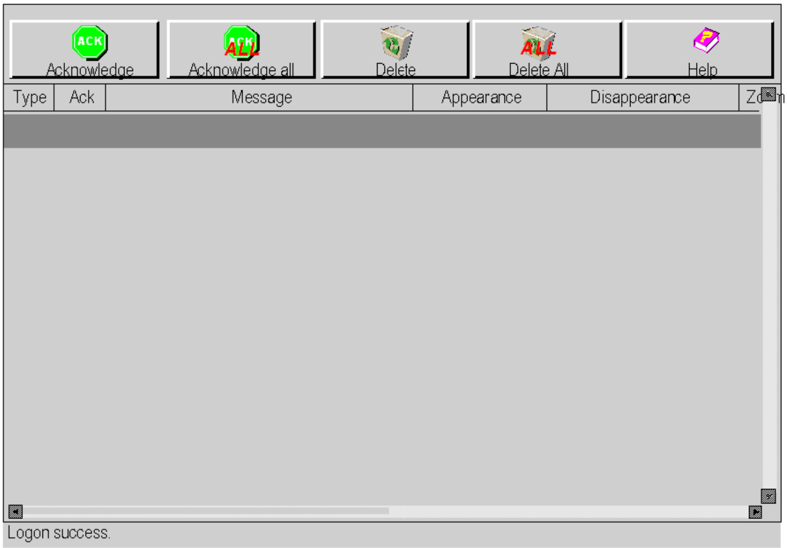
Button Bar

The button bar contains five buttons.

ACK	This button is used to acknowledge a selected alarm in the list. A request is sent to the PLC (Diagnostic Buffer).
ACK ALL	This button is used to acknowledge all alarms in the list that can be acknowledged. A request for each alarm is sent to the PLC.
♻	This button is used to delete a selected alarm in the list. There is no request sent to the PLC. This command affects only the alarm list in Alarm Viewer.
♻ ALL	This button is used to delete all alarms in the list that can be deleted. There is no request sent to the PLC. This command affects only the alarm list in Alarm Viewer.
?	This button displays a frame that contains help.

List of Alarms

Alarms in the list are displayed in historical order. The last alarm in runtime is added to the end of the list.



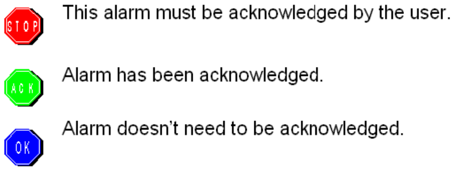
Logon success.

Each line displayed in the list corresponds to an alarm and contains the following information

Type: An icon that represents the alarm type. For each type of alarm there is a different icon.

DFB	DFB alarm
	Grafcet alarm
	System alarm
ASI	A-SI alarm

Ack: An icon that represents the acknowledgement status.



STOP This alarm must be acknowledged by the user.

ACK Alarm has been acknowledged.

OK Alarm doesn’t need to be acknowledged.

Message: Alarm text.

Appearance: Date and time when alarm occurs.

Disappearance: Date and time when alarm disappears.

Zone or area number: Area or geographical zone from which the alarm comes (common area: 0).

NOTE: Consider the following.

- You can change the column width by using the mouse.
- The number of alarms that can be displayed in the list is limited to 1000. When this limit is reached, an information message is displayed in the status frame. Alarm Viewer recognizes subsequent alarms, but doesn't display them. To see the next alarms, purge the list of alarms displayed by Alarm Viewer.
- Alarm Viewer displays alarms from all zones. The zone contains values from 0...15.

NOTE: An alarm that appears is displayed in red. An alarm that disappears is displayed in green.

Status Frame

This frame is used to display the alarm, an information message, or throughput messages. For example a message such as, "Diagnostic Application is not configured in this application."

Operation and Management of Alarms

Browsing

Use the UP, DOWN, PAGEUP, or PAGEDOWN keys or the mouse to select the alarms in the list. Use the scroll bar if the list contains more alarms than can be displayed in the dialog.

Acknowledgement

To acknowledge an alarm that requires acknowledgment, select the alarm and use the appropriate toolbar button.

Several alarms can be acknowledged at one time by using the **Ack All** button.

NOTE: An alarm can be acknowledged by another Alarm Viewer. In this case, Alarm Viewer is notified and the alarm is displayed as acknowledged.

Deleting an Alarm from the List

- An alarm that requires acknowledgment or that has not disappeared cannot be deleted.
- DELETE and DELETE ALL buttons can be used to delete only those alarms that have disappeared and have been acknowledged (if acknowledgment is required).

Limitations

Overview

For each alarm, there is additional information stocked in the diagnostic buffer. For example, many DFBs have outputs named STATUS (word) where the cause of the alarm is coded. Alarm Viewer doesn’t use this information. It displays only basic information about alarms.

Alarm Viewer Functionality

The Alarm Viewer works only under these conditions.

- Product: BMX NOE 0110
- Firmware platform: Modicon M340
- Software platform: Unity Pro 3.1 or later

  NOTE: Unity Pro is the former name of Control Expert for version 13.1 or earlier.

Chapter 4
Editors

Overview

This chapter describes the Data Editor and the Graphic Editor Java applets that enable you to create either dynamic data tables or dynamic graphic displays. Both editors are dynamically updated with run-time data from the PLC.

What Is in This Chapter?

This chapter contains the following sections:

Section 4.1
Data Editor

Subject of this Section

The Data Editor allows you to edit/create data monitoring tables or to display Data tables. Data tables provide read/write access to application data and devices registers. Write access is password protected.

Allowing write access can change system behavior.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Password-protect the embedded server.
- Carefully select the symbols and direct addresses you authorize to be modified online.
- Do not authorize online modifications of critical process variables.
- Do not use default or obvious password.
- Limit access to trained personnel.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

This section shows how to use the Data Editor to display and modify the values of the symbol variables and direct addresses.

What Is in This Section?

This section contains the following topics:

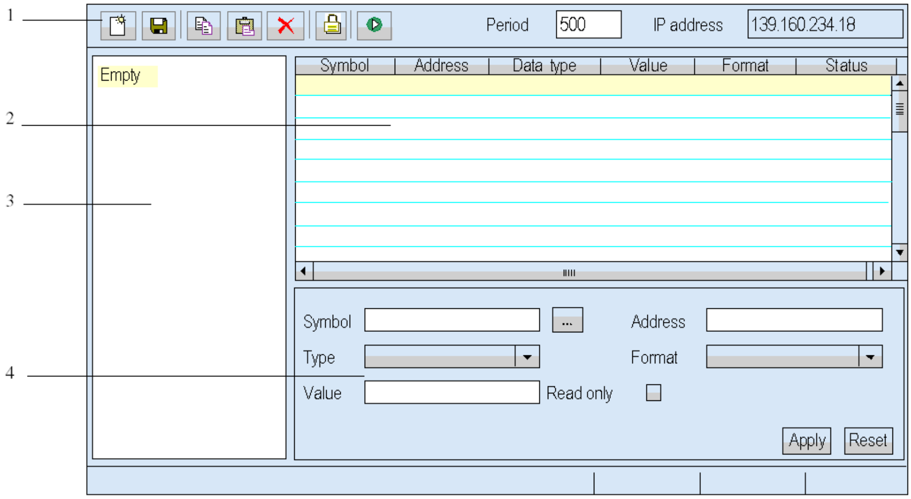
Data Editor

Overview

The Data Editor is a Java applet that enables you to create dynamic data tables that can be updated with run-time data from the PLC.

Elements of the Data Editor

The Data Editor comprises 4 screen elements:



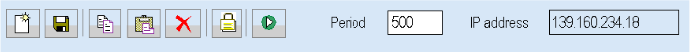
Period 500

IP address 139.160.234.18

Number	Description
1	Tool bar *(see page 72)*.
2	Data Editor Template *(see page 73)*. The data editor template is a spreadsheet that contains your data.
3	List of data templates.
4	The Configuration Area makes it possible to: ● select *(see page 79)* and/or modify *(see page 83)* a symbol, ● modify *(see page 83)* a symbol value, ● select *(see page 81)* and/or modify an address, ● select the variable type, ● select the variable's display format, ● check the read-only option.

Tool bar

Here is a closer view of the buttons on the Data Editor‘s tool bar:



From left to right, the buttons indicate:

- create a new object
- save the current application
- copy the object
- paste the object
- delete the object
- change the password
- start or stop the animation
- set the rate for polling the variable
- display the server’s IP address in format <`host`>[:<:`filename`>]

Data Editor Spreadsheet

Overview

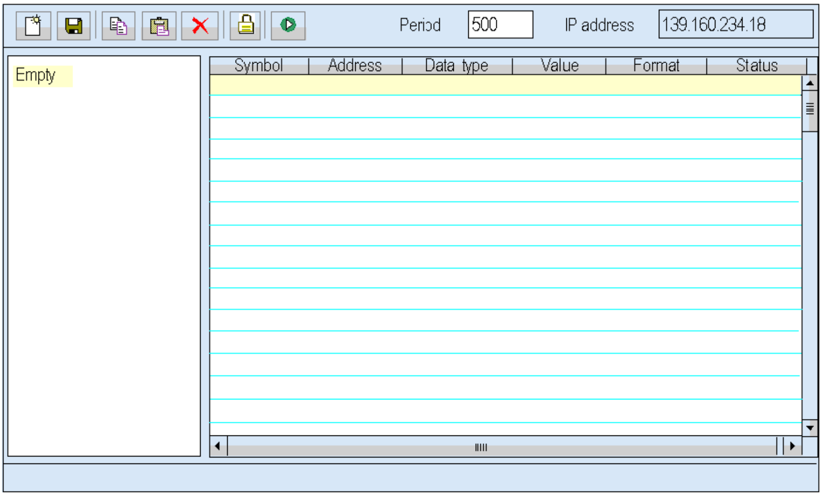
The Data Editor spreadsheet displays data with the following fields:

- Symbol
- Address
- Data Type
- Value
- Format
- Status

This topic provides a snapshot of the spreadsheet and an explanation of each field.

Spreadsheet

The figure shows the Data Editor spreadsheet:



Period: 500

IP address: 139.160.234.18

Empty

Symbol	Address	Data type	Value	Format	Status

Symbol Field

The **Symbol** column contains the names of Control Expert symbols (variables).

The only symbols (variables) that may be used in the Data Editor are those in the namespace on the Embedded Server.

NOTE: The Data Editor can only read values from a namespace that was created from the same program as the one running in the controller. The program used in the controller is displayed at the top of the Data Editor. If the namespace was created using a different program, its name is displayed at the bottom of the Data Editor.

Address Field

The **Address** column contains direct addresses and the addresses of Control Expert symbols (variables). Any direct address can be viewed by entering its reference in this field. It does not have to be included in the namespace.

Valid Direct Addresses for Modicon M340

The following table contains the valid direct addresses for Modicon M340:

Address	Type	R(ead) or W(rite) Access
%KWi	WORD 16	R
%KDi	WORD 32	R
%MDi	WORD 32	R/W
%SDi	WORD 32	R/W
%Ir.s.c	BOOLEAN	R
%Qr.s.c	BOOLEAN	R/W
%Mi	BOOLEAN	R/W
%Si	BOOLEAN	R/W
%MFi	REAL 32	R/W
%IWr.s.c.	WORD 16	R
%MWi	WORD 16	R/W
%SWi	WORD 16	R/W
%QWr.s.c.	WORD 16	R/W
%MWr.s.c.i	WORD 16	R/W
%KWr.s.c.i	WORD 16	R/W
r = rack number, s = slot number, c = channel number, and i = range number		

The following table contains the valid Fipio addresses for Modicon M340:

Address	Type	R(ead) or W(rite) Access
%I\bs.cp\0.m.c	BOOLEAN	R
%Q\bs.cp\0.m.c	BOOLEAN	R/W
%IW\bs.cp\0.m.c.i	WORD 16	R
%QW\bs.cp\0.m.c.i	WORD 16	R/W
%MW\bs.cp\0.m.c.i	WORD 16	R/W
%KW\bs.cp\0.m.c.i	WORD 16	R
bs = bus number, cp = connection point, m = module number, c = channel number and i = range number		

For Modicon M340, a single bit of any "word address" (for example, %MWi, %SWi, %KWi) can be specified by appending ".j" to the address, where "j" is a bit index in the range of 0 (least significant bit) to 15 (most significant bit). For example, bit 4 of the value at %MW101 would be specified as %MW101.4.

Also for Modicon M340 PLC, a direct address can include an index specification that allows it to be treated as an array variable. Indexed addressing can be used with a %Mi, %MWi, %MDi, %MFi, %KWi or %KD address by appending "[j]" to the address of the beginning of the array, where "[j]"is an unsigned integer value. For example, the third value of an array of float values starting at %MF201 would be specified as %MF201[2].

Data Type Field

The **Data Type** field contains the data type of the symbol (variable) or direct address. Symbol (variable) data types appear automatically when the symbol (variable) is located. Set the direct address data types from a drop-down list.

The following data types are valid:

Abbreviation	Data Type
INT	16-bit signed integer
UINT	16-bit unsigned integer
DINT	32-bit signed integer
UDINT	32-bit unsigned integer
REAL	32-bit IEEE floating point
TIME	32-bit unsigned integer (in ms)
DATE	Date (32-bit BCD)
TOD	Time of Day (32-bit BCD)
BOOL	1-bit discreet (boolean)

Value Field

The **Value** column is filled with the value of the symbol (variable) or direct address. This field is updated continuously.

Format Field

The **Format** field contains the format type for displaying the value of the symbol (variable) or direct address. The following formats are available:

Abbreviation	Format Type
bool	Boolean
dec	Decimal
hex	Hexadecimal
binary	Binary
ASCII	bytes displayed as ASCII characters
time	day_hr_min_sec_ms
date	YYYY-MM-DD or HH:MM:SS

Status Field

The **Status** column contains messages about the status of communications with the symbol (variable) or direct address. If communications are normal, the status message will be "OK."

If there is a communication interruption with the symbol (variable) or direct address, the **Status** column will contain a message describing the event.

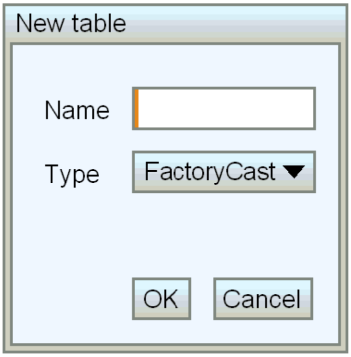
Creating a Data Template

Overview

To display some symbols (variables) or direct addresses, create a new data template.

Creating a Data Template

Follow the steps in the table below to create a data template:

Step	Description
1	If you would like a new one, select the 🗋 button on the toolbar. Result: The **New table** box appears.  New table Name Type FactoryCast
2	Enter the name of the new Data template.
3	Click **OK**.

NOTE: Save *(see page 82)* the current spreadsheet before selecting a new spreadsheet. Selecting a new spreadsheet deletes the current spreadsheet.

Using an Existing Data Template

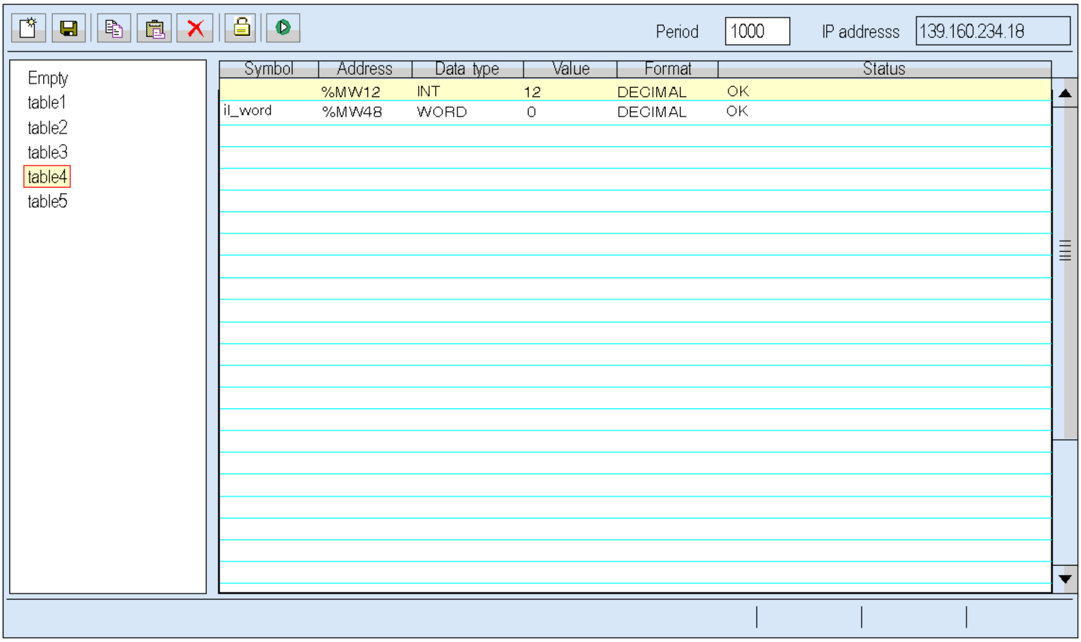
Overview

After you save a data template, you can use it to view or modify the values of the same symbols (variables) and direct addresses.

Finding a Data Template

A treeview menu on the Data Editor lists the saved data templates.

DATA EDITOR



Period 1000

IP addresss 139.160.234.18

Empty
table1
table2
table3
table4
table5

Symbol	Address	Data type	Value	Format	Status
	%MW12	INT	12	DECIMAL	OK
il_word	%MW48	WORD	0	DECIMAL	OK

Retrieving a Data Template

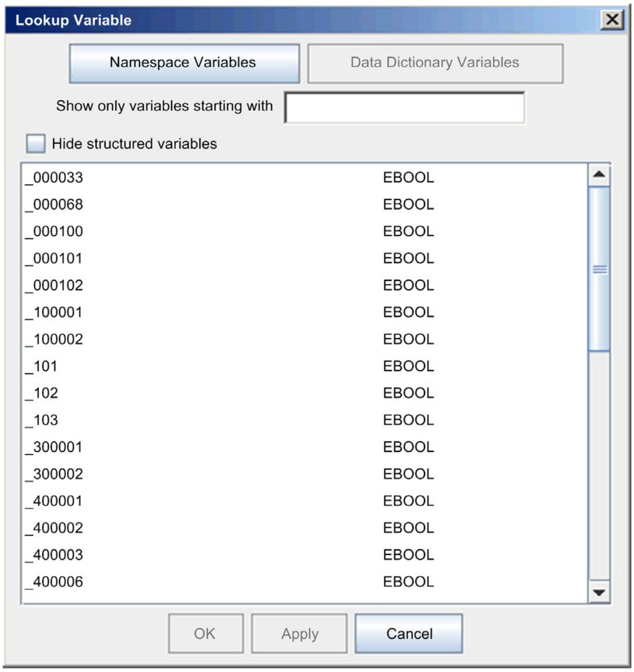
Select the data template you want from the treeview menu. It will appear on a spreadsheet.

Inserting a Symbol (Variable) in a Data Template

Presentation

To view or modify the value of a symbol (variable) in the namespace, insert that symbol (variable) in a data template.

Inserting a Symbol (Variable)

Step	Action
1	Double-click on an empty row in the spreadsheet. **Result**: The data editor's configuration area appears.
2	In the configuration area, click on the ... button. **Result:** The **Lookup Variable** dialog box appears. 

Lookup Variable

Namespace Variables | Data Dictionary Variables

Show only variables starting with

□ Hide structured variables

_000033	EBOOL
_000068	EBOOL
_000100	EBOOL
_000101	EBOOL
_000102	EBOOL
_100001	EBOOL
_100002	EBOOL
_101	EBOOL
_102	EBOOL
_103	EBOOL
_300001	EBOOL
_300002	EBOOL
_400001	EBOOL
_400002	EBOOL
_400003	EBOOL
_400006	EBOOL

OK | Apply | Cancel

Step	Action
3	Click one of the following buttons: ● **Namespace Variables**: to display variables assigned to the Namespace of the web server that resides in the Ethernet communication module. ● **Data Dictionary Variables**: to display variables from the PLC data dictionary. **NOTE:** Before you can select variables from the data dictionary, configure Control Expert so that the data dictionary is embedded in PLC memory.
4	Select the symbol (variable) you want to insert in the data template.
5	Click **OK**. **Result:** The symbol (variable) you selected is displayed in the **Symbol** field.
6	In the configuration area, click on **Apply**. **Result:** A new row corresponding to the symbol (variable) you selected is displayed in the spreadsheet.

Inserting a Direct Address in a Data Template

Presentation

If you want to view or modify the value of a direct address, insert that direct address in a data template.

Allowing write access can change system behavior.

⚠ **WARNING**

UNINTENDED EQUIPMENT OPERATION

- Limit embedded server access to qualified personnel.
- Restrict access to the embedded server by configuring passwords.
- Carefully select the symbols and direct addresses you authorize to be modified online.
- Do not authorize online modification of variables that can adversely affect human and material integrity.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Inserting a Direct Address

Step	Action
1	Double-click an empty row in the spreadsheet. **Result**: The data editor's configuration area appears.
2	In the **Address** field of the configuration area, enter the variable's direct address.
3	In the configuration area, click **Apply**. **Result:** A new row that corresponds to the variable address is displayed in the spreadsheet.

Saving a Data Template

Overview

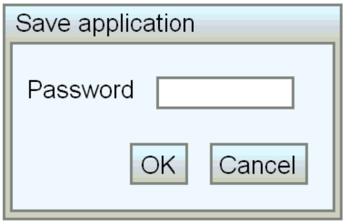
If you save a data template, you can use it again to view or modify the same symbols (variables) or direct addresses.

NOTE:

- When you modify and save a data table, the last modifications is saved and overwrites the existing table even if someone else has created it.
- If someone is viewing a data table which is overwritten, the modifications are seen only on the next access to the data editor.

Saving a Data Template

Follow the steps in the table below to save a data template.

Step	Action
1	Click the 💾 button on the tool bar. Result: The **Save application** box appears. 
2	Enter the password (USER by default).
3	Click **OK**.

Modifying Data Values in a Data Template

Overview

You can use the Data Editor to modify the value of a symbol (variable) and of a direct address, and send the new value(s) to the controller.

Allowing write access can change system behavior.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION
● Password-protect the embedded server. ● Carefully select the symbols and direct addresses you authorize to be modified online. ● Do not authorize online modifications of critical process variables. ● Do not use default or obvious password. ● Limit access to trained personnel.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

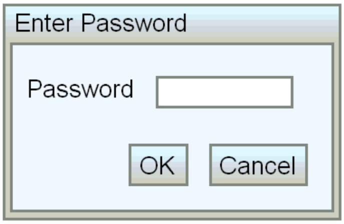
For example, say you have programmed a pushbutton object to jog a motor when the button is depressed and to stop jogging when the button is released. If communications are lost while the button is depressed, the motor will continue to jog even when the button is released. Graphic objects should not be used to control situations such as this unless other interlock methods are installed in the system.

Restrictions on Modifying Data

You can only modify the value of a symbol (variable) or of a direct address that is write-enabled in the namespace.

Modifying Data

Follow the steps in the table below to modify data:

Step	Action
1	Double-click on a spreadsheet row that corresponds to the symbol (variable). **Result**: The data editor’s configuration area appears.
2	In the **Value** field of the configuration area, change the value of the symbol (variable) or direct address.
3	Click **Apply**.
4	**Result**: The **Password** dialog appears. 
5	Enter the password (USER by default).
6	Click **OK**. **Result**: The new value is sent to the controller.

Variables via Tooltip box

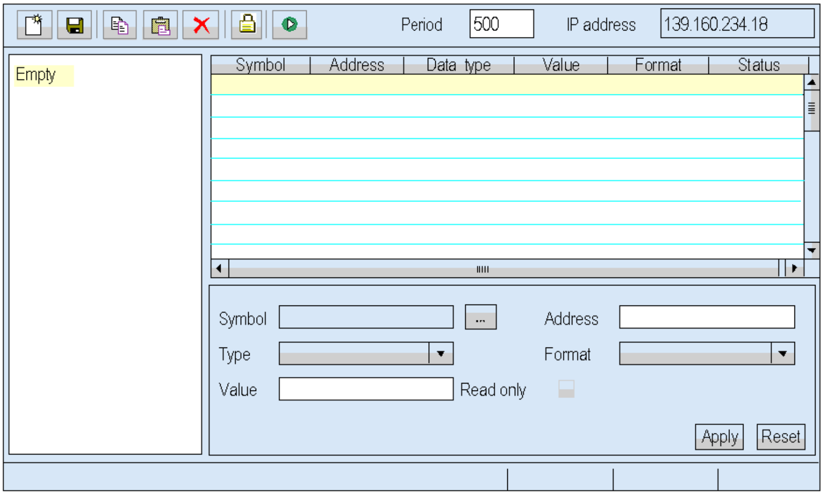
When the mouse is placed on the row containing the variable UNITY, the texts entered in the UNITY are displayed on a tooltip.

Section 4.2
Data Editor Lite

Data Editor Lite

Overview

Data Editor Lite is a compact version of Data Editor that is faster to download, especially for use over a dialup connection. It provides the same interface than the standard Data Editor but with some restrictions on accessible data type.



Period 500

IP address 139.160.234.18

Variables

Data Editor Lite accepts the following IEC variables:

Address	Type	Display
%MW IEC internal word	INT	DECIMAL
%MD IEC double word	DINT	DECIMAL
%M IEC internal bits	BOOL	BOOLEAN

Tables

Data Editor Lite can reuse the tables created with Data Editor and Pocket Data Editor. However, Data Editor tables use a wider range of variable types than in Data Editor Lite. When Data Editor Lite encounters a variable it cannot manage, it displays `Not Supported`. In this case, the variable cannot be edited.

Operating Mode

Refer to the Operating Mode section for Data Editor *(see page 70)*.

NOTE: You cannot insert symbols (variables) in a Data Template with Data Editor Lite, you can only insert direct addresses.

Section 4.3
Pocket Data Editor

Overview

This section describes the Pocket Data Editor, which runs on Pocket PC.

What Is in This Section?

This section contains the following topics:

Pocket Data Editor

Overview

Pocket Data Editor is a version of the Data Editor that operates in a Pocket PC environment. A Pocket PC with a WiFi connection can be used as a mobile client station to access data on remote equipment. Pocket Data Editor supports remote data monitoring, remote control, and remote diagnostics.

⚠ WARNING
LOSS OF PROCESS ALERTS AND CONTROL
● Do not use the Pocket Editor to retrieve and display or control events that could affect human, material or equipment integrity. ● Provide an appropriate and independent protection via your own application or process.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Pocket PC Requirements

The Pocket PC requirements are:

- only Pocket PCs with ARM processors
- Pocket PC Windows 2003 mobile edition

Pocket PC Windows 2002 mobile edition is not supported.

Capabilities

Pocket Data Editor can open and edit the same data tables as the PC version in a format appropriate for Pocket PCs.

NOTE:
- You cannot save the tables edited by the Pocket Data Editor.
- If you have developed custom pages for your Web server, Internet Explorer for Pocket PC can access the same custom HTML pages on an equipment Web site. A Pocket PC screen is smaller than one on a PC, so adapt the Web site to return the appropriate page based on the client's browser type.

Availability

NOTE:

- Pocket Data Editor is disabled when the Pocket PC is connected to and synchronized with a PC via USB using the ActivSync tool.
- Some communication interruptions can occur between a Pocket PC and the FactoryCast module when the WiFi (802.11) power save mode is activated on a Pocket PC. In such a case, it is not possible to connect to the module. See your Pocket PC user manual for the procedure to disable this feature.

Variables

Pocket Data Editor accepts only symbols (variables).

Acceptable types of symbols are:

- INT,
- DINT,
- BOOL,
- EBOOL.

NOTE:

- Direct Modbus addresses are not supported. The status bar displays **not supported** if you enter a direct Modbus address in a Pocket Data table.
- Variables that you want to monitor with the Pocket PC have to be declared to Persistent mode in Web Designer for Modicon M340.

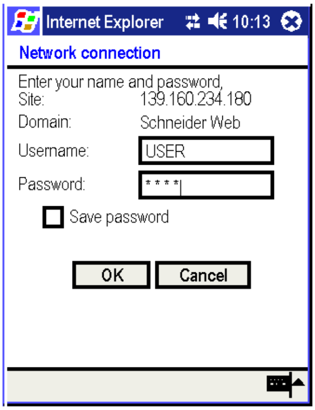
Using Pocket Data Editor

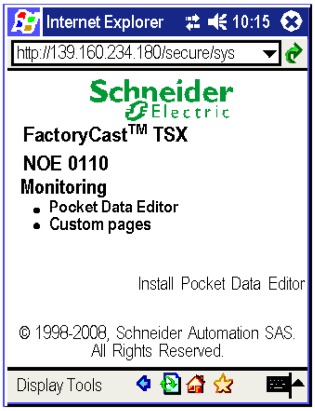
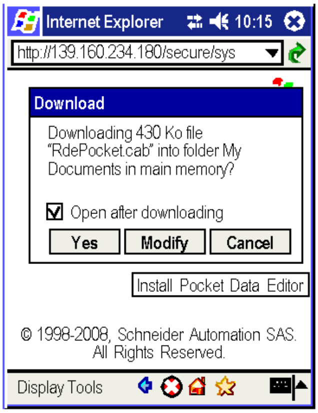
Overview

When you first connect a Pocket PC to the FactoryCast Web server, download and install the Pocket Data Editor software on your Pocket PC. After this, you can launch the Pocket Data Editor by clicking on the **Pocket Data Editor** link on the equipment Web site home page.

Installing Pocket Data Editor

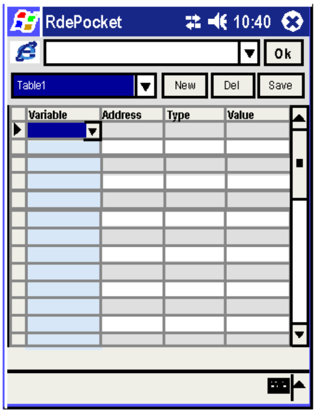
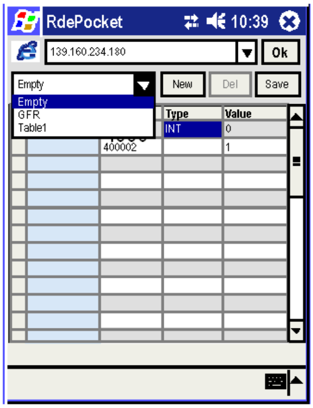
The following table describes the installation of the Pocket Data Editor:

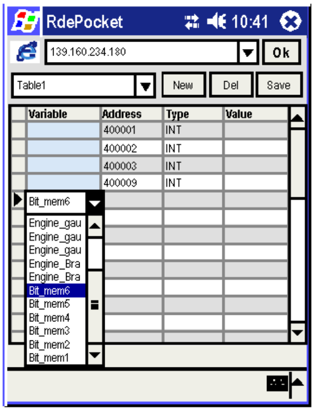
Step	Action
1	Turn on the Pocket PC.
2	Launch Internet Explorer.
3	Enter the IP address of the remote equipment. **Result**: The following window appears:  Network connection Enter your name and password. Site: 139.160.234.180 Domain: Schneider Web Username: USER Password: **** □ Save password

Step	Action
4	Enter the username and password. (The defaults are USER and USER.) **Result**: the FactoryCast Server home page appears:  Internet Explorer 10:15 http://139.160.234.180/secure/sys Schneider Electric FactoryCast™ TSX NOE 0110 Monitoring - Pocket Data Editor - Custom pages Install Pocket Data Editor © 1998-2008, Schneider Automation SAS. All Rights Reserved.
5	Click on **Install Pocket Data Editor**. **Result**: The following window appears:  Internet Explorer 10:15 http://139.160.234.180/secure/sys Download Downloading 430 Ko file "RdePocket.cab" into folder My Documents in main memory? ☑ Open after downloading Install Pocket Data Editor © 1998-2008, Schneider Automation SAS. All Rights Reserved.
6	Click **YES**.

Using Pocket Data Editor

The following table describes the use of Pocket Data Editor

Step	Action
1	Click on the **Pocket Data Editor** link on the equipment Web site home page. **Result**: The following window appears:  RdePocket 10:40 Ok Table1 New Del Save Variable Address Type Value Push the **New** button to create a new table after entering its name. Push the **Del** button to delete the current table. Push the **Save** button to save the new version of the table after entering the HTTP password.
2	Select the desired table via the drop-down menu:  RdePocket 10:39 139.160.234.180 Ok Empty New Del Save Empty GFR Table1 Type Value INT 0 400002 1 The Pocket PC version can use the same tables defined in the Data Editor for PC and in Data Editor Lite. Not all variables are accessible. (Restrictions are detailed at the Variables discussion *(see page 89)*.) A selected variable may be reported as **not supported** in the status bar. For an unsupported variable type, the string `???` appears in the value cell.

Step	Action
3	If you want to view or modify the value of a PLC variable, insert the variable symbol or address in the Pocket Data table.  RdePocket 10:41 139.160.234.180 Ok Table1 New Del Save Variable Address Type Value 400001 INT 400002 INT 400003 INT 400009 INT Bit_mem6 Engine_gau Engine_gau Engine_gau Engine_Bra Engine_Bra Bit_mem6 Bit_mem5 Bit_mem4 Bit_mem3 Bit_mem2 Bit_mem1 You can add new addresses, following the restrictions detailed in the Variables section *(see page 89)*. To modify a PLC variable, select the corresponding **value** cell and enter the new value.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

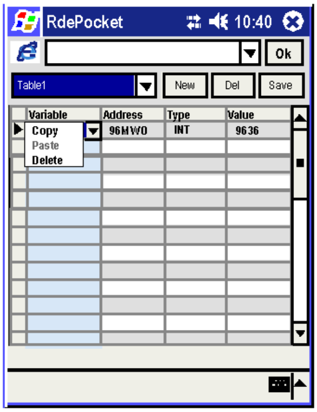
- Password-protect the embedded server.
- Carefully select the symbols and direct addresses you authorize to be modified online.
- Do not authorize online modifications of critical process variables.
- Do not use default or obvious password.
- Limit access to trained personnel.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Copy, Paste, or Delete Table Rows

To copy, paste, or delete, select one or more rows by clicking in the first cell (to the left of the **variable** cell). The selected rows are highlighted. Hold down the mouse button in this cell until a popup menu appears, then select the desired action.

- To copy, select one or more rows that have values (that are not empty).
- To paste the copied rows, select a row, empty or not. If several rows are copied, the other rows are pasted one after the other.
- To delete one or more rows, select rows that have values (that are not empty) and choose **delete**.



Variable	Address	Type	Value
	96MW0	INT	9636

Using customized pages

If you want to display a custom page, click on the **Custom pages** link on the equipment Web site home page.

Launch Pocket Data Editor from its icon

Pocket Data Editor can also be launched directly from its program icon.

Section 4.4
Graphic Editor

Overview

This section describes the functions and features of the Graphic Editor, a Java applet that enables you to create dynamic graphic displays via a Web browser, using a set of pre-defined graphic objects. The Graphic Editor is only used as an editor for creating and modifying displays. The Graphic Viewer is the run-time environment for viewing displays while they are dynamically animated with run-time data from the PLC. This viewer is lighter than the editor, resulting in faster loading times and operation.

What Is in This Section?

This section contains the following topics:

Graphic Editor Overview

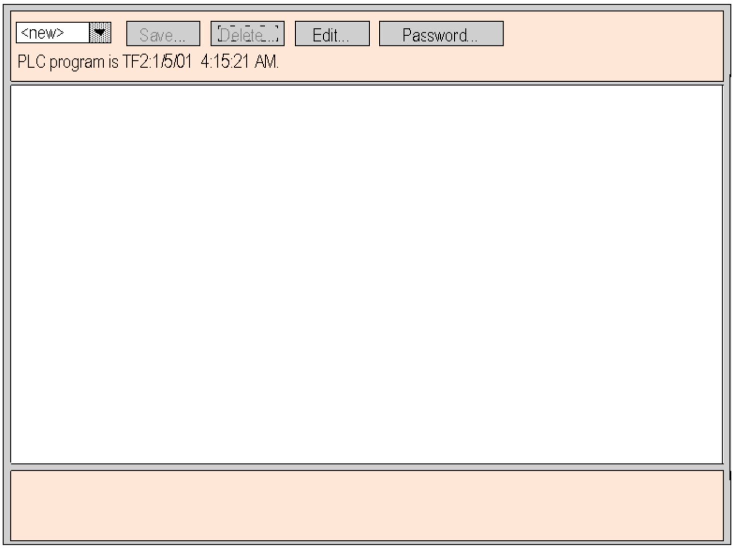
The Interface

The Graphic Editor applet is separated into three windows:

- Top Window: This window provides the area for presenting the user controls and dialogs for creating, saving, reading, and editing a graphic display.
- Display Window: This window provides an area for presenting the current graphic display. When you create a new graphic display, this window becomes an "empty canvas" on which you can add the graphic objects that will compose the desired graphic display.
- Message Window: This window presents any messages generated by the Graphic Editor.

Illustration

The figure below shows the Graphic Editor applet with its initial top window, and empty display and message windows.



PLC program is TF2:1/5/01 4:15:21 AM.

Graphic Objects

The graphic objects provided with the Graphic Editor are capable of communicating with the PLC from which the Graphic Editor applet was downloaded. There is no additional "wiring" of the graphic objects with "communication objects." Graphic objects are standalone, meaning there are no connections required between them and each is capable of operating independently.

Viewing a Graphic Display

After the Graphic Editor applet has been uploaded to a Web browser, you will usually want to either view a graphic display (for monitoring/controlling the PLC application) or create or modify a graphic display. A user who only wants to view and interact with existing graphic displays (e.g., an operator) can select the Graphic Viewer link instead of Graphic Editor. They will see a window with the widgets that does not include the Edit menu. This viewer loads faster than the standard Graphic Editor because it is lighter. You need only to enter a password to write data to the PLC.

Allowing write access can change system behavior.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Password-protect the embedded server.
- Carefully select the symbols and direct addresses you authorize to be modified online.
- Do not authorize online modifications of critical process variables.
- Do not use default or obvious password.
- Limit access to trained personnel.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Create and Modify Graphic Displays

To create and modify graphic displays, click the **Edit...** button to see the standard graphic editor functions. These include selecting objects from a palette, dropping them onto a canvas, moving and resizing them with a mouse, and setting object properties. You can immediately test the modified graphic display with run-time data from the PLC by clicking the **Done** button to exit edit mode. When satisfied with your creation, the graphic display can be saved to the PLC for re-use by clicking the **Save...** button, assuming you entered the correct password.

NOTE: Be careful when you modify and save a graphic page, the last modifications will be saved and will overwrite the existing page even if someone else has created it.

User Functions

Most of the Graphic Editor’s user functions are available as Top Window User Functions *(see page 98)*. From the display window, you can directly manipulate a graphic object’s size and location. All properties of a graphic object (e.g., scaling values, labels, colors, PLC addresses of the run-time data) are set in the Property Sheet *(see page 103)*.

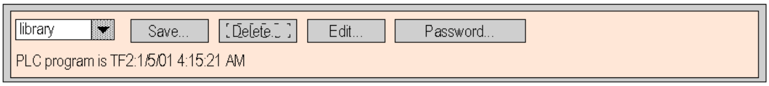
Top Window User Functions

Overview

The Graphic Editor applet's top window consists of several "dialog panels," only one of which is visible at any time. Switching from one dialog to another is done by clicking buttons on the current dialog. This topic describes the dialog panels that compose the top window.

Top Dialog

The **Top Dialog** is the dialog panel that is initially shown in the top window when the Graphic Editor applet is started. Access to all other dialog panels of the top window is from this dialog.



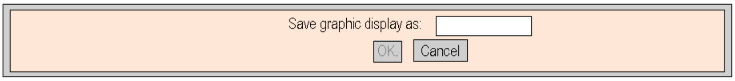
PLC program is TF2:1/5/01 4:15:21 AM

The controls of the **Top Dialog** provide the following functions:

- **Drop-down List**. The drop-down list box shows all graphic display files that have been saved to the Web server module and are available for retrieval. When you select a graphic display from this list, the graphic display currently in the window is replaced with the selected one. If the current graphic display has been modified since it was last saved, you are asked for confirmation that the changes are to be discarded. If the special entry **<new>** is chosen from the list, the display window is cleared and a new graphic display can be created.
- **Save**. The **Save** button makes the **Save Dialog** visible. This button is disabled until you have entered a correct write-enabled password.
- **Delete**. The **Delete...** button makes the **Delete Dialog** visible. This button is disabled until you have entered a correct password, or if the current graphic display has not yet been saved.
- **Edit**. The **Edit...** button makes the **Edit Dialog** visible.
- **Password**. The **Password...** button makes the **Password Dialog** visible.
- **Information display area**. The information display area shows the name and version of the Control Expert program that is running in the connected PLC.

Save Dialog

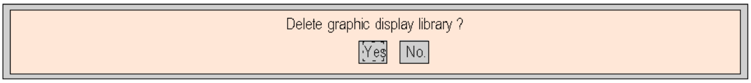
The **Save Dialog** allows you to save the current graphic display.



When the **Save Dialog** is presented, the name of the current graphic display is shown in the dialog's text field. If the current graphic display has not been saved (i.e., a "new" graphic display), then the text field is blank. Once you have either accepted the current name (with a "save" operation) or provided a new name (with a "save as" operation), then you can click the **OK** button to save the contents of the current graphic display to the Web server module. The **Cancel** button will cause the **Top Dialog** to be shown again, with no action being taken.

Delete Dialog

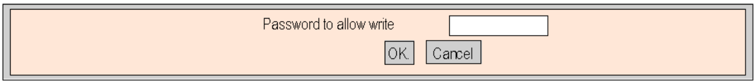
The **Delete Dialog** allows you to delete the current graphic display.



If you click **Yes**, the existing graphic display window is cleared and the graphics file on the Web server module is deleted. Clicking **No** will cause the **Top Dialog** to be shown again, with no action being taken.

Password Dialog

The **Password Dialog** allows you to enter the password that enables those user functions that modify graphic display files or PLC run-time data values.

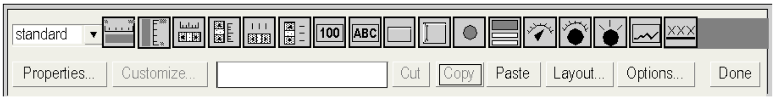


If you enter the correct password and click **OK**, then you will be allowed to save and delete the current graphic display. Correct entry of the password also permits you to write new values to the PLC (via those graphic objects that support writing values to a PLC, if any). Clicking **OK** when the text field is empty clears the current password permissions (if there are any). The **Cancel** button redisplays the **Top Dialog** without changing the current password permissions.

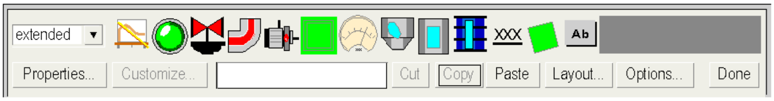
Edit Dialog

The **Edit Dialog** allows you to select a graphic object for placement in the display window, and provides access to all graphic editing functions. The available graphic objects are presented in a set of palettes, with one palette visible at a time. There are two palettes.

The standard palette:



The extended palette:

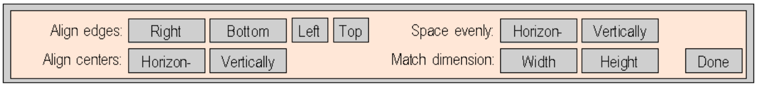


The controls of the **Edit Dialog** provide the following functions:

- The **Drop-down List Box** shows the set of available palettes. When you select the name of a palette from the list, the graphic objects in that palette appear in the palette display area of the dialog.
- The **Palette** shows the graphic objects in the current palette. An icon depicts each graphic object's type (meter, button, etc.). When you click any icon in the palette, a graphic object of the corresponding type is selected for insertion. If you click in an open area of the display window while the Graphic Editor is in "insert mode," an instance of the selected graphic object is inserted into the graphic display.
- The **Information Area** shows the name and size of the graphic object that is currently selected.
- The **Cut** button causes the currently selected graphic object(s) to be removed from the graphic display and saved to a buffer (i.e., an internal clipboard), replacing the existing contents of the buffer.
- The **Copy** button causes the currently selected graphic object(s) to be copied to the buffer, replacing the existing contents of the buffer.
- The **Paste** button causes the content of the clipboard to be inserted into the upper left corner of the graphic display. The pasted graphic objects can then be moved to the desired location in the display.
- The **Properties** button displays the currently selected graphic object’s Property Sheet *(see page 103)*.
- The **Customize** button displays the currently selected object’s Customizer (if the graphic object has one).
- The **Layout** button makes the **Layout Dialog** visible.
- The **Options** button makes the **Options Dialog** visible.
- The **Done** button makes the **Top Dialog** visible again.

Layout Dialog

The **Layout Dialog** allows you to change the position and size of a group of graphic objects.



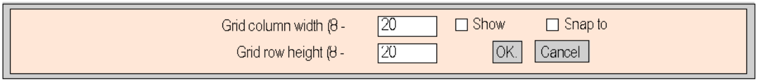
The controls of the **Layout Dialog** provide the following functions:

- For aligning the edges of graphic objects, the **Right**, **Bottom**, **Left**, and **Top** buttons move the selected graphic objects so that their specified sides are at the same position. Select at least two graphic objects to enable these buttons.
- For aligning the center lines of graphic objects, the **Horizontally** and **Vertically** buttons move the selected graphic objects so that their horizontal or vertical center lines, respectively, are at the same position. Select at least two graphic objects to enable these buttons.
- For positioning graphic objects so that they are evenly spaced, the **Horizontally** and **Vertically** buttons move the selected graphic objects so that either the horizontal or vertical spacing between the objects is the same. Select at least three graphic objects to enable these buttons.
- To automatically size graphic objects, use the **Width** and **Height** buttons to re-size the currently selected graphic objects so that either the widths or heights, respectively, of the objects match. Select at least two graphic objects to enable these buttons.
- The **Done** button makes the **Edit Dialog** visible again.

NOTE: For most layout operations (except **Space evenly**) one of the selected objects is considered the "reference object" to which all other selected objects adjust for their new position or dimension. For example, when the "Width" button is pressed, all selected objects have their widths changed to match that of the reference object. The reference object is differentiated from the other selected objects by making its selection box a different color than the others.

Options Dialog

The **Options Dialog** allows you to change the settings related to a grid that can be drawn in the display window. The grid is solely for assistance in editing a graphic display and is shown only when the Graphic Editor is in "edit mode." Edit mode starts when you switch to the **Edit Dialog** and ends when you return to the **Top Dialog**.



Grid column width (8 - 20 ☐ Show ☐ Snap to

Grid row height (8 - 20

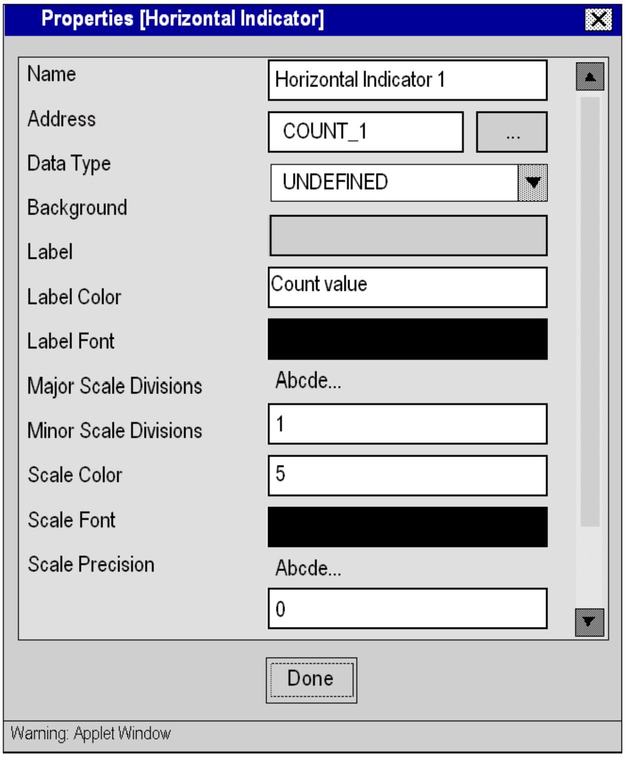
The **Options Dialog** controls provide the following functions:

- The cell size of the grid can be changed by the entering the grid's column width and row height in the dialog's text fields.
- If the **Show grid** check-box is checked, the grid is drawn; otherwise, no grid is shown.
- If the **Snap to grid** check-box is checked, then, when you change the size or position of a graphic object, the changed coordinate(s) or dimension(s) is automatically adjusted to coincide with a grid point.
- The **OK** button causes the current option settings to become active, and the **Edit Dialog** to be shown again.
- The **Cancel** button causes the **Edit Dialog** to be shown again, with no option settings being changed.

Property Sheet

Overview

The Property Sheet is a "floating" (non-modal) dialog that shows the configurable properties of the currently selected graphic object:



Properties [Horizontal Indicator]

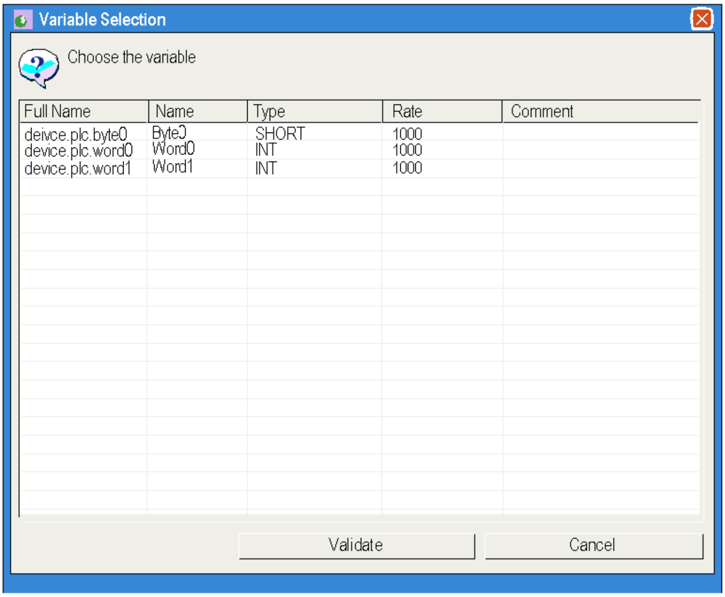
Name	Horizontal Indicator 1
Address	COUNT_1
Data Type	UNDEFINED
Background	▨
Label	Count value
Label Color	■
Label Font	Abcde...
Major Scale Divisions	1
Minor Scale Divisions	5
Scale Color	■
Scale Font	Abcde...
Scale Precision	0

The properties of a graphic object are specific to an object's type. The properties are presented in a scrollable list, with the name and the value of each property listed. The Graphic Editor comes with a description of graphic objects *(see page 107)*.

Lookup Dialog

For each of the graphic objects provided with the Graphic Editor, a property editor is provided for its **Address** property. This editor not only allows you to directly enter the address of a Control Expert variable name, but also provides access to the **Variable Selection**. The Variable Selection window allows you to pick a Control Expert symbol (variable) name from a list of symbol (variables) that have been "Web enabled" by Web Designer for Modicon M340.

The following figure shows the **Variable Selection** dialog box:



Full Name	Name	Type	Rate	Comment
deivce.plc.byte0	Byte0	SHORT	1000	
device.plc.word0	Word0	INT	1000	
device.plc.word1	Word1	INT	1000	

Security

Three security elements are provided to help you protect your data:

- The HTML page, which contains the Graphic Editor applet ,has been placed in the *secure* directory on the Web module. You need to provide the correct password which will allow him/her to download the HTML page.
- The **Password** dialog box enables you to save/delete files or to transfer data values. This dialog is password protected. When you transfer data values, the Graphic Editor reinforces the read-only mode by deactivating the user commands related to graphic objects.
- Web Designer for Modicon M340 enables you to indicate if an element is read only (read only is set by default). The **Graphic Editor** reinforces the read only attribute of a symbol (variable) by rejecting any request which would define a new data value and by displaying a message window.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Do not use graphic objects in a situation where loss of communication to the FactoryCast module can affect human or material integrity.
- Do not use graphic objects in safety critical machine functions.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

For example, say you have programmed a pushbutton object to jog a motor when the button is depressed and to stop jogging when the button is released. If communications are lost while the button is depressed, the motor will continue to jog even when the button is released. Graphic objects should not be used to control situations such as this unless other interlock methods are installed in the system.

Graphic Editor Applet Parameters

Overview

Three applet parameters can customize the behavior of the **Graphic Editor**. Applet parameters are specified with <PARAM> tags within the <APPLET> tag in the Graphic Editor's HTML page. The parameters recognized by the **Graphic Editor** applet are:

- **LOAD**: This parameter tells the **Graphic Editor** to auto-load a specific graphics file when it starts. If the file does not exist, a message appears. If this parameter is not provided in the <APPLET> tag, then a file is not auto-loaded at startup and you have to select an initial graphics file from the list provided by the **Graphic Editor**.
- **MODE**: This parameter tells the **Graphic Editor** whether to startup in its normal "Edit Mode" or in a special "View Mode." When started in view mode, the **Graphic Editor** shows only its display window. When this parameter is used with the LOAD parameter, a Web site can be designed using HTML pages that are dedicated to specific graphic displays. No explicit selection of graphic files is required by a user, providing more typical HMI screen behavior. The possible values for this parameter are:
  - **EDIT** (default value): The **Graphic Editor** starts up in its normal Edit Mode.
  - **VIEW_RO**: The **Graphic Editor** starts up in read-only view mode. The Web browser user will not be allowed to send data values to the PLC.
  - **VIEW_RW**: The **Graphic Editor** starts up in read/write view mode. The Web browser user will be allowed to send data values to the PLC after entering the write-access password.
- **AUTO_LOGIN**: This parameter tells the **Graphic Editor** to automatically enter the password that is required to permit writing to the PLC. If the **MODE** parameter is set to **VIEW_RW** or **EDIT**, then setting this parameter to **TRUE** will cause the **Graphic Editor** to allow writing to the PLC without requiring the user to enter the password. The possible values for this parameter are **FALSE** (default) and **TRUE**.

Example

The following is an example of an applet tag for the **Graphic Editor** that causes it to start up in view mode while automatically loading a graphics file named **UNIT_1**. In this case, the Web browser allows you to send values to the PLC via any graphic objects that support sending values (assuming you have entered the write-access password).

```
<APPLET codebase="/classes"
archive="SAComm.jar,GDE.jar,Widgets.jar"
code="com.schneiderautomation.gde.GdeApplet"
width="700" height="514">
<PARAM name="LOAD" value="UNIT_1">
<PARAM name="MODE" value="VIEW_RW">
<PARAM name="AUTO_LOGIN" value="FALSE">
</APPLET>
```

Graphic Objects

Overview

The set of graphic objects provided with the **Graphic Editor** supports the construction of graphic displays that mimic conventional instrument panels. The data monitoring and control objects have built-in communication capabilities and are designed as standalone graphic objects.

Be aware, however, that if communication to the device linked to the graphic object is lost, the object becomes inoperative without the end device's knowledge.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Do not use graphic objects in a situation where loss of communication to the FactoryCast module can affect human or material integrity.
- Do not use graphic objects in safety critical machine functions.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

For example, say you have programmed a pushbutton object to jog a motor when the button is depressed and to stop jogging when the button is released. If communications are lost while the button is depressed, the motor will continue to jog even when the button is released. Graphic objects should not be used to control situations such as this unless other interlock methods are installed in the system.

Additionally, each object in the **Graphic Editor** set is available in an applet version to support customers that want to put several simple applets on a single HTML page. When used in conjunction with `LiveBeanApplet`, the **Graphic Editor** graphic objects can be used in the same way as the `LiveLabelApplet`.

This topic describes standard graphic objects and their properties.

Horizontal Indicator

A Horizontal Indicator provides an analog representation of the value of a symbol (variable) or direct address in a PLC. It is a horizontal bar that represents the value as a percentage of its range in engineering units. Optionally, a digital indication of the value can be shown in the center of the bar area.

The following table describes the properties for the Horizontal Indicator:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 124*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 124*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used on the label	
Major Scale Divisions	The number of major (labeled) scale divisions	0 to 100
Minor Scale Divisions	The number of minor (unlabeled) scale divisions	0 to 100
Scale Color	The color of the scale and its labels	
Scale Font	The font used on scale labels	
Scale Precision	The number of fractional digits to be shown for scale labels (set to -1 to use a general exponential format)	-1 to 6
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Value Visible	Indicates whether a digital display of the scaled value is to be shown	
Value Font	The font for the digital display of the value (if shown)	
Bar Background	The background color of the bar indicator area	
Bar Color	The color of the indicator bar (when the scaled value within High/Low limits)	

Property	Description	Limits
High High Limit Value	The value in engineering units for the ‘High High’ limit	
High High Limit Color	The color of the indicator bar when the scaled value is greater than the ‘High High’ limit	
High Limit Value	The value in engineering units for the ‘High’ limit	
High Limit Color	The color of the indicator bar when the scaled value is greater than the ‘High’ limit	
Low Limit Value	The value in engineering units for the ‘Low’ limit	
Low Limit Color	The color of the indicator bar when the scaled value is less than the ‘Low’ limit	
Low Low Limit Value	The value in engineering units for the ‘Low Low’ limit	
Low Low Limit Color	The color of the indicator bar when the scaled value is less than the ‘Low Low’ limit	
Limit Deadband	The deadband (as percentage of EU range) to apply to High/Low limit checking	0 to 10
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	
PLC Value	A simulated, raw (unscaled) input value for testing the graphic object	See Note 3, *Notes, page 124*

Vertical Indicator

A Vertical Indicator provides an analog representation of the value of a symbol (variable) or direct address in a PLC. It is a vertical bar that represents the value as a percentage of its range in engineering units.

The following table describes the properties of the Vertical Indicator:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 124*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 124*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Major Scale Divisions	The number of major (labeled) scale divisions	0 to 100
Minor Scale Divisions	The number of minor (unlabeled) scale divisions	0 to 100

Property	Description	Limits
Scale Color	The color of the scale and its labels	
Scale Font	The font used for scale labels	
Scale Precision	The number of fractional digits to be shown for scale labels (set to -1 to use a general exponential format)	-1 to 6
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Bar Background	The background color of the bar indicator area	
Bar Color	The color of the indicator bar (when the scaled value within High/Low limits)	
High High Limit Value	The value in engineering units for the ‘High High’ limit	
High High Limit Color	The color of the indicator bar when the scaled value is greater than the ‘High High’ limit	
High Limit Value	The value of the ‘High’ limit in engineering units	
High Limit Color	The color of the indicator bar when scaled value is greater than the ‘High’ limit	
Low Limit Value	The value of the ‘Low’ limit in engineering units	
Low Limit Color	The color of the indicator bar when the scaled value is less than the ‘Low’ limit	
Low Low Limit Value	The value of the ‘Low Low’ limit in engineering units	
Low Low Limit Color	The color of the indicator bar when the scaled value is less than the ‘Low Low’ limit	
Limit Deadband	The deadband (as percentage of EU range) to apply to High/Low limit checking	0 to 10
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	
PLC Value	A simulated, raw (unscaled) input value for testing the graphic object	See Note 3, *Notes, page 124*

Horizontal or Vertical Slider

A Horizontal or Vertical Slider provides an analog representation of the value of a symbol (variable) or direct address in a PLC. It is a scroll bar with a "thumb" position that represents the value as a percentage of its range in engineering units. With the mouse, you can change the value of the scroll bar, sending a new value to the PLC.

The following table describes the properties for the Horizontal or Vertical Slider:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 124*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 124*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Major Scale Divisions	The number of major (labeled) scale divisions	0 to 100
Minor Scale Divisions	The number of minor (unlabeled) scale divisions	0 to 100
Scale Color	The color of the scale and its labels	
Scale Font	The font used for scale labels	
Scale Precision	The number of fractional digits to be shown for scale labels (set to -1 to use a general exponential format)	-1 to 6
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Block Increment	The amount that the scaled value should change when the scroll bar’s scroll area is clicked	
Unit Increment	The amount that the scaled value should change when the scroll bar’s arrow buttons are clicked	
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	

Horizontal or Vertical Selector

A Horizontal or Vertical Selector allows you select from a set of choices. When a selection is made, the value corresponding to the choice is sent to the PLC. The choices are shown as labels of a "scale," with the current selection indicated by the position of the "thumb" of a scroll bar.

The following table describes the properties of the Horizontal or Vertical Selector:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1, *Notes, page 124*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 124*
Background	The background color for the graphic object	
Choices	The choices for the selector. Each choice is given as a ‘label=value’ entry (when you select ‘label,’ ‘value’ is sent to PLC).	Minimum of two choices required
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Scale Visible	Indicates whether a "scale," labeled with the choices, is to be shown	
Scale Color	The color of the scale and its labels	
Scale Font	The font used for scale labels	
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	

Digital Indicator

A Digital Indicator provides a numeric representation of the value of a symbol (variable) or direct address in a PLC. The value can be shown in various formats, and can be made to change color when a preset high or low limit is exceeded.

The following table describes the properties of the Digital Indicator:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 124*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 124*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Value Format	The format (decimal, hex, etc.) to use in displaying the scaled value	
Value Precision	The number of fractional digits to be shown for the scaled value (set to -1 to use a general exponential format)	-1 to 6
Value Background	The background color of the value display area	
Value Color	The text color for the digital display of the value	
Value Font	The font used for the digital display of the value	
Units	The label for the value’s engineering units (appended to the numerical display of the value)	
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
High High Limit Value	The value of the ‘High High’ limit in engineering units	
High High Limit Color	The color of the indicator bar when the scaled value is greater than the ‘High High’ limit	
High Limit Value	The value of the ‘High’ limit in engineering units	
High Limit Color	The color of the indicator bar when the scaled value is greater than the ‘High’ limit	
Low Limit Value	The value of the ‘Low’ limit in engineering units	

Property	Description	Limits
Low Limit Color	The color for the indicator bar when scaled value is less than the 'Low' limit	
Low Low Limit Value	The value of the 'Low Low' limit in engineering units	
Low Low Limit Color	The color of the indicator bar when the scaled value is less than the 'Low Low' limit	
Limit Deadband	The deadband (as percentage of EU range) to apply to High/Low limit checking	0 to 10
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	
PLC Value	A simulated, raw (unscaled) input value for testing the graphic object	See Note 3

Message Display

A Message Display shows a text message based on the value of a symbol (variable) or direct address in a PLC. For each specified message, a specified value triggers its display.

The following table describes the properties of the Message Display:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 124*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 124*
Background	The background color of the graphic object	
Messages	The set of messages to display. Each message is given as a 'value=text' entry (when the PLC value equals 'value,' 'text' is displayed as the message).	Minimum of one message required
Message Background	The background color of the message display area	
Message Color	The color of the message text	
Message Font	The font used for the message text	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	
PLC Value	A simulated input value for testing the graphic object	See Note 3, *Notes, page 124*

Push Button

A Push Button allows you to send preset value(s) to a PLC when clicked with the mouse.

The following table describes the properties of the Push Button:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 124*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 124*
Background	The background color of the graphic object	
Values	The value(s) to send to the PLC	See Note 4, *Notes, page 124*
Reset Values	The value(s) to send to the PLC after the reset delay time has expired. If no reset values are provided, no reset action will occur.	
Reset Delay	The delay time (in milliseconds) that the Push Button should wait after sending the value(s) to the PLC before sending the reset value(s).	0-2000
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Button Label	The text label for the button	
Button Background	The color of the button	0 to 100
Button Label Color	The color of the button label	
Button Label Font	The font used for the button label	
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	

Direct Output Station

The Direct Output Station allows you to enter a numeric value in a text field directly with the keyboard. When the entered value is within preset high and low limits, a **Set** button is enabled. In this case, the entered value will be sent to the PLC when you press either the **Set** button or the ENTER key (if the input field has keyboard input focus).

The following table describes the properties of the Direct Output Station:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 124*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 124*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Maximum Input	The maximum value, in engineering units, that is valid for the entered input value	
Minimum Input	The minimum value, in engineering units, that is valid for the entered input value	
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	

Indicator Light

The Indicator Light provides a dual-state indication of the value of a symbol (variable) or direct address in a PLC. Unless the **Input Inverted** property is set to **TRUE**, an input value of zero is deemed **OFF** and a non-zero value is deemed **ON**. If the **Flash Interval** property is set to greater than zero, the light will flash while the input value is on.

The following table describes the properties of the Indicator Light:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1, *Notes, page 124*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 124*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Off Word	The text to show when the input value is off	
Off Word Background	The background color of the light when the **Off Word** is shown	
Off Word Color	The color of the **Off Word** text	
Off Word Font	The font used for the **Off Word** text	
On Word	The text to show when the input value is on	
On Word Background	The background color of the light when the **On Word** is shown	
On Word Color	The color of the **On Word** font	
On Word Font	The font used for the **On Word** text	
Flash Interval	The flashing time period (in milliseconds) of the light when the input value is on. Set to zero for no flashing.	200 to 2000
Shape	The shape (circle, rectangle, etc.) of the light	
Input Inverted	If **TRUE**, inverts the input value. (Light will show the **Off Word** when input value is on.)	
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	
PLC Value	A simulated input value for testing the graphic object	See Note 3, *Notes, page 124*

Motor Control Station

The Motor Control Station is designed to mimic the typical start/stop push button station that is often used to control motors. This graphic object is essentially a composite of two push buttons and an indicator light. A Customizer is provided to make it easier to set the object's many properties. Most properties (except Name) are set with its Customizer, not with the **Graphic Editor's** Property Sheet.

The following table describes the properties of the Motor Control Station:

Property	Description	Limits
Name	The name of the graphic object	
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	
Indicator Light	Same properties as the Indicator Light graphic object, excluding the shared properties listed above	
Top Push Button	Same properties as the Push Button graphic object, excluding the shared properties listed above	
Bottom Push Button	Same properties as the Push Button graphic object, excluding the shared properties listed above	

Analog Meter

An Analog Meter provides an analog representation of the value of a symbol (variable) or direct address in a PLC. It is represented as a pointer on a circular dial; its position indicates the value as a percentage of its range in engineering units. You can set the size of the meter's circular dial (degrees sweep of a circle), the colors of the dial, and the style of the pointer.

The following table describes the properties of the Analog Meter:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 124*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 124*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	

Property	Description	Limits
Major Scale Divisions	The number of major (labeled) scale divisions	0 to 100
Minor Scale Divisions	The number of minor (unlabeled) scale divisions	0 to 100
Scale Color	The color of the scale and its labels	
Scale Font	The font used for scale labels	
Scale Precision	The number of fractional digits to be shown for scale labels (set to -1 to use a general exponential format)	-1 to 6
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Dial Degrees Sweep	The amount of a circular arc to use for drawing the dial	60 to 300
Pointer Type	The type (needle, arrow head, etc.) of pointer to use	
Pointer Color	The color for the pointer	
Dial Color	The color of the dial (that part that is within the High/Low limits)	
High High Limit Value	The value of the ‘High High’ limit in engineering units	
High High Limit Color	The color of the indicator bar when the scaled value is greater than the ‘High High’ limit	
High Limit Value	The value of the ‘High’ limit in engineering units	
High Limit Color	The color of the indicator bar when the scaled value is greater than the ‘High’ limit	
Low Limit Value	The value of the ‘Low’ limit in engineering units	
Low Limit Color	The color of the indicator bar when the scaled value is less than the ‘Low’ limit	
Low Low Limit Value	The value of the ‘Low Low’ limit in engineering units	
Low Low Limit Color	The color of the indicator bar when the scaled value is less than the ‘Low Low’ limit	
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	
PLC Value	A simulated, raw (unscaled) input value for testing the graphic object	See Note 3, *Notes, page 124*

Rotary Slider

A Rotary Slider provides an analog representation of the value of a symbol (variable) or direct address in a PLC. It is represented as a knob on a circular dial; its position indicates the value as a percentage of its range in engineering units. The dial size and knob color can be set. With the mouse, you can change the position of the knob, sending a new value to the PLC.

The following table describes the properties of the Rotary Slider:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 124*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 124*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Major Scale Divisions	The number of major (labeled) scale divisions	0 to 100
Minor Scale Divisions	The number of minor (unlabeled) scale divisions	0 to 100
Scale Color	The color of the scale and its labels	
Scale Font	The font used for scale labels	
Scale Precision	The number of fractional digits to be shown for scale labels (set to -1 to use a general exponential format)	-1 to 6
Dial Degrees Sweep	The amount of a circular arc to use for drawing the dial	60 to 300
Dial Color	The color of the dial	
Knob Color	The color used for the knob	
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	

Rotary Selector

A Rotary Selector allows you to select from a set of choices. When a selection is made, the value corresponding to the choice is sent to the PLC. The choices are shown as labels of a "scale," with the current selection indicated by the position of the knob. The size of the circular dial (degrees sweep of a circle) and knob color can be set.

The following table describes the properties of the Rotary Selector:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 124*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 124*
Background	The background color of the graphic object	
Choices	The choices for the selector. Each choice is given as a 'label=value' entry. (When you select 'label,' 'value' is sent to PLC.)	Minimum of two choices required
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Scale Visible	Indicates whether a "scale," labeled with the choices, is to be shown	
Scale Color	The color of the scale and its labels	
Scale Font	The font used for scale labels	
Dial Degrees Sweep	The amount of a circular arc to use for drawing the dial	60 to 300
Knob Color	The color of the knob	
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	

Trend Recorder

A Trend Recorder provides a continuous, time-based charting of the value of up to six symbol(s) (variables) or direct addresses in a PLC. A Trend Recorder emulates a strip-chart recorder, with the pens on the right, and the "paper" moving from right to left. A vertical scale can be shown on the left side of the chart for showing the range of the values being recorded, and a horizontal scale can be shown below the chart for showing the time span of the chart. You can set the rate at which the chart is updated, and the appearance of the chart.

A Customizer is provided to make it easier to set this object’s many properties. Most properties (except Name) are set with its Customizer, not with the **Graphic Editor’s** Property Sheet.

The following table describes properties of the Trend Recorder. Properties available for each pen are described in the next table:

Property	Description	Limits
Name	The name of the graphic object	
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Major Scale Divisions	The number of major (labeled) scale divisions	0 to 100
Minor Scale Divisions	The number of minor (unlabeled) scale divisions	0 to 100
Scale Color	The color of the scale and its labels	
Scale Font	The font used for scale labels	
Scale Precision	The number of fractional digits to be shown for scale labels (set to -1 to use a general exponential format)	-1 to 6
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Update Period	The update interval (in seconds) for the chart	0.5 to 120
Time Scale Divisions	The number of horizontal scale divisions	0 to 6
Chart Background	The color of the chart area	
Grid Color	The color of the grid drawn in the chart area	
Vertical Grid Divisions	The number of vertical divisions for the grid	0 to 100
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	

These Trend Recorder properties are available for each pen:

Property	Description	Limits
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1, *Notes, page 124*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 124*
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 124*
Pen Color	The color of the "pen" used to record the scaled value	
Pen Label	The label used to identify the pen	

Display Link

A Display Link is a special graphic object that allows you to switch to another graphic display by clicking on it with a mouse. To indicate that the object is a link to another display, the text label for the link is underlined and the mouse cursor changes to a hand icon when the mouse is moved over it. This object is especially useful when the **Graphic Editor** is used in **view mode**, where no drop-down list of graphic displays is available for selecting a display.

A Display Link can also be used as a hyperlink to an HTML file. If a URL is entered as the **Link Display Name**, the URL can be opened in a new browser window if you press the SHIFT key while clicking the link; otherwise, the existing browser window is replaced with the URL when the link is clicked.

If the **Link Display Name** is blank, then the **Label** will be shown as not underlined, and the displayed object becomes a simple text label.

The following table describes the properties of the Display Link:

Property	Description	Limits
Label	The label of the link	
Link Display Name	The name of the graphic display to be loaded when the link is clicked, or a URL of a Web page	
Label Color	The color of the label	
Label Font	The font used for the label	

Notes

These are the notes for this topic:

1. If the Address property of a graphic object is a direct address, the Data Type property is set to UNDEFINED, a default Data Type (BOOL, INT, DINT, or REAL, based on the implied size of the data value) is used. If the Address property is a symbol (variable) name, the Data Type property does not have to be specified and can be set to UNDEFINED. If, however, the Data Type property is specified for a symbol (variable), it must exactly match the symbol's (variable's) actual data type.
If the Address property is a direct address for a discrete PLC reference (Quantum 0x/1x reference), set the Data Type property to BOOL. The Data Type property may be set to BOOL only for a discreet PLC reference.

2. The meanings of the possible values of the Data Type property are:

Data Type	Meaning
`UNDEFINED`	`no data type specified`
`BOOL`	`1-bit discreet (Boolean)`
`SHORT`	`8-bit signed integer`
`USHORT`	`8-bit unsigned integer`
`INT`	`16-bit signed integer`
`UINT`	`16-bit unsigned integer`
`DINT`	`32-bit signed integer`
`UDINT`	`32-bit unsigned integer`
`REAL`	`32-bit IEEE floating point`
`TIME`	`32-bit unsigned integer (in milliseconds)`
`DATE`	`Date (32-bit BCD)`
`TOD`	`Time of Day (32-bit BCD)`
`DT`	`Date and Time (64-bit BCD)`

3. The limits for the Maximum PLC Value and Minimum PLC Value properties are the natural limits of the Data Type property that is set. A Data Type setting of UNDEFINED is treated as REAL with respect to its limit values.

4. For a Push Button, provide at least one value. If the Address property is a symbol (variable) name, then only one value will ever be sent to the PLC, and any additional values are ignored. If the Address property is a direct address, then all of the values provided are sent to the PLC as an array of values starting at the specified direct address.

Extended Graphic Objects

Overview

The extended graphic objects provided in the Graphic Editor are used to build graphic displays that mimic advanced graphic panels. All of the data monitoring and control objects have built-in communication capabilities and are designed as standalone graphic objects.

Be aware, however, that if communication to the device linked to the extended graphic object is lost, the object becomes inoperative without the end device's knowledge.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Do not use graphic objects in situations where loss of communication to the FactoryCast module can affect human or material integrity.
- Do not use graphic objects in safety critical machine functions.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

For example, say you have programmed a pushbutton object to jog a motor when the button is depressed and to stop jogging when the button is released. If communications are lost while the button is depressed, the motor will continue to jog even when the button is released. Graphic objects should not be used to control situations such as this unless other interlock methods are installed in the system.

Additionally, to support customers that want to put several simple applets on a single HTML page, each object in the Graphic Editor set is provided in an applet version. When used in conjunction with the `LiveBeanApplet`, Graphic Editor graphic objects can be used in the same way as the `LiveLabelApplet`.

ASCII Text Writer

The ASCII text writer is based on the message display widget. It allows you to input new text.

The properties of the ASCII text writer are:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 134*
Max. Text Length	The maximum length of the text	
Text Color	The color of the text	
Text Font	The font of the text	
Swap Bytes	False if target order of bytes is same as PC one.	
Value	The text itself	

Bar Graph

A bar graph provides an analog representation of the value of a symbol (variable) or direct address in a PLC. It draws a vertical bar whose length is proportional to the value as a percentage of its range in engineering units.

The properties of the bar graph are:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 134*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 134*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used on the label	
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 134*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 134*
Bar Background	The background color of the bar indicator area	

Property	Description	Limits
Bar Color	The color of the indicator bar (when scaled value within High/Low limits)	
High High Limit Value	The value of the High High limit in engineering units	
High High Limit Color	The color of the indicator bar when the scaled value is greater than the High High limit	
High Limit Value	The value of the High limit in engineering units	
High Limit Color	The color of the indicator bar when scaled value is greater than the High limit	
Low Limit Value	The value of the Low limit in engineering units	
Low Limit Color	The color of the indicator bar when the scaled value is less than the Low limit	
Low Low Limit Value	The value of the Low Low limit in engineering units	
Low Low Limit Color	The color of the indicator bar when the scaled value is less than the Low Low limit	
Limit Deadband	The deadband (as percentage of EU range) to apply to High/Low limit checking	0 to 10
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	
PLC Value	A simulated, raw (unscaled) input value for testing the graphic object	See Note 3, *Notes, page 134*

Bitmap

The bitmap widget displays a static bitmap on the screen.

The properties of the bitmap widget are:

Property	Description	Limits
Name	The name of the graphic object	
Background	The background color of the graphic object	See Note 1, *Notes, page 134*
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Border Width	The width (in pixels) of the graphic object's border	
Border Color	The color of the graphic object's border	
Bitmap Choices	Represents the filenames of the custom bitmaps to display. The default root path of the file location directory is /FLASH1/wwwroot;; ":images/ thus refers to /FLASH1/wwwroot/images/ .	

Generic Bitmap

The generic bitmap widget lets you display one static bitmap for each distinct value of a PLC variable. It can be used to display dynamic animations, for instance the changing level in a tank.

The properties of the Generic Bitmap widget are:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 134*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 134*
Background	The background color of the graphic object	See Note 1, *Notes, page 134*
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Bitmap Choices	Represents the filenames of the custom bitmaps to display. Clicking on this property opens a text editor that makes it possible to type the PLC value conditions and related bitmaps to display, such as "0:key.gif:images/" where 0 is the PLC value, "key.gif" the bitmap file related to the value, "images" the directory in which the file is located. The default root path of the file location directory is **/FLASH1/wwwroot**; **images/** thus refers to **/FLASH1/wwwroot/images/**.	
Border Width	The width (in pixels) of the graphic object's border	
Border Color	The color of the graphic object's border	
PLC Value	A simulated input value for testing the graphic object behavior.	

Graphic Link

A graphic link is a special graphic object that lets you switch to another graphic display by clicking on it. Graphic links can also be recognized by their underlined labels, and the mouse cursor changes to a hand icon when the mouse moves over them. This object is especially useful when the Graphic Editor is used in view mode, where no drop-down list of graphic displays is available.

A graphic link can also be used as a hyperlink to an HTML file. If a URL is entered as the **Link Display Name**, the URL can be opened in a new browser window by simultaneously pressing the SHIFT key and clicking the link. Otherwise, the URL opens in the existing browser window when the link is clicked.

If the **Link Display Name** is blank, then the label is not underlined, and the displayed object becomes a simple text label.

The properties of the display link are:

Property	Description	Limits
Label	The link label	
Link Display Name	The name of the graphic display to be loaded when the link is clicked, or the URL of a Web page	
Label Color	The color of the label	
Label Font	The font used on the label	
Bitmap Choices	The filename of the bitmap on which to click	

Indicator Light

The indicator light displays the value of a symbol (variable) or direct address in a PLC with two possible states. An input value of 0 is considered off and a non-zero value is considered on. If the **Flash Interval** property is set to a value greater than 0, the light flashes while the input value is on. There is a bitmap for the on-state and a different one for the off-state.

The properties of the indicator light are:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 134*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 134*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
OFF Word	The text to display when the input value is off	

Property	Description	Limits
OFF Bitmap Choice	The light bitmap when the OFF word is displayed	
OFF Word Color	The color of the OFF word text	
OFF Word Font	The font used for the OFF word text	
ON Word	The text to display when the input value is on	
ON Bitmap Choice	The light bitmap when the ON word is displayed	
ON Word Color	The color of the ON word font	
ON Word Font	The font used for the ON word text	
Flash Interval	The flashing time period (in ms) of the light when the input value is on. Set to 0 for no flashing.	200 to 2000
Input Inverted	If true, inverts the input value. (Light will show the off-word when input value is on.)	
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	
PLC Value	A simulated input value for testing the graphic object	See Note 3, *Notes, page 134*

Motor

The motor widget displays the value of a symbol (variable) or direct address in a PLC with three possible states. An input value of 0 is considered off, a value of 1 is considered on and other values are considered default. The three states are represented by different bitmaps.

The properties of the motor widget are:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 134*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 134*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
OFF Word	The text to display when the input value is off	
OFF Bitmap Choice	The motor bitmap when the OFF word is displayed	
OFFWord Color	The color of the OFF word text	
OFF Word Font	The font used for the OFF word text	
ON Word	The text to display when the input value is ON	

Property	Description	Limits
ON Bitmap Choice	The motor bitmap when the ON word is displayed	
ON Word Color	The color of the ON word font	
ON Word Font	The font used for the ON word text	
DEFAULTWord	The text to display when the input value is ON	
DEFAULT Bitmap Choice	The motor bitmap when the DEFAULT word is displayed	
DEFAULT Word Color	The color of the DEFAULT word font	
DEFAULT Word Font	The font used for the DEFAULT word text	
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	
PLC Value	A simulated input value for testing the graphic object	See Note 3, *Notes, page 134*

Pipe

The pipe displays the value of a symbol (variable) or direct address in a PLC with two possible states. An input value of 0 is considered off and a non-zero value is considered on. There is a bitmap for the on-state and a different one for the off-state.

The properties of the pipe are:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 134*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 134*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
OFF Word	The text to display when the input value is off	
OFF Bitmap Choice	The pipe bitmap when the OFF word is displayed	
OFF Word Color	The color for the OFF word text	
OFF Word Font	The font used for the OFF word text	
ON Word	The text to display when the input value is on	
ON Bitmap Choice	The pipe bitmap when the ON word is displayed	

Property	Description	Limits
ON Word Color	The color of the ON word font	
ON Word Font	The font used for the ON word text	
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	
PLC Value	A simulated input value for testing the graphic object	See Note 3, *Notes, page 134*

Push Button

A push button sends preset value(s) to a PLC when the user clicks it with the mouse.

These are the properties of the Push Button.

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 134*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 134*
Background	The background color of the graphic object	
Values	The value(s) to send to the PLC	See Note 4, *Notes, page 134*
Reset Values	The value(s) to send to the PLC after the reset delay time has expired. If no reset values are provided, no reset action will occur.	
Reset Delay	The delay time (in milliseconds) that the Push Button should use after sending the value(s) to the PLC before sending the reset value(s).	0-2000
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Button Label	The text label for the button	
Button Label Color	The color of the button label	
Button Label Font	The font used for the button label	
OFF Bitmap Choice	The button bitmap when the OFF state is displayed	
ON Bitmap Choice	The button bitmap when the ON state is displayed	
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	

Valve

The valve displays the value of a symbol (variable) or direct address in a PLC with two possible states. An input value of 0 is considered off and a non-zero value is considered on. There is a bitmap for the on-state and a different one for the off-state.

The properties of the valve are:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 134*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 134*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
OFF Word	The text to display when the input value is off	
OFF Bitmap Choice	The valve bitmap when the OFF word is displayed	
OFF Word Color	The color of the OFF word text	
OFF Word Font	The font used for the OFF word text	
ON Word	The text to display when the input value is ON	
ON Bitmap Choice	The valve bitmap when the ON word is displayed	
ON Word Color	The color of the ON word font	
ON Word Font	The font used for the ON word text	
Flash Interval	The flashing time period (in ms) of the light when the input value is on. Set to 0 for no flashing.	200 to 2000
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	
PLC Value	A simulated input value for testing the graphic object	See Note 3, *Notes, page 134*

Notes

These are the notes for the chapter.

<table>
<tr><td>1.</td><td colspan="2">If the address property of a graphic object is a direct address, the data type property is set to **UNDEFINED**, a default data type (**BOOL**, **INT**, **DINT**, or **REAL** based on the implied size of the data value) is used. If the address property is a symbol (variable) name, the data type property does not have to be specified and can be set to **UNDEFINED**. If, however, the data type property is specified for a symbol, it is valid only if it exactly matches the symbol's actual data type.
If the address property is a direct address for a Quantum 0x/1x reference, the data type property must be set to **BOOL**. The data type property may be a **BOOL** only for a discrete PLC reference.</td></tr>
<tr><td rowspan="15">2.</td><td colspan="2">The meaning of the possible values of the Data Type property are:</td></tr>
<tr><td>Data Type</td><td>Meaning</td></tr>
<tr><td>UNDEFINED</td><td>no data type specified</td></tr>
<tr><td>BOOL</td><td>1-bit discreet (Boolean)</td></tr>
<tr><td>SHORT</td><td>8-bit signed integer</td></tr>
<tr><td>USHORT</td><td>8-bit unsigned integer</td></tr>
<tr><td>INT</td><td>16-bit signed integer</td></tr>
<tr><td>UINT</td><td>16-bit unsigned integer</td></tr>
<tr><td>DINT</td><td>32-bit signed integer</td></tr>
<tr><td>UDINT</td><td>32-bit unsigned integer</td></tr>
<tr><td>REAL</td><td>32-bit IEEE floating point</td></tr>
<tr><td>TIME</td><td>32-bit unsigned integer (in milliseconds)</td></tr>
<tr><td>DATE</td><td>Date (32-bit BCD)</td></tr>
<tr><td>TOD</td><td>Time of Day (32-bit BCD)</td></tr>
<tr><td>DT</td><td>Date and Time (64-bit BCD)</td></tr>
<tr><td>3.</td><td colspan="2">The limits for the Maximum PLC Value and Minimum PLC Value properties are the natural limits of the Data Type property that is set. A Data Type setting of UNDEFINED is treated as REAL with respect to its limit values.</td></tr>
<tr><td>4</td><td colspan="2">For a Push Button, enter at least one value. If the Address property is a symbol name, only one value is sent to the PLC, and any additional values are ignored. If the Address property is a direct address, all of the values are sent to the PLC as an array, starting at the specified direct address.</td></tr>
</table>

Section 4.5
Graphic Viewer

Graphic Viewer

Overview

Graphic Viewer is a lighter version of the Graphic Editor. Its small size enables you to download it faster. With Graphic Viewer you can only display widgets. You can not edit them.

Graphic Viewer is divided into two windows:

- **Top Window**: This area shows the user controls. This is where you can select a graphic to edit from a drop-down menu
- **Display Window**: This area displays the selected graphic.

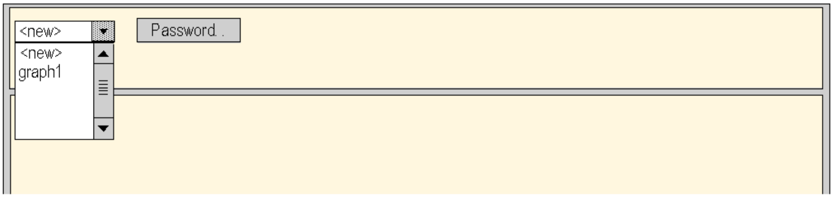
When viewing a graphic in full-screen mode, the **Top Window** is not displayed.

Selecting a Graphic

The following instructions tell you how to select a graphic created by the Graphic Editor:

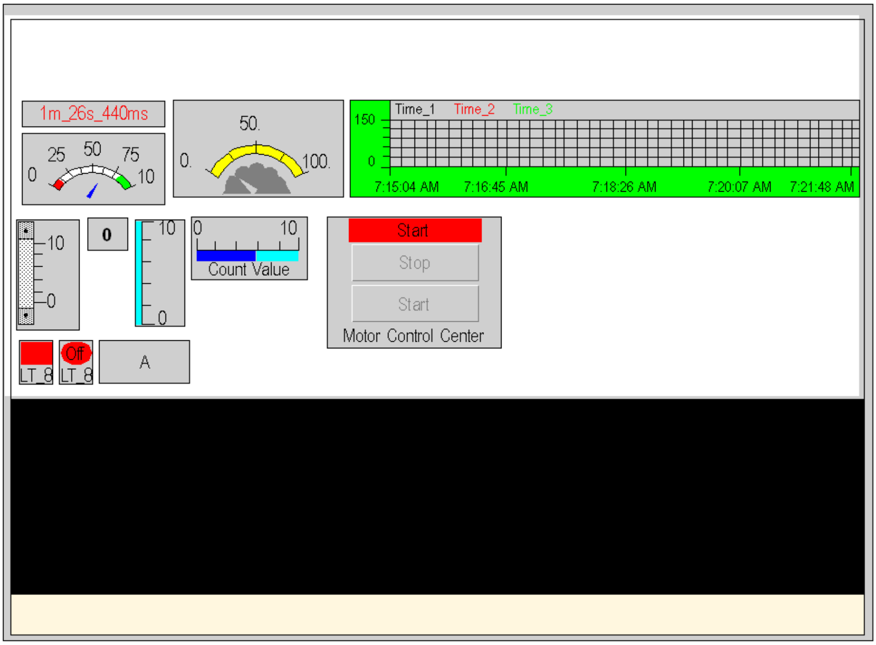
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>

Click on **Graphic Viewer**.
The **Top and Display Windows** appear:



</td></tr>
<tr><td>2</td><td>

Select a graphic from the drop-down menu.
Result: The selected graphic appears in the **Display Window**:



Note: Double-click in the **Display Window** area in the active graphic (unless the active graphic is a command widget) to refresh the **Top Window**.

</td></tr>
</table>

Section 4.6
PLC Program Viewer

PLC Program Viewer

Presentation

The PLC program viewer feature enables you to visualize and monitor Control Expert programs in run mode using a Web browser. PLC programs can be displayed in the following languages:

- IEC Ladder (LD)
- Instruction List (IL)
- Function Block Diagram (FBD)
- LL984 (984)
- Structured Text (ST)
- Sequential Function Chart (SFC)

For FactoryCast versions higher than 4.6, the FactoryCast module detects any changes to the PLC program, then automatically updates and synchronizes the web display to the PLC program.

Enabling Automatic Synchronization of PLC Program Viewer

For FactoryCast versions higher than 4.65, you can configure your Control Expert application so that the FactoryCast Ethernet communication module will automatically detect any changes to the PLC program, then automatically synchronize the web display to the PLC program.

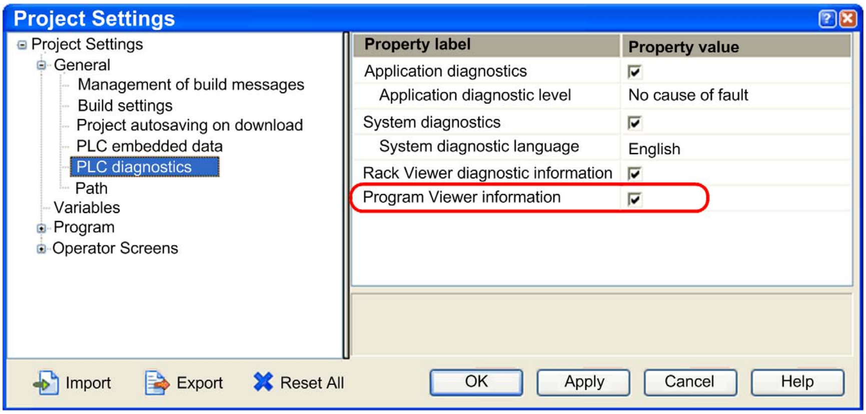
To enable automatic synchronization of the Control Expert application running in the PLC with the **PLC Program Viewer** display, you need to:

- enable program viewer information display
- embed data dictionary of variables in PLC memory

Use Control Expert to enter these settings, as described below.

To enable the **PLC Program Viewer** display, follow these steps:

Step	Action
1	In Control Expert, select **Tools** → **Project Settings...** to open the **Project Settings window**.
2	Select **Project Settings** → **General** → **PLC diagnostics** in the tree control on the left side of the window.
3	Select **Program Viewer information** as displayed below, then click **Apply**.

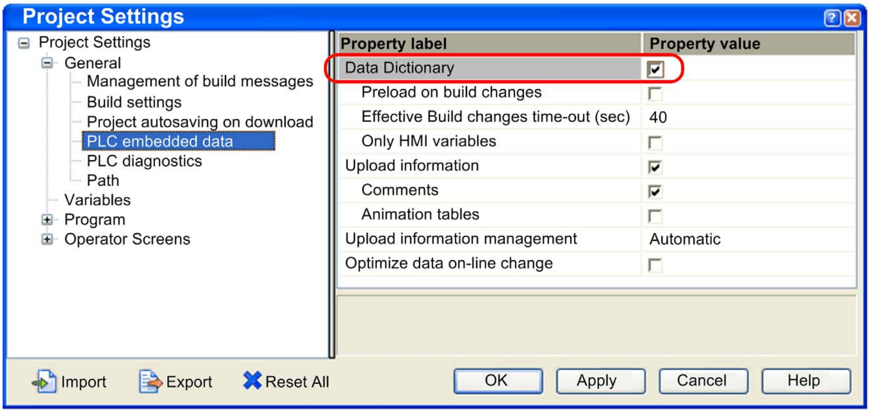


Property label	Property value
Application diagnostics	☑
Application diagnostic level	No cause of fault
System diagnostics	☑
System diagnostic language	English
Rack Viewer diagnostic information	☑
Program Viewer information	☑

NOTE: If you do not select **Program Viewer information**, as described above, the Java version of the **PLC Program Viewer** page will display.

To embed the data dictionary of PLC variables in PLC memory, follow these steps:

Step	Action
1	In Control Expert, select **Tools** → **Project Settings...** to open the **Project Settings window**.
2	Select **Project Settings** → **General** → **PLC embedded data** in the tree control on the left side of the window.
3	Select **Data Dictionary** as displayed below, then click **OK**.



Property label	Property value
Data Dictionary	☑
Preload on build changes	□
Effective Build changes time-out (sec)	40
Only HMI variables	□
Upload information	☑
Comments	☑
Animation tables	□
Upload information management	Automatic
Optimize data on-line change	□

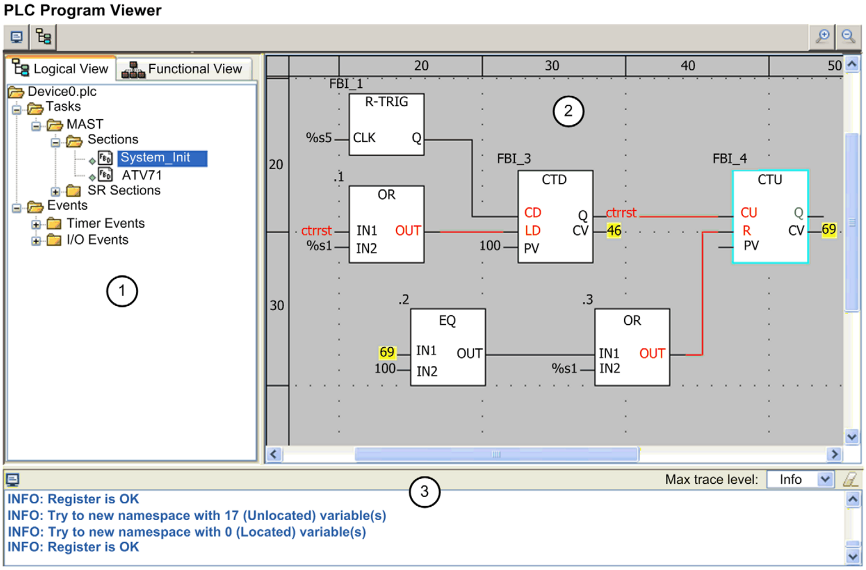
Accessing the PLC Program Viewer

The procedure below shows how to access the PLC Program Viewer page:

Step	Action
1	Launch the website of your module using a Web browser.
2	Click the **Monitoring** link on the Home page of the website.
3	Click the **PLC Program Viewer** link on the Monitoring page.

Representation and Color Convention

The PLC Program Viewer window:



INFO: Register is OK
INFO: Try to new namespace with 17 (Unlocated) variable(s)
INFO: Try to new namespace with 0 (Located) variable(s)
INFO: Register is OK

FactoryCast™, Copyright © 1998-2012, Schneider Electric All Rights Reserved.

1 Navigation tree: select the section of the PLC program you want to visualize
2 Display window: this zone displays the animated PLC program
3 Console: displays selected trace event data

Variables animation:

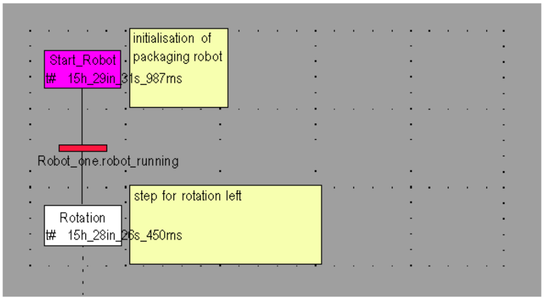
- Boolean are displayed in:
  - green if its value is true
  - red if its value is false

- Other types display the name or the value of the variable in yellow. Use the tool tip to see more information as the name of the variable, its type, its address and its comment.

Links animation:

- Links connected to boolean variables are displayed in green or red depending on the value of the variable they are connected to (green if true red if false).
- Other links are displayed in black.

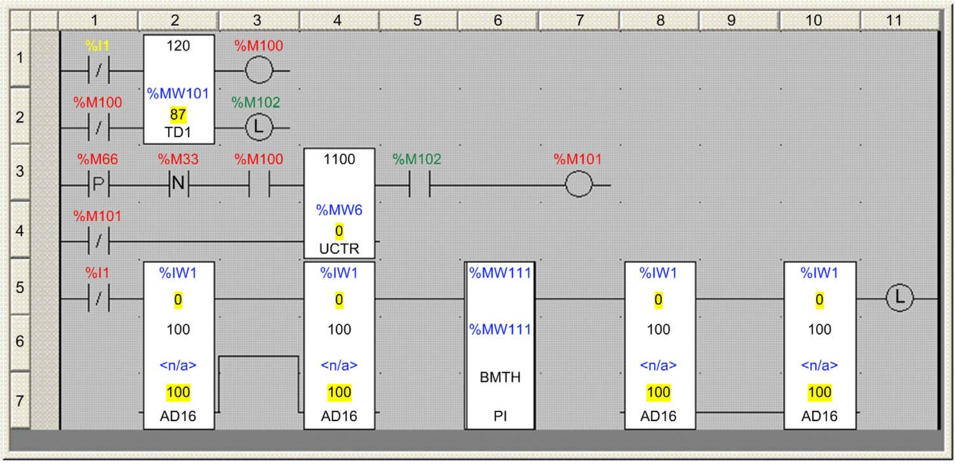
SFC animation:



The colors used for the different elements are:

- for steps:
  - green if the step is active,
  - white if the step is inactive,
  - yellow if the activity time of the step is less than the minimum programmed time,
  - pink if the activity time of the step is greater than the minimum programmed time.
- for macro-steps:
  - when a macro-step becomes active the upper half is shown in green,
  - when the OUT step of the macro-step is active the whole of the macro-step is shown in green,
  - when the macro-step becomes inactive it is then shown in white.
- for transitions associated with a Boolean element or a simple Boolean expression:
  - green if the element or the expression is TRUE,
  - red if the element or the expression is FALSE.
- for transitions associated with a section:
  - black as long as the previous step remains inactive,
  - green if the conditions in the section are TRUE,
  - red if the conditions in the section are FALSE.

LL984 animation:



The colors used for the different variables, and their meanings, are:
- Black: not monitored
- Yellow: real time value isn’t available
- Red: a boolean set to false
- Green: a boolean set to true
- Black with yellow background: a non-boolean (for example, a constant)

Tool Tip

The tool tip function is a help bubble which is displayed when you move the cursor over a variable.

It displays information about:
- The value of the variable if only its name is visible in the viewer.
- The type, name, address and comment if only its value is visible in the viewer.

Click on the variable to display the bubble permanently. Right click on it to make it disappear.

Zoom In / Out

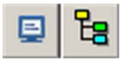
The PLC program viewer presents two buttons you can use to zoom in and zoom out of the program display. These buttons are located at the top right corner of the PLC program viewer:

Click on the button on the left (with the “+” sign) to zoom in up to three magnification levels.

Click on the button on the right (with the “–” sign) up to three times to zoom out and display the entire program in the PLC program viewer.

Show / Hide Console and Navigation Tree

The PLC program viewer presents two buttons you can toggle to show or hide the the PLC program viewer’s console and navigation tree. These buttons are located at the top left corner of the PLC program viewer:



Toggle the button on the left to show or hide the console.

Toggle the button on the right to show or hide the navigation tree.

Limitations

- Only PLC programs created using Unity Pro 4.0 or later can be viewed.
- You can only monitor PLC programs, changes are not allowed.
- The following objects are not animated, they appear in black:
  - Objects for which the result depends on an expression
  - Uninstantiated function blocks for which there is no information on the input/output variables
  - Standard DFB (i.e., ALARM_DIA)
  - Multiple dimension tables

NOTE: Unity Pro is the former name of Control Expert for version 13.1 or earlier.

Parameters

You can launch the PLC program viewer in two ways:

- from the **Home** page, by selecting **Monitoring → Plc program viewer**
- by typing into your browser the full URL for the PLC program viewer page

The full URL syntax for the PLC program viewer page is:

“http://<IP address>/unsecure/system/plcv.htm? showTreeview = 1&showConsole = 1&
section = <sectionName>|<ObjectID>@<deviceName>.plc”

When you open this page using its full URL, only the PLC program viewer itself is displayed. The web page header and the main menu (that ordinarily appear at the top of the page), and the Monitoring menu (that normally appears at the left of the page) are not displayed.

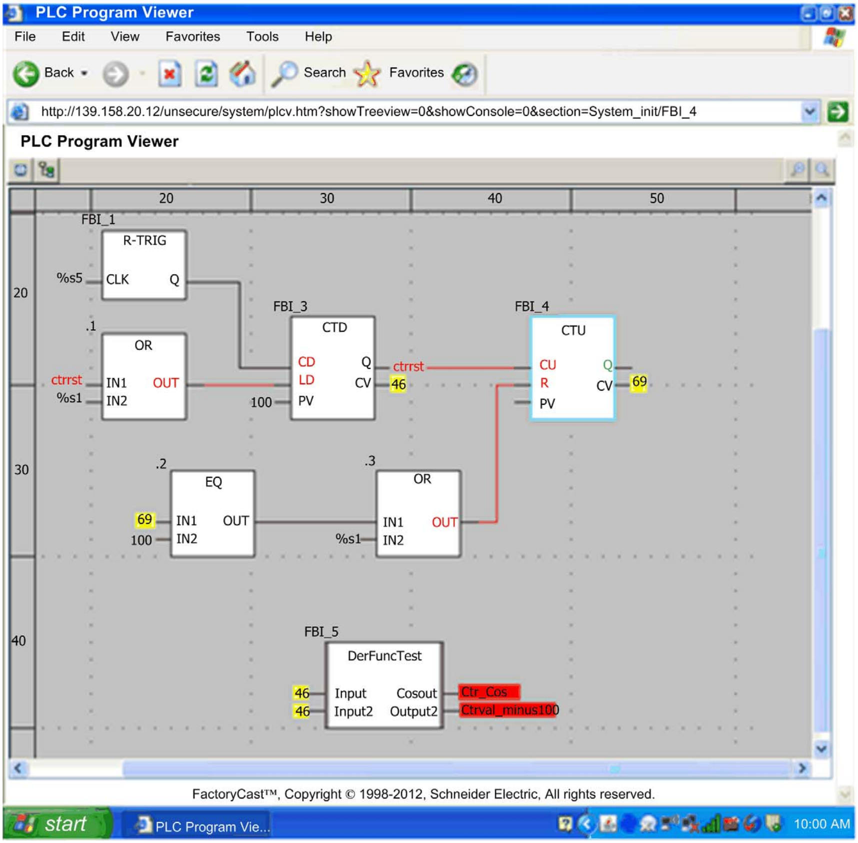
In addition, you can configure parameters contained in the URL to show or hide the Treeview control (at the left of the PLC program viewer), to show or hide the Console (at the bottom of the PLC program viewer), and to focus on a specific section or object in the PLC program. These parameters are:

- showTreeview: 0 to hide the treeview, 1 to show it.
- showConsole: 0 to hide the console, 1 to show it.
- section: access to a program section or to an object in the section. The applets directly display the section or object if it is found. The syntax for this parameter is:
  section = sectionName|ObjectID@deviceName.plc

An example of a full URL address that presents only the PLC program viewer display window is:

“http://139.158.20.12/unsecure/system/plcv.htm?showTreeview = 0 &showConsole = 0& section = System_Init|FBI_4@device0.plc”

Because only the program is displayed, it is easier to monitor program execution, as shown below:



Chapter 5
Adding Custom Pages to the Site

Overview

You may choose to add your own Web pages to the site on the embedded Server. Web Designer allows you to use the same passwords for both your custom pages and the default pages. You can also put them in an unprotected area where anyone can view them without a password.

FactoryCast provides you animated graphical objects. These objects are written in HTML so you can use them in your own Web pages. They enable you to monitor and control PLC variables by associating PLC variables with objects.

This section discusses how to create you own HTML web pages and how to use the real time animated objects provided with FactoryCast.

NOTE:
When planning custom Web pages, keep them within the limits of the memory available for customization:

- If you are using a third party software to transfer a website and if you reach the limits of the memory, some pages of the website will not appear.
- If you are using Web Designer to transfer a website, it will prompt you if your website exceeds the memory capacity.

What Is in This Chapter?

This chapter contains the following sections:

Section 5.1
Using FactoryCast's Java Applets Using HTML Code

Overview

This section describes how to use the Java applets that come with FactoryCast. Use these applets to create custom Web pages. To create custom pages, use any plain text editor, such as Microsoft Notepad.

Creating custom Web pages with HTML makes it possible to view live PLC data in your browser.

What Is in This Section?

This section contains the following topics:

Inserting Applets on a Web Page

Overview

FactoryCast software includes several graphic objects (or JavaBeans) that are used to visualize live PLC data on a graphic display. Generally, an end-user builds a graphic display using the **Graphic Editor** tool that comes with FactoryCast. However, you can also build a graphic display using graphic objects inserted into a Web page via the LiveBeanApplet. Any of the graphic objects, such as an Analog Meter or a Push Button, can be inserted into a Web page so that dynamic data may be visualized outside the context of the Graphic Editor.

To view live PLC data with a browser, you may opt for one of two methods to insert a FactoryCast applet on a Web page.

- (1) Enter the HTML code found in these sections:
  - Inserting a `LiveBeanApplet` using HTML code
  - Inserting a `LiveLabelApplet` using HTML code
  - Inserting a `LiveTableApplet` using HTML code
- (2) Insert a Java applet and then fill in the dialog boxes using the FactoryCast extension for Microsoft FrontPage or Expression Web.

Inserting LiveBeanApplet

Overview

This topic discusses general concepts about inserting a `LiveBeanApplet`.

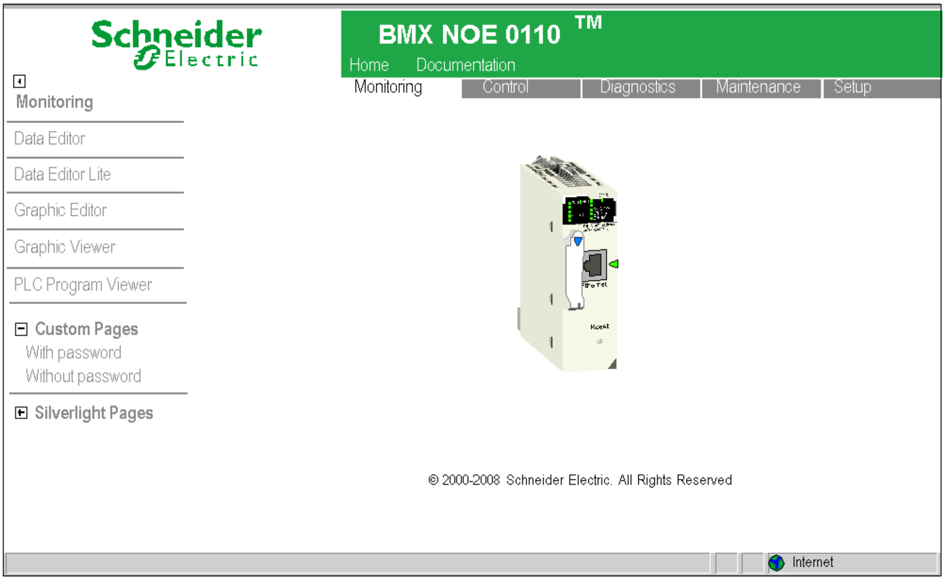
NOTE:
Elsewhere in this guide is information for inserting a `LiveBeanApplet` using:

- HTML *(see page 151)*
- Microsoft’s FrontPage Extension *(see page 173)*

NOTE: Before inserting a `LiveBeanApplet` into a Web page, create a JavaBeans library using the Graphic Editor that comes with the FactoryCast software. Generally you should create a JavaBeans library that has one instance of every object that you want to use in a Web page. Think of this library as a set of templates that are copied to and customized for your Web pages. For example, a library may have one analog meter, one rotary selector, and one pushbutton. Multiple instances of each bean can then be added to a Web page, each with a set of unique parameters such as an address.

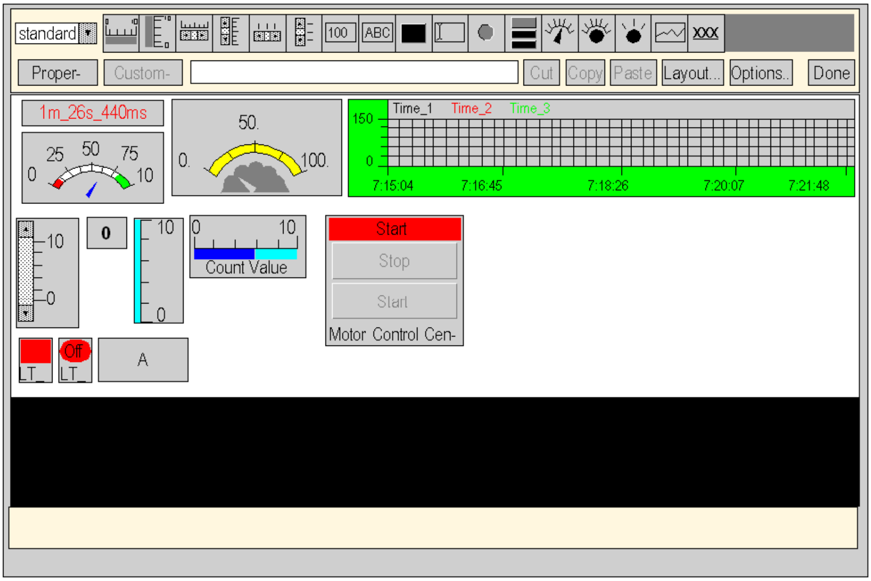
Inserting a LiveBeanApplet

To insert a `LiveBeanApplet`:

Step	Action
1	Using your Web browser, enter your Web server module’s name or IP address and go to the Graphic Editor applet found via the Monitoring and Online Editors Web link.  Schneider Electric BMX NOE 0110 TM Home Documentation Monitoring Control Diagnostics Maintenance Setup Monitoring Data Editor Data Editor Lite Graphic Editor Graphic Viewer PLC Program Viewer Custom Pages With password Without password Silverlight Pages © 2000-2008 Schneider Electric. All Rights Reserved Internet

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>

Enter the **Edit** mode and create a JavaBeans library you want to use on your Web pages. Customize each bean with a common set of parameters so that you only have to configure unique parameters, such as the register address.



Note: For help using the Graphic Editor applet, see the section Graphic Editor *(see page 95)*.

</td></tr>
<tr><td>3</td><td>Once you have created your library, click the <b>Done</b> button in the editor window and save your library.</td></tr>
</table>

Inserting LiveBeanApplet using HTML Code

Overview

To graphically visualize data, use graphic objects such as Analog Meter or Push Button. Before any beans are inserted into a Web page, install a special applet—called `LiveBeanMgrApplet`—on the server.

LiveBeanMgrApplet

The `LiveBeanMgrApplet` allows the Web page to display dynamic data from the controller. This applet must be included once on the page if any instances of `LiveBeanApplet` are included in the page.

The `LiveBeanMgrApplet` can be included on a Web page in two possible forms.

- Invisible applet—if the Web page is used only to monitor PLC values, then no input is needed from the user
- Icon of a key—if the Web page is used both to send new values and to monitor values to the PLC, then input is needed from the user in order to send new values.

NOTE: If the applet is used as an icon of a key, the user enters a password in order to send values to the PLC. From the Web browser click on the applet (icon of a key), a dialog box appears requesting the user to enter a password; entering the password enables the user to write to the PLC.

Here is the HTML code that you use to include the applet on a Web page that is used only for monitoring:

```
<APPLET codebase="/classes"
archive="SAComm.jar,GDE.jar,Widgets.jar"
code="com.schneiderautomation.gde.LiveBeanMgrApplet"
width=0 height=0>
</APPLET>
```

Here is the HTML code that you use to include the applet on a Web page that is used for sending values to a PLC as well as monitoring:

```
<APPLET codebase="classes"
archive="SAComm.jar,GDE.jar,Widgets.jar"
code="com.schneiderautomation.gde.LiveBeanMgrApplet"
width=32 height=32>
<PARAM name=MODE value="READWRITE">
<PARAM name=AUTO_LOGIN value="FALSE">
<PARAM name=FTP_ROOT value="/SDCA/Web">
</APPLET>
```

In the above example, if the value of the **AUTO_LOGIN** parameter is set to **TRUE**, instead of **FALSE**, then the password is automatically entered and the user is not required to enter the password.

LiveBeanApplet

The `LiveBeanApplet` is included one time for each symbol (variable) or direct address monitored/controlled on the Web page. For instance, if you are monitoring three symbols (variables), you would include the applet three times. `LiveBeanApplet` allows any graphic object/Java Bean that was created with the **Graphic Editor** *(see page 95)* to be included on a Web page as a separate applet. Any graphic object that has been saved as part of a **Graphic Editor** graphic display can be retrieved from the graphic file and presented by the applet.

LiveBeanApplet Parameters

The `LiveBeanApplet` uses parameters that allow you to specify the graphic object to be presented by the applet and to set the applet's background color.

The applet's parameters and their meanings are shown below.

Parameter...	Defines...
LIBRARY	The name of the graphic display which contains the graphic object that is to be presented by the applet. (This will be the same name that was used when the graphic display was saved with the **Graphic Editor**.) **This parameter is required.**
BEAN	The name of the graphic object that is to be retrieved from the graphic display specified by the LIBRARY parameter. (This will be the name that appears as the 'Name' property of the graphic object.) **This parameter is required.**
BACKGRND	The background color for the applet. Acceptable values are WHITE, LT_GRAY, GRAY, DK_GRAY, BLACK, RED, PINK, ORANGE, YELLOW, GREEN, MAGENTA, CYAN, and BLUE. Also, a RGB color value can be entered using the format "0xRRGGBB" where RR, GG, and BB are the hexadecimal values for the red, green, and blue components, respectively. This parameter is optional but is normally set to match the color of the HTML page.

In addition to the above parameters, the <APPLET> tag for a `LiveBeanApplet` must include **width** and **height** attributes. Normally, the size of a `LiveBeanApplet` is set to match the size of the graphic object that it is presenting. To get the size of a graphic object, select the object while the **Graphic Editor** is in editing mode. The selected object's name and size are shown in the **Information Area** at the top of the **Graphic Editor** applet.

LiveBeanApplet Example

All instances of `LiveBeanApplet` that are included in a Web page follow the same pattern, with only the applet's parameters and size varying for each instance.

Here is the HTML code for including a `LiveBeanApplet` that will present the graphic object named **MyMeter** which was saved by the **Graphic Editor** as part of the graphic display, **Library1**.

```
<APPLET codebase="/classes"
archive="SAComm.jar,GDE.jar,Widgets.jar"
code="com.schneiderautomation.gde.LiveBeanApplet"
width=180 height=160>
<PARAM name=LIBRARY value="Library1">
<PARAM name=BEAN value="MyMeter">
<PARAM name=BACKGRND value="0xDDEEFF">
</APPLET>
```

More Information

For more information about creating Java applets and graphic objects to obtain runtime data from a PLC, refer to the Software Developer's Kit included in the FactoryCast installation.

Inserting LiveLabelApplet using HTML Code

Overview

Before any live labels are inserted into a Web page, install a special applet—called `LiveLabelMgrApplet`—on the page.

NOTE: However, if a Web page contains both `LiveLabelApplet` and `LiveBeanApplet`, then that page must contain a single instance of `LiveBeanMgrApplet`, not `LiveLabelMgrApplet`. (`LiveBeanMgrApplet` supports both `LiveLabelApplet` and `LiveBeanApplet`, while `LiveLabelMgrApplet` supports only `LiveLabelApplet`.)

LiveLabelMgrApplet

The `LiveLabelMgrApplet` allows the Web page to display dynamic data from the controller. This applet must be included once on the page if any instances of `LiveLabelApplet` are included on the page.

Here is the HTML code that you use to include the applet on a page.

```
<APPLET>
codebase="/classes" archive="SAComm.jar"
code="com.schneiderautomation.factorycast.LiveLabelMgrApplet"
<PARAM Name=FTP_ROOT value="/SDCA/Web">
width=0 height=0>
</APPLET>
```

LiveLabelApplet

Use one `LiveLabelApplet` for every symbol (variable) or direct-address monitored on the Web page used. For example, if you are monitoring three symbols (variables), you would include the applet three times.

This applet displays the following three fields.

Field	Description
Label	Your label for the symbol (variable) or direct address
Value	Run-time value of the symbol (variable) or direct address
Units	The units you specify for the value

Data Parameters

The applet’s parameters, their meaning, and the default values are shown below.

Parameter...	Defines...		With Default Value of...
LABEL	A text label to identify the data item		No label
UNITS	A text label to identify the value’s engineering units		No units displayed
ADDRESS	The name of Concept/PL7/Control Expert symbol (variable) or Quantum/Premium direct address		None
DATATYPE	The data type of the symbol (variable) or direct address. Acceptable values for this parameter are		UNDEFINED
	SHORT	8-bit signed integer	
	USHORT	8-bit unsigned integer	
	INT	16-bit signed integer	
	UINT	16-bit unsigned integer	
	DINT	32-bit signed integer	
	UDINT	32-bit unsigned integer	
	REAL	32-bit IEEE floating point	
	TIME	32-bit unsigned integer (in ms)	
	DATE	Date (32-bit BCD)	
	TOD	Time of Day (32-bit BCD)	
	DT	Date and Time 64-bit BCD	
	BOOL	1-bit discrete (boolean)	
NOTES: If the ADDRESS parameter is a direct address, and the DATATYPE parameter is not specified, a default DATATYPE (BOOL, INT,DINT or REAL based on the implied size of the data value) is used. If ADDRESS is a direct address for a Quantum 0x/1x reference, set the DATATYPE to BOOL. DATATYPE may be set to BOOL only for discrete PLC references. If the ADDRESS parameter is the name of a Concept, PL7 or Control Expert symbol, the DATATYPE parameter is optional. If the DATATYPE is specified for a symbol, it must exactly match its actual data type. TIME is not a valid data type for PL7 Premium.			

<table>
<tr><th>Parameter...</th><th colspan="2">Defines...</th><th>With Default Value of...</th></tr>
<tr><td rowspan="8">FORMAT</td><td colspan="2">The display format for the value.
Acceptable values for this parameter are</td><td rowspan="8">DEC for most data types
TIME for data type TIME
BOOL for data type BOOL
DATE for data types DATE, TOD and DT.</td></tr>
<tr><td>DEC</td><td>decimal</td></tr>
<tr><td>HEX</td><td>hexadecimal</td></tr>
<tr><td>BIN</td><td>binary</td></tr>
<tr><td>ASCII</td><td>bytes displayed as ASCII characters</td></tr>
<tr><td>TIME</td><td>‘day_hr_min_sec_ms’</td></tr>
<tr><td>DATE</td><td>‘yyyy-mm-dd-hh and /or hh:mm:ss</td></tr>
<tr><td>BOOL</td><td>ON_WORD or OFF_WORD (see below)</td></tr>
<tr><td colspan="4"><b>NOTE:</b> If DATA TYPE is REAL, a FORMAT other than DEC will give unpredictable results if the value cannot be converted to an integer.</td></tr>
<tr><td>GAIN</td><td colspan="2">The gain (multiplier) used for scaling the retrieved value to engineering units.</td><td>1.0</td></tr>
<tr><td colspan="4"><b>NOTE:</b> Scaling is to be performed only if GAIN or BIAS is set and FORMAT is DEC.
Linear scaling is performed by the formula: SCALED_VALUE=GAINxRAW_VALUE+BIAS</td></tr>
<tr><td>BIAS</td><td colspan="2">The bias (offset) used for scaling the retrieved value to engineering units. See NOTE for GAIN.</td><td>0.0</td></tr>
<tr><td>ON_WORD</td><td colspan="2">A text value to be shown when value is non-zero
(Use only if the FORMAT is BOOL).</td><td>ON</td></tr>
<tr><td>OFF_WORD</td><td colspan="2">A text value to be shown when value is zero
(Use only if the FORMAT is BOOL).</td><td>OFF</td></tr>
<tr><td>FOREGRND</td><td colspan="2">Foreground color of the applet.
Acceptable values are: WHITE, LT_GRAY, DK_GRAY, BLACK, RED, PINK, ORANGE, YELLOW, GREEN, MAGENTA, CYAN, and BLUE
Also, a RGB color value can be entered using the format "0xRRGGBB" where RR, GG, and BB are the hexadecimal values for the red, green, and blue components, respectively.</td><td>BLACK</td></tr>
<tr><td>BACKGRND</td><td colspan="2">Background color for the applet.
For acceptable values, see FOREGRND.</td><td>LT_GRAY</td></tr>
<tr><td>ERROR_ COLOR</td><td colspan="2">Foreground color of the VALUE field when unable to retrieve the value from the PLC.
For acceptable values, see FOREGRND.</td><td>MAGENTA</td></tr>
<tr><td>LABEL_ ALIGN</td><td colspan="2">Alignment of the text in the LABEL field, if the width of the field is greater than the length of the text.
Acceptable values are: LEFT, CENTER, and RIGHT.</td><td>LEFT</td></tr>
<tr><td>VALUE_ ALIGN</td><td colspan="2">Alignment of the text in the VALUE field, if the width of the field is greater than the length of the text.
Acceptable values are: LEFT, CENTER, and RIGHT.</td><td>LEFT</td></tr>
</table>

Parameter...	Defines...	With Default Value of...
UNITS_ ALIGN	Alignment of the text in the UNITS field, if the width of the field is greater than the length of the text. Acceptable values are: LEFT, CENTER, and RIGHT.	LEFT
FONT_ NAME	Name of the font used by the applet. Acceptable values are: SERIF, SANSSERIF, and MONOSPACE.	SANSSERIF
FONT_ BOLD	If set, displays text in the applet as bold. Acceptable values are: TRUE and FALSE.	FALSE
FONT_ ITALIC	If set, displays text in the applet in italics. Acceptable values are: TRUE and FALSE.	FALSE
FONT_SIZE	Sets the point size of the font used by the applet.	12
LABEL_ WIDTH	The width of the LABEL field.	
UNITS_ WIDTH	The width of the UNITS field.	

Size Parameters

The size of a `LiveLabelApplet` is specified in the width and height attributes of its <APPLET> tag. Unless the width of the Label or Units field is set with the LABEL_WIDTH or UNITS_WIDTH parameters, display the text values of the associated applet parameters with the proper width. The remaining width of the applet is given to its VALUE field.

LiveLabelapplet Example #1

The applet example in this section contains almost every applet parameter. Here is the HTML code for this example.

```
<APPLET codebase="/classes" archive="SAComm.jar"
code="com.schneiderautomation.factorycast.LiveLabelApplet"
width=300 height=30>
<PARAM name=LABEL value="Reactor 1 Temperature">
<PARAM name=UNITS value="F">
<PARAM name=ADDRESS value="40101">(ForPremium value="%MW100")
<PARAM name=DATATYPE value="UINT">
<PARAM name=FORMAT value="DEC">
<PARAM name=GAIN value="2.0">
<PARAM name=BIAS value="100.0">
<PARAM name=FOREGRND value="WHITE">
<PARAM name=BACKGRND value="BLACK">
<PARAM name=ERROR_COLOR value="RED">
<PARAM name=FONT_NAME value="SERIF">
<PARAM name=FONT_BOLD value="TRUE">
<PARAM name=FONT_ITALIC value="FALSE">
<PARAM name=FONT_SIZE value="10">
</APPLET>
```

LiveLabelApplet Example #2

This is an example of a minimal applet, using default values for most parameters. Here is the HTML code for this example.

```
<APPLET codebase="/classes" archive="SAComm.jar"
code="com.schneiderautomation.factorycast.LiveLabelApplet"
width=300 height=30>
<PARAM name=LABEL value="Reactor 1 Pressure">
<PARAM name=UNITS value="PSI">
<PARAM name=ADDRESS value="PT_101">
</APPLET>
```

Inserting LiveTableApplet using HTML Code

Overview

LiveTableApplet is used to display dynamic, runtime PLC data on a Web page, and LiveTableApplet operates in a manner similar to LiveLabelApplet. However, there is a difference: LiveTableApplet can show multiple input values using a tabular format. Only a single input value can be shown with LiveLabelApplet.

LiveTableApplet

LiveTableApplet supports applet parameters for

- Setting the number of rows in its table (N_ROWS)
- Default settings for a row's properties
- Unique settings for each row's properties

The properties (default or unique) that can be set for a row of LiveTableApplet are the same properties that can be set for LiveLabelApplet. (The parameter names are the same as those for LiveLabelApplet, except that **Rx_** prefixes them, where **x** equals the applicable row number. A default setting is specified by not including the **Rx_** prefix.

LiveTableApplet Example

This is an example of `LiveTableApplet` that has two rows and an overall width of 200. The common row properties set the width of the **Label** field to 100 and the width of the **Units** field to 40 (leaving a width of 60 for the **Value** field). Also, every row has a black background with white text of size-10 font. The **Label** text is centered; the **Value** text is right-aligned; and the **Units** text is left-aligned. The address, data type, and the text for the **Label** and **Units** fields are set individually for each of the two rows.

Here is the HTML code for this example.

```
<APPLET codebase="/classes" archive="SAComm.jar"
code="com.schneiderautomation.factorycast.LiveTableApplet"
width="200" height="40">
<PARAM name=N_ROWS value="2">
<PARAM name=LABEL_WIDTH value="100">
<PARAM name=UNITS_WIDTH value="40">
<PARAM name=BACKGRND value="BLACK">
<PARAM name=FOREGRND value="WHITE">
<PARAM name=FONT_SIZE value="10">
<PARAM name=LABEL_ALIGN value="CENTER">
<PARAM name=VALUE_ALIGN value="RIGHT">
<PARAM name=UNITS_ALIGN value="LEFT">

<PARAM name=R1_LABEL value="Reactor Pressure">
<PARAM name=R1_UNITS value="PSIG">
<PARAM name=R1_ADDRESS value="400101">
» (for Premium value="%MW101")
<PARAM name=R1_DATATYPE value="INT">

<PARAM name=R2_LABEL value="Reactor Temperature">
<PARAM name=R2_UNITS value="F">
<PARAM name=R2_ADDRESS value="400102">
» (for Premium value="%MW102")
<PARAM name=R2_DATATYPE value="INT">
</APPLET>
```

Section 5.2
Using FactoryCast Extension for Microsoft FrontPage or Expression Web

Overview

The *Using FactoryCast's Java Applets* section described how to add FactoryCast's Java applets to a HTML document using any text editor. This section describes using an extension for Microsoft's FrontPage or Expression Web application. The extension allows a user of FrontPage or Expression Web to easily insert FactoryCast applets to view real-time PLC data on a Web page.

What Is in This Section?

This section contains the following topics:

Installing FactoryCast’s Microsoft Expression Web Extension

Overview

This section describes how to install/remove the FactoryCast extension for Microsoft Expression Web (MS Expression Web).

Installing FactoryCast

MS Expression Web installed before Web Designer for Modicon M340:

During the installation of Web Designer for Modicon M340, the FactoryCast Extension for MS Expression Web is automatically installed as a MS Expression Web "Macro File." If this is the case, then proceed to the next section, which provides instructions for adding the extension to MS Expression Web's menu.

MS Expression Web installed after Web Designer for Modicon M340:

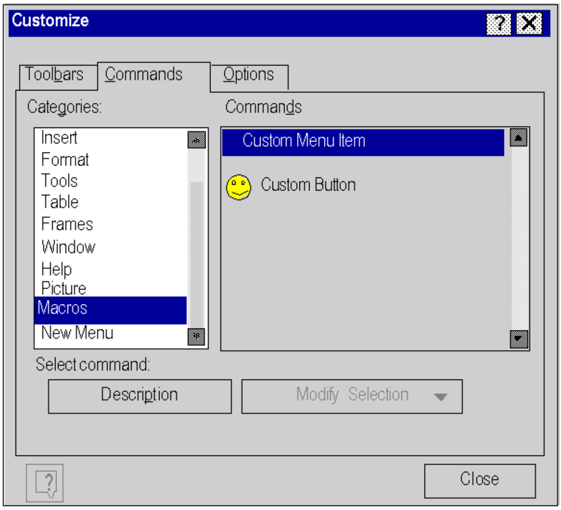
The extension can be installed by either re-installing Web Designer for Modicon M340 or manually copying the macro file to MS Expression Web's macro folder. For manual installation, after installing MS Expression Web, the following will install the extension:

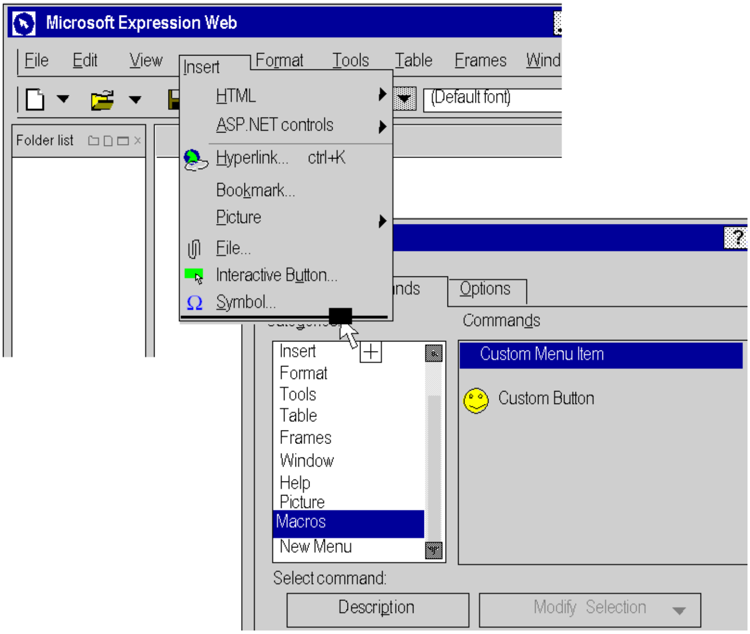
Copy the file "Microsoft Expression Web.wdmacro" from CD-ROM to the following folder (create the final 'Macros' folder, if it does not already exist):
C:\Documents and Settings\<username>\Application Data\Microsoft\Expression\Macros

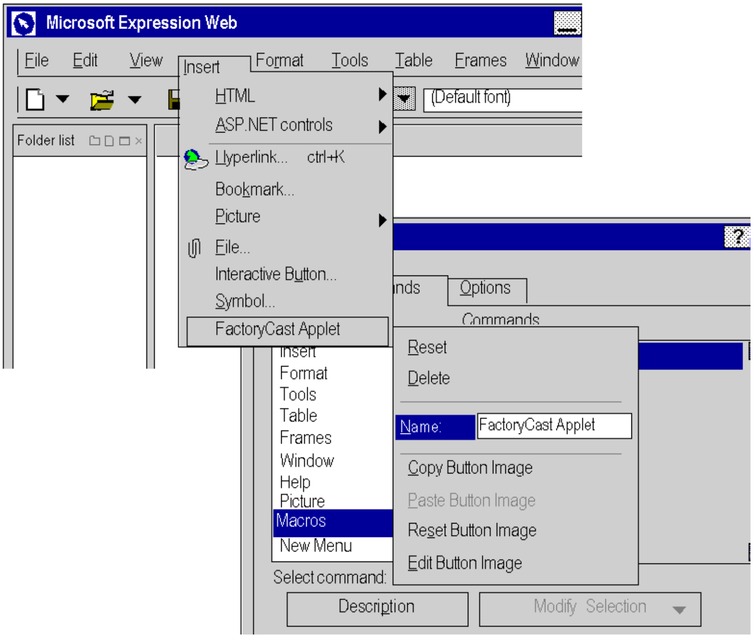
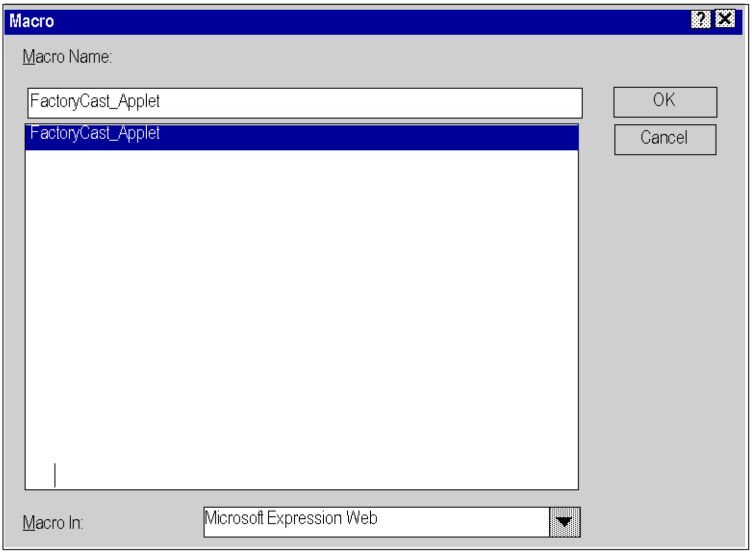
NOTE: MS Expression Web uses this address by default. For a custom installation, you can determine this address by typing SET from a Command Prompt.

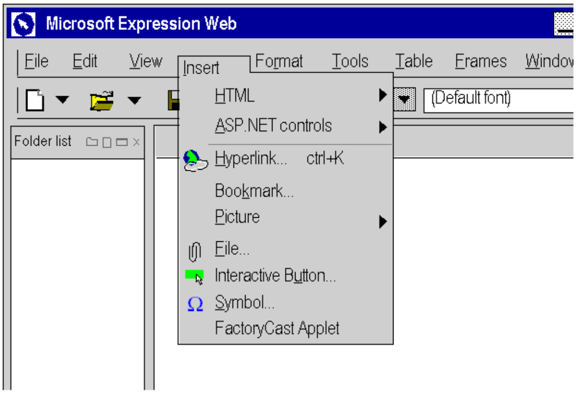
Adding FactoryCast Extension

To add the FactoryCast Extension to the MS Expression Web **Insert** menu, do the following.

Step	Action
1	Start MS Expression Web.
2	Click **Customize** on the **Tools** menu.
3	Click the **Commands** tab, and then select **Macros** from the **Categories** list.  Customize Toolbars Commands Options Categories: Insert Format Tools Table Frames Window Help Picture Macros New Menu Commands Custom Menu Item Custom Button Select command: Description Modify Selection Close

Step	Action
4	Drag the entry **Custom Menu Item** from the **Commands** list to the **Insert** menu, beneath the **Symbol** command (the **Insert** menu will automatically expand when the pointer rolls over it.). 

Step	Action
5	Click **Modify Selection**, change the name to **FactoryCast Applet**, and then press ENTER key.  Microsoft Expression Web File Edit View Insert Format Tools Table Frames Window (Default font) Folder list HTML ASP.NET controls Hyperlink... ctrl+K Bookmark... Picture File... Interactive Button... Symbol... FactoryCast Applet ?nds Options Commands Insert Format Tools Table Frames Window Help Picture Macros New Menu Select command: Reset Delete Copy Button Image Paste Button Image Reset Button Image Edit Button Image Description Modify Selection Name: FactoryCast Applet
6	Click **Modify Selection** again, and then choose **Assign Macro**.  Macro Name: FactoryCast_Applet FactoryCast_Applet Macro In: Microsoft Expression Web

Step	Action
7	Choose **FactoryCast_Applet** from the list, and then click **OK**.
8	Click **Close**. **Result:** The **FactoryCast Applet** command is added to the **Insert** menu.  Microsoft Expression Web File Edit View Insert Format Tools Table Frames Window (Default font) Folder list HTML ASP.NET controls Hyperlink... ctrl+K Bookmark... Picture File... Interactive Button... Symbol... FactoryCast Applet

Removing FactoryCast Extension

To remove the FactoryCast Extension from the MS Expression Web menu, proceed as follow:

Step	Action
1	In MS Expression Web, click **Customize** on the **Tools** menu.
2	Click the **Insert** menu, and then select **FactoryCast Applet**.
3	Right click, and then select **Delete** from the popup menu.

Editing Applets

There are two ways to edit an applet that has been inserted into your Web page:

- You can double-click on the object and make changes via dialog boxes.
- You can switch to the HTML editor in MS Expression Web and do your editing in this environment.

NOTE: It is suggested you edit via the first approach unless you are comfortable programming in the HTML language used to build Web pages.

Installing FactoryCast’s Microsoft FrontPage Extension

Overview

This section describes how to install/remove the FactoryCast Extension for Microsoft FrontPage 2000.

Installing FactoryCast

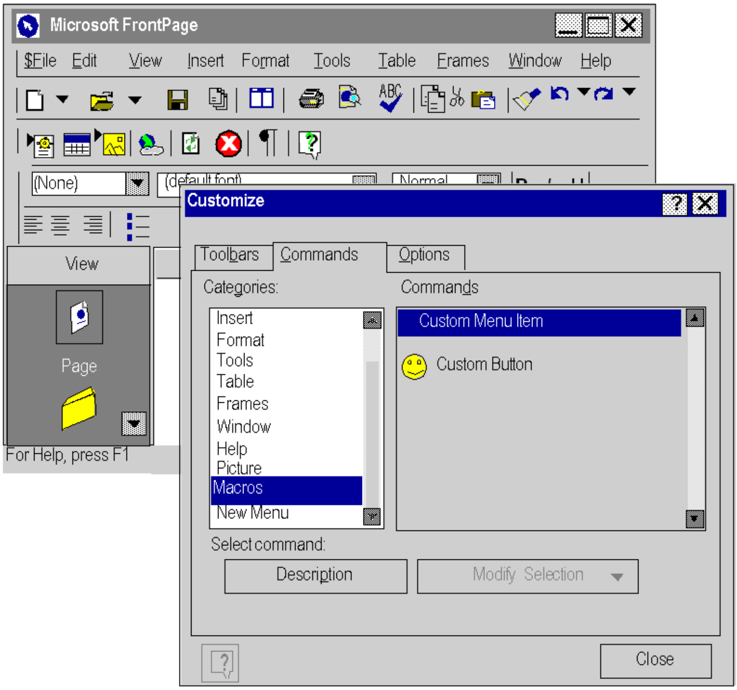
During the installation of Web Designer for Modicon M340, if FrontPage 2000 has been installed on the same PC, then the FactoryCast Extension for FrontPage 2000 is automatically installed as a FrontPage "Macro File." If this is the case, then proceed to the next section, which provides instructions for adding the Extension to FrontPage's menu.

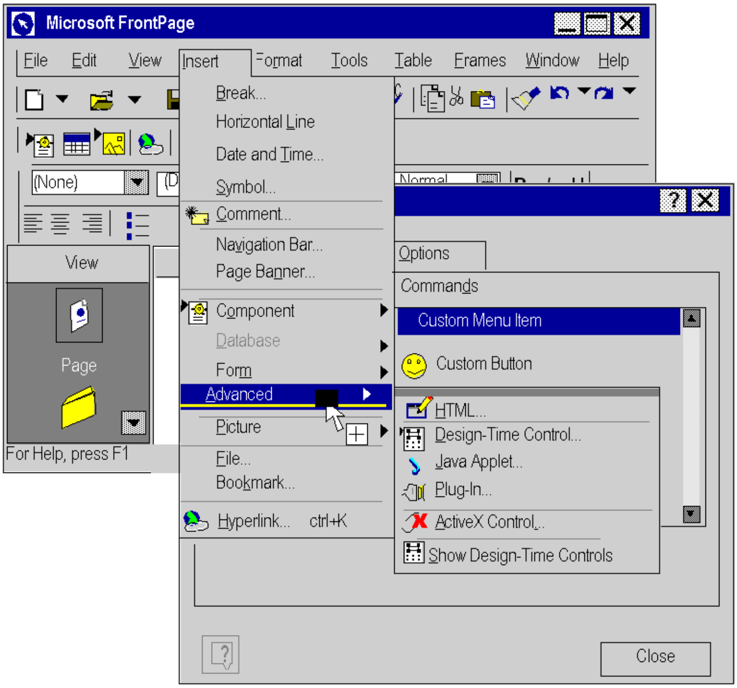
However, if FrontPage 2000 is installed after Web Designer for Modicon M340 has been installed, then the Extension can be installed by either re-installing Web Designer for Modicon M340 or manually copying the macro file to FrontPage's macro folder. For manual installation, after installing FrontPage 2000, the following will install the Extension:

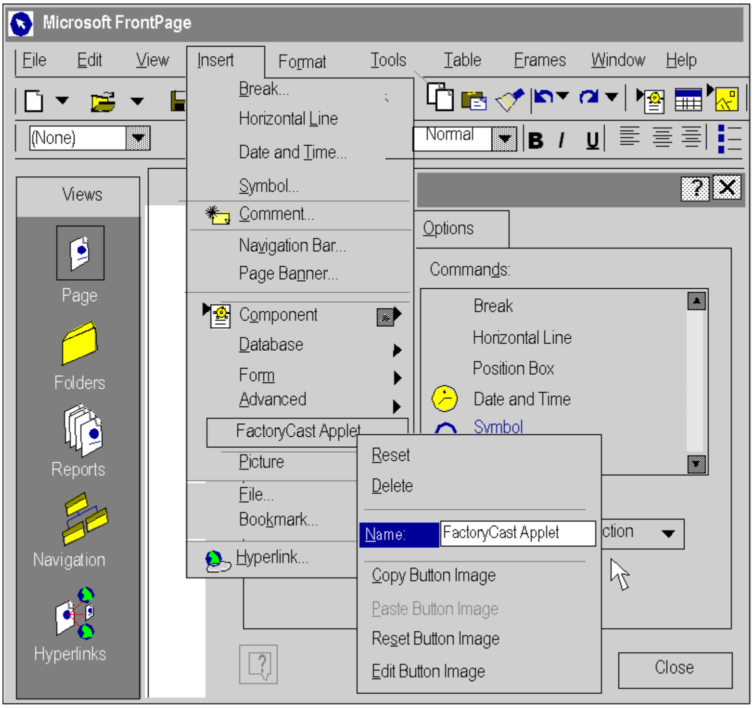
- **For Windows 95/98:** Copy the file "Microsoft FrontPage.fpm" to the folder C:\WINDOWS\Application Data\Microsoft\FrontPage\Macros. (Create the final "Macros" folder, if it does not already exist.)
- **For Windows NT/2000/XP:** Copy the file "Microsoft FrontPage.fpm" from CD-ROM to the folder %USERPROFILE%\Application Data\Microsoft\FrontPage\Macros. (Create the final 'Macros' folder, if it does not already exist.) The value of the USERPROFILE environment variable is usually C:\WINNT\Profiles\<username> for NT, or C:\Documents and Settings\<username> for Windows XP. Its value can be determined by typing SET from a Command Prompt.

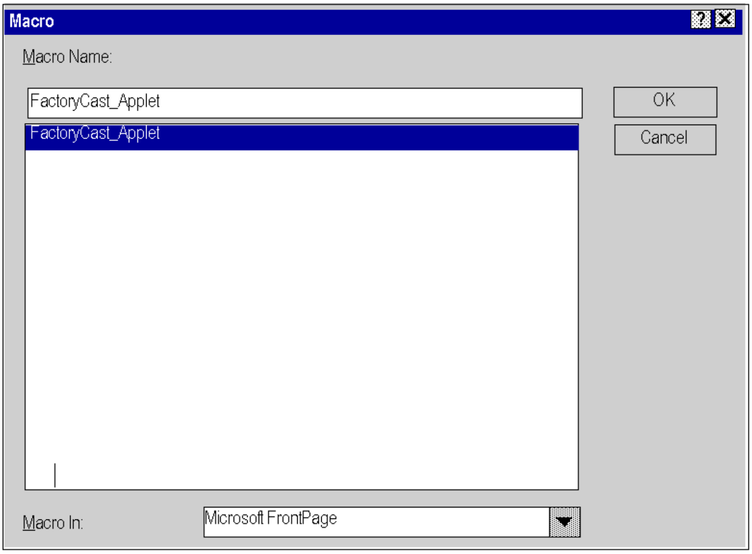
Adding FactoryCast Extension

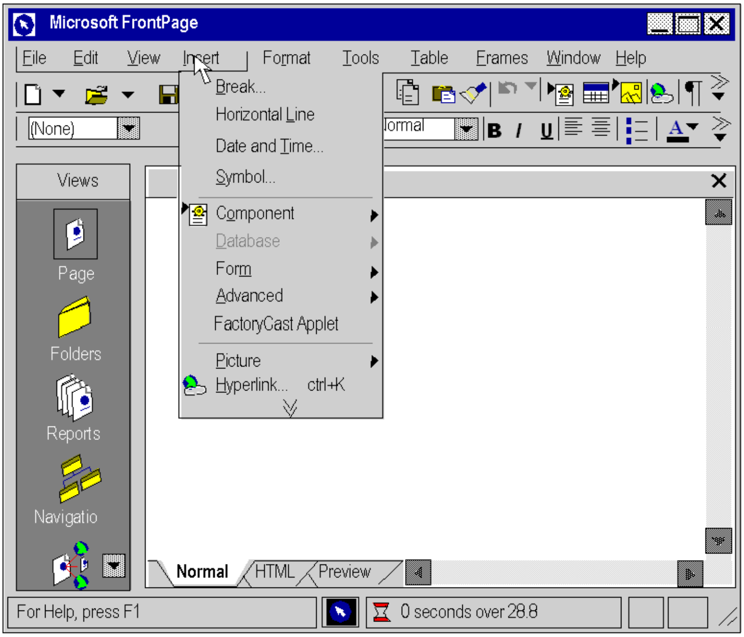
To add the FactoryCast Extension to the FrontPage **Insert** menu, do the following.

Step	Action
1	Start FrontPage 2000.
2	Click **Customize** on the **Tools** menu.
3	Click the **Commands** tab, and then select **Macros** from the **Categories** list.  Microsoft FrontPage $File Edit View Insert Format Tools Table Frames Window Help (None) (default font) Normal View Page For Help, press F1 Customize Toolbars Commands Options Categories: Insert Format Tools Table Frames Window Help Picture Macros New Menu Commands Custom Menu Item Custom Button Select command: Description Modify Selection Close

Step	Action
4	Drag and drop the entry **Custom Menu Item** from the **Commands** list to the **Insert** menu, beneath the **Advanced** command. (The **Insert** menu will automatically drop down when you drag over it.)  Microsoft FrontPage File Edit View Insert Format Tools Table Frames Window Help Break... Horizontal Line Date and Time... Symbol... Comment... Navigation Bar... Page Banner... Component Database Form Advanced Picture File... Bookmark... Hyperlink... ctrl+K (None) Normal View Page For Help, press F1 Options Commands Custom Menu Item Custom Button HTML... Design-Time Control... Java Applet... Plug-In... ActiveX Control... Show Design-Time Controls Close

Step	Action
5	Click **Modify Selection**, change the name to **FactoryCast Applet**, and then press ENTER key.  Name: FactoryCast Applet

Step	Action
6	Click **Modify Selection** again, and then choose **Assign Macro**.  Macro Macro Name: FactoryCast_Applet FactoryCast_Applet Macro In: Microsoft FrontPage
7	Choose **FactoryCast_Applet** from the list, and then click **OK**.

Step	Action
8	Click **Close**. The **FactoryCast Applet** command is added to the **Insert** menu.  Microsoft FrontPage File Edit View Insert Format Tools Table Frames Window Help Break... Horizontal Line Date and Time... Symbol... Component Database Form Advanced FactoryCast Applet Picture Hyperlink... ctrl+K (None) lormal Views Page Folders Reports Navigatio Normal HTML Preview For Help, press F1 0 seconds over 28.8

Removing FactoryCast Extension

To remove the FactoryCast Extension from the FrontPage menu, do the following.

Step	Action
1	In FrontPage, click **Customize** on the **Tools** menu.
2	Click the **Insert** menu, and then select **FactoryCast Applet**.
3	Right click, and then select **Delete** from the popup menu.

Editing Applets

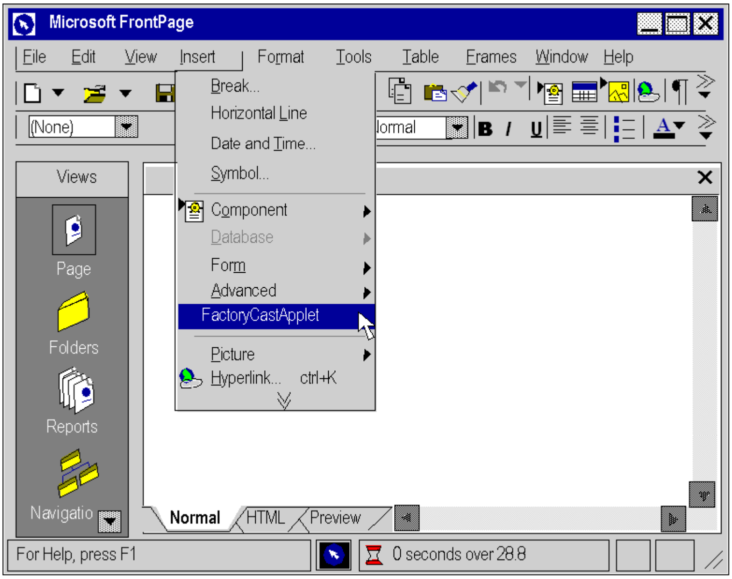
There are two ways to edit an applet that has been inserted into your Web page. First, you can double-click on the object and make changes via dialog boxes. Or you can switch to the HTML editor in FrontPage and do your editing in this environment. It is suggested you edit via the first approach unless you are comfortable programming in the HTML language used to build Web pages.

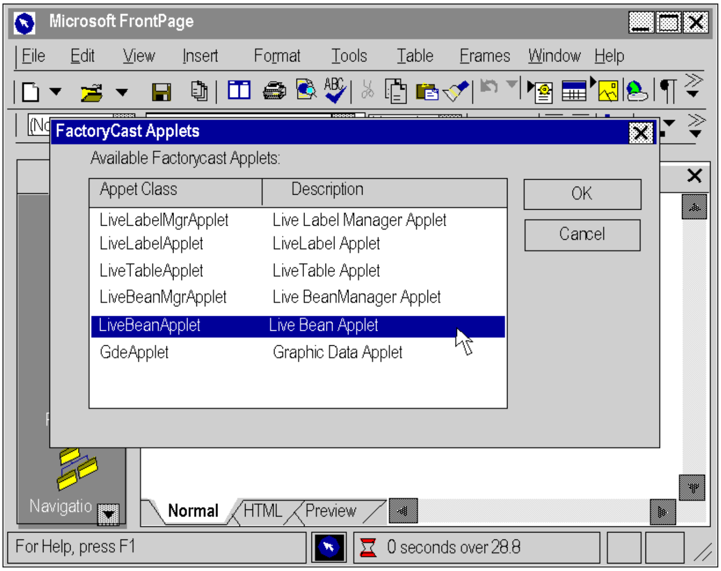
Inserting LiveBeanApplet Using FrontPage or Expression Web

Overview

NOTE: The following procedure shows you how inserting **LiveBeanApplet** using Frontpage. Use the same procedure if you use Expression Web.

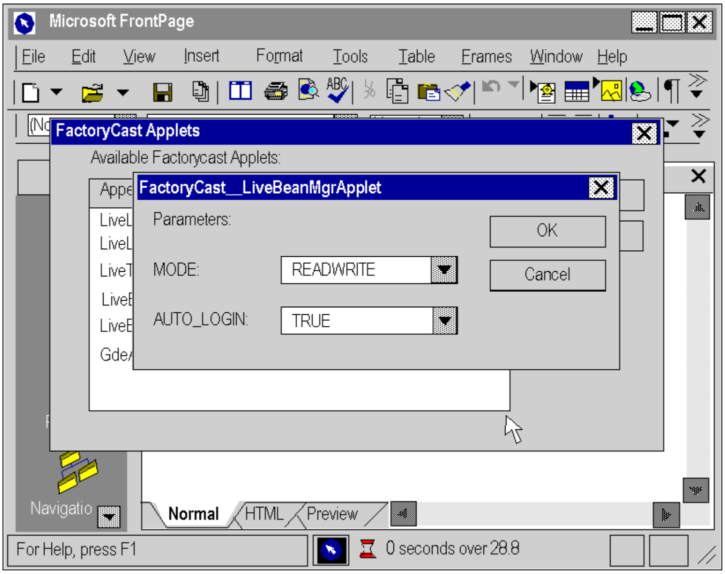
Inserting a LiveBeanApplet

Step	Action
1	In FrontPage, Select **Insert | FactoryCast Applet** to view the FactoryCast applet selection window.  Microsoft FrontPage File Edit View Insert Format Tools Table Frames Window Help Break... Horizontal Line Date and Time... Symbol... Component Database Form Advanced FactoryCastApplet Picture Hyperlink... ctrl+K (None) Normal Views Page Folders Reports Navigatio Normal HTML Preview For Help, press F1 0 seconds over 28.8

Step	Action
2	**Note:** The **LiveBeanMgrApplet** is a special applet that is inserted one time only into your Web page. Select the applet **LiveBeanMgrApplet** and then click the OK button. 

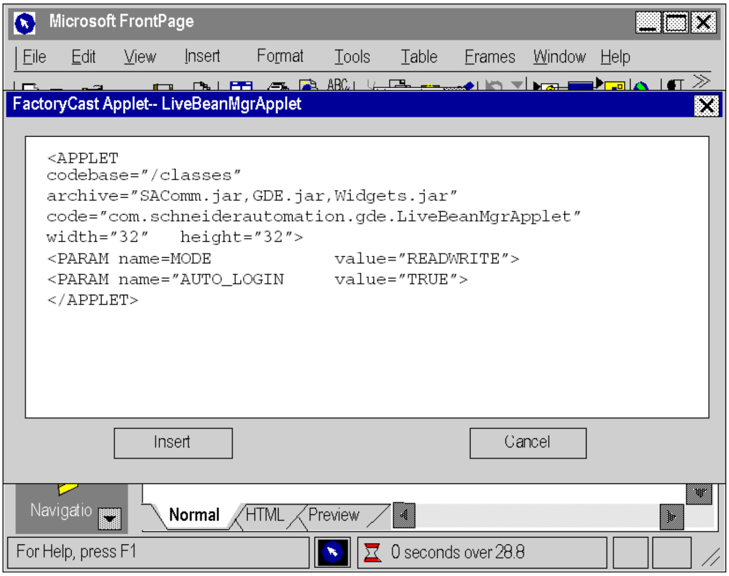
Available Factorycast Applets:

Appet Class	Description
LiveLabelMgrApplet	Live Label Manager Applet
LiveLabelApplet	LiveLabel Applet
LiveTableApplet	LiveTable Applet
LiveBeanMgrApplet	Live BeanManager Applet
LiveBeanApplet	Live Bean Applet
GdeApplet	Graphic Data Applet

Step	Action
3	Configure the **Mode** and **Auto Login** parameters and then click the **OK** button.  FactoryCast__LiveBeanMgrApplet Parameters: MODE: READWRITE AUTO_LOGIN: TRUE Microsoft FrontPage File Edit View Insert Format Tools Table Frames Window Help FactoryCast Applets Available Factorycast Applets: OK Cancel Navigatio Normal HTML Preview For Help, press F1 0 seconds over 28.8

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>

A window will appear showing the Java code that will be inserted into your HTML document.



Microsoft FrontPage
File Edit View Insert Format Tools Table Frames Window Help
FactoryCast Applet-- LiveBeanMgrApplet
Insert
Cancel
Navigatio
Normal
HTML
Preview
For Help, press F1
0 seconds over 28.8

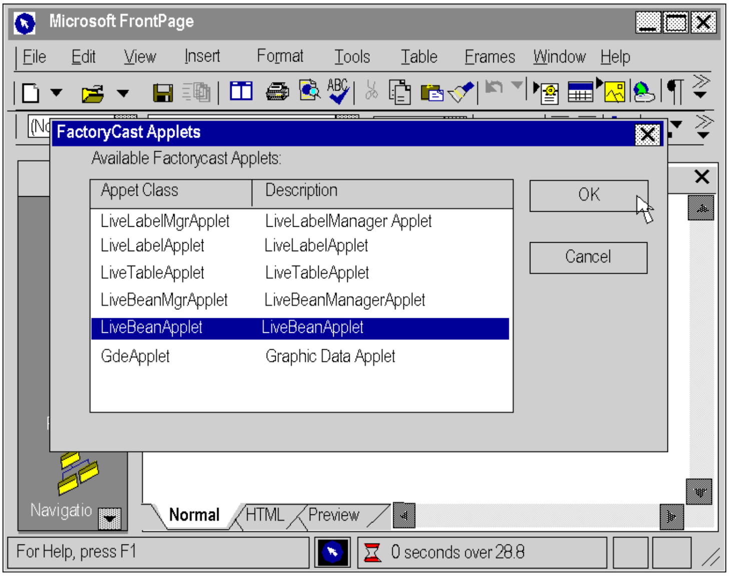
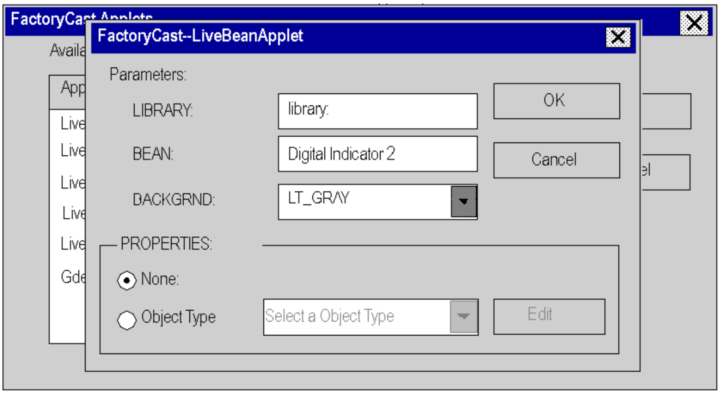
```
<APPLET
codebase="/classes"
archive="SAComm.jar,GDE.jar,Widgets.jar"
code="com.schneiderautomation.gde.LiveBeanMgrApplet"
width="32"   height="32">
<PARAM name=MODE            value="READWRITE">
<PARAM name="AUTO_LOGIN     value="TRUE">
</APPLET>
```

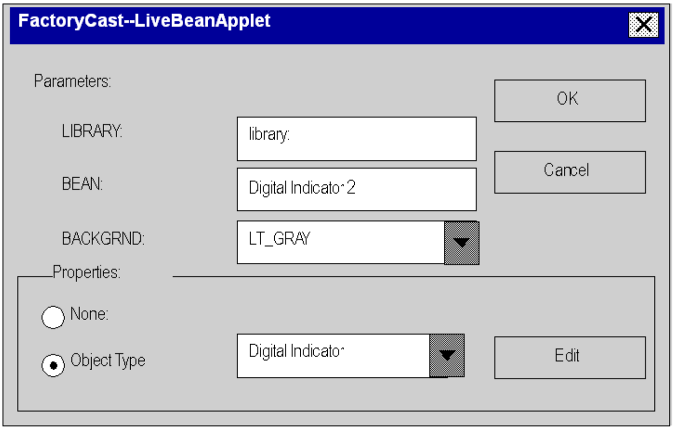
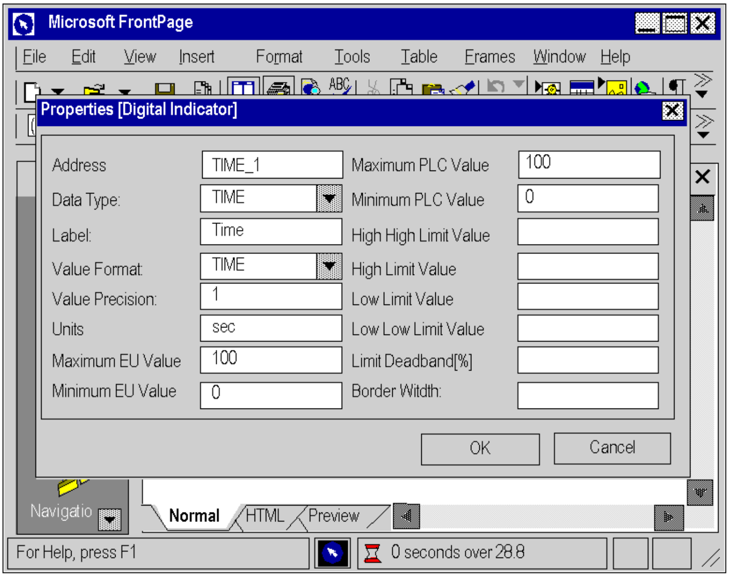
Do the following:

1. Add the following line to the applet:
   `<PARAM Name=FTP_ROOT value="/SDCA/Web">`
2. Click the **Insert** button to finish inserting this applet.

Note: Though a gray box with a blue letter "J" will appear on your Web page during editing, this applet will only be visible on your loaded Web page if you configured the Mode parameter as **ReadWrite** (it will appear as a Key when viewed via your browser). Otherwise, if **Mode** is set as **ReadOnly** then the width and height of the applet will be set to zero and you will not see it in your browser.

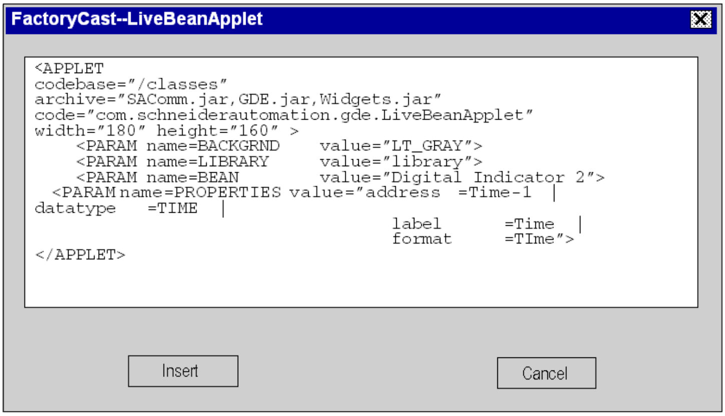
</td></tr>
</table>

Step	Action
5	Select the **LiveBeanApplet** from the FactoryCast applet selection window and then click the **OK** button.  Microsoft FrontPage File Edit View Insert Format Tools Table Frames Window Help Navigatio Normal HTML Preview For Help, press F1 0 seconds over 28.8 FactoryCast Applets Available Factorycast Applets: Appet Class — Description LiveLabelMgrApplet — LiveLabelManager Applet LiveLabelApplet — LiveLabelApplet LiveTableApplet — LiveTableApplet LiveBeanMgrApplet — LiveBeanManagerApplet LiveBeanApplet — LiveBeanApplet GdeApplet — Graphic Data Applet OK Cancel
6	Enter the library name and the name of a particular bean in this library that you want to show. If none of the bean’s properties, such as address, need to be customized, then click the **OK** button.  FactoryCast Applets Availa App Live Live Live Live Live Gde el FactoryCast--LiveBeanApplet Parameters: LIBRARY: library BEAN: Digital Indicator 2 BACKGRND: LT_GRAY PROPERTIES: ● None: ○ Object Type Select a Object Type OK Cancel Edit

Step	Action
7	Usually at a minimum the address property will need to be customized for your beans. Thus after entering the library name and bean name, select the object type from the **Object Type** box.  FactoryCast--LiveBeanApplet Parameters: LIBRARY: library BEAN: Digital Indicator2 BACKGRND: LT_GRAY Properties: ○ None: ● Object Type: Digital Indicator OK Cancel Edit Click the **Edit** button when finished.
8	In the properties edit window, change only those parameters, such as address, that are unique to this bean. Other parameters are set according to the saved bean in your library. Click the **OK** button when finished.  Properties [Digital Indicator] Address: TIME_1 Data Type: TIME Label: Time Value Format: TIME Value Precision: 1 Units: sec Maximum EU Value: 100 Minimum EU Value: 0 Maximum PLC Value: 100 Minimum PLC Value: 0 High High Limit Value: High Limit Value: Low Limit Value: Low Low Limit Value: Limit Deadband[%]: Border Width: OK Cancel

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>9</td><td>

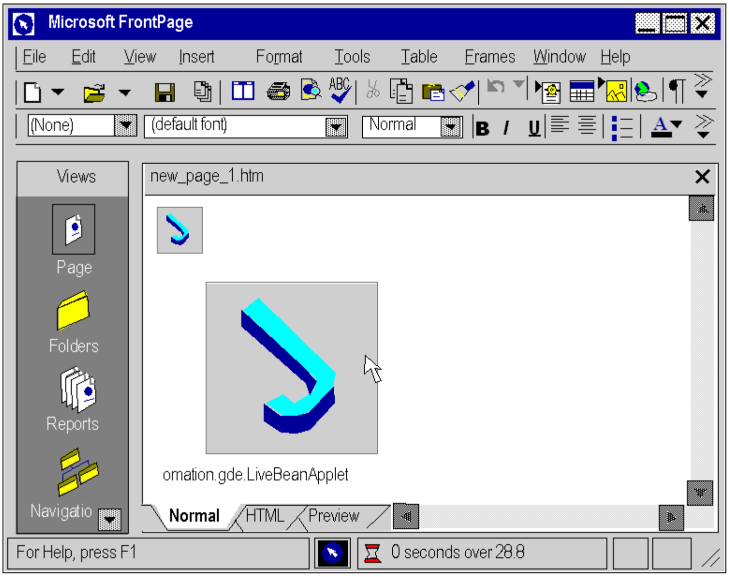
Click the **OK** button. A window will appear showing the Java code that will be inserted into your HTML document. Select the **Insert** button to finish inserting this bean.



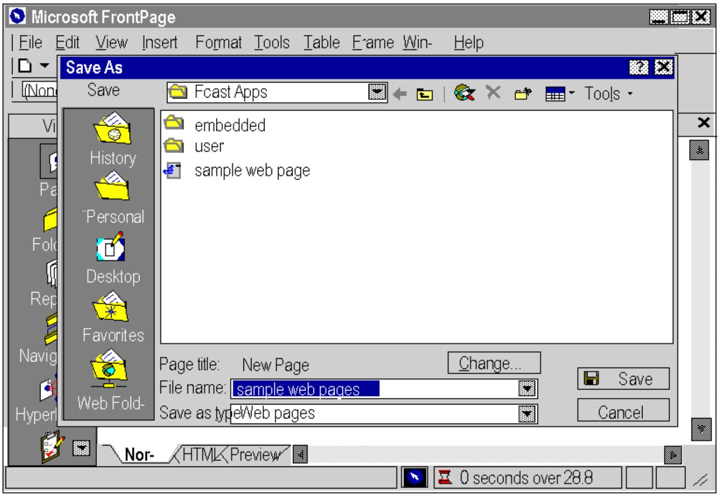
```
<APPLET
codebase="/classes"
archive="SAComm.jar,GDE.jar,Widgets.jar"
code="com.schneiderautomation.gde.LiveBeanApplet"
width="180" height="160" >
    <PARAM name=BACKGRND     value="LT_GRAY">
    <PARAM name=LIBRARY      value="library">
    <PARAM name=BEAN         value="Digital Indicator 2">
  <PARAM name=PROPERTIES value="address  =Time-1  |
datatype   =TIME  |
                                   label      =Time  |
                                   format     =TIme">
</APPLET>
```

</td></tr>
<tr><td>10</td><td>

Continue to add instances of `LiveBeanApplet` to your Web page. Once you have inserted your last applet, click the **Cancel** button on the FactoryCast applet selection window to return to editing your Web page.



</td></tr>
</table>

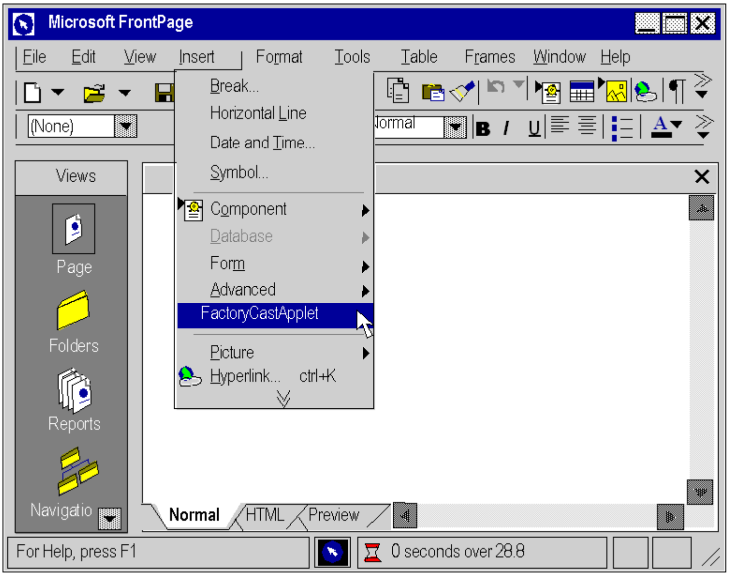
Step	Action
11	Save the customized Web page.  Fcast Apps embedded user sample web page Page title: New Page File name: sample web pages Save as type: Web pages Microsoft FrontPage File Edit View Insert Format Tools Table Frame Win- Help Save As Save Tools History Personal Desktop Favorites Web Fold- Change... Save Cancel Nor- HTML Preview 0 seconds over 28.8
12	Transfer the customized Web page to the NOE Web server.
13	Test the application.

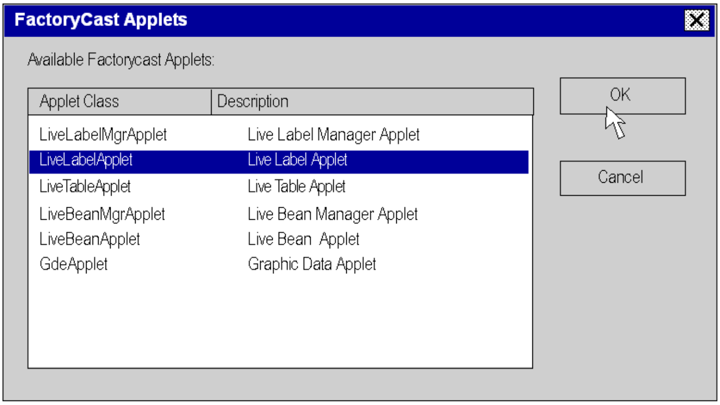
Inserting LiveLabelApplet Using FrontPage and Expression Web

Overview

NOTE: The following procedure shows you how inserting **LiveLabelApplet** using Frontpage. Use the same procedure if you use Expression Web.

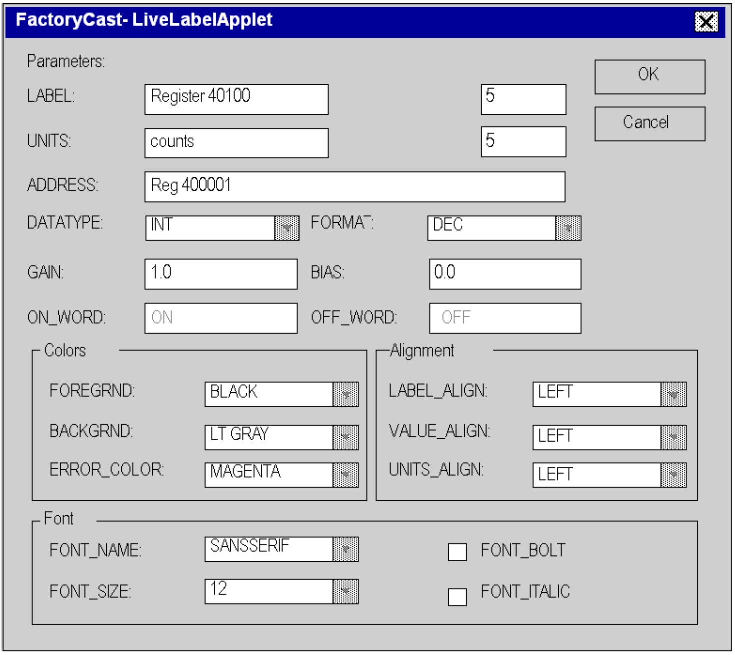
Inserting LiveLabelApplet

Step	Action
1	Select **Insert | FactoryCast** to view the FactoryCast applet selection window.  Microsoft FrontPage File Edit View Insert Format Tools Table Frames Window Help (None) Normal Break... Horizontal Line Date and Time... Symbol... Component Database Form Advanced FactoryCastApplet Picture Hyperlink... ctrl+K Views Page Folders Reports Navigatio Normal HTML Preview For Help, press F1 0 seconds over 28.8

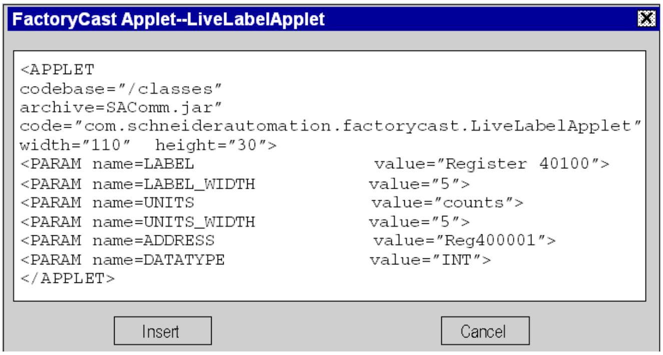
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>Select the applet <b>LiveLabelMgrApplet</b> and then click the <b>OK</b> button.


<table>
<tr><th>Applet Class</th><th>Description</th></tr>
<tr><td>LiveLabelMgrApplet</td><td>Live Label Manager Applet</td></tr>
<tr><td>LiveLabelApplet</td><td>Live Label Applet</td></tr>
<tr><td>LiveTableApplet</td><td>Live Table Applet</td></tr>
<tr><td>LiveBeanMgrApplet</td><td>Live Bean Manager Applet</td></tr>
<tr><td>LiveBeanApplet</td><td>Live Bean Applet</td></tr>
<tr><td>GdeApplet</td><td>Graphic Data Applet</td></tr>
</table>

A window will appear showing the HTML code that will be inserted into your HTML document. Do the following:
1. Add the following line of code to the applet:
<code><PARAM Name=FTP_ROOT value="/SDCA/Web"></code>
2. Click the <b>Insert</b> button to finish inserting this applet.
<b>Note:</b> The <b>LiveLabelMgrApplet</b> is a special applet that is inserted one time only into your Web page. Though a gray box with a blue letter "J" will appear on your Web page during editing, this applet will not be visible when you view your Web page from the Web server module.</td></tr>
<tr><td>3</td><td>Select the applet <b>LiveLabelApplet</b> and then click the <b>OK</b> button. The Parameters edit window appears.</td></tr>
</table>

Step	Action
4	Fill in the parameters for the label and then click the **OK** button.  FactoryCast- LiveLabelApplet Parameters: LABEL: Register 40100 5 UNITS: counts 5 ADDRESS: Reg 400001 DATATYPE: INT FORMAT: DEC GAIN: 1.0 BIAS: 0.0 ON_WORD: ON OFF_WORD: OFF Colors FOREGRND: BLACK BACKGRND: LT GRAY ERROR_COLOR: MAGENTA Alignment LABEL_ALIGN: LEFT VALUE_ALIGN: LEFT UNITS_ALIGN: LEFT Font FONT_NAME: SANSSERIF FONT_SIZE: 12 □ FONT_BOLT □ FONT_ITALIC **Note:** For a description on each parameter, see *Using FactoryCast's Java Applets Using HTML Code, page 146*.

Step	Action

5	A window will appear showing the HTML code that will be inserted into your HTML document. Click the **Insert** button to finish inserting this applet.
	
	
	`<APPLET`
	`codebase="/classes"`
	`archive=SAComm.jar"`
	`code="com.schneiderautomation.factorycast.LiveLabelApplet"`
	`width="110"   height="30">`
	`<PARAM name=LABEL               value="Register 40100">`
	`<PARAM name=LABEL_WIDTH         value="5">`
	`<PARAM name=UNITS               value="counts">`
	`<PARAM name=UNITS_WIDTH         value="5">`
	`<PARAM name=ADDRESS             value="Reg400001">`
	`<PARAM name=DATATYPE            value="INT">`
	`</APPLET>`

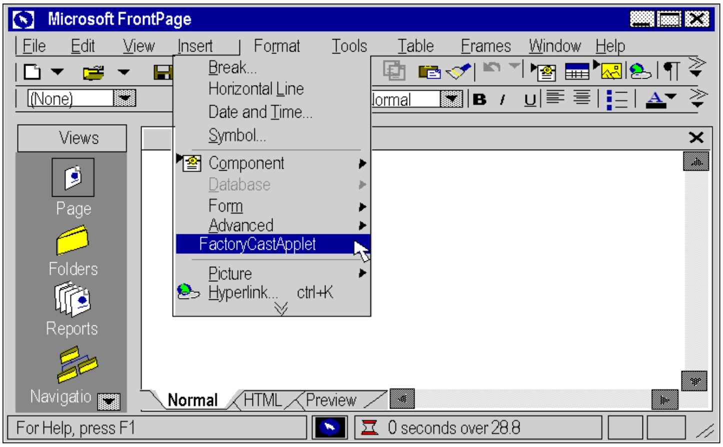
6	Continue to add additional instances of `LiveLabelApplet` to your Web page. Once you have inserted your last applet, click the **Cancel** button on the FactoryCast applet selection window to return to editing your Web page.

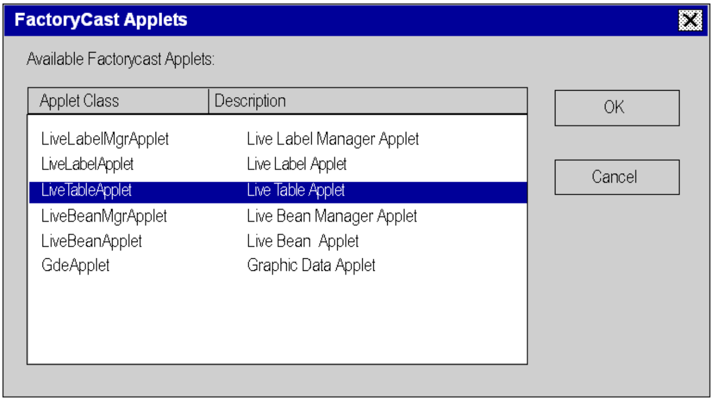
Inserting LiveTableApplet Using FrontPage or Expression Web

Overview

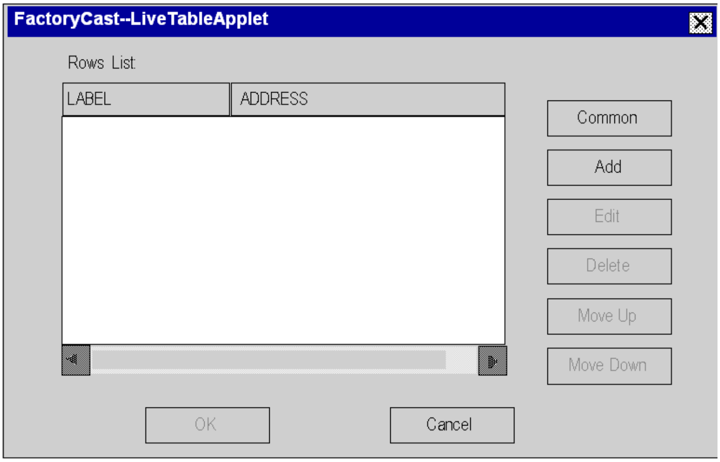
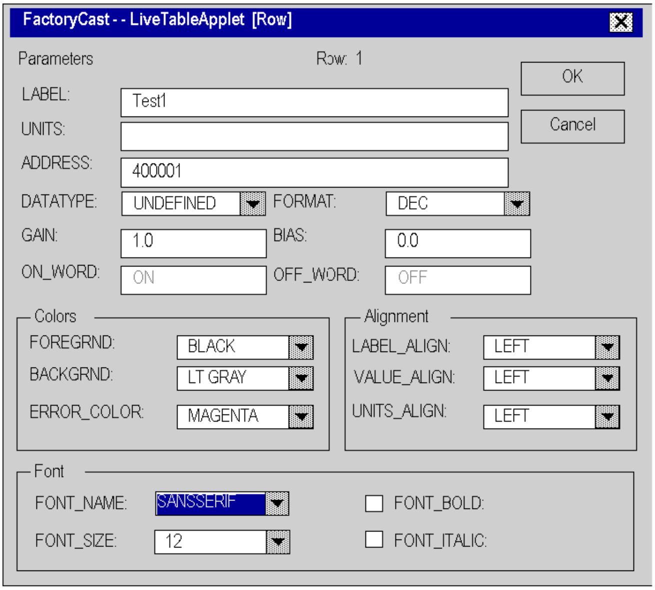
NOTE: The following procedure shows you how inserting **LiveTableApplet** using Frontpage. Use the same procedure if you use Expression Web.

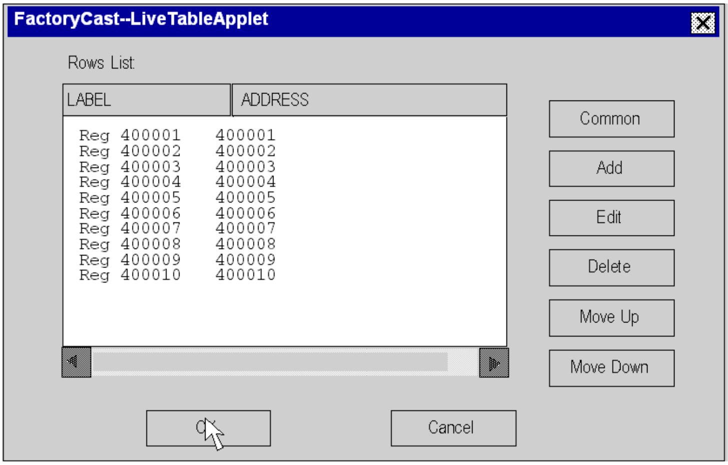
Inserting LiveTableApplet

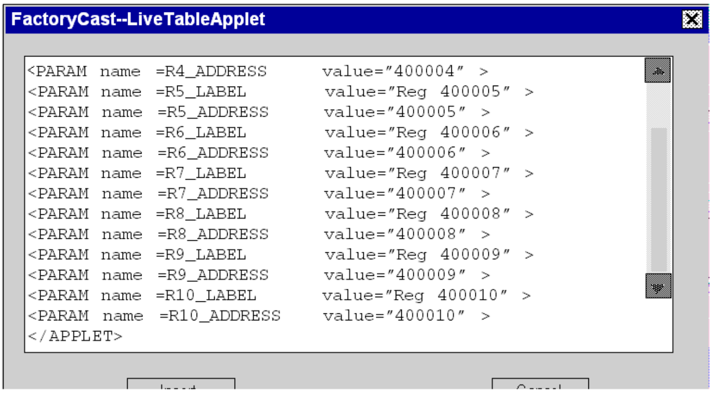
Step	Action
1	Select **Insert | FactoryCast** to view the FactoryCast applet selection window.  Microsoft FrontPage File Edit View Insert Format Tools Table Frames Window Help (None) Normal Break... Horizontal Line Date and Time... Symbol... Component Database Form Advanced FactoryCastApplet Picture Hyperlink... ctrl+K Views Page Folders Reports Navigatio Normal HTML Preview For Help, press F1 0 seconds over 28.8
2	Select the applet **LiveLabelMgrApplet** and then click the **OK** button. A window will appear showing the HTML code that will be inserted into your HTML document. Do the following: 1. Add the following line of code to the applet: `<PARAM Name=FTP_ROOT value="/SDCA/Web">` 2. Click the **Insert** button to finish inserting this applet. **Note:** The **LiveLabelMgrApplet** is a special applet that is inserted one time only into your Web page.

Step	Action
3	Select the **LiveTableApplet** and then click the **OK** button. 

Applet Class	Description
LiveLabelMgrApplet	Live Label Manager Applet
LiveLabelApplet	Live Label Applet
LiveTableApplet	Live Table Applet
LiveBeanMgrApplet	Live Bean Manager Applet
LiveBeanApplet	Live Bean Applet
GdeApplet	Graphic Data Applet

Step	Action
4	Click the **Common** button to configure the parameters that are common to all rows of the table.  FactoryCast--LiveTableApplet Rows List LABEL | ADDRESS Common Add Edit Delete Move Up Move Down OK Cancel This procedure speeds configuration by simultaneously entering many values that have the same parameter fields. You many still edit parameters for rows individually during the add process.  FactoryCast - - LiveTableApplet [Row] Parameters Row: 1 LABEL: Test1 UNITS: ADDRESS: 400001 DATATYPE: UNDEFINED FORMAT: DEC GAIN: 1.0 BIAS: 0.0 ON_WORD: ON OFF_WORD: OFF Colors FOREGRND: BLACK BACKGRND: LT GRAY ERROR_COLOR: MAGENTA Alignment LABEL_ALIGN: LEFT VALUE_ALIGN: LEFT UNITS_ALIGN: LEFT Font FONT_NAME: SANSSERIF FONT_SIZE: 12 ☐ FONT_BOLD: ☐ FONT_ITALIC:

Step	Action
5	Click the **Add** button to add a row to your table. The parameters edit dialog window appears. Configure parameters and then click the **Add** button when you are done.  FactoryCast--LiveTableApplet Rows List LABEL ADDRESS Reg 400001 400001 Reg 400002 400002 Reg 400003 400003 Reg 400004 400004 Reg 400005 400005 Reg 400006 400006 Reg 400007 400007 Reg 400008 400008 Reg 400009 400009 Reg 400010 400010 Common Add Edit Delete Move Up Move Down OK Cancel
6	Continue to add rows using the **Add** button. You may also select a row and move it up or down the rows of your table using the **Move Up** and **Move Down** buttons. When you are finished adding and editing rows in your table, click the **OK** button from the **Rows List** window.

7	A window will appear showing the HTML code that will be inserted into your HTML document. Click the **Insert** button to finish inserting this applet.
	
	FactoryCast--LiveTableApplet
	`<PARAM name =R4_ADDRESS value="400004" >`
	`<PARAM name =R5_LABEL value="Reg 400005" >`
	`<PARAM name =R5_ADDRESS value="400005" >`
	`<PARAM name =R6_LABEL value="Reg 400006" >`
	`<PARAM name =R6_ADDRESS value="400006" >`
	`<PARAM name =R7_LABEL value="Reg 400007" >`
	`<PARAM name =R7_ADDRESS value="400007" >`
	`<PARAM name =R8_LABEL value="Reg 400008" >`
	`<PARAM name =R8_ADDRESS value="400008" >`
	`<PARAM name =R9_LABEL value="Reg 400009" >`
	`<PARAM name =R9_ADDRESS value="400009" >`
	`<PARAM name =R10_LABEL value="Reg 400010" >`
	`<PARAM name =R10_ADDRESS value="400010" >`
	`</APPLET>`

Refer to Graphic Editor Applet Parameters *(see page 106)* for inserting a custom GDEApplet applet.

Section 5.3
Using Graphic Objects Lite

Overview

This section describes how to use the Graphic Objects Lite library. This library uses a modem and allows faster download of the graphical interface by the user. This is a lighter version of the standard graphical library.

What Is in This Section?

This section contains the following topics:

Downloading the Graphic Objects Lite Library

Overview

Download the Graphic Objects Lite library before you use it. Use the Transfer Web File utility provided with Web Designer, and transfer the file **widgetslite.jar** to the module’s Web site.

NOTE: The **widgetslite.jar** file is located in the installation folder, in the subdirectory **/addons/jar**.

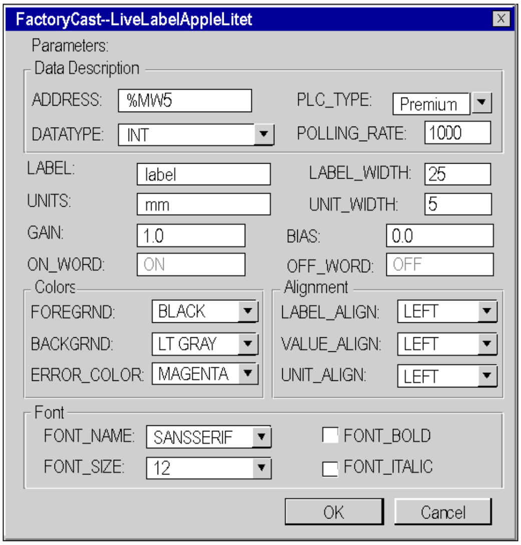
Description of Graphic Objects Lite

Presentation

The set of graphic objects provided in the FactoryCast_Applet can help you create graphic displays similar to the human-machine interface (HMI) screens. The data control and monitoring objects have integrated communication functions and are designed as standalone graphic objects.

LiveLabel AppletLite Setup

The **LiveLabel AppletLite** window displays the direct address value of a Modbus slave in a text field.



FactoryCast--LiveLabelAppleLitet

Parameters:

Data Description

ADDRESS: %MW5

PLC_TYPE: Premium

DATATYPE: INT

POLLING_RATE: 1000

LABEL: label

LABEL_WIDTH: 25

UNITS: mm

UNIT_WIDTH: 5

GAIN: 1.0

BIAS: 0.0

ON_WORD: ON

OFF_WORD: OFF

Colors

FOREGRND: BLACK

BACKGRND: LT GRAY

ERROR_COLOR: MAGENTA

Alignment

LABEL_ALIGN: LEFT

VALUE_ALIGN: LEFT

UNIT_ALIGN: LEFT

Font

FONT_NAME: SANSSERIF

FONT_SIZE: 12

□ FONT_BOLD

□ FONT_ITALIC

The properties of this widget are as follows:

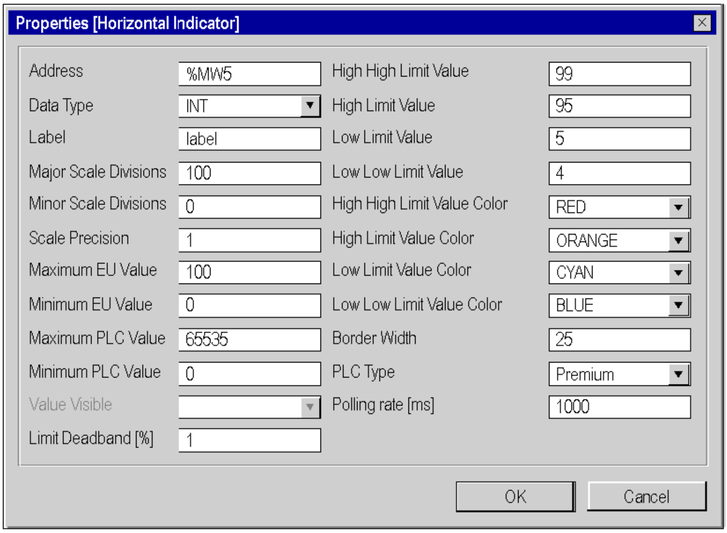
Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 207)*
Data type	PLC address data type	Note 2 *(see page 207)*
PLC Type	Type of PLC	Premium or Quantum
Label	Label displayed as part of the graphic object	Note 5 *(see page 207)*
Label Width	Width of label	
Unit Width	Width of unit	
Polling rate [ms]	Scanning value	
Gain	The gain (multiplier) is used for the scale of the value retrieved in physical units	1.0
Bias	The bias (multiplier) is used for the scale of the value retrieved in physical units	0.0
ON_Word	Text value displayed when the value is not zero (use if the data type is binary)	ON
OFF_Word	Text value displayed when the value is not zero (use if the data type is binary)	OFF
Foregrnd	Color of the applet foreground	BLACK
Backgrnd	Color of the applet background	LT_GRAY
Error_Color	Color of the applet foreground if the address value cannot be retrieved	MAGENTA
Label_Align	Alignment of text in the Label field if the size is greater than the text length	LEFT
Value_Align	Alignment of text in the Value field if the size is greater than the text length	LEFT
Units_Align	Alignment of text in the Units field if the size is greater than the text length	LEFT
Font_Name	Font name for the applet text	SANSSERIF
Font_Bold	Applet text is bold if configured as TRUE	FALSE
Font_Italic	Applet text is italic if configured as TRUE	FALSE
Font_Size	Applet text size	12

The HTML code with the parameters of the widget above is as follows:

```
<APPLET
codebase="/classes"
archive="mbclient.jar,widgetslite.jar"
code="com.schneiderautomation.factorycast.gateway.widgetsl
ite.LiveLabelApplet"
width="130" height="30">
   <PARAM name="progressbar"    value="true">
   <PARAM name="progresscolor"    value="#000000">
   <PARAM name="ADDRESS"    value="1">
   <PARAM name="UNITID"    value="0">
   <PARAM name="RATE"    value="1000">
   <PARAM name="DATATYPE"    value="REGISTER">
   <PARAM name="LABEL"    value="label">
   <PARAM name="LABEL_WIDTH"    value="25">
   <PARAM name="UNITS"    value="mm">
   <PARAM name="UNITS_WIDTH"    value="5">
</APPLET>
```

Horizontal or Vertical Indicator Setup

The horizontal or vertical indicator gives an analog representation of the value of a direct address of a Modbus slave. It is a horizontal or vertical bar whose length is proportional to the value. It represents a percentage of its range in physical units.



Properties [Horizontal Indicator]

Address	%MW5	High High Limit Value	99
Data Type	INT	High Limit Value	95
Label	label	Low Limit Value	5
Major Scale Divisions	100	Low Low Limit Value	4
Minor Scale Divisions	0	High High Limit Value Color	RED
Scale Precision	1	High Limit Value Color	ORANGE
Maximum EU Value	100	Low Limit Value Color	CYAN
Minimum EU Value	0	Low Low Limit Value Color	BLUE
Maximum PLC Value	65535	Border Width	25
Minimum PLC Value	0	PLC Type	Premium
Value Visible		Polling rate [ms]	1000
Limit Deadband [%]	1		

The properties of the indicator are as follows:

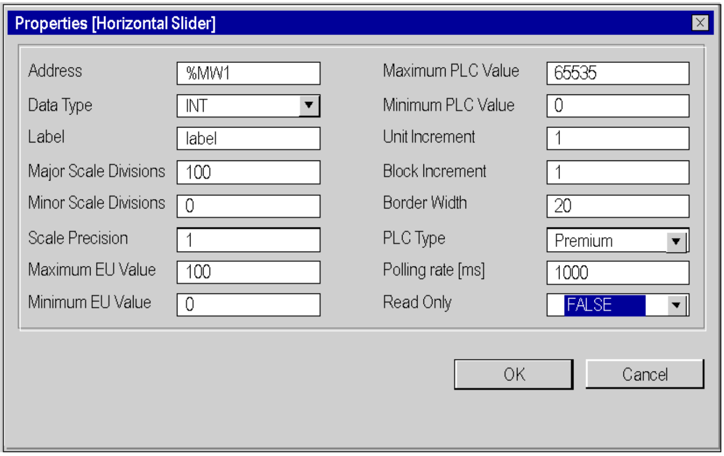
Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 207)*
Data type	PLC address data type	Note 2 *(see page 207)*
Label	Label displayed as part of the graphic object	Note 5 *(see page 207)*
Major scale gradation	Number of major gradations (marked) in the scale	0 to 100
Minor scale gradation	Number of minor gradations (not marked) in the scale	0 to 100
Scale precision	Number of decimal places shown for the scale gradations (set to -1 to use a general exponential format)	-1 to 6
Maximum EU Value	Maximum value of the direct address for scaling, in physical units	
Minimum EU Value	Minimum value, in physical units, of the direct address for scaling	
Maximum PLC Value	Gross maximum value (without scale) of the direct address in the PLC	Note 3 *(see page 207)*
Minimum PLC Value	Gross minimum value (without scale) of the direct address in the PLC	Note 3 *(see page 207)*
Limit Deadband[%]	Neutral range (as a percentage of the UP range) to apply to verification of the High/Low limit	0 to 10
High High Limit Value	Value expressed in physical units of the "High High" limit	
High Limit Value	Value expressed in physical units of the "High" limit	
Low Limit Value	Value expressed in physical units of the "Low" limit	
Low Low Limit Value	Value expressed in physical units of the "Low Low" limit	
High High Limit Value Color	Color of the indicator bar if the scale value is greater than the "High High" limit	
High Limit Value Color	Color of the indicator bar if the scale value is greater than the "High" limit	
Low Limit Value Color	Color of the indicator bar if the scale value is less than the "Low" limit	
Low Low Limit Value Color	Color of the indicator bar if the scale value is less than the "Low Low" limit	
Border Width	Width of border	
PLC Type	Type of PLC	Quantum or Premium
Polling rate [ms]	Scanning value	

The HTML code with the parameters of the widget above is as follows:

```
<APPLET
codebase="/classes"
archive="mbclient.jar,widgetslite.jar"
code="com.schneiderautomation.factorycast.gateway.
indicators.LiveHorizontalIndicatorApplet"
width="180" height="160">
   <PARAM name="progressbar"    value="true">
   <PARAM name="progresscolor"    value="#000000">
   <PARAM name="debug"    value="0">
   <PARAM name="BACKGRND"    value="LT_GRAY">
   <PARAM name="address"    value="1">
   <PARAM name="datatype"    value="REGISTER">
   <PARAM name="label"    value="label">
   <PARAM name="majorTics"    value="100">
   <PARAM name="minorTics"    value="0">
   <PARAM name="precision"    value="1">
   <PARAM name="maximum"    value="100">
   <PARAM name="minimum"    value="0">
   <PARAM name="maxValue"    value="65535">
   <PARAM name="minValue"    value="0">
   <PARAM name="borderWidth"    value="25">
   <PARAM name="limitHiHi"    value="99">
   <PARAM name="limitHi"    value="95">
   <PARAM name="limitLo"    value="5">
   <PARAM name="limitLoLo"    value="4">
   <PARAM name="deadband"    value="1">
   <PARAM name="rate"    value="1000">
   <PARAM name="unitId"    value="0">
   <PARAM name="colorHiHi"    value="RED">
   <PARAM name="colorHi"    value="ORANGE">
   <PARAM name="colorLoLo"    value="BLUE">
   <PARAM name="colorLo"    value="CYAN">
</APPLET>
```

Horizontal or Vertical Slider Setup

A horizontal or vertical slider gives an analog representation of the direct address of a Modbus device. It is a cursor that is proportional to the value, and it represents a percentage of its range in physical units. Using the mouse, you can change the value of the slider and trigger, and you can send a new value to the Modbus slave.



Properties [Horizontal Slider]

Address: %MW1
Data Type: INT
Label: label
Major Scale Divisions: 100
Minor Scale Divisions: 0
Scale Precision: 1
Maximum EU Value: 100
Minimum EU Value: 0
Maximum PLC Value: 65535
Minimum PLC Value: 0
Unit Increment: 1
Block Increment: 1
Border Width: 20
PLC Type: Premium
Polling rate [ms]: 1000
Read Only: FALSE

The properties of the slider are as follows:

Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 207)*
Data type	Data type of the PLC address	Note 2 *(see page 207)*
Label	Label displayed as part of the graphic object	Note 5 *(see page 207)*
Major Scale Division	Number of major gradations (marked) in the scale	0 to 100
Minor Scale Division	Number of minor gradations (not marked) in the scale	0 to 100
Scale Precision	Number of decimal places shown for the scale gradations (set to -1 to use a general exponential format)	-1 to 6
Maximum EU value	Maximum value, in physical units, of the direct address for scaling	
Minimum EU Value	Minimum value of the direct address for scaling, in physical units	
Maximum PLC Value	Gross maximum value (without scale) of the direct address in the PLC	Note 3 *(see page 207)*

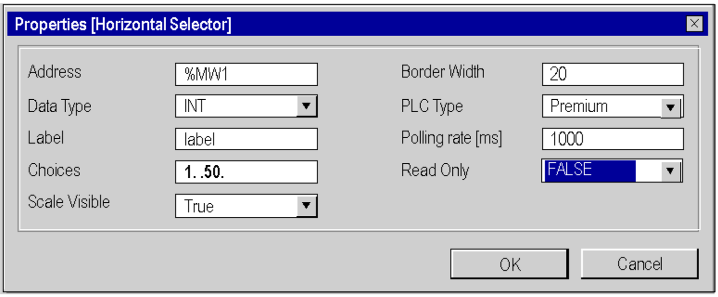
Property	Description	Limits
Minimum PLC Value	Gross minimum value (without scale) of the direct address in the PLC	Note 3 *(see page 207)*
Unit Increment	Amount by which the scale value is modified when you click on the slider arrows	
Block Increment	Amount by which the scale value is modified when you click on the slide area of the bar	
Border Width	Width (in pixels) of the border of the graphic object	
PLC Type	Type of PLC	Premium or Quantum
Polling rate [ms]	Scanning value	
Read Only	TRUE = read value, FALSE = read/write value	

The HTML code with the parameters of the widget above is as follows:

```
<APPLET
codebase="/classes"
archive="mbclient.jar,widgetslite.jar"
code="com.schneiderautomation.factorycast.gateway.sliders.
LiveHorizontalSliderApplet"
width="180" height="160">
   <PARAM name="progressbar"    value="true">
   <PARAM name="progresscolor"    value="#000000">
   <PARAM name="debug"    value="0">
   <PARAM name="BACKGRND"    value="LT_GRAY">
   <PARAM name="address"    value="1">
   <PARAM name="datatype"    value="REGISTER">
   <PARAM name="label"    value="label">
   <PARAM name="majorTics"    value="100">
   <PARAM name="minorTics"    value="0">
   <PARAM name="precision"    value="1">
   <PARAM name="maximum"    value="100">
   <PARAM name="minimum"    value="0">
   <PARAM name="maxValue"    value="65535">
   <PARAM name="minValue"    value="0">
   <PARAM name="unitIncrement"    value="1">
   <PARAM name="blockIncrement"    value="1">
   <PARAM name="rate"    value="1000">
   <PARAM name="unitId"    value="0">
   <PARAM name="borderWidth"    value="20">
   <PARAM name="readOnly"    value="False">
</APPLET>
```

Horizontal or Vertical Selector

A horizontal or vertical selector allows the user to choose from a number of options. Once the selection has been made, the value corresponding to the choice is sent to the PLC. The choices are represented by marks on a "scale", the current selection being indicated by the position of the cursor on a slider.



Properties [Horizontal Selector]

Address	%MW1	Border Width	20
Data Type	INT	PLC Type	Premium
Label	label	Polling rate [ms]	1000
Choices	1..50.	Read Only	FALSE
Scale Visible	True		

The properties of the slider are as follows:

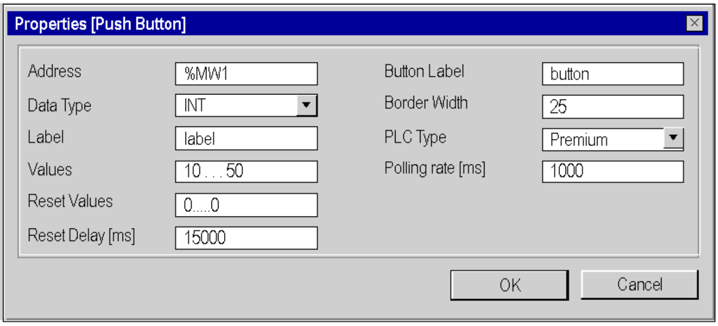
Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 207)*
Data type	PLC address data type	Note 2 *(see page 207)*
Label	Label displayed as part of the graphic object	Note 5 *(see page 207)*
Choices	Setpoints to apply to the address value	
Border Width	Width (in pixels) of the border of the graphic object	
PLC Type	Type of PLC	Premium or Quantum
Polling rate [ms]	Scanning value	
Read Only	True = read value, False = read/write value	

The HTML code with the parameters of the widget above is as follows:

```
<APPLET
codebase="/classes"
archive="mbclient.jar,widgetslite.jar"
code="com.schneiderautomation.factorycast.gateway.selectors
.LiveHorizontalSelectorApplet"
width="180" height="160">
   <PARAM name="progressbar"    value="true">
   <PARAM name="progresscolor"    value="#000000">
   <PARAM name="debug"    value="0">
   <PARAM name="BACKGRND"    value="LT_GRAY">
   <PARAM name="address"    value="1">
   <PARAM name="datatype"    value="REGISTER">
   <PARAM name="label"    value="label">
   <PARAM name="scaleVisible"    value="True">
   <PARAM name="choices"    value="1=1000,9=9000,50=50000">
   <PARAM name="rate"    value="1000">
   <PARAM name="unitId"    value="0">
   <PARAM name="borderWidth"    value="20">
   <PARAM name="readOnly"    value="False">
</APPLET>
```

Push Button Setup

You can send a preset value to one or more Modbus slaves with a push button. A push button is activated with the mouse.



Address	%MW1	Button Label	button
Data Type	INT	Border Width	25
Label	label	PLC Type	Premium
Values	10 . . . 50	Polling rate [ms]	1000
Reset Values	0.....0		
Reset Delay [ms]	15000		

The properties of the push button are as follows:

Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 207)*
Data type	PLC address data type	Note 2 *(see page 207)*
Label	Label displayed as part of the graphic object	Note 5 *(see page 207)*
Values	Setpoints to apply to the address value when the button is pressed	Note 4 *(see page 207)*
Reset Values	Value to apply when the Reset Delay ends	
Reset Delay [ms]	Time in ms counted down after the button is pressed	
Button Label	Label to display on the button	
Border Width	Width (in pixels) of the border of the graphic object	
PLC Type	Type of PLC	Premium or Quantum
Polling rate [ms]	Scanning value	

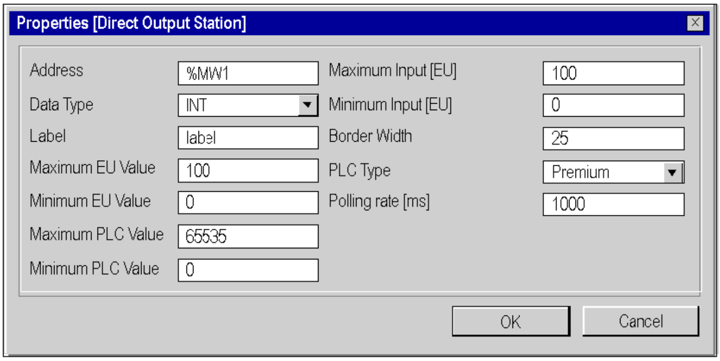
The HTML code with the parameters of the widget above is as follows:

```
<APPLET
codebase="/classes"
archive="mbclient.jar,widgetslite.jar"
code="com.schneiderautomation.factorycast.gateway.pushButton
.LivePushButtonApplet"
width="180" height="160">
   <PARAM name="progressbar"    value="true">
   <PARAM name="progresscolor"    value="#000000">
   <PARAM name="debug"    value="0">
   <PARAM name="BACKGRND"    value="LT_GRAY">
   <PARAM name="address"    value="1">
   <PARAM name="datatype"    value="REGISTER">
   <PARAM name="label"    value="label">
   <PARAM name="values"    value="10,50">
   <PARAM name="resetValues"    value="0">
   <PARAM name="resetDelay"    value="15000">
   <PARAM name="borderWidth"    value="25">
   <PARAM name="buttonLabel"    value="button">
   <PARAM name="rate"    value="1000">
   <PARAM name="unitID"    value="0">
</APPLET>
```

NOTE: In this example, when you press the button, the value 10 is applied to address 1 and the value 50 is applied to address 2 for 15,000 ms. After 15,000 ms, the two addresses are reset to 0.

Direct Output Window

With the Direct Output Window applet, you can enter a value in a text input field directly from the keyboard. If you enter a numerical value between the upper and lower preset limits, an OK button is activated. The value is sent to the Modbus slave each time you click OK or press the ENTER key (if the input field is active for keyboard input).



Properties [Direct Output Station]

Address	%MW1	Maximum Input [EU]	100
Data Type	INT	Minimum Input [EU]	0
Label	label	Border Width	25
Maximum EU Value	100	PLC Type	Premium
Minimum EU Value	0	Polling rate [ms]	1000
Maximum PLC Value	65535		
Minimum PLC Value	0		

The direct output properties are as follows:

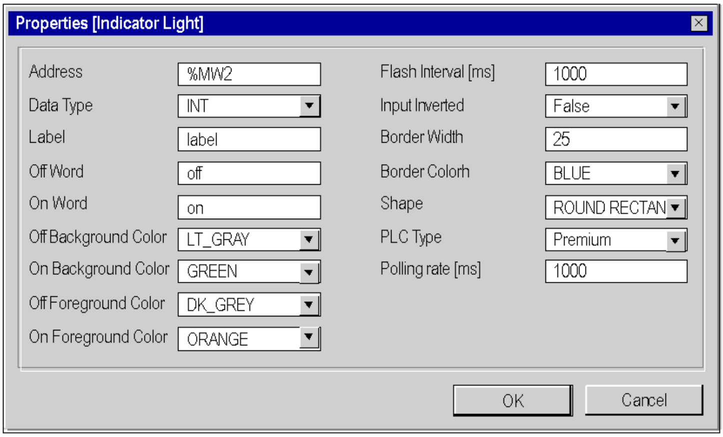
Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 207)*
Data type	PLC address data type	Note 2 *(see page 207)*
Label	Label displayed as part of the graphic object	Note 5 *(see page 207)*
Maximum EU Value	Maximum value, in physical units, of the direct address for scaling	
Minimum EU Value	Minimum value, in physical units, of the direct address for scaling	
Maximum PLC Value	Gross maximum value (without scale) of the direct address in the PLC	Note 3 *(see page 207)*
Minimum PLC Value	Gross minimum value (without scale) of the direct address in the PLC	Note 3 *(see page 207)*
Maximum Input [EU]	Maximum setpoint for the input	
Minimum Input [EU]	Minimum setpoint for the input	
Border Width	Width (in pixels) of the border of the graphic object	
PLC Type	Type of PLC	Premium or Quantum
Polling rate [ms]	Scanning value	

The HTML code with the parameters of the widget above is as follows:

```
<APPLET
codebase="/classes"
archive="mbclient.jar,widgetslite.jar"
code="com.schneiderautomation.factorycast.gateway.direct
Output.LiveDirectOutputApplet"
width="180" height="160">
   <PARAM name="progressbar"    value="true">
   <PARAM name="progresscolor"    value="#000000">
   <PARAM name="debug"    value="0">
   <PARAM name="BACKGRND"    value="LT_GRAY">
   <PARAM name="address"    value="1">
   <PARAM name="datatype"    value="REGISTER">
   <PARAM name="label"    value="label">
   <PARAM name="maximum"    value="100">
   <PARAM name="minimum"    value="0">
   <PARAM name="maxValue"    value="65535">
   <PARAM name="minValue"    value="0">
   <PARAM name="maxInputValue"    value="100">
   <PARAM name="minInputValue"    value="0">
   <PARAM name="rate"    value="1000">
   <PARAM name="unitId"    value="0">
   <PARAM name="borderWidth"    value="25">
</APPLET>
```

Indicator Light Setup

The indicator light provides a dual indication of the value of a direct address in a PLC. If the Input inverted property is not set to TRUE, a 0 input value is declared OFF, and a non-zero value is declared ON. If the Flash Interval property is set to a positive value, the indicator light flashes when the input value is equivalent to ON.



Properties [Indicator Light]

Address	%MW2	Flash Interval [ms]	1000
Data Type	INT	Input Inverted	False
Label	label	Border Width	25
Off Word	off	Border Colorh	BLUE
On Word	on	Shape	ROUND RECTAN
Off Background Color	LT_GRAY	PLC Type	Premium
On Background Color	GREEN	Polling rate [ms]	1000
Off Foreground Color	DK_GREY		
On Foreground Color	ORANGE		

The properties of the indicator light are as follows:

Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 207)*
Data type	PLC address data type	Note 2 *(see page 207)*
Label	Label displayed as part of the graphic object	Note 5 *(see page 207)*
OFF Word	Text displayed when the input value is OFF	
ON Word	Text displayed when the input value is ON	
OFF Background Color	Background color of the indicator light when OFF Word is displayed	
ON Background Color	Background color of the indicator light when ON Word is displayed	
OFF Foreground Color	OFF Word text color	
ON Foreground Color	ON Word text color	
Flash Interval	The flashing time for the indicator light (expressed in ms) when the input value is ON. Set to zero for no flashing	200 to 2000
Input inverted	On TRUE, inverts the input value (the indicator light displays the OFF Word when the input value is ON)	
Border Width	Width (in pixels) of the border of the graphic object	
Border Color	Color of the border	

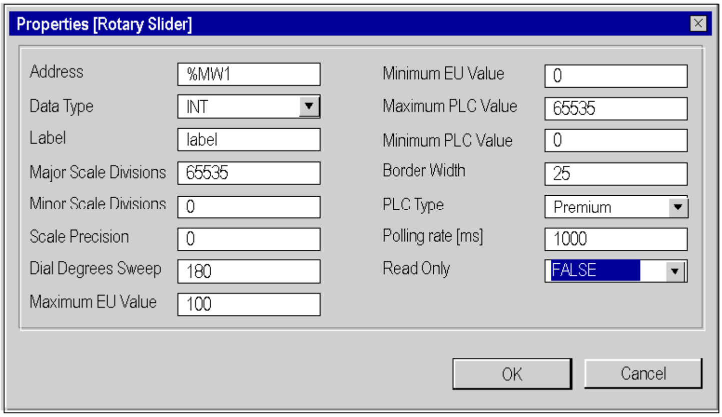
Property	Description	Limits
Shape	Shape (circular, rectangular, etc.) of the indicator light	
PLC Type	Type of PLC	Premium or Quantum
Polling rate [ms]	Scanning value	

The HTML code with the parameters of the widget above is as follows:

```
<p><APPLET
codebase="/classes"
archive="mbclient.jar,widgetslite.jar"
code="com.schneiderautomation.factorycast.gateway.indica-
torLight.LiveIndicatorLightApplet"
width="180" height="160">
   <PARAM name="progressbar"    value="true">
   <PARAM name="progresscolor"    value="#000000">
   <PARAM name="debug"    value="0">
   <PARAM name="BACKGRND"    value="LT_GRAY">
   <PARAM name="address"    value="2">
   <PARAM name="datatype"    value="REGISTER">
   <PARAM name="label"    value="label">
   <PARAM name="offWord"    value="off">
   <PARAM name="onWord"    value="on">
   <PARAM name="offWordBackground"    value="LT_GRAY">
   <PARAM name="onWordBackground"    value="GREEN">
   <PARAM name="offWordForeground"    value="DK_GRAY">
   <PARAM name="onWordForeground"    value="ORANGE">
   <PARAM name="flashInterval"    value="1000">
   <PARAM name="inputInverted"    value="False">
   <PARAM name="borderWidth"    value="25">
   <PARAM name="borderColor"    value="BLUE">
   <PARAM name="shape"    value="ROUND RECTANGLE">
   <PARAM name="rate"    value="1000">
   <PARAM name="unitId"    value="0">
</APPLET>
```

Rotary Slider Setup

A rotary slider gives an analog representation of the direct address of a Modbus device. It draws a position proportional to the value of the address and represents a percentage of its range in physical units on a circular dial. The size of the circular dial (cycle in degrees of a circle) and the button color can be configured. Using the mouse, the user can change the value of the rotary slider and trigger sending a new value to the PLC.



Properties [Rotary Slider]

Address	%MW1	Minimum EU Value	0
Data Type	INT	Maximum PLC Value	65535
Label	label	Minimum PLC Value	0
Major Scale Divisions	65535	Border Width	25
Minor Scale Divisions	0	PLC Type	Premium
Scale Precision	0	Polling rate [ms]	1000
Dial Degrees Sweep	180	Read Only	FALSE
Maximum EU Value	100		

The properties of the slider are as follows:

Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 207)*
Data type	PLC address data type	Note 2 *(see page 207)*
Label	Label displayed as part of the graphic object	Note 5 *(see page 207)*
Major scale gradation	Number of major gradations (marked) in the scale	0 to 100
Minor scale gradation	Number of minor gradations (not marked) in the scale	0 to 100
Scale precision	Number of decimal places in the scale gradations (set to -1 to use a general exponential format)	-1 to 6
Dial Degrees Sweep	Portion of circular dial used to draw the scale	60 to 300
Maximum EU Value	Maximum value of the direct address for scaling, in physical units	
Minimum EU Value	Minimum value of the direct address for scaling, in physical units	
Maximum PLC Value	Gross maximum value (without scale) of the direct address in the PLC	Note 3 *(see page 207)*

Property	Description	Limits
Minimum PLC Value	Gross minimum value (without scale) of the direct address in the PLC	Note 3 *(see page 207)*
Border Width	Width (in pixels) of the border of the graphic object	
PLC Type	Type of PLC	Premium or Quantum
Polling rate [ms]	Scanning value	
Read Only	True = read value, False = read/write value	

The HTML code with the parameters of the widget above is as follows:

```
<APPLET
codebase="/classes"
archive="mbclient.jar,widgetslite.jar"
code="com.schneiderautomation.factorycast.gateway.sliders.
LiveRotosliderApplet"
width="180" height="160">
   <PARAM name="progressbar"    value="true">
   <PARAM name="progresscolor"    value="#000000">
   <PARAM name="debug"    value="0">
   <PARAM name="BACKGRND"    value="LT_GRAY">
   <PARAM name="address"    value="1">
   <PARAM name="datatype"    value="REGISTER">
   <PARAM name="label"    value="label">
   <PARAM name="majorTics"    value="65535">
   <PARAM name="minorTics"    value="0">
   <PARAM name="precision"    value="0">
   <PARAM name="degSweep"    value="180">
   <PARAM name="maximum"    value="100">
   <PARAM name="minimum"    value="0">
   <PARAM name="maxValue"    value="65535">
   <PARAM name="minValue"    value="0">
   <PARAM name="borderWidth"    value="25">
   <PARAM name="rate"    value="1000">
   <PARAM name="readOnly"    value="False">
   <PARAM name="unitID"    value="0">
```

Notes

1.	The address can be:	
	%MW	IEC internal word
	%MD	IEC double word
	%M	IEC internal bit
	400000	Concept integer
	100000	Concept boolean
2.	The various values of the Data type property have the following meaning:	
	Data type	Meaning
	`INT`	`integer`
	`DINT`	`double integer`
	`BOOL`	`boolean`
3.	The limits of the Maximum PLC Value and Minimum PLC Value properties are the natural limits of the configured Data type property.	
4.	Specify at least one value for a push button. If several values are entered, they will be assigned to an address table starting with the direct address indicated.	
5.	If you specify `param name="label" value="$data$'` in the HTML code, the applet displays the numerical value of the data in place of a label.	

Section 5.4
Adding Microsoft Silverlight® Pages

Adding Microsoft Silverlight® Pages

Introduction

Use the commands in Web Designer to create new—or import existing—Microsoft Expression Blend applications, and add them to your Web Designer project. Expression Blend is a powerful development platform you can use to create graphically rich Silverlight pages for your website.

When you launch Expression Blend from Web Designer, the following tools are available for use in Expression Blend:

- a collection of custom FactoryCast objects, developed by Schneider Electric, that you can embed in your Silverlight pages
- a list of PLC variables imported from your Control Expert project that you can use to animate the Factory Cast objects you add to your Silverlight pages

Expression Blend Commands

With a Web Designer project open, you can use the following commands to add an Expression Blend application to your website.

To create a new Expression Blend application:

<Project> → <Target> → GraphicScreens → New Microsoft Blend Application

To import an existing Expression Blend application:

<Project> → <Target> → GraphicScreens → Import Blend Application

Further Reference

For instructions on how to use Expression Blend and the Schneider Electric Factory Cast custom objects, refer to the following documents:

- Expression Blend help
- *Web Designer Objects Library for Microsoft Expression Blend®*, which you can access as follows:
  - as a help file, from the Web Designer software **Help** menu, or
  - in .pdf format from the Web Designer installation disk

Chapter 6
SOAP Web Services

Purpose of this chapter

This chapter describes the various SOAP Web services implemented in FactoryCast Web servers. SOAP Web services are fully compliant with the W3C WS-I Web services standards. They provide a new efficient and standard means of making controller devices interact directly with IT/management level applications.

⚠ **WARNING**

LOSS OF PROCESS ALERTS AND CONTROL

- Do not rely solely on the SOAP Web Services to retrieve and display or control events that could affect human, material or equipment integrity.
- Provide appropriate and independent protection in your own application or process.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Web services are based on standards such as:

- SOAP (Single Object Access Protocol), the exchange protocol carried out via the HTTP (HyperText Transfer Protocol) channel.
- WSDL (Web Services Description Language), in XML format.
- XML (eXtensible Markup Language), the universal data exchange standard.

FactoryCast SOAP Web services act as SOAP server interfaces. They allow developers to easily design client applications that can exchange data directly with FactoryCast Web servers. Applications such as Microsoft.NET, SQL Server, Microsoft Office, IBM (WebSphere), SUN (Java, Eclipse), Lotus, Oracle, SAP, MES, ERP and so forth can be interfaced directly with FactoryCast using SOAP Web services.

Three types of web services are provided in FactoryCast modules as SOAP server interfaces:

- ModbusXMLDA: Web service to implement data access to Modbus variables
- SymbolicXMLDA: Web service to implement Symbolic data access
- ExtendedSymbolicXMLDA: Web service to extend Symbolic data access to Control Expert.

The Web services provided by FactoryCast are compatible with the WS-I basic profile 1.1.

NOTE: For more information on Web services, please refer to specialized publications on the subject.

What Is in This Chapter?

This chapter contains the following topics:

Principles for Designing a SOAP Client Interface

Overview

An interface enables a SOAP client application to communicate directly with a FactoryCast Web server module. Exchanges are initiated by the SOAP client application. The FactoryCast server responds to these requests.

A SOAP Client Interface

The following table describes the process of designing a SOAP client interface:

Stage	Action
1	Create the client application: The development environment (for example, Visual Studio.net) connects to a FactoryCast Web server module where it can access a list of available Web services. The Web server returns descriptions of the requested services as WSDL objects.
2	Develop the client application: The developer integrates the Web service APIs using the code retrieved in the previous stage as a Web reference and generates the client application.
3	Execute the client application: In run mode, the client application communicates in real time with the FactoryCast Web server module using the SOAP protocol.

ModBusXMLDA SOAP Services

Overview

The ModbusXMLDA Web Service is implemented to provide data access to Modbus variables using a syntax similar to the Modbus protocol.

This Web service offers the following APIs:
- ReadDeviceIdentification,
- ReadMultipleRegisters,
- WriteMultipleRegisters,
- ReadCoils,
- WriteMultipleCoils,
- ReadInt32,
- WriteInt32.

Data amounts for API parameters

This table presents the maximum amount of data that can be used for API parameters:

APIs	BMX NOE 0110
ReadMultipleRegisters	124 (Register)
WriteMultipleRegisters	96 (Register)
ReadCoils	255 (Coils)
WriteMultipleCoils	254 (Coils)
ReadInt32	61 (Int32)
WriteInt32	48 (Int32)

ReadDevice-Identification

The ReadDeviceIdentification request returns the entire device identification of the destination specified by the ID provided in the request.

This API is mapped on Modbus Function Code 43 14.

Parameters

Input: int UnitID

Output: string VendorName, ProductCode, MajorMinorRevision, VendorURL, ProductName, ModelName, UserApplicationName, TRImplementationClass, TRCommunicationServices

Example

The following is an example of a SOAP request and its response:

Request

```
POST /ws/ModbusXmlDa
HTTP/1.0 Host: 139.160.65.83:8080
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.schneider-electric.com/ws/ModbusXmlDa/ReadDevi-
ceIdentification"
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <ReadDeviceIdentification xmlns="http://www.schneider-
electric.com/ws/ModbusXmlDa/">
       <UnitID>int</UnitID>
     </ReadDeviceIdentification>
   </soap:Body>
 </soap:Envelope>
```

Response

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length <?xml version="1.0" encoding="utf-8"
?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <ReadDeviceIdentificationResponse xmlns="http://www.schneider-
electric.com/ws/ModbusXmlDa/">
       <ReadDeviceIdentificationResult>
         <DeviceIdentification>
                 <VendorName>string</VendorName>
                 <ProductCode>string</ProductCode>
           <MajorMinorRevision>string</MajorMinorRevision>
           <VendorUrl>string</VendorUrl>
           <ProductName>string</ProductName>
```

```
                <ModelName>string</ModelName>
                <UserApplicationName>string</UserApplicationName>
                <TRImplementationClass>string</TRImplementationClass>
                <TRCommunicationServices>string</TRCommunicationServices>
              </DeviceIdentification>
            </ReadDeviceIdentificationResult>
          </ReadDeviceIdentificationResponse>
        </soap:Body>
     </soap:Envelope>
```

ReadMultiple-Registers

This is the ReadMultipleRegisters API used to implement a read multiple registers action.

The request provides three parameters:
- the destination of the read request,
- the starting address,
- the number of values to read (base index is provided by starting address).

In response, the list of values is provided.

This API is mapped on Modbus Function Code 03.

Parameters

Input: int UnitID, Address, Quantity

Output: array of int Result

Example

The following is an example of a SOAP request and its response:

Request

```
POST /ws/ModbusXmlDa
 HTTP/1.0 Host: 139.160.65.83:8080
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.schneider-
electric.com/ws/ModbusXmlDa/ReadMultipleRegisters"
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <ReadMultipleRegisters xmlns="http://www.schneider-
electric.com/ws/ModbusXmlDa/">
       <UnitID>int</UnitID>
       <Address>int</Address>
       <Quantity>int</Quantity>
     </ReadMultipleRegisters>
   </soap:Body>
 </soap:Envelope>
```

Response

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length <?xml version="1.0" encoding="utf-8" ?>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <ReadMultipleRegistersResponse xmlns="http://www.schneider-
electric.com/ws/ModbusXmlDa/">
       <ReadMultipleRegistersResult>
         <int>int</int>
         <int>int</int>
       </ReadMultipleRegistersResult>
     </ReadMultipleRegistersResponse>
   </soap:Body>
 </soap:Envelope>
```

WriteMultiple-Registers

This is the WriteMultipleRegisters API used to implement a write multiple registers action.

The request provides three parameters:

- the destination of the write request,
- the starting address,
- the values to write.

This API is mapped on Modbus Function Code 16.

Parameters

Input: int UnitID, Address, array of int Value

Output: none

Example

The following is an example of a SOAP request and its response:

Request

```
POST /ws/ModbusXmlDa
HTTP/1.0 Host: 139.160.65.83:8080
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.schneider-electric.com/ws/ModbusXmlDa/WriteMul-
tipleRegisters"
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <WriteMultipleRegisters xmlns="http://www.schneider-
electric.com/ws/ModbusXmlDa/">
       <UnitID>int</UnitID>
       <Address>int</Address>
       <Value>
         <int>int</int>
         <int>int</int>
       </Value>
     </WriteMultipleRegisters>
 </soap:Body>
</soap:Envelope>
```

Response

```
HTTP/1.0 200 OKContent-Type: text/xml; charset=utf-8
Content-Length: length <?xml version="1.0" encoding="utf-8" ?>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
     <WriteMultipleRegistersResponse xmlns="http://www.schneider-
electric.com/ws/ModbusXmlDa/">
     </WriteMultipleRegistersResponse>
  </soap:Body>
 </soap:Envelope>
```

ReadCoils

This is the ReadCoils API used to implement a read multiple coils action.

The request provides three parameters:

- the destination of the read request,
- the starting address,
- the number of values to read (base index is provided by starting address).

In response, the list of values is provided.

This API is mapped on Modbus Function Code 01.

Parameters

Input: int UnitID, Address, Quantity

Output: array of int Result

Example

The following is an example of a SOAP request and its response:

Request

```
POST /ws/ModbusXmlDa
HTTP/1.0 Host: 139.160.65.83:8080
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.schneider-
electric.com/ws/ModbusXmlDa/ReadCoils"
<?xml version="1.0" encoding="utf-8" ?>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <ReadCoils xmlns="http://www.schneider-
electric.com/ws/ModbusXmlDa/">
       <UnitID>int</UnitID>
       <Address>int</Address>
       <Quantity>int</Quantity>
     </ReadCoils>
   </soap:Body>
</soap:Envelope>
```

Response

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8" ?>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
     <ReadCoilsResponse xmlns="http://www.schneider-
electric.com/ws/ModbusXmlDa/">
       <ReadCoilsResult>
         <int>int</int>
         <int>int</int>
       </ReadCoilsResult>
     </ReadCoilsResponse>
  </soap:Body>
</soap:Envelope>
```

WriteMultiple-Coils

This is the WriteMultipleCoils API used to implement a write multiple coils action.

The request provides three parameters:

- the destination of the write request,
- the starting address,
- the values to write.

This API is mapped on Modbus Function Code 15.

Parameters

Input: int UnitID, Address, array of int Value

Output: none

Example

The following is an example of a SOAP request and its response:

Request

```
POST /ws/ModbusXmlDa HTTP/1.0
Host: 139.160.65.83:8080
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.schneider-electric.com/ws/ModbusXmlDa/WriteMul-
tipleCoils"
<?xml version="1.0" encoding="utf-8" ?>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <WriteMultipleCoils xmlns="http://www.schneider-
electric.com/ws/ModbusXmlDa/">
       <UnitID>int</UnitID>
       <Address>int</Address>
       <Value>
         <int>int</int>
         <int>int</int>
       </Value>
     </WriteMultipleCoils>
   </soap:Body>
 </soap:Envelope>
```

Response

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8" ?>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <WriteMultipleCoilsResponse xmlns="http://www.schneider-
electric.com/ws/ModbusXmlDa/">
     </WriteMultipleCoilsResponse>
   </soap:Body>
</soap:Envelope>
```

ReadInt32 SOAP request/response

The ReadInt32 request provides three parameters:

- The destination of the read request,
- the starting address,
- the number of values to read (base index is provided by starting address).

Parameters

Input: int UnitID, Address, Quantity

Output: array of int Result

Example

In response, the list of values is provided. The following is an example of a SOAP request and response:

Request

```
POST /ws/ModbusXmlDa
HTTP/1.0 Host: 139.160.65.83:8080
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.schneider-
electric.com/ws/ModbusXmlDa/ReadMultipleRegisters"
<?xml version="1.0" encoding="utf-8" ?>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <ReadInt32 xmlns="http://www.schneider-
electric.com/ws/ModbusXmlDa/">
       <UnitID>int</UnitID>
       <Address>int</Address>
       <Quantity>int</Quantity>
     </ ReadInt32 >
   </soap:Body>
 </soap:Envelope>
```

Response

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8" ?>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     < ReadInt32Response xmlns="http://www.schneider-
electric.com/ws/ModbusXmlDa/">
       < ReadInt32Result>
         <int>int</int>
         <int>int</int>
       </ ReadInt32Result>
     </ ReadInt32Response>
   </soap:Body>
 </soap:Envelope>
```

WriteInt32 SOAP request/response

The WriteInt32 request provides three parameters:

- The destination of the write request,
- the starting address,
- the values to write.

Parameters

Input: int UnitID, Address, array of int Value

Output: none

Example

The following is an example of a SOAP request and response:

Request

```
POST /ws/ModbusXmlDa
HTTP/1.0 Host: 139.160.65.83:8080
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.schneider-electric.com/ws/ModbusXmlDa/WriteMul-
tipleRegisters"
<?xml version="1.0" encoding="utf-8" ?>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <WriteInt32 xmlns="http://www.schneider-
electric.com/ws/ModbusXmlDa/">
       <UnitID>int</UnitID>
       <Address>int</Address>
       <Value>
         <int>int</int>
         <int>int</int>
       </Value>
     </ WriteInt32 >
   </soap:Body> </soap:Envelope>
```

Response

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8" ?>
 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
   < WriteInt32Response xmlns="http://www.schneider-
electric.com/ws/ModbusXmlDa/">
      </ WriteInt32Response>
   </soap:Body>
 </soap:Envelope>
```

SymbolicXMLDA SOAP Services

Overview

The SymbolicXMLDA is the Web Service provided to implement a Symbolic Data Access based on a FactoryCast namespace. The services are:

- read,
- write,
- browse.

Limitation on symbolic access

As symbolic access relies on low-level Modbus access for actual communication, only symbols mapped on data accessible through native Modbus requests (Read/Write registers and Read/Write Coils) are supported. Consequently, only data of these types are supported:

- bit
- 16-bit
- 32-bit double word

Unlocated data of all types are not supported.

Read service

The aim of this request is to read one or several symbols. The symbols passed as request arguments are translated using the namespace to get the address of the value to read. When all addresses are known, a Modbus request is issued by contiguous addresses (i.e., if registers 10, 11, and 13 are requested, two requests are issued: one for 10 and 11, and one for 13) and by the type of variable desired (coils, registers). Once the values are retrieved, the SOAP response is built using the variable name, type and values.

Parameters

Input: list of string ItemName

Output: array of Item ReadResult

Example

The following is a example SOAP request and response:

Request

```
POST /ws/SymbolicXmlDa.asmx  HTTP/1
Host: 139.160.65.83:8080
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.schneider-
electric.com/ws/SymbolicXmlDa/1.0/Read"
  <?xml version="1.0" encoding="utf-8"?>
  <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
      <Read xmlns="http://www.schneider-
electric.com/ws/SymbolicXmlDa/1.0/">
        <ItemList>
          <Items>
            <ItemName>string</ItemName>
          </Items>
          <Items>
            <ItemName>string</ItemName>
          </Items>
        </ItemList>
      </Read>
    </soap:Body>
  </soap:Envelope>
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
      <WriteResponse xmlns="http://www.schneider-
electric.com/ws/SymbolicXmlDa/">
      </WriteResponse>
    </soap:Body>
</soap:Envelope>
```

Write service

The aim of this request is to write one or several symbols. The symbols, their types, and values are passed as request arguments. The names of the variables are translated using the namespace to get the address of the value to read. When all addresses are known, a Modbus request is issued by contiguous addresses (i.e., if register 10, 11, and 13 are requested, two requests are issued: one for 10 and 11, and one for 13) and by type of variable desired (coils, registers).

After the values are written, the SOAP response is built.

Parameters

Input: list of (string ItemName, Value, Type)

Output: none

Example

The following is a example SOAP request and response:

Request

```
POST /Recipe/ws/SymbolicXmlDa.asmx HTTP/1.1
Host: 139.160.65.83:8080
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.schneider-
electric.com/ws/SymbolicXmlDa/1.0/Write"
<?xml version="1.0" encoding="utf-8"?>
 <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Body>
     <Write xmlns="http://www.schneider-
electric.com/ws/SymbolicXmlDa/1.0/">
       <ItemList>
         <Items>
           <ItemName>string</ItemName>
           <Value>string</Value>
           <Type>string</Type>
         </Items>
         <Items>
           <ItemName>string</ItemName>
           <Value>string</Value>
           <Type>string</Type>
         </Items>
      </ItemList>
     </Write>
   </soap:Body>
 </soap:Envelope>
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ReadResponse xmlns="http://www.schneider-
electric.com/ws/SymbolicXmlDa/">
       <ReadResult>
       <Items>
          <Name>string</Name
          <SymbolType>string</SymbolType>
          <IntValue>int</IntValue>
       </Items>
       <Items>
          <Name>string</Name>
          <SymbolType>string</SymbolType>
          <IntValue>int</IntValue>
     </Items>
     </ReadResult>
     </ReadResponse>
   </soap:Body>
</soap:Envelope>
```

Browse service

The aim of this request is to read a list of symbolic variables present on the namespace of the FactoryCast module.

This functionality does not interface with the Modbus layer since its only action is to return the namespace items. The response contains the symbols and their types.

Parameters

Input: list of (string ItemName, Type)

Output: list of Descriptions BrowseResult

Example

The following is a example SOAP request and response:

Request

```
POST /Recipe/ws/SymbolicXmlDa.asmx
HTTP/1.1 Host: 139.160.65.83:8080
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.schneider-
electric.com/ws/SymbolicXmlDa/1.0/Browse"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <Browse xmlns="http://www.schneider-
electric.com/ws/SymbolicXmlDa/1.0/" />
   </soap:Body> </soap:Envelope>
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length <?xml version="1.0" encoding="utf-8"?>
 <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <BrowseResponse xmlns="http://www.schneider-
electric.com/ws/SymbolicXmlDa/1.0/">
       <Elements>
        <BrowseElement>
           <ItemName>string</ItemName>
           <Type>string</Type>
         </BrowseElement>
         <BrowseElement>
          <ItemName>string</ItemName>
           <Type>string</Type>
         </BrowseElement>
      </Elements>
     </BrowseResponse>
   </soap:Body>
 </soap:Envelope>
```

ExtendedSymbolicXMLDA SOAP Services

Presentation

The ExtendedSymbolicXMLDA extends the SymbolicXmlDA SOAP services for Control Expert. It allows you to access direct addresses data and unlocated data. The services are:

- read,
- write,
- browse.

Limitation on variables access

As variables access relies on low-level Control Expert protocol access for actual communication, only symbols and direct addresses mapped on data accessible through native Control Expert requests (Read/Write registers and Read/Write Coils) are supported. Consequently, only data of these types are supported:

Type	Syntax
Internal Variables	%Mi, %MXi, %MWi, %MWi.j
Constants	%KWi
System Variables	%Si, %SXi, %SWi
Simulated	%MDi, %MFi, %KDi, %KFi

Supported types are all basic Control Expert types including elements of arrays.

For direct addressing, supported syntax are:

Syntax	Type	Comment
%Mi or %MXi	EBOOL	mapped on XML/W3C type unsignedByte
%MWi	INT	mapped on XML/W3C type short
%MWi.j	BOOL	mapped on XML/W3C type boolean
%MDi	DINT	mapped on XML/W3C type int
%MFi	REAL	mapped on XML/W3C type float
%KWi	INT	mapped on XML/W3C type short
%KWi.j	BOOL	mapped on XML/W3C type boolean
%KDi	DINT	mapped on XML/W3C type int
%KFi	REAL	mapped on XML/W3C type float
%Si or %SXi	BOOL	mapped on XML/W3C type boolean
%SWi	INT	mapped on XML/W3C type short
%SWi.j	BOOL	mapped on XML/W3C type boolean

For Control Expert symbols, supported types are:

Type	Comment
BOOL	mapped on XML/W3C type boolean
EBOOL	mapped on XML/W3C type unsignedByte
INT	mapped on XML/W3C type short
DINT	mapped on XML/W3C type int
UINT	mapped on XML/W3C type unsignedShort
UDINT	mapped on XML/W3C type unsignedInt
TIME	mapped on XML/W3C type unsignedInt
DATE	mapped on XML/W3C type date or dateTime (dateTime is mandatory for write, date part is ignored)
TOD	mapped on XML/W3C type time or dateTime (dateTime is mandatory for write, time part is ignored)
DT	mapped on XML/W3C type dateTime
REAL	mapped on XML/W3C type float or double (double is only for write request, cast the value in float)
BYTE	mapped on XML/W3C type unsignedByte
WORD	mapped on XML/W3C type unsignedShort
DWORD	mapped on XML/W3C type unsignedInt
STRING	mapped on XML/W3C type string
STRING[n]	mapped on XML/W3C type string

NOTE:

- The name of the variable can be either a direct address or any symbols that appear in the FactoryCast namespace.
- If a Web service function has an optional parameter, Visual Studio .NET creates an additional boolean parameter in the corresponding proxy. Visual Studio .NET adds "Specified" at the end of the name of that parameter.

Read service

The aim of this request is to read symbols and direct addresses. The symbols passed as request arguments are translated using the namespace to get the address of the value to read. Direct addresses are passed as request argument. When all addresses are known, a set of Control Expert requests is issued by contiguous addresses (i.e., if registers 10, 11, and 13 are requested, two requests are issued: one for 10 and 11, and one for 13) and by type of variable desired (coils, registers). After the values are retrieved, the SOAP response is built using the variable name, type and values.

Parameters

Input: list of string ItemName

Output: array of Item ReadResult

Example

The following is a example SOAP request and response:

Request

```
POST /ws/ExtendedSymbolicXmlDa.asmx  HTTP/1
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.schneider-electric.com/ws/ExtendedSymbol-
icsXmlDa/Read"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
      <Read xmlns="http://www.schneider-electric.com/ws/ExtendedSymbol-
icsXmlDa/">
        <VariableList>
            <Name>string</Name>
            <Name>string</Name>
        </VariableList>
      </Read>
    </soap:Body>
  </soap:Envelope>
```

Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

                 <soap:Body>

    <ReadResponse xmlns="http://www.schneider-
electric.com/ws/ExtendedSymbolicXMLDa/">

    <ReadResult>

    <Item>

     <Name>string</Name>
```

```
          <VariableType>string</VariableType>
          <Value />
        </Item>
        <Item>
          <Name>string</Name>
          <VariableType>string</VariableType>
          <Value />
        </Item>
        </ReadResult>
      </ReadResponse>
  </soap:Body>
</soap:Envelope>
```

Write service

The aim of this request is to write symbols or direct addresses. The symbols, their types, and values are passed as request arguments. The names of the variables are translated using the namespace to get the address of the value to read. Direct addresses are passed as request argument. When all addresses are known, a set of Control Expert requests is issued by contiguous addresses (i.e., if register 10, 11, and 13 are requested, two requests are issued: one for 10 and 11, and one for 13) and by type of variable desired (coils, registers).

After the values are written, the SOAP response is built.

Parameters

Input: list of (string ItemName, Value, Type)

Output: none

Example

The following is a example SOAP request and response:

Request

```
POST /maquette/ExtendedSymbolicXmlDa.asmx HTTP/1.1
Host: 139.160.65.83:8080
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.schneider-electric.com/ws/ExtendedSymbol-
icsXmlDa/Write"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
 <soap:Body>
     <Write xmlns="http://www.schneider-electric.com/ws/ExtendedSymbol-
icXmlDa/">
       <ItemList>
         <Item>
           <Name>string</Name>
           <VariableType>string</VariableType>
           <Value />
         </Item>
         <Item>
           <Name>string</Name>=
           <VariableType>string</VariableType>
           <Value />
         </Item>
      </ItemList>
     </Write>
   </soap:Body>
</soap:Envelope>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/">
   <soap:Body>
     <WriteResponse xmlns="http://www.schneider-
electric.com/ws/ExtendedSymbolicXmlDa/">
    </WriteResponse>

          </soap:Body>
</soap:Envelope>
```

Browse service

The aim of this request is to read a list of symbolic variables (PLC, unlocated and I/O) present on the namespace of the FactoryCast module.

The response contains all the symbols and their types.

Parameters

Input: none or the optional parameter

Output: list of descriptions (name, address and symbol type)

Example

The following is a example SOAP request and response:

Request

```
POST /maquette/ExtendedSymbolicXmlDa.asmx
HTTP/1.1 Host: 139.160.65.83:8080
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
SOAPAction: xmlns="http://www.schneider-
electric.com/ws/ExtendedSymbolicXmlDa/Browse"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/2003/05/soap-envelope/">
   <soap:Body>
```

```
<Browse xmlns="http://www.schneider-electric.com/ws/ExtendedSymbol-
icXmlDa/Browse" />
   </soap:Body>
         </soap:Envelope>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/">
   <soap:Body>
     <BrowseResponse xmlns="http://www.schneider-
electric.com/ws/ExtendedSymbolicXmlDa/">
        <BrowseResult>
        <Description>
           <Name>string</Name>
           <Address>string</Address>
           <VariableType>string</VariableType>
         </Description>>
         <Description>
           <Name>string</Name>
           <Address>string</Address>
           <VariableType>string</VariableType>
         </Description>
        </BrowseResult>
     </BrowseResponse>
   </soap:Body>
 </soap:Envelope>
```

How to optimize massive symbolic requests

Overview

The symbolic read/write request allows you to perform requests of up to 128 symbols.

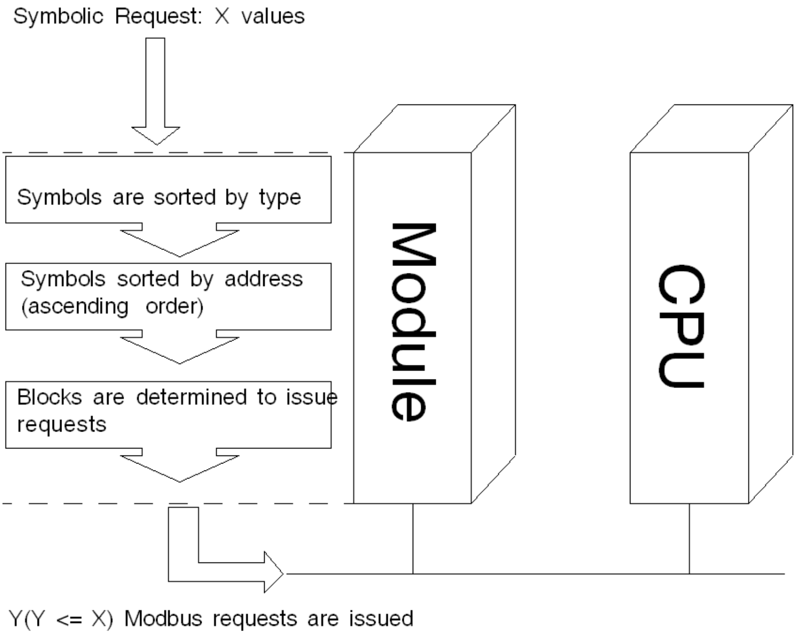
You can experience a lag when using a large amount of symbols. There are several rules you can apply to improve your application’s speed, by reducing the number of internal requests.

The purpose of this topic is to help you create a coherent namespace and use symbolic requests efficiently.

Symbolic request process

Symbolic requests enable the user to use symbols instead of addresses to handle PLC data. This request system uses the namespace on the module to resolve the addresses. Once the addresses are known, there is an optimization process to sort the addresses by type (coils, registers) in ascending order.

The sorted addresses should be grouped by coherent block to minimize the number of requests sent to the CPU. The graphic below shows the three steps of the symbolic request optimization:


Symbolic Request: X values
Symbols are sorted by type
Symbols sorted by address (ascending order)
Blocks are determined to issue requests
Module
CPU
Y(Y <= X) Modbus requests are issued

Optimization method

To optimize a massive symbolic request, consider grouping symbols as follows:

Step	Action
1	Group symbols by type: don't mix coils and registers
2	If you can, sort symbols in ascending order
3	Try to determine blocks of symbols (i.e., contiguous symbol addresses). These blocks will be processed in only one low-level request.

Optimization example

Considering a relatively simple namespace file:

```
|STATION:0.0
16
|Bit_mem1|%M0|0|0|1|1|1|false
|Bit_mem2|%M1|0|0|1|1|1|false
|Bit_mem3|%M2|0|0|1|1|1|false
|Bit_mem4|%M4|0|0|1|1|1|false
|Bit_mem5|%M5|0|0|1|1|1|false
|Bit_mem6|%M6|0|0|1|1|1|false
|Engine_Brake_lvl|%MW0|0|0|4|1|1|false
|Engine_Brake_dsk|%MW1|0|0|4|1|1|false
|Engine_gauge_1|%MW2|0|0|4|1|1|false
|Engine_gauge_2|%MW3|0|0|4|1|1|false
|Engine_gauge_3|%MW4|0|0|4|1|1|false
|Engine_gauge_4|%MW5|0|0|4|1|1|false
|Engine2_Brake_lvl|%MW100|0|0|4|1|1|false
|Engine2_Brake_dsk|%MW101|0|0|4|1|1|false
|Engine2_gauge_1|%MW102|0|0|4|1|1|false
|Engine2_gauge_2|%MW103|0|0|4|1|1|false
|Engine2_gauge_3|%MW104|0|0|4|1|1|false
|Engine2_gauge_4|%MW105|0|0|4|1|1|false
0
1
false
```

There are values for two engines (engine and engine2).

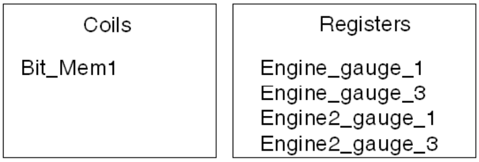
There are also Boolean values (Bit_mem).

If you request every symbol in the namespace order, you will receive one SOAP request with 16 values. Because type and addresses (in ascending order) sort these values, the optimization will only determine blocks of addresses. Here we can see three blocks:

- coils from address 0 to 6,
- registers from address 0 to 5,
- registers from address 100 to 105.

There will be three low-level requests sent to the CPU.

If the end user asks for five symbols in one SOAP request: Engine_gauge_3, Engine_gauge_1, Bit_Mem1, Engine2_gauge_3, Engine2_gauge_1.

Step	Action
1	Since symbols are mixed, the first optimization step is to sort between coils and registers.
2	In this step, symbols of the same type are sorted by address. As the SOAP request is not well ordered, this step has to process the data. At the end of this step we have two arrays: an array of one coil and an array of four registers:  Coils Bit_Mem1 Registers Engine_gauge_1 Engine_gauge_3 Engine2_gauge_1 Engine2_gauge_3
3	This step should find contiguous addresses. Since there are no contiguous addresses, we have five blocks of one address. The final result is that five low-level requests are sent to the CPU.

In brief

When you require intensive use (in number, in time constraints) of symbolic requests, follow these rules:

- group the symbols by types,
- sort the symbols in ascending order,
- group the symbols by blocks of contiguous addresses.

Visual Basic Examples

Overview

To help you start writing your application, here is a Visual Basic .NET (2003 or 2005) example of accessing SOAP requests.

Preliminary Steps

The following table describes the actions to carry out before using the example.

Step	Action
1	In Visual Basic .NET, select **Project** → **Add WebReference**.
2	Type `http://Module @IP/ws/ModbusXmlDa`.
3	Select the desired function.
4	Click the **Add Reference** button.

Example

The following example reads ten consecutive registers from register 5.

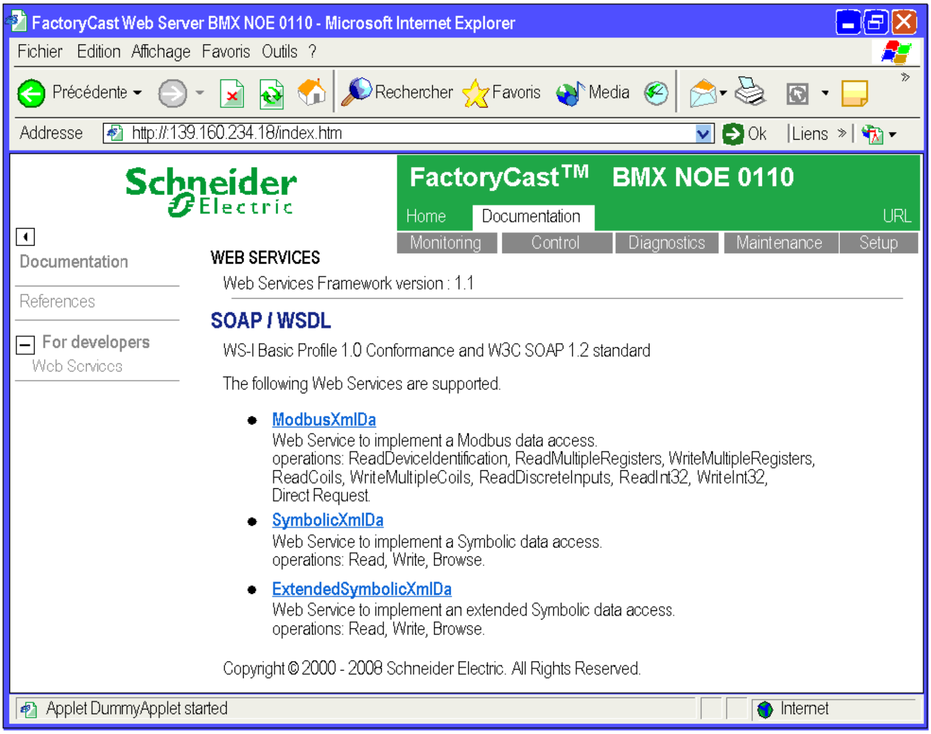
```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

        Dim ws As New WindowsApplication26.WebReference.ModbusXmlDa

        ListBox1.DataSource = ws.ReadMultipleRegisters(0, 5, 10)

    End Sub

End Class
```

Online SOAP Documentation

Overview

When you open a FactoryCast Web server page in a browser, you can find a **Documentation** hyperlink.

This link takes you to SOAP online documentation, which describes the SOAP Web services available, explains the syntax of SOAP APIs, and provides Visual Basic example.



FactoryCast Web Server BMX NOE 0110 - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Précédente
Rechercher
Favoris
Media
Addresse
http://139.160.234.18/index.htm
Ok
Liens
Schneider Electric
FactoryCast™ BMX NOE 0110
Home
Documentation
URL
Monitoring
Control
Diagnostics
Maintenance
Setup
Documentation
References
For developers
Web Services
Applet DummyApplet started
Internet

WEB SERVICES

Web Services Framework version : 1.1

SOAP / WSDL

WS-I Basic Profile 1.0 Conformance and W3C SOAP 1.2 standard

The following Web Services are supported.

- ModbusXmlDa
  Web Service to implement a Modbus data access.
  operations: ReadDeviceIdentification, ReadMultipleRegisters, WriteMultipleRegisters, ReadCoils, WriteMultipleCoils, ReadDiscreteInputs, ReadInt32, WriteInt32, Direct Request.
- SymbolicXmlDa
  Web Service to implement a Symbolic data access.
  operations: Read, Write, Browse.
- ExtendedSymbolicXmlDa
  Web Service to implement an extended Symbolic data access.
  operations: Read, Write, Browse.

Copyright © 2000 - 2008 Schneider Electric. All Rights Reserved.

Appendices

Appendix A
Browser Requirements, Settings, and Security Considerations

Overview

To view the Java applets on FactoryCast Web sites, use a browser that supports both Java® and Silverlight® pages. This appendix discusses that requirement and other browser considerations.

What Is in This Chapter?

This chapter contains the following topics:

Browser Version

Overview

To display FactoryCast web pages, your browser must support both Java® and Silverlight® pages. Accomplish this by installing the following software:

- Java Standard Edition (Java SE), beginning with Java 2 version 1.6 or higher. Use the Sun Java plug-in, available at the Web site: http://java.sun.com. Choose download and install the Java SE software.
- Silverlight 4 plug-in. You can download the installation software for the Silverlight 4 plug-in at the www.microsoft.com web site.
  NOTE: The Silverlight plug-in supports the display of the Silverlight Rack Viewer pages. If this plug-in is not installed, Rack Viewer displays as Java pages.

Which Browsers Qualify?

Silverlight pages are supported by the following combinations of operating system platforms and browsers:

OS Platform	Browser					
	IE 8	IE 7	IE 6	Firefox 3+	Safari 3+	Chrome 4+
Windows Vista	X	X		X		X
Windows 7	X			X		X
Windows Server 2008	X	X		X		X
Windows Server 2008 R2	X					X
Windows Server 2003, Windows XP SP2, SP3	X	X	X	X		X
Windows 2000 SP4+			X			
Macintosh OS 10.4.11+ (Intel-based)				X	X	
X = supported						

For additional information, refer to the Microsoft Silverlight website at:

http://www.microsoft.com/silverlight/faq/#sys-req

Which Browser Do I Have?

Your browser name and version are displayed at the bottom of the FactoryCast Home Page. If you need to upgrade your browser, the Internet addresses for the two major browsers are listed below. (Upgrades are free.You only need to download the latest version.)

How Do I Upgrade?

Supported browsers include the following:

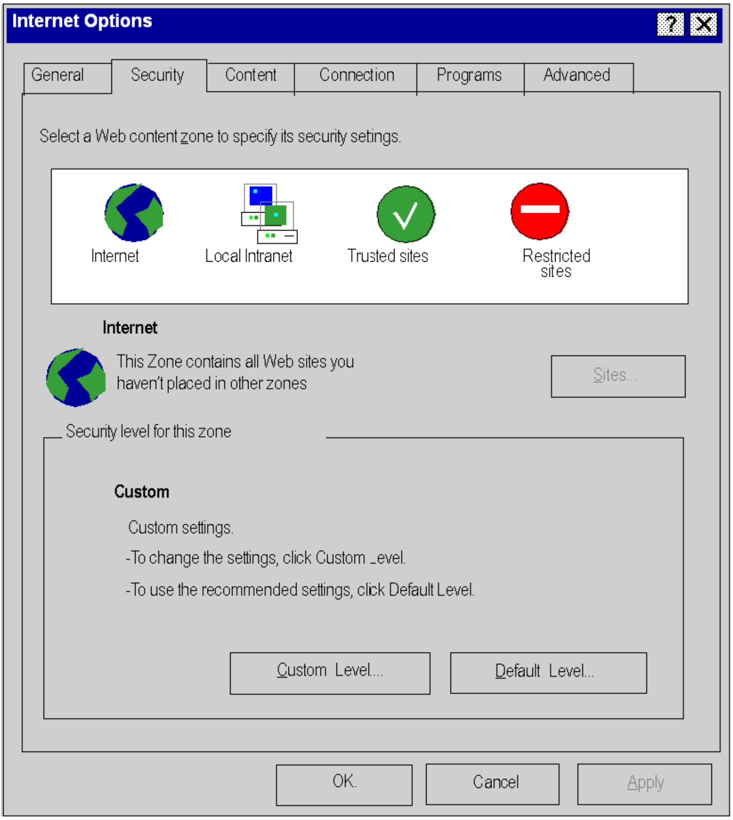
Download the browser:	At Web Site:
Mozilla Firefox	http://www.mozilla.com
Internet Explorer	http://www.microsoft.com

Browser Settings

Modifying Security Settings

If you are using Microsoft Internet Explorer and you cannot view Java applets, modify your browser security settings, as follows:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Open Internet Explorer.</td></tr>
<tr><td>2</td><td>Select <b>Tools | Internet Options</b> from the menu bar.
<b>Result:</b> The <b>Internet Options</b> window appears.</td></tr>
<tr><td>3</td><td>Select the <b>Security</b> tab.



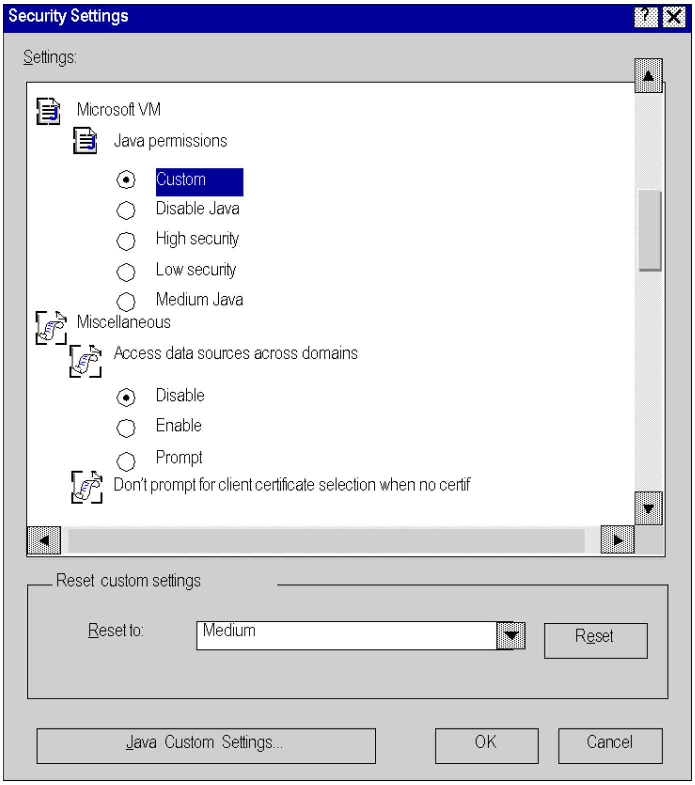
Select a Web content zone to specify its security setings.

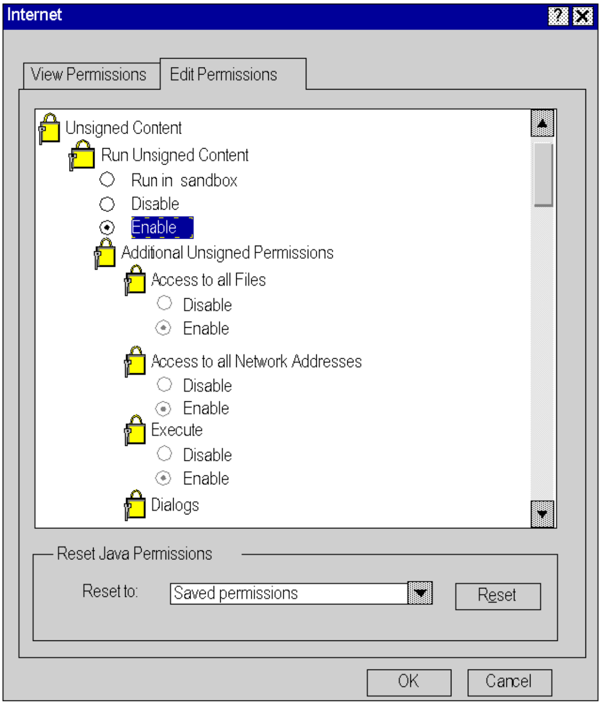
This Zone contains all Web sites you haven't placed in other zones

Custom settings.

-To change the settings, click Custom _evel.

-To use the recommended settings, click Default Level.
</td></tr>
<tr><td>4</td><td>Select the <b>Custom Level</b> button in the <b>Security Level</b> settings section.</td></tr>
</table>

Step	Action
5	Select the **Settings** button. **Result:** The Security Settings dialog appears.  Security Settings Settings: Microsoft VM Java permissions ● Custom ○ Disable Java ○ High security ○ Low security ○ Medium Java Miscellaneous Access data sources across domains ● Disable ○ Enable ○ Prompt Don't prompt for client certificate selection when no certif Reset custom settings Reset to: Medium Reset Java Custom Settings... OK Cancel
6	Scroll down until **Microsoft VM** is found under **Java Permissions**. Click the **Custom** radio button. Result: The **Java Custom Settings** button appears.

Step	Action
7	Select the **Java Custom Settings** button. **Result:** The Internet Zone dialog appears.  Internet View Permissions Edit Permissions Unsigned Content Run Unsigned Content ○ Run in sandbox ○ Disable ● Enable Additional Unsigned Permissions Access to all Files ○ Disable ● Enable Access to all Network Addresses ○ Disable ● Enable Execute ○ Disable ● Enable Dialogs Reset Java Permissions Reset to: Saved permissions Reset OK Cancel
8	Select the **Edit Permissions** tab.
9	Under the **Run Unsigned | Content**, select **Enable**.
10	Click **OK**.

Browser Security Considerations

Overview

Depending on your settings, both Mozilla Firefox and Internet Explorer remember a user name and password once entered for a Web site.

We Recommend

Close the browser after each session so that unauthorized people do not use your passwords to gain read or write access to your site.

If you are using Internet Explorer, you may also have to change your security settings using the custom settings option. Under the options for User Authentication, select "Prompt for user name and password."

Index