FactoryCast User Guide

For Quantum and Premium

8/2012

31001229.13

www.schneider-electric.com

Schneider Electric

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 that is contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

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.

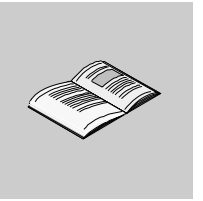
Failure to observe this information can result in injury or equipment damage.

Registered Trademarks

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

© 2012 Schneider Electric. All rights reserved.

Table of Contents



Safety Information **9**
About the Book **11**
Chapter 1 Introduction to FactoryCast **13**
About FactoryCast 14
FactoryCast components 15
FactoryCast Server 16
Web Designer for FactoryCast 17
FactoryCast Client 18
Variable Consistency Management 19
Installation 22
Chapter 2 Web Site Security **27**
Internal Security 28
External Security 29
Variable Access Security, Symbol, Direct Address 31
Chapter 3 Default Web Site for Quantum **33**
3.1 Quantum Home Page 34
Quantum Home Page 34
3.2 Monitoring Page 36
Monitoring Home Page 36
3.3 Diagnostics Page 38
Diagnostics Home Page 39
Rack Viewer Page 40
Navigating the Rack Viewer Silverlight Page 46
Properties Page 48
3.4 Module Diagnostics (Silverlight) 49
Quantum Controller Status 50
Quantum Ethernet Module Statistics 52
Remote I/O Communication Status (Head Module) 53
Remote I/O (Drop Adapter) 55
Discrete I/O Modules 58
Analog I/O Modules 60

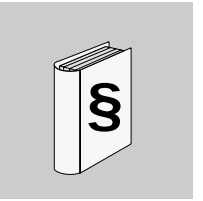
3.5 I/O Diagnostics (Java) 62
Controller Status Page 63
Ethernet Module Diagnostic Pages 64
Remote I/O Status 71
Configured Remote I/O Page 72
Remote I/O Drop Pages 74
Remote I/O Drop Module Pages 76
Distributed I/O Drops Page 78
Distributed I/O Specific Drop Page 79
Distributed I/O Module Page 81
3.6 Setup Page 83
Setup Home Page 84
SNMP Configuration 86
Configuring Faulty Device Replacement 89
Global Data (Publish/Subscribe) Utility Configuration 93
Changing the HTTP and Write Passwords 96
Changing the FTP Password 99
Chapter 4 Default Web Site for Premium 101
4.1 Premium Home Page 102
Premium Home Page 102
4.2 Monitoring Page 104
Monitoring Home Page 104
4.3 Diagnostics Page 106
Diagnostics Home Page 107
Rack Viewer Page 108
Properties Page 113
4.4 Module Diagnostics (Silverlight) 114
Premium Controller Status 115
Premium CPU Serial Port Status 117
CPU PCMCIA Card Status 119
Premium Ethernet Communication Module Status 123
Discrete I/O Modules 124
Analog I/O Modules 127
AS-i Module Diagnostics Page 128
Fipio Diagnostics Pages 129
4.5 Module Diagnostics (Java) 133
PLC Personality Page 134
Ethernet Module Statistics Page 136
Fipio Module Diagnostics Page 141
Digital I/O Module Diagnostics Page 147
Analog I/O Module Diagnostics Page 149
AS-i Module Diagnostics Page 151
PCMCIA Communication Card Diagnostics Page 153
Standard Module Diagnostics Page 155

4.6 Setup Page 157
Setup Home Page 158
Address Server Page for the HTTP Server 159
Changing the HTTP and Write Passwords 161
Changing the FTP Password 164
Chapter 5 Common Quantum and Premium Setup Page 167
Mail Service Configuration 168
Configuring the Time Synchronization Service 170
Chapter 6 Editors 175
6.1 Data Editor 176
Data Editor 177
Data Editor Spreadsheet 179
Creating a Data Template 185
Saving a Data Template 186
Using an Existing Data Template 187
Inserting a Symbol (Variable) in a Data Template 188
Inserting a Direct Address in a Data Template 190
Modifying Data Values in a Data Template 191
6.2 Data Editor Lite 193
Data Editor Lite 193
6.3 Pocket Data Editor 195
Pocket Data Editor 196
Using Pocket Data Editor 198
6.4 Graphic Editor 204
Graphic Editor Overview 205
Top Window User Functions 208
Display Window User Functions 213
Property Sheet 216
Security 218
Graphic Editor Applet Parameters 219
Graphic Objects 221
Extended Graphic Objects 239
6.5 Graphic Viewer 249
Graphic Viewer 249
6.6 PLC Program Viewer 251
PLC Program Viewer 251
Chapter 7 Adding Custom Pages to the Site 259
7.1 Working with Custom Pages 260
Downloading a Custom Home Page 261
Downloading Supporting Pages 263

7.2 Using FactoryCast's Java Applets Using HTML Code 264
Inserting Applets on a Web Page 265
Inserting LiveBeanApplet 266
Inserting LiveBeanApplet using HTML Code 269
Inserting LiveLabelApplet using HTML Code 272
Inserting LiveTableApplet using HTML Code 277
7.3 Using FactoryCast Extension for Microsoft FrontPage or Expression Web 279
Installing FactoryCast Extension for Microsoft Expression Web 280
Installing FactoryCast Extension for Microsoft FrontPage 284
Inserting LiveBeanApplet Using FrontPage or Expression Web 290
Inserting LiveLabelApplet Using FrontPage and Expression Web 298
Inserting LiveTableApplet Using FrontPage or Expression Web 302
7.4 Using Graphic Objects Lite 306
Downloading the Graphic Objects Lite Library 307
Description of Graphic Objects Lite 308
7.5 Adding Microsoft Silverlight® Pages 325
Adding Microsoft Silverlight® Pages 325
Chapter 8 Alarm Viewer 327
Display 328
Operation and Management of Alarms 331
Limitations 332
Chapter 9 SOAP Web Services 333
Principles for Designing a SOAP Client Interface 335
ModBusXMLDA SOAP Services 336
SymbolicXMLDA SOAP Services 347
ExtendedSymbolicXMLDA SOAP Services 353
How to optimize massive symbolic requests 360
Visual Basic Examples 363
Online SOAP Documentation 364
Appendices 365
Appendix A Browser Requirements, Settings, and Security Considerations 367
Browser Version 368
Browser Settings 370
Browser Security Considerations 373
Appendix B Performance Benchmarks 375
B.1 Premium Performance Benchmarks 376
Premium Performance Benchmarks 377
Monitoring Applet Performance on ETY5103 379
SOAP Performance Benchmarks 381

B.2 Quantum Performance Benchmarks 382
Quantum Performance Benchmarks 383
Performance on NOE77111 384
SOAP Performance Benchmarks 386
Index 387

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, 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 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 an imminently hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

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

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

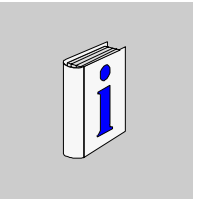
NOTICE
NOTICE is used to address practices not related to physical injury.

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.

About the Book



At a Glance

Document Scope

This user guide introduces the FactoryCast 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 Quantum or Premium programmable logic controller (PLC). FactoryCast provides the Web pages and Java applets needed to view run-time data from a controller. FactoryCast offers two 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

The present documentation is intended for qualified technical personnel responsible for the implementation, operation and maintenance of the products described. It contains information necessary for the proper use of the products. However, those who wish to make a more "advanced" use of our products may find it necessary to consult our nearest distributor in order to obtain additional information.

The technical characteristics of the devices described in this manual also appear online. To access this information online:

Step	Action
1	Go to the Schneider Electric home page *www.schneider-electric.com*.
2	In the **Search** box type the reference of a product or the name of a product range. ● Do not include blank spaces in the model number/product range. ● To get information on a grouping similar modules, use asterisks (*).
3	If you entered a reference, go to the **Product datasheets** search results and click on the reference that interests you. If you entered the name of a product range, go to the **Product Ranges** search results and click on the product range that interests you.
4	If more than one reference appears in the **Products** search results, click on the reference that interests you.
5	Depending on the size of your screen, you maybe need to scroll down to see the data sheet.
6	To save or print a data sheet as a .pdf file, click **Download XXX product datasheet**.

The characteristics that are presented in this manual should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the manual and online information, use the online information as your reference.

Related Documents

Title of Documentation	Reference Number
Web Designer Objects Library for Microsoft Expression Blend®	S1A75812
Quantum NOE 771 xx Ethernet Modules User Guide	840USE11600
PL7 Junior/Pro Setup Diagnostic Functions	TLXDSDIAGPL7
Web Designer for Modicon M340, Premium and Quantum	35016149

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

User Comments

We welcome your comments about this document. You can reach us by e-mail at techcomm@schneider-electric.com.

Introduction to FactoryCast

1

Purpose

This chapter describes FactoryCast for Quantum and Premium, including its functions, components and system requirements.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
About FactoryCast	14
FactoryCast components	15
FactoryCast Server	16
Web Designer for FactoryCast	17
FactoryCast Client	18
Variable Consistency Management	19
Installation	22

About FactoryCast

Overview

FactoryCast is a software package that you use to customize a Web site on an embedded Web server module. The site can be accessed via a browser to view and modify data from a Quantum or Premium 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 three FactoryCast components are:

- FactoryCast Server
- Web Designer for FactoryCast
- FactoryCast Client

Quantum Version

For Quantum, the FactoryCast components are integrated as follows.

This Component...	Is Used To...
140 NOE 771 11	Store the Web site on its embedded server.
Concept v. 2.1 or higher Modsoft v. 2.5 and higher Unity Pro v 4.0 and higher	Program the controller.
Web Designer for FactoryCast	Configure the Web site, download data and maintain the site on the server.
Web browser (IE, Firefox...)	View the Web site.

Premium Version

For Premium, the FactoryCast components are integrated as follows.

This Component...	Is Used To...
TSX ETY 5103	Store the Web site on its embedded server.
PL7 v. 3.0 and higher Unity Pro v 4.0 and higher	Program the controller.
Web Designer for FactoryCast	Configure the Web site, download data and maintain the site on the server.
Web browser (IE, Firefox...)	View the Web site.

Required Browser Plug-Ins

FactoryCast requires the installation of the following software:

- Sun Java Virtual Machine (JVM) version 1.6 or higher. 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 Quantum or Premium 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.

Versions

FactoryCast server is embedded on the following PLC modules.

Part Number	Description	Memory Available for Customization
140 NOE771 11	Quantum FactoryCast module with Ethernet TCP/IP 10/100 BaseT twisted pair/100 Base Fx fiber optic, 1 channel	up to 8 Mb
TSX ETY 5103	Premium Web Server Module with Fast Ethernet 10/100 BaseT Twisted Pair	up to 8 Mb

Web Designer for FactoryCast

Overview

You can use Web Designer for FactoryCast 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 FactoryCast is to provide general FTP capabilities enabling you to load custom Web pages to the FactoryCast module.

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

Configuring a Site

Web Designer for FactoryCast 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 Concept, PL7 or Unity Pro 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 FactoryCast enables you to backup files, restore files and, if necessary, reflash files to the embedded server.

FactoryCast Client

Overview

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

- diagnostic tools:
  - a rack viewer
  - an alarm viewer
- monitoring tools:
  - a data editor
  - a graphic editor and a graphic viewer
  - a PLC program viewer

Rack Viewer

The Rack Viewer allows you to display the status and configuration of the controller, embedded server module, other option modules and I/O modules.

Alarm Viewer

When the PLC application has diagnostic properties activated, the Alarm Viewer allows you to display application event messages.

Data Editor

The Data Editor allows you to view and modify variables (symbols and direct addresses).

Graphic Editor and Viewer

The Graphic Editor allows you to create and view graphical objects. Each graphical object can be linked to a variable or address in the embedded server. The Graphic Viewer lets you monitor the graphic page you created with the Graphic Editor.

PLC Program Viewer

Use the PLC program Viewer to monitor the PCL program during run time.

Variable Consistency Management

Overview

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

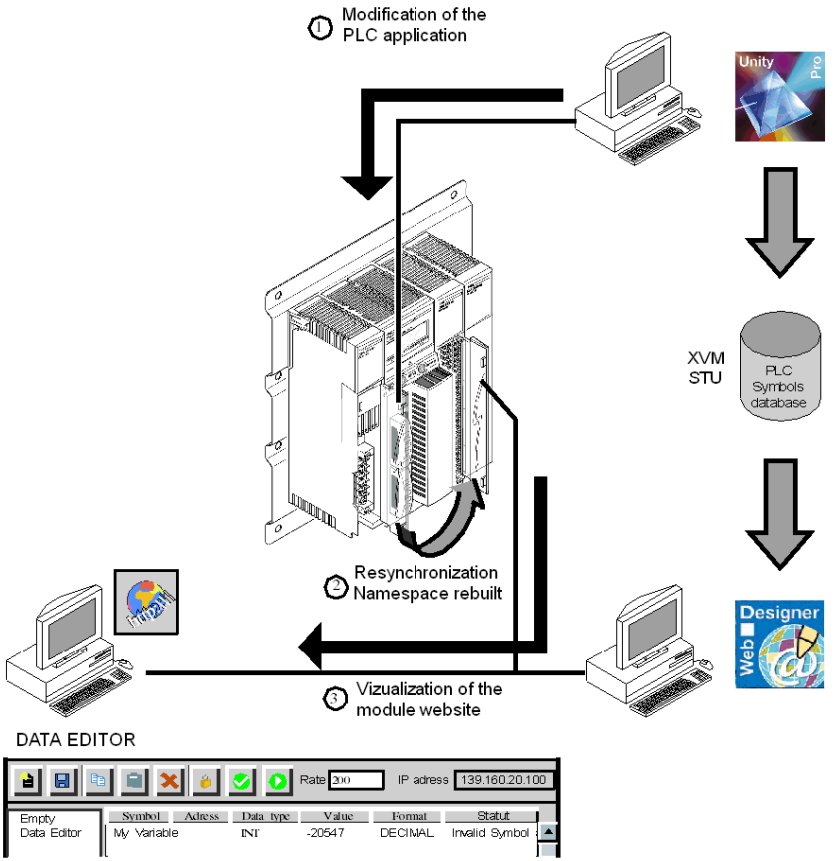
This can lead to inconsistency between the variables exchanged with the PLC and the variables of the PLC application (i.e. modification of the Unity Pro application). In this case, the real-time communication between the Web page and the PLC is stopped.

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 Unity Pro.

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.

Synchronization Process

The following figure shows you the synchronization process:



The following procedure assumes you already have:

- transferred your UnityPro application into the PLC with the Data Dictionary feature enabled (refer to the following note).
- imported the symbols from your UnityPro application to Web Designer for FactoryCast.
- transferred the Web Designer for FactoryCast project to the FactoryCast module.

NOTE: Enable the Data Dictionary feature in your UnityPro project in order to use automatic synchronization. This feature increases the size of your application. Refer to the Memory Usage page to know the size of your application.

1. Modification of the UnityPro application that runs on the PLC (UnityPro 4.0 or higher).
2. The FactoryCast module database synchronizes automatically with the PLC database using the data dictionary feature of UnityPro.
3. Visualization of the Web pages (Web browser).

NOTE:

- Variable Consistency Management operates between the PLC and the FactoryCast module. Therefore, you need to synchronize manually the Web Designer for FactoryCast 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.

Installation

Overview

FactoryCast comes on a CD and runs automatically. Once the CD has loaded on your PC, follow the installation dialogs.

Quantum Downloaded Files before V3.1

FactoryCast downloads the following files for Quantum modules during installation.

Rack Viewer	Data Editor	Graphic Editor	Real-Time Communication
wwwroot/classes » Sys Diag.jar wwwroot/conf » Gcnftcop.sys wwwroot/images » module.gif » miniplc.gif » eight_io.gif » empty.gif » hiendcpu.gif secure/system » ctrlstat.htm » ethernet.htm » plccfg.htm » riostat.htm	wwwroot/classes » RDE.jar secure/system » rde.htm	wwwroot/classes » GDE.jar » Widgets.jar secure/system » gde.htm wwwroot/images » key.gif	wwwroot/classes » SAComm.jar

Quantum Downloaded Files in V3.1 (only for NOE77111 V3.1 and Higher)

FactoryCast downloads the following files for Quantum modules during installation.

Rack Viewer	Data Editor	Graphic Editor	Real-Time Communication
wwwroot/classes » SysDiag.jar wwwroot/conf » Gcnftcop.sys wwwroot/images » module.gif » miniplc.gif »eight_io.gif » empty.gif »hiendcpu.gif wwwroot/secure/system » ctrlstat.htm » ethernet.htm » plccfg.htm » riostat.htm	wwwroot/classes » RDE.jar secure/system » rde.htm	wwwroot/classes » GDE.jar » Widgets.jar » user.jar wwwroot/secure/system » gde.htm wwwroot/images » key.gif	wwwroot/classes » SAComm.jar

Graphic Viewer	Data Editor Lite	Alarm Viewer	Pocket Data Editor
wwwroot/classes » GDV.jar » Widgets.jar » user.jarw wwroot/secure/system » gdv.htm wwwroot/images » key.gif	wwwroot/classes » rdelite.jar wwwroot/secure/system » rdelite.htm	wwwroot/classes » RAE.jar wwwroot/secure/system » rae.htm	wwwroot/cabs » RdePocket.cab wwwroot/secure/system » rdepocket.htm

Premium Downloaded Files before V3.1

FactoryCast downloads the following files for Premium modules during installation.

Rack Viewer	Data Editor	Graphic Editor	Alarm Viewer	Real-Time Communication
wwwroot/classes » JL.jar » SysDiag.jar wwwroot/conf » business.pbf » modules.pbf » products.pbf » ranges.pbf wwwroot/images » anim1.gif » backbtn.gif secure/system » plccfg.htm » ethernet.htm	wwwroot/classes » RDE.jar secure/system » rde.htm	wwwroot/classes » GDE.jar » Widgets.jar secure/system » gde.htm wwwroot/images » key.gif	wwwroot/classes » RAE.jar secure/system » rae.htm wwwroot/images » ack.gif » ackall.gif » dfb.gif » gr7.gif » help.gif » nack.gif » stop.gif » sys.gif » trasall.gif » trash.gif » asi.gif	wwwroot/classes » SAComm.jar

Premium Downloaded Files in V3.1 (only for ETY5103 V3.1 and Higher)

FactoryCast downloads the following files for Premium modules during installation.

Rack Viewer	Data Editor	Graphic Editor	Alarm Viewer	Real-Time Communication
wwwroot/classes » JL.jar » SysDiag.jar wwwroot/conf » business.pbf » modules.pbf » products.pbf » ranges.pbf wwwroot/images » anim1.gif » backbtn.gif secure/system » plccfg.htm » ethernet.htm	wwwroot/classes » RDE.jar secure/system » rde.htm	wwwroot/classes » GDE.jar » Widgets.jar » user.jar wwwroot/secure/system » gde.htm wwwroot/images» key.gif	wwwroot/classes » RAE.jar wwwroot/secure/system » rae.htm	wwwroot/classes » SAComm.jar

Graphic Viewer	Data Editor Lite	Pocket Data Editor
wwwroot/classes » GDV.jar » Widgets.jar » user.jar wwwroot/secure/system » gdv.htm wwwroot/images »key.gif	wwwroot/classes » rdelite.jar wwwroot/secure/system » rdelite.htm	wwwroot/cabs » RdePocket.cab wwwroot/secure/system » rdepocket.htm

Web Site Security

2

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:

Topic	Page
Internal Security	28
External Security	29
Variable Access Security, Symbol, Direct Address	31

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.

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.

External Security

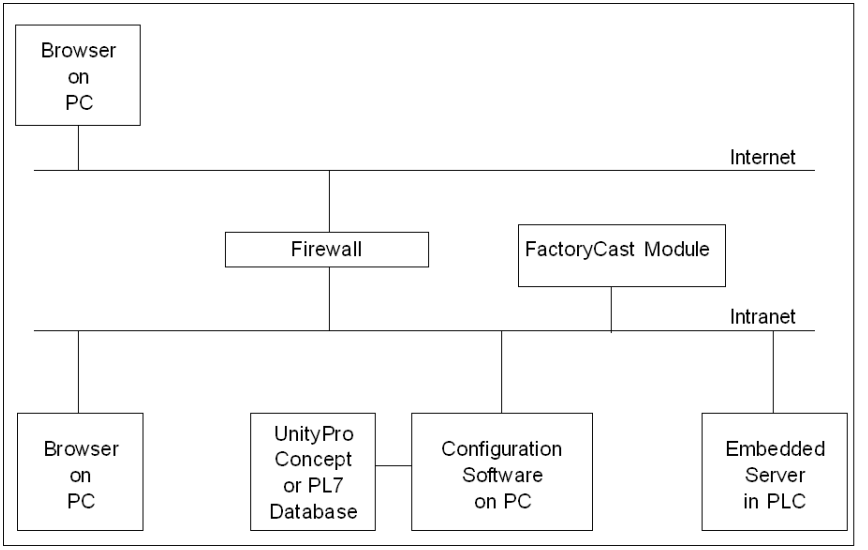
Overview

If your network has been configured to let users consult your Internet site, your security system is similar to an intranet site, only you have an additional security measure: a firewall.

Architecture of a Firewall

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

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 network.

Application-Level Firewalls

An application firewall works for an application, for example FTP. It intercepts 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 FactoryCast uses HTTP, FTP and 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 behind a firewall, you need to configure the firewall to authorize HTTP, FTP and MBAP traffic.

Port	Protocol	Access to...
21	FTP	Embedded server files (behind firewall)
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 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.

Default Web Site for Quantum

3

Overview

When you receive the Embedded Server, it already contains a default Web site with diagnostic pages, Data Editor, and Graphic Editor.

You may view these pages and view direct addresses in the editors simply by installing the module and configuring its IP address. To access the site, type the module's IP address in your browser and enter the default user name and password of *USER*.

This chapter describes the pages in the default Web site.

What Is in This Chapter?

This chapter contains the following sections:

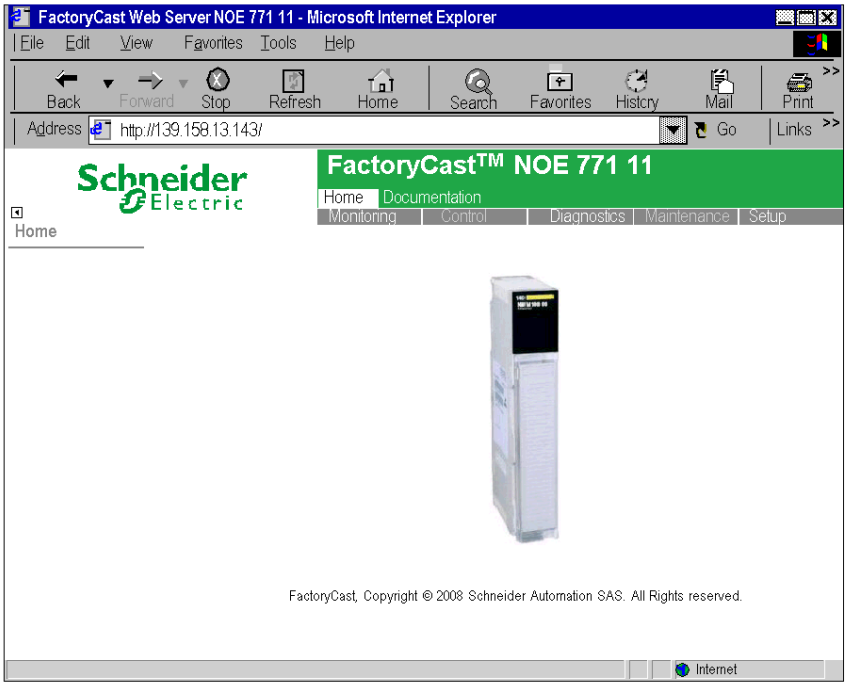
Section	Topic	Page
3.1	Quantum Home Page	34
3.2	Monitoring Page	36
3.3	Diagnostics Page	38
3.4	Module Diagnostics (Silverlight)	49
3.5	I/O Diagnostics (Java)	62
3.6	Setup Page	83

3.1 Quantum Home Page

Quantum Home Page

Home Page

To access the Quantum home page enter the IP address of the module in his web browser. No password is required to display the home page:



Links

From the Quantum home page, you can access the following pages:

- Monitoring *(see page 36)*
- Diagnostics *(see page 39)*
- Setup *(see page 84)*
- Documentation

Enter a user name and a password to access the services on these pages.

Language

In the Quantum home page, you can display the Quantum web pages in either English or Chinese.

3.2 Monitoring Page

Monitoring Home Page

Home 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** home page looks like this:



To access a service, click on a link. The services include:

- Data editor *(see page 176)*: for creating variable data tables to determine their values when the table is animated.
- Data editor lite *(see page 193)*: for creating variable data tables to determine their values when the table is animated. (This editor contains fewer features than the standard Data editor.)
- Graphic editor *(see page 204)*: for creating graphics to determine the values of variables when the graphic is animated.
- Graphic viewer *(see page 249)*: for viewing graphics to determine the values of variables when the graphic is animated.
- PLC Program viewer: Display UnityPro programs in run mode using a web browser.
- Password-protected custom pages: for restricting access to web pages created by the user.
- Custom pages without password protection: for unrestricted access to web pages created by the user.
- Silverlight pages *(see page 325)*: for viewing user-defined pages created with *Microsoft Expression Blend®* software that have been added to the web site

3.3 Diagnostics Page

Overview

This section describes the different services proposed by the **Diagnostics** page.

What Is in This Section?

This section contains the following topics:

Topic	Page
Diagnostics Home Page	39
Rack Viewer Page	40
Navigating the Rack Viewer Silverlight Page	46
Properties Page	48

Diagnostics Home Page

Home Page

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

Illustration

The **Diagnostics** home page looks like this:



Links

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

- Rack Viewer *(see page 40)*
- Controller Status *(see page 63)*
- RIO Status *(see page 71)*
- Alarm Viewer *(see page 327)*
- Ethernet *(see page 64)*
- Properties *(see page 48)*

Rack Viewer Page

Overview

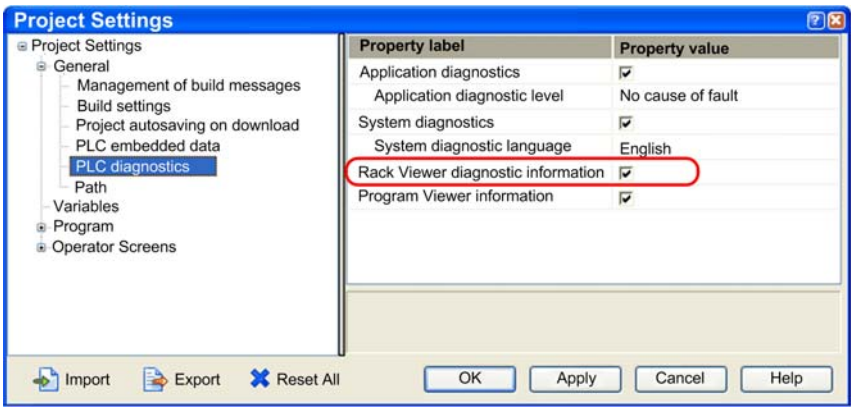
The **Rack Viewer** page displays the current configuration of your network, as either a Silverlight or a Java page. The technology used to display your network depends on both:

- how you configured your Unity pro 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 Unity Pro, 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 diagnostic information** as displayed below, then click **OK**. **NOTE:** The **Rack Viewer diagnostic information** selection is set by default.



NOTE: If you do not select **Rack Viewer diagnostic information**, as described above:

- the Java version of the **Rack Viewer** page displays
- Ethernet remote I/O drops are not displayed

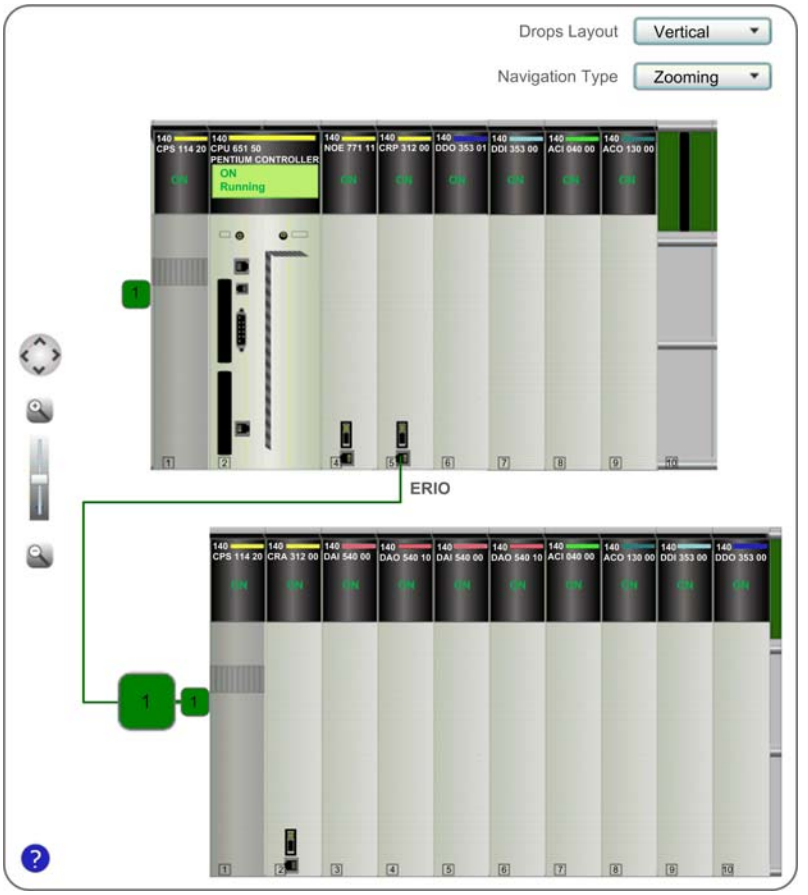
Browser Supporting Software

If you selected **Rack Viewer diagnostic information**, above, the **Rack Viewer** display depends on the browser-supporting software installed on your PC.

If your PC has installed on it:	The Rack Viewer page displays:
• Java® Standard Edition software • Microsoft Silverlight® 4 plug-in	A Silverlight page containing the entire configuration of the network, including both the local rack plus up to 31 remote racks.
Java Standard Edition software (without the Microsoft Silverlight 4 plug-in)	A Java page containing the current configuration of the local rack.

Sample Silverlight Pages

Here is an example of the **Rack Viewer** page presented in Silverlight, for a network with a local rack and a single remote rack:



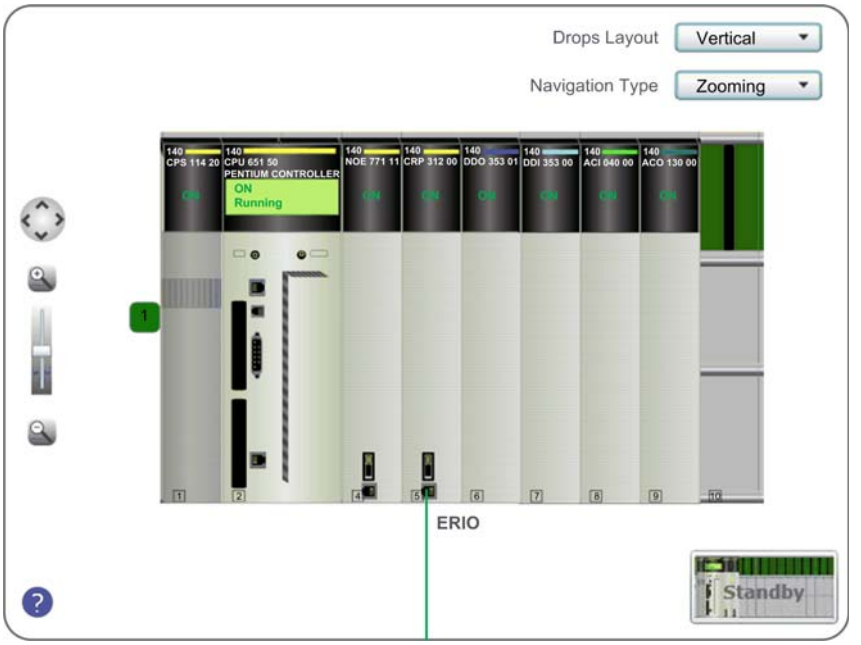
The Silverlight version of the **Rack Viewer** presents the configured network modules in one screen. In the above example, the network consists of a local rack and one remote rack.

Each 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 module to open a pop-up page with detailed information about that module.

If your network consists of both a primary local rack and a secondary hot standby local rack, the **Rack Viewer** page includes an icon—in the lower right corner of the page—that you can click to display the secondary hot standby rack:



Silverlight Page Diagnostics

Color indicators display the operational health of modules, racks, drops and the network bus.

Modules:

Each 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 module to open a pop-up page with detailed information about that module.

Racks and Drops:

The node next to each drop and rack indicates the health of that node:

- Green indicates every module in the rack or drop is functioning properly.
- Red indicates at least one module in the rack or drop is not functioning properly.

Bus:

The color of the line representing the network bus indicates the health of the bus:

- Green indicates every module on the bus is functioning properly.
- Red indicates at least one module on the bus is not functioning properly.

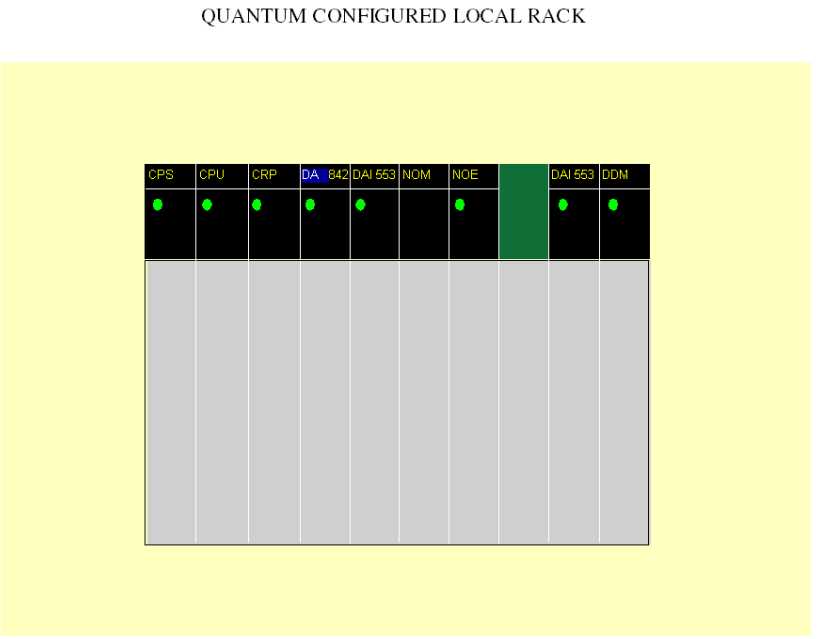
When you place your mouse pointer over a module, the **Rack Viewer** displays information describing the module and its location in the network.

Silverlight Page Navigation

For information on displaying and navigating the **Rack Viewer** page, refer to the topic Navigating the Rack Viewer Silverlight Page *(see page 46)*.

Sample Java Page

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



The Java version of the **Rack Viewer** displays the local rack, with each module appearing in its configured slot in the rack, along with the following information:

- A label at the top of the module tells what type it is. Question marks indicate that the module type is unknown or the slot is empty.
- An LED below the label reports the `Health Bit` status for the module:
  - Green indicates that the module is functioning properly.
  - Red indicates that the module is not functioning properly.

When you click on a CRP head module or an NOE module in the local rack, you begin the process of drilling-down through a series of Web pages displaying properties for the connected racks and modules.

When you click on an input or output module, a window opens displaying detailed information about that module.

Navigating the Rack Viewer Silverlight Page

Overview

Use the controls on the **Rack Viewer** page to specify your preferred layout of drops and modules, and your intended method of navigating around the page.

Selecting a Layout

Use the **Drops Layout** setting to select the order in which the racks are displayed. if you select:

- **Horizontal**: the page displays each remote drop in a top-to-bottom order beneath the primary bus, with the lowest-numbered drop at the top.
- **Vertical**: the page displays each remote drop in a left-to-right order beneath the primary bus, with the lowest-numbered drop at the left.

The main and extension racks in the primary bus appear at the top of the layout, in a left-to-right sequence.

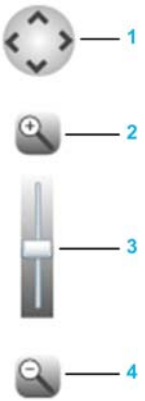
Selecting a Navigation Method

Use the **Navigation Type** setting to enable either:

- **ScrollBar**, then use the vertical and horizontal scrollbars to display racks and modules.
- **Zooming**, then use the zooming controls to navigate the **Rack Viewer** page.

Zooming

If you select **Zooming** as the **Navigation Type**, you can use your mouse, the keyboard, or the **Rack Viewer** navigation controls, depicted below:



1 Move up, down, left, and right arrows
2 Zoom in button
3 Zoom slider
4 Zoom out button

Navigate around the **Rack Viewer** page as follows:

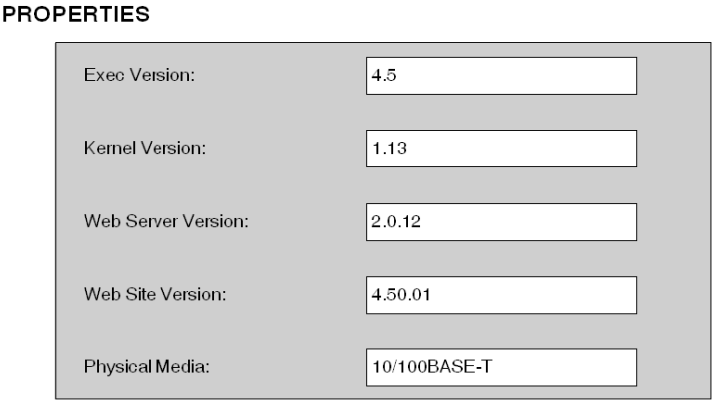
- Zoom In:
  - Mouse: scroll the mouse wheel forward
  - Keyboard: hold down the **Ctrl** key, then use the up or right arrow keys
  - Navigation panel: click the zoom in (2) button
  - Navigation panel: move the slider (3) up
- Zoom Out:
  - Mouse: scroll the mouse wheel backward
  - Keyboard: hold down the **Ctrl** key, then use the down or left arrow keys
  - Navigation panel: click the zoom out (4) button
  - Navigation panel: move the slider (3) down
- Move the **Rack Viewer** screen:
  - Mouse: hold down the left mouse button, then drag the screen
  - Keyboard: use the up, down, left or right arrow keys
  - Navigation panel: click the up, down, left or right (1) arrows

NOTE: If you select **Zooming** as the **Navigation Type**, you can place your mouse pointer over the help icon to view help related to zooming.

Properties Page

Introduction to the Properties Page

You can navigate to the **Properties** page from the Diagnostics page *(see page 38)*.The **Properties** page displays the versions of the Exec, Kernel, Web Server, Web Pages and the Physical Media:



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

3.4 Module Diagnostics (Silverlight)

Overview

Many Quantum FactoryCast modules present diagnostic web pages. Each module's web pages—and status—are displayed in the **Rack Viewer**.

What Is in This Section?

This section contains the following topics:

Topic	Page
Quantum Controller Status	50
Quantum Ethernet Module Statistics	52
Remote I/O Communication Status (Head Module)	53
Remote I/O (Drop Adapter)	55
Discrete I/O Modules	58
Analog I/O Modules	60

Quantum Controller Status

Overview

The **Quantum Controller Status** page is a read-only Silverlight page that presents data describing the controller's:

- operating status
- rack and slot location
- model number and platform
- configuration settings

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

To access this page, click on the CPU module in **Rack Viewer**.

Sample Pages

The **Quantum Controller Status** page looks like this, when your network does not include a second hot standby local rack:



NOTE: You can access the same—plus additional—data about the controller in the **Diagnostics** → **Controller Status** Java-based web page.

When your network includes both a primary local rack and a hot standby rack, the **Quantum Controller Status** page looks like this:



Quantum Ethernet Module Statistics

Overview

The **Quantum Ethernet Module Statistics** page presents data describing the Ethernet communication module's:

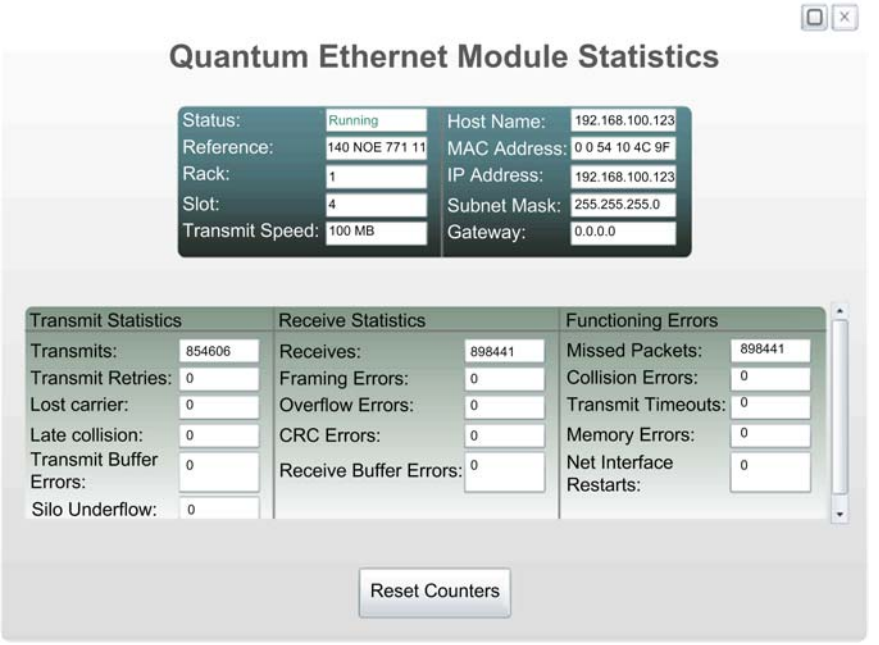
- operating status
- rack and slot location
- model number
- IP addressing settings
- Ethernet operational counting statistics

NOTE: For a description of the data presented for a specific Ethernet communication module, consult the user-documentation for that module.

To access this page, click on the Ethernet communication module in **Rack Viewer**.

Sample Page

The **Quantum Ethernet Module Statistics** page looks like this:



Click the **Reset Counters** button to reset the counting statistics—in the lower half of the page—to zero.

Remote I/O Communication Status (Head Module)

Overview

The **Remote I/O Communication Status** page is a read-only Silverlight page describing the remote I/O head module in the local rack. Remote I/O head modules are protocol-specific and can support either:

- Ethernet remote I/O networks
- S908 remote I/O networks

NOTE: For a description of a specific remote I/O head module's parameters, consult the user-documentation for that module.

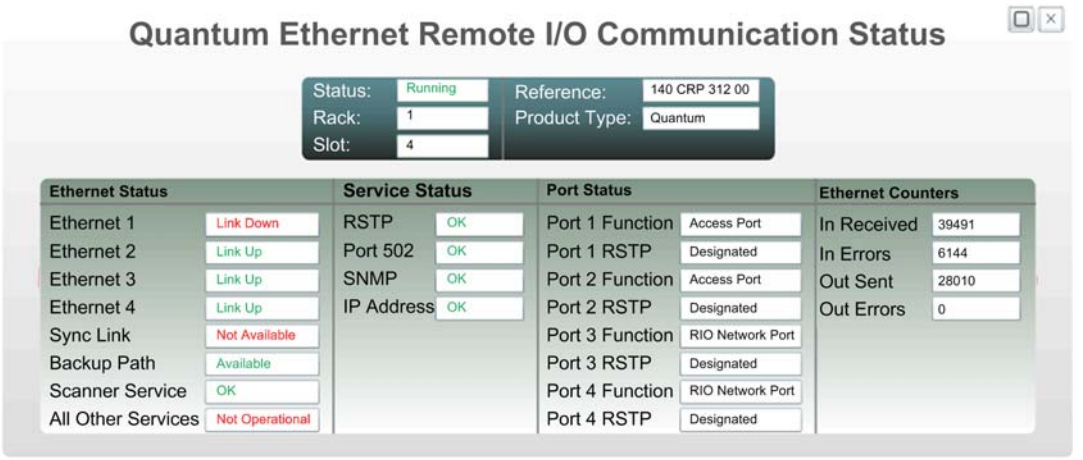
To access this page, click on the remote I/O head module in **Rack Viewer**.

Ethernet Remote I/O Communication Status

The **Ethernet Remote I/O Communication Status** page presents data describing the remote I/O head module, including:

- Rack and slot location
- Ethernet status
- Service status
- Port status
- Ethernet packet counters

The **Ethernet Remote I/O Communication Status** page looks like this:

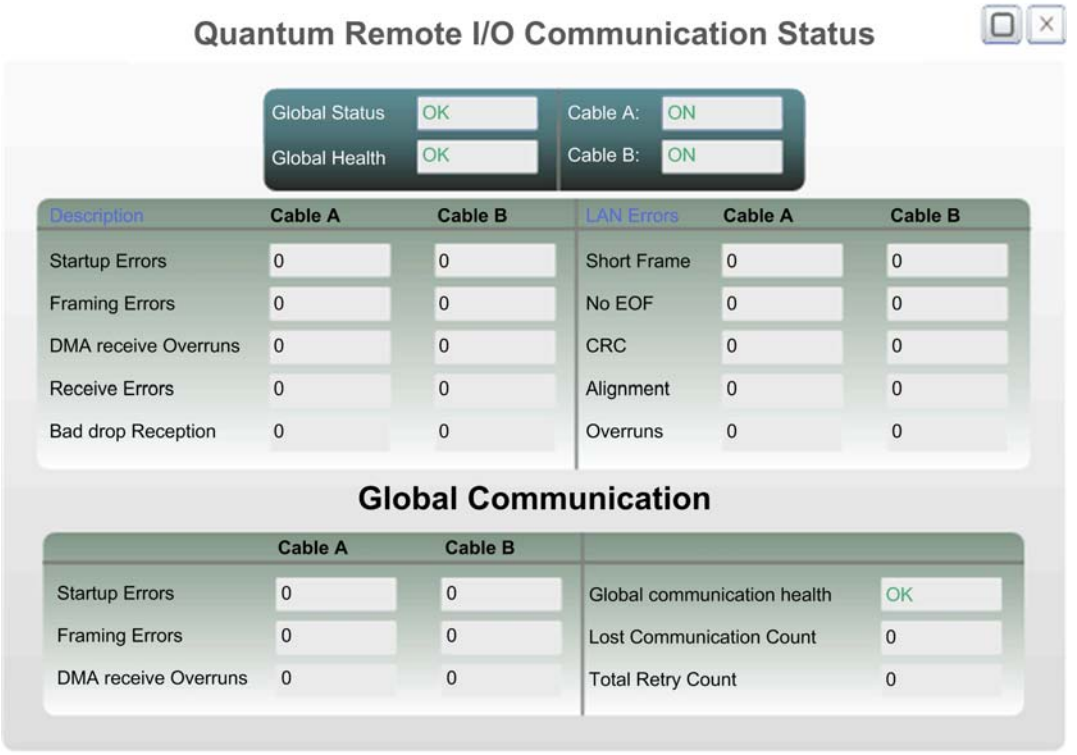


S908 Remote I/O Communication Status

The **S908 Remote I/O Communication Status** page presents data describing the remote I/O head module, including:

- Global status
- Global health status
- Port status (cables A and B)
- Module-specific detected error status
- Global detected error and communication health status

The **S908 Remote I/O Communication Status** page looks like this:



Remote I/O (Drop Adapter)

Overview

The **Remote I/O Communication Status** page is a read-only Silverlight page that presents data describing the remote I/O adapter module in the local rack. Remote I/O adapter modules are protocol-specific and can support either:

- Ethernet remote I/O networks
- S908 remote I/O networks

NOTE: For a description of a specific remote I/O drop adapter's parameters, consult the user-documentation for that module.

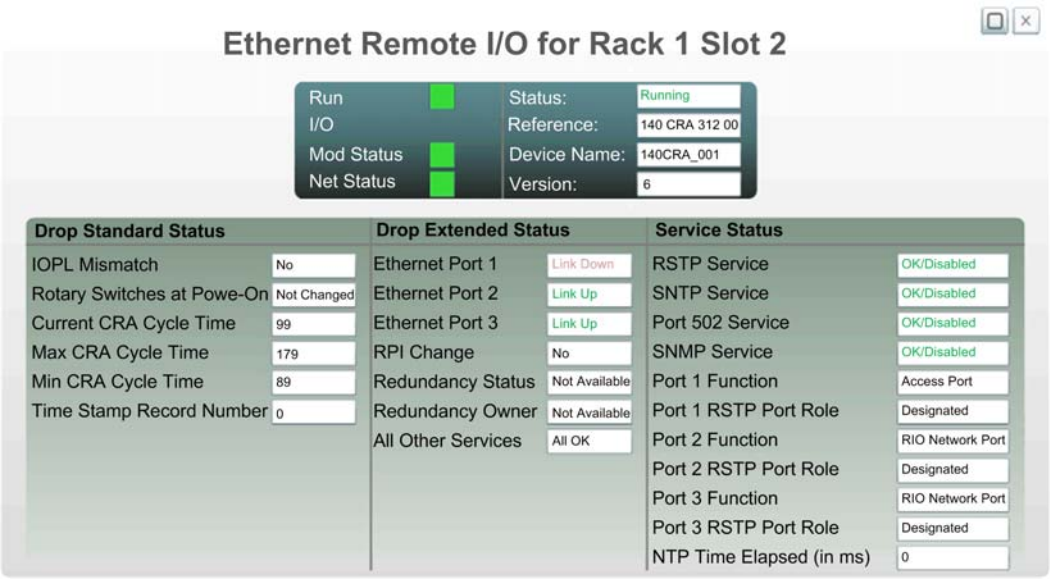
To access this page, click on the remote I/O drop adapter module in **Rack Viewer**.

Ethernet Remote I/O Communication Status

The **Ethernet Remote I/O** page presents data describing the remote I/O adapter, including:

- rack and slot location
- LEDs indicating:
  - Run status
  - I/O status
  - Module status
  - Network status
- operating status
- model number and platform
- Remote I/O drop statistics
- Ethernet services status

The **Ethernet Remote I/O** page looks like this:

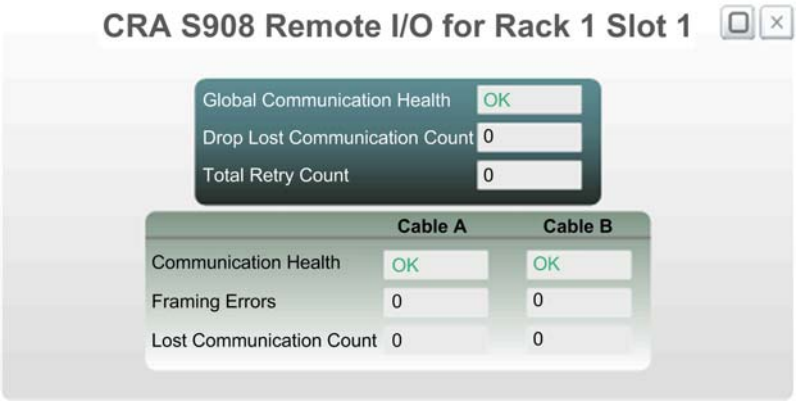


S908 Remote I/O Communication Status

The **S908 Remote I/O** page presents data describing the remote I/O adapter, including:

- rack and slot location
- for the drop:
  - communication health
  - lost communication count
  - total retry count
- for cables A and B:
  - communication health
  - detected framing errors
  - lost communication count

The **S908 Remote I/O** page looks like this:



Discrete I/O Modules

Overview

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

- the discrete input or output module type
- rack and slot location
- the range of bits or words assigned to the module
- the status of the inputs or outputs

The page displays I/O data as discrete points in an LED panel. When the LED is:

- green: the point is active
- off (or gray): the point is inactive

Tooltip Display

When you place your mouse pointer over an input or output channel (or point), a tooltip appears displaying the following information for that channel:

- variable name (if available)
- forced status (Forced or NotForced)
- operational status (ON or OFF)

The following is an example of a channel tooltip display:

Variable Name: VarCh1 /Forced Status: NotForced /Status: ON

Sample Discrete Input Module Page

The discrete input module, depicted below, displays the status of each bit comprising two words:



Each bit is active, as denoted by the green color of the bit number.

Sample Discrete Output Module Page

The discrete output module, depicted below, also displays the status of each bit in two words:



Each bit is active, as denoted by the green color of the bit number.

Analog I/O Modules

Overview

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 numeric status of each word

NOTE: Analog modules display data exclusively as words.

Tooltip Display

When you place your mouse pointer over an input or output channel (or point), a tooltip appears displaying the following information for that channel:

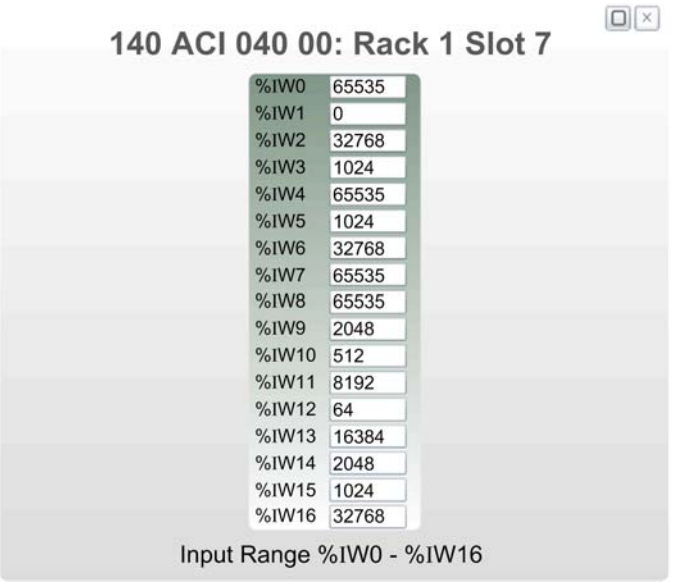
- variable name (if available)
- forced status (Forced or NotForced)
- operational status (ON or OFF)

The following is an example of a channel tooltip display:

Variable Name: VarCh1 /Forced Status: NotForced /Status: ON

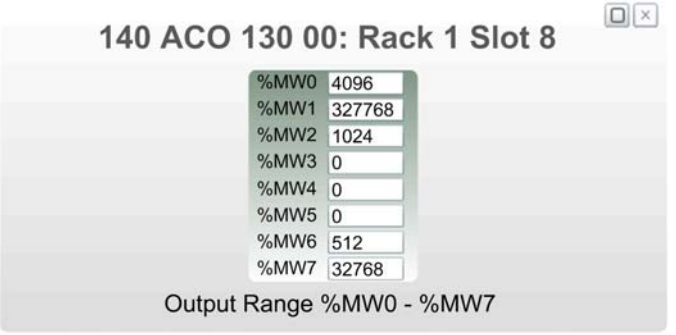
Sample Analog Input Module Page

An analog input module looks like this:



Sample Analog Output Module Page

An analog output module looks like this:



3.5 I/O Diagnostics (Java)

Overview

This section presents the Java version of diagnostic web pages. Your browser displays these pages if your application is configured to display the diagnostic pages in Java format *(see page 40)*, or if Silverlight supportive files are not installed on your PC.

Java-based Web pages include:
- Controller Status page
- Ethernet Module Diagnostic page
- Remote I/O Status page
- Configured I/O Remote page
- Remote I/O Drop pages
- Remote I/O Drop Module pages
- Distributed I/O Drops page
- Distributed I/O Specific Drop page
- Distributed I/O Module pages

Distributed I/O pages can be accessed by selecting either the CPU or a NOM from the **Rack Viewer** page assuming that Distributed I/O is configured in the controller.

What Is in This Section?

This section contains the following topics:

Topic	Page
Controller Status Page	63
Ethernet Module Diagnostic Pages	64
Remote I/O Status	71
Configured Remote I/O Page	72
Remote I/O Drop Pages	74
Remote I/O Drop Module Pages	76
Distributed I/O Drops Page	78
Distributed I/O Specific Drop Page	79
Distributed I/O Module Page	81

Controller Status Page

Overview

The **Controller Status Page** provides up-to-date information about the controller and its configuration. Access this page by selecting the CPU model from the Rack Viewer or the hyperlink Controller Status.

Sample Page

Here is an example of a **Controller Status** page.

QUANTUM CONTROLLER STATUS

Status:	Running	Reference:	CPU 534 14
Battery:	OK	Product Type:	Quantum
Rack:	1	Exec ID:	883
Slot:	2	Logged In:	No

Description		Registers		ASCII	
System Memory [Kb]	64 Kb	0xxxxx	000001-001536	Total Words	0
Extended Memory [Kb]	96 Kb	1xxxxx	100001-100512	Total Messages	0
Total Memory [Bytes]	163840	3xxxxx	300001-300512	Word Used	0
I/O Map Words	161	4xxxxx	400001-401872	Messages Used	0
Segments	32	6xxxxx	600000-	Available Words	0
DCP Drop ID	0	Battery Coil	0-----	Available Mes-	0
Memory Protect	Off	Timer Register	4-----	# ASCI Ports	0
Constant Sweep	Off	Time of Day	4-----	ASCII Inputs	4------
Optimize	No	Stopped Codes	0x0000	ASCII Outputs	4------

Dynamic Data

Some of the data provided on this page is dynamic. Dynamic data is constantly refreshed at a rate determined by the performance of the Embedded Server, network, and client CPU.

Ethernet Module Diagnostic Pages

Overview

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

- Global Data
- I/O Scanning
- Messaging
- Bandwidth Monitoring
- Statistics
- NTP (network time protocol)
- Email
- Upload MIB File
- Crash Log File

Click a link to access the desired diagnostics utility.

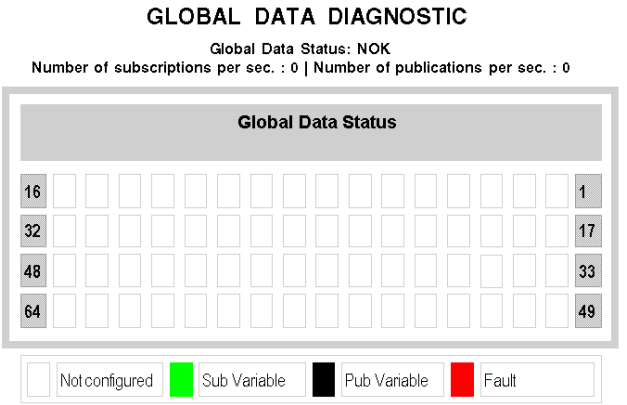
Global Data Page

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 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



I/O Scanning Page

General diagnostics for the I/O scanning service are shown at the top of this page:

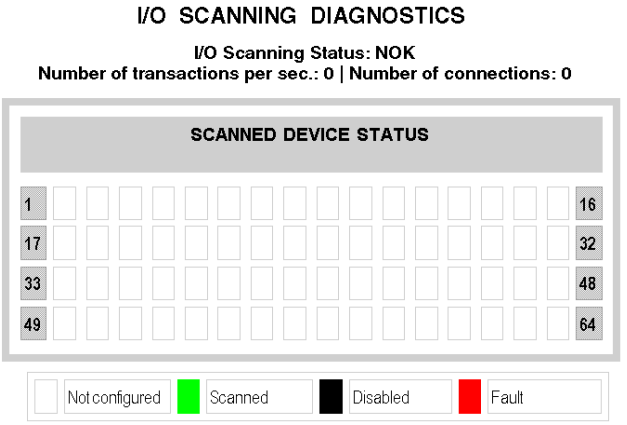
- the I/O scanning status
- the number of transactions per second
- the number of connections

A value of NOK in the **I/O Scanning Status** field indicates that the local system is not scanning. In this case, any data that appears in the **Scanned Device Status** display is meaningless.

A value of OK in the **I/O Scanning Status** field indicates that the values in the **Scanned Device Status** display are reporting the state of scanned devices.

In the **Scanned Device Status** display, the colors that appear in each block indicate the following states for specific remote devices:

- Green indicates that a device is being scanned
- Black indicates that I/O scanning of the specific device has been intentionally disabled via the Device Control Block
- White indicates an unconfigured device
- Red indicates a suspect device



NOTE: Be aware that the green **Scanning** indicator may incorrectly remain green for a remote scanned device after the Ethernet cable gets removed from that device. This inaccurate indication occurs when the health timeout value in the I/O Scanning configuration screen is set to 0.

You should configure an operational health timeout value in the range 1...65535 (in 1 ms increments). If the health timeout value is set to any value in this range, the **Scanning** indicator reports I/O scanning health correctly.

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):

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

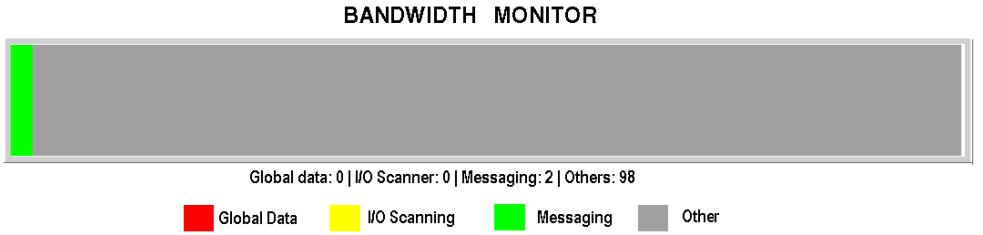
NOTE: Following a request to close a connection, the PLC may hold the connection open in its memory for a few minutes, during which the table will reflect the open connection.

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.

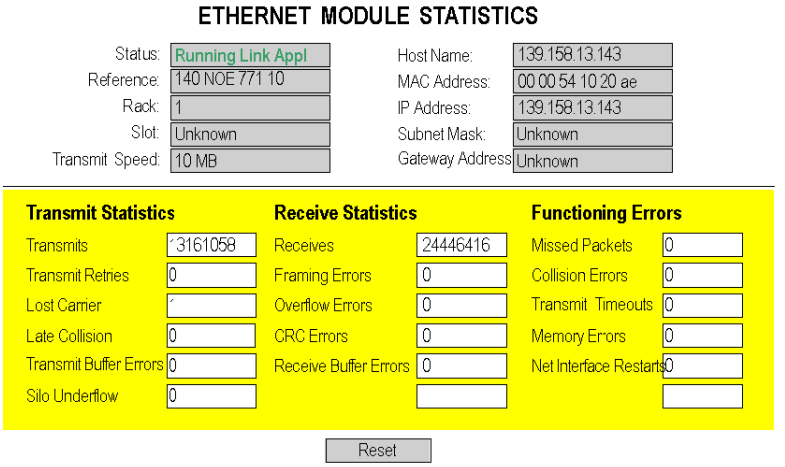
Bandwidth Monitoring Page

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



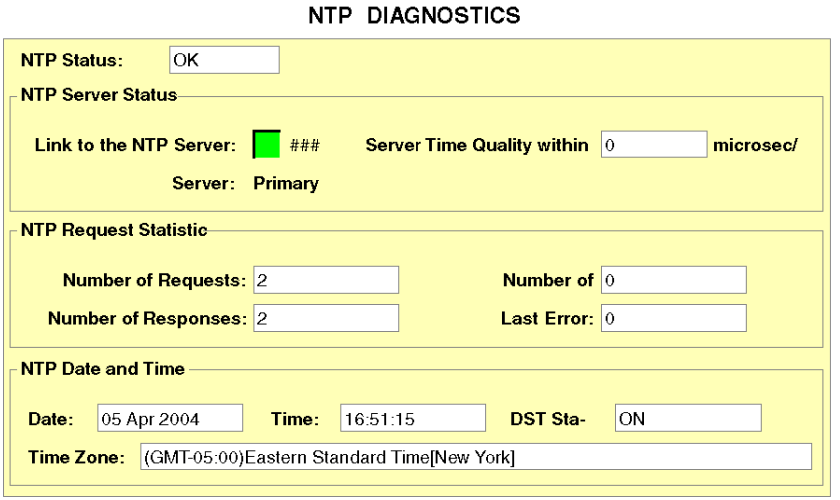
Ethernet Statistics Page

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 from the local rack or use the hyperlink Statistics. The following graphic is an example **Ethernet Module Statistics** page:



NOTE: Refer to *Modicon Quantum Ethernet TCP/IP Module User Guide* (840 USE 107) and *Modicon Quantum Ethernet TCP/IP Module User Guide* (840 USE 115) for definitions of terms.

NTP Diagnostics Page



Time synchronization service parameters:

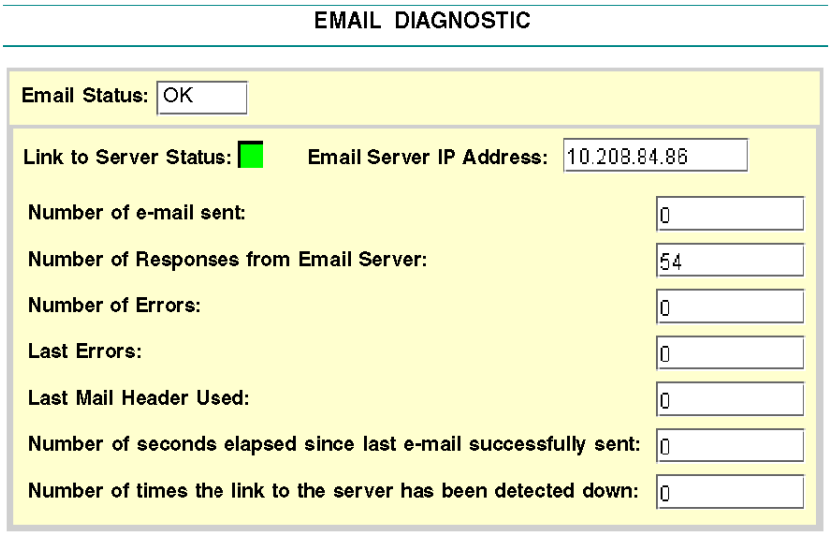
Parameter	Description
NTP status	Service is correctly configured (OK)
NTP server status	NTP client is connected to the NTP server, and if the server is Primary or Standby
NTP requests	Total number of client requests sent to the NTP server
NTP responses	Total number of server responses sent from the NTP 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

Type of detected event	Value
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

Email Diagnostics Page

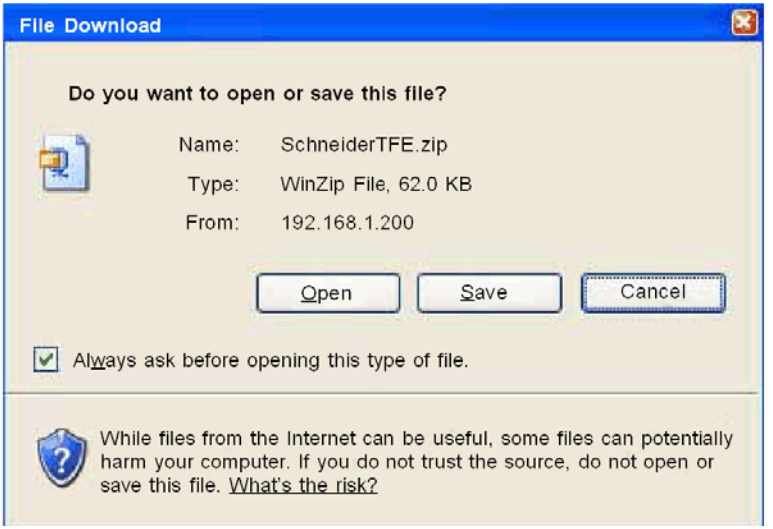


Parameter	Description
Email status	Email service is correctly configured (OK).
Link to Server Status	Ethernet module is connected to the SMTP server. Status is checked at start-up and at least every 30 minutes after start-up: ● Green = module connected to server ● Red = module NOT connected to server
SMTP Server IP Address	IP address of the SMTP server
Number of e-mails sent	Total number of e-mails sent successfully
Number of Responses from SMTP Server	Total number of SMTP messages received from the SMTP server
Number of Errors	Total number of e-mails not sent because of a detected error

Parameter	Description
Last Errors	Reason for the last detected error with a code in hexadecimal. 0 indicates no detected error.
Last Mail Header Used	Last header used by giving the number.
Number of seconds elapsed since last e-mail successfully sent	Counts the number of seconds since the last email was successfully sent.
Number of times the link to the server has been detected down	Number of times the SMTP server could not be reached. (Link checked every 30 minutes.)

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:



Crash Log Diagnostics Page

The **Crash Diagnostics** Page displays a crash log file when an unrecoverable event has occurred, and a status message otherwise.

Press **Clear Crash Log File** to clear the log file.

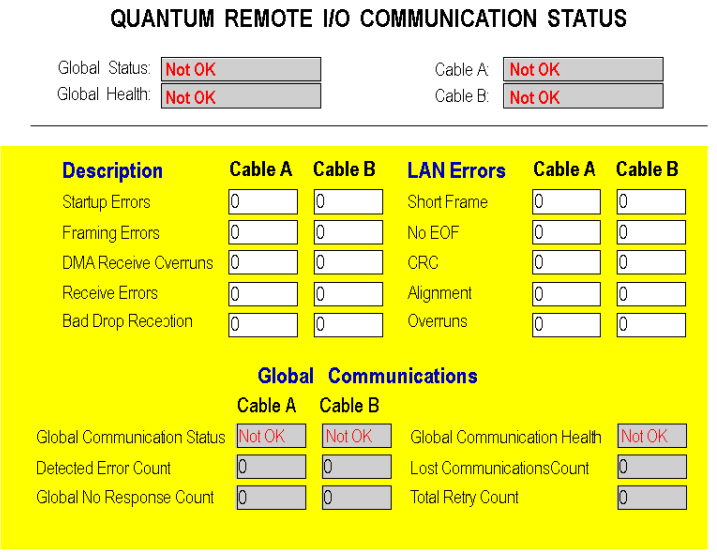
Remote I/O Status

Overview

The **Remote I/O Status** page gives an overview of the status and health of the Remote I/O network communications. Access this page by selecting the CRP Drop down menu item **Remote I/O Status** or using the hyperlink **RIO Status**.

Sample Page

Here is an example of a **Remote I/O Status** page.



Dynamic Data

Some of the data provided on this page is dynamic. Dynamic data is constantly refreshed at a rate determined by the performance of the embedded server, network, and client CPU.

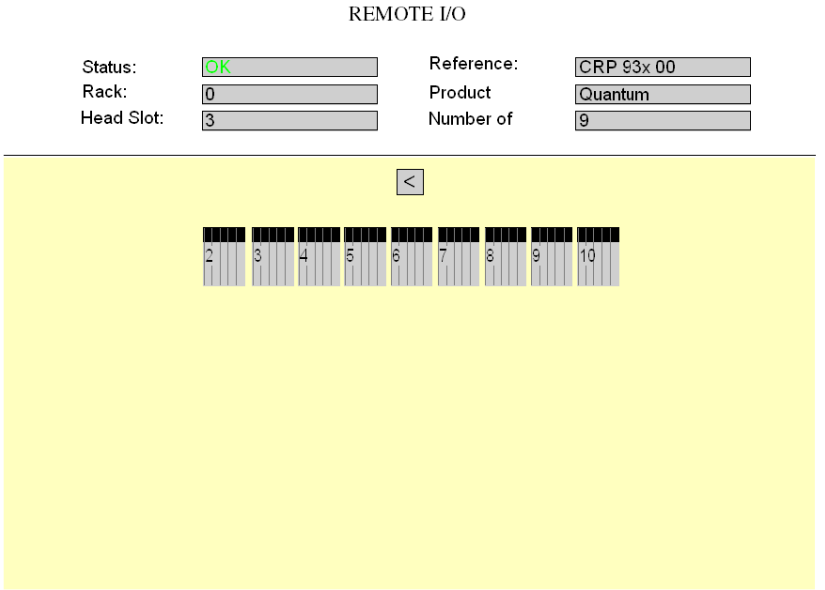
Configured Remote I/O Page

Overview

The **Configured Remote I/O** page displays information about the Remote I/O Head Processor and the number of remote I/O drops. This page can be accessed by selecting the CRP (RIO Head) module in the **Rack Viewer** Page described previously.

Sample Page

Here is an example of a **Configured Remote I/O** page.



Data

The top half of the screen provides the current status (dynamic) and other data about the Remote I/O Head Processor.

The bottom half of the screen displays an icon for each Remote I/O Drop and the drop number. Moving the cursor across the icons will display a text message in the Browser status window indicating whether the drop is 800 series or Quantum I/O.

Links

Click on a drop adapter icon to get detailed information about each drop.

The "<" back button returns you to the previous page.

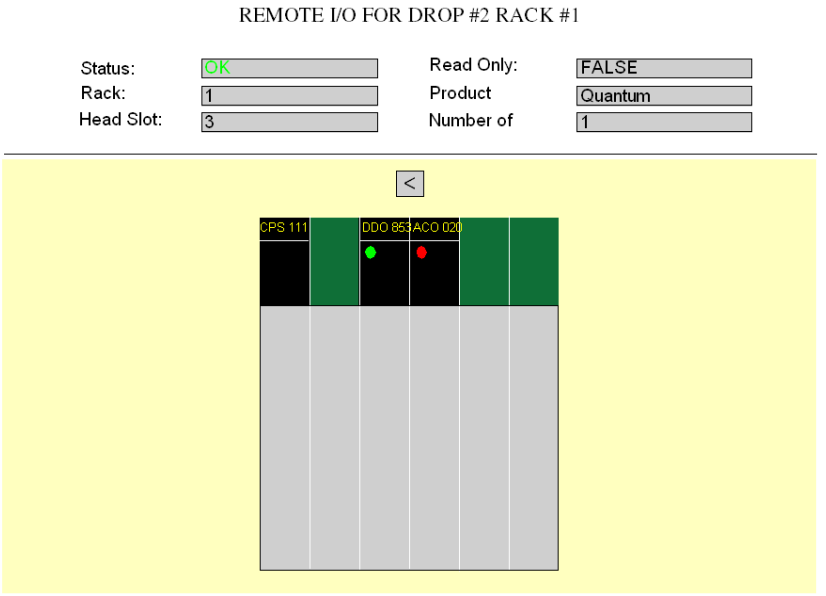
Remote I/O Drop Pages

Overview

When you click the icon for a Remote I/O Drop Adapter on the **Configured Remote I/O** page, you reach a **Remote I/O Drop** page with detailed information about that drop.

Sample Page

Here is an example of a **Remote I/O Drop** page.



Data

The top part of the page reports the current status of the drop adapter and the number of modules in the drop.

The bottom part of the page provides an icon for each module in the drop. A label at the top of the module identifies the module type. Question marks indicated that the module type is unknown or the slot is empty. A colored LED reports module status:

- Green indicates that the module is functioning properly
- Red indicates that the module is not functioning properly

Links

Click a module icon to get detailed information about it.

The "<" back button returns you to the previous page.

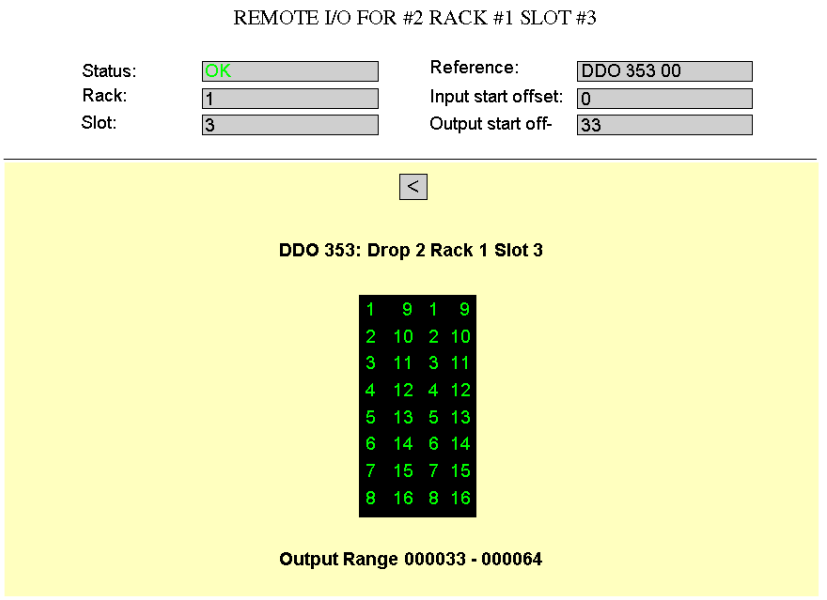
Remote I/O Drop Module Pages

Overview

When you click on a specific module on a **Remote I/O Drop** page, you reach a **Remote I/O Module** page with information about that module.

Sample Page

Here is an example of a **Remote I/O Discreet Module** page.



Data

The top part of the screen provides information about the current status of the I/O module, its location, module type, and input or output offset.

The LED panel in the lower part of the screen displays the status of the discreet I/O points:

- For discrete mdoules:
  - Green indicates the point is active
  - Off indicates the point is not active
- For analog modules:
  - Register values are displayed in integer format

LInks

A "<" back button at the bottom of the screen allows you to return to the previous page.

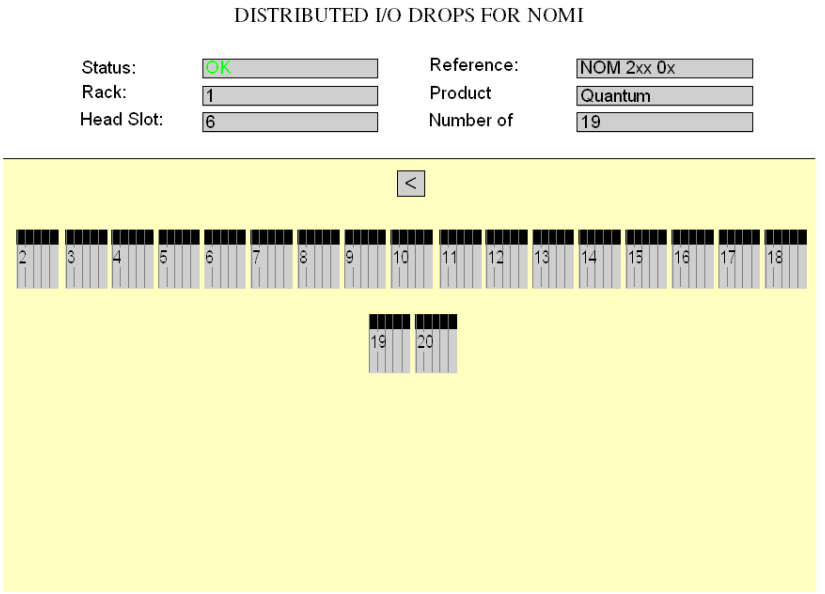
Distributed I/O Drops Page

Overview

When you select a module on the **Rack Viewer** page configured for distributed I/O, you reach a page with detailed information about the distributed I/O network drops controlled by the module.

Sample Page

Here is an example of a **Distributed I/O Drops** page connected to the CPU.



Data

The top half of the screen provides the current status (dynamic) and other data about the controller or NOM module running the distributed I/O network.

The bottom half of the screen displays an icon for each distributed I/O drop.

Links

Click a drop icon to get detailed information abut each drop.

The "<" back button returns to the previous page.

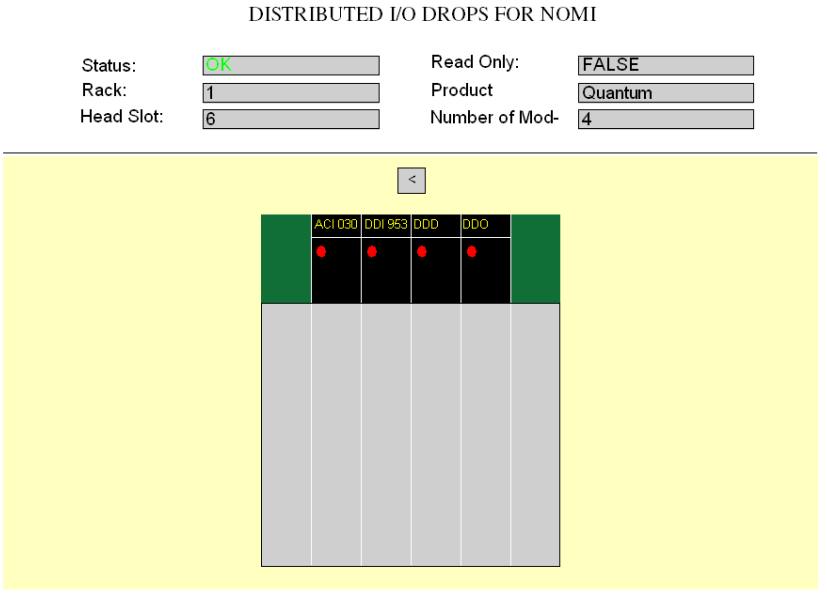
Distributed I/O Specific Drop Page

Overview

When you click a drop icon on the **Distributed I/O Network** page, you reach a **Distributed I/O Drop** page with information about that drop.

Sample Page

Here is an example of a **Distributed Specific I/O Drop** page.



Data

The top part of the page reports the current status of the drop and the number of modules in the drop.

The bottom part of the page provides an icon for each module in the drop. A label at the top of the module identifies the module type.

- Green indicates that the module is functioning properly
- Red indicates that the module is not functioning properly

Links

Click a module icon to get detailed information about that module.

The "<" back button returns to the previous page.

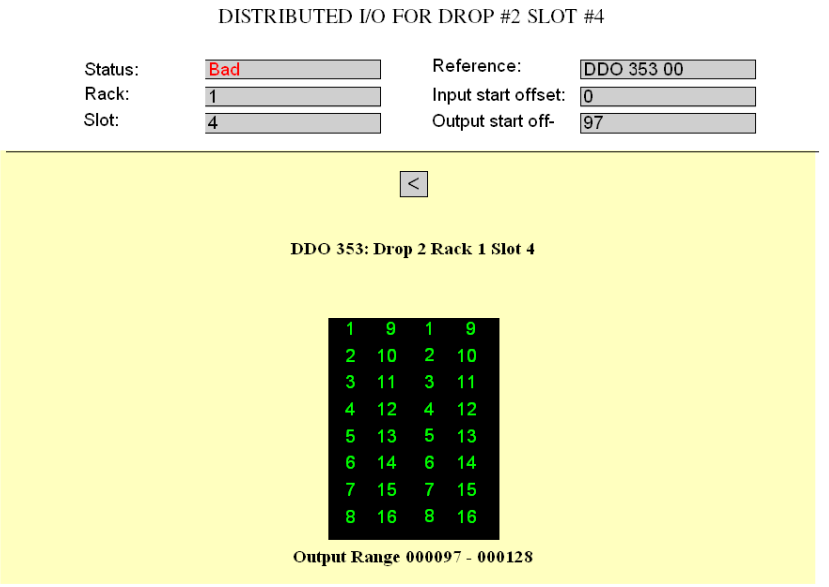
Distributed I/O Module Page

Overview

When you click a module icon on a **Distributed I/O Drop** page, you reach a **Distributed I/O Module** page with information about that module.

Sample Page

Here is an example of a **Distributed I/O Module** page.



Data

The top part of the screen provides information about the current status of the I/O module, its location, module type, and input or output offset.

The LED panel in the lower part of the screen displays the status of the discreet I/O points:

- For discrete mdoules:
  - Green indicates the point is active
  - Off indicates the point is not active
- For analog modules:
  - Register values are displayed in integer format

Links

The "<" back button returns you to the previous page.

3.6 Setup Page

Overview

This section describes the different configuration services proposed by the **Setup** page.

What Is in This Section?

This section contains the following topics:

Topic	Page
Setup Home Page	84
SNMP Configuration	86
Configuring Faulty Device Replacement	89
Global Data (Publish/Subscribe) Utility Configuration	93
Changing the HTTP and Write Passwords	96
Changing the FTP Password	99

Setup Home Page

Home Page

The NOE 771 11 **Setup** page lists the various services used to configure the module. You can navigate to this page from the link on the Home Page *(see page 34)*.

Illustration

The **Setup** page looks like this:



Setup Page Links

These links are on the **Setup** page:

Link	Result
SNMP	Provides the ability to configure the SNMP Agent in the NOE
Address Server	Configure the IP assignments, including showing the BOOTP and DHCP database
Global Data	Displays the **Global Data Configuration** page. Configure the Group Address, Multicast filtering, Distribution period, Health Time Out, Health Bits, and Data Zones. The **Global Data Configuration** page also displays a Variable Table.
NTP	Configure the IP address of the primary and secondary NTP server. Set a polling period. Select a time zone from the drop-down list, and automatic adjustment for daylight savings time.
Email	Configure the email server's IP and Port (default port is 25). If security is required, select **Enable** and set a Login and Password. Create three mail headers.
Security *(see page 96)*	View and edit: • the username and password for accessing the home page • the password for writing variables to the data editor
FTP *(see page 99)*	View and edit the username and password for the FTP service.

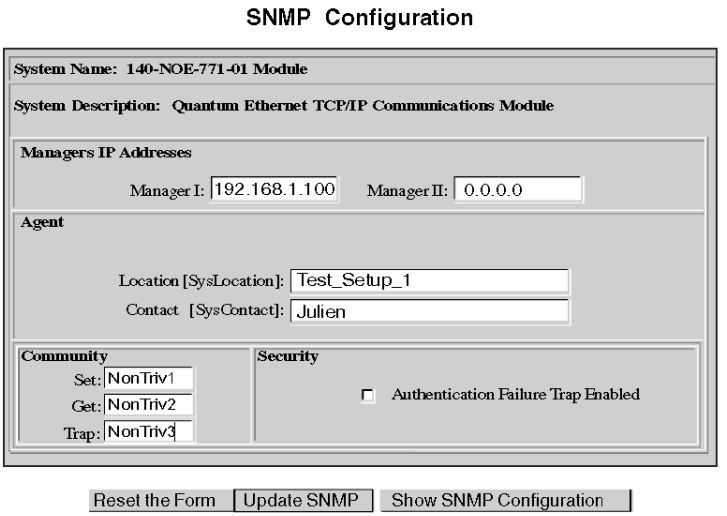
SNMP Configuration

Overview

This topic describes SNMP configuration for the Quantum Ethernet module through the **SNMP Configuration** page.

SNMP Configuration Page

Click the **SNMP** link in the **Setup** home page to display the Configure SNMP page:



The following table describes the SNMP configuration tasks that you can perform through the Web page:

Task	How To
To display the current SNMP configuration	Click **Show SNMP Configuration**.
To clear the fields	Click **Reset the Form**.
To change the SNMP configuration	Change the information on the page and click **Update SNMP**.

SNMP Page Fields

The following table describes the SNMP fields that you can modify.

Field	Information To Be Supplied
Manager I	IP Address of first SNMP Manager
Manager II	IP Address of second SNMP Manager
Location [SysLocation]	Location of the module
Contact [SysContact]	Name of the responsible systems engineer
Set	Designation of user level who can set the configuration
Get	Designation of user level who can view the configuration
Trap	Designation of user level who can capture information
Authentication Failure Trap Enabled	Turns on Community Name Checking

After you make your modifications to any of the above parameters, click **Update SNMP**. A new page displays the following message: "Successfully updated SNMP database." Note that this page contains the same links as those on the **Configure SNMP** page.

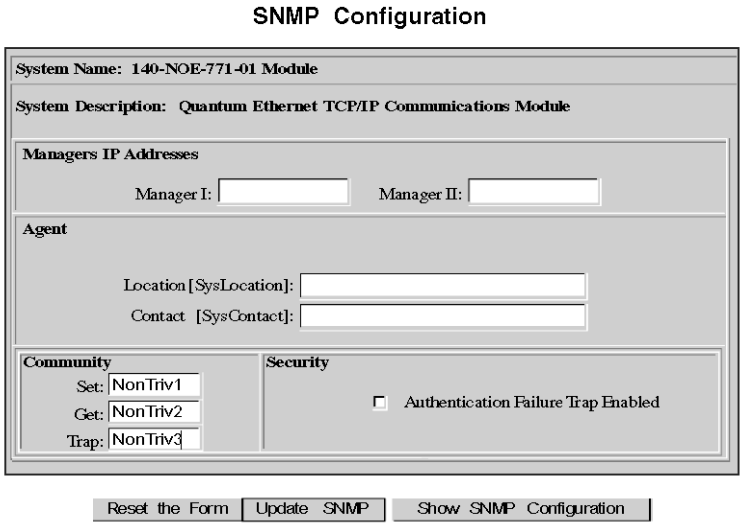
NOTE: Reset the module to implement the changes.

SNMP Community Strings

Use strings to restrict access to the SNMP Agent. These strings should be set to non-trivial names during module installation.

Modifying the SNMP Community Strings

To configure the SNMP community strings:

Step	Action
1	Enter the following URL into your browser: **http://hostname/secure/embedded/builtin?submit=Configure+SNMP** or click the **SNMP** link in the **Setup** home page to navigate to the **SNMP Configuration** page.
2	Enter the **Community** names for **Set**, **Get**, and **Trap** into the **SNMP Configuration** page as shown below. 
3	Click **Update SNMP**.
4	To set your changes, use hot swap to reboot the module.

Configuring Faulty Device Replacement

Configuring the Address Server

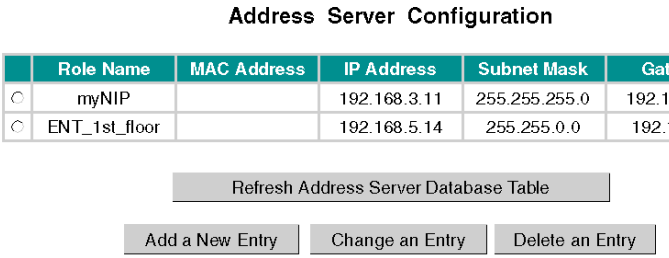
To configure the Address Server you use Web pages generated by the embedded Web server. The first page that appears is the **Address Server Configuration** page. The first column contains buttons used for selecting devices. The **Address Server Configuration** page displays configuration information for each device in the system and has seven columns in the table.

Displayed on this page is information about:

- Role Name
- MAC Address
- IP Address
- Subnet Mask
- Gateway

An additional, unnamed column indicates any difference between the current and stored configuration. If a difference exists, an exclamation point is displayed.

This is the **Address Server Configuration** page. In this sample page, the configured and stored configurations are the same for each device.



On the Address Server Configuration page you can:

- Add a New Entry
- Change an Entry
- Delete an Entry
- Refresh the data table

Choosing Options

The Address Server Configuration page allows you to choose different options for adding or altering the configurations of your NOE. The options available to you are:

- select an entry
- add an entry
- change an entry
- delete an entry

Below we describe the method and options chosen to perform any of the four options listed above. Screen shots are presented to accompany the *Adding an entry* section.

Selecting an Entry When the page displays, by default no entries are selected. Use the radio buttons in the first column to select an entry. Only one entry may be selected at a time.

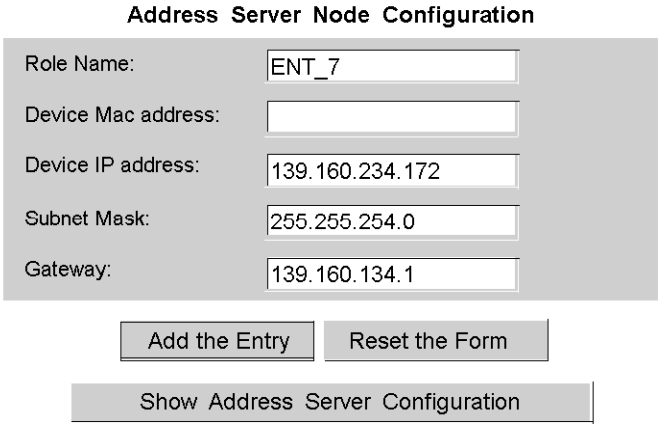
Adding an Entry When the **Add a New Entry** button is selected, the **Address Server Node Configuration** page appears. This page displays information about a device.

If you selected a device, this page displays the device's configuration. Configuration information appears in four of the five fields of the dialog box. Only the Role information field is blank. You should enter a Role name, for example *ENT_7*.

If no device is selected, default values appear in each field.

Changing an Entry Before using this button, you must select an entry in the database by choosing one of the radio buttons in the first column. If you do not choose an entry, a message appears.

When the **Change an Entry** button is selected, the **Address Server Node Configuration** page appears. The information displayed is for the device selected.



Deleting an Entry Before using this button, select an entry in the database by choosing one of the radio buttons in the first column. If you do not choose an entry, a message appears.

The entry selected will be removed from the database. Before completely deleting an entry, a warning message appears. Click **Yes** if you want to delete the entry, **No** if you don't.

If you click **Yes**, a dialog box appears.



Click **OK**. Another dialog box appears notifying you that the deletion was successful.



Highlighting Anomalies

If there are anomalies with the entered configuration parameter information, the system indicates anomalies using a highlighting mechanism. Valid device configurations appear in purple and italic. A device with configuration anomalies appears in red and bold.

The system detects the following anomalies:

- Bad **Role Name**
  The valid **Role Name** characters are:
  - a through z (lower case)
  - A through Z (upper case)
  - "_" (underscore)
- Bad **MAC Address**
  The valid **MAC Address** characters are:
  - 0 through 9
  - A through F
- Wrong **IP Address**
- Wrong **Subnet Mask**
- Wrong **Gateway**
- Double Entry

Each entry needs a unique **Role Name** or **MAC Address**. If a duplicate **Role Name** or **MAC Address** is entered, the system highlights the duplication.

Erroneous entries are not loaded into the DHCP server. Therefore, correct erroneous before loading. There are two ways of correcting the erroneous entries:

- **Correcting through Web page:** Make the changes on the Web page and submit the change.
- **Correcting through the Address Server configuration file:** Make the changes in the file and reboot the server.

Global Data (Publish/Subscribe) Utility Configuration

Overview

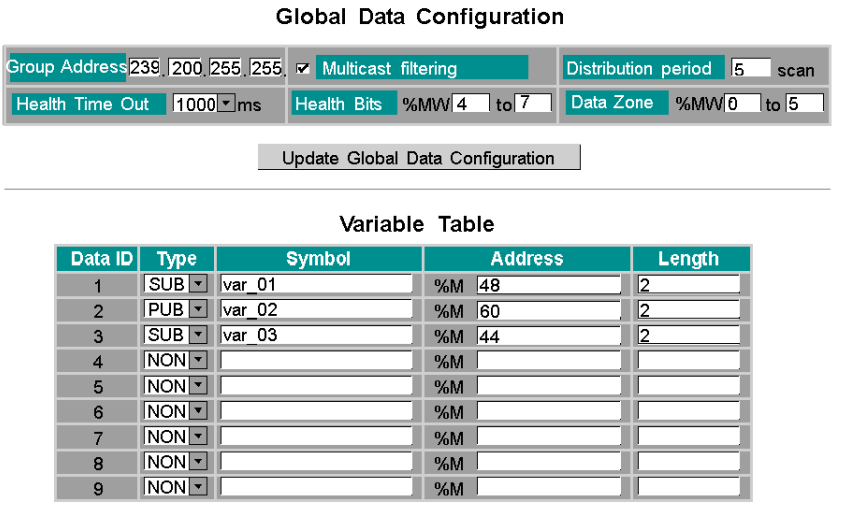
Whether you use either the Configure Each Device Separately or the Copy Configuration method, the procedure to configure individual parameters is the same. Therefore, in order to use the Global Data (publish/subscribe) utility in the NOE, you need to configure the Global Data parameters including:

- Distribution period
- Multicast filtering
- Health bit location
- Global Data base address
- Group IP address

The following sections describe in detail the exact steps to configure each parameter on the **Global Data Configuration** page.

Illustration

You can change the configuration in the **Global Data Configuration** page:



Configuring Global Data

After you have completed the Modeling System Configuration process using the second method, Copy Configuration, you modify the following parameters:

- Distribution period
- Health Time Out
- Health Bits location

- Start address
- Type: Pub / Sub / None

Please **do not** change Symbol (description), and Length.

To change the Global Data variables of the group box on the **Global Data Configuration** page, follow the instructions below.

Step	Action
1	Adjust the **Distribution Period** cycle. Enter a value from 1 through 50. **Note: Distribution Period** is the minimum number of controller scan times before an update occurs.
2	Before entering a value in the **Group Address** field, identify the station's distribution group. The **Group Address** entry will be an IP address from **`224.0.0.0`** through **`239.255.255.255`**. **Group Address:** the Class D Multicast IP address used for a distribution group. Members of this distribution group are configured to use the same group address; therefore, members can communicate with each other using Global Data.
3	Set the timeout in the **Health Time Out** field. This value is measured in milliseconds and can be set to a value that ranges from 50 through 15000 ms (in 50 ms increments) **Note: Health Time Out** is the maximum time between received subscriptions before a subscription is declared unhealthy (inoperable).
4	In the 4x starting address, set the **Data Zone** field.
5	If you are connected to an Ethernet switch that supports multicast filtering, click the **Multicast filtering** check box.
6	Enter %MW word (4x register) location for the Health Bits. This is the location for storing health bits.

NOTE: Health bits run in different directions.

- I/O scanner health bits run left to right.
- Global Data health bits run right to left.

Changing Global Data Variables

To change the Global Data variables that appear in the **Variable Table** area, follow the instructions below.

Step	Action
1	Highlight the identification number in the **Data ID** column.
2	In the **Type** column select the publish/subscribe variable type from the drop down list. Three options are available publish, subscribe, or none, displayed as: • **NONE** • **SUB** • **PUB**
3	In the **Symbol** column you may enter text to describe the variable.
4	In the **Address** column you see the application address for this variable. **Note:** This is a read only field.
5	In the Length column for each row, type a value, which represents the number of 4x registers. The ending 4x register field is automatically updated. If you are using the second method, **Copy Configuration**, update **Length** the first time only.
6	When you are finished, click the **Update Global Data Configuration** button.

Verifying System Operation

To verify that the system is operational, do the following:

Step	Action
1	Verify that every controller is running.
2	Look at the health of variables using the **Global Data Diagnostics** page. Follow these links: **Diagnostics** → **Ethernet** → **Global Data**

Changing the HTTP and Write Passwords

Presentation

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 15 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 procedure below 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.

View of the Security Page

The Security page looks like this:



Modifying HTTP Access Rights

The procedure is as follows:

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

The procedure is as follows:

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 restricting Web page access are **USER**/**USER**.

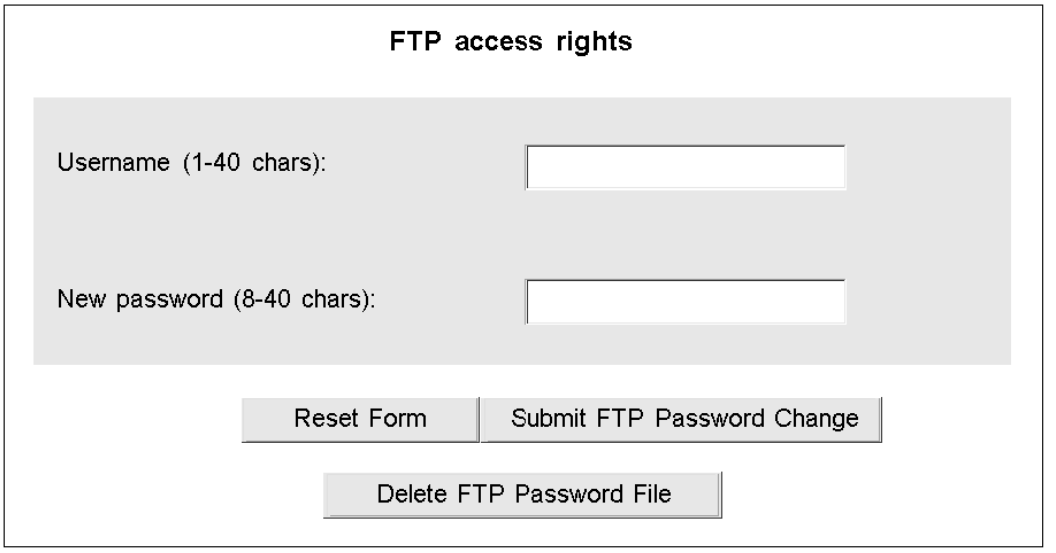
Accessing the FTP Configuration Page

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:



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 **Submit FTP Password Change**. **Result**: a confirmation window appears.

Default Web Site for Premium

4

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 may 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 of *USER*.

This chapter describes the pages in the default Web site.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
4.1	Premium Home Page	102
4.2	Monitoring Page	104
4.3	Diagnostics Page	106
4.4	Module Diagnostics (Silverlight)	114
4.5	Module Diagnostics (Java)	133
4.6	Setup Page	157

4.1 Premium Home Page

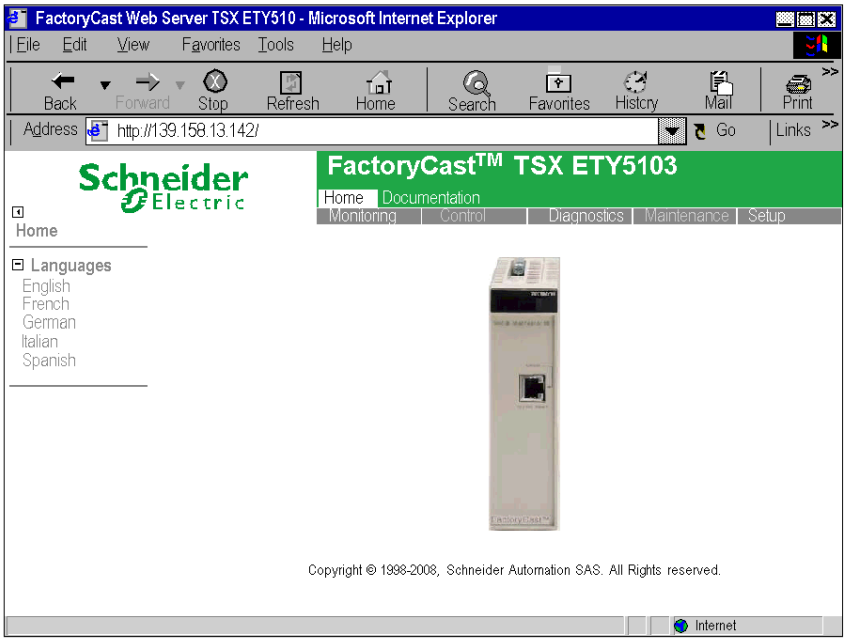
Premium Home Page

Overview

To access the Premium home page enter the IP address of the module in his web browser. No password is required to display this page.

Home Page

This Premium home page looks like this:



Links

From the Premium home page, you can access the following pages :

- Monitoring *(see page 104)*
- Diagnostics *(see page 107)*
- Setup
- Documentation
- Foreign language links exist for French, German, Italian and Spanish versions

Enter a user name and a password to access the services proposed in these pages.

4.2 Monitoring Page

Monitoring Home Page

Home 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** home page looks like this:



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

- Data Editor *(see page 176)*: for creating variable data tables, so as to be able to determine their value when the table is animated.
- Data Editor Lite *(see page 193)*: for creating variable data tables, so as to be able to determine their value when the table is animated (lighter than Data editor).
- Graphic Editor *(see page 204)*: for creating graphics, so as to be able to determine the value of variables when the graphic is animated.
- Graphic Viewer *(see page 249)*: for viewing graphics, so as to be able to determine the value of variables when the graphic is animated.
- PLC Program Viewer *(see page 251)*: Display UnityPro programs in run mode using a web browser.
- Password-protected custom pages *(see page 259)*: for viewing screen pages (accessible with password) created by the user
- Custom pages without password protection *(see page 259)*: for viewing screen pages (accessible to everyone) created by the user.
- Silverlight pages *(see page 325)*: for viewing user-defined pages created with *Microsoft Expression Blend®* software, that have been added to the web site.

4.3 Diagnostics Page

Overview

This section describes the different services proposed by the **Diagnostics** page.

What Is in This Section?

This section contains the following topics:

Topic	Page
Diagnostics Home Page	107
Rack Viewer Page	108
Properties Page	113

Diagnostics Home Page

Home Page

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

Illustration

The **Diagnostics** home page looks like this:



Links

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

- Rack Viewer
- Alarm viewer *(see page 327)*
- Ethernet

Rack Viewer Page

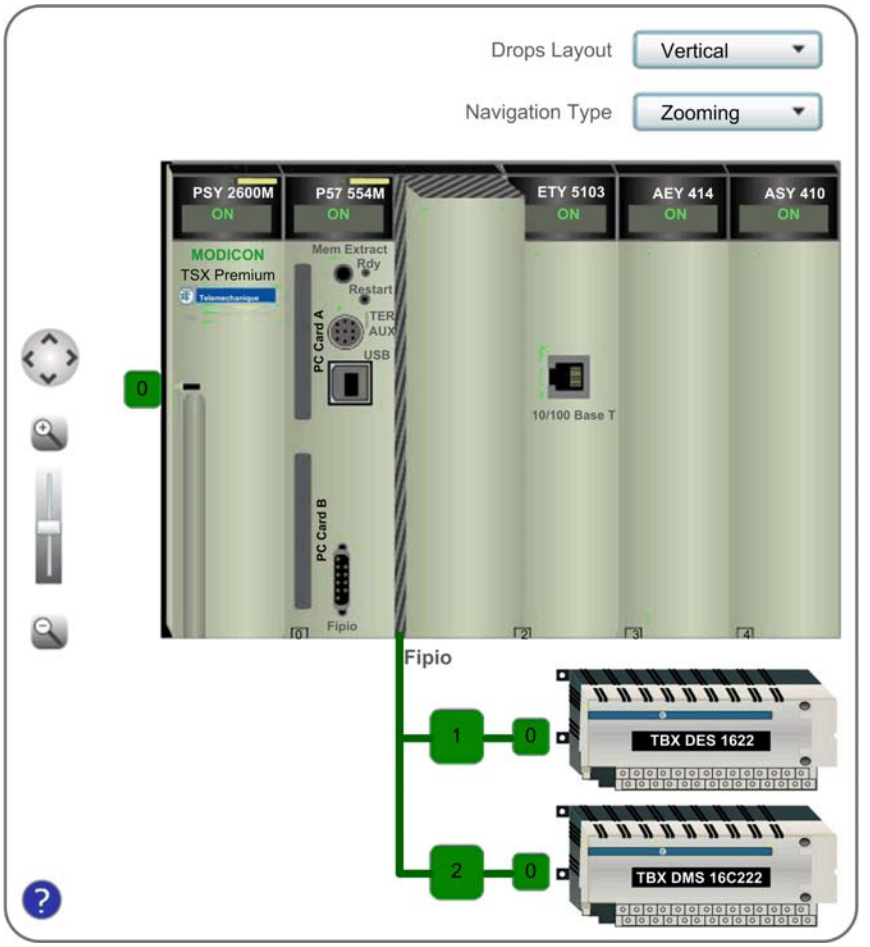
Silverlight and Java Page Versions

The **Rack Viewer** page displays the current configuration of your network, as either a Silverlight or a Java page. The technology used to display your network in this page depends on:

- How you configured your Unity pro application **Project Settings**. Refer to the description of Project Settings *(see page 40)* for the diagnostic web pages for an explanation of your configuration options.
- The browser-supporting software that is installed on your PC. Refer to the description of Browser Supporting Software *(see page 41)* for an explanation of software that supports Java and Silverlight pages.

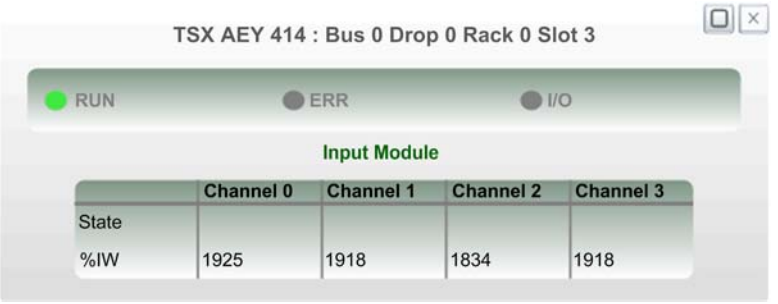
Silverlight Page Example

The Silverlight version of the **Rack Viewer** presents the network modules in a single screen. The following graphic depicts a main rack with two remote Fipio drops.



Silverlight Page Premium Module Addressing

Each Premium module in the **Rack Viewer** has an address. When you open the diagnostic page for a Premium module, its address appears at the top of the page. For example, the address of the TSX AEY 414 analog output module, above, appears below:



Each Premium module address consists of the following zero-based numerical elements:

- Bus: the number of the network. Because there can be only one X Bus, each bus value is set to “0”. **Rack Viewer** does not display this value in the network diagram. The bus value appears in the status page for an individual module.
- Drop: the node on the network (large node number). There can be only one drop—set to a the value “0”—for each X Bus. **Rack Viewer** does not display this value in the network diagram. the drop value appears in the status page for an individual module.
- Rack: a drop can consist of multiple racks. The rack number appears in the small node next to a main or extension rack. The preceding diagram depicts the rack number (“0”) in the small node next to both the main rack and extension rack.
- Slot: each rack can consist of multiple modules. Slot numbers are assigned to each module in a Premium rack, except power supply modules. In the preceding graphic, the numbers 0...4 indicate the slot number of modules in the Premium main rack.

When you place your mouse pointer over a module, the **Rack Viewer** displays information describing the module and its location in the network.

Silverlight Page Diagnostics

Color indicators display the operational health of modules, racks, drops and the network bus.

Modules:

Each 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 module to open a pop-up page with detailed information about that module.

NOTE: In the following cases, **Rack Viewer** cannot display module information:

- If the module is configured in Unity Pro, but not physically present in the rack, **Rack Viewer** displays the message *Module is missing*.
- If the module physically present in the rack is not the same as the module configured in Unity Pro, **Rack Viewer** displays the message *Detected Module is different from the one configured in this position*.

Racks and Drops:

The node next to each drop and rack indicates the health of that node:
- Green indicates every module in the rack or drop is functioning properly.
- Red indicates at least one module in the rack or drop is not functioning properly.

Bus:

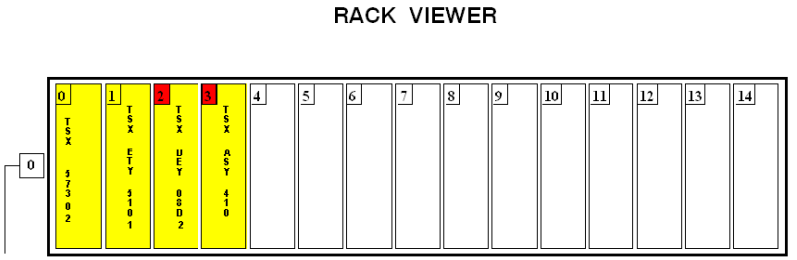
The color of the line representing the network bus indicates the health of the bus:
- Green indicates every module on the bus is functioning properly.
- Red indicates at least one module on the bus is not functioning properly.

Silverlight Page Navigation

For information on displaying and navigating the **Rack Viewer** page, refer to the topic Navigating the Rack Viewer Silverlight Page *(see page 46)*.

Java Page Example

The following graphic depicts a Java version of the **Rack Viewer** page.



Java Page Diagnostics

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

- A vertical label displays the module type and part number
- The box in the upper left-hand corner of the module displays the slot number and module health:
  - A yellow box indicates that the module is functioning properly
  - A red box indicates that the module is not functioning properly

Java Page Navigation

Click a module icon to obtain detailed information about it.

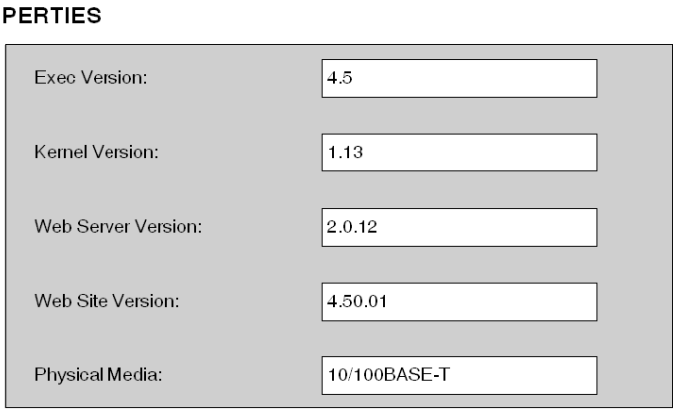
The CPU module icon (slot 1) contains two links. The top link leads to the **Fipio Module Diagnostics** page. The tower link leads to the PLC Personality page.

A line leading down from the Rack#0 icon (to the left of the rack) is a link to the next rack. When you place your mouse over this link, it turns into a red arrow. Each rack in the configuration may be viewed in turn. Upward links will take you back toward Rack#0.

Properties Page

Introduction to the Properties Page

You can navigate to the **Properties** page from the Diagnostics page *(see page 106)*. The **Properties** page displays the versions of the Exec, Kernel, and Web Pages and the Physical Media:



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

4.4 Module Diagnostics (Silverlight)

Overview

Many Premium FactoryCast modules and connections present diagnostic web pages. These diagnostic web pages are displayed in the **Rack Viewer**.

What Is in This Section?

This section contains the following topics:

Topic	Page
Premium Controller Status	115
Premium CPU Serial Port Status	117
CPU PCMCIA Card Status	119
Premium Ethernet Communication Module Status	123
Discrete I/O Modules	124
Analog I/O Modules	127
AS-i Module Diagnostics Page	128
Fipio Diagnostics Pages	129

Premium Controller Status

Overview

Click on different parts of the controller image in **Rack Viewer** to access diagnostic pages for that module. The pages you can access depend on the specific controller in your application. For example, if you are using the TSX P57 6634M controller, you can display the following pages:

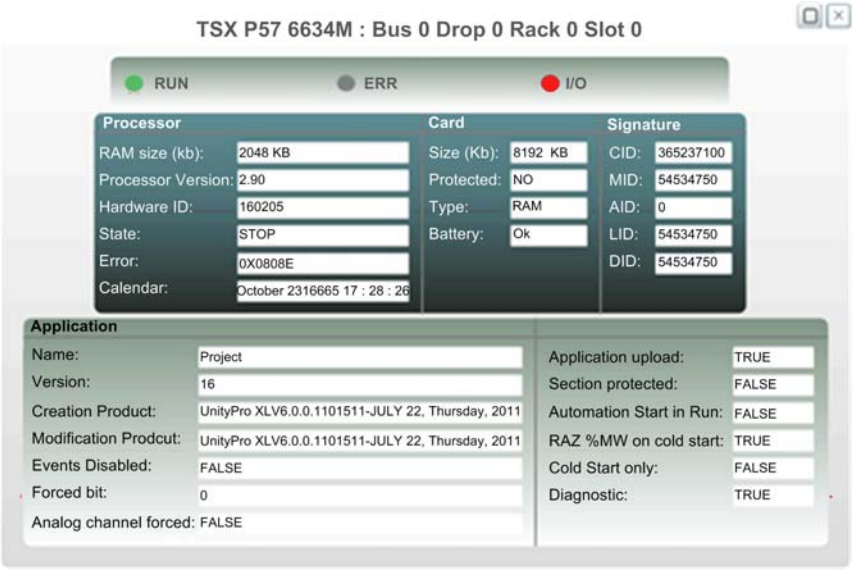
If you click on the...	You can view a diagnostic page describing...
CPU module face	The CPU processor, card, signature and application.
Ethernet port	The CPU Ethernet configuration and activity.
Serial port *(see page 117)*	The CPU serial—or terminal—port status and configuration.
PCMCIA slot (A or B) *(see page 119)*	The selected PCMCIA card status and configuration.

The diagnostic pages are read-only Silverlight pages that present data describing the selected feature of the CPU.

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

Controller Status

Click on the face of the TSX P57 6634M CPU in the **Rack Viewer**, but not on a port image, to open the following page describing both the CPU and the application running in the CPU:



NOTE: You can access additional data about the controller in the **PLC Personality** Java web page.

Ethernet Port Status

Click on the Ethernet port of the TSX P57 6634M CPU in the **Rack Viewer** to open a page describing the CPU IP address settings and recent Ethernet activity:



Premium CPU Serial Port Status

Serial Port Status

Click on the serial port (also called the terminal port) of the CPU in the **Rack Viewer** to open a page describing the:

- detected fault status of the CPU serial port, in the top panel
- the configuration of the serial port, in the bottom panel

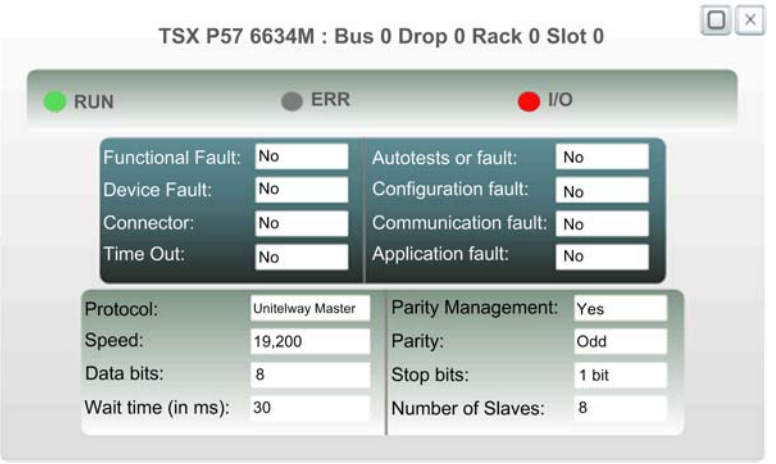
NOTE: The parameters presented in the configuration (bottom) panel of this page depend on the selected protocol. The serial port supports the following protocols:

- Unitelway master
- Unitelway slave
- Character mode

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

Unitelway Master Protocol

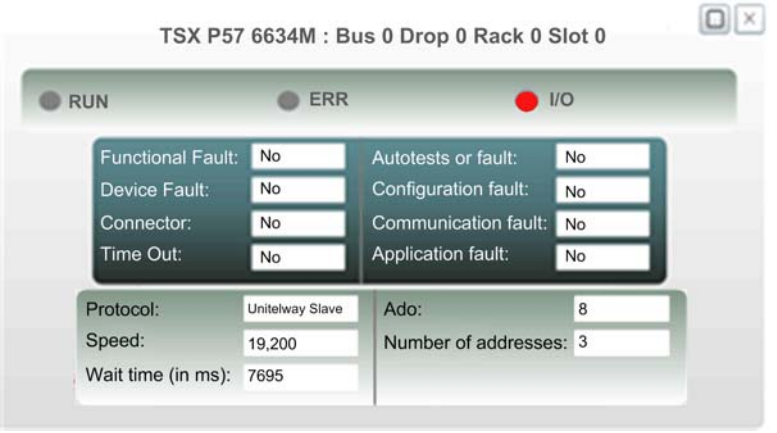
The following graphic depicts the CPU serial port status page for the Unitelway master protocol:



NOTE: The serial port status page for the Unitelway master protocol contains the same fields as the PCMCIA card page for the same protocol.

Unitelway Slave Protocol

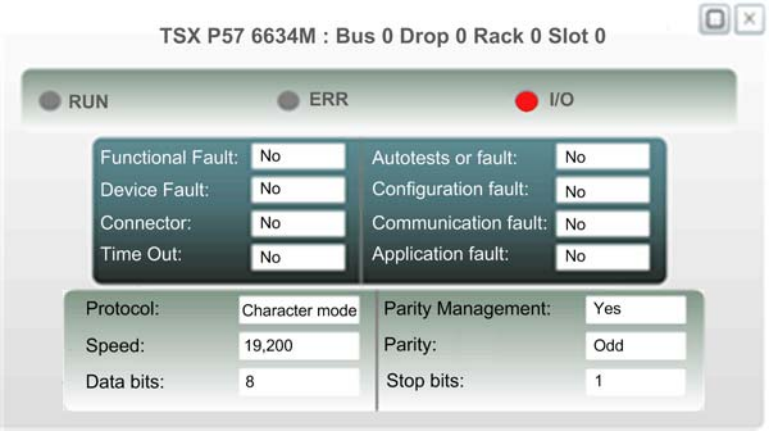
The following graphic depicts the CPU serial port status page for the Unitelway slave protocol:



NOTE: The serial port status page for the Unitelway slave protocol contains the same fields as the PCMCIA card page for the same protocol.

Character Mode Protocol

The following graphic depicts the CPU serial port status page for the Character mode protocol:



NOTE: The serial port status page for the character mode protocol contains the same fields as the PCMCIA card page for the same protocol.

CPU PCMCIA Card Status

PCMCIA Card Status

If port B of the CPU contains a PCMCIA card that is configured in your application, you can display a diagnostic page for that card. To determine if the CPU is equipped with a configured PCMCIA card, place your mouse pointer over Port B and look for the tooltip message *Click to get PCMCIA diagnostic page*.

The diagnostic page for a PCMCIA card describes:

- the detected fault status of the selected PCMCIA card (top panel)
- the configuration of the selected PCMCIA card (bottom panel)

NOTE: The parameters presented in the configuration (bottom) panel of this page depend on the selected protocol. PCMCIA slot B supports the following protocols:

- Unitelway master *(see page 117)*
- Unitelway slave *(see page 118)*
- Character mode *(see page 118)*
- Modbus master
- Modbus slave
- Modbus Plus
- Fipway
- CANopen

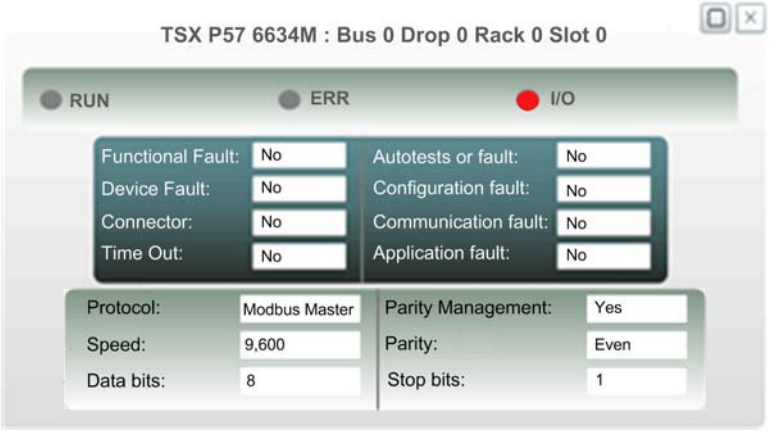
The PCMCIA card pages for Unitelway master, Unitelway slave and Character protocols are the same as for the serial port pages. Refer to the topic Premium CPU Serial Port Status *(see page 117)* for a description of these pages.

These pages are read-only Silverlight pages that present data describing the selected PCMCIA card. In the following examples, each PCMCIA card is placed in a slot on a TSX P57 6634M CPU.

NOTE: For a description of a specific PCMCIA card parameters, consult the documentation for that card.

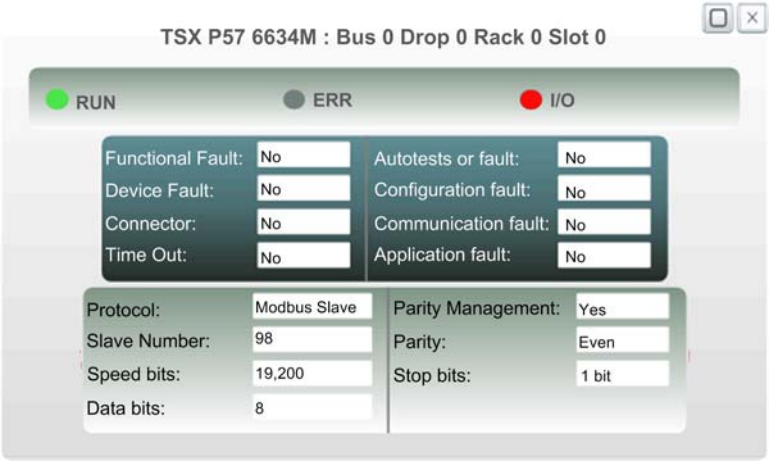
Modbus Master Protocol

Click on a PCMCIA slot on the CPU to display the following Modbus master protocol status and configuration data:



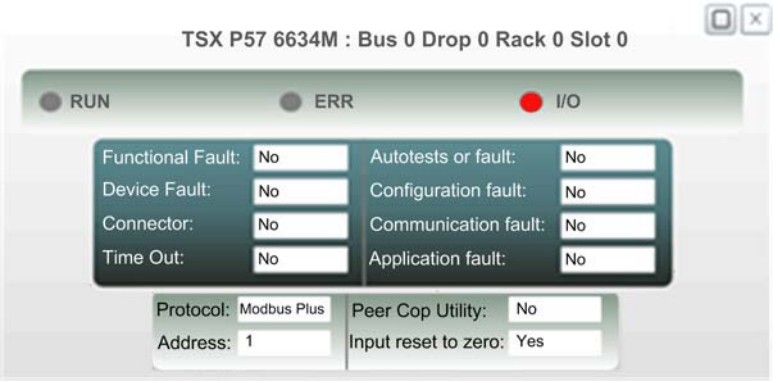
Modbus Slave Protocol

Click on a PCMCIA slot on the CPU to display the following Modbus slave protocol status and configuration data:



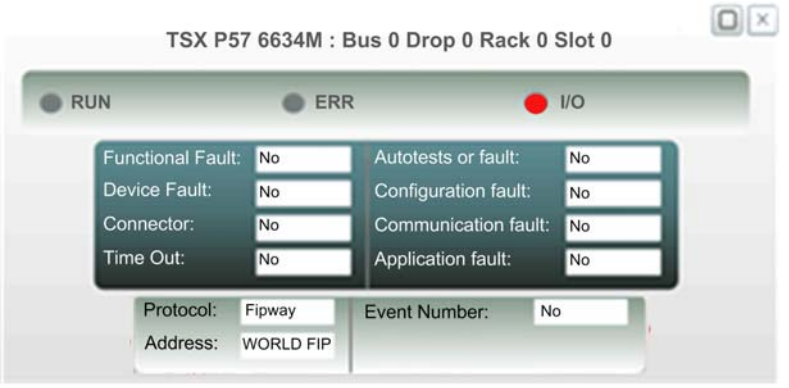
Modbus Plus Protocol

Click on a PCMCIA slot on the CPU to display the following Modbus plus protocol status and configuration data:



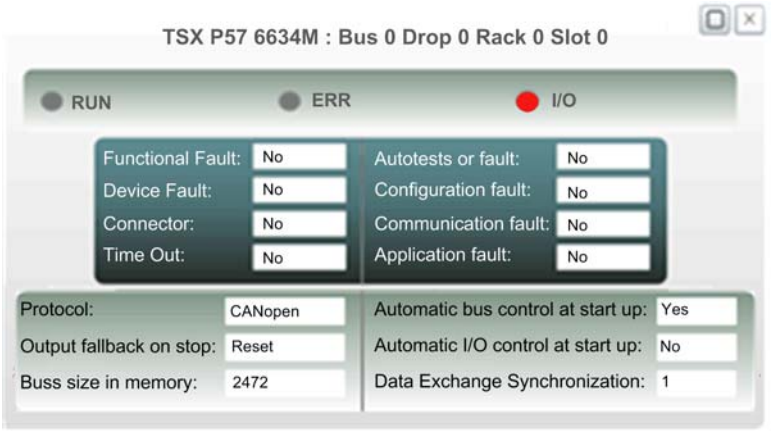
Fipway Protocol

Click on a PCMCIA slot on the CPU to display the following Fipway protocol status and configuration data:



CANopen Protocol

Click on a PCMCIA slot on the CPU to display the following CANopen protocol status and configuration data:



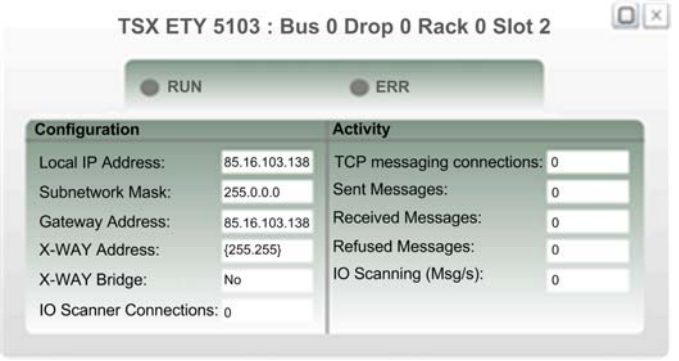
Premium Ethernet Communication Module Status

Ethernet Communication Module Diagnostic Page

The Premium Ethernet communication module status page presents data describing:

- the Ethernet communication module IP address settings, and
- recent Ethernet messaging activity

Click on the Ethernet communication module in **Rack Viewer** to access this page. The following graphic depicts the diagnostic page for the TSX ETY 5103 Ethernet communication module:



This page is a read-only Silverlight page.

NOTE: For a description of the data presented for a specific Ethernet communication module, consult the documentation for that module.

Discrete I/O Modules

Overview

When you click on a discrete input or discrete output module in a Premium main or extension rack, a Silverlight web page opens displaying information about that module, including:

- the discrete input or output module type
- bus, drop, rack and slot location
- the status of the RUN, ERR and I/O module LEDs
- the status of the inputs or outputs

The page displays I/O data as discrete points in an LED panel. When the I/O point number is:

- light green: the I/O point is active
- dark green the I/O point is inactive

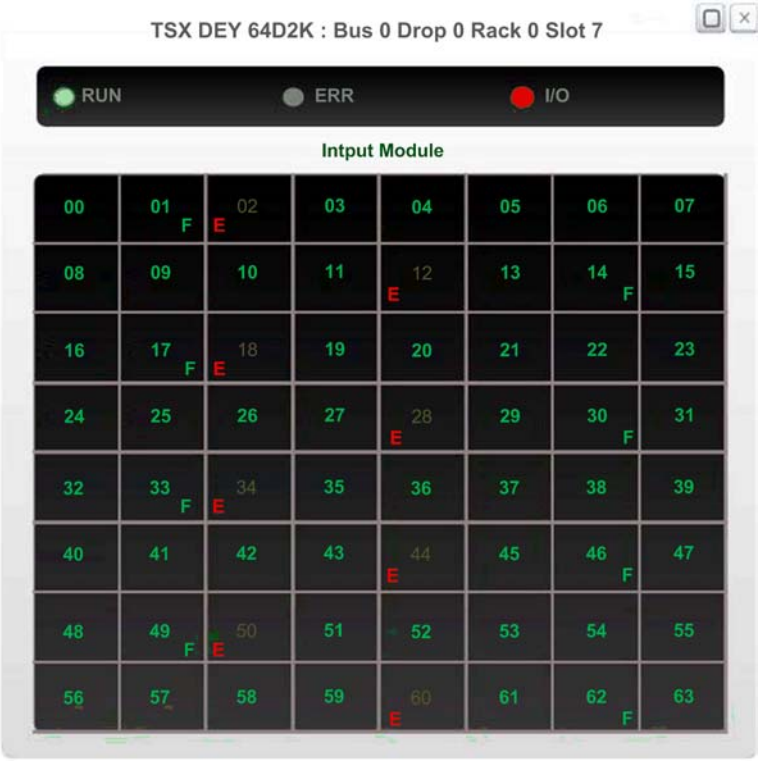
Each I/O point can also include a letter indicating its state, as follows:

- a red E indicates a detected error on that point
- a red FB indicates the point is in its fall back state
- a green F indicates the point has been forced

NOTE: For a description of the data presented for a specific I/O module, consult the documentation for that module.

Sample Discrete Input Module Page

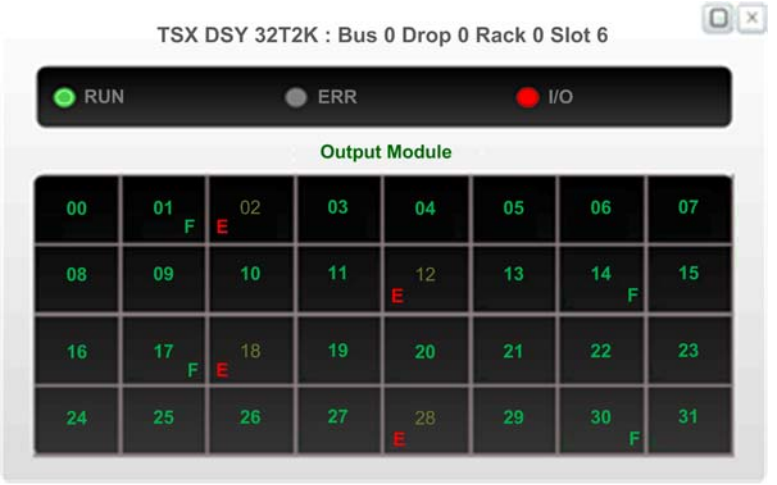
The discrete input module, depicted below, displays the status of each bit comprising four words:



Each bit is active, as denoted by the green color of the bit number.

Sample Discrete Output Module Page

The discrete output module, depicted below, also displays the status of each bit in two words:



Each bit is active, as denoted by the green color of the bit number.

Analog I/O Modules

Overview

When you click on an analog input or analog output module in a Premium main or extension rack, a Silverlight page opens displaying information about that module, including:

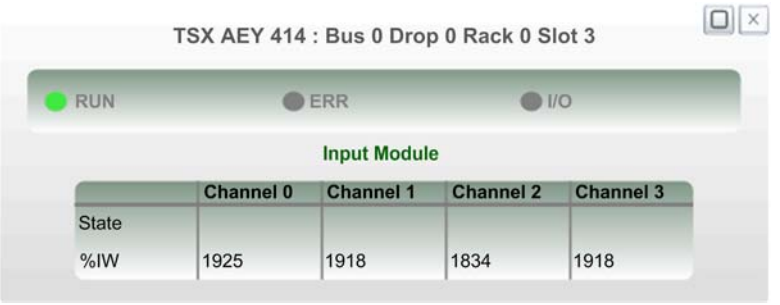
- the analog input or output module type
- bus, drop, rack and slot location
- a register address of each channel assigned to the module
- the status of each channel

Analog modules display data exclusively as words.

NOTE: For a description of the data presented for a specific analog I/O module, consult the documentation for that module.

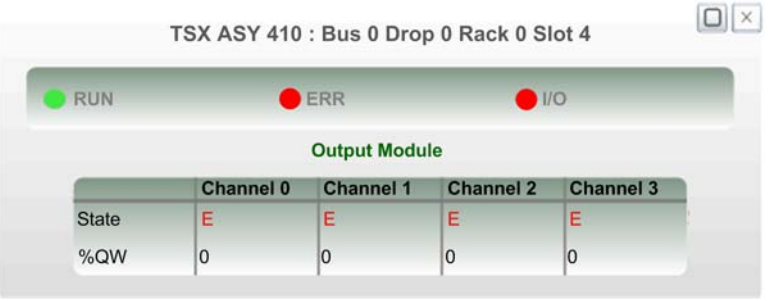
Sample Analog Input Module Page

The analog input module, below, displays the status of four channels:



Sample Analog Output Module Page

The analog output module, below, displays the status of four channels:



AS-i Module Diagnostics Page

AS-i Module Diagnostics

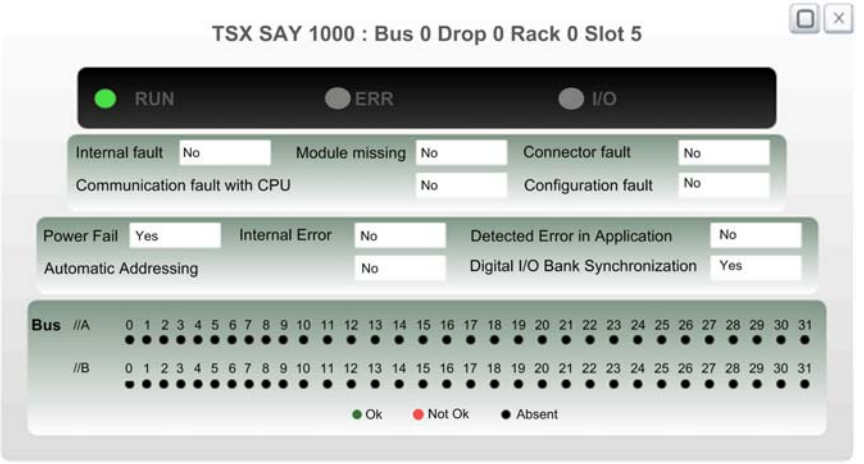
Click on an actuator sensor interface (AS-i) module in the **Rack Viewer** to display a diagnostics page for that module. The page presents the following data:

- the state of module LEDs
- the status of ongoing event detection
- the state of up to 64 nodes on the AS-i bus

NOTE: For a description of the data presented for a specific AS-i module, consult the documentation for that module.

Sample Page

The TSX ASY 1000 module AS-i diagnostics page displays the state of the module.



LED Data

The LEDs provide a dynamic report on the module status.

LED	Color if On	Meaning if On	Meaning if blinking	Meaning if Off
RUN	Green	Module **OK** and configured	Awaiting configuration	Module out of order
ERR	Red	Serious non-rectifiable Module event	Rectifiable module event (PL7 configuration, AS-i power supply)	Module **OK**
I/O	Red	AS-i bus detected fault	Invalid configuration detected, or awaiting user configuration	AS-i Bus **OK**

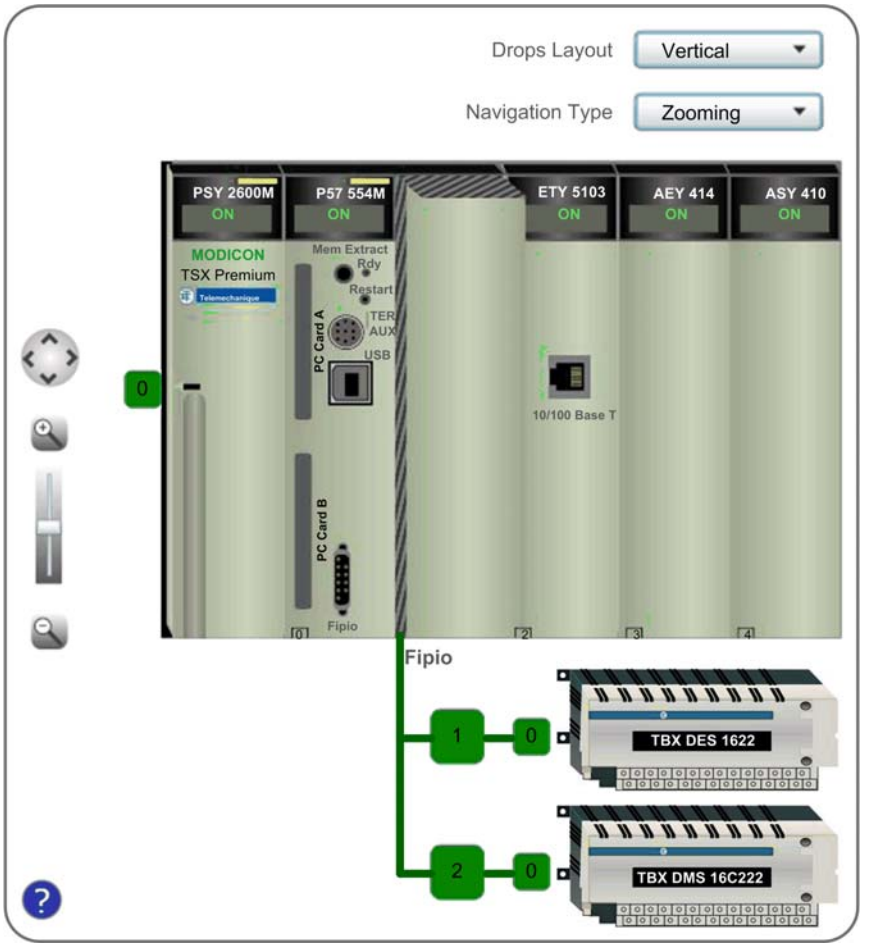
Fipio Diagnostics Pages

Fipio Bus

If your Premium CPU supports Fipio, the Silverlight **Rack Viewer** presents both the Premium X Bus and the Fipio bus in a single diagram. The Fipio modules include:

- the CPU module, located in the main rack
- individual Fipio devices, as nodes on the Fipio bus

The **Rack Viewer** displays the Fipio bus connected to the CPU, as follows:



Fipio Addressing

Each Fipio node on the bus also has an address, which consists of the following zero-based numerical elements:

- Bus: The number of the Fipio bus. Because there can be only one Fipio Bus, the bus value is set to “0”. Neither the **Rack Viewer** nor the status page for individual devices displays this value.
- Connection Point: Comparable to a “Drop” on the X Bus, the Fipio bus can include multiple connection points. A connection point is identified by a large node. For example, the preceding diagram displays two Fipio drops—drop 1 and 2—are indicated by numbers in the large nodes.
- Rack: This addressing element is not used in Fipio and is set to “0”. The status page for an individual device displays this value at the top of the page.
- Module: The **Rack Viewer** page displays the Module number in a small node next to the rack or device. For example, the preceding diagram displays two Fipio racks at addresses:
  - Connection Point 1, Rack 0, Module 0
  - Connection Point 2, Rack 0, Module 0

For a description of the addressing for the P57 554M CPU module in the Premium rack on the X Bus, refer to the Rack Viewer *(see page 109)* topic.

Fipio Bus Diagnostics

Color indicators display the operational health of the Fipio bus, connection points, and modules.

Modules:

The color of the small node, next to a module, indicates the status for that module:

- Green indicates that the module is functioning properly.
- Red indicates that the module is not functioning properly

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

Connection Points:

The node next to each drop and rack indicates the status of modules at that connection point:

- Green indicates every module at the connection point is functioning properly.
- Red indicates at least one module at the connection point is not functioning properly.

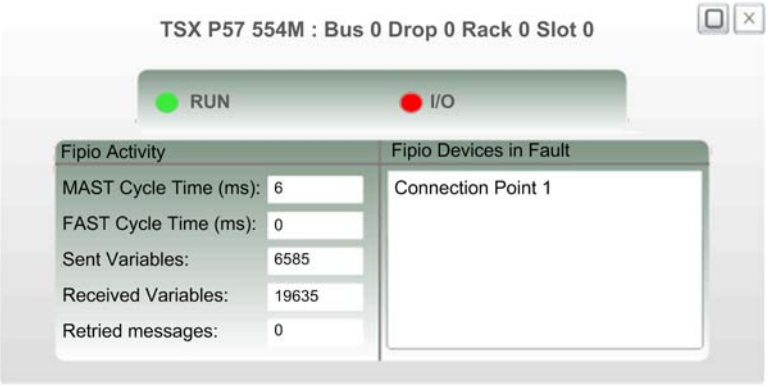
Bus:

The color of the line representing the Fipio bus indicates the health of the bus:

- Green indicates every module on the bus is functioning properly.
- Red indicates at least one module on the bus is not functioning properly.

Fipio Bus Status

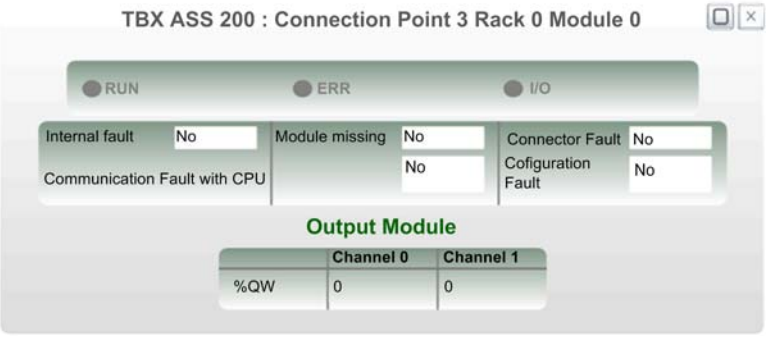
Click on the 9-pin Fipio connector on the CPU to open a page that displays the status of the Fipio bus, Fipio bus activity, and a list of Fipio devices with a detected fault.



NOTE: For a description of the data presented for a specific module, consult the documentation for that module.

Fipio Analog Module Status

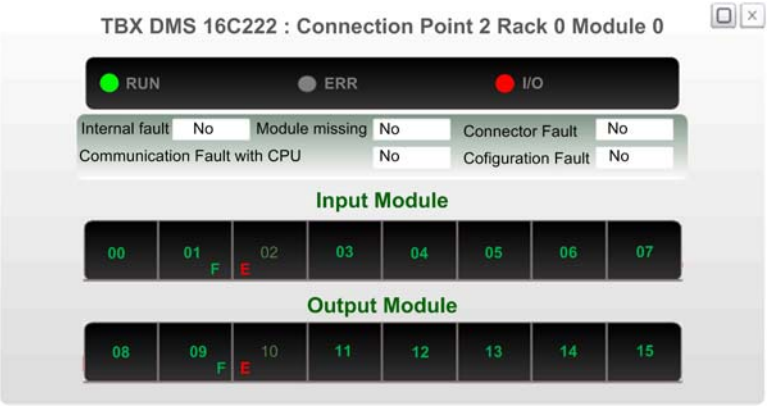
Click on an Fipio analog device to open a page that displays its operating status, detected faults, and channel status.



NOTE: For a description of the data presented for a specific module, consult the documentation for that module.

Fipio Digital Module Status

Click on a Fipio digital device to open a page that displays its operating status, detected faults, and the status of each digital bit.



The page displays I/O data as discrete points in an LED panel. When the I/O point number is:

- light green: the I/O point is active
- dark green the I/O point is inactive

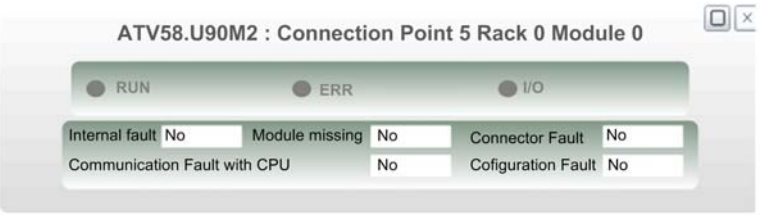
Each I/O point can also include a letter indicating its state, as follows:

- a red E indicates a detected error on that point
- a red FB indicates the point is in its fall back state
- a green F indicates the point has been forced

NOTE: For a description of the data presented for a specific module, consult the documentation for that module.

Fipio Generic Module Status

Some Fipio devices are not designed to display module-specific diagnostic parameters. When you click on a Fipio generic device, a page opens displaying its operating status and detected faults.



4.5 Module Diagnostics (Java)

Overview

This section presents the Java version of diagnostic web pages. Your browser displays these pages if your application is configured to display the diagnostic pages in Java format, *(see page 108)*or if Silverlight supportive files are not installed on your PC.

What Is in This Section?

This section contains the following topics:

Topic	Page
PLC Personality Page	134
Ethernet Module Statistics Page	136
Fipio Module Diagnostics Page	141
Digital I/O Module Diagnostics Page	147
Analog I/O Module Diagnostics Page	149
AS-i Module Diagnostics Page	151
PCMCIA Communication Card Diagnostics Page	153
Standard Module Diagnostics Page	155

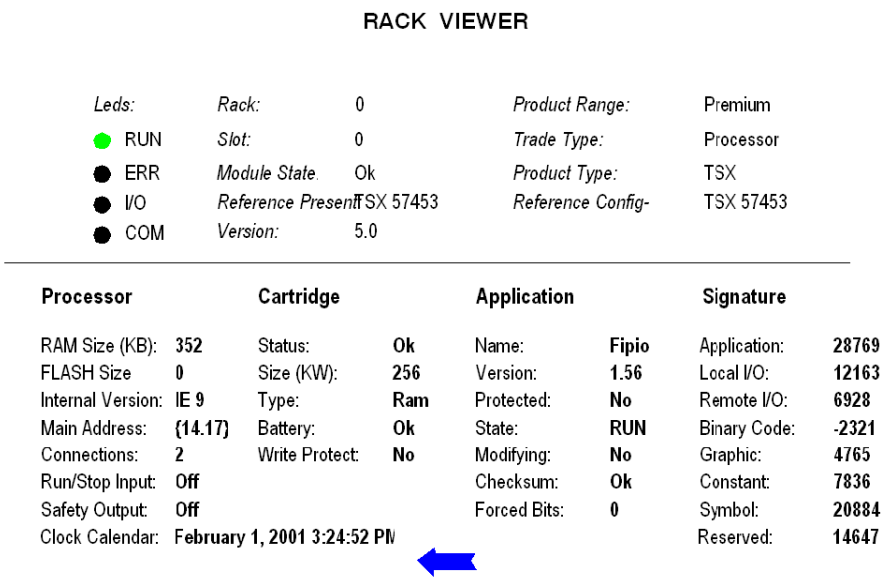
PLC Personality Page

Overview

The **PLC Personality** page provides information about the controller and its configuration.

Sample Page

Here is an example of a **PLC Personality** page.



Dynamic Data

The LEDs in the upper left hand corner of the screen provide a dynamic report on the controller status.

LEDs	Color if On	Meaning if On	Meaning if Blinking	Meaning if Off
RUN	Green	Application running	Stopped	PLC detected fault
ERR	Red	PLC detected fault	Not configured	OK
I/O	Red	I/O event	---	OK
COM	Yellow	Communication interruption	---	OK

Links

The back arrow will take you to the **Rack Viewer** page for this controller.

Ethernet Module Statistics Page

Home Page

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

- Global Data
- I/O scanning *(see page 137)*
- Messaging
- Bandwidth monitoring
- Ethernet module statistics

A link also allows the private MIB source file to be downloaded.

Click on a link to access the desired diagnostics page.

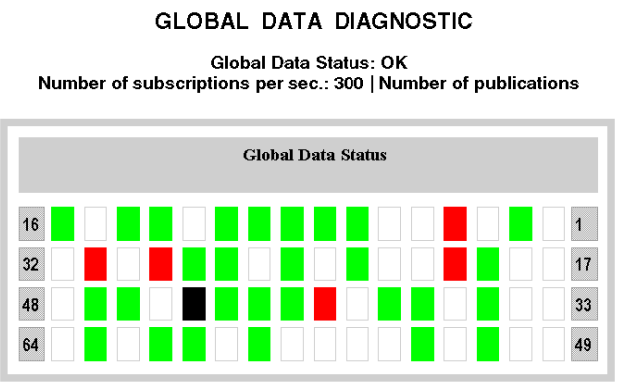
Global Data Page

If you click **Global Data** on the **Diagnostics** home page *(see page 107)*, the following information can be found:

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

This page also shows a table of the published and subscribed variables in the same distribution group. The nature of each variable is identified by a color code:

- green: subscribed variables
- black: published variables
- white: unconfigured variables
- red: variables with detected communication faults



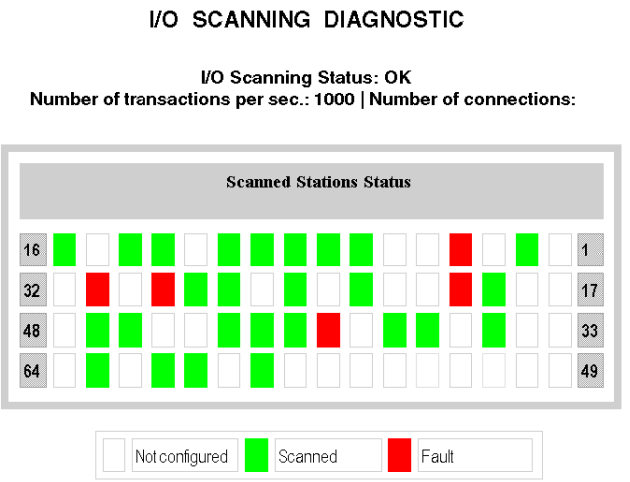
I/O Scanning Page

If you click **I/O scanning** on the **Diagnostics** home page *(see page 107)*, the following information can be found:

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

This page also displays a summary of the status of modules with color coding:

- green for the scanned modules
- white for the unconfigured modules
- red for suspect modules
- black for the modules that are temporarily unscanned.



Messaging Page

The **Messaging** page provides current information on the open TCP connection on port 502.

MESSAGING DIAGNOSTIC

Number of Messages sent: 150 | Number of Messages received: 50

Conn.#	Remote addr.	Remote port	Local Port	Mess. sent	Mess. received	Err. sent.
1	192.160.10.20	1920	502	20	12	0
2	139.160.235.90	2020	502	0	30	02
3	192.160.10.21	502	300	3	60	0
4	139.160.234.20	1050	502	15	42	0
5	139.160.234.18	5120	502	0	39	1

The number of sent/received messages on the port can be found at the top of this page. For each connection (numbered from 1 to 64), a table provides:

- the remote IP address (Remote addr.)
- the remote TCP port (Remote port:)
- the local TCP port (Local Port)
- the number of messages sent from this connection (Mess. sent)
- the number of messages received from this connection (Mess. received)
- the detected error number on this connection (Err. sent)

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.

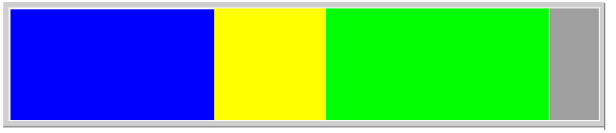
Number of Messages received is not reset after a port 502 connection is closed. Therefore, the count indicates the total number of messages that have been received since the module was started.

The remote address ‘127.0.0.1’ is used as Private System Connection For Diagnostic Feature or SOAP Communications.

Bandwidth Monitoring Page

The **Bandwidth Monitoring** page shows the load distribution of the TSX ETY 4103/5103 module between the Global Data, I/O Scanning, Messaging and other services:

BANDWIDTH MONITOR



Global data: 30% | I/O Scanner: 20% | Messaging: 40% | Others: 10%

Statistics Page

When you click the Embedded Server module in the **Rack Viewer**, you reach the **Ethernet Module Statistics** page. This page provides up-to-date information about the status, configuration, and activity of the Embedded Server module.

Here is an example of an **Ethernet Module Statistics** page.

LEDs:	*Rack:*	0	*Product Range:*	Premium
RUN	*Slot:*	2	*Trade Type:*	Communication
ERR	*Module State:*	Ok	*Product Type:*	Ethernet
STS	*Reference Present:*	TSX ETY 5103	*Reference Configured:*	TSX ETY 5103
	Version:	1.1		

Configuration		**Activity**	
Local IP Address:	**139.158.12.110**	TCP Messaging connections:	**1**
Subnetwork Mask:	**255.255.218.0**	Sent Messages:	**485851**
Gateway Address:	**139.158.8.1**	Received Messages:	**485790**
X-WAY Address:	**{0,0}**	Refused Messages:	**0**
X-WAY Bridge:	**No**	IO Scanning (Msg/s):	**3**
IO Scanner Connections:	**0**		

Back

Dynamic Data

The LEDs in the upper left-hand corner of the screen provide a dynamic report on the Embedded Server module status.

LEDs	Color if On	Meaning if On	Meaning if Blinking	Meaning if Off
RUN	Green	Running normally	---	Power Off
ERR	Red	Module detected error	Not configured	Running normally
STS	Red	Invalid network address or station out of range	---	OK

Links

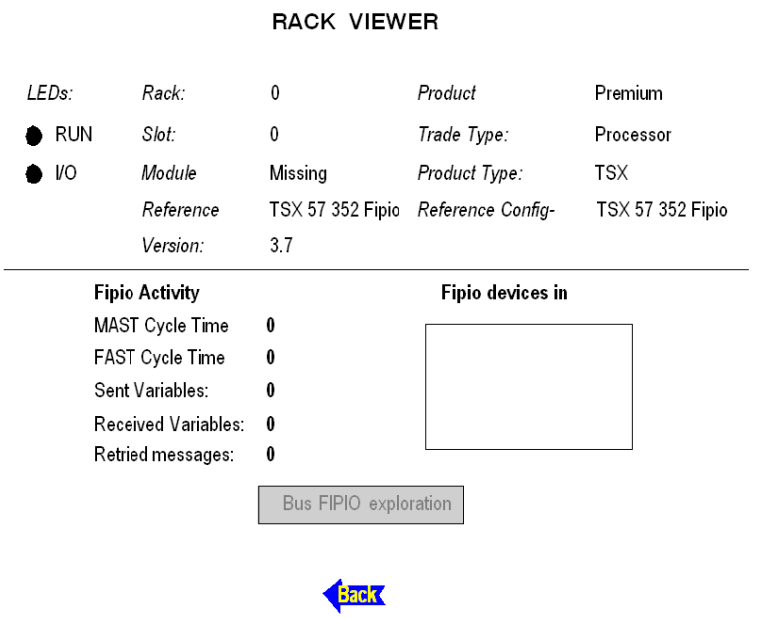
Clicking the back arrow takes you to the **Rack Viewer** page for this controller.

Fipio Module Diagnostics Page

Overview

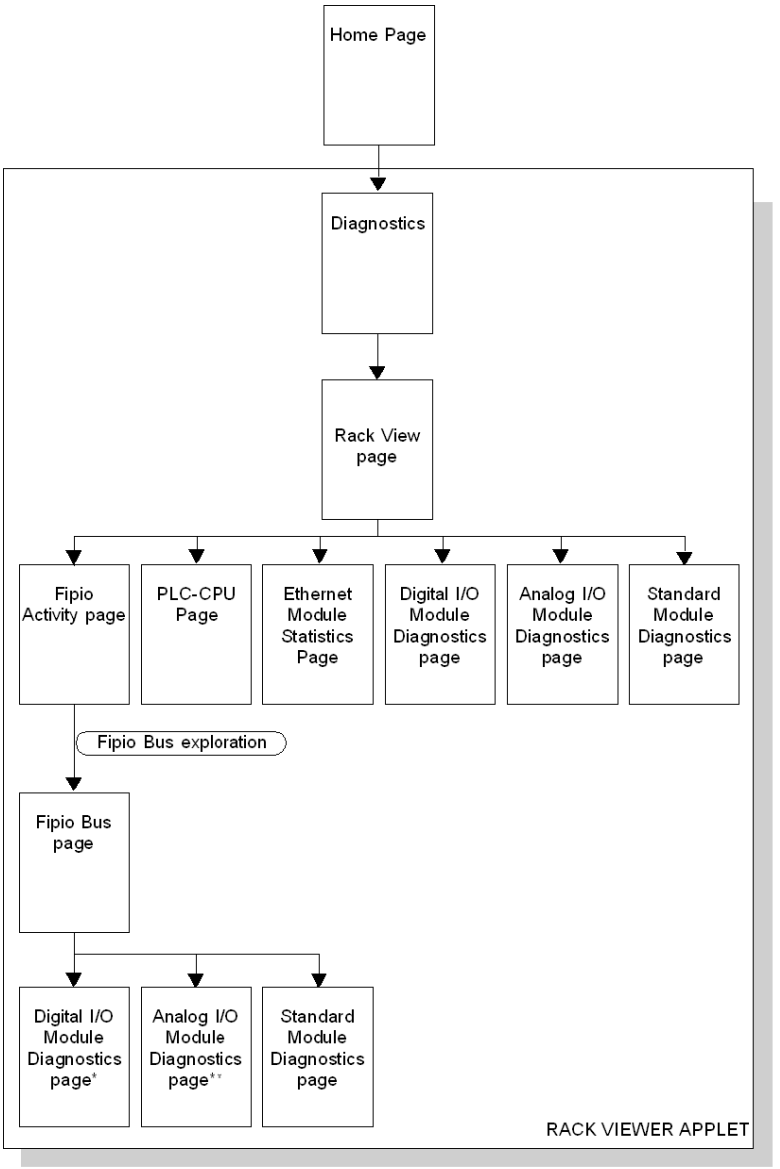
If you click the Fipio link on the controller icon in the **Rack Viewer** page, you will reach the **Fipio Module Diagnostics** page.

This is the **Fipio** page.



Rack Viewer Navigation

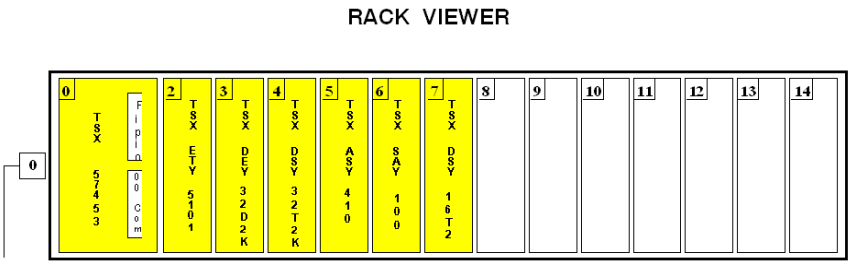
The following illustration represents the navigation page tree of the **Rack Viewer** Applet.



* Only for Digital TBX-7 modules
** Only for Analog TBX-7 modules

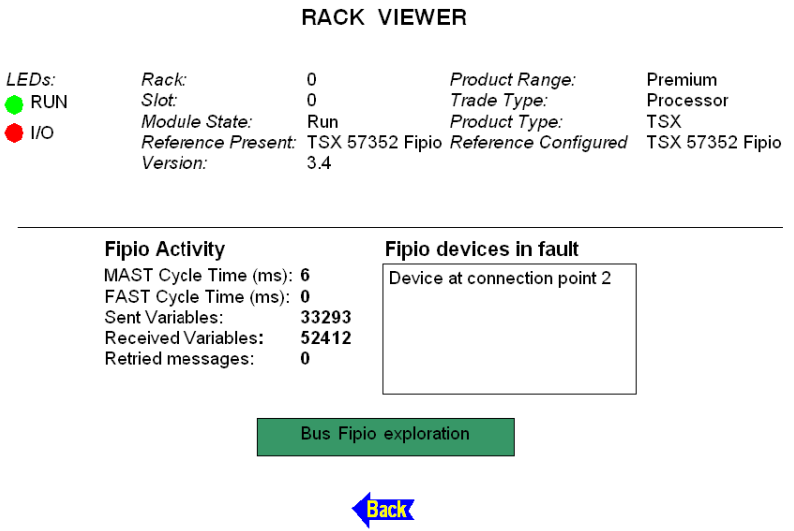
Rack Viewer

Click the **Rack Viewer** link. The applet starts and displays the current configuration of the local rack, including the controller, Embedded Server module, and any I/O modules.



FIP bus link

If Premium has a CPU with FIP bus link, click the **Fipio** link - the first-level **Fipio diagnostic** page is displayed.

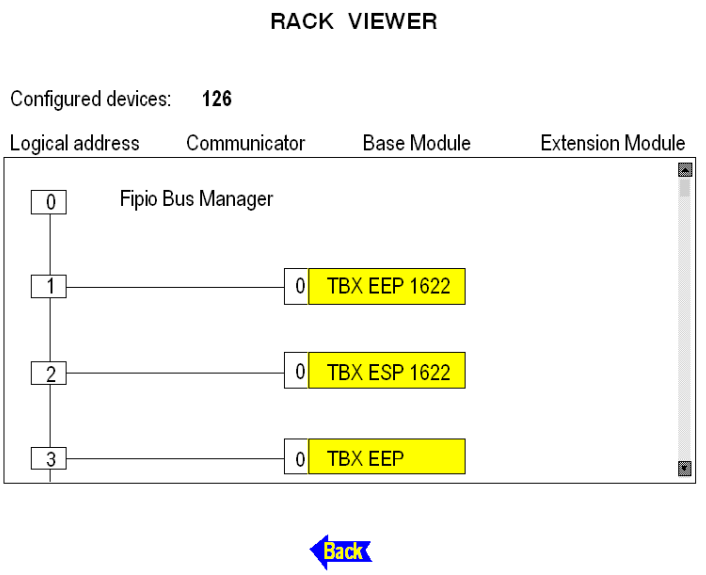


Fipio Bus Page

In FactoryCast Client, there is a button at the bottom of the **Fipio activity** page, named **Bus Fipio exploration**.If you click on this button, the **Fipio Bus** page is displayed.

NOTE: The **Fipio Bus** exploration button is enabled only if a Fipio bus is configured in the PLC application. Otherwise, the button is disabled (grayed out).

This is the **Fipio Bus** Page.



At the top of the page, the number of Fipio devices configured on Fipio Bus is displayed. A scroll window displays the configured devices on Fipio Bus.

A small circle indicates the device connection point. If a Fipio Device is inoperable, the background color of its logical address indicator is red. An inoperable Fipio device is also displayed when the box containing the module number has a red background color.

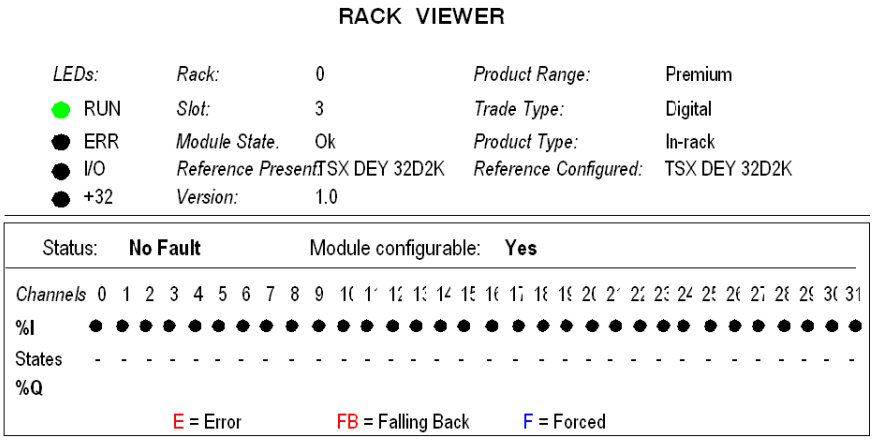
NOTE: An inoperable Fipio device is notified as inoperable both in the **Fipio Bus** page and the **Activity** page.

If the Base Module of the Fipio device is a "modular" type, a **Communicator** is displayed. If the Base Module of the Fipio device is a "compact" type, there is no **Communicator**.

Base Module links and Extension Module links Diagnostic Page

If present, Base Module links and Extension Module links are mouse sensitive

Clicking the module allows you to access its diagnostic page in the same way as an in-rack module.



Types of Diagnostic Pages

On Fipio modules, there are three kinds of diagnostic pages. The display depends on Product Range and of Trade Type.

A *Standard Module Diagnostics page* is displayed except for Digital TBX-7 and for Analog TBS-7 modules.

They are displayed respectively as the **Digital I/O Module Diagnostics** page and the **Analog I/O Module Diagnostics** page.

Dynamic Data

The LEDs in the upper left-hand corner of the screen provide a dynamic report on the Fipio module status.

LEDs	Color if On	Meaning if On	Meaning if Off
RUN	Green	Link active	Link inactive
I/O	Red	Remote device inoperable	Remote device operating normally

Links

The back arrow will take you to the **Rack Viewer** page for this controller.

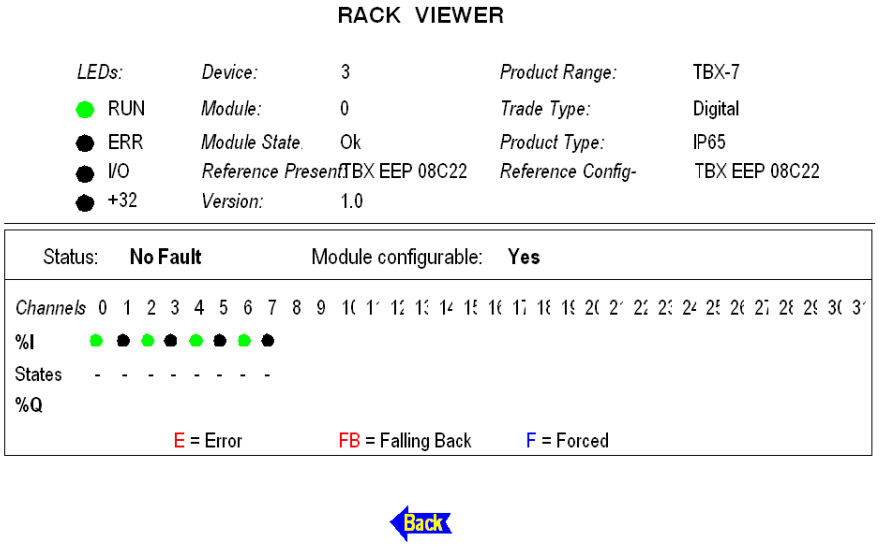
Digital I/O Module Diagnostics Page

Overview

If you click a Digital I/O module in the **Rack Viewer**, you will reach a **Digital I/O Module Diagnostics** page with detailed information about it.

Sample Page

Here is an example of **Digital I/O Module Diagnostics** page.



Dynamic Data

The LEDs in the upper left-hand corner of the screen provide a dynamic report on the module status.

LED	Color if On	Meaning if On	Meaning if Blinking	Meaning if Off
RUN	Green	Running normally	---	Not operating - analysis needed
ERR	Red	Module inoperable	Communication interruption	OK
I/O	Red	Circuit overload or server/preactuator overvoltage detected	Terminal Block detected fault	OK
+32	Green	Channels 32 ... 63 displayed	---	Channels 0 ... 31 displayed

The LED panel in the lower part of the screen provides a dynamic status report for each channel.

Links

The back arrow will take you to the **Rack Viewer** page for this controller.

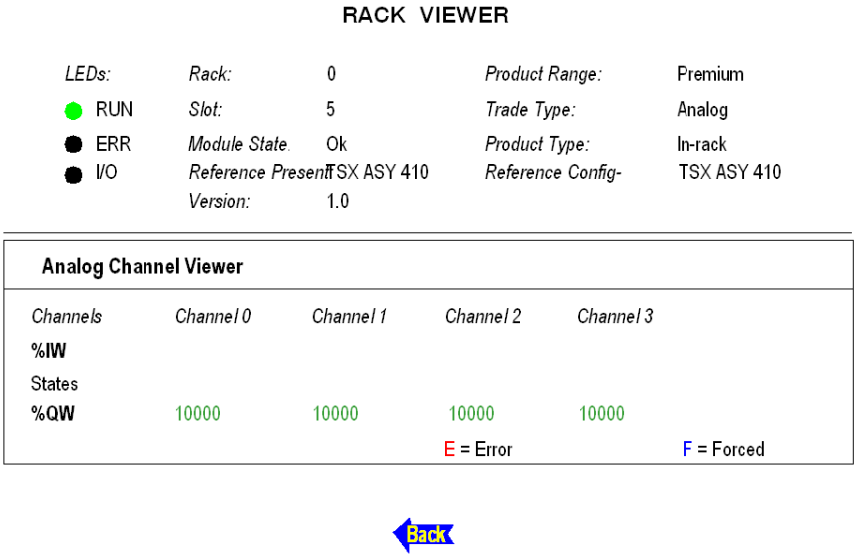
Analog I/O Module Diagnostics Page

Overview

If you click an Analog I/O module icon in the Rack Viewer, you will reach an **Analog I/O Module Diagnostics** page with detailed information about it.

Sample Page

Here is an example of an **Analog I/O Module Diagnostics** page.



Dynamic Data

The LEDs in the upper left-hand corner of the screen provide a dynamic report on the module status.

LED	Color if On	Meaning if On	Meaning if Blinking	Meaning if Off
RUN	Green	Running normally	---	Faulty module or no power
ERR	Red	Module inoperable	Communication interruption with PLC	OK
I/O	Red	Range overshoot or sensor link inoperable	Terminal Block detected fault	OK

Links

The back arrow will take you to the **Rack Viewer** page for this controller.

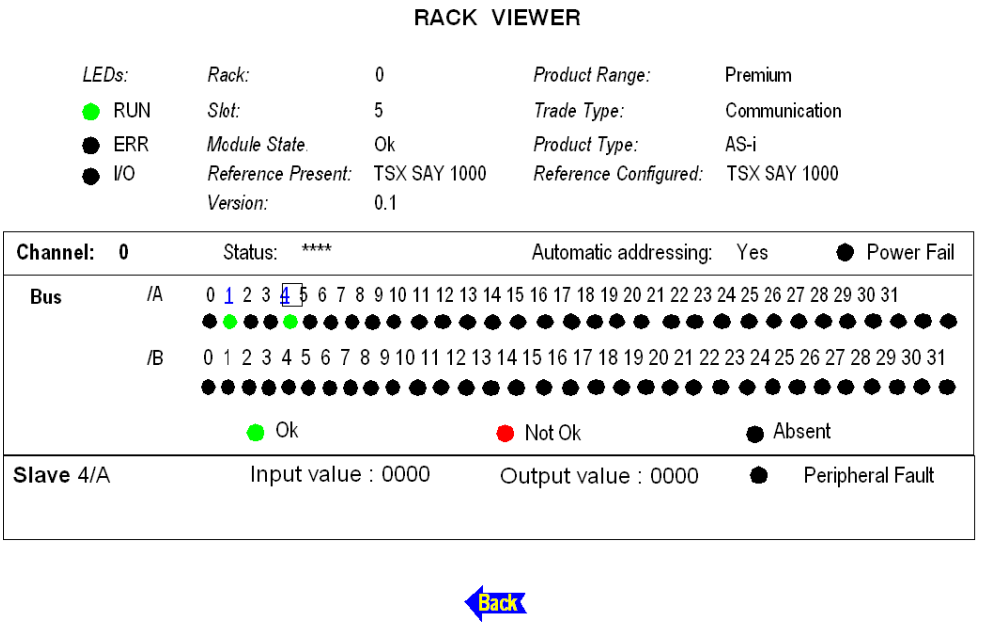
AS-i Module Diagnostics Page

Overview

If you click an AS-i module icon in the **Rack Viewer**, you will reach an **AS-i Module Diagnostics** page with detailed information about it.

Sample Page

Here is an example of an **AS-i Module Diagnostics** page.



NOTE: Click on the green slave LED to display the slave I/O values. When a slave is selected, I/O values and "Peripheral Fault" LED are displayed.

Dynamic Data

The LEDs in the upper left-hand corner of the screen provide a dynamic report on the module status.

LED	Color if On	Meaning if On	Meaning if blinking	Meaning if Off
RUN	Green	Module **OK** and configured	Awaiting configuration	Module out of order
ERR	Red	Serious non-rectifiable Module event	Rectifiable module event (PL7 configuration, AS-i power supply)	Module **OK**
I/O	Red	AS-i bus detected fault	Improper configuration or awaiting user configuration	AS-i Bus **OK**

Links

The back arrow will take you to the **Rack Viewer** page for this controller.

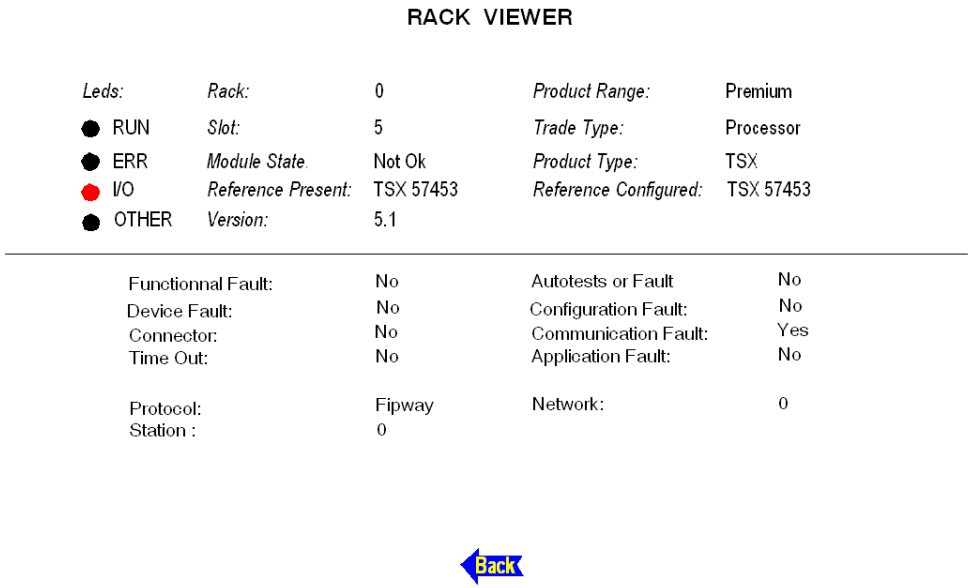
PCMCIA Communication Card Diagnostics Page

Overview

If you click a PCMCIA icon in the Rack Viewer, you will reach a **PCMCIA Communication Card Diagnostics** page with detailed information about that module.

Sample Page

Here is an example of a **PCMCIA Communication Card Diagnostics** page.



Dynamic Data

The following table describes the role of each LED.

LED	Color if On	Meaning if On	Meaning if Flashing	Meaning if Off
RUN	Green	PLC running normally, program executing.	PLC in STOP mode or blocked by software event.	PLC not configured: application missing, invalid or incompatible.
ERR	Red	Processor or system detected event.	• PLC not configured (application missing, invalid or incompatible), • PLC blocked by a software event, • memory card battery detected error, • X Bus detected error.	OK.
I/O	Red	Input/output detected error coming from a module, a channel or an improper configuration.	X Bus detected error.	OK

Links

The back arrow will take you to the **Rack Viewer** page for this controller.

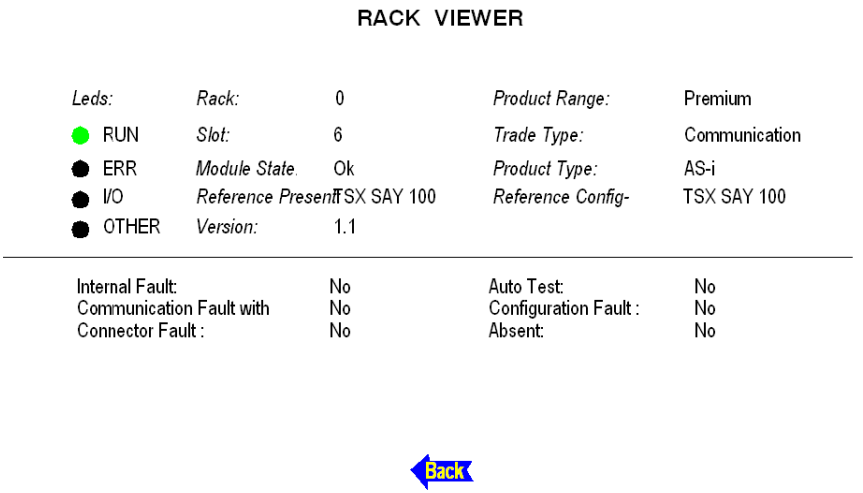
Standard Module Diagnostics Page

Overview

If you click another type of module in the **Rack Viewer** page, you reach a **Standard Module Diagnostics** page with detailed information about that module.

Sample Page

Here is an example of a **Standard Module Diagnostics** page.



Dynamic Data

The LEDs in the upper left hand corner of the screen provide a dynamic report on the module status.

LED	Color if On	Meaning if On	Meaning if Blinking	Meaning if Off
RUN	Green	Running normally	***	***
ERR	Red	Module event	Not configured	OK
I/O	Red	I/O event	---	OK
OTHER	Yellow	***	***	***
*** The meaning depends on the module type. For more information, refer to the user manual for the appropriate module.				

Links

The back arrow will take you to the **Rack Viewer** page for this controller.

4.6 Setup Page

Overview

This section describes the different configuration services proposed by the **Setup** page.

What Is in This Section?

This section contains the following topics:

Topic	Page
Setup Home Page	158
Address Server Page for the HTTP Server	159
Changing the HTTP and Write Passwords	161
Changing the FTP Password	164

Setup Home Page

Home Page

This page lists the various services used to configure the module.

Illustration

The **Setup** home page looks like this :



Links

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

- Address Server,
- NTP *(see page 170)*,
- Email *(see page 168)*.

Address Server Page for the HTTP Server

At a Glance

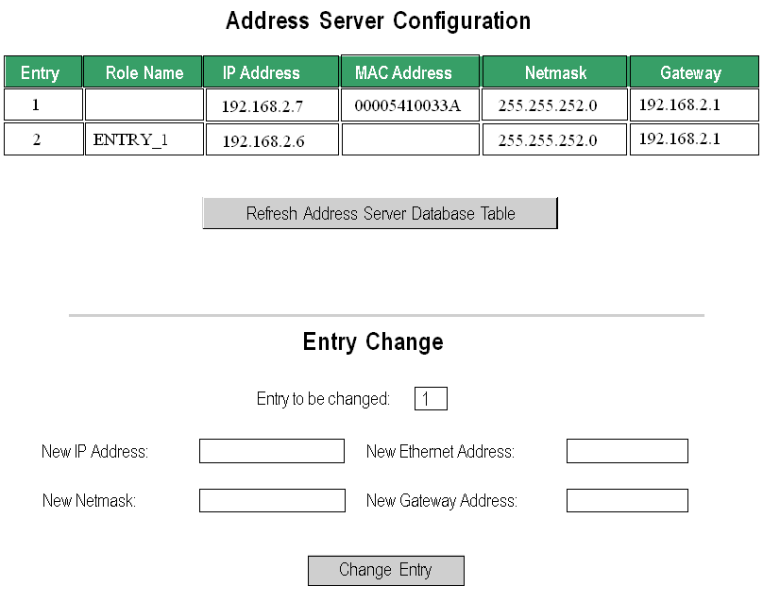
This page is used to display or modify the correspondence table between the MAC addresses or the Name (Role Name) and the IP addresses of the module if the latter is configured as a BOOTP server.

This function is useful when replacing an inoperative remote device.

NOTE: This page does not allow the addition of new inputs, nor the modification of the Role name for a remote device.

Illustration

By way of example, the **Address Server Configuration** page of a TSX ETY4103, a TSX ETY PORT or the Ethernet link of the TSX P57 5634 is as follows.



How to Modify the Table

The procedure is as follows:

Step	Action
1	Enter the entry number to be modified in the field provided.
2	Enter the new IP address to be modified in the field provided.
3	Enter the new MAC address to be modified in the field marked: **New Ethernet Address**.
4	Is the server locked in run mode? ● If so: enter the password associated to the address server then go to step 5. ● If not: go to step 5.
5	Confirm the modification with the **Change Entry** button.
6	Click on the **Refresh Address Server Database Table** button to display the modification on the screen.

Recognition of Modifications

Configuration modifications are recognized either after a cold restart of the PLC, or on the next loading of the PLC application.

Changing the HTTP and Write Passwords

Presentation

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 15 characters (a-z, A-Z and 0-9).

NOTE:

- Access to this page requires a password. The default values for the username/password fields restricting Web page access are **USER**/**USER**.
- The default username and password for FTP are USER/USERUSER.

Accessing the Security 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 **Security** link on the setup page

View of the Password Page

The password editing page looks like this:



Modifying HTTP Access Rights

The procedure is as follows:

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

The procedure is as follows:

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

Use this page 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 restricting Web page access are **USER**/**USER**.

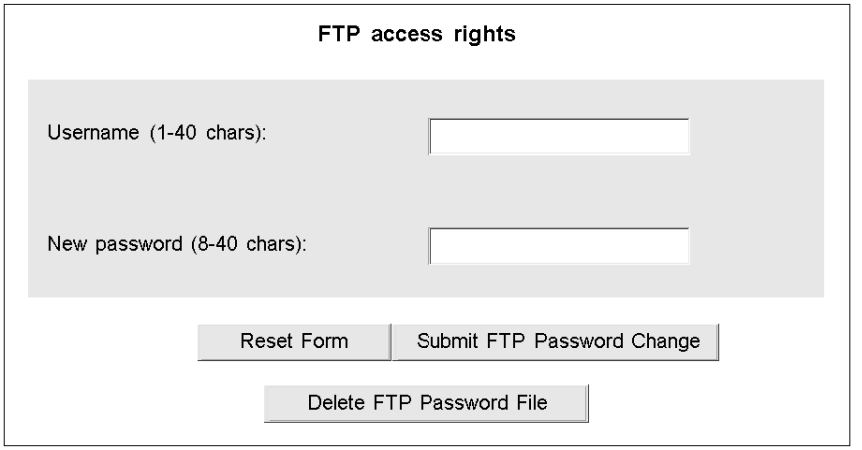
Accessing the FTP Configuration Page

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:



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 **Submit FTP Password Change**. **Result**: a confirmation window appears.

Common Quantum and Premium Setup Page

5

Overview

This chapter describes setup pages that are common to both the Quantum and Premium.

What Is in This Chapter?

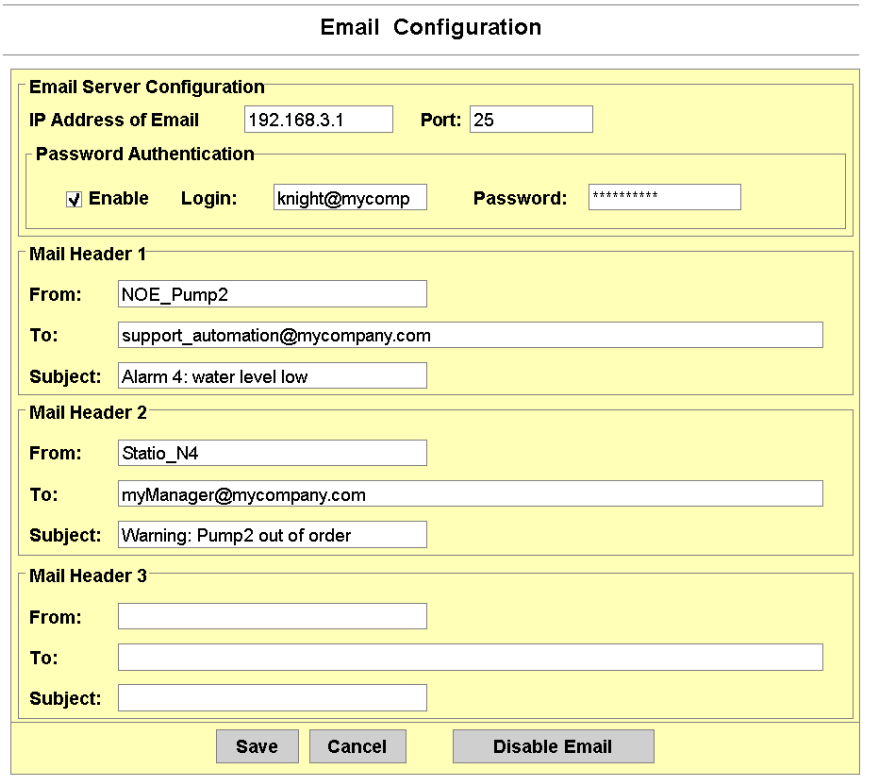
This chapter contains the following topics:

Topic	Page
Mail Service Configuration	168
Configuring the Time Synchronization Service	170

Mail Service Configuration

Configuring the Mail Service with the Email Configuration Page

Use the module's embedded Web page to configure the Electronic Mail Notification service. No other method is available.



Mail Service Command Buttons

Button	Description
Save	Saves the new Email configuration. **Note**: The previous configuration is no longer valid and it is not stored.
Cancel	Cancels the entries in the fields. The previous configuration is valid.
Disable Email	Clears the stored configuration, and disables the email service. **Note**: The next time the service is enabled, a new configuration is required.

Configurable Mail Service Parameters

Parameter	Description
IP address of Email	Enter a valid IP address. (This parameter identifies the SMTP server.)
Port	Default = 25 (If necessary, you may enter a new value.)

Password Authentication	If you want to restrict access, enable **Password Authentication** by entering a check mark in the box.
	Enter values for:
	● Login:
	  ● Any printable character allowed
	  ● 64 character maximum
	● Password:
	  ● Any printable character allowed
	  ● 64 character maximum

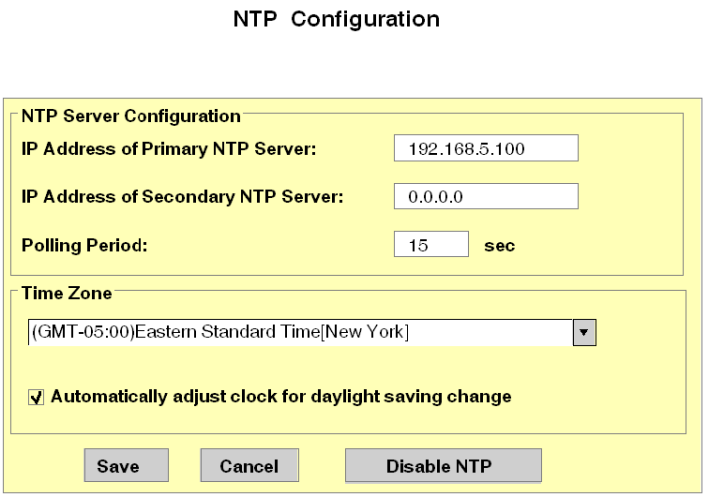
3 mail headers	Each header contains:
	● sender's ID in the **From:** field
	  ● 32 character maximum; no spaces
	● list of recipients in the **To:** field
	  ● Separate each email address with a comma.
	  ● 128 character maximum
	● fixed part of message in the **Subject:** field
	  ● (32 character maximum)

The **Subject** field consists of two parts: **1.** Fixed (32 character maximum) **2.** Dynamic (206 character maximum)	

Configuring the Time Synchronization Service

Configuring the Time Service with the NTP Configuration Page

Use the module's embedded Web page to configure the time service. No other method is available.



Time Service Command Buttons

Execute the following commands

Command Button	Description
Save	Stores new NTP (time service) configuration. Previous configuration is no longer valid.
Cancel	Cancels new NTP (time service) configuration. Previous configuration is valid
Disable NTP	IP of Primary and Standby set = 0. NTP server not polled Time in controller not updated.

Configurable Time Service Parameters

Configure or change the following parameters on the NTP Configuration page.

- IP address of primary NTP server:
  - Enter a valid IP address
- IP address of secondary NTP server:
  - Enter a valid IP address
- Polling Period (in seconds):
  Enter a value
  - min = 1sec
  - max = 120sec
  - default = 5 sec
- Time Zone:
  - Select from drop-down menu
    Universal Time, Coordinated (GMT) = default
  - Custom time zone
- Automatically adjust clock for daylight saving change:
  - Parameter is selected by default (check mark appears) if daylight saving time is chosen.

Changing Time Service Parameters

To make any changes to the time synchronization service:

Step	Action
1	Enter changes in the appropriate field on the NTP Configuration page for one or more of the configurable parameters.
2	Click Save.

Information about the Time Service

NOTE: About the Time Service

- Enable/Disable Daylight Savings Time parameter:
  If the Enable/Disable check box is selected, the module automatically corrects the local time to account for daylight savings time. Therefore, requiring no action, as the daylight savings time start and end are automatically changed each year.
- Polling Time Parameter:
  The time (in seconds) is the time between time updates from the NTP server. The default is 5 Seconds.
- Storing the Time Service Configuration:
  The last time service configuration is saved internally in the Ethernet module.
- Replacing the Ethernet Module:
  If the Ethernet module has to be replaced, the stored configuration is lost, and the system returns to the default configuration.

Customizing Time Zone Parameters

If you want a time zone not listed in the time zone table,

Step	Action	Comment
1	Write the text rules for the custom time zone.	
2	Using an FTP client, store your rules in the file: */FLASH0/wwwroot/conf/NTP/customrules* user ID: ntpupdate password: ntpupdate	Root directory to store 'customrules' is set by the FTP server as */FLASH0/wwwroot/conf/NTP*
3	When the rules are written, choose the drop down menu on the NTP Configuration web page, and configure (or reboot) the module by selecting Time Zone = Custom	The NTP component looks for customrules, calls the tz compiler and generates a new file called 'tz_custom'. This file is binary file and should not be edited. If the tz compiler detects an incorrect syntax in customrules, it is logged in the file: */FLASH0/wwwroot/conf/NTP/error.log* **1.** NTP component is not launched **2.** NTP Status field in diagnostic web page displays NOT OK.
4	If you want more information, the syntax to write those rules along with a few examples are found in the module in */FLASH0/wwwroot/conf/NTP/instructions.txt*	

Time Zone Parameters

Select one from drop-down menu.

Time Zone	Description	DST Available
Custom		Yes
(GMT-12:00)	Dateline Standard Time [Eniwetok Kwajalein]	No
(GMT-11:00)	Samoa Standard Time [Midway Is Samoa]	No
(GMT-10:00)	Hawaiian Standard Time [Hawaii Honolulu]	No
(GMT-09:00)	Alaskan Standard Time [Anchorage]	Yes
(GMT-08:00)	Pacific Standard Time [Los Angeles Tijuana]	Yes
(GMT-07:00)	Mexican Standard Time [Chihuahua La Paz Mazatlan]	Yes
(GMT-07:00)	Mountain Standard Time [Arizona Phoenix]	No
(GMT-07:00)	Mountain Standard Time [Denver]	Yes
(GMT-06:00)	Central Standard Time [Chicago]	Yes
(GMT-06:00)	Mexico Standard Time [Tegucigalpa]	No
(GMT-06:00)	Canada Central Standard Time [Saskatchewan Regina]	No
(GMT-06:00)	Central America Standard Time [Mexico_city]	Yes
(GMT-05:00)	SA Pacific Standard Time [Bogota Lima Quito]	No
(GMT-05:00)	Eastern Standard Time [New York]	Yes

Time Zone	Description	DST Available
(GMT-05:00)	Eastern Standard Time [Indiana (East)] [Indianapolis]	No
(GMT-04:00)	SA Western Standard Time [Caracas La Paz]	No
(GMT-04:00)	Pacific SA Standard Time [Santiago]	Yes
(GMT-03:30)	Newfoundland Standard Time [Newfoundland St Johns]	Yes
(GMT-03:00)	E. South America Standard Time [Brasilia Sao_Paulo]	Yes
(GMT-03:00)	SA Eastern Standard Time [Buenos Aires Georgetown]	No
(GMT-02:00)	Mid-Atlantic Standard Time [South_Georgia]	No
(GMT-01:00)	Azores Standard Time [Azores Cape Verde Island]	Yes
(GMT)	Universal Coordinated Time [Casablanca, Monrovia]	No
(GMT0)	Greenwich Mean Time [Dublin Edinburgh Lisbon London]	Yes
(GMT+01:00)	Romance Standard Time [Amsterdam CopenHagen Madrid Paris Vilnius]	Yes
(GMT+01:00)	Central European Standard Time [Belgrade Sarajevo Skopje Sofija Zagreb]	Yes
(GMT+01:00)	Central Europe Standard Time [Bratislava Budapest Ljubljana Prague Warsaw]	Yes
(GMT+01:00)	W. Europe Standard Time [Brussels Berlin Bern Rome Stockholm Vienna]	Yes
(GMT+02:00)	GTB Standard Time [Athens Istanbul Minsk]	Yes
(GMT+02:00)	E. Europe Standard Time [Bucharest]	Yes
(GMT+02:00)	Egypt Standard Time [Cairo]	Yes
(GMT+02:00)	South Africa Standard Time [Johannesburg Harare Pretoria]	No
(GMT+02:00)	FLE Standard Time [Helsinki Riga Tallinn]	Yes
(GMT+02:00)	Israel Standard Time [Israel Jerusalem]	Yes
(GMT+03:00)	Arabic Standard Time [Baghdad]	Yes
(GMT+03:00)	Arab Standard Time [Kuwait Riyadh]	No
(GMT+03:00)	Russian Standard Time [Moscow St. Petersburg Volgograd]	Yes
(GMT+03:00)	E. Africa Standard Time [Nairobi]	No
(GMT+03:30)	Iran Standard Time [Tehran]	Yes
(GMT+04:00)	Arabian Standard Time [Abu Dhabi Muscat]	No
(GMT+04:00)	Caucasus Standard Time [Baku Tbilisi]	Yes
(GMT+04:00)	Afghanistan Standard Time [Kabul]	No
(GMT+05:00)	Ekaterinburg Standard Time [Ekaterinburg]	Yes
(GMT+05:00)	West Asia Standard Time [Islamabad Karachi Tashkent]	No
(GMT+05:30)	India Standard Time [Bombay Calcutta Madras New Delhi]	No
(GMT+06:00)	Central Asia Standard Time [Almaty Dhaka]	Yes
(GMT+06:00)	Sri Lanka Standard Time [Columbo]	No
(GMT+07:00)	SE Asia Standard Time [Bangkok Hanoi Jakarta]	No

Time Zone	Description	DST Available
(GMT+08:00)	China Standard Time [Beijing Chongqing Hong Kong Urumqi]	No
(GMT+08:00)	W. Australia Standard Time [Perth]	No
(GMT+08:00)	Singapore Standard Time [Singapore]	No
(GMT+08:00)	Taipei Standard Time [Taipei]	No
(GMT+09:00)	Tokyo Standard Time [Osako Sapporo Tokyo]	No
(GMT+09:00)	Korea Standard Time [Seoul]	No
(GMT+09:00)	Yakutsk Standard Time [Yakutsk]	Yes
(GMT+09:30)	Cen. Australia Standard Time [Adelaide]	Yes
(GMT+09:30)	AUS Central Standard Time [Darwin]	No
(GMT+10:00)	E. Australia Standard Time [Brisbane]	No
(GMT+10:00)	AUS Eastern Standard Time [Canberra Melbourne Sydney]	Yes
(GMT+10:00)	West Pacific Standard Time [Guam Port Moresby]	No
(GMT+10:00)	Tasmania Standard Time [Hobart]	Yes
(GMT+10:00)	Vladivostok Standard Time [Vladivostok]	Yes
(GMT+11:00)	Central Pacific Standard Time [Magadan Solomon Is New Caledonia]	Yes
(GMT+12:00)	New Zealand Standard Time [Auckland Wellington]	Yes
(GMT+12:00)	Fiji Standard Time [Fiji Kamchatka Marshall Is]	No

Editors

6

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	Topic	Page
6.1	Data Editor	176
6.2	Data Editor Lite	193
6.3	Pocket Data Editor	195
6.4	Graphic Editor	204
6.5	Graphic Viewer	249
6.6	PLC Program Viewer	251

6.1 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:

Topic	Page
Data Editor	177
Data Editor Spreadsheet	179
Creating a Data Template	185
Saving a Data Template	186
Using an Existing Data Template	187
Inserting a Symbol (Variable) in a Data Template	188
Inserting a Direct Address in a Data Template	190
Modifying Data Values in a Data Template	191

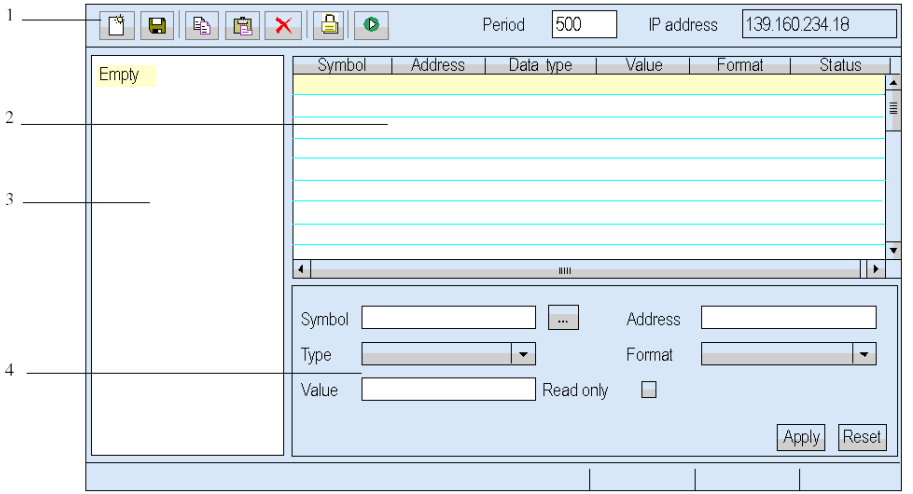
Data Editor

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:



Number	Description
1	Tool bar *(see page 178)*.
2	Data Editor Template *(see page 179)*. 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 188)* and/or modify *(see page 191)* a symbol, ● modify *(see page 191)* a symbol value, ● select *(see page 190)* 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

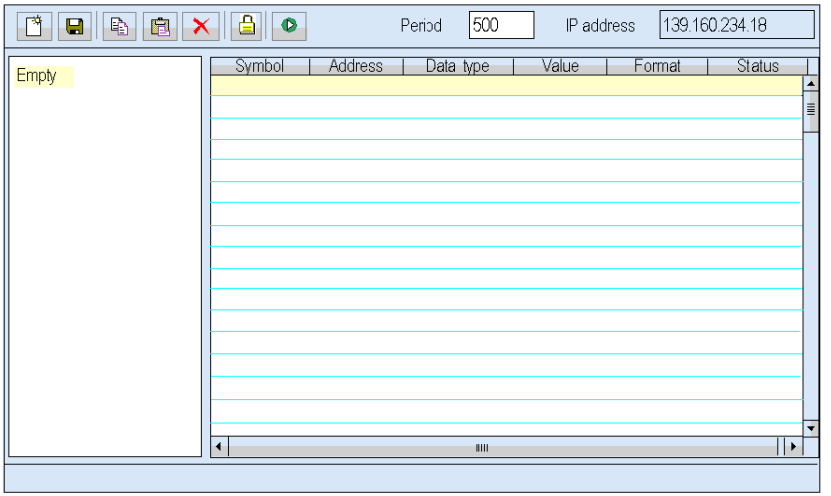
The Data Editor spreadsheet displays data with the following fields:

- **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:



Symbol Field

The **Symbol** column contains the names of Concept, PL7, or Unity Pro 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 Concept, PL7, or Unity Pro 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 Quantum

Valid direct addresses for Quantum are:

- Coils (0x)
- Discreet inputs (1x)
- Input registers (3x)
- Output/holding registers (4x)
- Extended memory registers (6x)

For Unity Quantum PLC, direct addresses also include:

- %Mi (same as for 0X coils)
- %Ii (same as 1x for discreet inputs)
- %IWi (same as 3x for input registers)
- %MWi, %MDi, %MFi (same as 4x for holding registers)
  For a Unity Quantum PLC, a single bit of any word address (for example, %MWi, %IWi) can be specified by appending ".j" to the address, where "j" is a bit index in the range of 0 (LSB) to 15 (MSB). For example, bit 4 of the value at %MW101 would be specified as %MW101.4.
  Also for a Unity Quantum 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, or %MFi 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].

Valid Direct Addresses for Premium

The following table contains the valid direct addresses for Premium, except Unity Premium:

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
%Irs.c	BOOLEAN	R
%Qrs.c	BOOLEAN	R/W
%Mi	BOOLEAN	R/W
%Si	BOOLEAN	R/W
%MFi	REAL 32	R/W
%IWrs.c.i	WORD 16	R
%MWi	WORD 16	R/W
%SWi	WORD 16	R/W
%QWrs.c.i	WORD 16	R/W
%MWrs.c.i	WORD 16	R/W
%MWrs.MOD.i	WORD 16	R/W
%KWrs.c.i	WORD 16	R/W
r = rack number, s = slot number, c = channel number, and i = range number		

Valid Direct Addresses for Unity Premium

The following table contains the valid direct addresses for Unity Premium:

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 Premium, except Unity Premium:

Address	Type	R(ead) or W(rite) Access
%I\rs.2.d\m.c	BOOLEAN	R
%Q\rs.2.d\m.c	BOOLEAN	R/W
%IW\rs.2.d\m.c.i	WORD 16	R
%QW\rs.2.d\m.c.i	WORD 16	R/W
%MW\rs.2.d\m.c.i	WORD 16	R/W
%KW\rs.2.d\m.c.i	WORD 16	R
r = rack number, s = slot number, d = device number, m = module number, c = channel number and i = range number		

The following table contains the valid Fipio addresses for Unity Premium:

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 Unity Premium, 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 Unity Premium 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. Direct address data types are set by the user 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 communicating with the symbol (variable) or direct address, the **Status** column will contain a system message describing the interruption.

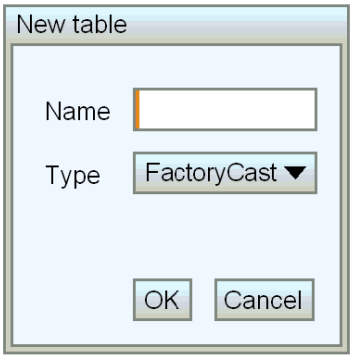
Creating a Data Template

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. 
2	Enter the name of the new Data template.
3	Click **OK**.

NOTE: Save *(see page 186)* the current spreadsheet before selecting a new spreadsheet. Selecting a new spreadsheet deletes the current spreadsheet.

Saving a Data Template

Overview

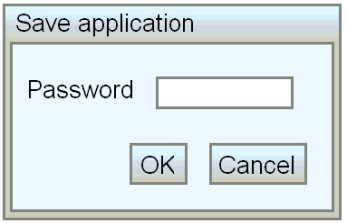
If you save a data template, you can use it again to view or modify the same symbols (variables) or direct addresses.

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**.

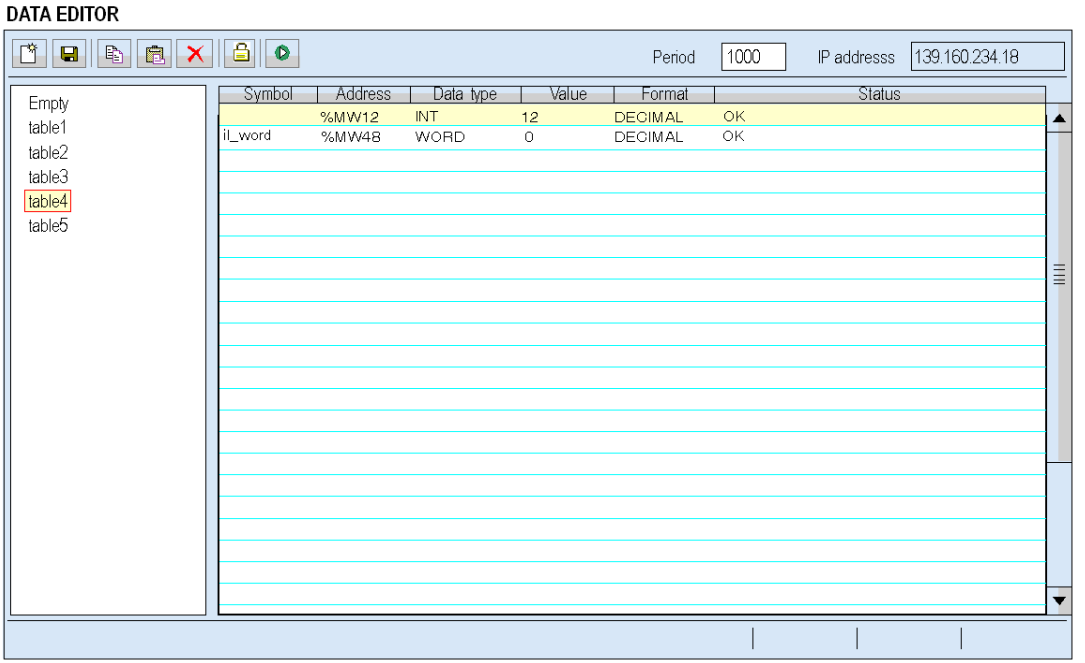
Using an Existing Data Template

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.



Retrieving a Data Template

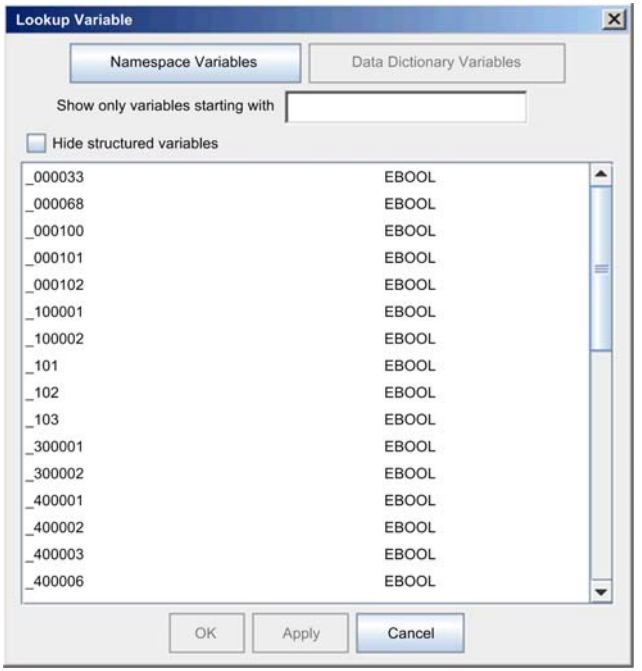
Select the data template you want from the treeview menu. It will appear on a spreadsheet.

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. 
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 Unity Pro so that the data dictionary is embedded in PLC memory *(see page 251)*.

Step	Action
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.
- Password-protect access to 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.

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

Inserting a Direct Address

Step	Action
1	Double-click on 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 on **Apply**. **Result:** A new row that corresponds to the variable address is displayed in the spreadsheet.

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.

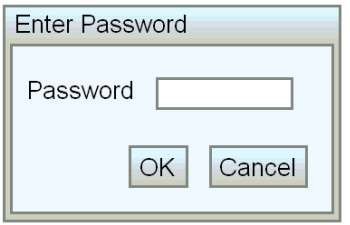
For example, say you have programmed a pushbutton object to jog a motor when the button is depressed and to stop jogging when the button is released. If communications are lost while the button is depressed, the motor will continue to jog even when the button is released. Graphic objects should not be 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

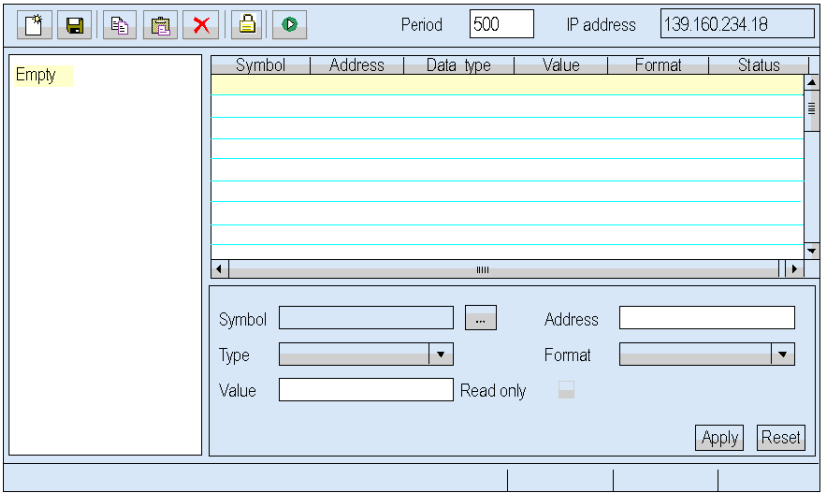
When the mouse is placed on the row containing the variable UNITY, the texts entered in the UNITY are displayed on a tooltip.

6.2 Data Editor Lite

Data Editor Lite

Overview

Data Editor Lite is a version of the Data Editor that is smaller in size and thus faster to download, especially for use via a dialup connection. It provides the same interface.



Variables

Data Editor Lite accepts the following IEC and Concept variables:

Address	Type	Display
%MW IEC internal word	INT	DECIMAL
%MD IEC double word	DINT	DECIMAL
%M IEC internal bits	BOOL	BOOLEAN
400000 Concept variable	INT or DINT	DECIMAL
000000 Concept variable	BOOL	BOOLEAN

Tables

Data Editor Lite can reuse the same tables created with Data Editor and Pocket Data Editor. However, Data Editor tables use a wider range of variable types than 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 176)*.

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

6.3 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:

Topic	Page
Pocket Data Editor	196
Using Pocket Data Editor	198

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 rely upon 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

- 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, edit, and save 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. Because a Pocket PC screen is smaller than a PC screen, adapt your Web site to display the delivered page based on the client's browser type.

Availability

Pocket Data Editor is available on the FactoryCast v3.1 and higher Web server. It can be used with the following modules:

- a TSX ETY 5103 module for Premium
- an NOE 771 11 module for Quantum

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 the following variables:

Address	Type	Display
%MW IEC internal word	INT	DECIMAL
%MD IEC double word	DINT	DECIMAL
%M IEC internal bits	BOOL	BOOLEAN
400000 Concept variable	INT or DINT	DECIMAL
000000 Concept variable	BOOL	BOOLEAN

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 FactoryCast.

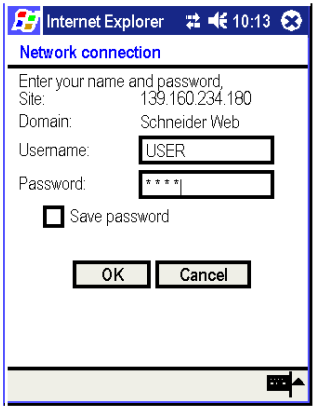
Using Pocket Data Editor

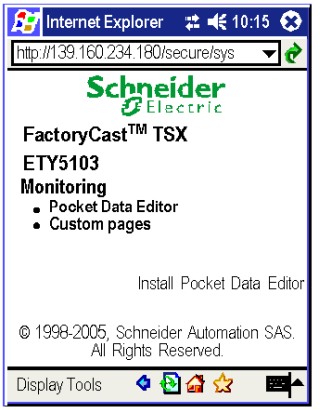
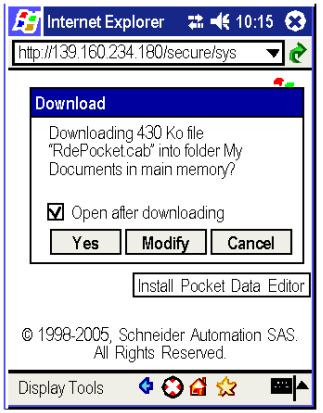
Overview

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

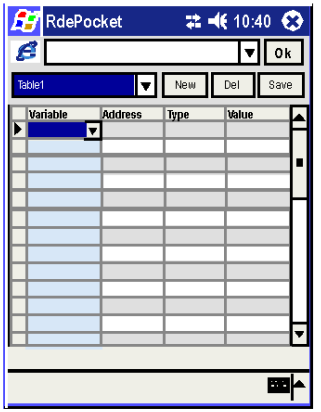
The following table describes the installation of the Pocket Data Editor:

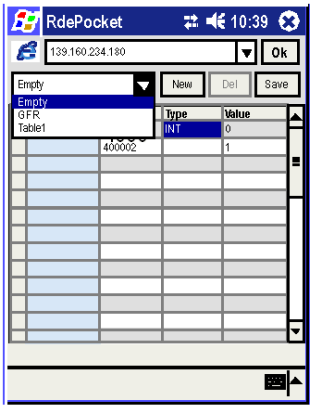
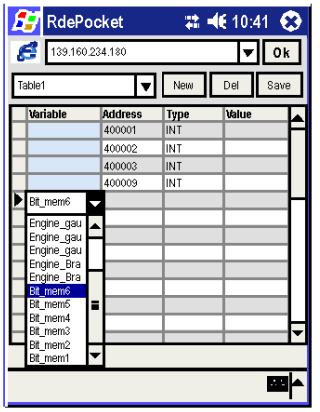
Step	Action
1	Turn on the Pocket PC.
2	Launch Internet Explorer.
3	Enter the IP address of the remote equipment. **Result**: The following window appears: 

Step	Action
4	Enter the username and password. (The defaults are USER and USER.) **Result**: the FactoryCast Server home page appears: 
5	Click on **Install Pocket Data Editor**. **Result**: The following window appears: 
6	Click on **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:  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.

Step	Action
2	Select the desired table via the drop-down menu:  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 197)*.) 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.
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.  You can add new addresses, following the restrictions detailed in the Variables section *(see page 197)*. To modify a PLC variable, select the corresponding **value** cell and enter the new value.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

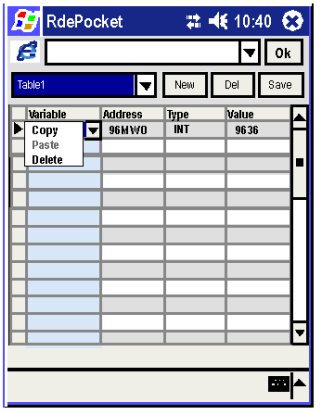
- Password-protect the embedded server.
- Carefully select the symbols and direct addresses you authorize to be modified online.
- Do not authorize online modifications of critical process variables.
- Do not use default or obvious password.
- Limit access to trained personnel.

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**.



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.

6.4 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:

Topic	Page
Graphic Editor Overview	205
Top Window User Functions	208
Display Window User Functions	213
Property Sheet	216
Security	218
Graphic Editor Applet Parameters	219
Graphic Objects	221
Extended Graphic Objects	239

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.



NOTE: Owing to memory restrictions, you can not download both the Data Editor and Graphic Editor on the 140 NOE 211 10 and NOE 251 10 FactoryCast Servers. Before downloading both, first remove an equivalent amount of plug-ins.

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.

User Functions

Most of the Graphic Editor's user functions are available as Top Window User Functions *(see page 208)*. From the display window, you can directly manipulate a graphic object's size and location. The 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 216)*.

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 other dialog panels of the top window is from this dialog.



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

- **Drop-down List**. The drop-down list box shows 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 Concept, PL7, or Unity Pro 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 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:

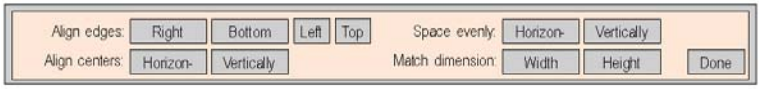


The controls of the **Edit Dialog** provide the following functions:

- 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.
- The **Customize** button displays the currently selected object's Customizer *(see page 215)* (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.



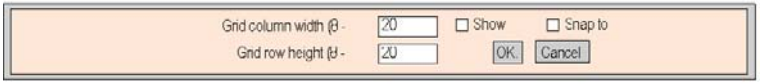
The controls of the **Layout Dialog** provide the following functions:

- 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 the other selected objects adjust for their new position or dimension. For example, when the "Width" button is pressed, the 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**.



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

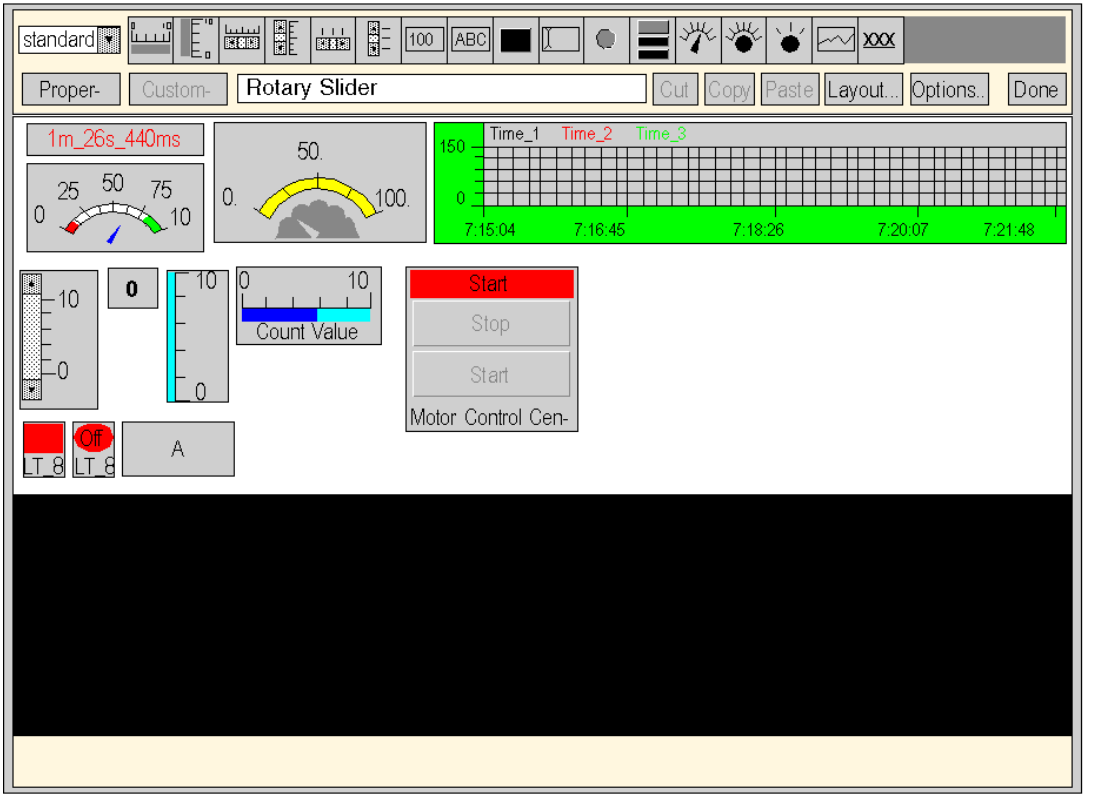
- The cell size of the grid can be changed by the entering the grid's column width and row height in the dialog's text fields.
- If the **Show grid** check-box is checked, the grid is drawn; otherwise, no grid is shown.
- If the **Snap to grid** check-box is checked, then, when you change the size or 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.

Display Window User Functions

Overview

The user functions available from the **Graphic Editor** display window allow you to select, move, and size objects. Moving and sizing operations require that your first select those graphic object(s) that you want to modify. A selected object is indicated by its surrounding selection box; an unselected or deselected object has no surrounding selection box.

The figure below shows the **Graphic Editor** display.



Selecting Graphic Objects

A graphic object's selection state (selected/deselected) can be set via the following user actions:

- A single graphic object can be selected by simply clicking on it with a mouse. If any other objects are currently selected, they will be deselected.
- Multiple graphic objects can be selected with a selection box in the display window. If you press a mouse button in an open area of the display window (i.e., not on a graphic object) and drag the mouse without releasing it, you will see a dotted outline box. One corner of the box is fixed where the mouse button was initially pressed while the opposite corner tracks the current mouse position. When the mouse button is released, every object that intersects the selection box is selected. Objects outside the box are deselected.
- A graphic object's selection state can be toggled between selected and deselected without affecting the selection state of other objects by pressing the CTRL key when clicking on the object. With this action, graphic objects can be individually added or removed from the group of selected objects.
- A graphic object can be selected without affecting the selection state of any other objects by pressing the SHIFT key when clicking on the object. When an object is selected this way, it becomes the *reference object* (see Layout Dialog *Top Window User Functions, page 208*) for the group of selected objects. The primary purpose of this action is to change the reference object in a group of selected objects prior to invoking one of the **Layout** operations.
- Previously selected graphic objects can be deselected by clicking the mouse in an open area of the display window, that is, not on a graphic object.

Sizing Graphic Objects

A graphic object's size can be changed by first selecting it, then using the mouse to change the size of the object's selection box. As you move the mouse over an object's selection box, the mouse pointer changes to reflect the type of sizing operation to be performed. If you press a mouse button while the mouse is over an object's selection box and drag the mouse without releasing it, a dotted outline box appears. When the mouse button is released, the object's size is changed to match the size of the outline. There are eight possible sizing actions depending on which part of an object's selection box is dragged. Each corner of the box will allow only its adjacent sides to move; each side of the box will allow only that side to move.

Moving Graphic Objects

A graphic object can be moved in the display window with the mouse. If you press a mouse button while the cursor is over an object and drag without releasing the button, then a selection box will be shown. When the mouse button is released, the object moves to the location of the selection box.

Multiple graphic objects can be moved by first selecting the objects to be moved, and then dragging the entire group of objects in the same way a single object is moved. While a group of objects is moved, a selection box is shown for each object in the group.

Setting Graphic Object Properties

You can set a graphic object's properties via the Property Sheet. If the Property Sheet is visible, the properties of the selected graphic object are presented for editing. Display the Property Sheet by pressing the **Properties...** button or by double-clicking the mouse anywhere on the selected object in the display window.

Customizing Complex Graphic Objects

Some complex graphic objects have a very large number of properties. Configuring such an object with the Property Sheet can be cumbersome. An available Customizer can ease the configuration of complex graphic objects. A Customizer is a dialog window designed specifically to configure its associated graphic object. When the Graphic Editor detects that a selected graphic object has a Customizer, it will enable the **Customizer...** button, which brings up the graphic object's Customizer. When you double-click on a graphic object that is associated with a Customizer, the Customizer comes up (instead of the Property Sheet). If a graphic object is associated with a Customizer, the only item in the Property Sheet is its name.

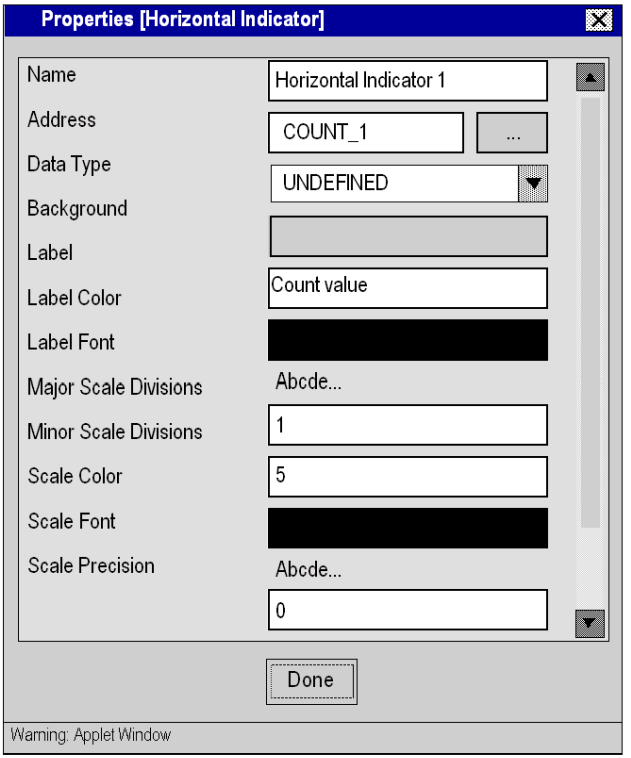
Display Background Image

A Graphic Editor display has a **Background Image** property that can be used to designate an image as the display's background. The image can be a GIF or JPEG file. File locations are relative to the Embedded Server's /wwwroot directory. For example, if the image "cool.gif" was put in the Embedded Server's /wwwroot/images directory, then the Background Image property should be set to /images/cool.gif.

Property Sheet

Overview

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

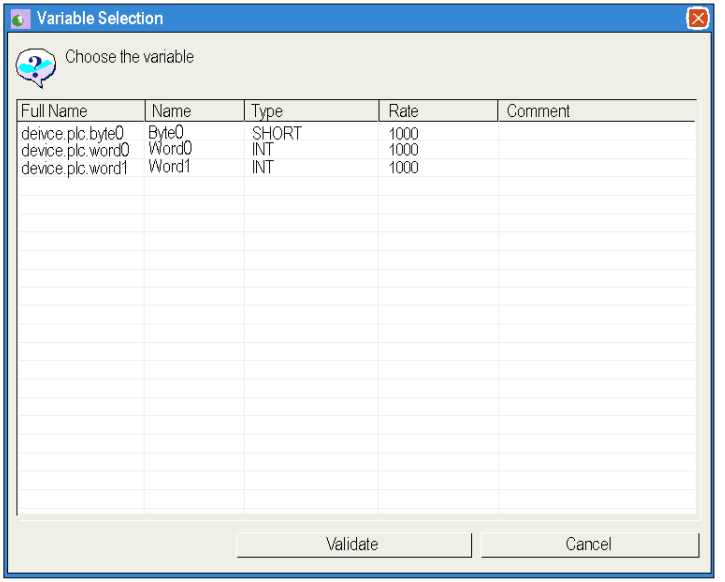


The properties of a graphic object are specific to an object's type. The properties are presented in a scrollable list, with the name and the value of each property listed. The Graphic Editor comes with a description of graphic objects *(see page 221)*.

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 Quantum/Premium register (or Concept/PL7/Unity Pro variable name), but also provides access to the **Lookup Dialog**. The Lookup Dialog allows you to pick a Concept/PL7/Unity Pro symbol (variable) name from a list of symbol (variables) that have been "Web enabled" by Web Designer for FactoryCast.

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



Security

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

- The HTML page containing the Graphic Editor applet has been placed in the *secure* directory on the Web module, then the Web browser user is asked for a password before being allowed 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 FactoryCast allows you to specify that an item is read-only. The **Graphic Editor** will enforce the read-only attribute of a symbol (variable) or address by rejecting any request to set a new value for the data item, and informing the user in the **Graphic Editor** 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 238*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 238*
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 238*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 238*
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)	
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	

Property	Description	Limits
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 238*

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 238*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 238*
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)	

Property	Description	Limits
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 238*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 238*
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 238*

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 238*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 238*
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 238*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 238*
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 238*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 238*
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 238*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 238*
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	

Property	Description	Limits
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 238*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 238*
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 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 238*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 238*
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 238*

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 238*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 238*
Background	The background color of the graphic object	
Values	The value(s) to send to the PLC	See Note 4, *Notes, page 238*
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.	

Property	Description	Limits
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 238*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 238*
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 238*

Property	Description	Limits
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 238*
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 238*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 238*
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

Property	Description	Limits
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 238*

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 238*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 238*
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 238*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 238*
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	

Property	Description	Limits
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 238*

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 238*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 238*
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

Property	Description	Limits
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 238*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 238*
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 238*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 238*
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 238*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 238*
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 238*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 238*
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 248*
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 248*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 248*
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 248*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 248*
Bar Background	The background color of the bar indicator area	
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	

Property	Description	Limits
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 248*

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 248*
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 248*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 248*
Background	The background color of the graphic object	See Note 1, *Notes, page 248*
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 248*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 248*
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 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

Property	Description	Limits
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 248*

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 248*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 248*
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	
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	

Property	Description	Limits
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 248*

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 248*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 248*
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	
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 248*

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 248*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 248*
Background	The background color of the graphic object	
Values	The value(s) to send to the PLC	See Note 4, *Notes, page 248*
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 248*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 248*
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 248*

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>2.</td><td colspan="2">The meaning of the possible values of the Data Type property are:</td></tr>
<tr><td></td><td>Data Type</td><td>Meaning</td></tr>
<tr><td></td><td>UNDEFINED</td><td>no data type specified</td></tr>
<tr><td></td><td>BOOL</td><td>1-bit discreet (Boolean)</td></tr>
<tr><td></td><td>SHORT</td><td>8-bit signed integer</td></tr>
<tr><td></td><td>USHORT</td><td>8-bit unsigned integer</td></tr>
<tr><td></td><td>INT</td><td>16-bit signed integer</td></tr>
<tr><td></td><td>UINT</td><td>16-bit unsigned integer</td></tr>
<tr><td></td><td>DINT</td><td>32-bit signed integer</td></tr>
<tr><td></td><td>UDINT</td><td>32-bit unsigned integer</td></tr>
<tr><td></td><td>REAL</td><td>32-bit IEEE floating point</td></tr>
<tr><td></td><td>TIME</td><td>32-bit unsigned integer (in milliseconds)</td></tr>
<tr><td></td><td>DATE</td><td>Date (32-bit BCD)</td></tr>
<tr><td></td><td>TOD</td><td>Time of Day (32-bit BCD)</td></tr>
<tr><td></td><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>

6.5 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.

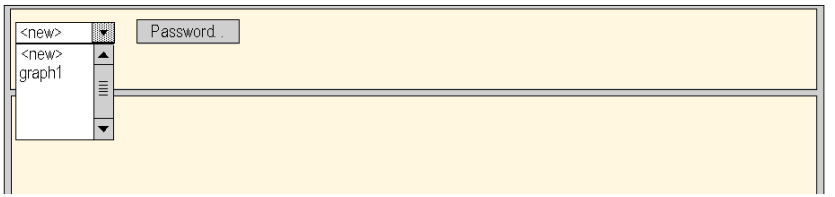
Graphic Viewer is divided into two windows:

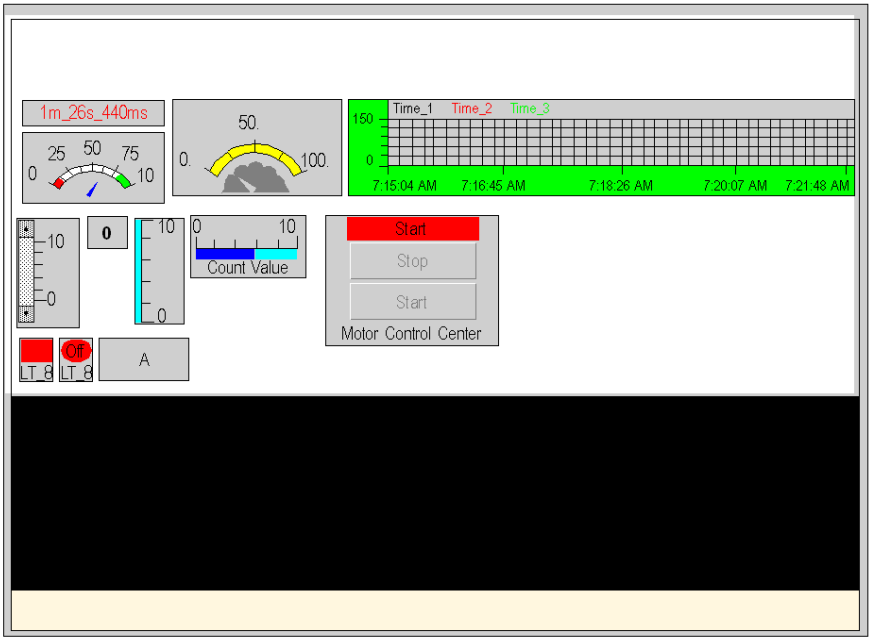
- **Top Window**: This area shows the user controls. This is where you can select a graphic to edit from a drop-down menu
- **Display Window**: This area displays the selected graphic.

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:

Step	Action
1	Click on **Graphic Viewer**. The **Top and Display Windows** appear: <new> <new> graph1 Password..

Step	Action
2	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**.

6.6 PLC Program Viewer

PLC Program Viewer

Presentation

The PLC program viewer feature enables you to monitor UnityPro 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)
- 984 Ladder Logic (LL984)
- 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 4.6, you can configure your Unity Pro 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.

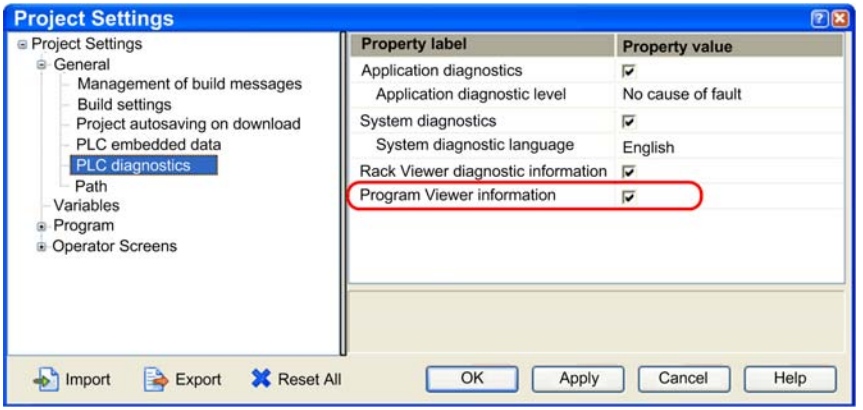
To enable automatic synchronization of the Unity Pro application running in the PLC with the **PLC Program Viewer** display, you need to:

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

Use Unity Pro to enter these settings, as described below.

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

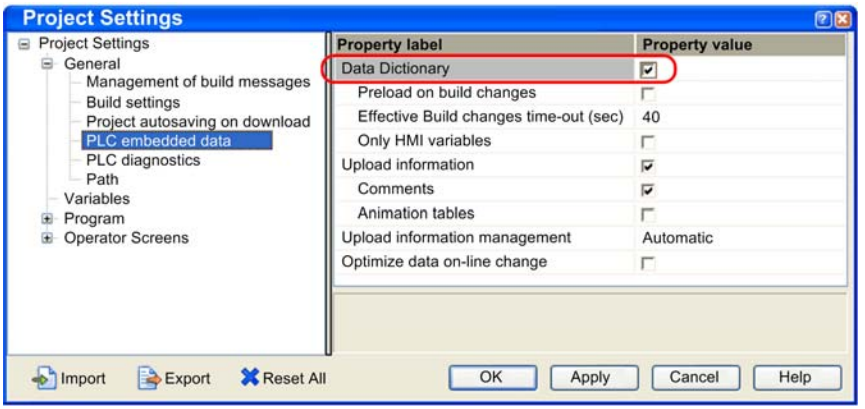
Step	Action
1	In Unity Pro, select **Tools** → **Project Settings...** to open the **Project Settings window**.
2	Select **Project Settings** → **General** → **PLC diagnostics** in the tree control on the left side of the window.
3	Select **Program Viewer information** as displayed below, then click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



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 Unity Pro, 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**.



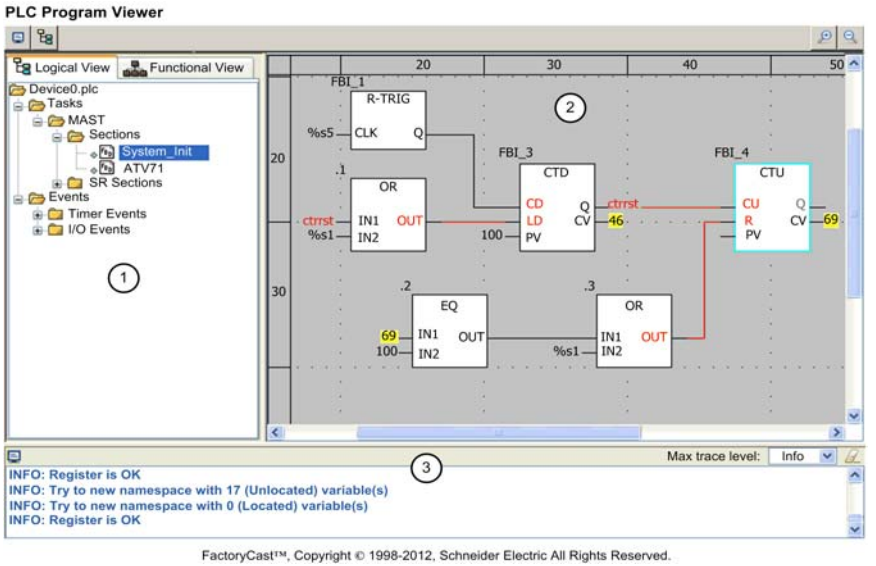
Accessing the PLC Program Viewer

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

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:



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

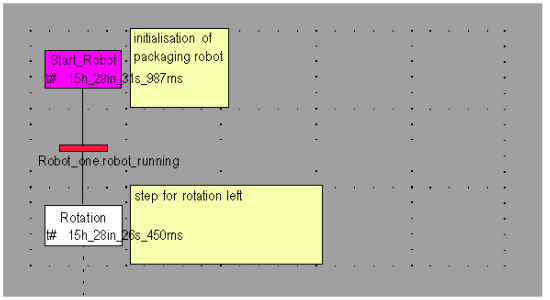
Variables animation:

- 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.

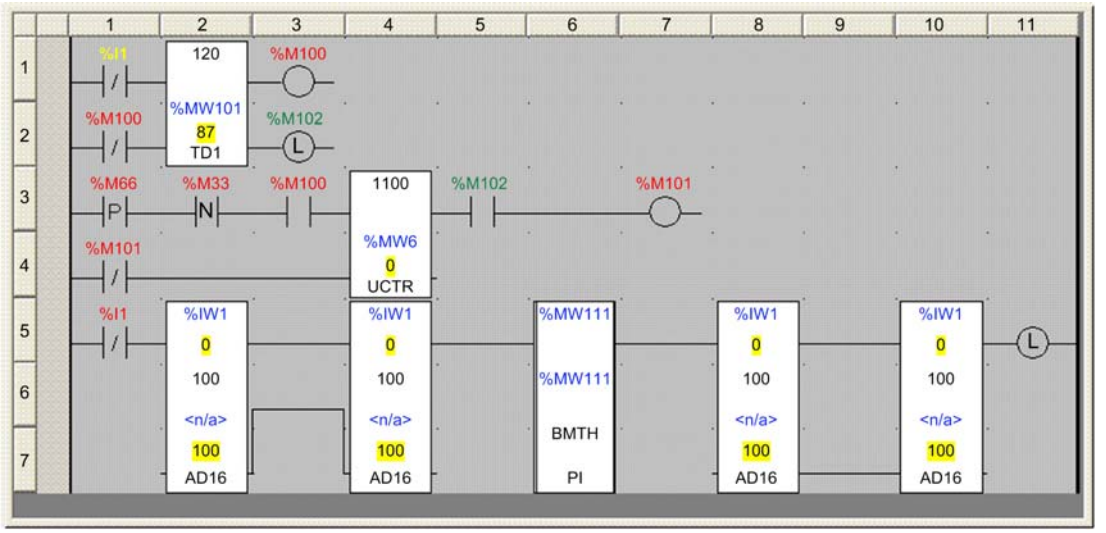
SFC animation:



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 UnityPro 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

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.

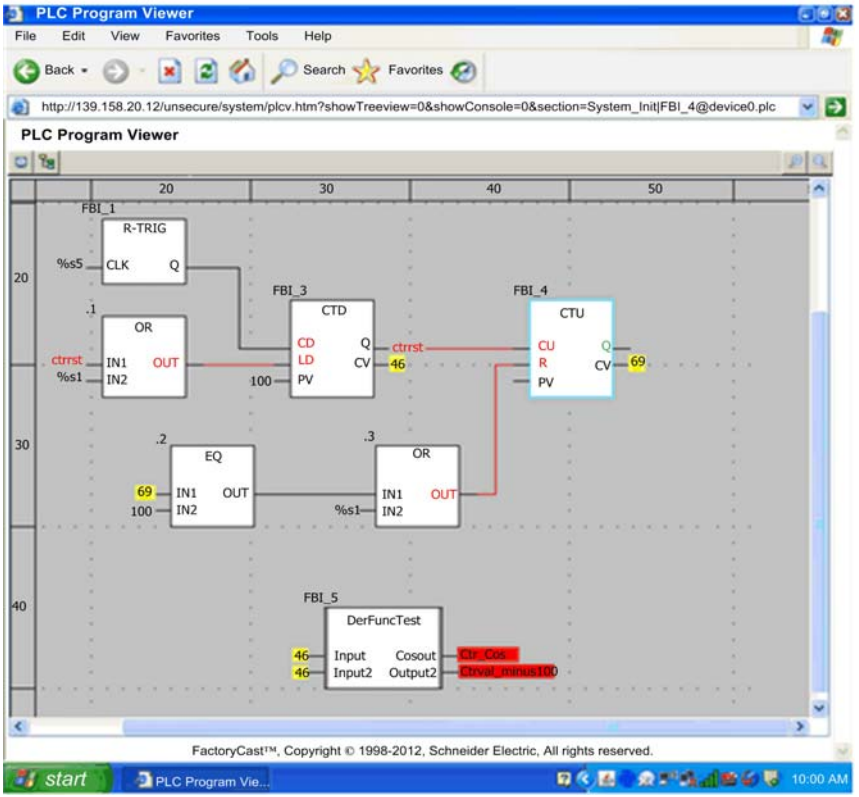
Additionally, you can configure parameters contained in the URL to show or hide the Treeview control (at the left of the PLC program viewer), to show or hide the Console (at the bottom of the PLC program viewer), and to focus on a specific section or object in the PLC program. These parameters are:

- showTreeview: 0 to hide the treeview, 1 to show it.
- showConsole: 0 to hide the console, 1 to show it.
- section: access to a section or to an object in a section. The applets directly display the program 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:



Adding Custom Pages to the Site

7

Overview

You may choose to add your own Web pages to the site on the embedded Server. Web Designer for FactoryCast allows you to use the same passwords for your 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 Web site and if you reach the limits of the memory, some pages of the Web site will not appear.
- If you are using Web Designer for FactoryCast to transfer a Web site, it will prompt you if your website exceeds the memory capacity.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
7.1	Working with Custom Pages	260
7.2	Using FactoryCast's Java Applets Using HTML Code	264
7.3	Using FactoryCast Extension for Microsoft FrontPage or Expression Web	279
7.4	Using Graphic Objects Lite	306
7.5	Adding Microsoft Silverlight® Pages	325

7.1 Working with Custom Pages

Overview

You may develop any pages you want in order to customize the FactoryCast site. Web Designer for FactoryCast gives you two different methods for adding these pages to the site:

- You may add supporting pages and give them password protection.
- You may add supporting pages and make them available to users.

What Is in This Section?

This section contains the following topics:

Topic	Page
Downloading a Custom Home Page	261
Downloading Supporting Pages	263

Downloading a Custom Home Page

Introduction

If you want to replace the default home page with one of your own, follow these steps:

Step	Action
1	Back up the original configuration, so that you can restore it later if needed.
2	Create your home page.
3	Place your home page in its location.
4	Download your home page to the Embedded Server

Backing Up the Original Configuration

Before you change the default home page, you should back up the configuration. It's permit to restore the server to its original configuration.

Creating Your Home Page

Give your home page the same name as the default home page: index.htm.

Placing Your Home Page

Copy your home page to:

- the *wwwroot**secure**user*\ directory for custom pages with password protection.
- the *wwwroot**unsecure**user*\ directory for custom pages without password protection.

What About Image Files?

Any image files for your new home page should be placed in the same directory as your company logo.

Only create filenames that comply with the DOS 8.3 naming convention: a filename that does not exceed eight letters, and a three-letter extension.

Downloading

Follow the steps in the table below to download your new home page to the Embedded Server:

Step	Action
1	Select **Transfer | Web Files** from the menu bar. **Result**: The **Confirm Host** dialog box appears with the Embedded Server's IP address and any host name. **Note:** If the IP address or host name is incorrect, make the appropriate changes in the **IP Address** dialog box.
2	Click **OK**. **Result:** The **Transfer Web Files** dialog box appears.
3	Confirm that the right-hand window displays the Embedded Server's *\wwwroot* directory. Locate and select the new home page in the left-hand window.
4	Click the → button to transfer the home page, overwriting the index.htm file in the Embedded Server's *\wwwroot* directory.

Downloading Supporting Pages

Overview

If you are adding supporting Web pages to the default Web site, you may elect to restrict access to them using the same user name and password as the default pages, or you may choose to make them accessible without a password.

In order to add pages to the site, specify the directory where they are located and download them to the server.

Specifying File Locations

Place the files for your Web pages and any images for the pages in a single directory. Each subdirectory should have a page named `index.htm`.

Use filenames that comply with the DOS 8.3 naming convention (a filename of no more than eight letters and its three-letter extension).

Use the **Settings** dialog box to specify the files' location.

Download Options

Copy pages that are to be password protected to the Embedded Server's *\wwwroot\secure\user* directory.

NOTE: Remember to create new hyperlinks to the other Web pages so you can access them in the browser.

Downloading to a Server

Follow the steps below to download to the server:

Step	Action
1	Select **Transfer | Web Files** from the menu. **Result:** The **Confirm Host** dialog box appears with the Embedded Server's IP address and any host name. **Note:** If the IP address or host name is incorrect, make the appropriate changes in the **IP Address** dialog.)
2	Click **OK** to confirm the IP address and begin the download. **Result:** The **Transfer Web Files** dialog appears.
3	Confirm that the window on the right shows the desired directory in the Embedded Server. Locate and select the new pages in the left-hand window.
4	Click the → button to transfer the pages to the Embedded Server.

7.2 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:

Topic	Page
Inserting Applets on a Web Page	265
Inserting LiveBeanApplet	266
Inserting LiveBeanApplet using HTML Code	269
Inserting LiveLabelApplet using HTML Code	272
Inserting LiveTableApplet using HTML Code	277

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 264)*
- Microsoft's FrontPage Extension *(see page 284)*

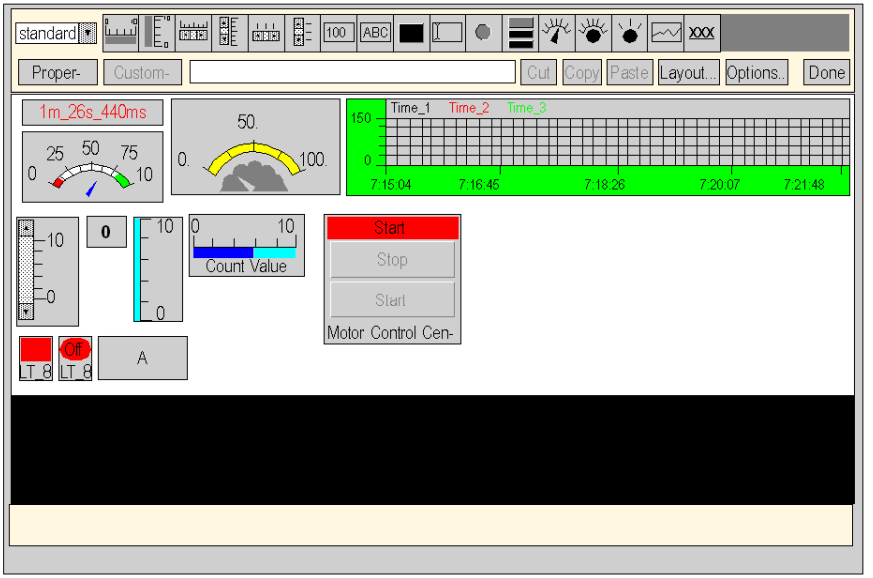
NOTE: Before inserting a `LiveBeanApplet` into a Web page, create a JavaBeans library using the Graphic Editor that comes with the FactoryCast software. Generally you should create a JavaBeans library that has one instance of every object that you want to use in a Web page. Think of this library as a 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.



Step	Action
2	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 204)*.
3	Once you have created your library, click the **Done** button in the editor window and save your library.

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. Include this applet 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">
</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 204)* 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, include the **width** and **height** attributes of the the <APPLET> tag for a `LiveBeanApplet`. 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

Each instance of `LiveBeanApplet` that is included in a Web page follows 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 needs to 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. Include this applet 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.schneiderauto-
mation.factorycast.LiveLabelMgrApplet"
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/Unity Pro 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 Unity Pro symbol, the DATATYPE parameter is optional. If the DATATYPE is specified for a symbol, it needs to exactly match its actual data type. TIME is not a valid data type for PL7 Premium.

Parameter...	Defines...		With Default Value of...
FORMAT	The display format for the value. Acceptable values for this parameter are		DEC for most data types TIME for data type TIME BOOL for data type BOOL DATE for data types DATE, TOD and DT.
	DEC	decimal	
	HEX	hexadecimal	
	BIN	binary	
	ASCII	bytes displayed as ASCII characters	
	TIME	'day_hr_min_sec_ms'	
	DATE	'yyyy-mm-dd-hh and /or hh:mm:ss	
	BOOL	ON_WORD or OFF_WORD (see below)	
NOTE: If DATA TYPE is REAL, a FORMAT other than DEC will give unpredictable results if the value cannot be converted to an integer.			
GAIN	The gain (multiplier) used for scaling the retrieved value to engineering units.		1.0
NOTE: 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			
BIAS	The bias (offset) used for scaling the retrieved value to engineering units. See NOTE for GAIN.		0.0
ON_WORD	A text value to be shown when value is non-zero (Use only if the FORMAT is BOOL).		ON
OFF_WORD	A text value to be shown when value is zero (Use only if the FORMAT is BOOL).		OFF
FOREGRND	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.		BLACK
BACKGRND	Background color for the applet. For acceptable values, see FOREGRND.		LT_GRAY
ERROR_ COLOR	Foreground color of the VALUE field when unable to retrieve the value from the PLC. For acceptable values, see FOREGRND.		MAGENTA

Parameter...	Defines...	With Default Value of...
LABEL_ ALIGN	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.	LEFT
VALUE_ ALIGN	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.	LEFT
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>
```

7.3 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:

Topic	Page
Installing FactoryCast Extension for Microsoft Expression Web	280
Installing FactoryCast Extension for Microsoft FrontPage	284
Inserting LiveBeanApplet Using FrontPage or Expression Web	290
Inserting LiveLabelApplet Using FrontPage and Expression Web	298
Inserting LiveTableApplet Using FrontPage or Expression Web	302

Installing FactoryCast Extension for Microsoft Expression Web

Overview

The following procedures describe how to install/remove the FactoryCast extension for Microsoft Expression Web (MS Expression Web).

Installing FactoryCast

MS Expression Web installed before Web Designer for FactoryCast:

During the installation of Web Designer for FactoryCast, 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 FactoryCast:

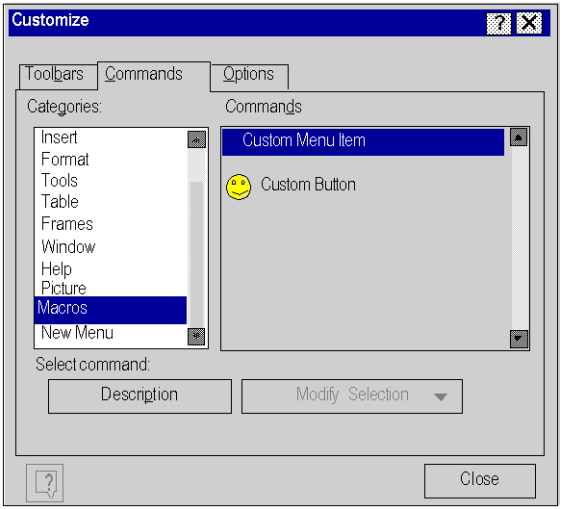
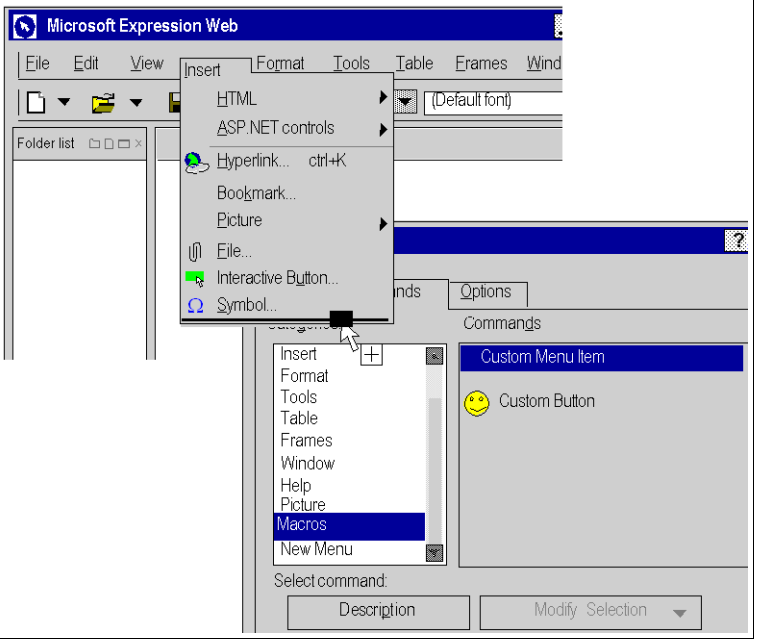
The extension can be installed by either re-installing Web Designer for FactoryCast or manually copying the macro file to MS Expression Web's macro folder. For manual installation, after installing MS Expression Web, the following will install the extension:Copy the file "Microsoft Expression Web.wdmacro"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

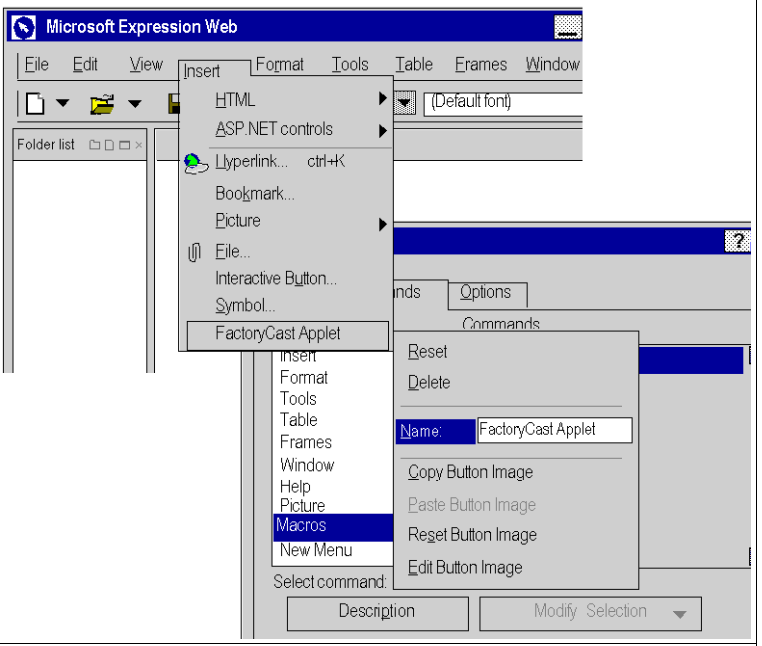
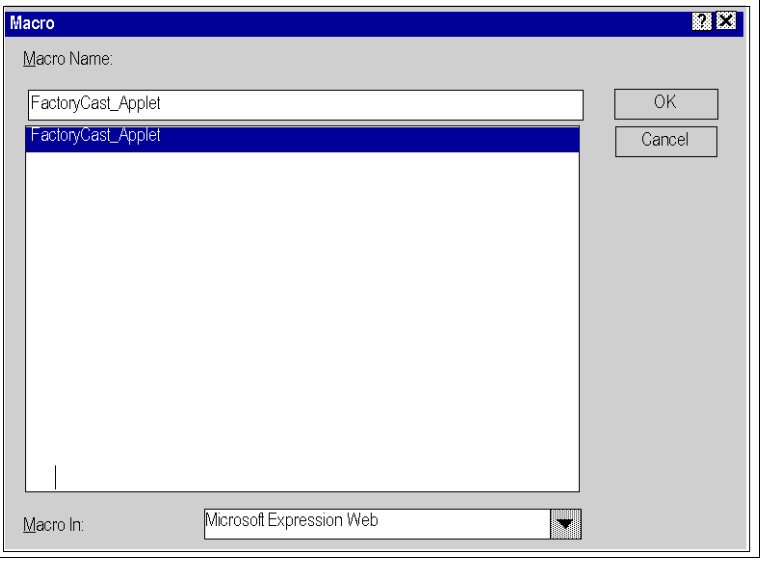
NOTE: MS Expression Web uses this address by default. For a custom installation, you can determine this address by typing SET from a Command Prompt.

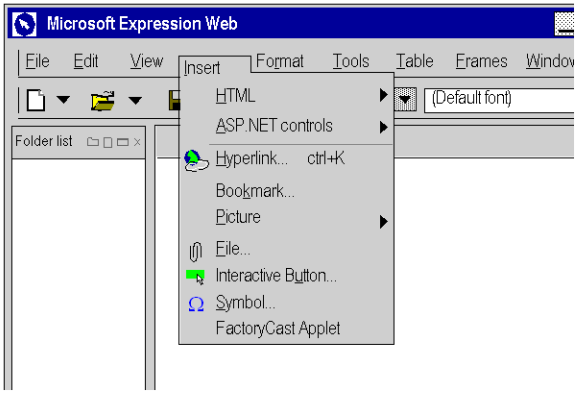
Adding FactoryCast Extension

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.

Step	Action
3	Click the **Commands** tab, and then select **Macros** from the **Categories** list. 
4	Drag and drop the entry **Custom Menu Item** from the **Commands** list to the **Insert** menu, beneath the **Symbol** command (the **Insert** menu will automatically drop down when you drag over it). 

Step	Action
5	Click **Modify Selection**, change the name to **FactoryCast Applet**, and then press ENTER key. 
6	Click **Modify Selection** again, and then choose **Assign Macro**. 

Step	Action
7	Choose **FactoryCast_Applet** from the list, and then click **OK**.
8	Click **Close**. The **FactoryCast Applet** command is added to the **Insert** menu. 

Removing FactoryCast Extension

To remove the FactoryCast Extension from the MS Expression Web menu, do the following.

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. First, you can double-click on the object and make changes via dialog boxes. Or you can switch to the HTML editor in MS Expression Web 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.

Installing FactoryCast Extension for Microsoft FrontPage

Overview

The following procedures describe how to install/remove the FactoryCast Extension for Microsoft FrontPage 2000.

Installing FactoryCast

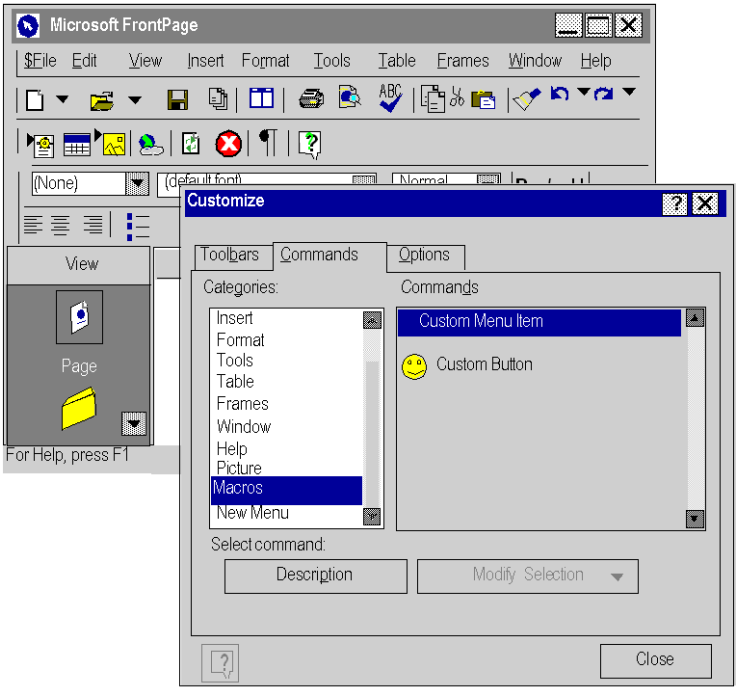
During the installation of Web Designer for FactoryCast, if FrontPage 2000 has been installed on the same PC, then the FactoryCast Extension for FrontPage 2000 is automatically installed as a FrontPage Macro File. If this is the case, then proceed to the next section, which provides instructions for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

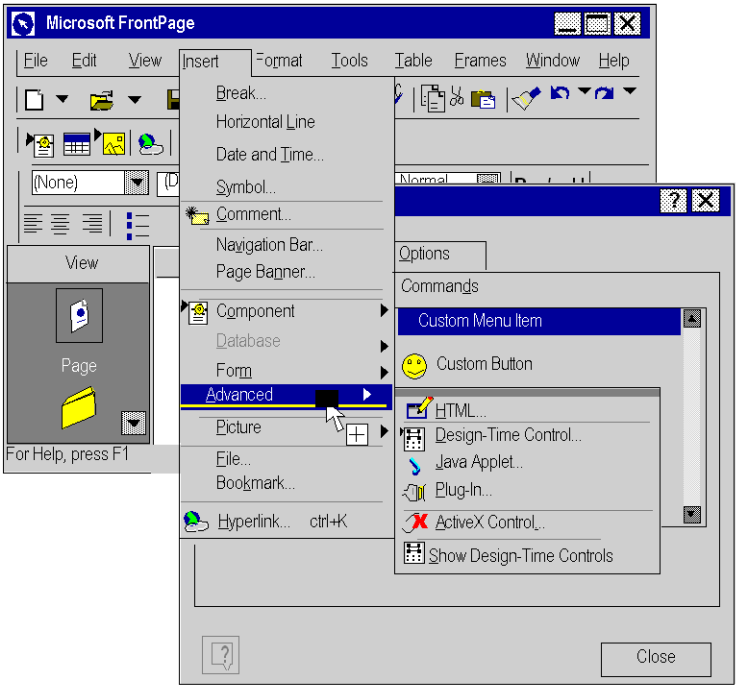
However, if FrontPage 2000 is installed after Web Designer for FactoryCast has been installed, then the Extension can be installed by either re-installing Web Designer for FactoryCast or manually copying the macro file to FrontPage's macro folder. For manual installation, after installing FrontPage 2000,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

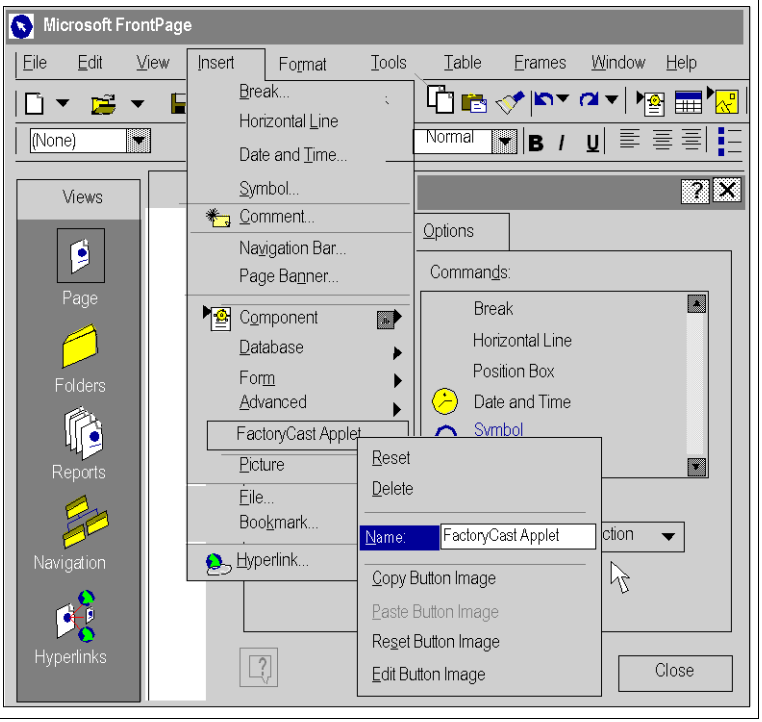
- **For Windows 95/98:** Copy the file *Microsoft FrontPage.fpm* to the folder *C:\WINDOWS\Application Data\Microsoft\FrontPage\Macros*. (Create the final *Macros* folder, if it does not already exist.)
- **For Windows NT/2000/XP:** Copy the file *Microsoft FrontPage.fpm* from CD-ROM to the folder *%USERPROFILE%\Application 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.

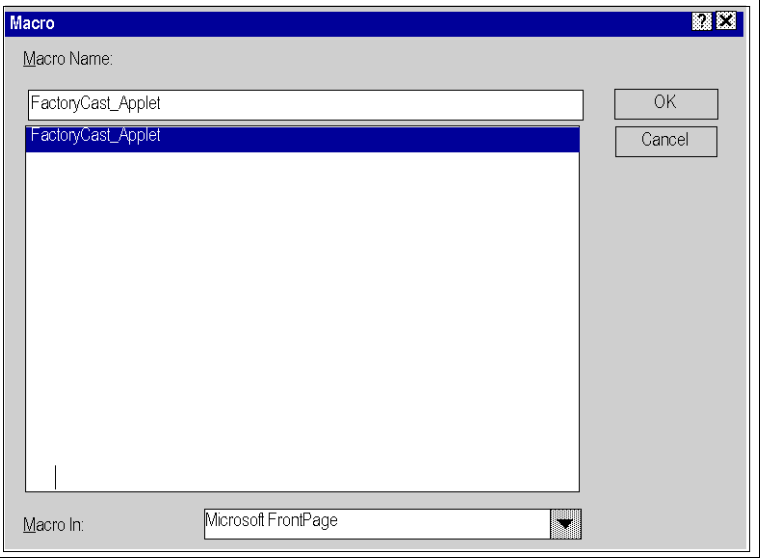
Adding FactoryCast Extension

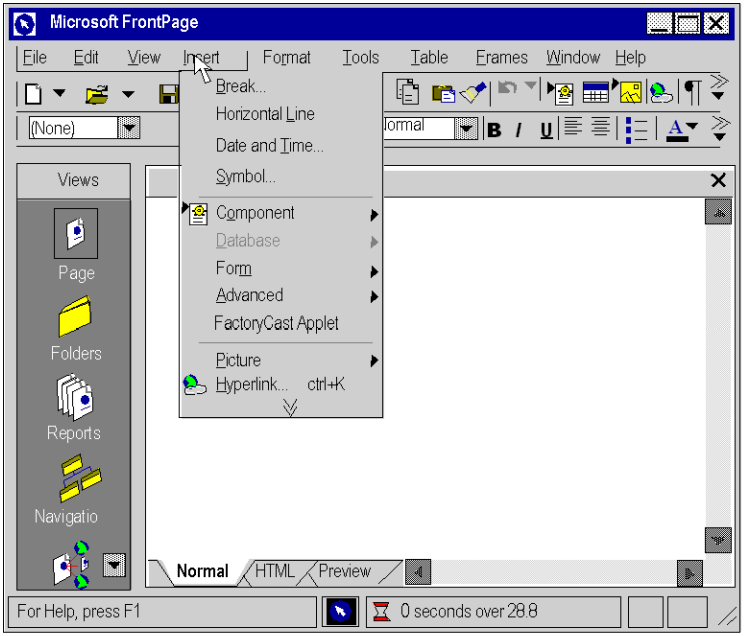
To add the FactoryCast Extension to the FrontPage **Insert** menu, do the following.

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. 

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.) 

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

Step	Action
6	Click **Modify Selection** again, and then choose **Assign Macro**. 
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. 

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

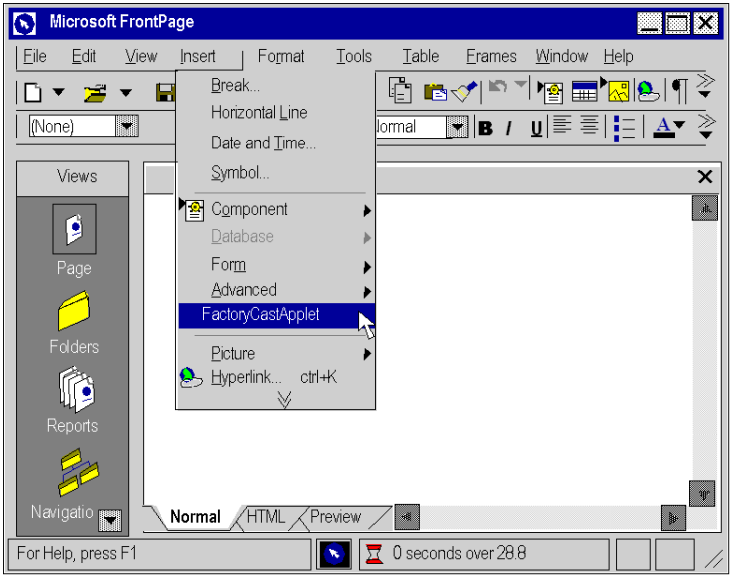
There are two ways to edit an applet that has been inserted into your Web page. You can double-click on the object and make changes via dialog boxes, or you can switch to the HTML editor in FrontPage and do your editing in this environment. It is suggested you edit via the first approach unless you are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

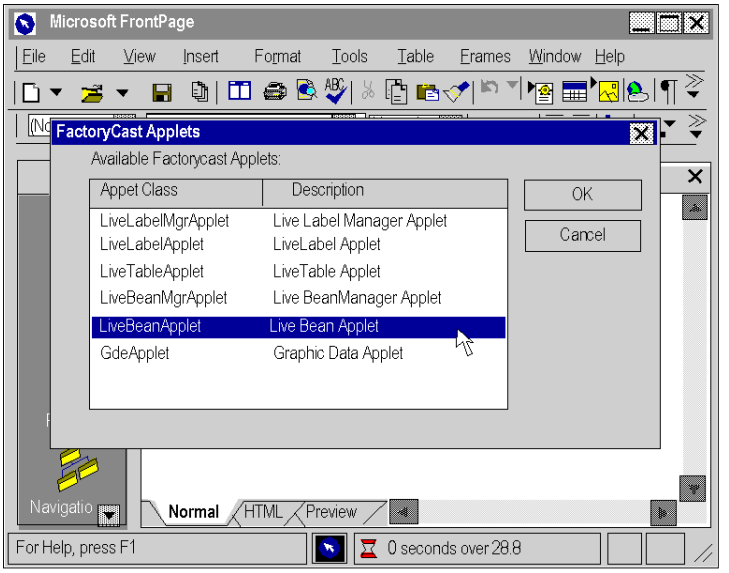
Inserting LiveBeanApplet Using FrontPage or Expression Web

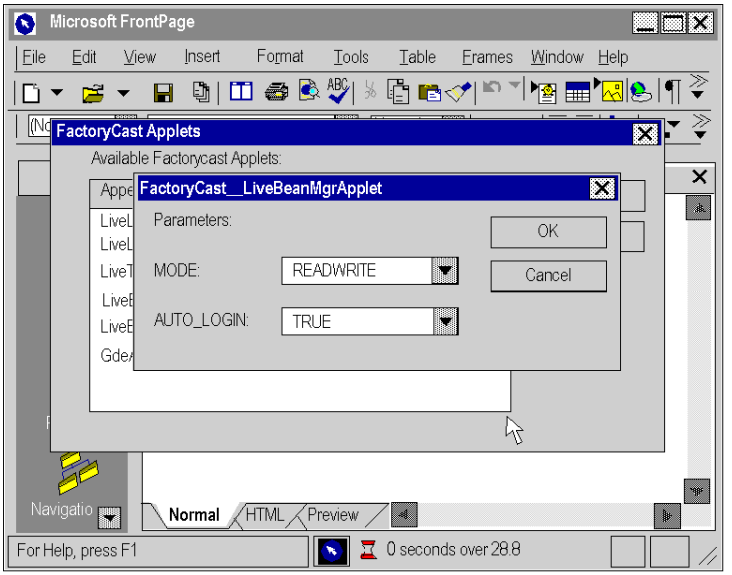
Overview

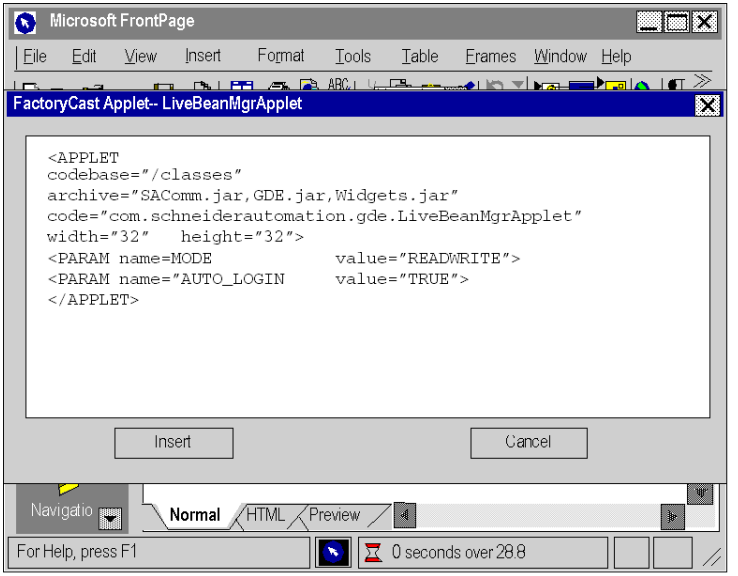
NOTE: The following procedure shows you how inserting **LiveBeanApplet** using Frontpage. Use the same procedure if you use Expression Web.

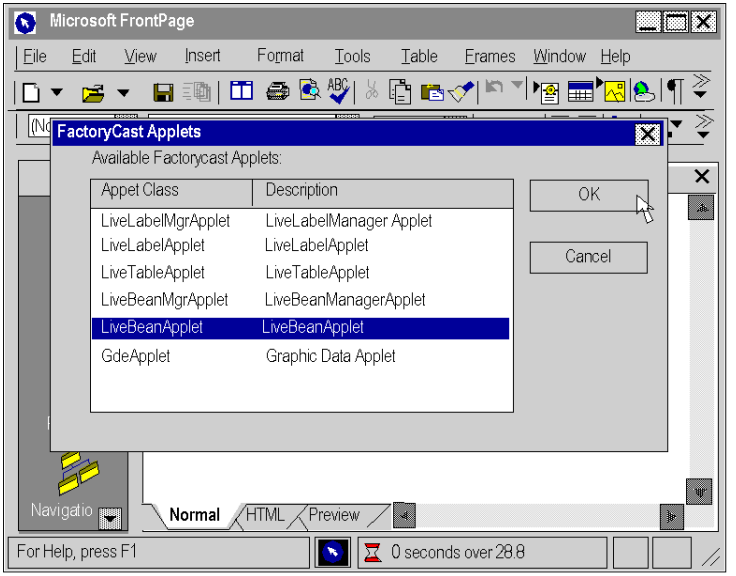
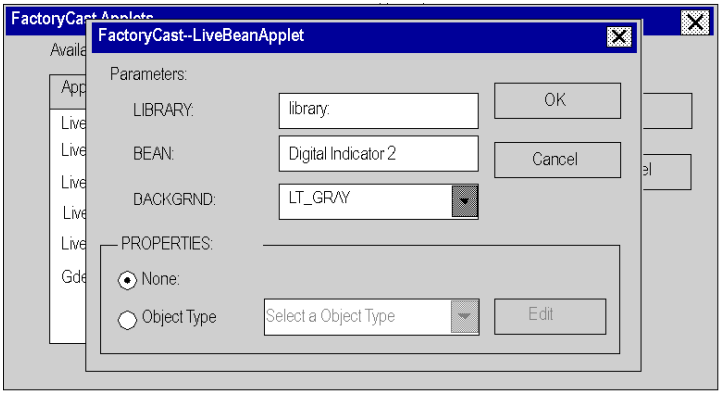
Inserting a LiveBeanApplet

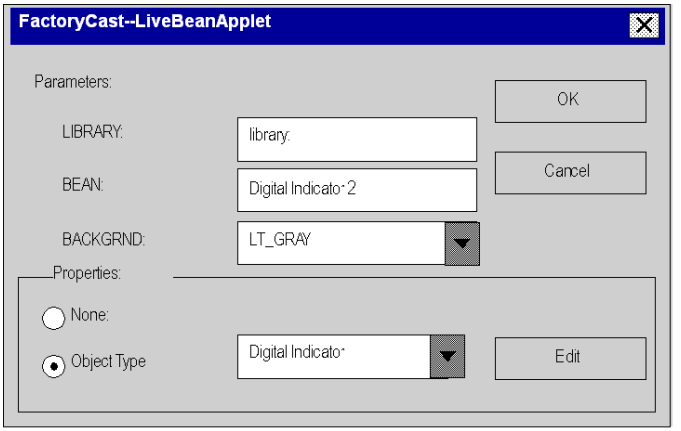
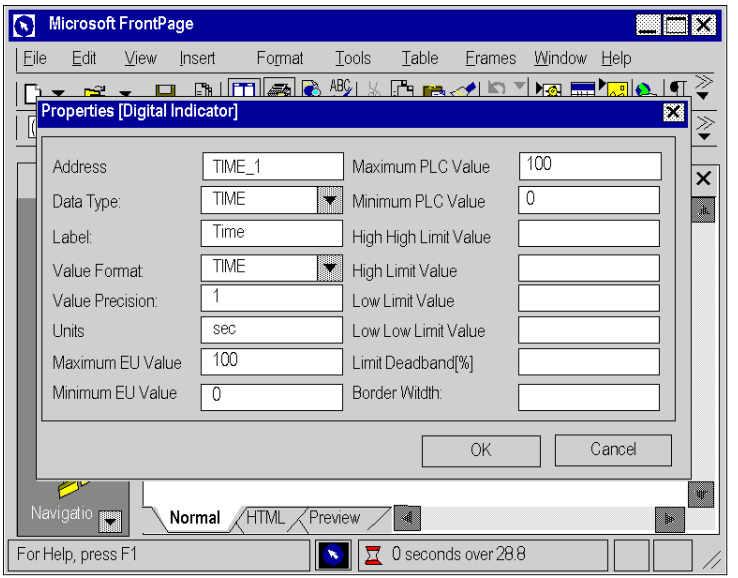
Step	Action
1	In FrontPage, Select **Insert | FactoryCast Applet** to view the FactoryCast applet selection window. 

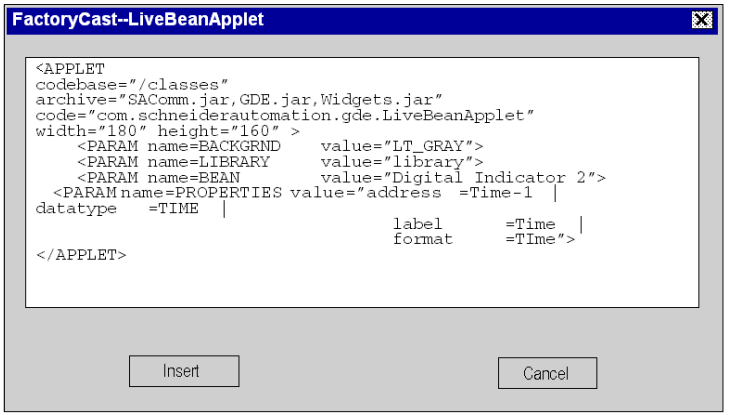
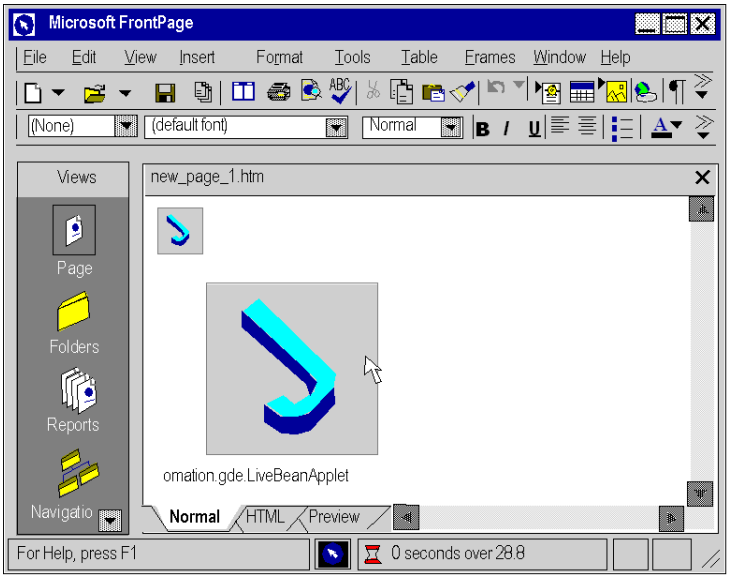
Step	Action
2	**Note:** The **LiveBeanMgrApplet** is a special applet that is inserted one time only into your Web page. Select the applet **LiveBeanMgrApplet** and then click the OK button. 

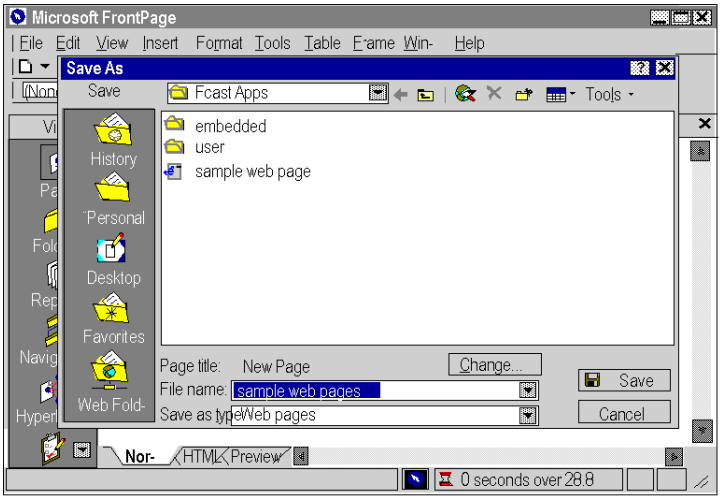
Step	Action
3	Configure the **Mode** and **Auto Login** parameters and then click the **OK** button. 

Step	Action
4	A window will appear showing the Java code that will be inserted into your HTML document.  Select 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** *(see page 272)* 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.

Step	Action
5	Select the **LiveBeanApplet** from the FactoryCast applet selection window and then click the **OK** button. 
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. 

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.  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. 

Step	Action
9	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. 
10	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. 

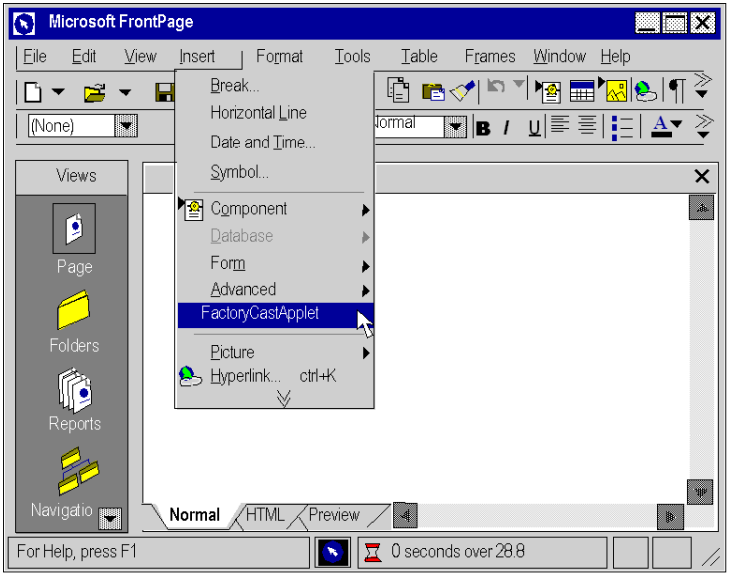
Step	Action
11	Save the customized Web page. 
12	Transfer the customized Web page to the NOE Web server.
13	Test the application.

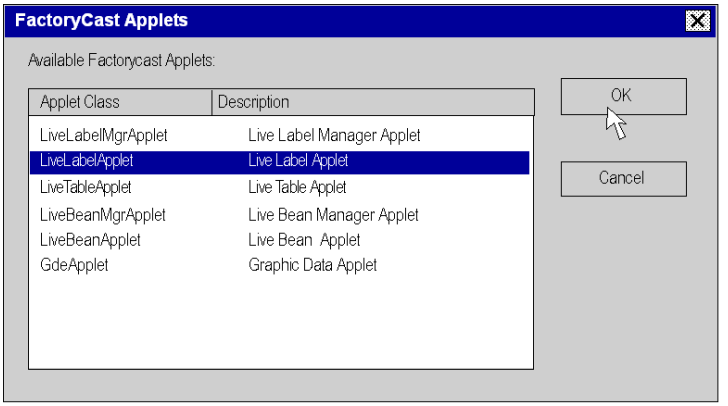
Inserting LiveLabelApplet Using FrontPage and Expression Web

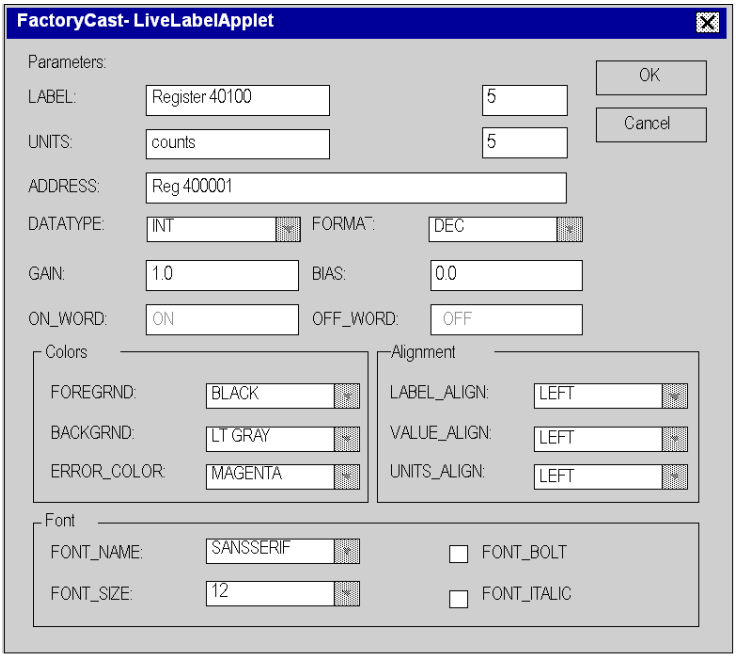
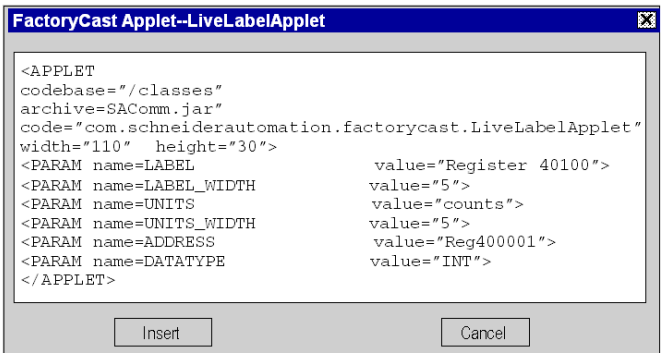
Overview

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. 

Step	Action
2	Select the applet **LiveLabelMgrApplet** and then click the **OK** button.  **Note:** The **LiveLabelMgrApplet** 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.
3	Select the applet **LiveLabelApplet** and then click the **OK** button. The Parameters edit window appears.

Step	Action
4	Fill in the parameters for the label and then click the **OK** button.  **Note:** For a description on each parameter, see *Using FactoryCast's Java Applets Using HTML Code, page 264*.
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. 

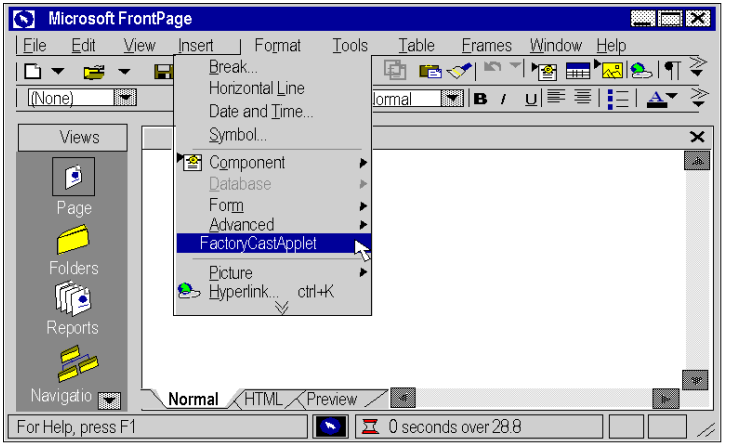
Step	Action
6	Continue to add additional instances of `LiveLabelApplet` to your Web page. Once you have inserted your last applet, click the **Cancel** button on the FactoryCast applet selection window to return to editing your Web page.

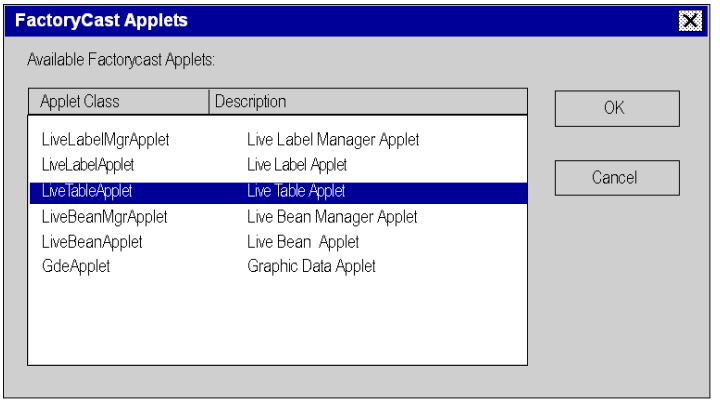
Inserting LiveTableApplet Using FrontPage or Expression Web

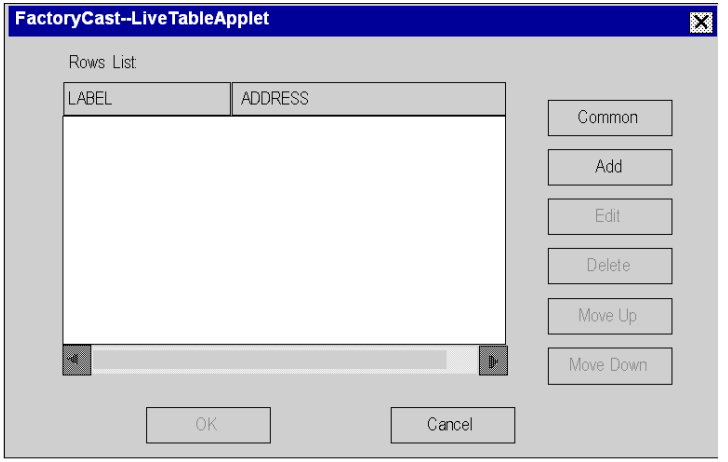
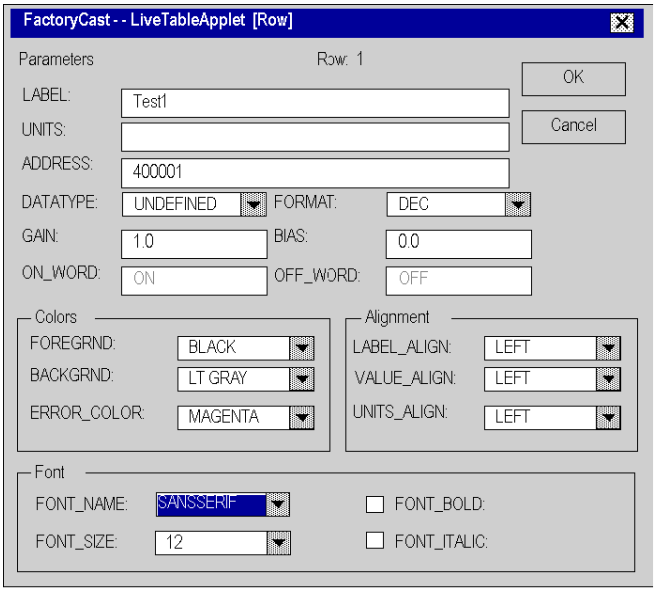
Overview

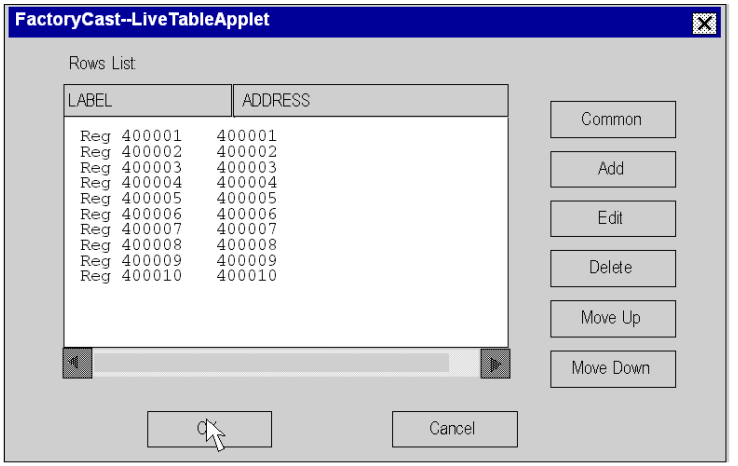
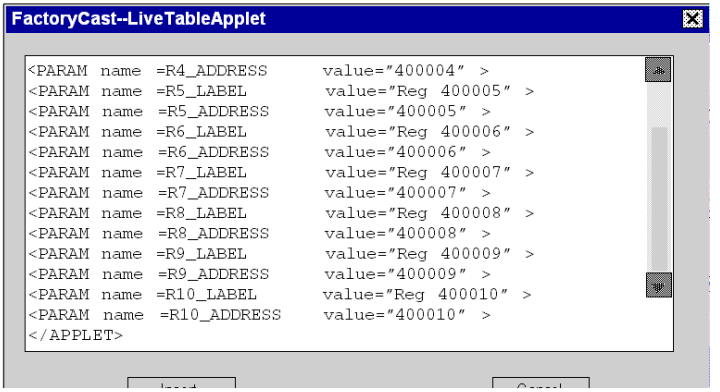
NOTE: The following procedure shows you how inserting **LiveTableApplet** using Frontpage. Use the same procedure if you use Expression Web.

Inserting LiveTableApplet

Step	Action
1	Select **Insert | FactoryCast** to view the FactoryCast applet selection window. 
2	Select the applet **LiveLabelMgrApplet** and then click the **OK** button. **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. 

Step	Action
4	Click the **Common** button to configure the parameters that are common to all rows of the table.  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. 

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. 
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. 

Refer to Graphic Editor Applet Parameters *(see page 219)* for inserting a custom GDEApplet applet.

7.4 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:

Topic	Page
Downloading the Graphic Objects Lite Library	307
Description of Graphic Objects Lite	308

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**.

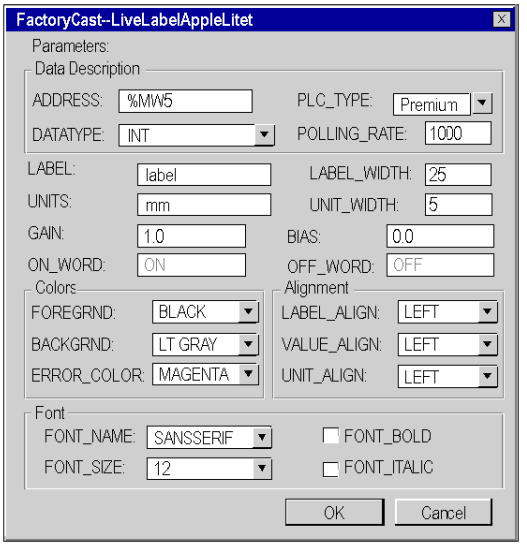
Description of Graphic Objects Lite

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.



The properties of this widget are as follows:

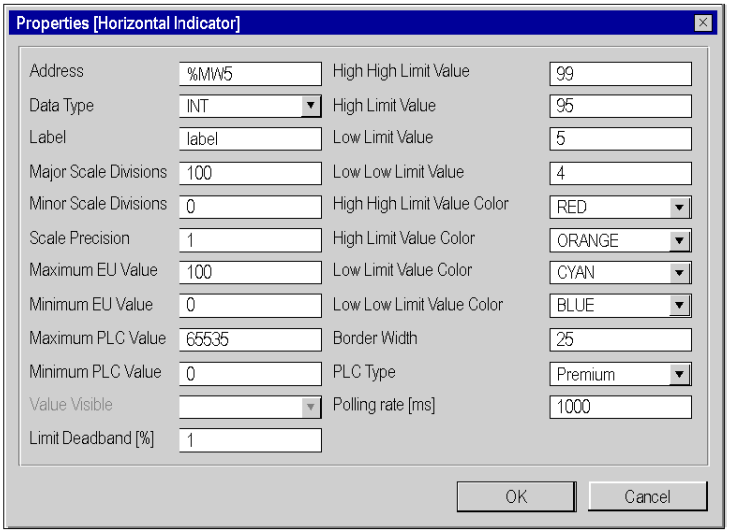
Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 324)*
Data type	PLC address data type	Note 2 *(see page 324)*
PLC Type	Type of PLC	Premium or Quantum
Label	Label displayed as part of the graphic object	Note 5 *(see page 324)*
Label Width	Width of label	
Unit Width	Width 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.



The properties of the indicator are as follows:

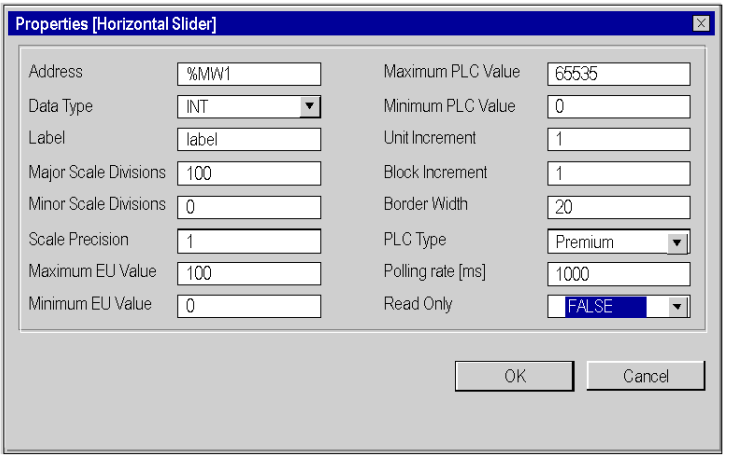
Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 324)*
Data type	PLC address data type	Note 2 *(see page 324)*
Label	Label displayed as part of the graphic object	Note 5 *(see page 324)*
Major scale gradation	Number of major gradations (marked) in the scale	0 to 100
Minor 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 324)*
Minimum PLC Value	Gross minimum value (without scale) of the direct address in the PLC	Note 3 *(see page 324)*
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.



The properties of the slider are as follows:

Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 324)*
Data type	Data type of the PLC address	Note 2 *(see page 324)*
Label	Label displayed as part of the graphic object	Note 5 *(see page 324)*
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 324)*
Minimum PLC Value	Gross minimum value (without scale) of the direct address in the PLC	Note 3 *(see page 324)*
Unit Increment	Amount by which the scale value is modified when you click on the slider arrows	

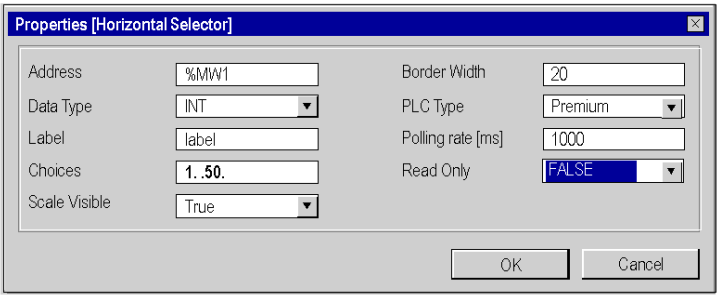
Property	Description	Limits
Block Increment	Amount by which the scale value is modified when you click on the slide area of the bar	
Border Width	Width (in pixels) of the border of the graphic object	
PLC Type	Type of PLC	Premium or Quantum
Polling rate [ms]	Scanning value	
Read Only	TRUE = read value, 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.



The properties of the slider are as follows:

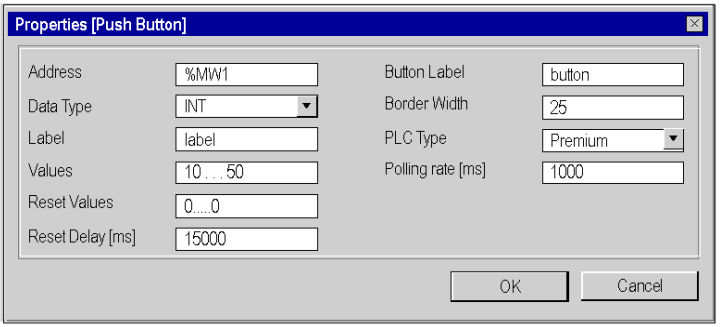
Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 324)*
Data type	PLC address data type	Note 2 *(see page 324)*
Label	Label displayed as part of the graphic object	Note 5 *(see page 324)*
Choices	Setpoints to apply to the address value	
Border Width	Width (in pixels) of 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.



The properties of the push button are as follows:

Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 324)*
Data type	PLC address data type	Note 2 *(see page 324)*
Label	Label displayed as part of the graphic object	Note 5 *(see page 324)*
Values	Setpoints to apply to the address value when the button is pressed	Note 4 *(see page 324)*
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	

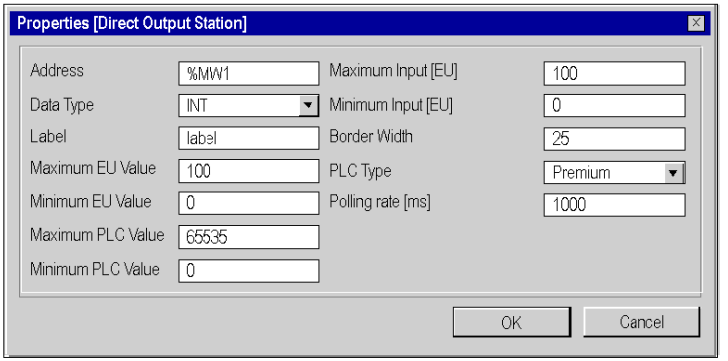
The HTML code with the parameters of the widget above is as follows:

```
<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).



The direct output properties are as follows:

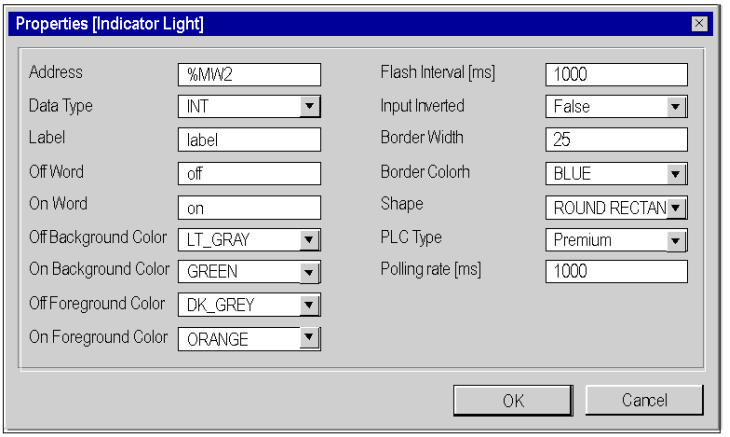
Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 324)*
Data type	PLC address data type	Note 2 *(see page 324)*
Label	Label displayed as part of the graphic object	Note 5 *(see page 324)*
Maximum EU Value	Maximum value, in physical units, of the direct address for scaling	
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 324)*
Minimum PLC Value	Gross minimum value (without scale) of the direct address in the PLC	Note 3 *(see page 324)*
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.



The properties of the indicator light are as follows:

Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 324)*
Data type	PLC address data type	Note 2 *(see page 324)*
Label	Label displayed as part of the graphic object	Note 5 *(see page 324)*
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	
Shape	Shape (circular, rectangular, etc.) of the indicator light	
PLC Type	Type of PLC	Premium or Quantum
Polling rate [ms]	Scanning value	

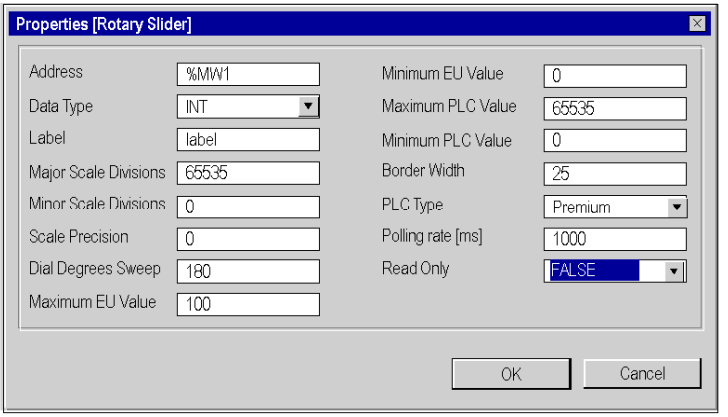
The HTML code with the parameters of the widget above is as follows:

```
<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.

The properties of the slider are as follows:



Property	Description	Limits
Address	Address of the PLC variable	Note 1 *(see page 324)*
Data type	PLC address data type	Note 2 *(see page 324)*
Label	Label displayed as part of the graphic object	Note 5 *(see page 324)*
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 324)*
Minimum PLC Value	Gross minimum value (without scale) of the direct address in the PLC	Note 3 *(see page 324)*
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.	

7.5 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 Unity Pro 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

Alarm Viewer

8

Overview

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

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Display	328
Operation and Management of Alarms	331
Limitations	332

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

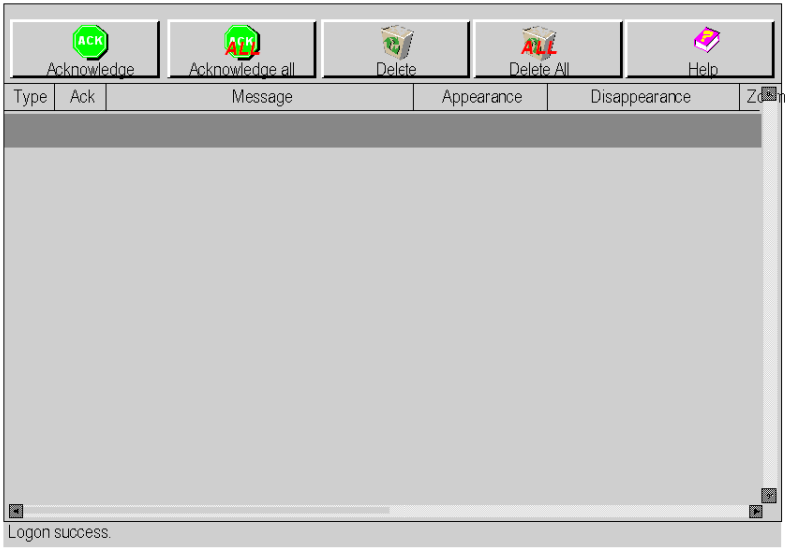
Button Bar

The button bar contains five buttons.

	This button is used to acknowledge a selected alarm in the list. A request is sent to the PLC (Diagnostic Buffer).
	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.
	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.



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 alarm


Grafcet alarm


System alarm



A-SI alarm

Ack: An icon that represents the acknowledgement status.



This alarm must be acknowledged by the user.



Alarm has been acknowledged.



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 alarm cause 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	Firmware Platform	Software Platform
TSX ETY 110 WS	Premium PLC TSX57/PCX57/PMX57 V3.3	PL7 PRO PL-7 Junior V3.3 Unity Pro
TSX ETY 5103	Premium PLC TSX57/PCX57/PMX57 V3.3	PL7 PRO PL-7 Junior V3.3 Unity Pro
140 NOE 771 11	Quantum Unity Platform	Unity Pro

SOAP Web Services

9

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.

Two 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 Unity Pro.

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:

Topic	Page
Principles for Designing a SOAP Client Interface	335
ModBusXMLDA SOAP Services	336
SymbolicXMLDA SOAP Services	347
ExtendedSymbolicXMLDA SOAP Services	353
How to optimize massive symbolic requests	360
Visual Basic Examples	363
Online SOAP Documentation	364

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, according to the modules, that can be used for API parameters:

APIs	ETY 5103	NOE 77111
ReadMultipleRegisters	125 (Register)	124 (Register)
WriteMultipleRegisters	120 (Register)	96 (Register)
ReadCoils	512 (Coils)	255 (Coils)
WriteMultipleCoils	511 (Coils)	254 (Coils)
ReadInt32	62 (Int32)	61 (Int32)
WriteInt32	59 (Int32)	48 (Int32)

ReadDevice-Identification

This 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/ReadDeviceIdentification"
<?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</TRImplementa-
```

```
tionClass>
           <TRCommunicationServices>string</TRCommunication-
Services>
         </DeviceIdentification>
       </ReadDeviceIdentificationResult>
     </ReadDeviceIdentificationResponse>

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

ReadMultiple-Registers

This is the 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 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/WriteMultipleRegisters"
<?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 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 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/WriteMultipleCoils"
<?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 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 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/WriteMultipleRegisters"
<?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 UnityPro. 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 Unity protocol access for actual communication, only symbols and direct addresses mapped on data accessible through native Unity 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 UnityPro 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 Unity 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 Unity 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/ExtendedSymbolicsXmlDa/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/ExtendedSymbolicsXmlDa/">
        <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 Unity 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/ExtendedSymbolicsXmlDa/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/ExtendedSymbolicXmlDa/">
       <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/ExtendedSymbolicXmlDa/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.

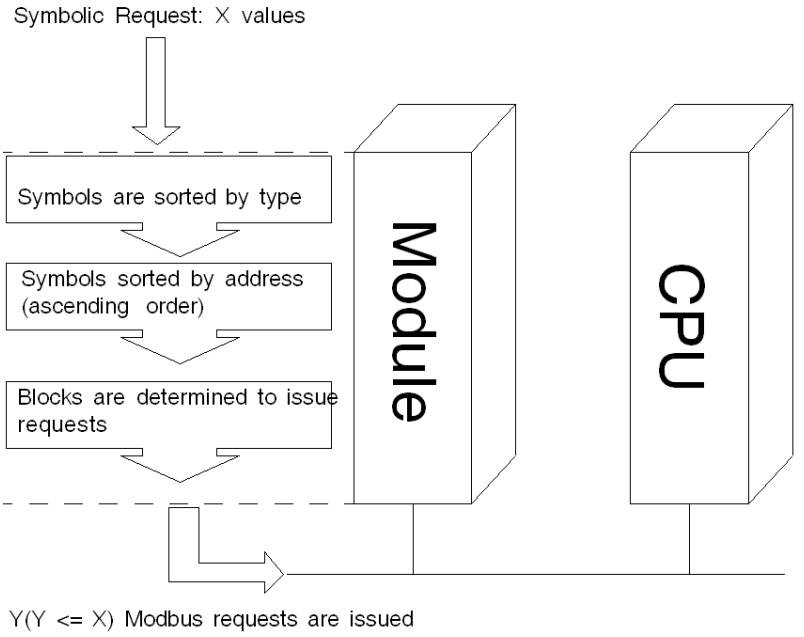
You can experience a lag when using a large amount of symbols. There are several rules you can apply to improve your application's speed, by reducing the number of internal requests.

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:



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).

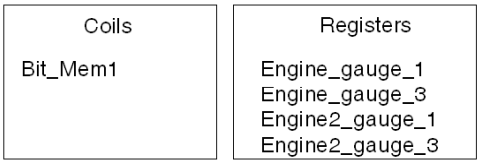
There are also Boolean values (Bit_mem).

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: 
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

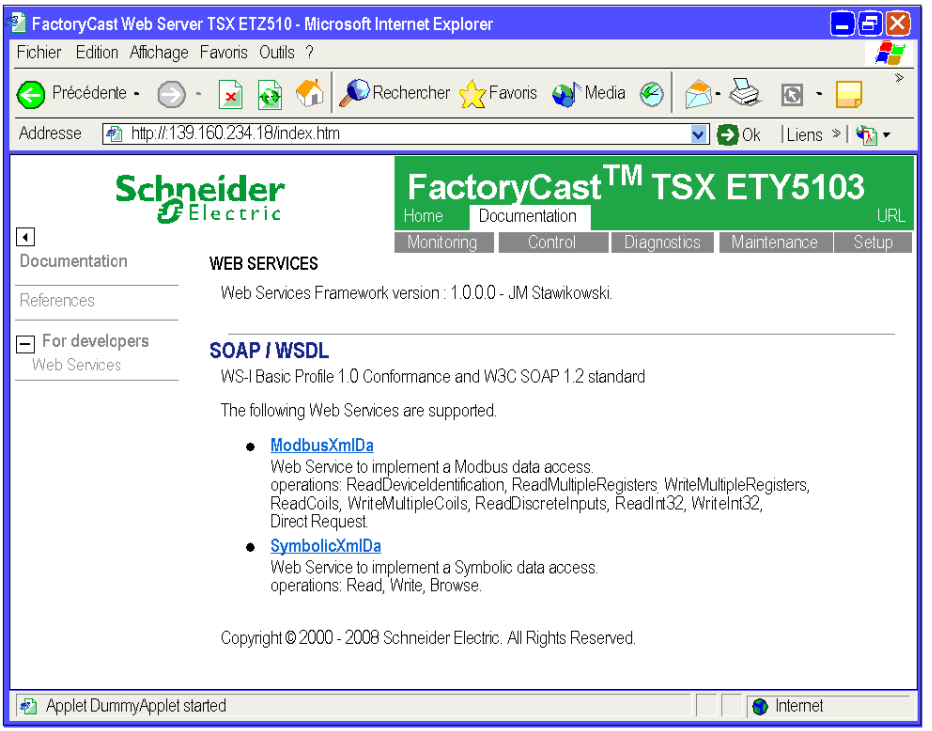
End Class
```

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.



Appendices



Overview

There are two sets of appendices for the FactoryCast User's Guide. The first appendix is browser requirements, settings, and security considerations, and the second appendix is performance benchmarks.

What Is in This Appendix?

The appendix contains the following chapters:

Chapter	Chapter Name	Page
A	Browser Requirements, Settings, and Security Considerations	367
B	Performance Benchmarks	375

Browser Requirements, Settings, and Security Considerations

A

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:

Topic	Page
Browser Version	368
Browser Settings	370
Browser Security Considerations	373

Browser Version

Overview

To display FactoryCast web pages, use a browser that supports 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?

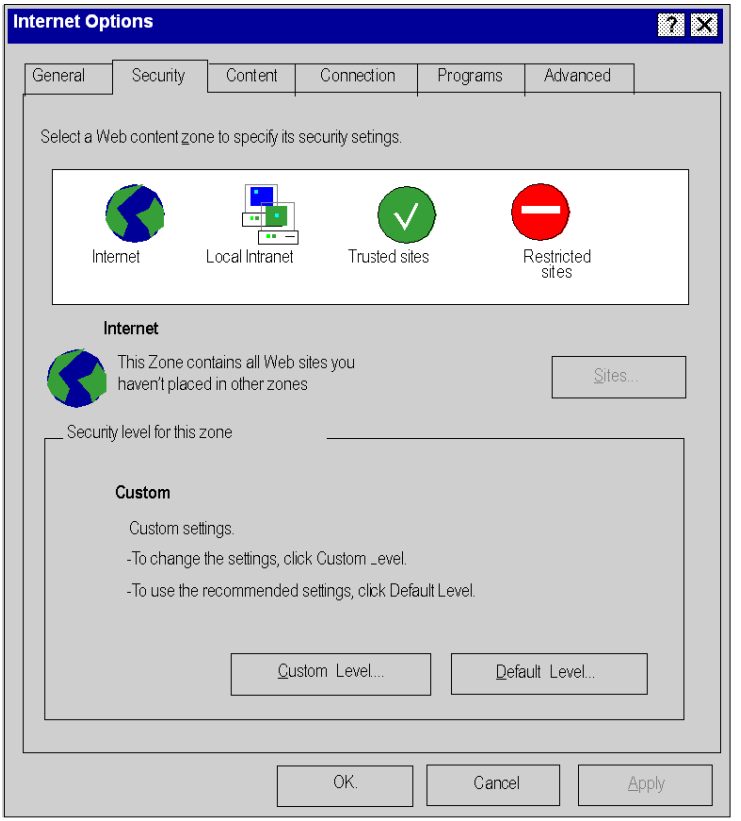
Supported browsers are listed below:

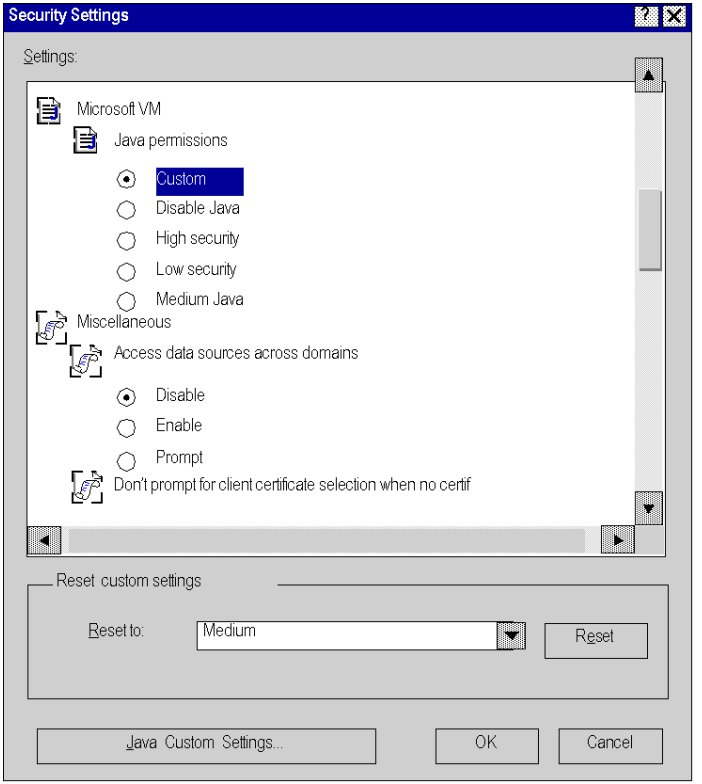
Download the browser:	At Web Site:
Mozilla Firefox	http://www.mozilla.com
Internet Explorer	http://www.microsoft.com

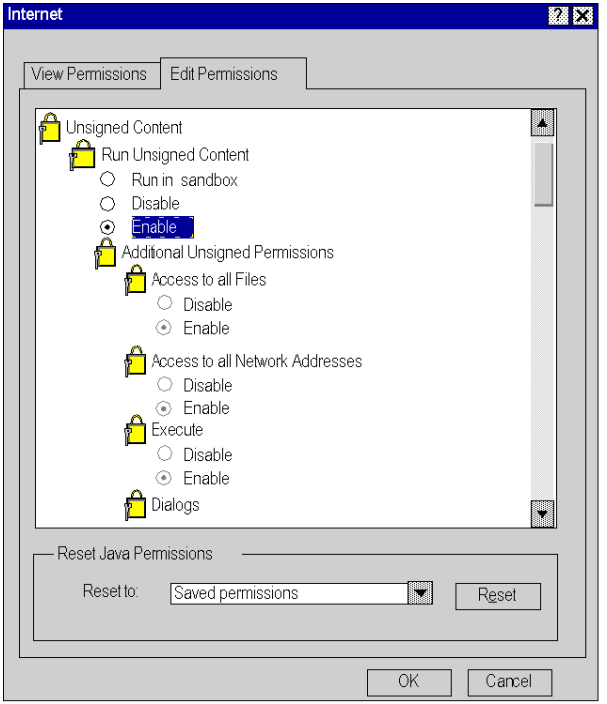
Browser Settings

Modifying Security Settings

If you are using Microsoft Internet Explorer and you cannot view Java applets, modify your browser security settings, as follows:

Step	Action
1	Open Internet Explorer.
2	Select **Tools | Internet Options** from the menu bar. **Result:** The **Internet Options** window appears.
3	Select the **Security** tab. 
4	Select the **Custom Level** button in the **Security Level** settings section.

Step	Action
5	Select the **Settings** button. **Result:** The Security Settings dialog appears. 
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. 
8	Select the **Edit Permissions** tab.
9	Under the **Run Unsigned | Content**, select **Enable**.
10	Click **OK**.

Browser Security Considerations

Overview

Both Mozilla Firefox and Internet Explorer remember a user name and password after it has been 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."

Performance Benchmarks

B

Overview

This appendix contains performance benchmarks for several configurations.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
B.1	Premium Performance Benchmarks	376
B.2	Quantum Performance Benchmarks	382

B.1 Premium Performance Benchmarks

Overview

This section describes the performance of the Premium embedded Web server.

What Is in This Section?

This section contains the following topics:

Topic	Page
Premium Performance Benchmarks	377
Monitoring Applet Performance on ETY5103	379
SOAP Performance Benchmarks	381

Premium Performance Benchmarks

Overview

This section contains information on the impact that access to the embedded Web server has on the TSX ETY 110 messaging module's performances.

All the response times are in ms. The client and the server have the same cycle time.

Average time of 1 EF READ-VAR of 50 words

ETHWAY

	Cyclic	**5**	**10**	**20**	**50**
TSX ETY 110	73	73	76	82	103
TSX ETY 110 WS installed	73	72	76	83	138
TSX ETY 110 WS running with data editor	73	73	75	83	127
TSX ETY 110 WS running with sysdiag on ETY client	86	88	91	98	149
TSX ETY 110 WS running with sysdiag on ETY client and server	93	98	99	99	148

TCP/IP

	Cyclic	**5**	**10**	**20**	**50**
TSX ETY 110	74	75	77	84	102
TSX ETY 110 WS installed	74	73	77	84	106
TSX ETY 110 WS running with data editor	74	73	77	85	108
TSX ETY 110 WS running with sysdiag on ETY client	80	85	90	98	109
TSX ETY 110 WS running with sysdiag on ETY client and server	92	94	100	108	128

Average time of 8 EF READ-VAR of 50 words

ETHWAY

	Cyclic	10	50
TSX ETY 110	221	229	247
TSX ETY 110 WS installed	223	229	247
TSX ETY 110 WS running with data editor	224	231	249
TSX ETY 110 WS running with sysdiag on ETY client	236	240	254
TSX ETY 110 WS running with sysdiag on ETY client and server	246	252	275

TCP/IP

	Cyclic	10	50
TSX ETY 110	244	254	287
TSX ETY 110 WS installed	244	261	291
TSX ETY 110 WS running with data editor	245	259	293
TSX ETY 110 WS running with sysdiag on ETY client	262	270	309
TSX ETY 110 WS running with sysdiag on ETY client and server	304	307	337

Throughput Time of Bridge

ETHWAY

	Cyclic	10	20	50
TSX ETY 110	55	56	55	26
TSX ETY 110 WS running	51	51	47	18
TSX ETY 110 WS running with Comm ETY Bridge	61	58	58	56
TSX ETY 110 WS running with Comm ETY client and server	62	63	62	23

TCP/IP

	Cyclic	10	20	50
TSX ETY 110	57	57	60	31
TSX ETY 110 WS running	58	56	55	33
TSX ETY 110 WS running with Comm ETY Bridge	65	63	64	67
TSX ETY 110 WS running with Comm ETY client and server bridge	66	69	67	54

Monitoring Applet Performance on ETY5103

Ethernet Configuration

- PC Intel Pentium IV 1.7 GHz
- JVM 1.4.2_09 (Java virtual machine)
- 100 Mb/s Ethernet

Modem Configuration

- PC Intel Pentium IV 1.7 GHz
- JVM 1.4.2_09
- 56k modem (38400 bauds)

Data Editor

This table displays Data Editor performances:

RDE	100 Mb/s	Modem
1st page loading	2 s	2 m 55 s
Following loading	2 s	14 s
Time refresh value, full table	300 ms	300 ms

Data Editor Lite

This table displays Data Editor Lite performances:

RDE	Modem
1st page loading	1 m 20 s
Following loading	4 s
Time refresh value, full table	300 ms

Graphic Editor

This table displays Graphic Editor performances:

GDE	100 Mb/s	Modem
First page loading without graphic page	3 s	5 m 46 s
Following loading	2 s	20 s
Loading page with 20 widgets	2 s	11 s
Loading page with 50 widgets	2 s	18 s
Loading page with 100 widgets	2 s	37 s
Loading page with 200 widgets	3 s	1 m 05 s

Custom Pages

Concerning Custom Pages performances with widgets refer to the previous table.

This table displays Custom Pages performances with new widgets lite:

Custom pages	Modem
First page loading with 20 widgets lite	55 s
Following loading page with 20 widgets lite	3 s
First page loading with 50 widgets lite	1 m 04 s
Following loading page with 50 widgets lite	4 s
First page loading with 100 widgets lite	1 m 30 s
Following loading page with 100 widgets lite	5 s
First page loading with 200 widgets lite	1 m 45 s
Following loading page with 200 widgets lite	8 s

SOAP Performance Benchmarks

SOAP communication

Microsoft Application Center test tools (supplied with Visual Studio.net 2003) were used for the performance measurements.

The following table shows measured performance:

Nature of measurement	Time spent for 100 iterations	Average request rate
The request reads 50 registers (from %MW0 to %MW49)	20 s	5/s
The request writes 50 registers (one Modbus request)	20 s	5/s
The request reads 50 symbols of noncontiguous registers	4 m 09 s	0.30/s
The request reads 50 symbols of contiguous registers	27 s	3.7/s

B.2 Quantum Performance Benchmarks

Overview

This section describes the performance of the Quantum embedded Web server.

What Is in This Section?

This section contains the following topics:

Topic	Page
Quantum Performance Benchmarks	383
Performance on NOE77111	384
SOAP Performance Benchmarks	386

Quantum Performance Benchmarks

Overview

These Quantum Web Embedded Server performance tests were run on Win95 on a Dell OptiPlex Gxi at 200 MHZ. The PLC was a Quantum 424.

Performance Data

The following table contains the performance data for the Quantum Web Embedded Server.

Web Scanner w/MSTR Block	50 scans	100 scans	200 scans	500 scans
Web scanner loaded, 1 MSTR	48.2ms	33.4ms 134.3ms	34.3ms	34.3ms
Web scanner loaded, 2 MSTR	43.7ms 66.4ms	45.6ms	43.8ms	46.7ms
Web scanner loaded, 4 MSTR	72.35ms 86.19ms	73ms	74ms	74ms 90ms
Web scanner loaded, 4 MSTR, 1 browser	107.3ms	72.5ms	73ms	72ms
Web scanner loaded, 4 MSTR, 3 browsers	78.9ms 94.2ms	80.5ms	81.1ms	120ms
Web scanner loaded, 4 MSTR, 5 browsers	105ms	80.5ms	80ms	90ms
Web scanner loaded, 4 MSTR, 5 instances of Internet Explorer (no cache)	104.1ms 92.5ms	85.2ms	88.9ms	90ms
Web scanner loaded, 4 MSTR, 5 instances of Internet Explorer (cache on)	108ms 83.7ms	85.2ms	90ms	120ms 92.5ms

Performance on NOE77111

Ethernet Configuration

- PC Intel Pentium IV 1.7 GHz
- JVM 1.4.2_09 (Java virtual machine)
- 100 Mb/s Ethernet

Modem Configuration

- PC Intel Pentium IV 1.7 GHz
- JVM 1.4.2_09
- 56k modem (38400 bauds)

Data Editor

This table displays Data Editor performances:

RDE	100 Mb/s	Modem
1st page loading	2 s	2 m 55 s
Following loading	2 s	14 s
Time refresh value, full table	300 ms	300 ms

Data Editor Lite

This table displays Data Editor Lite performances:

RDE	Modem
1st page loading	1 m 20 s
Following loading	4 s
Time refresh value, full table	300 ms

Graphic Editor

This table displays Graphic Editor performances:

GDE	100 Mb/s	Modem
First page loading without graphic page	3 s	5 m 46 s
Following loading	2 s	20 s
Loading page with 20 widgets	2 s	11 s
Loading page with 50 widgets	2 s	18 s
Loading page with 100 widgets	2 s	37 s
Loading page with 200 widgets	3 s	1 m 05 s

Custom Pages

Concerning Custom Pages performances with widgets refer to the previous table.

This table displays Custom Pages performances with new widgets lite:

Custom pages	Modem
First page loading with 20 widgets lite	55 s
Following loading page with 20 widgets lite	3 s
First page loading with 50 widgets lite	1 m 04 s
Following loading page with 50 widgets lite	4 s
First page loading with 100 widgets lite	1 m 30 s
Following loading page with 100 widgets lite	5 s
First page loading with 200 widgets lite	1 m 45 s
Following loading page with 200 widgets lite	8 s

SOAP Performance Benchmarks

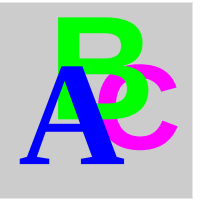
SOAP Communication

Microsoft Application Center test tools (supplied with Visual Studio.net 2003) were used for the performance measurements.

The following table shows measured performance:

Nature of measurement	Time spent for 100 iterations	Average request rate
The request reads 50 registers (from %MW0 to %MW49)	21 s	4.77/s
The request writes 50 registers (one Modbus request)	22 s	4.85/s
The request reads 50 symbols of noncontiguous registers	4 m 59 s	0.33/s
The request reads 50 symbols of contiguous registers	31 s	3.22/s

Index



0-9

140 NOE 771 11, *15*

A

Alarm Viewer display, *328*
alarms
  operation and management of, *331*
analog I/O diagnostics, *149*
Analog I/O module status web page
  Premium, *127*
  Quantum, *60*
applets
  Java, *269*
  using, *269*
AS-i diagnostics, *151*
AS-i module status web page
  Premium, *128*

B

benchmarks, *375*
browsers
  requirements, *367*
  security, *367*, *373*
  setting, *367*
  supported, *368*
  versions, *367*

C

changing passwords, *96*, *99*, *161*, *164*
configuration tool
  overview, *17*
controller status
  Premium, *134*
Controller Status web page
  Quantum, *50*
CPU Ethernet port status web page
  Premium, *116*
CPU serial port status web page
  Premium, *117*
CPU status web page
  Premium, *116*
custom Web pages
  restricted access, *263*
  unrestricted access, *263*
  using Java applets, *269*

D

Data Editor
  spreadsheets, *179*
  symbols, *188*
  templates, *185*
  variables, *188*
data management, *19*
data templates, *185*
default Web sites
  read-only, *27*
defaults
  restore, *22*

diagnostics
Analog I/O, *149*
AS-i, *151*
Digital I/O, *147*
distributed I/O, *62*
Fipio module, *141*
Java web pages, *62*, *133*
PCMCIA communication, *153*
remote I/O, *62*
standard module, *155*
Diagnostics web page
Quantum, *39*
direct addresses, *190*
write-enabled, *191*
Discrete I/O module status web page
Premium, *124*
Quantum, *58*
drops page
distributed I/O, *78*

E

Embedded Servers, *16*
benchmarks: Premium, *375*
benchmarks: Quantum, *375*
Ethernet communication module status web page
Premium, *123*
Ethernet Module Statistics web page
Quantum, *52*
Ethernet Remote I/O (adapter) web page
Quantum, *55*
Ethernet Remote I/O (head) web page
Quantum, *53*

F

Fipio diagnostics web page
Premium, *129*
Fipio module diagnostics, *141*
FTP, *16*

H

home pages
custom, *261*
HTTP, *16*
Address server, *159*
Graphic Objects, *308*

I

internet browsers
supported, *368*

J

Java applets, *269*

N

namespace, *19*

P

parameters
setting daylight savings time, *171*
setting NTP server IP address, *171*
setting synchronization period, *171*
setting time zone, *171*
passwords, *96*, *99*, *161*, *164*
PCMCIA card status web page
Premium, *119*
PCMCIA communication diagnostics, *153*
performance benchmarks, *375*
PLC program viewer, *251*
pocket data editor
presentation, *196*
Premium
Analog I/O module status web page, *127*
AS-i module status web page, *128*
CPU Ethernet port status web page, *116*
CPU serial port status web page, *117*
CPU status web page, *116*
Discrete I/O module status web page, *124*
Ethernet communication module status

web page, *123*
Fipio diagnostics web page, *129*
PCMCIA card status web page, *119*
Rack Viewer web page, *108*
Properties web page
Quantum, *48*

Q

Quantum
Analog I/O module status web page, *60*
Controller Status web page, *50*
Diagnostics web page, *39*
Discrete I/O module status web page, *58*
Ethernet Module Statistics web page, *52*
Ethernet Remote I/O (adapter) web page, *55*
Ethernet Remote I/O (head) web page, *53*
Properties web page, *48*
Rack Viewer web page, *40*
S908 Remote I/O (adapter) web page, *56*
S908 Remote I/O (head) web page, *54*

R

Rack Viewer web page
navigating, *46*
Premium, *108*
Quantum, *40*
restore defaults, *22*

S

S908 Remote I/O (adapter) web page
Quantum, *56*
S908 Remote I/O (head) web page
Quantum, *54*
security, *373*
access, *28*, *31*
firewall, *29*
passwords, *28*
Silverlight pages, *325*
SOAP, *333*
standard module diagnostics, *155*
symbols
definition, *14*
inserting, *188*
modifying, *191*

T

time zones, *172*
customizing, *172*
traps
setting, *171*
TSX ETY5103, *15*
tz compilers, *172*

V

variables
definition, *14*
inserting, *188*
modifying, *191*

W

Web Designer, *17*
Web pages
NTP Configuration, *170*