FactoryCast HMI

Setup Manual
Premium and Quantum HMI Modules

FCHMI eng

Table of Contents

About the Book

At a Glance

Document Scope

This document describes the implementation procedure for FactoryCast HMI, the configuration of the Premium TSX WMY 100 module under PL7 and Unity Pro, and the configuration of the Quantum 140 NWM 100 00 module under Concept and Unity Pro.

Related Documents

Title of Documentation	Reference Number
Communication Applications Setup Manual, Volume 3	TLX DS COM PL7 •• E

User Comments

We welcome your comments about this document. You can reach us by e-mail at TECHCOMM@modicon.com

Configuration of FactoryCast HMI

I

At a Glance

Subject of this Part

This part describes how to configure FactoryCast HMI software.

What's in this Part?

This part contains the following chapters:

Introduction to FactoryCast HMI

1

At a Glance

Subject of this Chapter

This chapter aims to provide you with a description of FactoryCast HMI.

What's in this Chapter?

This chapter contains the following topics:

Introduction to FactoryCast HMI

General

FactoryCast HMI is a package comprising configuration software and a PLC module which can be used to execute HMI applications built into an Ethernet module and based on web technologies. Two types of FactoryCast HMI modules are available according to the PLC:

- one module for Premium: TSX WMY 100,
- one module for Quantum: 140 NWM 100 00,

FactoryCast HMI modules have two profiles:

- they have a communication profile comprising all the standard communication functions,
- and they have an HMI profile enabling a FactoryCast HMI application to be executed.

The FactoryCast HMI configuration tool is a Windows-based software program which can be used to create, test and manage FactoryCast HMI applications.

FactoryCast HMI Built-In Services

The following FactoryCast HMI services are built into a module:

- **PLC** (See *"PLC" service, p. 27*): used to declare and define PLC parameters, and to create and configure FactoryCast HMI variables,
- **Database connection** (See *"Database connection" Service, p. 42*): used to archive FCHMI variables in an external database (SQL Server, Oracle, etc.),
- **Email** (See *"Email" Service, p. 32*): used to send an Email when the status of a variable changes, a threshold is exceeded, or an alarm triggered, etc.
- **Calculation** (See *"Calculation" Service, p. 38*): used to perform calculations using FCHMI variables without employing CPU resources on the PLC.

Terminal Configuration

Required Operating System

FactoryCast HMI configuration software requires one of the following operating systems to be present on the terminal:

- Windows 2000 SP2,
- Windows XP Computing and Professional.

Minimum configuration

The following table provides the minimum terminal characteristics necessary to implement FactoryCast HMI software:

Elements	Characteristics
System	Pentium 800 MHz
RAM	Windows 2000/XP 256 Mb
Hard disk	100 Mb for the software
Ports	TCPIP / Ethernet communication
Monitor	800*600 minimum

Typical configuration

The following table shows a configuration for attaining optimal performance with FactoryCast HMI software.

Elements	Characteristics
Systems	Pentium 1.2 GHz
RAM	Windows 2000/XP 512 Mb
Hard disk	130 Mb for the software
Ports	TCPIP / Ethernet communication
Monitor	800*600 or SVGA with 24 bit color management recommended)

Please note:

A different configuration may be required if other software is used simultaneously with FactoryCast HMI.

Installing a JVM

FactoryCast HMI software requires the installation of Sun JVM version 1.4.1-02 or above. The installation software for the Sun JVM is supplied on the FactoryCast HMI CD.

Preparing for Factory Cast HMI installation

Foreword

If a previous version of FCHMI is already installed, it must be uninstalled first (FactoryCast HMI offers to perform the uninstall for you if this is the case).

The installation procedure is as follows: insert the CD-ROM into the CD drive. The CD is Auto-run, therefore if your PC is set up for this feature you should see the FCHMI main window. If Auto-run is disabled or does not work:

- click on the Start button in the task bar,
- select Settings -> Control panel,
- Click on the Add/Remove Programs icon in the Control Panel,
- click on the Install/Uninstall tab then click on the Install button and follow the instructions, Click on Add New Programs, then on the CD-ROM or floppy disk button and follow the instructions,
- the Install Tool will automatically find the FactoryCast HMI Setup.exe program on the CD and will also display the path and file name then prompt you to perform the installation.

Importing existing projects

Existing projects are not visible once FactoryCast HMI is reinstalled. To make these projects visible with FactoryCast HMI, click on: `Project` then on `Import` and `Existing projects in the workspace`.

Configuration of FactoryCast HMI

2

At a Glance

Subject of this Chapter

This chapter describes the procedure to be followed to configure FactoryCast HMI. It describes all the built-in services (PLC, E-mail, Calculation and Database).

What's in this Chapter?

This chapter contains the following sections:

2.1 The FactoryCast HMI configuration tool: available functions

FactoryCast HMI configuration tool: main functions

Overview

The main functions of the FactoryCast HMI configuration tool are described in the table below:

Menu	Sub-menu	Overview
Project	New	- Used to create a new project or `service`, - Used to create personal `files` or `folders`. These files or folders will have no effect on FactoryCast HMI operations.
	Save all	- Used to save all your services and folders with one click of the mouse.
	Import	- Used to `import` an existing project.
	Export	- Used to `export` the current project. The exported project will be zipped (.zip extension).
	Check project	- Used to check the state of the project before transferring the application to the module (recommended procedure).

Menu	Sub-menu	Overview
Module	Total transfer	● Used to transfer the whole application, either from your local PC to a module, or from a module to your local PC.
	Connection	● Used to connect to the `module` or to the `simulator`.
	Disconnection	● Used to disconnect from the `module` or from the `simulator`.
	Shut down all services	● Used to shut down all current services. The application will be disabled.
	Start up all services	● Used to start up all services after they have been shut down.
	Re-start module	● Used to re-start the module. Re-starting the module has no effect on the state of current services.
	Format module	● Used to restore the module to its original state. Once the module has been formatted, only the built-in web site remains in the module.
	Locate module address	● Provides the IP address for the module.
	Properties	● Provides access to module properties.
Service	Save.......	● Saves the selected service.
	Stop	● Shuts down the selected service.
	Run	● Runs the selected service, if the service has been shut down.
	Statistics	● Shows current state of the selected service (incoming messages, outgoing messages ...).
Options	Configure external tool	● Configures an external tool you want to run under FactoryCast HMI (e.g. Unity Pro).
	Graphic Editor	● Creates graphic pages in online mode.
	Data Editor	● Creates tables of variables in online mode.
	Default display	● Restores default three dimensional display.
Help	Help	● Accesses FactoryCast HMI Help files.

2.2 Description of Services

At a Glance

Subject of this Section

This section provides you with a description of how to configure the different services available with FactoryCast HMI.

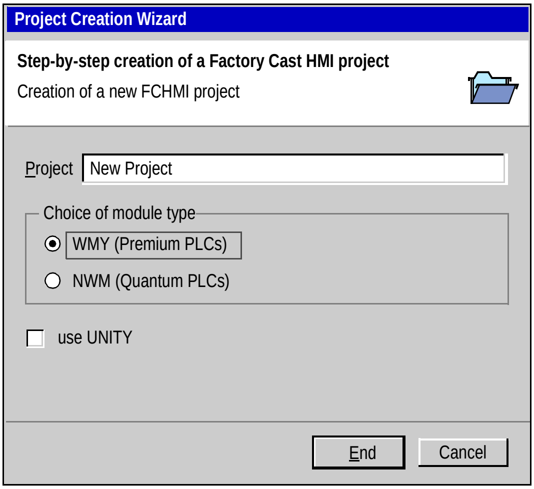
What's in this Section?

This section contains the following topics:

How to Create a New Project

Creating a Project

When you start FactoryCast HMI the first time, the workspace is empty. You must therefore either import a project, or create a new project:



Project Creation Wizard

Step-by-step creation of a Factory Cast HMI project

Creation of a new FCHMI project

Project: New Project

Choice of module type

● WMY (Premium PLCs)

○ NWM (Quantum PLCs)

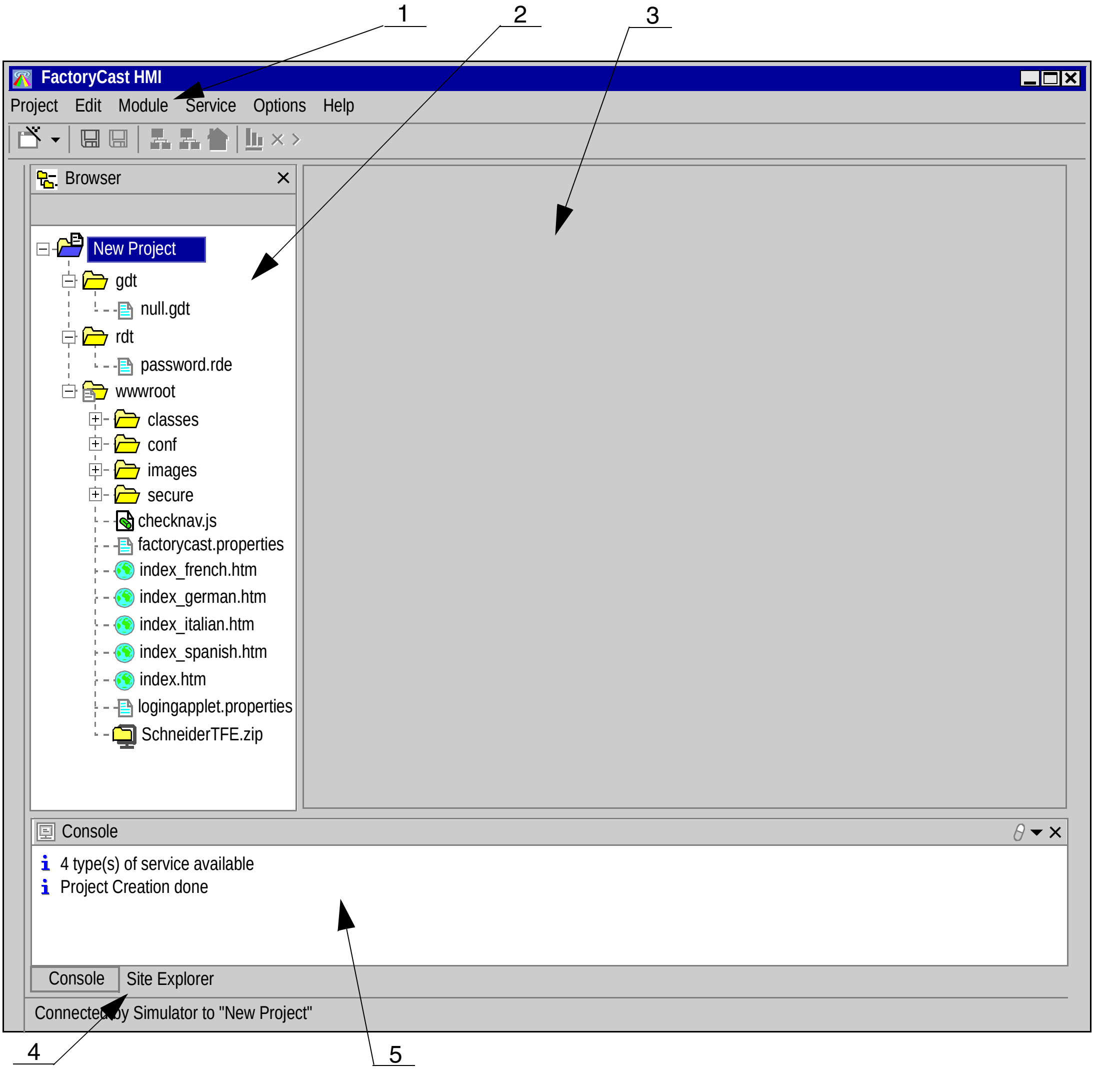
□ use UNITY

Several configurations are available, according to the FactoryCast HMI module type and software workshop being used. The configuration selected will determine a driver and a protocol for the new project as follows:

	Without "Unity in use"	With "Unity in use"
WMY (Premium PLCs)	- Driver: BUSX_ASYNC - Protocol: UNITE	- Driver: BUSX_ASYNC - Protocol: UMAS
NWM (Quantum PLCs)	- Driver: BUSX - Protocol: MODBUS	- Driver: BUSX - Protocol: UMAS

Main window contents

When the new project has been created, the project structure appears in the "Browser".



i 4 type(s) of service available
i Project Creation done

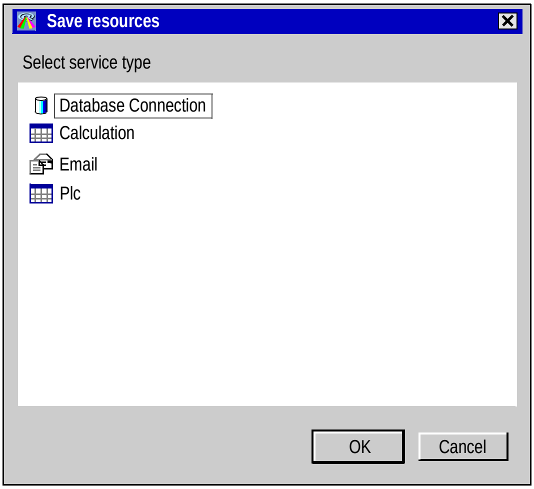
Connected by Simulator to "New Project"

The screen is divided into four main zones which are:

Zone	Description
1	This zone contains the toolbar with its main functions (project creation, archiving, application transfer, site viewing, and creation of services, etc.).
2	This zone is called the "Browser". When you create a project, it is displayed in this window. This window provides an overall view of the application. Three main directories are visible for the new project: ● gdt: Graphic editor (See *Graphic Editor, p. 118*). The directory displays pages created using the graphic editor. There are two different ways of creating pages with a graphic editor: ● either via the FactoryCast HMI configuration tool, by connecting to the simulator then clicking on "Options" and "Graphic editor", ● or via the module web site, by clicking on "Diagnosis and Configuration" then on "Graphic editor". In this case, the pages will be directly created in the module. ● rdt: Data Editor (See *Data Editor, p. 102*). This directory displays pages created using the data editor. There are two ways of creating pages using the data editor: ● either via the FactoryCast HMI configuration tool, by connecting to the simulator then clicking on "Options" and "Data editor", ● or via the module web site, by clicking on "Diagnosis and Configuration" then on "Data editor". In this case, the pages will be directly created in the module. ● wwwroot: This directory contains all the web site files for the FactoryCast HMI module. You can therefore customise the web site by using these files.
3	This zone is used to configure project services (See *Description of Services, p. 22*).
4	This zone is used to display the console or to see the contents of a given site.
5	This zone shows the operations performed using the configuration tool.

Services available

Once you have created the project, you can add HMI services by clicking on "Project" then on "New" and "Service". Four services are available:



The services are as follows:

- **PLC** (See *"PLC" service, p. 27*)
- **Database connection** (See *"Database connection" Service, p. 42*)
- **Email** (See *"Email" Service, p. 32*)
- **Calculation** (See *"Calculation" Service, p. 38*)

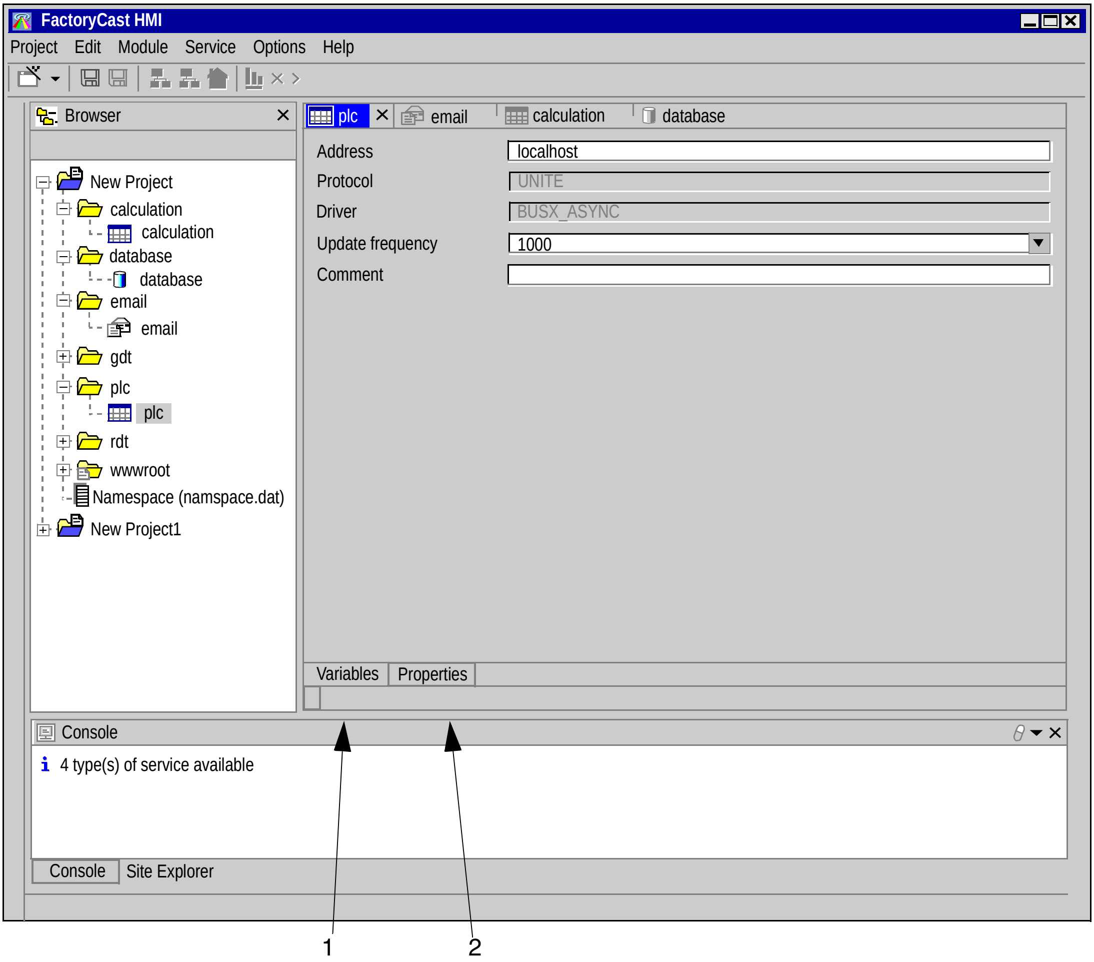
"PLC" service

General

The "PLC" service is used to create FactoryCast HMI variables associated with PLC variables. These variables are associated with symbols which will be used by other services (calculation, database or email).

Note: You must create the "PLC" service, as all the other services and the data and graphic editor use this service.

Properties configuration screen for the "PLC" service:



Address: localhost
Protocol: UNITE
Driver: BUSX_ASYNC
Update frequency: 1000
Comment:

4 type(s) of service available

Meaning of numbering on above screen:

Number	Function
1	This button is used to configure the service variables (See *Configuration of variables, p. 30*).
2	This button is used to configure the service properties (See *Configuration of Properties, p. 29*).

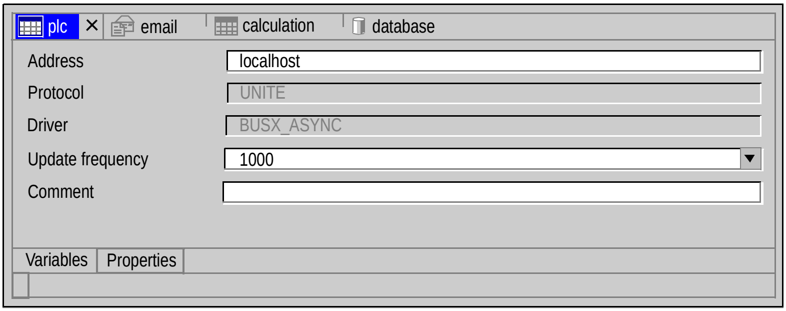
How to Create a "PLC" Service

This table below describes the procedure you need to follow to create a "PLC" service:

Step	Action
1	Click on the name of your project in the browser,
2	To add the new service you can: ● click on "Project" then "New" and select "Service" or, ● right-click in the browser, then click on "New" and select "Service".
3	A new window opens. Click on "Select",
4	Click on "Plc" then "Ok",
5	Enter the service name, or click on "End" to keep the name by default,
6	The "PLC" service has been created. You now have to configure it. You may create up to 5 "PLC" services.

Configuration of Properties

The properties configuration screen for the "PLC" service is as follows:



Address: localhost

Protocol: UNITE

Driver: BUSX_ASYNC

Update frequency: 1000

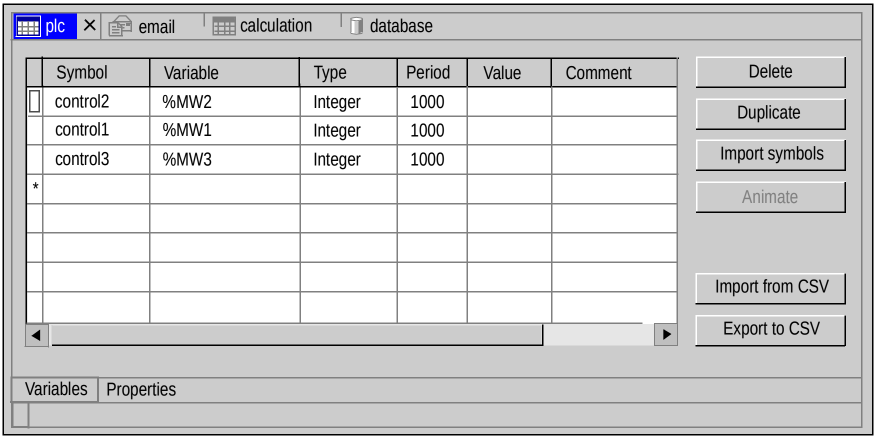
Comment:

This table describes the various fields that make up the properties configuration screen:

Fields	Function
Address	FactoryCast HMI variables may be associated with remote PLCs. The field is used to define the PLC address. If the FactoryCast HMI variable is associated with a variable in the local PLC, enter "localhost" in this field. If this is not the case, use X-Way transparent communication (remote PLC), by entering the X-Way address of the PLC concerned. In this case, the address is of the network.station type (i.e. on two levels only).
Protocol	The protocol is determined by the project type (shaded), i.e. MODBUS (for Quantum PLCs), or UNITE (for Premium PL7s), or UMAS (for Premium/ Quantum Unity).
Driver	The driver is determined by the project type (shaded), i.e. either BUSX_ASYNC for Premium modules, or BUSX for Quantum modules.
Update frequency	Used to define the default frequency with which PLC variables are read.
Comment	Used to add a comment to the "PLC" service properties. This comment is only visible in this screen.

Configuration of variables

The "PLC" variables configuration screen is as follows:



Symbol	Variable	Type	Period	Value	Comment
control2	%MW2	Integer	1000		
control1	%MW1	Integer	1000		
control3	%MW3	Integer	1000		

This table describes the various fields that make up the variables configuration screen:

Fields	Function
`Symbol`	Name of FactoryCast HMI variable.
`Variable`	Physical address of the PLC variable.
`Type`	Used to define the type of variable (See *Notes, p. 152*). The following types may be used with FactoryCast HMI: ● BOOL, ● EBOOL, ● SHORT, ● USHORT, ● INT, ● UINT, ● DINT, ● REAL, ● TIME, ● DATE, ● TOD, ● STR,
`Period`	PLC variable scan frequency (in ms).
`Comment`	Used to add a comment to the "PLC" service properties. This comment is only visible in this screen.

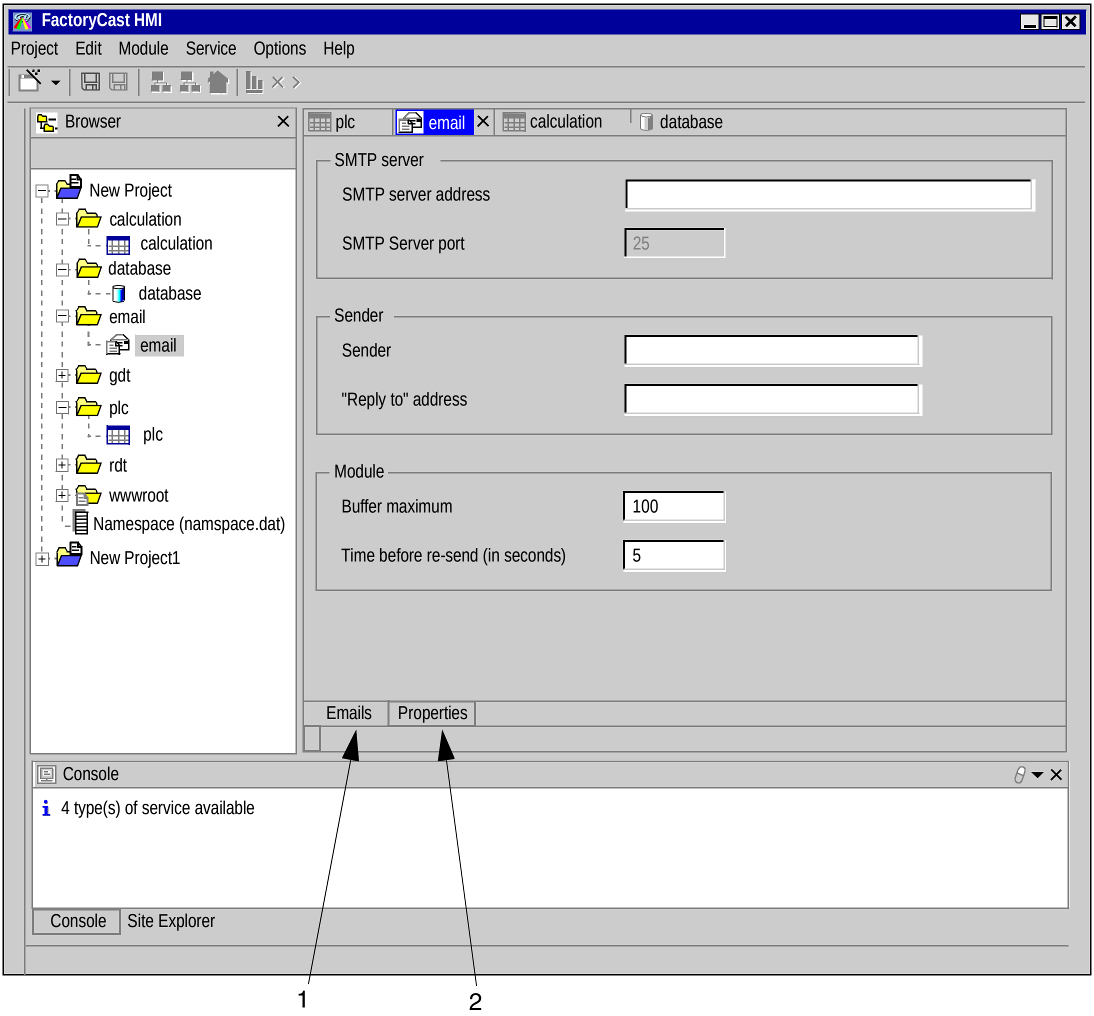
Fields	Function
`Import symbols`	This button is used to import PLC symbols from a FEF (PL7), STU (Unity Pro), XVM (Unity Pro) or PRJ (Concept) file. **PLEASE NOTE**: Concept projects (.PRJ) can only be imported if the ExportVariables parameter in concept.ini is set to 1. You must modify this parameter before saving the project under Concept (re-start Concept after modifying the parameter setting).
`Animate`	This button is used to animate variables when the project is in online mode, with connections either to the simulator (See *Connecting to/Disconnecting from the Simulator, p. 53*) or to the module (See *Connecting / Disconnecting the Module, p. 51*).
`Import from CSV`	This button is used to import symbols from a CSV file. This file may be created using Excel or a text editor (in this case, change the .txt file extension to .csv). The following syntax must be used: `symbol;variable;type;period;comment`
`Export to CSV`	This button is used to export the variable table for the current service to a CSV file.

Note: You may define up to 1000 FactoryCast variables.

"Email" Service

General

The Email service is used to send an email when an alarm is triggered, when there is a change in the status of a variable, or a threshold is overrun, etc.
It is associated with the FactoryCast HMI variables created in the "PLC" or the "Calculation" services.
Properties configuration screen for the Email service:



SMTP server

SMTP server address:

SMTP Server port: 25

Sender

Sender:

"Reply to" address:

Module

Buffer maximum: 100

Time before re-send (in seconds): 5

Console

4 type(s) of service available

Meaning of numbering on above screen:

Number	Function
1	This button is used to configure emails sent by the Email service (See *Configuring the Email service, p. 36*).
2	This button is used to configure email properties (See *Configuring the Server, p. 34*).

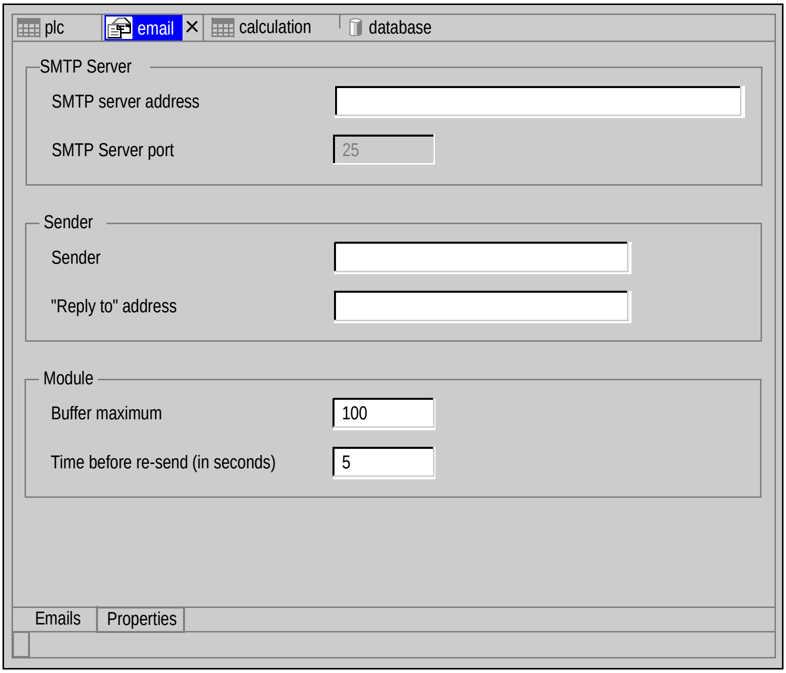
How to Create an "Email" Service

The table below describes the procedure you need to follow to create an "Email" service:

Step	Action
1	Click on the name of your project in the browser,
2	To add the new service you can: ● click on "Project" then "New" and select "Service" or, ● right-click in the browser, then click on "New" and select "Service".
3	A new window will open. Click on "Select",
4	Click on "Email" then "Ok",
5	Enter the service name, or click on "End" to keep the name by default,
6	The "Email" service has been created. You must now configure it.

Configuring the Server

The properties configuration screen for the "Email" service is shown below:



SMTP Server

SMTP server address

SMTP Server port 25

Sender

Sender

"Reply to" address

Module

Buffer maximum 100

Time before re-send (in seconds) 5

This table describes the various fields that make up the properties configuration screen:

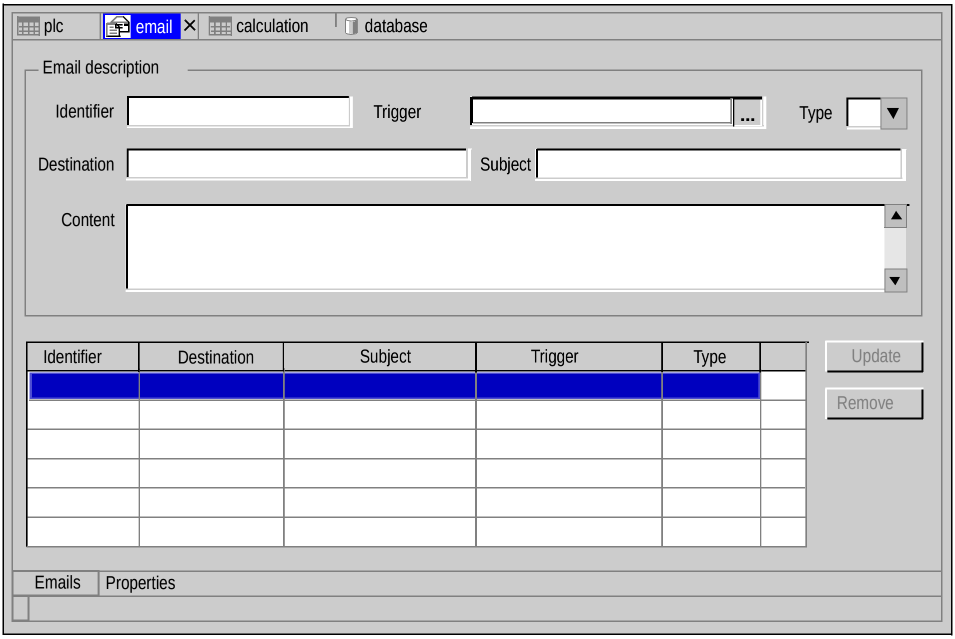
Fields	Function
`SMTP server address`	SMTP address server address.
`SMTP Server port`	TCP port used by the SMTP server (generally port 25).
`Sender`	Email address for the PLC sending the email. This address will identify the PLC when the user receives the email.
`"Reply to" address`	Email address to which a reply will be sent if the user clicks on the "Reply" button.
`Buffer maximum`	Maximum number of mails which can be stored in the buffer memory before being sent. ● Default value = 100, ● Minimum value = 30, ● Maximum value= 200.

Fields	Function
`Time before re-send (in seconds)`	Delay before emails stored in the buffer memory are re-sent after failure. ● Default value = 15s, ● Minimum value = 5s, ● Maximum value= 3600s.

Note: When the maximum number of mails is reached (e.g.: 100), no further messages can be stored.

Configuring the Email service

The Email configuration screen is shown below:



This table describes the various fields that make up the properties configuration screen:

Fields	Function	
`Identifier`	Name for the email.	
`Trigger`	Event that will trigger the email.	
`Type`	**NY**: Notification	Triggered by bit status change or word value change.
	RE: Rising Edge	Triggered by a bit rising edge or by an increasing word value.
	FE: Falling Edge	Triggered by a bit falling edge or by a decreasing word value.
	BQ: "Bad Quality"	Triggered if the trigger status is "Bad quality".
`Destination`	Destination email address.	
`Subject`	Subject of mail.	
`Content`	Content of mail.	

Note: Dynamic data can be included in the message. To include dynamic data, place brackets before and after the FactoryCast HMI variable name. If, for instance, you want to know the value of the "value1" variable created in the PLC service, simply write {plc.plc1.value1}. You can also include comments before and after the brackets.
The number of e-mails you can configure in the Email service is restricted to 100.

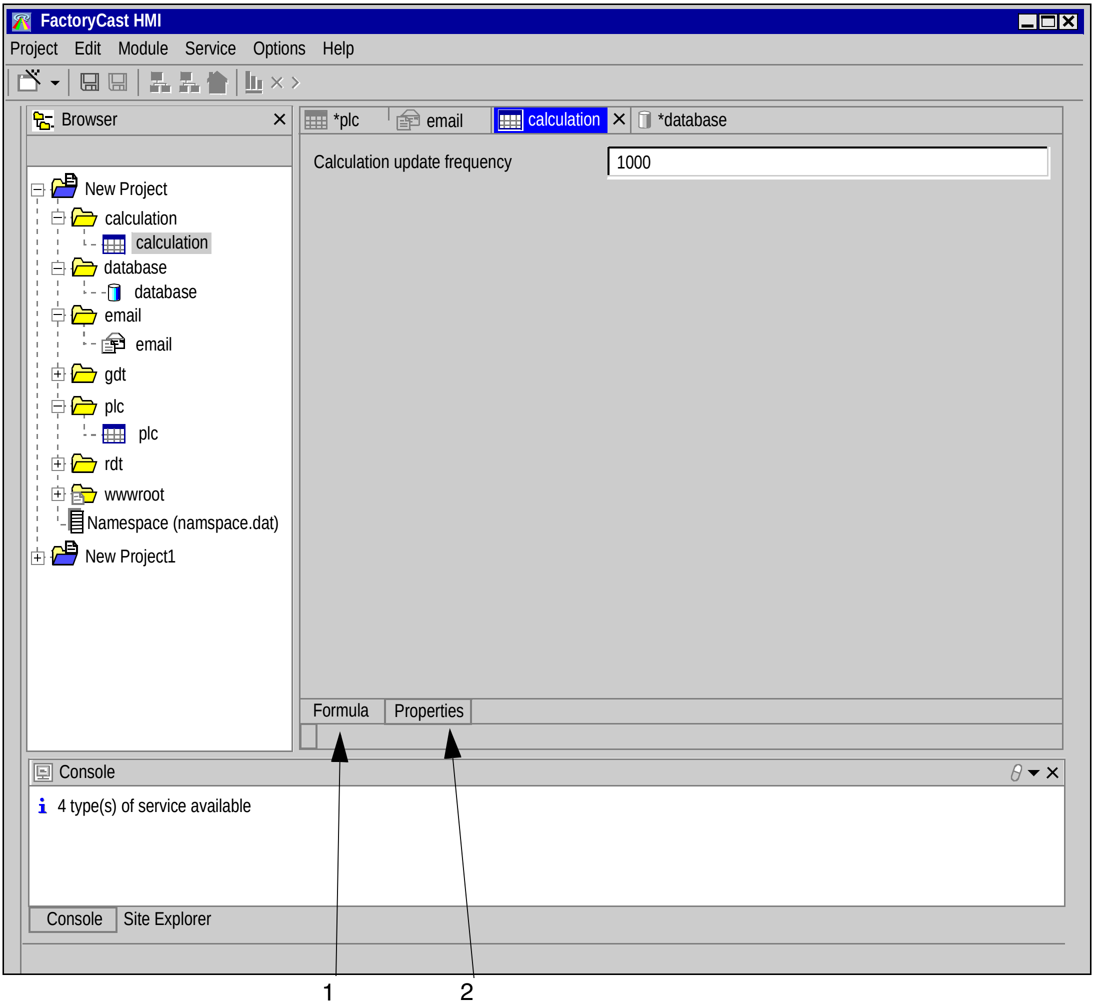
Note: The following fields are mandatory to record and save an email:`Identifier`, `Trigger` and `Destination`.

"Calculation" Service

General

The "Calculation" service is used to perform operations on or to combine FactoryCast HMI variables. It can also be used to view the result of operations in real time to facilitate application debugging.

The properties configuration screen for the "Calculation" service is shown below:



Calculation update frequency: 1000

4 type(s) of service available

Meaning of numbering on above screen:

Number	Function
1	This button is used to configure formulae (See *Configuring formulae, p. 40*).
2	This button is used to configure properties common to all formulae (See *Configuration of Properties, p. 39*).

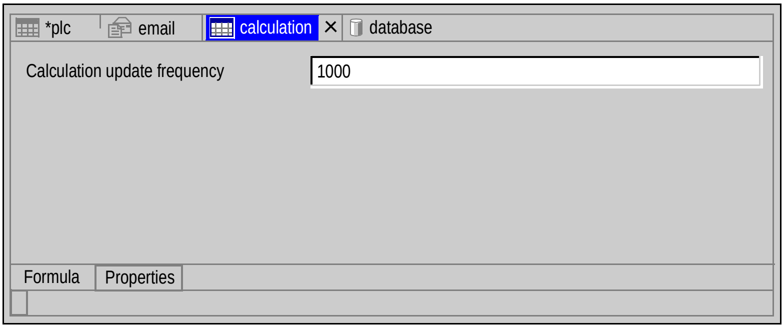
How to Create a "Calculation" Service

The table below describes the procedure you need to follow to create a "Calculation" service:

Step	Action
1	Click on the name of your project in the browser,
2	To add the new service you can: ● click on "Project" then "New" and select "Service" or, ● right-click in the browser, then click on "New" and select "Service".
3	A new window will open. Click on "Select",
4	Click on "Calculation" then "Ok",
5	Enter the service name, or click on "End" to keep the name by default,
6	The "Calculation" service has been created . You now have to configure it.

Configuration of Properties

The properties configuration screen for the "Calculation" service is shown below:



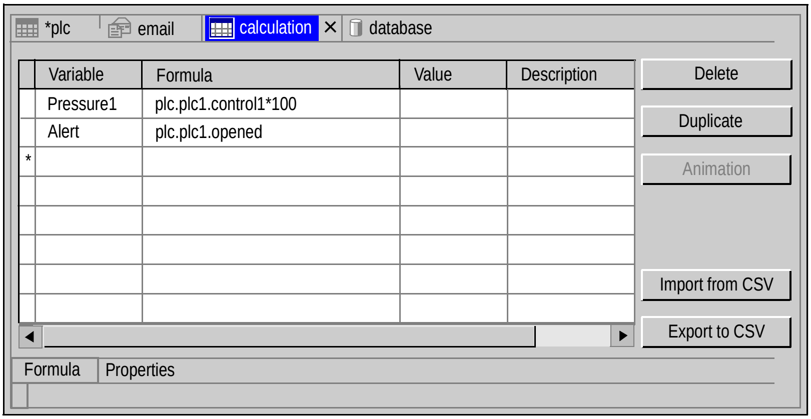
Calculation update frequency 1000

This table describes the various fields that make up the properties configuration screen:

Fields	Function
`Calculation update frequency`	Frequency of execution of formulae using FactoryCast HMI variables.

Configuring formulae

The formula configuration screen for the "Calculation" service is shown below:



Variable	Formula	Value	Description
Pressure1	plc.plc1.control1*100		
Alert	plc.plc1.opened		
*			

This table describes the various fields that make up the formula configuration screen:

Fields	**Function**
`Variable`	Name of FactoryCast HMI variable.
`Formula`	Formula associated with the FactoryCast HMI variable using the PLC variables created and the "Formula" variables. The operators are limited,
`Value`	Gives the result of the formula (value of the variable) in real time,
`Description`	Comment on the variable created,
`Animation`	This button is used to switch to simulation mode (See *Connecting to/Disconnecting from the Simulator, p. 53*).
`Import from CSV`	This button is used to import symbols from a CSV file. This file may be created using Excel or a text editor (in this case, change the .txt file extension to .csv). The following syntax must be used: `variable;formula;description`
`Export to CSV`	This button is used to export the variable table for the current service to a CSV file.

Note: The maximum number of formulae you can configure in this service is 1000.

List of Available Operators

The following list provides details of numeric binary, Boolean and unary operators. The result of the operation is displayed in the "Result type" column:

Type	Operator	Description	Example
Binary numeric	+	addition	1+2 displays the result "3"
	-	subtraction	5-2 displays the result "3"
	*	multiplication	5*2 displays the result "10"
	/	division	15/3 displays the result "5"
	<	less than	1<2 displays the result "true"
	>	greater than	1>2 displays the result "false"
	<=	less or equal to	1<=2 displays the result "true"
	>=	greater or equal to	2>=2 displays the result "true"
	==	equal to	3==2 displays the result "false"
	!=	not equal to	1!=2 displays the result "true"
Boolean numeric	&	and	1+1 displays the result "1"
	|	or	1 | 0 displays the result "1"
unary	!	inverse	!1 displays the result "0"
	-	sign inversion	-1 displays -1
	.<. ? . : .		a<b ? c:d returns c if a<b otherwise the result is d

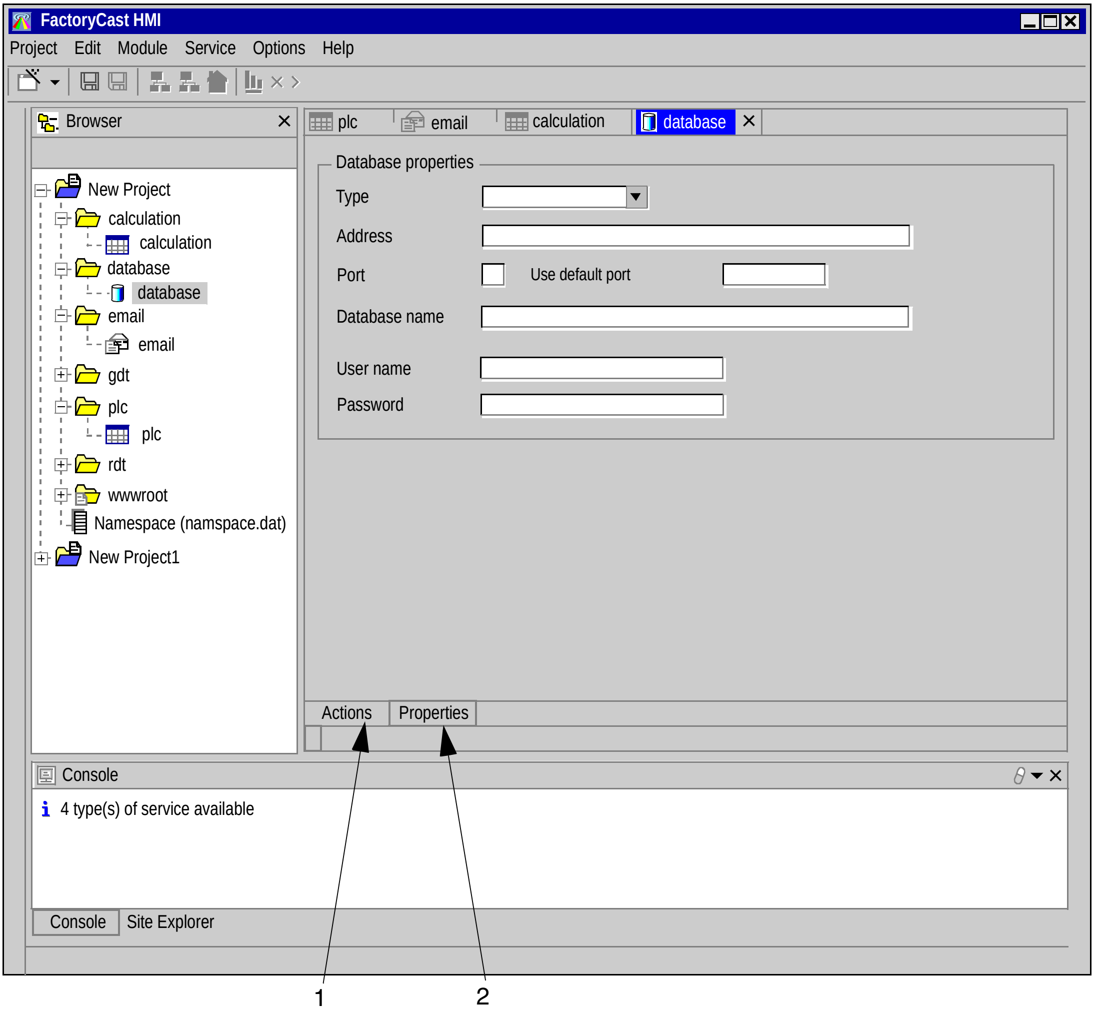
Execution of Formulae

Formulae are executed periodically, according to the frequency configured in the `Properties` screen. The `formula` cells are interpreted then executed one by one from top to bottom. The associated variable configured in the variable field is updated with the new value.

"Database connection" Service

General

The "Database connection" service is used to archive FactoryCast HMI variables in an external database.
The properties configuration screen for the "Database connection" service is shown below:



4 type(s) of service available

Meaning of numbering on above screen:

Number	Function
1	This button is used to configure the variables to be archived (See *Configuring the log file, p. 45*).
2	This button is used to configure the general properties of the database (See *Configuration of Properties, p. 44*).

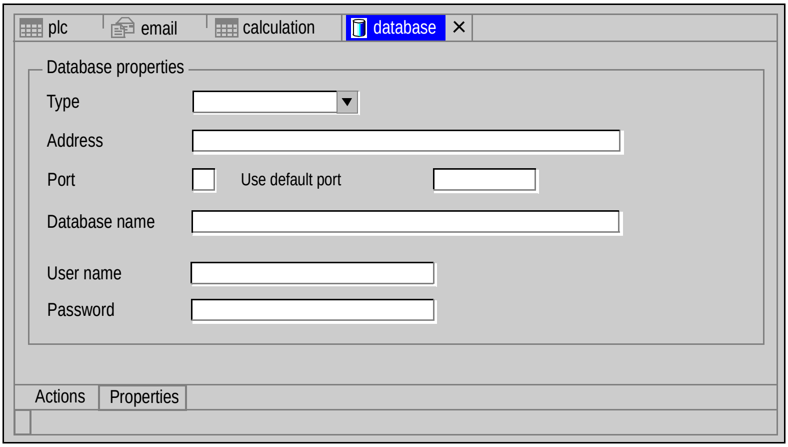
How to Create a "Database connection" Service

The table below describes the procedure you need to follow to create a "Database connection" service:

Step	Action
1	Click on the name of your project in the browser,
2	To add the new service you can: ● click on "Project" then "New" and select "Service" or, ● right-click in the browser, then click on "New" and select "Service".
3	A new window will open. Click on "Select",
4	Click on "Database" then "Ok",
5	Enter the service name, or click on "End" to keep the name by default,
6	The "Database connection" service has been created. You now have to configure it.

Configuration of Properties

The properties configuration screen for the "Database connection" service is shown below:



Database properties

Type

Address

Port □ Use default port

Database name

User name

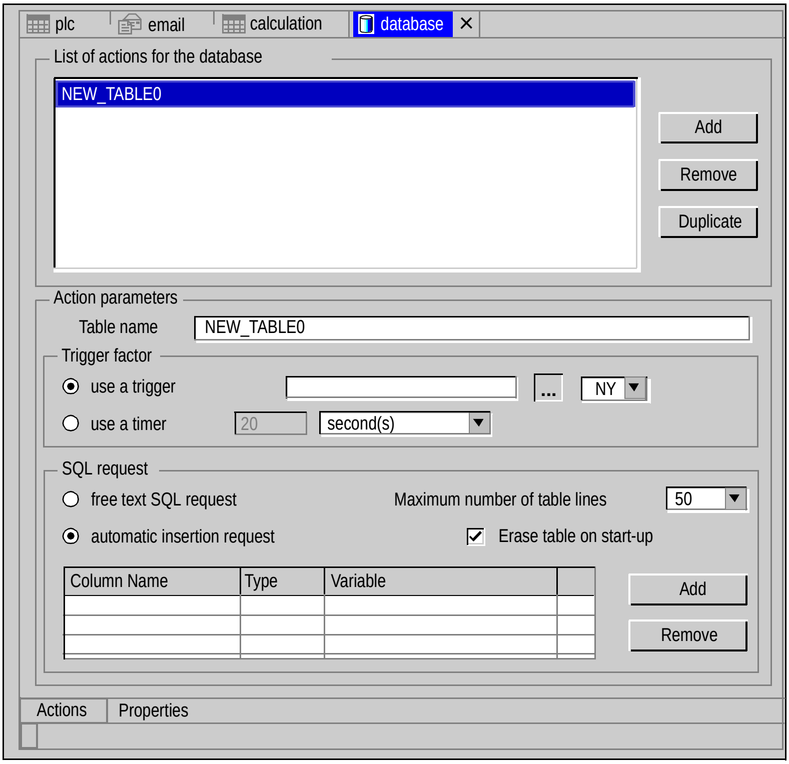
Password

This table describes the various fields that make up the database server configuration screen:

Fields	Function
`Type`	● Oracle 8i, ● MySql, ● SQL Server.
`Address`	IP address of the database server,
`Port`	Used to validate a default address,
`Use default port`	Port used by the database to access the server,
`Database name`	Name of the database that will contain all data configured by the user.
`User name` & `Password`	Protection of database and access by username and password (database connection string).

Configuring the log file

The configuration screen for variables to be archived for the "Database connection" service is shown below:



List of actions for the database

NEW_TABLE0

Action parameters

Table name NEW_TABLE0

Trigger factor

● use a trigger NY

○ use a timer 20 second(s)

SQL request

○ free text SQL request

Maximum number of table lines 50

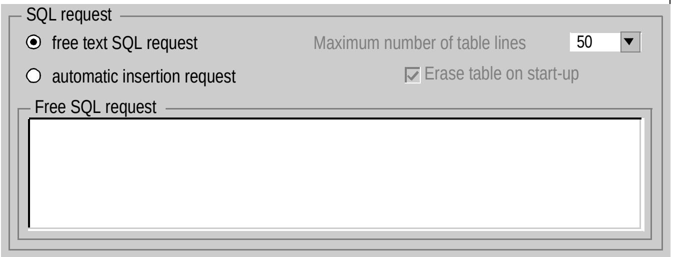
● automatic insertion request

☑ Erase table on start-up

Column Name	Type	Variable

This table describes the various fields that make up the log file configuration screen:

Fields	Function
`List of actions for the database`	List of all tables configured in the FactoryCast HMI application on the server. Use the `Add`, `Remove` and `Duplicate` buttons to create, delete or copy a table. When an existing database is duplicated, the new name will have the suffix `_copy` added to its initial name,

Fields	Function
`Action parameters`	Name of the new archiving table. If the table does not exist in the database, FactoryCast HMI will create it on the first connection. To trigger logging of variables, an event must be configured. This will either be a periodic event (`use timer`) in which case you will have to set a timebase, or it will be an event associated with a FactoryCast HMI variable (`use trigger`) in which case you will have to provide the name of this variable (e.g.: calculation.calculation1.Pressure1).
`SQL request`	Two options are available: ● You can either develop requests yourself with SQL requests in the `Free Text SQL Request` window,  SQL request ◉ free text SQL request Maximum number of table lines 50 ▼ ○ automatic insertion request ☑ Erase table on start-up Free SQL request ● or you can use the table provided for this purpose specifying the name of the column, the type of the variable and the name of the variable. In this case, the column name syntax must comply with the database syntax. As a general rule, avoid using punctuation marks, upper case letters, and hyphens, (refer to your database documentation).

Note: You can configure a maximum of:
- 2 database services,
- 10 tables per database,
- 100 columns per table.

Database properties

The table below shows the correspondences for Xopen types according to the type of database used:

Xopen type	**MySql,**	**Oracle 8i,**	**SQL Server.**
BOOLEAN	SMALLINT	Device number 1	SMALLINT
TINYINT	SMALLINT	SMALLINT	SMALLINT
SMALLINT	SMALLINT	NUMBER	SMALLINT
INTEGER	INTEGER	NUMBER	INTEGER
BIGINT	(decimal)	NUMBER	NUMERIC(38,0)
DOUBLE	DECIMAL(50,80)	NUMBER	FLOAT(32)
REAL	REAL	REAL	REAL
TIMESTAMP	TIMESTAMP	TIMESTAMP	TIMESTAMP
DATE	DATETIME	DATE	DATETIME
VARCHAR	VARCHAR(255)	VARCHAR(255)	VARCHAR(255)
All other types	VARCHAR(255)	VARCHAR(255)	VARCHAR(255)

2.3 Operating modes

At a Glance

Subject of this Section

This section introduces the different FactoryCast HMI operating modes.

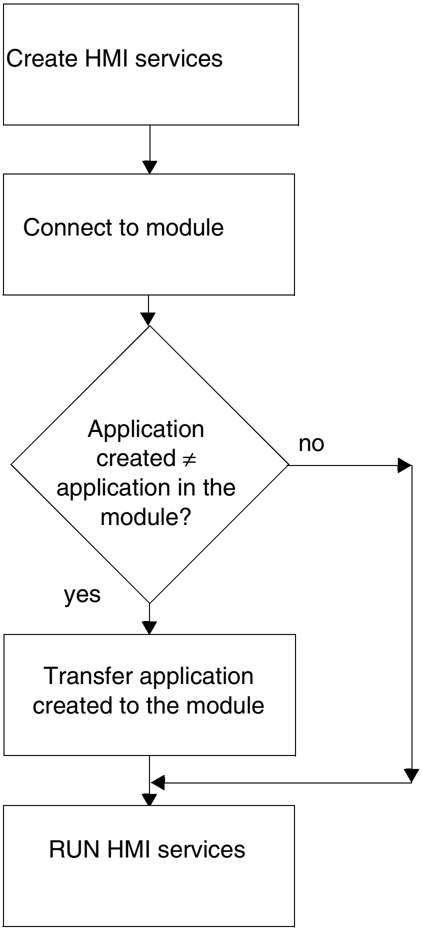
What's in this Section?

This section contains the following topics:

Operating Mode Description

At a Glance

To execute an application in a HMI module, the first step is to create the services, then to transfer the application to the module, then switch to online mode.
The following diagram describes in detail the operating mode for starting up an application:


Create HMI services
Connect to module
Application created ≠ application in the module?
no
yes
Transfer application created to the module
RUN HMI services

Note:

- Configuration of the HMI services (See *Description of Services, p. 22*) is simple, and requires no specific method. The order in which the different services are created in unimportant.
- The connection to the module (See *Connecting / Disconnecting the Module, p. 51*) is used to transfer an application to or from a module. As soon as you connect, the HMI services are set to RUN.
- When you start up a module, the HMI services which were stopped by the user at the end of the previous work session do not start up again automatically.

Connecting / Disconnecting the Module

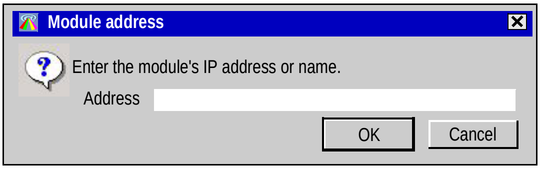
At a Glance

The information below explains the operating mode for executing an application once the HMI services have been created.

Note: Transferring a project to an HMI module clears the existing project. When a project is transferred, if an application is in RUN mode, the new project will only be taken into account once the module is restarted and the old one overwritten.

Connecting to the HMI Module and Recovering the Module Project

The following table shows the procedure for connecting to an HMI module with a view to recovering its application:

Step	Function	Procedure
1	Creating a project	Create a new project: `Project`, then `New`, then `Project`. The project is displayed in the left window.
2	Defining the module address	Click on: `Module` then `Locate module address`. The following window appears:  Enter the module's IP address or name. Address Enter the module’s IP address or name.
3	Transferring the application	Click on `Module` then select `Total transfer` , then `Module -> Pc`. The FactoryCast HMI configuration tool recovers the project for the module. When the transfer is complete, the project is in offline mode, so you can change its content if you wish.
4	Connecting to the module	Click on: `Module` then `Connection` and `Module`. FactoryCast analyses the changes between your project and the module content. If you have modified the project, FactoryCast HMI will ask you to transfer the project. Otherwise the application moves automatically to online mode. You are now connected to the module and the application executes.

Connecting to the HMI Module and Transferring the Project to the Module

The following table shows the procedure for connecting to an HMI module with a view to transferring the project to the module:

Step	Function	Procedure
1	Transferring the application	Click on `Module` then select `Total transfer` , then `Pc -> Module`. FactoryCast HMI transfers the project to the module. Once the transfer is complete, the project is in the module.
2	Connecting to the module	Click on: `Module` then `Connection` and `Module`. FactoryCast analyses the changes between your project and the module content. If no changes have been made to the project, the project will not be transferred to the module. Otherwise the application will be loaded. To take the new application into account, you must restart the module. You are now connected to the module and the application executes.

Disconnecting from the Module

The following table shows the procedure for disconnecting from a HMI module:

Step	Function	Procedure
1	Disconnecting from the Module	Click on: `Module` then `Disconnection`. You have just switched back to offline mode.

Connecting to/Disconnecting from the Simulator

At a Glance

The following information describes the operating mode for launching an application through the FactoryCast HMI simulator.
The FactoryCast simulator lets you execute an application without having to connect to a module. It is therefore possible to test the application from a PC before transferring it to the module.

Note: The graphic and data editors (See *Data and Graphic Editors, p. 101*) are both active in simulation mode. You can therefore modify these pages in simulation mode.
Once the changes are made, carry out a part transfer in order to reduce transfer time.

Connecting to the Simulator

The following table shows the procedure for connecting to the simulator and transferring the application:

Step	Function	Procedure
1	Creating a project	Create a new project: `Project`, then `New`, then `Project`. The project is displayed in the left window.
2	Connecting to the Simulator	Click on: `Module` then `Connection` and `Simulator`. You are now connected to the simulator, the application is in simulation mode.

Disconnecting from the Module

The following table shows the procedure for disconnecting from a HMI module:

Step	Function	Procedure
1	Disconnecting from the Simulator	Click on: `Module` then `Disconnection`. You have just switched back to configuration mode.

Animations

In simulation mode, the variables are animated as follows (value update frequency depends on the update frequency setting):

- bit: value change, 0 or 1,
- word: increment step 1.

Default Web Site for Quantum

3

Default Web Site for Quantum

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 IP address of the module in your browser and enter the default user name and password of "USER". However, Schneider Automation recommends that you complete the setup procedures as outlined in .

This section describes the pages in the default Web site.

What's in this Chapter?

This chapter contains the following sections:

3.1 Quantum Home Pages

Overview of Quantum Home Pages

Overview

When a user accesses the default Web site for Quantum, the user first encounters a FactoryCast Home Page followed by a Quantum Welcome Page. This section describes those two pages.

What's in this Section?

This section contains the following topics:

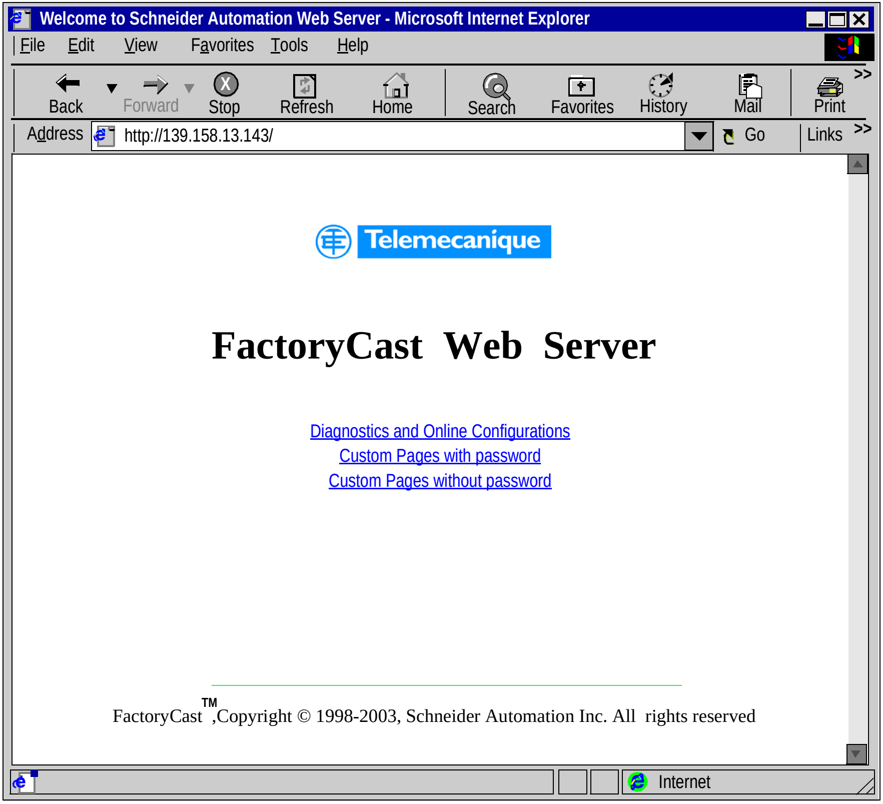
FactoryCast Home Page

Overview

The FactoryCast Home Page is the first page a visitor will encounter. No password is required to access this page.

Home Page

This is how the FactoryCast Home Page looks.



FactoryCast Web Server

Diagnostics and Online Configurations

Custom Pages with password

Custom Pages without password

FactoryCast™,Copyright © 1998-2003, Schneider Automation Inc. All rights reserved

Data

The home page reports on the following.

- Operating system
- Screen resolution
- Browser type and version

Links

The home page offers three links.

- Diagnostics and Online Configurations links to the default diagnostic Web pages, the Data Editor, and the Graphic Editor
- Custom Pages with Password will link to any Web pages you add to the site with password security
- Custom Pages without Password will link to any Web pages you add to the site without password security

Customize

The home page can be customized to meet your specific needs. Simply edit the HTML and download the page to the Embedded Server.

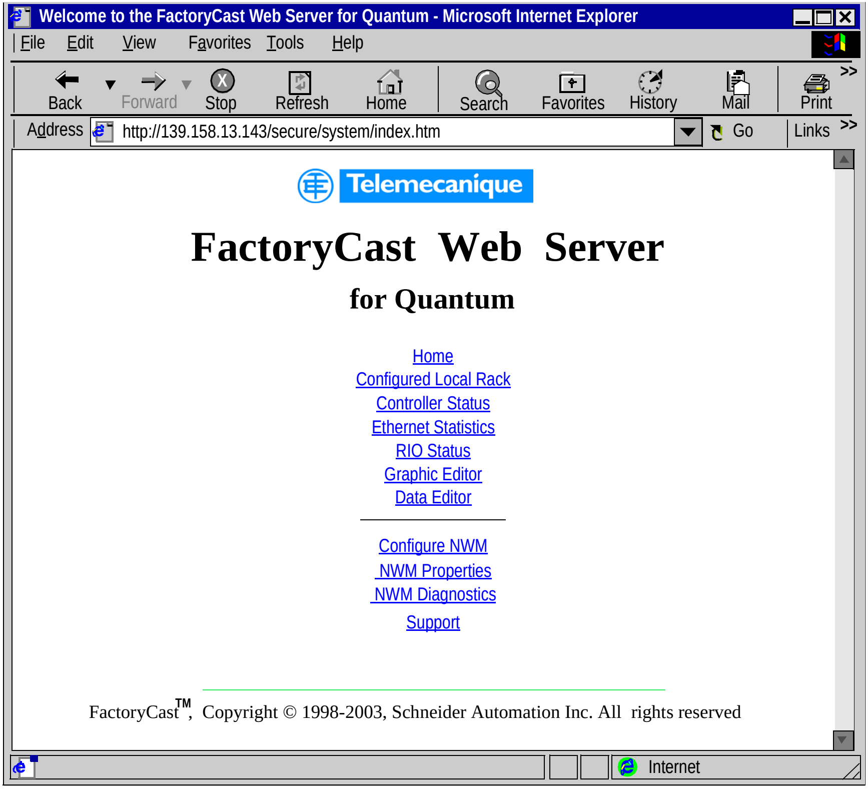
Quantum Welcome Page

Overview

When a visitor clicks on the Diagnostics and Online Configurations link on the FactoryCast Home Page, the visitor is directed to the Quantum Welcome Page. The visitor must supply a user name and password to view this page.

Welcome Page

This is how the Quantum Welcome Page looks.


Welcome to the FactoryCast Web Server for Quantum - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Forward
Stop
Refresh
Home
Search
Favorites
History
Mail
Print
Address
Go
Links
Internet

http://139.158.13.143/secure/system/index.htm

Telemecanique

FactoryCast Web Server

for Quantum

Home
Configured Local Rack
Controller Status
Ethernet Statistics
RIO Status
Graphic Editor
Data Editor

Configure NWM
NWM Properties
NWM Diagnostics
Support

FactoryCast™, Copyright © 1998-2003, Schneider Automation Inc. All rights reserved

Links

The Quantum Welcome Page provides links to several major Quantum diagnostic pages and to the Data Editor and Graphic Editor.

3.2 Local Rack Diagnostics

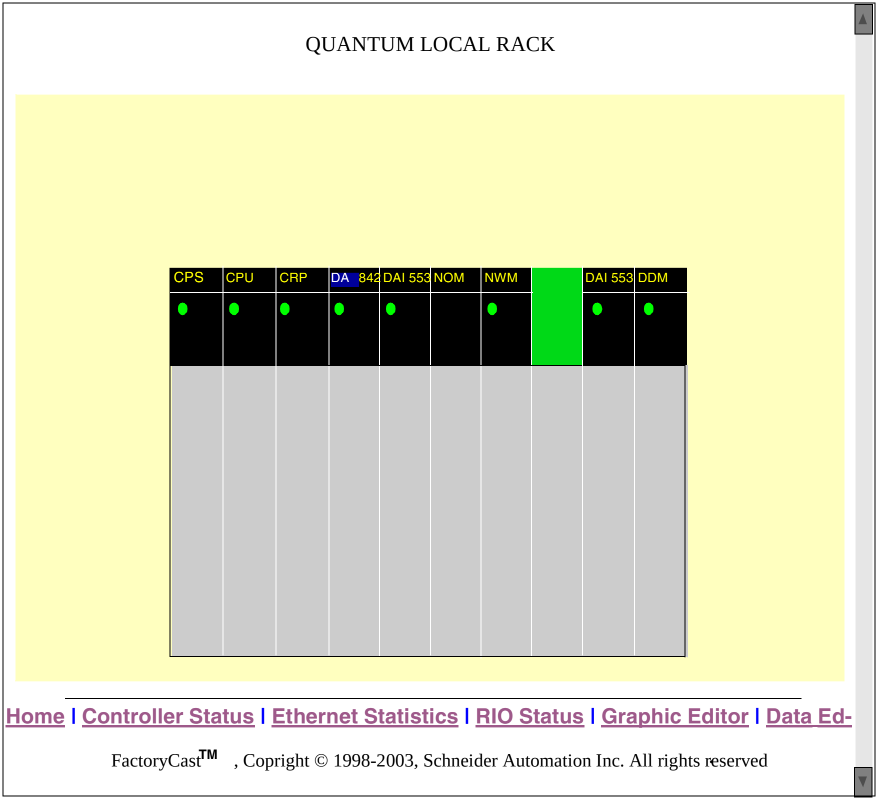
Configured Local Rack Page

Overview

The Quantum Configured Local Rack page displays the current configuration of the local rack, including the controller, Embedded Server module and any I/O modules. The rack can contain up to 16 slots.

Sample Page

Here is an example of a Configured Local Rack page.



QUANTUM LOCAL RACK

CPS
CPU
CRP
DA 842
DAI 553
NOM
NWM
DAI 553
DDM
Home | Controller Status | Ethernet Statistics | RIO Status | Graphic Editor | Data Ed-

FactoryCast^{TM} , Copright © 1998-2003, Schneider Automation Inc. All rights reserved

Data

Each module is displayed in its configured slot in the rack and the following information is provided.

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

Links

If you click on any of the modules, you will reach another Web page with detailed information about that module.

Links at the bottom of the page connect to other major diagnostic pages and the Data Editor and Graphic Editor.

3.3 Controller Diagnostics

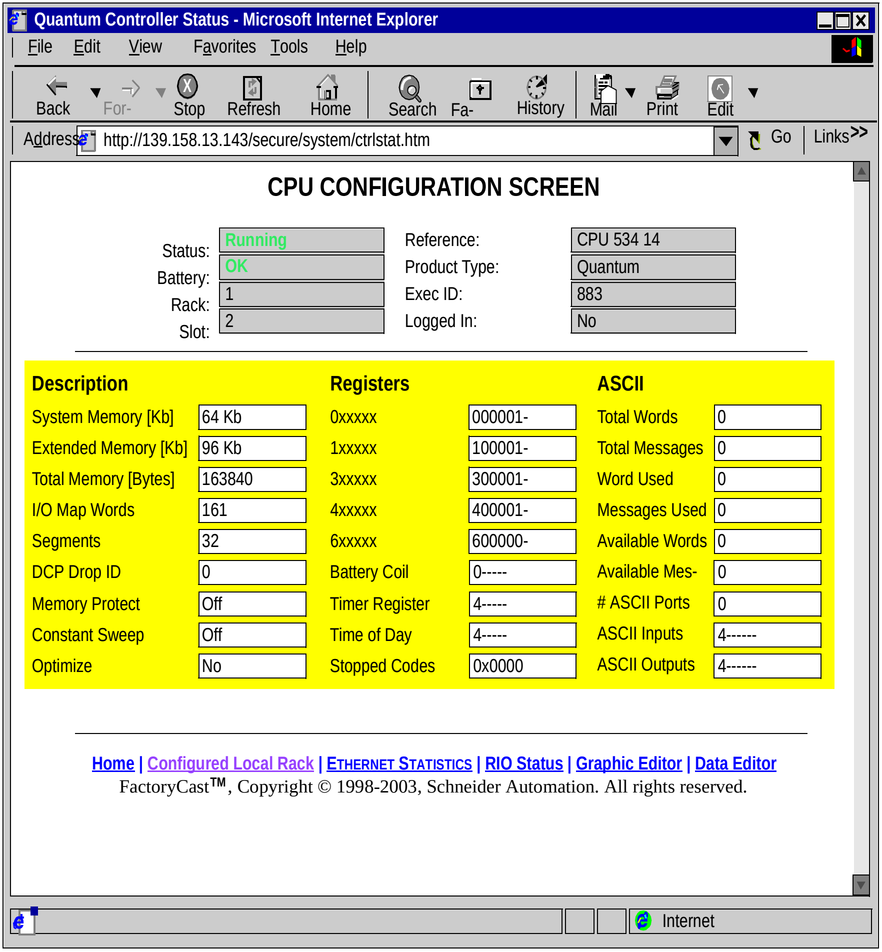
CPU Configuration Page

Overview

The CPU Configuration Page provides up-to-date information about the controller and its configuration. Access this page by selecting the CPU model from the "Configured Local Rack," or the hyperlink "Controller Status" at the bottom of the page.

Sample Page

Here is an example of a CPU Configuration page.


Quantum Controller Status - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back For- Stop Refresh Home Search Fa- History Mail Print Edit
Address
Go
Links>>
Internet

http://139.158.13.143/secure/system/ctrlstat.htm

CPU CONFIGURATION SCREEN

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-	Total Words	0
Extended Memory [Kb]	96 Kb	1xxxxx	100001-	Total Messages	0
Total Memory [Bytes]	163840	3xxxxx	300001-	Word Used	0
I/O Map Words	161	4xxxxx	400001-	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-----	# ASCII Ports	0
Constant Sweep	Off	Time of Day	4-----	ASCII Inputs	4------
Optimize	No	Stopped Codes	0x0000	ASCII Outputs	4------

Home | Configured Local Rack | ETHERNET STATISTICS | RIO Status | Graphic Editor | Data Editor

FactoryCast™, Copyright © 1998-2003, Schneider Automation. All rights reserved.

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.

Links

Links at the bottom of the page connect to other major diagnostic pages and the Data Editor and Graphic Editor.

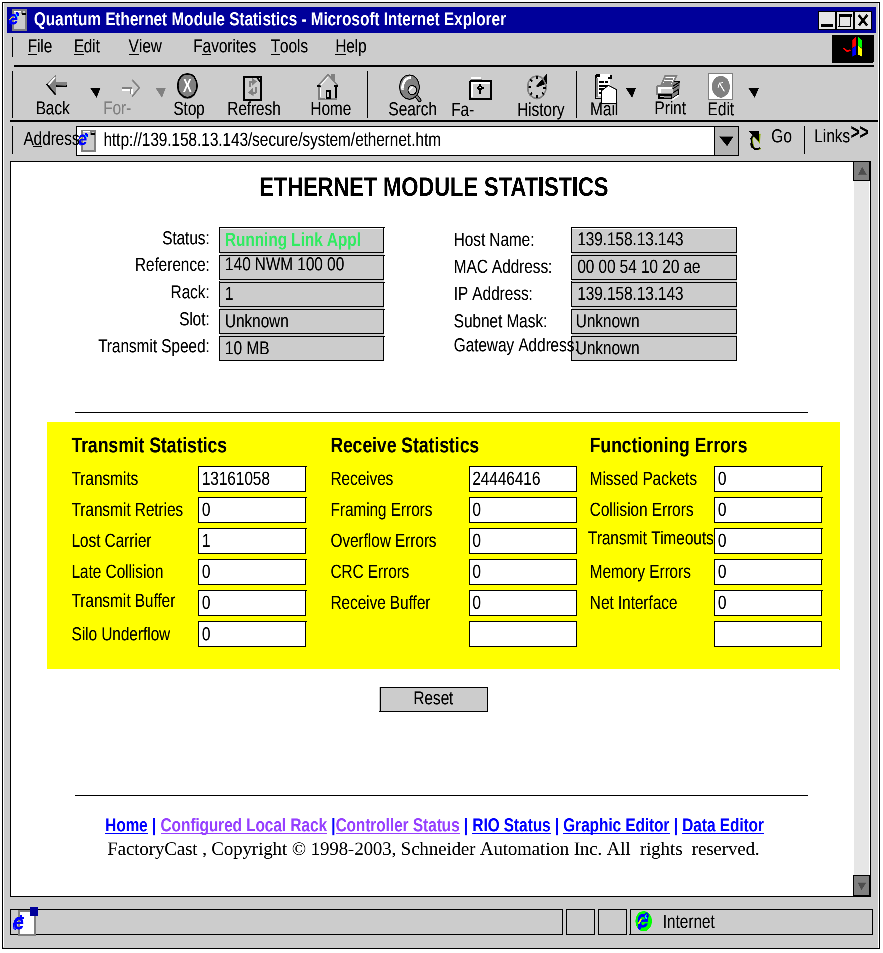
3.4 Ethernet Statistics

Ethernet Statistics Page

Ethernet Statistics Page

The Ethernet Module Statistics page provides information about the status, transmit and receive statistics, and errors for the Embedded Server module.

Here is an example of an Ethernet Module Statistics page.


Quantum Ethernet Module Statistics - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back For- Stop Refresh Home Search Fa- History Mail Print Edit
Address
Go Links>>
Internet

http://139.158.13.143/secure/system/ethernet.htm

ETHERNET MODULE STATISTICS

Status:	Running Link Appl	Host Name:	139.158.13.143
Reference:	140 NWM 100 00	MAC Address:	00 00 54 10 20 ae
Rack:	1	IP Address:	139.158.13.143
Slot:	Unknown	Subnet Mask:	Unknown
Transmit Speed:	10 MB	Gateway Address	Unknown

Transmit Statistics		Receive Statistics		Functioning Errors	
Transmits	13161058	Receives	24446416	Missed Packets	0
Transmit Retries	0	Framing Errors	0	Collision Errors	0
Lost Carrier	1	Overflow Errors	0	Transmit Timeouts	0
Late Collision	0	CRC Errors	0	Memory Errors	0
Transmit Buffer	0	Receive Buffer	0	Net Interface	0
Silo Underflow	0				

Reset

Home | Configured Local Rack |Controller Status | RIO Status | Graphic Editor | Data Editor

FactoryCast , Copyright © 1998-2003, Schneider Automation Inc. All rights reserved.

Note: Refer to *Modicon Quantum Ethernet TCP/IP Module User Guide* 840 USE 107 00 and *Modicon Quantum Ethernet TCP/IP Module User Guide 840* USE 115 00 for definitions of terms.

3.5 RIO Status

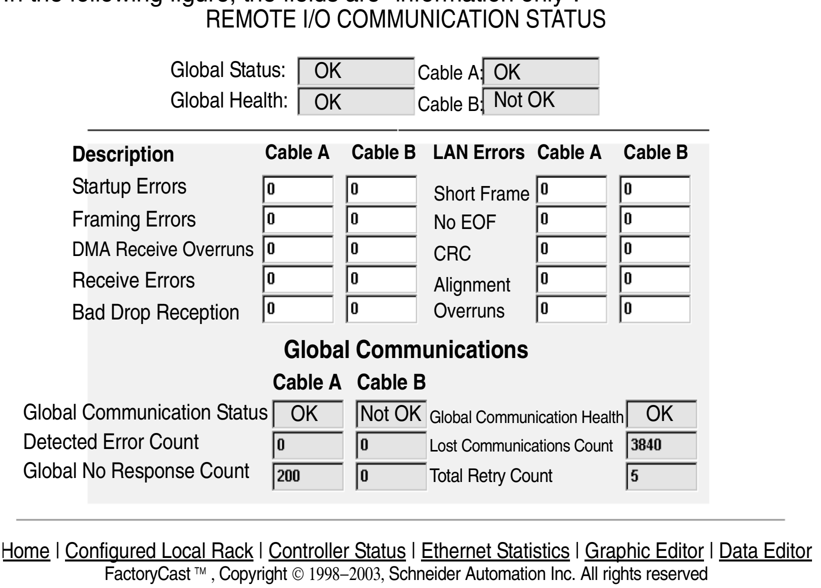
Remote I/O Communication Status Page

Overview

The following information describes the Remote I/O Communication Status Page.

Introduction to the Remote I/O Communications Status Page

In the following figure, the fields are "information only".



REMOTE I/O COMMUNICATION STATUS

Global Status:	OK	Cable A:	OK
Global Health:	OK	Cable B:	Not OK

Description	Cable A	Cable B	LAN Errors	Cable A	Cable B
Startup Errors	0	0	Short Frame	0	0
Framing Errors	0	0	No EOF	0	0
DMA Receive Overruns	0	0	CRC	0	0
Receive Errors	0	0	Alignment	0	0
Bad Drop Reception	0	0	Overruns	0	0

Global Communications

	Cable A	Cable B		
Global Communication Status	OK	Not OK	Global Communication Health	OK
Detected Error Count	0	0	Lost Communications Count	3840
Global No Response Count	200	0	Total Retry Count	5

Home | Configured Local Rack | Controller Status | Ethernet Statistics | Graphic Editor | Data Editor

FactoryCast ™ , Copyright © 1998–2003, Schneider Automation Inc. All rights reserved

For more information about the Remote I/O Communications Status, see the RIO Manual, 890 USE 101 00.

Remote I/O Communications Status Page Links

The following table describes the information that is available for each cable, using the links on the Remote I/O Communication Status Page.

Link	**Results**
Home	Displays the Quantum Welcome Page
Configured Local Rack	Displays the Quantum Local Rack with NWM and CPU
Controller Status	Displays the CPU Configuration
Graphic Editor	Displays the graphic tool for creating input/output screens
Data Editor	Allows access to the Quantum PLC Data with editing capabilities
Ethernet Statistics	Displays the Ethernet Module Statisitics with the Reset Counters link

3.6 Editors

Overview of Editors

Overview

The following section describes the Graphic Editor page and the Data Editor page.

What's in this Section?

This section contains the following topics:

Graphic Editor Overview

Top Window

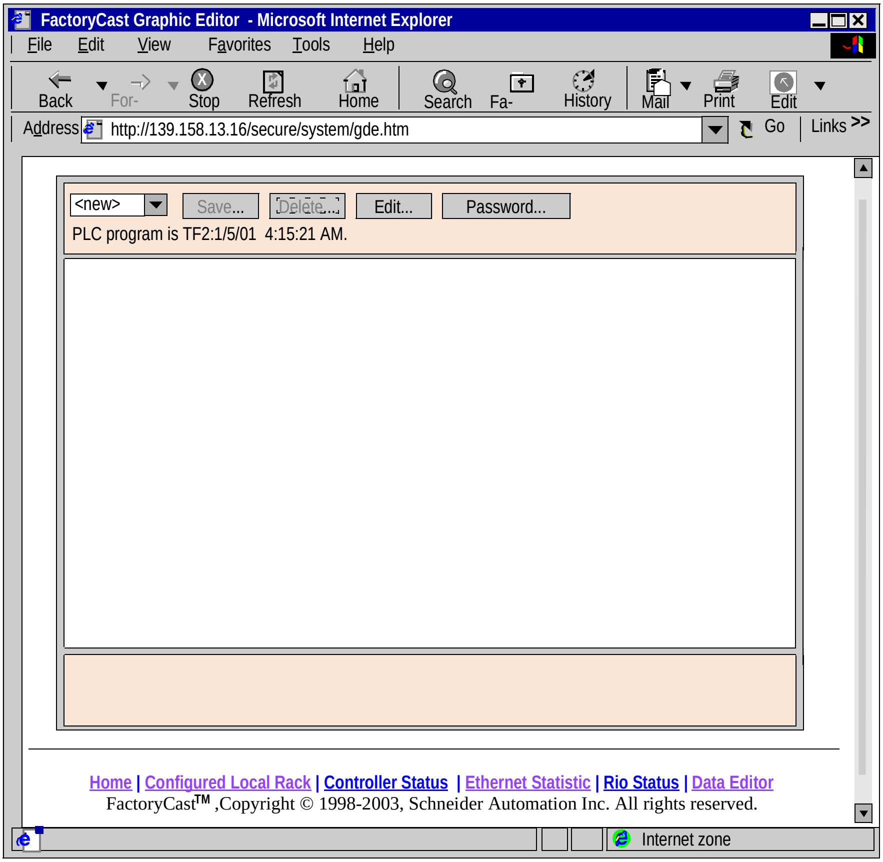
The Graphic Editor applet is divided into three windows. The Top Window provides the area for presenting all the user controls and dialogs for creating, saving, reading and editing a graphic display.

Display Window

The Display Window provides the area for presenting the current graphic display. When you create a new graphic display, this window becomes an "empty canvas", ready for you to add the graphic objects that will make up the desired graphic display.

Message Window

The Message Window provides a scrollable message window for presenting any messages that are generated by the Graphic Editor. The figure below shows the Graphic Editor applet with its initial top window and empty display and message windows.



FactoryCast Graphic Editor - Microsoft Internet Explorer

Address http://139.158.13.16/secure/system/gde.htm

<new>

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

Home | Configured Local Rack | Controller Status | Ethernet Statistic | Rio Status | Data Editor

FactoryCast^{TM} ,Copyright © 1998-2003, Schneider Automation Inc. All rights reserved.

Graphic Objects

All the graphic objects that are 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". All the graphic objects are designed as stand-alone, meaning that there are no connections required between the objects, 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 typically be interested in either viewing a graphic display (for monitoring/controlling the PLC application) or creating/modifying a graphic display. For a user who just wants to view and interact with existing graphic displays (e.g., an operator), they can select the desired graphic displays from the list that is available as soon as the applet is presented. They will need to enter a password only if they want to write data to the PLC.

Create and Modify Graphic Displays

If you want to create and modify graphic displays, click the **Edit...** button and you are presented with the customary editing capabilities for a graphic editor. That is, select objects from a palette, drop them onto a canvas, move and size them with a mouse, and set their properties. You can immediately test the modified graphic display with run-time data from the PLC by exiting editor mode (click the **Done** button). When satisfied with your creation, the graphic display can be saved to the PLC for re-use by clicking the **Save...** button, if you have entered the correct password.

User Functions

Most of the Graphic Editor's user functions are available from the top window, which is discussed in Top Window User Functions From the display window, you can directly manipulate a graphic object's size and location. All properties of a graphic object (e.g., scaling values, labels, colors, PLC addresses of the run-time data) are set by means of the Property Sheet, which is described in Property Sheet.

The Data Editor Spreadsheet

Overview

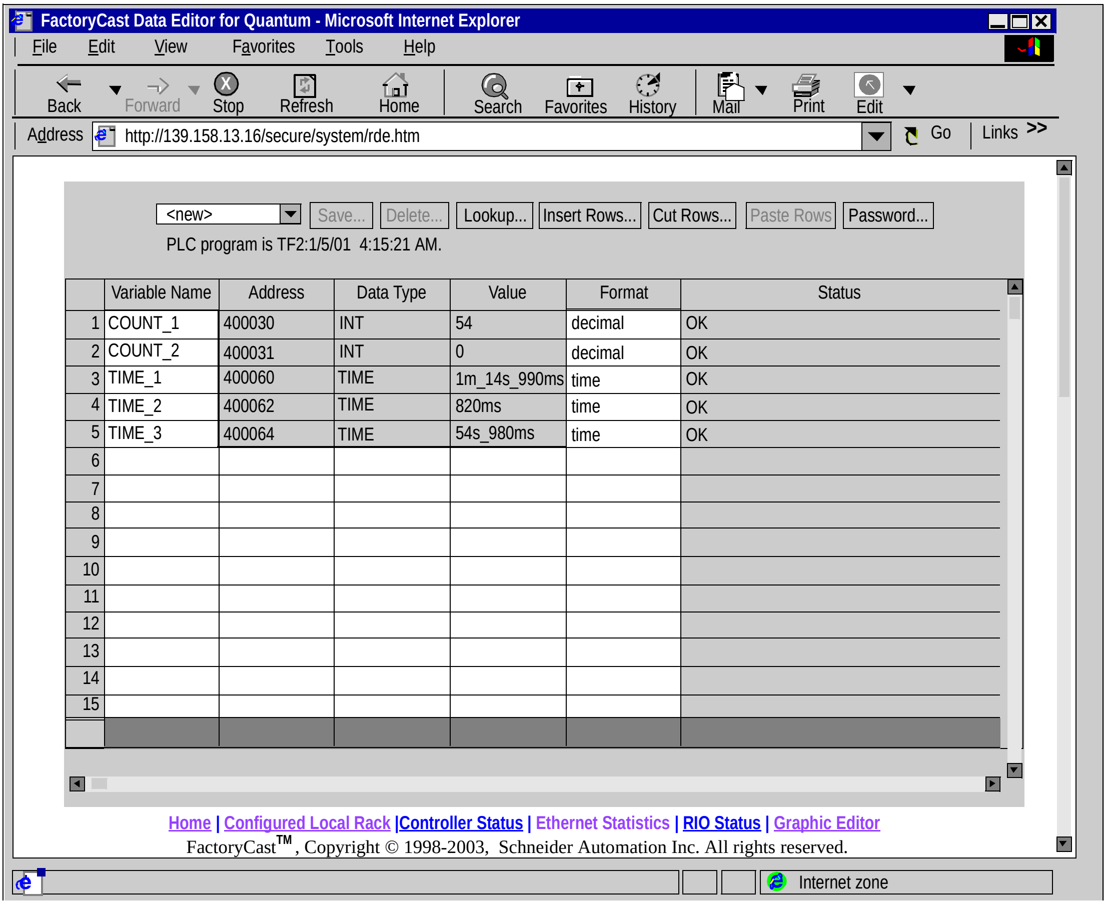
The Data Editor displays data in a spreadsheet with the following fields:

- **Variable Name**
- **Address**
- **Data Type**
- **Value**
- **Format**
- **Status**

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

Spreadsheet

Here is a Data Editor spreadsheet.



Address: http://139.158.13.16/secure/system/rde.htm

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

	Variable Name	Address	Data Type	Value	Format	Status
1	COUNT_1	400030	INT	54	decimal	OK
2	COUNT_2	400031	INT	0	decimal	OK
3	TIME_1	400060	TIME	1m_14s_990ms	time	OK
4	TIME_2	400062	TIME	820ms	time	OK
5	TIME_3	400064	TIME	54s_980ms	time	OK
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						

Home | Configured Local Rack |Controller Status | Ethernet Statistics | RIO Status | Graphic Editor

FactoryCast™, Copyright © 1998-2003, Schneider Automation Inc. All rights reserved.

Variable Name Field

The **Variable Name** column contains the names of Concept or PL7 symbols (variables).
The only symbols (variables) which may be used in the Data Editor are the ones in the namespace on the Embedded Server. If you try to enter a symbol (variable) which is not in the namespace, a "variable not found" message appears.

Note: The Data Editor can only read values from a namespace which 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 or PL7 symbols (variables). Any direct address may be viewed by entering its reference in this field. It does not have to be included in the namespace.

Valid Direct Addresses for Quantum

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

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 must be 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	16-bit unsigned integer
REAL	32-bit IEEE floating point
TIME	32-bit unsigned integer (in ms)
BOOL	1-bit discrete (boolean)

Value Field

The Value column will be 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

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 problem communicating with the symbol (variable) or direct address, the **Status** column will contain an error message describing the problem.

3.7 Configure NWM

Overview of Configure NWM

Overview

This section describes the Configure NWM Page and SNMP Page.

What's in this Section?

This section contains the following topics:

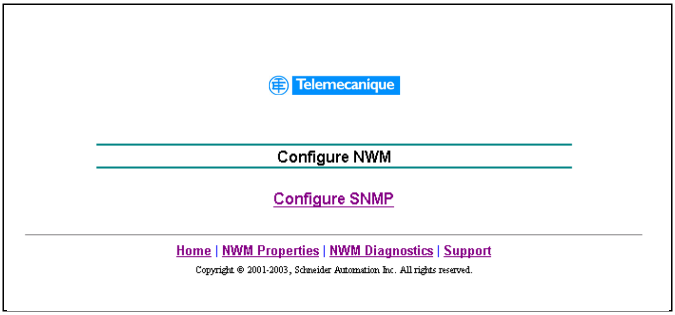
Configure NWM Page

Overview

The following information describes the **Configure NWM** page and the links on that page.

Introduction to the NWM Configuration Pages

The following figure shows the **Configure NWM** page, which provides links to the individual configuration pages for the 140 NWM 100 00.



Telemecanique

Configure NWM

Configure SNMP

Home | NWM Properties | NWM Diagnostics | Support

Copyright © 2001-2003, Schneider Automation Inc. All rights reserved.

NWM Configuration Page Links

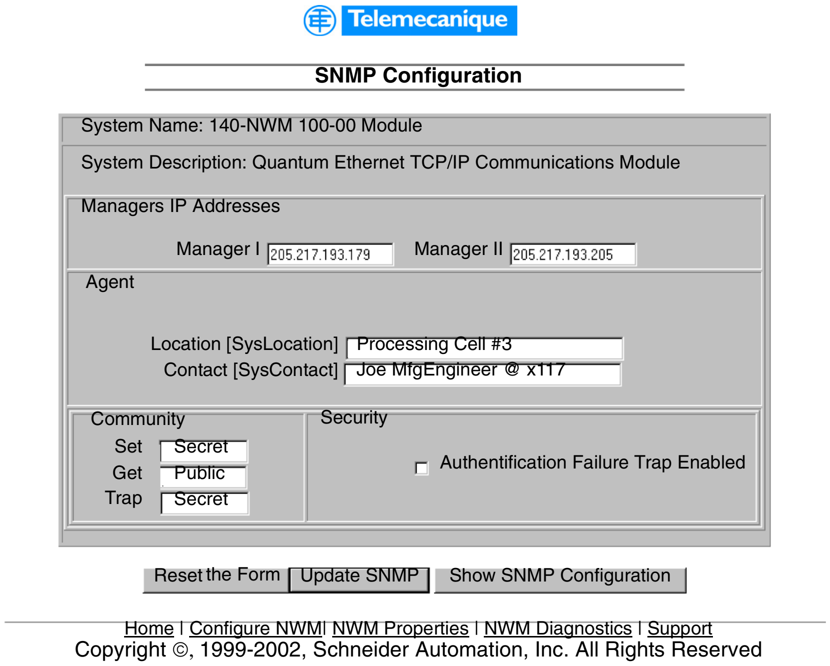
The following table describes the links on the **Configure NWM** page.

Link	Results
Home	Returns you to the Quantum Welcome Page
Configure SNMP	Provides the ability to configure the SNMP Agent in the NWM
NWM Properties	Provides information about NWM properties
NWM Diagnostics	Displays links to Ethernet Statistics and the Crash Log File Diagnostics
Support	Provides you with contact information for technical assistance, sales, and feedback

Configure SNMP Page

Introduction to the Configure SNMP Page

SNMP may already be configured on your 140 NWM 100 00. If it is not, complete the SNMP Configuration form, which is shown in the following figure.



SNMP Configuration

System Name: 140-NWM 100-00 Module

System Description: Quantum Ethernet TCP/IP Communications Module

Managers IP Addresses

Manager I: 205.217.193.179 Manager II: 205.217.193.205

Agent

Location [SysLocation]: Processing Cell #3

Contact [SysContact]: Joe MfgEngineer @ x117

Community

Set: Secret

Get: Public

Trap: Secret

Security

□ Authentification Failure Trap Enabled

Reset the Form
Update SNMP
Show SNMP Configuration
Home | Configure NWM| NWM Properties | NWM Diagnostics | Support

Copyright ©, 1999-2002, Schneider Automation, Inc. All Rights Reserved

The following table shows how to enter the required information for configuring SNMP on your 140 NWM 100 00.

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 specific 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 level of user who can set the configuration
Get	Designation of level of user who can view the configuration
Trap	Designation of level of user who can capture information
Authentication Failure Trap Enabled	Turns on Community Name Checking

Configure SNMP Page Links

The following table describes the links on the Configure SNMP Page.

Link	Results
Home	Returns you to the Quantum Welcome Page.
Configure NWM	Provides the ability to configure and change the NWM through the Ethernet Configuration Page.
NWM Properties	Provides information about NWM properties.
NWM Diagnostics	Displays links to Ethernet Statistics and the Crash Log File Diagnostics.
Support	Provides you with contact information for technical assistance, sales, and feedback.

Completion Message

Clicking on the **Update SNMP** button displays a new page containing the message "Successfully updated SNMP database." Note that this page contains the same links as those on the Configure SNMP Page.

Note: The 140 NWM 100 00 module has to be reset for the changes to take effect.

3.8 NWM Properties

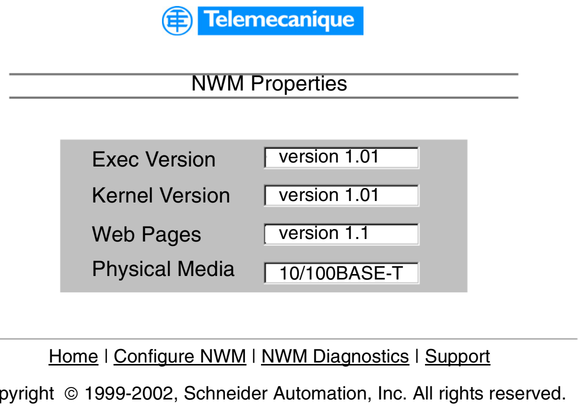
NWM Properties Page

Overview

The following information describes the 140 NWM 100 00 Properties Page.

Introduction to the NWM Properties Page

The following figure shows the NWM Properties Page, which displays the Exec, Kernel, Web Pages versions, and the Physical Media being used.



NWM Properties

Exec Version	version 1.01
Kernel Version	version 1.01
Web Pages	version 1.1
Physical Media	10/100BASE-T

Copyright © 1999-2002, Schneider Automation, Inc. All rights reserved.

Note: The NWM Properties Page is "information only." The fields cannot be changed.

NWM Properties Page Links

The following table describes the links on the NWM Properties Page.

Link	Results
Home	Returns you to the Quantum Welcome Page.
Configure NWM	Provides the ability to configure and change the NWM through the Ethernet Configuration Page.
NWM Diagnostics	Displays links to Ethernet Statistics and the Crash Log File Diagnostics.
Support	Provides you with contact information for technical assistance, sales, and feedback.

3.9 NWM Diagnostics

Overview NWM Diagnostics

Overview

This section provides information on the Diagnostics Page and the Crash Log Diagnostics.

What's in this Section?

This section contains the following topics:

Ethernet Module Diagnostic Pages

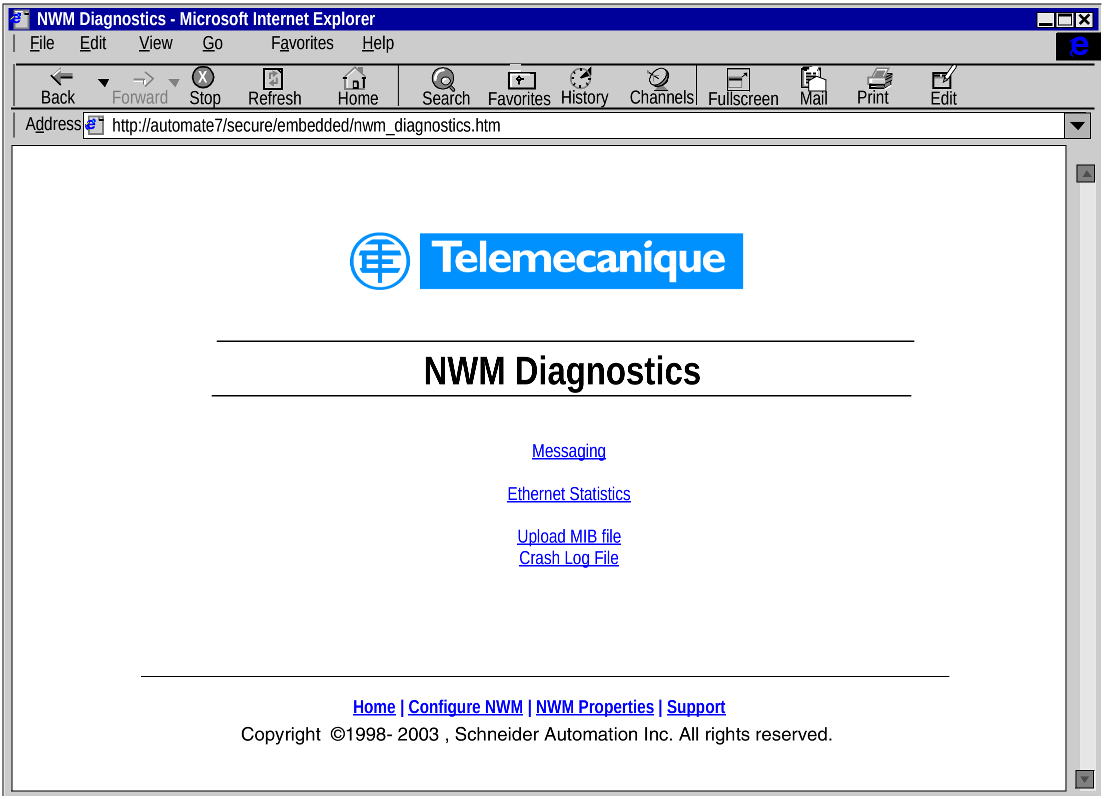
Home Page

This page contains a list of links for accessing the different diagnostic pages for the Ethernet module:

- Messaging utility
- Ethernet module statics

A link also allows the uploading of the private MIB source file.

View of the Diagnostics Utilities home page:



NWM Diagnostics

Messaging

Ethernet Statistics

Upload MIB file
Crash Log File

Home | Configure NWM | NWM Properties | Support

Copyright ©1998- 2003 , Schneider Automation Inc. All rights reserved.

Click on a link to access the desired diagnostics page.

Link	Results
Messaging	Displays the number of messages sent and received

Link	Results
Ethernet Statistics	Displays the Ethernet Statistics Page where you can display the Ethernet statistics and reset the counters
Crash Log File	Displays the Crash Log enteries for use in diagnosing the cause of crash
Home	Returns you to the Quantum Welcome Page
Configure NWM	Provides the ability to configure and change the NWM.
NWM Properties	Provides information about NWM properties
Support	Provides you with contact information for technical assistance, sales and feedback

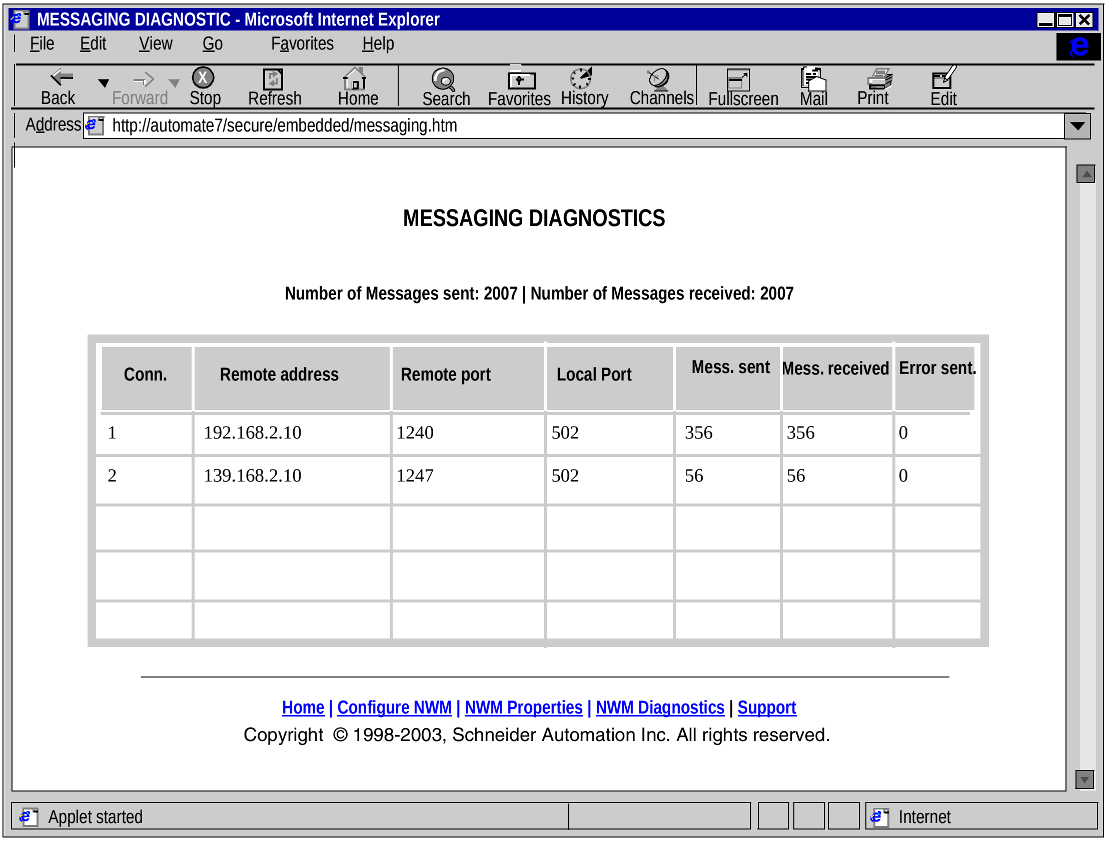
Messaging Page

This page provides current information on the open TCP connection 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 connections (numbered from 1 to 64):

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

View of the messaging diagnostics page:



http://automate7/secure/embedded/messaging.htm

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	192.168.2.10	1240	502	356	356	0
2	139.168.2.10	1247	502	56	56	0

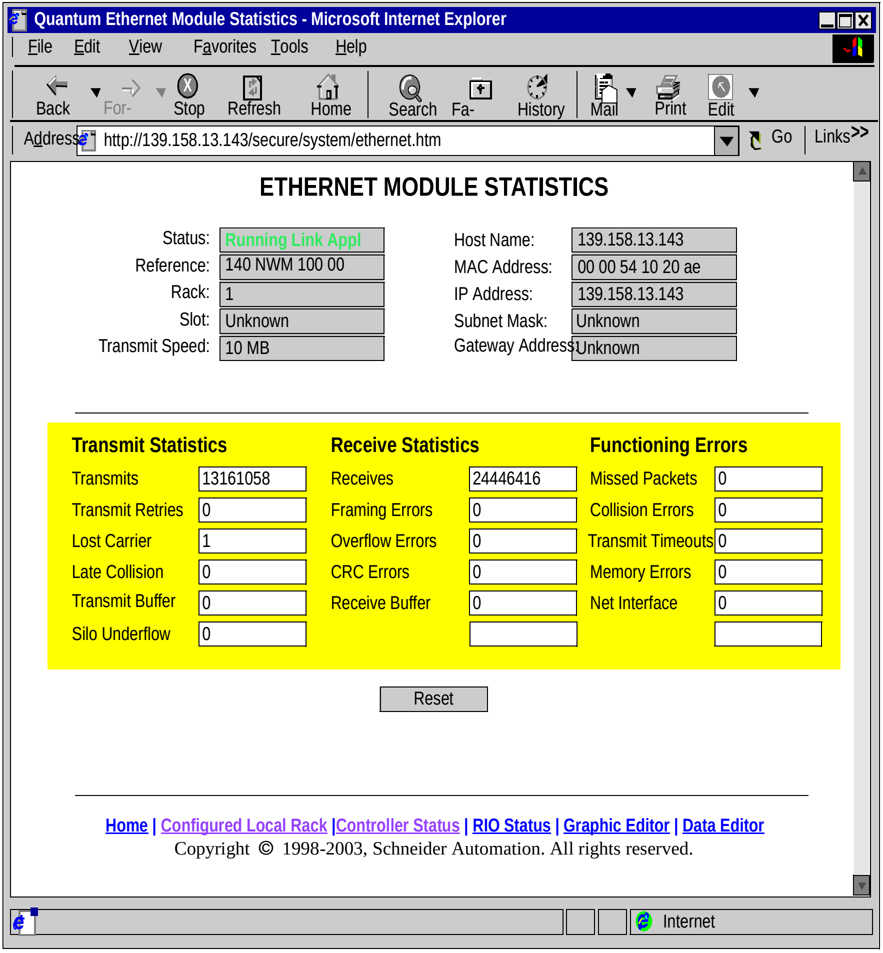
Home | Configure NWM | NWM Properties | NWM Diagnostics | Support

Copyright © 1998-2003, Schneider Automation Inc. All rights reserved.

Ethernet Statistics Page

The Ethernet Module Statistics page provides information about the status, transmit and receive statistics, and errors for the Embedded Server module. Access this page by selecting the NWM module from the local rack or use the hyperlink at the bottom of the page.

Here is an example of an Ethernet Module Statistics page.


Quantum Ethernet Module Statistics - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back For- Stop Refresh Home Search Fa- History Mail Print Edit
Address
Go
Links
Internet

http://139.158.13.143/secure/system/ethernet.htm

ETHERNET MODULE STATISTICS

Status:	Running Link Appl	Host Name:	139.158.13.143
Reference:	140 NWM 100 00	MAC Address:	00 00 54 10 20 ae
Rack:	1	IP Address:	139.158.13.143
Slot:	Unknown	Subnet Mask:	Unknown
Transmit Speed:	10 MB	Gateway Address	Unknown

Transmit Statistics		Receive Statistics		Functioning Errors	
Transmits	13161058	Receives	24446416	Missed Packets	0
Transmit Retries	0	Framing Errors	0	Collision Errors	0
Lost Carrier	1	Overflow Errors	0	Transmit Timeouts	0
Late Collision	0	CRC Errors	0	Memory Errors	0
Transmit Buffer	0	Receive Buffer	0	Net Interface	0
Silo Underflow	0				

Reset

Home | Configured Local Rack |Controller Status | RIO Status | Graphic Editor | Data Editor

Copyright © 1998-2003, Schneider Automation. All rights reserved.

Note: Refer to *Modicon Quantum Ethernet TCP/IP Module User Guide* 840 USE 107 00 and *Modicon Quantum Ethernet TCP/IP Module User Guide 840* USE 115 00 for definitions of terms.

Crash Log Diagnostics

Introduction to the Crash Log Diagnostics Page

When you click on Crash Log File in the NWM Diagnostics page, the Crash Log Diagnostics page appears. The Crash Log Diagnostics page displays a crash log file when a crash has occurred, and a status message when no crash has occurred. Press **Clear Crash Log File** to clear the file.

Crash Log Diagnostics Links

The following table describes the links on the Crash Log Diagnostics Page.

Link	Results
Home	Returns you to the Quantum Welcome Page
Configure NWM	Provides the ability to configure and change the NWM through the Ethernet Configuration Page
NWM Properties	Provides information about NWM properties
NWM Diagnostics	Displays links to Ethernet Statistics and the Crash Log File Diagnostics
Support	Provides you with contact information for technical assistance, sales, and feedback

Default Premium Web site

4

At a Glance

Subject of this Chapter

When you receive the TSX WMY 100 FactoryCast HMI module, it already contains a Web site by default.

To display these pages, simply install the module and configure its IP address. To access the site, enter the IP address of the module in your browser and the default username along with the "USER" password.

This chapter describes the pages of the default Web site.

What's in this Chapter?

This chapter contains the following sections:

4.1 Premium Home Pages

At a Glance

Subject of this Section

When you connect to the default Premium Web site, the site home page is displayed. This section provides a description of these pages.

What's in this Section?

This section contains the following topics:

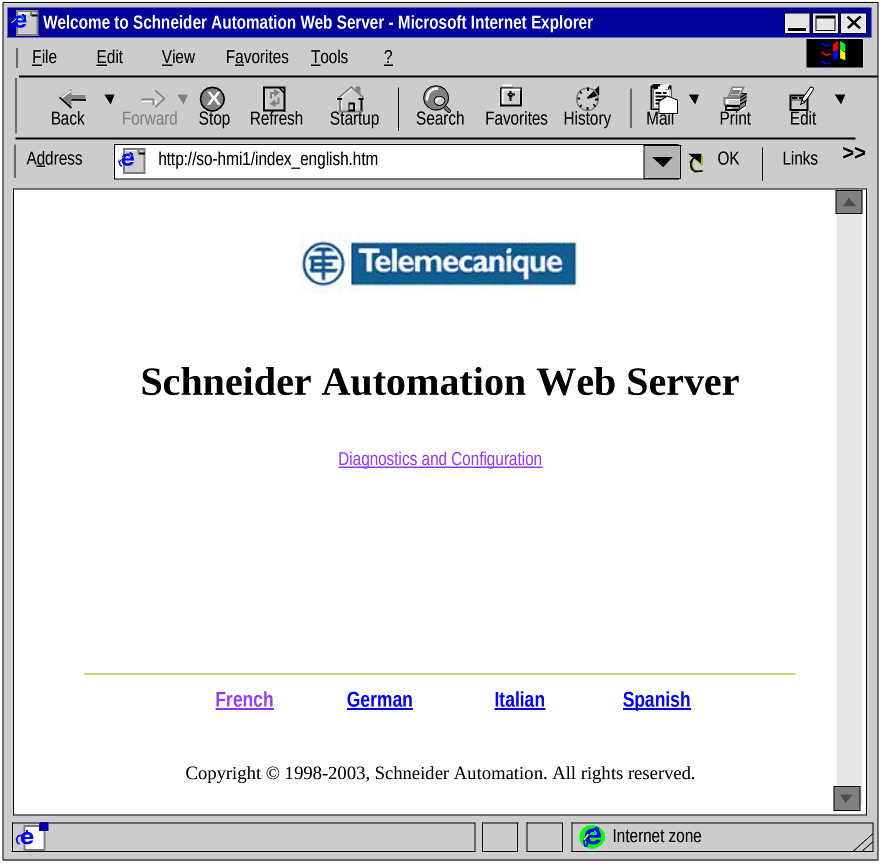
FactoryCast Home Page

Overview

The FactoryCast home page is the first page a visitor will encounter. No password is required to access this page.

Home Page

This is how the FactoryCast Home Page looks.


Welcome to Schneider Automation Web Server - Microsoft Internet Explorer
File Edit View Favorites Tools ?
Back Forward Stop Refresh Startup Search Favorites History Mail Print Edit
Address
http://so-hmi1/index_english.htm
OK
Links
>>
Telemecanique

Schneider Automation Web Server

Diagnostics and Configuration

French German Italian Spanish

Copyright © 1998-2003, Schneider Automation. All rights reserved.

Links

The home page offers a link to.

- `Diagnostics and Configuration`: which points to the various embedded Premium options available on the site. These options are password-secured.
- Foreign Language links exist for French, German, Italian, and Spanish versions.

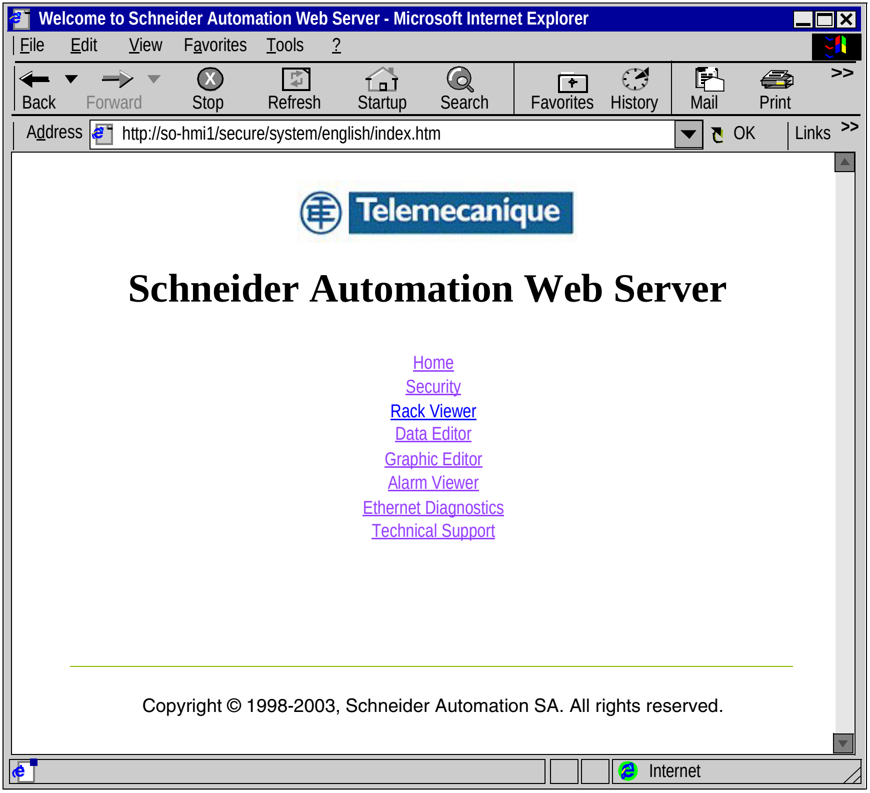
Premium Home Page

Overview

When a visitor clicks on the Diagnostics and Configuration link on the FactoryCast home page, this gives direct access to the Premium home page. The visitor must enter a username and a password to display the page.

Home Page

This is how the Premium Home Page looks.



http://so-hmi1/secure/system/english/index.htm

Telemecanique

Schneider Automation Web Server

Home
Security
Rack Viewer
Data Editor
Graphic Editor
Alarm Viewer
Ethernet Diagnostics
Technical Support

Copyright © 1998-2003, Schneider Automation SA. All rights reserved.

Links

The Premium home page offers links to.

- Home,
- Security,
- Rack Viewer,
- Data Editor,
- Graphic Editor,
- Alarm Viewer,
- Ethernet Diagnostics,
- Technical Support.

4.2 Rack Viewer

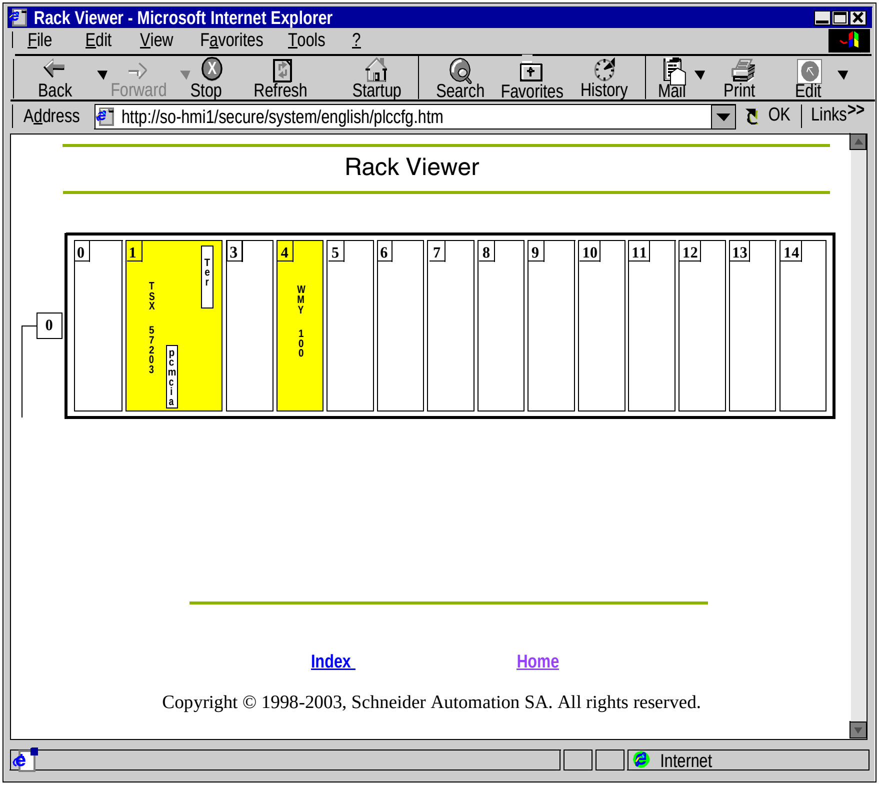
Rack Viewer Page

Overview

The Rack Viewer page shows the current configuration of the (local or remote) racks, in particular the PLC, the built-in server module and the I/O modules.

Example

Here is an example:



Rack Viewer - Microsoft Internet Explorer

Address: http://so-hmi1/secure/system/english/plccfg.htm

Rack Viewer

Copyright © 1998-2003, Schneider Automation SA. All rights reserved.

Data

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

- A vertical label indicating the type of module and the product reference
- The box in the upper lefthand corner of the module indicates the slot number and operating state of the module:
  - A yellow box indicates that the module is healthy
  - A red box indicates that the module is not operating correctly

Links

Click on an element in the rack to obtain more detailed information (processor, module, Ter port, etc.).

A line descending from Rack#0 (to the left of the rack) is a link to the next rack. When you roll over this link, it changes into a red arrow. Each rack in the configuration can be displayed. The links at the top of each slot will take you back to Rack#0.

4.3 PLC Diagnostics

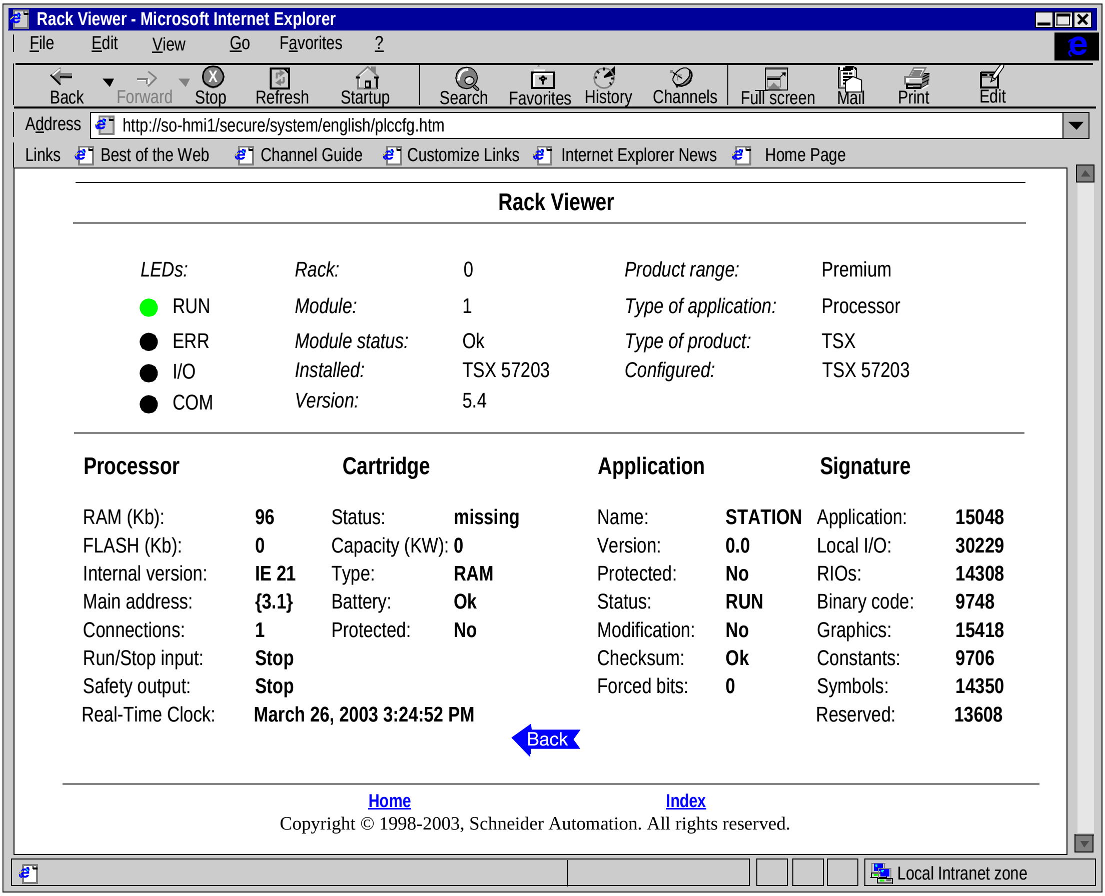
PLC Characteristics Page

Overview

The PLC characteristics page provides information on the PLC and its configuration.

Example

Here is an example of the PLC characteristics which corresponds to the rack configured above (See *Rack Viewer Page, p. 92*).



Address http://so-hmi1/secure/system/english/plccfg.htm

Rack Viewer

LEDs:	*Rack:*	0	*Product range:*	Premium
🟢 RUN	*Module:*	1	*Type of application:*	Processor
● ERR	*Module status:*	Ok	*Type of product:*	TSX
● I/O	*Installed:*	TSX 57203	*Configured:*	TSX 57203
● COM	*Version:*	5.4		

Processor		Cartridge		Application		Signature	
RAM (Kb):	**96**	Status:	**missing**	Name:	**STATION**	Application:	**15048**
FLASH (Kb):	**0**	Capacity (KW):	**0**	Version:	**0.0**	Local I/O:	**30229**
Internal version:	**IE 21**	Type:	**RAM**	Protected:	**No**	RIOs:	**14308**
Main address:	**{3.1}**	Battery:	**Ok**	Status:	**RUN**	Binary code:	**9748**
Connections:	**1**	Protected:	**No**	Modification:	**No**	Graphics:	**15418**
Run/Stop input:	**Stop**			Checksum:	**Ok**	Constants:	**9706**
Safety output:	**Stop**			Forced bits:	**0**	Symbols:	**14350**
Real-Time Clock:	**March 26, 2003 3:24:52 PM**					Reserved:	**13608**

Back

Home Index

Copyright © 1998-2003, Schneider Automation. All rights reserved.

Dynamic Data

The indicators at the top, and to the left of the screen provide dynamic references to the state of the PLC.

LEDs	Color when On	Meaning when On	Meaning when flashing	Meaning when Off
RUN	green	Application is running	Stopped	PLC error
ERR	Red	PLC error	Not configured	No error
I/O	Red	I/O event	---	No error
COM	Yellow	Communication error	---	No error

Links

The `Back` arrow returns you to the Rack Viewer page for this PLC. The links at the bottom of the PLC characteristics page can be used to go back to the FactoryCast or Premium home page.

4.4 FactoryCast HMI Module Diagnostics

Ethernet Module Statistics Page

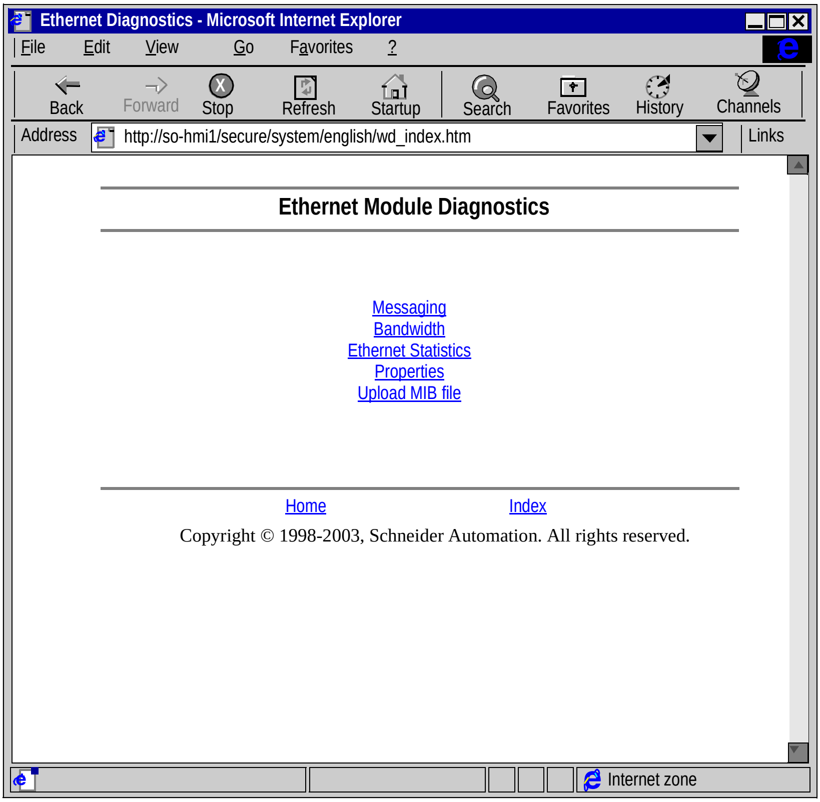
Home Page

This page contains a list of links for accessing the different diagnostic pages for the Ethernet module:

- Messaging service,
- Bandwidth service,
- Ethernet Statistics service,
- Properties service.

A link also allows the downloading of the private MIB source file.

View of the Diagnostic Services home page:



Ethernet Module Diagnostics

Messaging
Bandwidth
Ethernet Statistics
Properties
Upload MIB file

Home Index

Copyright © 1998-2003, Schneider Automation. All rights reserved.

Click on a link to access the desired diagnostics page.

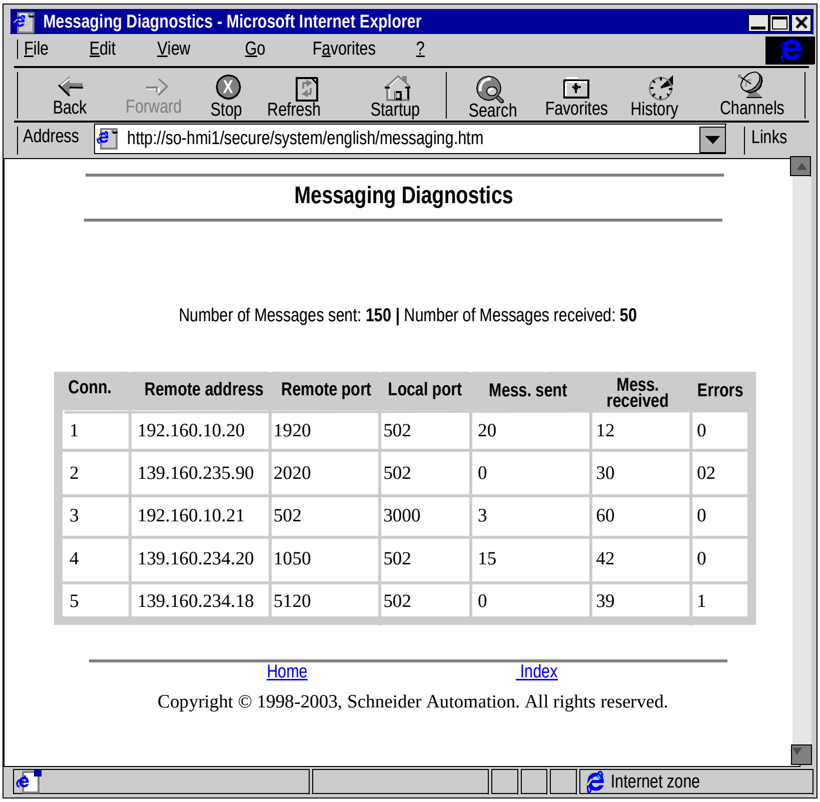
Messaging Diagnostics Page

This page provides current information on the open TCP connection on port 502. The number of messages sent/received 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
- The number of messages sent from this connection
- The number of messages received from this connection
- The error number on this connection.

View of the messaging diagnostics page:



http://so-hmi1/secure/system/english/messaging.htm

Messaging Diagnostics

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

Conn.	Remote address	Remote port	Local port	Mess. sent	Mess. received	Errors
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	3000	3	60	0
4	139.160.234.20	1050	502	15	42	0
5	139.160.234.18	5120	502	0	39	1

Home Index

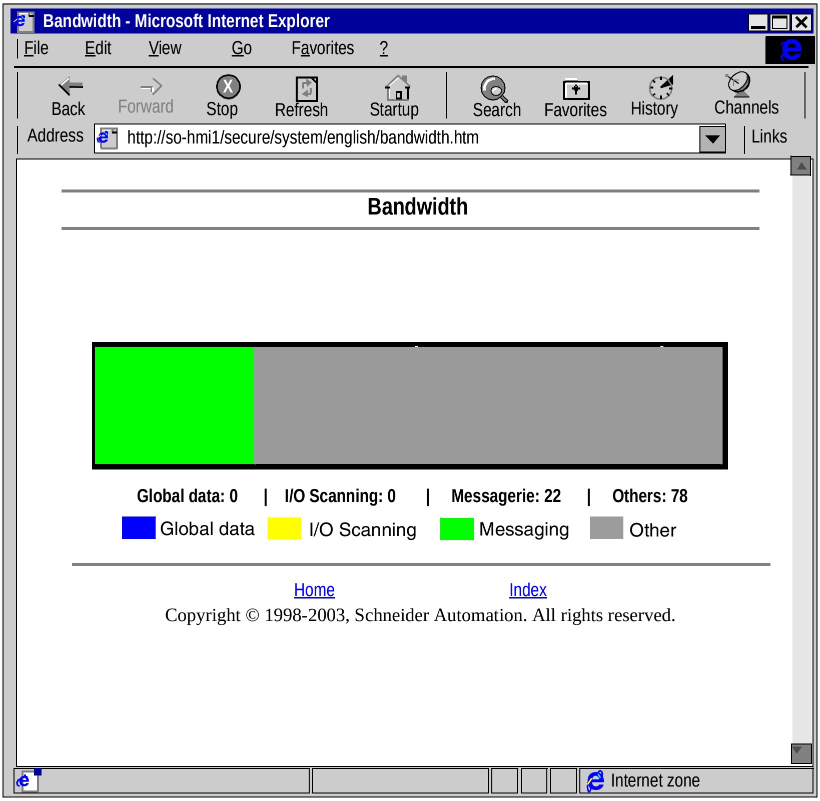
Copyright © 1998-2003, Schneider Automation. All rights reserved.

Bandwidth Page

This page displays the load distribution of the TSX WMY 100 module between the Global Data services, I/O Scanning, Messaging and other services.

Note: As Global data and I/O Scanning services are not supported on this module, they will always be set to 0.

View of the Bandwidth monitoring page:



Address: http://so-hmi1/secure/system/english/bandwidth.htm

Bandwidth

Global data: 0 | I/O Scanning: 0 | Messagerie: 22 | Others: 78

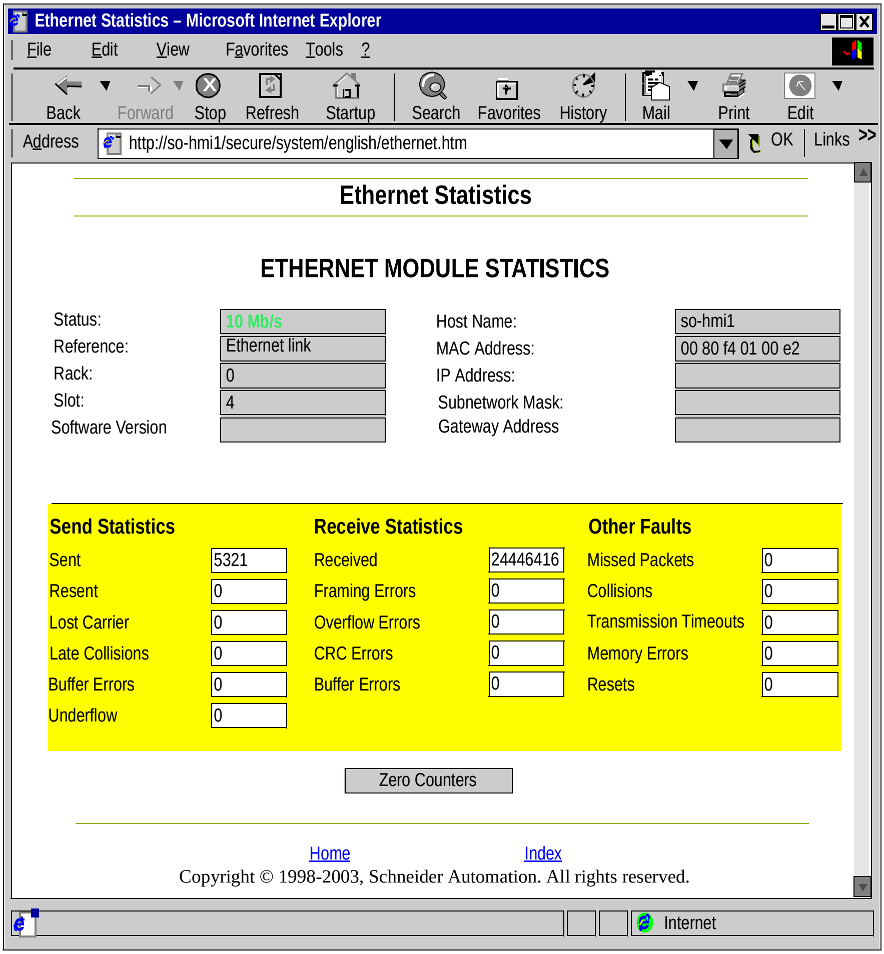
🟦 Global data 🟨 I/O Scanning 🟩 Messaging ▨ Other

Home Index

Copyright © 1998-2003, Schneider Automation. All rights reserved.

Statistics Page

Click on the embedded server module in the Rack Viewer to display the Statistics page for the Ethernet module. This page provides all the latest information on the state, configuration and functions of the embedded server module.
The Statistics Page for the Ethernet module is shown below.


Ethernet Statistics – Microsoft Internet Explorer
File Edit View Favorites Tools ?
Back
Forward
Stop
Refresh
Startup
Search
Favorites
History
Mail
Print
Edit
Address
http://so-hmi1/secure/system/english/ethernet.htm
OK
Links >>
Internet

Ethernet Statistics

ETHERNET MODULE STATISTICS

Status:	10 Mb/s	Host Name:	so-hmi1
Reference:	Ethernet link	MAC Address:	00 80 f4 01 00 e2
Rack:	0	IP Address:	
Slot:	4	Subnetwork Mask:	
Software Version		Gateway Address	

Send Statistics		Receive Statistics		Other Faults	
Sent	5321	Received	24446416	Missed Packets	0
Resent	0	Framing Errors	0	Collisions	0
Lost Carrier	0	Overflow Errors	0	Transmission Timeouts	0
Late Collisions	0	CRC Errors	0	Memory Errors	0
Buffer Errors	0	Buffer Errors	0	Resets	0
Underflow	0				

Zero Counters

Home Index

Copyright © 1998-2003, Schneider Automation. All rights reserved.

Dynamic Data

The indicators at the top, and to the left of the screen provide dynamic references to the state of the embedded server module.

LEDs	Color	LED on	LED flashing	LED off
RUN	Green	Operating normally	---	Power off
ERR	Red	Module fault	Not configured	Operating normally
STS	Red	Network address incorrect or station outside limits	---	No error

Links

The links at the bottom of the PLC `Ethernet Statistics` page can be used to go back to the FactoryCast or Premium home page.

Data and Graphic Editors

5

At a Glance

Subject of this Chapter

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's in this Chapter?

This chapter contains the following sections:

5.1 Data Editor

At a Glance

Subject of this Section

The standard Web page by default contains a link to the Data Editor. The Data editor is used to create dynamic data tables of PLC variables.
This section describes how to use the Data editor to display and modify the values of the symbol variables and direct addresses.

What's in this Section?

This section contains the following topics:

The Data Editor Spreadsheet

Overview

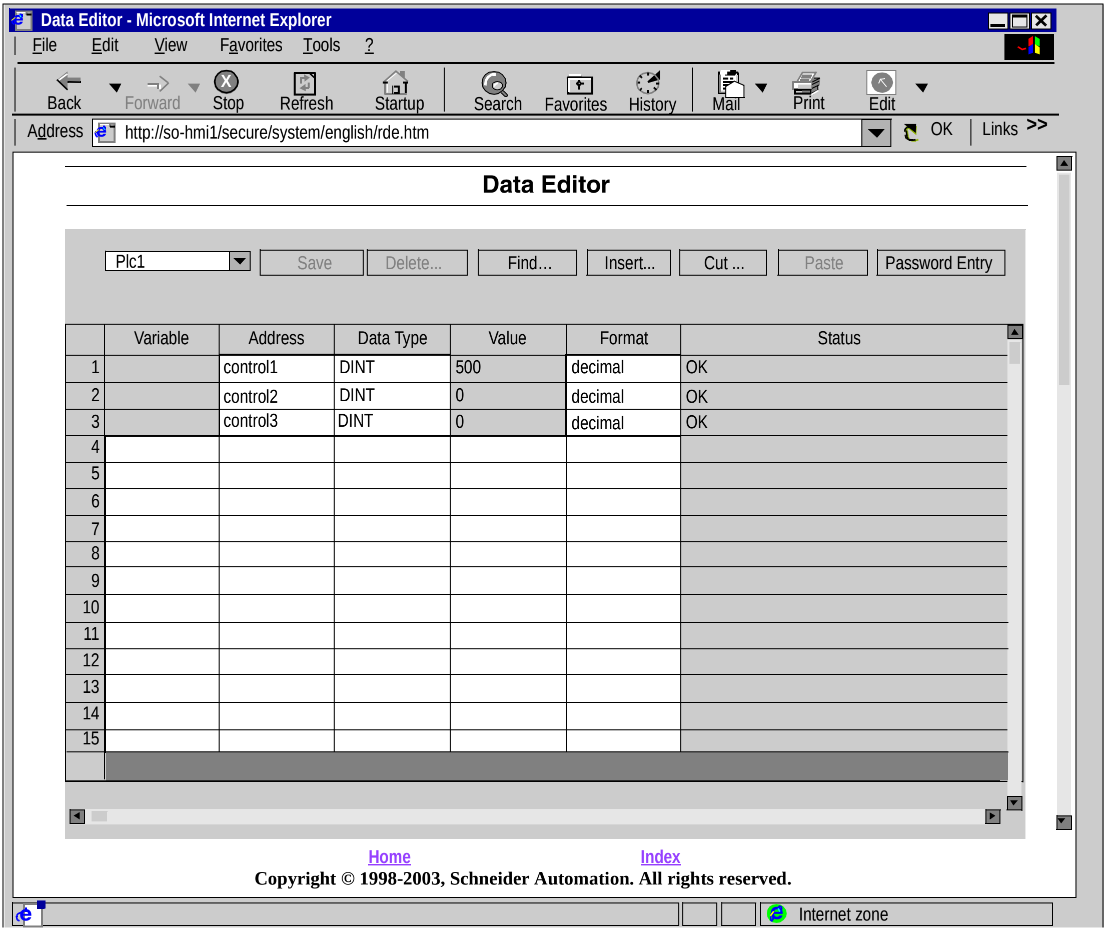
The Data Editor displays data in a spreadsheet with the following fields:

- **Variable**
- **Address**
- **Data Type**
- **Value**
- **Format**
- **State**

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

Spreadsheet

Here is a Data Editor spreadsheet.



Address: http://so-hmi1/secure/system/english/rde.htm

Data Editor

Plc1

	Variable	Address	Data Type	Value	Format	Status
1		control1	DINT	500	decimal	OK
2		control2	DINT	0	decimal	OK
3		control3	DINT	0	decimal	OK
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						

Variable Name Field

The **Variable** column contains the names of Concept, PL7 or Unity Pro symbol variables.
The only variable symbols which may be used in the Data Editor are the probably defined by the configuration tool. The symbol variables are grouped in a file called "Namespace". If you try to enter a symbol variable which is not in the namespace, a "variable not found" message appears.

Note: The Namespace must have been created using an identical program to the one operating in the PLC. 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 symbol variables. Any direct address may be viewed by entering its reference in this field. This direct address does not need to be referenced in the name space.

Valid Direct Addresses for Quantum

- Coils (0x)
- Discrete 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 discrete 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" (e.g.%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 Micro

Here are the valid direct addresses for Micro.

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
%Is.c	BOOLEAN	R
%Qs.c	BOOLEAN	R/W
%Mi	BOOLEAN	R/W
%Si	BOOLEAN	R/W
%MFi	REAL 32	R/W
%IWs.c.i	WORD 16	R
%MWi	WORD 16	R/W
%SWi	WORD 16	R/W
%QWs.c.i	WORD 16	R/W
%MWs.c.i	WORD 16	R/W
%MWs.MOD.i	WORD 16	R/W
%KWs.c.i	WORD 16	R/W
s=slot number, c=channel number, and i=range number		

Valid Direct Addresses for Premium

Here are 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		

Here are 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		

Here are the valid FIP I/O 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		

Here are the valid FIP I/O 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

Address	Type	R(ead) or W(rite) Access
%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" (e.g. %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 must be 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	16-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 internal bit (boolean)

Value Field

The Value column will be 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 problem communicating with the symbol variable or direct address, the **Status** column will contain an error message describing the problem.

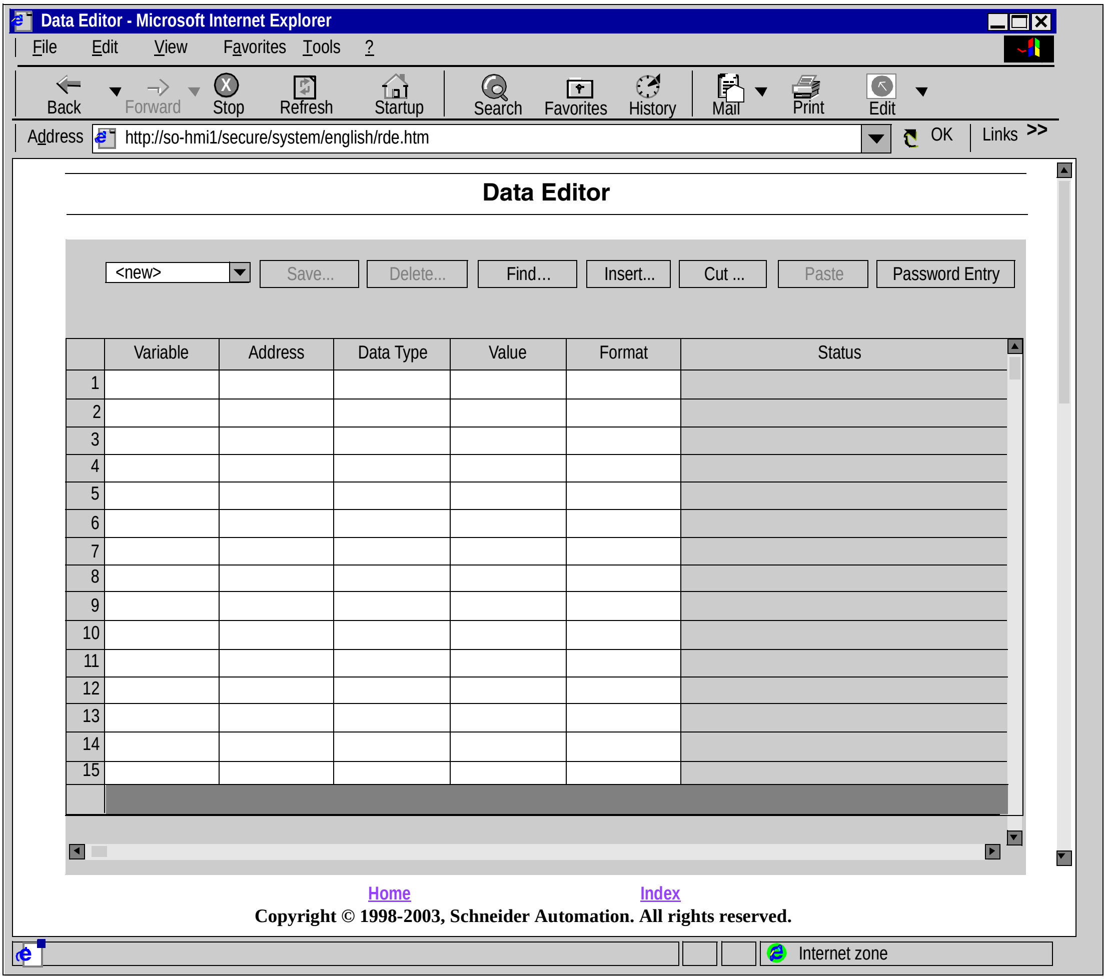
Creating a data table

Overview

When you want to view the values of symbol variables or direct addresses, you must enter the name of the symbol variable or direct address in a Data Editor spreadsheet. The spreadsheet containing your data is called a data table.

To obtain a new spreadsheet

An empty spreadsheet appears when you open the Data Editor.
If you have already worked with a spreadsheet in Data Editor and you want to obtain a new one, select **<new>** from the drop-down menu in the top left corner of the applet. This is the browser window containing an empty spreadsheet.



	Variable	Address	Data Type	Value	Format	Status
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						

Copyright © 1998-2003, Schneider Automation. All rights reserved.

Note: Save the current spreadsheet before selecting a new one.
If you select a new spreadsheet, it will overwrite the current spreadsheet.

Inserting Symbol Variables in a Data Table

Overview

If you want to display or modify the value of a symbol variable in the namespace, you must insert the variable in a Data Table.

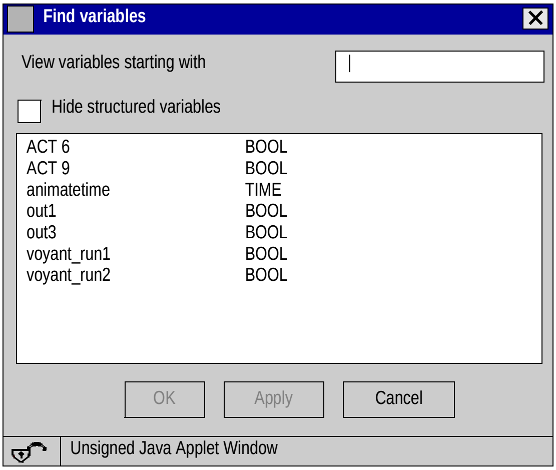
Inserting Variables

There are two possible methods for inserting a symbol variable in a Data Table.

- Enter the name of the symbol variable in the **Variable** cell of an empty line in the spreadsheet and press ENTER.
- Use the **Find variables** dialog box.

Use the Consult variables dialog box

How to use the Consult variables dialog box.

Step	Action
1	Click **Find** at the top of the spreadsheet. **Result:** The **Find Consult variables** dialog box appears.  Find variables View variables starting with □ Hide structured variables ACT 6 BOOL ACT 9 BOOL animatetime TIME out1 BOOL out3 BOOL voyant_run1 BOOL voyant_run2 BOOL
2	Select the symbol variables you want to insert in the Data Table.
3	Click **OK**. **Result:** The symbol variables you have selected appear in the table.

Inserting Direct Addresses in a Table

Overview

If you want to display or modify the value of a direct address, you must insert the address in a Data Table.

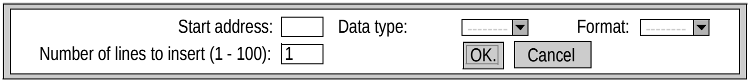
Inserting Direct Addresses

There are two possible methods for inserting one or more direct addresses in a Data Table.

- Enter the name of the address in the **Address** cell of an empty line in the spreadsheet and press ENTER.
- Use the **Insert Lines** dialog box.

Use the Insert Lines dialog box.

To use the **Insert Lines** dialog box, follow the steps in the table below.

Step	Action
1	Click **Insert Lines** at the top of the spreadsheet. **Result:** The **Insert Lines** dialog box appears.  Number of lines to insert (1 - 100): 1
2	In the **Start address** field, enter the address and insert the required number of lines in the **Number of lines to insert** field. In the Data Type field, select the type Select. A block of lines starting with the start address is inserted. **Result:** The data type and a default format appear automatically.
3	You can adjust the **Data Type** and **Format** options by making selections from the drop-down menus.
4	Click **OK**. **Result:** The direct addresses you have specified appear in the table.

Modifying Data Values

Overview

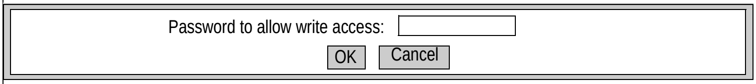
You can use the Data Editor to read and modify the values of symbol variables or direct addresses.

Modifying Data: Restrictions

Variables can only be modified if they have been declared write accessible by the HMI FactoryCast configurator.

Modifying Data

The data modification procedure involves the following steps.

Step	Action
1	Display the symbol variable or direct address in a data table.
2	Click **Password** in the Data Editor menu. **Result:** The **Password** dialog box appears.  Password to allow write access:
3	Enter the password to obtain write access.
4	Click **OK**.
5	Change the value of the symbol variable or direct address in the Value field of the table.
6	Press ENTER. **Result:** The new value is sent to the PLC.

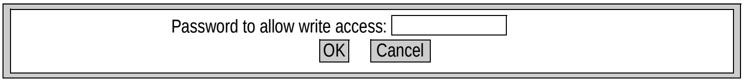
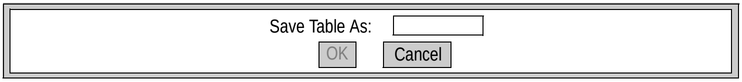
Saving a Data Table

Overview

It is possible to save a set of variables (direct or symbol) that have to be viewed and modified in a data table.

Saving a Data Table

To save a data table, follow the steps in the table below.

Step	Action
1	Insert the required direct or symbol variables in the table.
2	Click **Password** in the **Data Editor** menu. Result: The **Password** dialog box appears. 
3	Enter the **Write password**.
4	Click **OK**. **Result:** The **Save** button is activated in the **Data Editor** menu.
5	Click on the **Save** button. Result: The **Save Table As:** dialog box appears.  In the text field, enter a name for the data table. A table name must comprise a maximum of eight characters. Role Names are case sensitive. They can include upper and lower case letters, figures, underscores, hyphens and dollar signs.
6	Click **OK**.

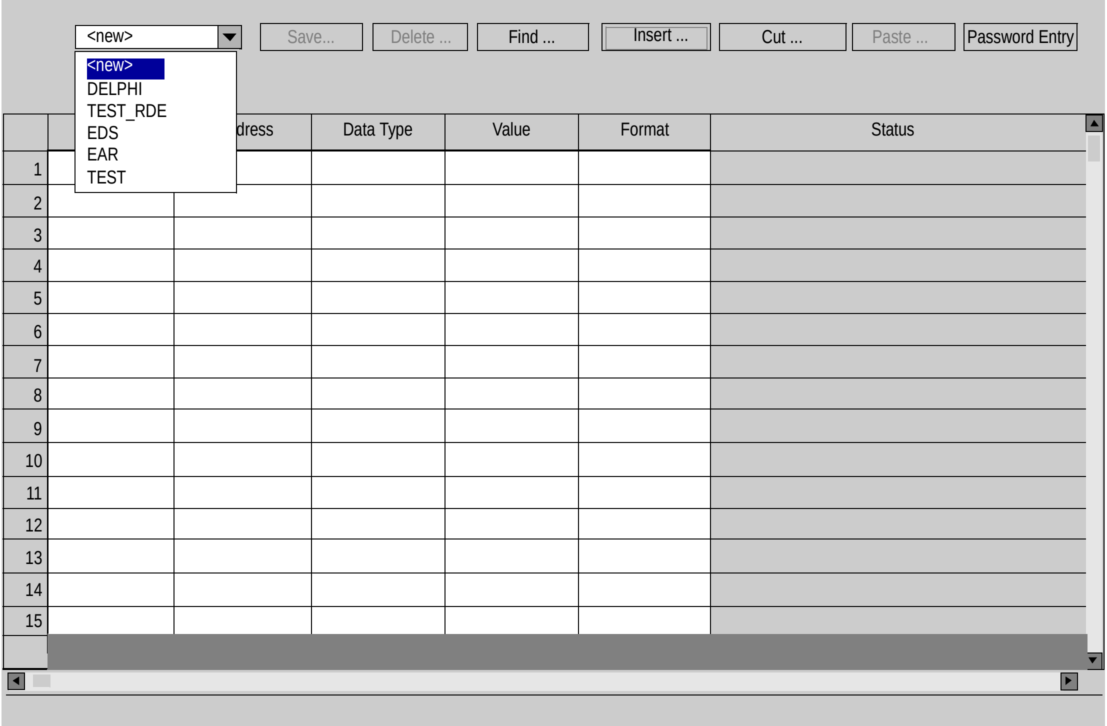
Using an Existing Data Table

Overview

Once you have saved a Data Table, you can use it to display or modify the values of symbol variables or direct addresses.

Finding a Data Table

A drop down menu from the Data Editor displays all the Data Tables that were saved.



		dress	Data Type	Value	Format	Status
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						

Retrieving a Data Table

Select the data table of your choice from the drop-down menu. The table is then automatically displayed on the screen.

Modifying a Data Table

If you enter the **Write Password**, you can modify and save a new version of the data table.

Parameters of the Data Editor Applet

Overview

The Data Editor supports two applet parameters for customizing its behavior. The applet parameters are indicated by the <PARAM> labels in the <APPLET> label of the Data Editor HTML page. The following parameters are recognized by the Data Editor applet

- **TEMPLATE**—This parameter informs the Data Editor that it must automatically load a specific table file on startup. If the specified file does not exist, an error message is delivered. If the parameter does not appear in the <APPLET> label, no table file is automatically loaded on startup. In this case you must select the initial table file from the list provided by the Data Editor.
- **AUTO-LOGIN**—This parameter informs the Data Editor that it must automatically enter the password providing write access to the PLC. If this parameter is set to **TRUE**, the Data Editor authorizes write access to the PLC without asking the user to enter a password. The possible values for this parameter are **FALSE** (by default) and **TRUE**.

Example

The following example shows an applet invocation marker from the Data Editor, enabling it to load automatically the "UNIT_1" file. The Web browser user will be authorized to send values to the PLC, having first entered the write access password.

```
<APPLET codebase="/classes" archive="SAComm.jar,RDE.jar"
code="com.schneiderautomation.rde.LiveDataApplet"
width="700" height="514">
<PARAM name="TEMPLATE" value="UNIT_1">
<PARAM name="AUTO_LOGIN" value="FALSE">
</APPLET>
```

5.2 Graphic Editor

At a Glance

Subject of this Section

This section describes the functions and characteristics of the Graphic Editor. The Graphic Editor is a Web page that enables the user to create dynamic graphic displays with a browser and using a predefined set of graphic objects. The Graphic Editor is both a graphic editor that can be used to create and modify displays, and a Runtime environment that allows the user to view animated displays using data from the PLC.

What's in this Section?

This section contains the following topics:

Overview of the Graphic Editor

Top Window

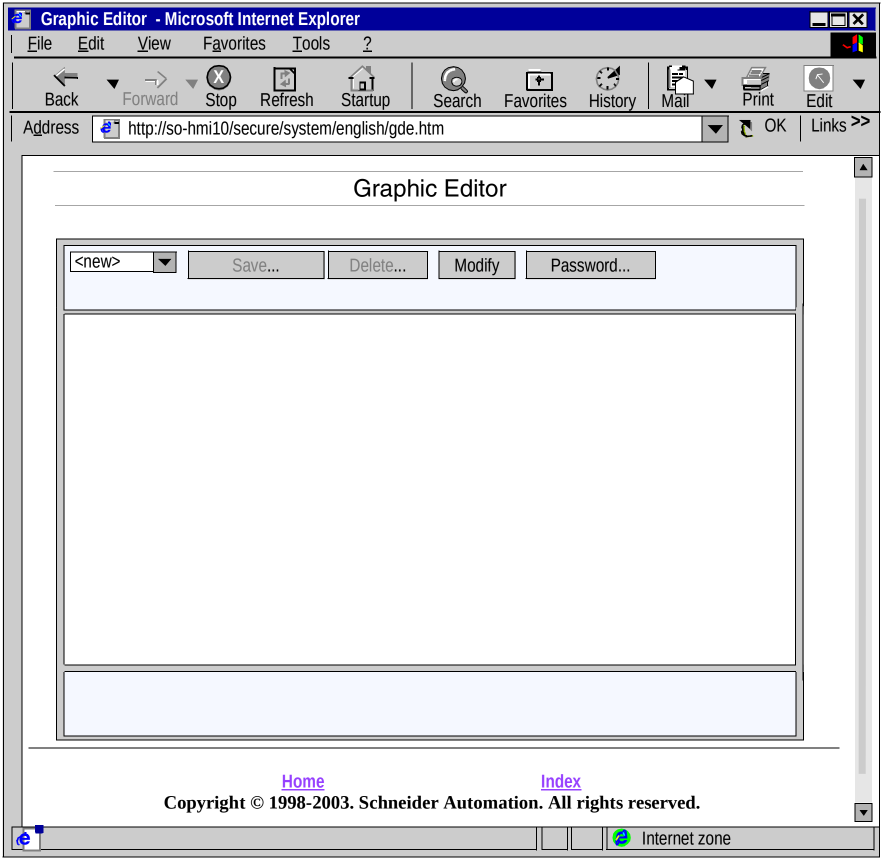
The Graphic Editor is made up of three windows. The top window features an area for presenting the user commands and dialog boxes for creating, saving, reading and editing a graphic editor.

Display Window

The display window proposes an presentation area of the current graphic display. When you create a new graphic display, this window turns into a blank space into which you can add the graphic objects that will make up the required graphic display.

Message window

This window proposes a scroll message window used to display the messages generated by the Graphic Editor. The figure below shows the Graphic Editor with its initial top window and display window.



Copyright © 1998-2003. Schneider Automation. All rights reserved.

Graphic Objects

All the graphic objects supplied with the Graphic Editor are able to communicate with the PLC. They are also designed as standalone objects, which means no connection is needed between the objects, and that each object is capable of operating on its own.

Viewing the Existing Graphic Pages

Once the Graphic Editor is loaded in the Web browser, you can:

- either view the existing graphic pages,
- or create or modify a graphic page.

If you only wish to view an existing graphic page, simply select it from the specially provided scroll menu.

The user only has to enter a password if he wishes to write data to the PLC.

Creating and Modifying Graphic Displays

If you wish to create and modify a graphic page, click **Modify...** which will give you access to the specific editing functions. Then, simply select the objects from the palette, place them in a space, move them and adjust their size, and define their properties. You can immediately test the graphic display updated with the PLC data by quitting edit mode (click **Done**). If you want to keep the graphic display you have created, you can save it in the PLC for future use by clicking **Save...**, on condition you have entered the correct password.

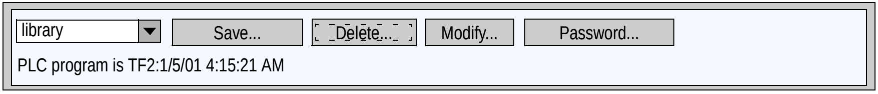
Top Window User Functions

Overview

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

Top Dialog

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



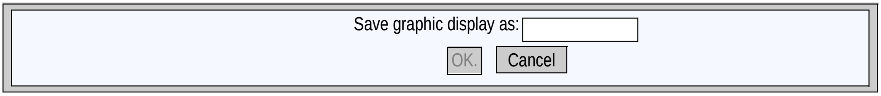
PLC program is TF2:1/5/01 4:15:21 AM

The controls of the **Top dialog box** provide the following functions.

- **Drop-down List**. The drop-down list box shows all the graphic pages that have been saved to the FactoyCast HMI module, and are available for retrieving. When you select a graphic page from this list, the graphic display currently visible in the window is replaced with the one selected. If the current graphic page has been modified since it was last saved, you will be asked for confirmation that the changes are to be discarded. If the special entry <new> is chosen from the list, then the display window is cleared, and a new graphic page can be created.
- **Save**. The **Save** button causes the **Save dialog box** to become visible. This button is disabled until you have entered a correct write-enable password.
- **Delete**. The **Delete...** button causes the **Delete dialog box** to become visible. This button is disabled until you have entered a correct password, or if the current graphic display has not yet been saved.
- **Modify**. The **Modify...** button causes the **Edit dialog box** to become visible.
- **Password**. The **Password...** button causes the **Password dialog box** to become 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

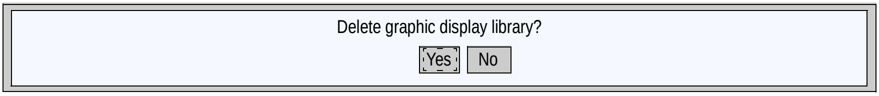
The **Save dialog box** allows you to save the current graphic display.



When the **Save dialog box** is presented, the name of the current graphic page is shown in the dialog’s text field. If the current graphic page has never been saved (i.e., a "new" graphic display), then the text field is blank. Once you have either accepted the current name (a "save" operation) or provided a new name (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 box** to be shown again, with no action being taken.

Delete Dialog

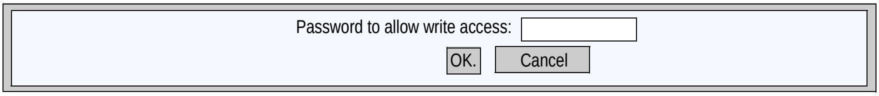
The **Delete dialog box** allows you to delete the current graphic page.



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

Password Dialog

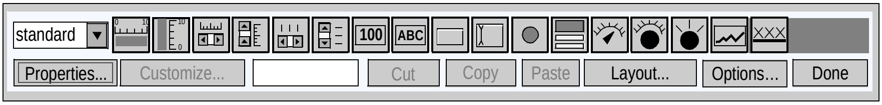
The **Password dialog box** allows you to enter the password that enables those user functions that modify graphic display files or PLC run-time data values.



If you enter the correct password and click the **OK** button, then you will be allowed to save and delete the current graphic display. Correct password entry also permits you to write new values to the PLC (via those graphic objects that support writing values to a PLC, if any). If you click the **OK** button when the text field is empty, then the current password permissions, if any, are cleared. The **Cancel** button will cause the **Top dialog box** to be shown again, with no changes made to current password permissions.

Edit Dialog

The **Edit dialog box** allows you to create or modify a graphic page, by selecting a graphic object for placement in the display window, and accessing all the graphic editing functions. The graphic objects available to you are presented in a single object palette.

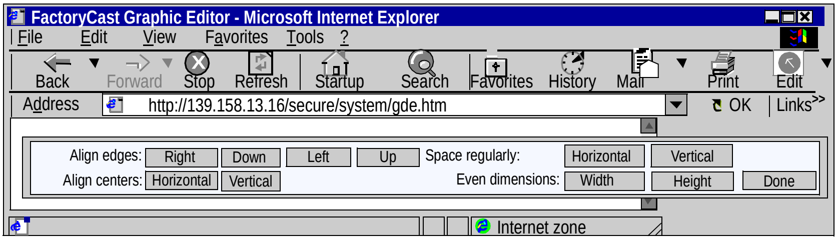


The controls of the **Edit dialog box** provide the following functions.

- The **Drop-down List Box** shows the set of palettes that are available. When you select the name of a palette from the list, the graphic objects that are in that palette are presented in the palette display area of the dialog.
- The **Palette** shows the graphic objects that are in the current palette with an icon that depicts each graphic object's type (meter, button, etc.). When you click any of the icons in the palette, a graphic object of the corresponding type becomes selected for insertion. While the Graphic Editor is in "insert mode," if you click in an open area of the display window, 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 any existing contents of the buffer.
- The **Copy** button causes the currently selected graphic object(s) to be copied to the buffer, replacing any existing contents.
- 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 causes the Properties Sheet for the currently selected graphic object to be shown. See .
- The **Customize** button causes the Customizer for the currently selected object to be shown, if the graphic object has been provided with one. See .
- The **Layout** button causes the **Layout dialog box** to become visible.
- The **Options** button causes the **Options dialog box** to become visible.
- The **Done** button causes the **Top dialog box** to be shown again.

Layout Dialog

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


FactoryCast Graphic Editor - Microsoft Internet Explorer
File Edit View Favorites Tools ?
Back Forward Stop Refresh Startup Search Favorites History Mail Print Edit
OK Links>>
Align edges: Right Down Left Up
Space regularly: Horizontal Vertical
Align centers: Horizontal Vertical
Even dimensions: Width Height Done
Internet zone

Address http://139.158.13.16/secure/system/gde.htm

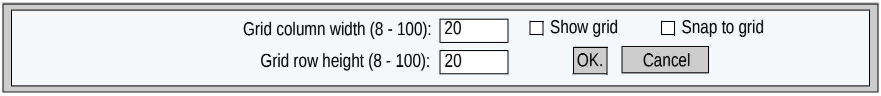
The controls of the **Layout dialog box** provide the following functions.

- The **Right**, **Bottom**, **Left**, and **Top** buttons can be used to align the edges of the selected graphic objects so that their specified sides are at the same position. At least two graphic objects must be selected for these buttons to be enabled.
- The **Horizontal**, and **Vertical** buttons are used to align the centers of the graphic objects. At least two graphic objects must be selected for these buttons to be enabled.
- The **Horizontal** and **Vertical** buttons are used to space the selected graphic objects regularly, in order that the horizontal or vertical spacing between the objects is the same. At least three graphic objects must be selected for these buttons to be enabled.
- The **Width** and **Height** buttons are used to achieve parity in dimensions of the graphic objects, so the selected width or height corresponds. At least two graphic objects must be selected for these buttons to be enabled.
- The **Done** button causes the **Edit dialog box** to be shown again.

Note: For all layout operations (except **Space evenly**) one of the selected objects is considered the "reference object" to which all other selected objects refer in order to know their new position or dimension. For example, when the "Width" button is pressed, all of the selected objects will have their width changed to match the width 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 box** is used to change the settings related to a grid drawn in the display window. The grid is solely for assistance in editing or creating a graphic display and is shown only when the Graphic Editor is in "edit mode."



Grid column width (8 - 100): 20 ☐ Show grid ☐ Snap to grid

Grid row height (8 - 100): 20

The controls of the **Options dialog box** provide the following function.

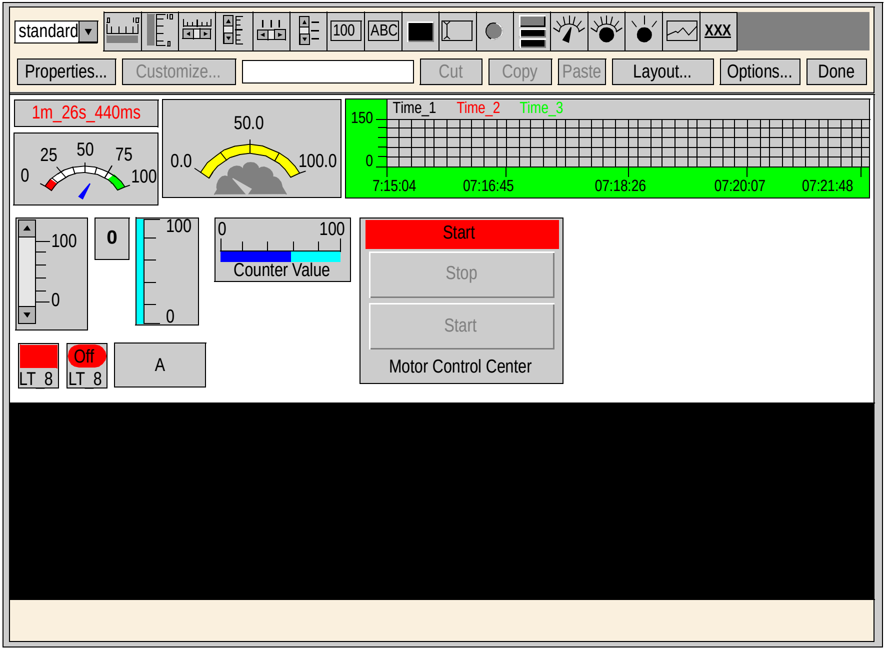
- The cell size of the grid can be changed by the entering the grid's column width and row height into the dialog's text fields.
- If the **Show grid** check-box is checked, the grid will be drawn; otherwise, no grid will be shown.
- If the **Snap to grid** check-box is checked, then, when you change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

User Functions of the Display Window

Overview

The user functions available in the display window of the **Graphic Editor** are used to select and move objects, and redefine their size. All moving or size redefinition operations require the graphic object(s) to be selected before being modified. The object selected appears framed on the screen.

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



Selecting Graphic Objects

The selection status of a graphic object (selected/deselected) can be defined by the following user actions:

- You can select a single graphic object by simply clicking on it. If other objects are already selected, they are automatically deselected.
- You can select several graphic objects by creating a rectangle over them in the display window. Left-click with the mouse in an open area of the display window (not on a graphic object) and, holding down the button, drag the curser to draw a dotted frame. One of the corners of the rectangle will be where you first clicked, and the corner diagonally opposite will follow the path of the curser. When you release the button, all the objects within the box will be selected. All objects outside this box will be deselected.
- You can select or deselect a graphic object by holding down the CTRL key and clicking on an object. Here, the selection status of all other objects will remain unchanged. With this action, each graphic object can be added to the current group of selected or deleted objects.
- You can select a graphic object by holding down the SHIFT key and clicking on an object. Here, the selection status of all other objects will remain unchanged. With this method, when one object is selected it becomes the *reference object* (see the layout dialog box in section) for the group of objects selected. The main aim of this action is to change the reference object in a group of selected objects. The **layout** operations come later.
- You can deselect all the graphic objects by clicking in an open area of the display window (and not on a graphic object).

Defining the Size of the Graphic Objects

You can modify the size of a graphic object by first selecting it then using the mouse to change the size of the object's selection box. As you move the curser on an object's selection box, this changes to reflect the type of operation to perform to redefine the size. If you press the left mouse button with the curser on an object's selection box, then hold down the button and move the object, a dotted frame appears. When you release the button, the size of the object is modified to correspond to the frame you have drawn. There are eight possible ways of redefining the size, according to the part of the object's selection box from which you drag the mouse. The corners of the frame will only move the adjacent sides, and each side of the frame can be moved on its own.

Moving Graphic Objects

The position of a graphic object in the display window can be modified using the mouse. If you press the left mouse button with the curser on an object, then hold down the button and move the object, a dotted frame appears. When you release the button, the position of the object is modified to correspond to the frame you have drawn.

You can move several graphic objects by first selecting the objects to move, then moving the whole group as you would a single object. When you move a group of objects, a box delimits each object of the group.

Defining the Properties of the Graphic Objects

You can define the properties of a graphic object using the **Property Sheet** (see property sheet in section). If the property sheet is displayed, the characteristics of the graphic object selected can be edited. You can display the property sheet by clicking **Properties...** or double-clicking anywhere on the selected object in the display window.

Customizing Complex Graphic Objects

Certain complex graphic objects have a large number of properties. The configuration process for graphic objects like these using the property sheet can be long. To simplify the configuration of complex graphic objects, you can use a customization module. This is a dialog window designed specially to configure associated graphic objects. . This button presents the customization module of the graphic object. When you double-click on a graphic object with a customization module, this is what is presented, not its properties sheet. If a graphic object has a customization module, the only characteristic to appear in its properties sheet is its name.

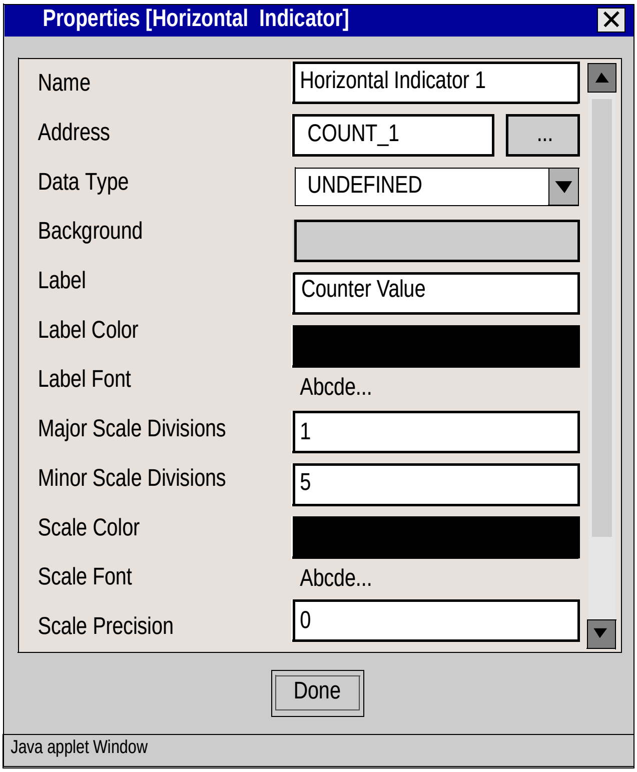
Displaying a Background Image

The Graphic Editor display has a **Background Image** characteristic which can be used to specify an image to appear in the background of the display. This image can be a GIF file or a JPEG file. All files are placed in the /wwwroot directory of the embedded server. For example, if the image "cool.gif" was placed in the /wwwroot/images directory of the embedded server, the background image property must be set to /wwwroot/images.

Property Sheet

Overview

The Property Sheet is a "floating" (non-modal) dialog that presents all the settable properties of the currently selected graphic object.



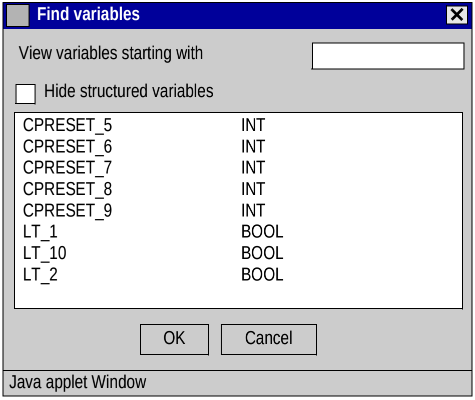
Name	Horizontal Indicator 1
Address	COUNT_1
Data Type	UNDEFINED
Background	▒
Label	Counter Value
Label Color	■
Label Font	Abcde...
Major Scale Divisions	1
Minor Scale Divisions	5
Scale Color	■
Scale Font	Abcde...
Scale Precision	0

The properties of a graphic object are specific to an object’s type. The properties are presented in a scrollable list, with the name and the value of each property listed. See for a description of the graphic objects provided with the Graphic Editor.

Find variables dialog box

For each of the graphic objects provided with the Graphic Editor, a property editor is provided for its **Address** property. This property editor not only allows you to directly enter the address of a Quantum/Premium/Micro register (or Concept/PL7/Unity Pro variable name), but also provides access to the **Find variables dialog box**. 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 the FactoryCast Configurator.

This is the **Find variables dialog box**.



View variables starting with

□ Hide structured variables

CPRESET_5	INT
CPRESET_6	INT
CPRESET_7	INT
CPRESET_8	INT
CPRESET_9	INT
LT_1	BOOL
LT_10	BOOL
LT_2	BOOL

Security

Security

Your API data is protected by three security measures.

- The HTML page containing the Graphic Editor applet has been placed in a "secure" directory on the Web module; the Web browser user is therefore asked to enter a password enabling him/her to download the HTML page.
- You must enter the correct password in the **Password** dialog box to be able to save/delete the files or send the data values to the connected PLC. For the transfer of data values to the PLC, the Graphic Editor reinforces the "read only" mode by deactivating the user commands of all the graphic objects.
- The FactoryCast configuration program lets you indicate that an element is in read only. The **Graphic Editor** reinforces the "read only" attribute of a symbol variable or address by rejecting all requests in order to define a new value for the data, and warning the user through the message window **Graphic Editor.**

Parameters of the Graphic Editor Applet

Overview

The **Graphic Editor** supports three applet parameters for customizing its behavior. The applet parameters are indicated by the <PARAM> labels in the <APPLET> label of the Graphic Editor HTML page. The parameters recognized by the **Graphic Editor** applet are:

- **LOAD**—This parameter asks the **Graphic Editor** to automatically load a graphic file when it is launched. If the specified file does not exist, an error message is delivered. If the parameter does not appear in the <APPLET> label, no graphic file is automatically loaded on startup. In this case you must select the initial table file from the list provided by the **Graphic Editor**.
- **MODE**—This parameter tells the **Graphic Editor** whether it has to start normally in edit mode or in view mode. If it is launched in view mode, it will **only** display the display window. When this parameter is used with the LOAD parameter, a website can be designed using HTML pages that are reserved for specific graphic display. The user is not required to explicitly select any graphic files, which is typical HMI behavior. The possible values for this parameter are
  - **EDIT**—The **Graphic Editor** starts normally in edit mode (default value).
  - **VIEW_RO**—The **Graphic Editor** starts normally in read only. The Web browser user will not be authorized to send values to the PLC.
  - **VIEW_RW**—The **Graphic Editor** starts normally in read/write. The Web browser user will be authorized to send values to the PLC, having first entered the write access password.
- **AUTO_LOGIN**—This parameter tells the **Graphic Editor** to automatically enter the password required to write access the PLC. If the **MODE** parameter is set to **VIEW_RW** or **EDITI** and also set to **TRUE**, the **Graphic Editor** grants write access to the PLC without asking the user to enter the password. The possible values for this parameter are **FALSE** (by default) and **TRUE**.

Example

The following example shows an applet invocation marker that starts the **Graphic Editor** in view mode and automatically changes the graphic file called **UNIT_1**. The Web browser user will be authorized to send values to the PLC though graphic objects that take into account the sending of values if the password for write access in entered.

```
<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 in the **Graphic Editor** is intended to support building graphic displays that mimic conventional instrument panels. All of the data monitoring and control objects have built-in communication capabilities and are designed as stand-alone graphic objects.

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`, the **Graphic Editor** graphic objects can be used in the same way as the `LiveLabelApplet`.

This section provides a description of the 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 by drawing a horizontal bar whose length is proportional to 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.

These are the properties for the Horizontal Indicator.

Property	**Description**	**Limits**
Name	The name for the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1,*Notes, p. 152*
Data Type	The data type of the direct address or symbol (variable)	See Note 2,*Notes, p. 152*
Background	The background color for the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color for the label	
Label Font	The font 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 for the scale and its labels	
Scale Font	The font 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, p. 152*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3,*Notes, p. 152*
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 for the bar indicator area	
Bar Color	The color for the indicator bar (when 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 for the indicator bar when 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 for the indicator bar when scaled value is greater than the ‘High’ limit	
Low Limit Value	The value in engineering units for the ‘Low’ limit	
Low Limit Color	The color for the indicator bar when 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 for the indicator bar when 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) for the graphic object’s border	0 to 32
Border Color	The color for the graphic object’s border	
PLC Value	A simulated, raw (unscaled) input value for testing the graphic object	See Note 3,*Notes, p. 152*

Vertical Indicator

A Vertical Indicator provides an analog representation of the value of a symbol (variable) or direct address in a PLC by drawing a vertical bar whose length is proportional to the value as a percentage of its range in engineering units. These are the properties for the Vertical Indicator.

Property	Description	Limits
Name	The name for the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1,*Notes, p. 152*
Data Type	The data type of the direct address or symbol (variable)	See Note 2,*Notes, p. 152*
Background	The background color for the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color for the label	
Label Font	The font 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 for the scale and its labels	
Scale Font	The font 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, p. 152*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3,*Notes, p. 152*
Bar Background	The background color for the bar indicator area	
Bar Color	The color for the indicator bar (when 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 for the indicator bar when scaled value is greater than the ‘High High’ limit	

Property	Description	Limits
High Limit Value	The value in engineering units for the ‘High’ limit	
High Limit Color	The color for the indicator bar when scaled value is greater than the ‘High’ limit	
Low Limit Value	The value in engineering units for the ‘Low’ limit	
Low Limit Color	The color for the indicator bar when 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 for the indicator bar when 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) for the graphic object’s border	0 to 32
Border Color	The color for the graphic object’s border	
PLC Value	A simulated, raw (unscaled) input value for testing the graphic object	See Note 3,*Notes, p. 152*

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 by drawing a scroll bar whose "thumb" position is proportional to the value as a percentage of its range in engineering units. With a mouse, a user can change the value of the scroll bar and cause a new value to be sent to the PLC.

These are the properties for the Horizontal or Vertical Slider.

Property	**Description**	**Limits**
Name	The name for the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1,*Notes, p. 152*
Data Type	The data type of the direct address or symbol (variable)	See Note 2,*Notes, p. 152*
Background	The background color for the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color for the label	
Label Font	The font 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 for the scale and its labels	
Scale Font	The font 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, p. 152*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3,*Notes, p. 152*
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 scrollbar's arrow buttons are clicked	

Property	Description	Limits
Border Width	The width (in pixels) for the graphic object's border	0 to 32
Border Color	The color for the graphic object's border	

Horizontal or Vertical Selector

A Horizontal or Vertical Selector allows a user to make a selection 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.
These are the properties for the Horizontal or Vertical Selector.

Property	Description	Limits
Name	The name for the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1,*Notes, p. 152*
Data Type	The data type of the direct address or symbol (variable)	See Note 2,*Notes, p. 152*
Background	The background color for the graphic object	
Choices	The choices for the selector. Each choice is given as a 'label=value' entry (when a user selects '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 for the label	
Label Font	The font for the label	
Scale Visible	Indicates whether a "scale," labeled with the choices, is to be shown	
Scale Color	The color for the scale and its labels	
Scale Font	The font for scale labels	
Border Width	The width (in pixels) for the graphic object's border	0 to 32
Border Color	The color for 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. These are the properties for the Digital Indicator.

Property	Description	Limits
Name	The name for the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1,*Notes, p. 152*
Data Type	The data type of the direct address or symbol (variable)	See Note 2,*Notes, p. 152*
Background	The background color for the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color for the label	
Label Font	The font 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 for the value display area	
Value Color	The text color for the digital display of the value	
Value Font	The font for the digital display of the value	
Units	The label for the engineering units of the value (appended to the numeric 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, p. 152*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3,*Notes, p. 152*
High High Limit Value	The value in engineering units for the 'High High' limit	
High High Limit Color	The color for the indicator bar when scaled value is greater than the ‘High High’ limit	
High Limit Value	The value in engineering units for the ‘High’ limit	

Property	Description	Limits
High Limit Color	The color for the indicator bar when scaled value is greater than the ‘High’ limit	
Low Limit Value	The value in engineering units for the ‘Low’ limit	
Low Limit Color	The color for the indicator bar when 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 for the indicator bar when 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) for the graphic object’s border	0 to 32
Border Color	The color for the graphic object’s border	
PLC Value	A simulated, raw (unscaled) input value for testing the graphic object	See Note 3,*Notes, p. 152*

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 value is also specified that will trigger its display.

These are the properties for the Message Display.

Property	Description	Limits
Name	The name for the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1,*Notes, p. 152*
Data Type	The data type of the direct address or symbol (variable)	See Note 2,*Notes, p. 152*
Background	The background color for 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 for the message display area	
Message Color	The color for the message text	
Message Font	The font for the message text	
Label	The label to be displayed as part of the graphic object	
Label Color	The color for the label	
Label Font	The font for the label	
Border Width	The width (in pixels) for the graphic object's border	0 to 32
Border Color	The color for the graphic object's border	
PLC Value	A simulated input value for testing the graphic object	See Note 3,*Notes, p. 152*

Push Button

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

These are the properties for the Push Button.

Property	Description	Limits
Name	The name for the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1,*Notes, p. 152*
Data Type	The data type of the direct address or symbol (variable)	See Note 2,*Notes, p. 152*
Background	The background color for the graphic object	
Values	The value(s) to send to the PLC	See Note 4,*Notes, p. 152*
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 (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 for the label	
Label Font	The font for the label	
Button Label	The text label for the button	
Button Background	The color for the knob	0 to 100
Button Label Color	The color for the button label	
Button Label Font	The font for the button label	
Border Width	The width (in pixels) for the graphic object's border	0 to 32
Border Color	The color for the graphic object's border	

Direct Output Station

The Direct Output Station allows a user to enter a value into a text input field directly with their keyboard. When the entered text represents a numeric value that is within preset high and low limits, a **Set** button is enabled. While the **Set** button is enabled, the entered value will be sent to the PLC whenever the user presses either the **Set** button or the ENTER key (if the input field has keyboard input focus).

These are the properties for the Direct Output Station.

Property	Description	Limits
Name	The name for the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1,*Notes, p. 152*
Data Type	The data type of the direct address or symbol (variable)	See Note 2,*Notes, p. 152*
Background	The background color for the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color for the label	
Label Font	The font 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, p. 152*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3,*Notes, p. 152*
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) for the graphic object's border	0 to 32
Border Color	The color for the graphic object's border	

LED

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.

These are the properties for the Indicator Light.

Property	Description	Limits
Name	The name for the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1,*Notes, p. 152*
Data Type	The data type of the direct address or symbol (variable)	See Note 2,*Notes, p. 152*
Background	The background color for the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color for the label	
Label Font	The font 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 for the **Off Word** text	
Off Word Font	The font 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 for the **On Word** font	
On Word Font	The font for the **On Word** text	
Flash Interval	The flashing time period (in milliseconds) of the light when the input value is on. Set to zero for no flashing.	200 to 2000
Shape	The shape (circle, rectangle, etc.) of the light	
Input Inverted	If **TRUE**, inverts the input value. (Light will show the **Off Word** when input value is on.)	
Border Width	The width (in pixels) for the graphic object's border	0 to 32
Border Color	The color for the graphic object's border	
PLC Value	A simulated input value for testing the graphic object	See Note 3,*Notes, p. 152*

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. In order to make it easier to set this object's many properties, a Customizer is provided. All of the properties (except Name) are set with its Customizer, not with the **Graphic Editor's** Property Sheet.

These are the properties for the Motor Control Station.

Property	Description	Limits
Name	The name for the graphic object	
Background	The background color for the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color for the label	
Label Font	The font for the label	
Border Width	The width (in pixels) for the graphic object's border	0 to 32
Border Color	The color for the graphic object's border	
LED	Same properties as the Indicator Light graphic object, excluding the common properties listed above	
Top Push Button	Same properties as the Push Button graphic object, excluding the common properties listed above	
Bottom Push Button	Same properties as the Push Button graphic object, excluding the common 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 by drawing a pointer on a circular dial whose position is proportional to the value as a percentage of its range in engineering units. The size of the meter's circular dial (degrees sweep of a circle), the colors for the dial, and the style of the pointer can all be set.

These are the properties for the Analog Meter.

Property	Description	Limits
Name	The name for the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1,*Notes, p. 152*
Data Type	The data type of the direct address or symbol (variable)	See Note 2,*Notes, p. 152*
Background	The background color for the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color for the label	
Label Font	The font 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 for the scale and its labels	
Scale Font	The font 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, p. 152*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3,*Notes, p. 152*
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	

Property	**Description**	**Limits**
Dial Color	The color for the dial (that part that is within the High/Low limits)	
High High Limit Value	The value in engineering units for the ‘High’ limit	
High High Limit Color	The color for the indicator bar when 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 for the indicator bar when scaled value is greater than the ‘High’ limit	
Low Limit Value	The value in engineering units for the ‘Low’ limit	
Low Limit Color	The color for the indicator bar when 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 for the indicator bar when scaled value is less than the ‘Low Low’ limit	
Border Width	The width (in pixels) for the graphic object’s border	0 to 32
Border Color	The color for the graphic object’s border	
PLC Value	A simulated, raw (unscaled) input value for testing the graphic object	See Note 3,*Notes, p. 152*

Rotary Slider

A Rotary Slider provides an analog representation of the value of a symbol (variable) or direct address in a PLC by drawing a knob on a circular dial whose position is proportional to the value as a percentage of its range in engineering units. The size of the circular dial (degrees sweep of a circle) and knob color can be set. With a mouse, a user can change the position of the knob and cause a new value to be sent to the PLC.

These are the properties for the Rotary Slider.

Property	**Description**	**Limits**
Name	The name for the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1,*Notes, p. 152*
Data Type	The data type of the direct address or symbol (variable)	See Note 2,*Notes, p. 152*
Background	The background color for the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color for the label	
Label Font	The font 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 for the scale and its labels	
Scale Font	The font for scale labels	
Scale Precision	The number of fractional digits to be shown for scale labels (Set to -1 to use a general exponential format.)	-1 to 6
Dial Degrees Sweep	The amount of a circular arc to use for drawing the dial	60 to 300
Dial Color	The color for the dial	
Knob Color	The color 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, p. 152*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3,*Notes, p. 152*

Property	Description	Limits
Border Width	The width (in pixels) for the graphic object’s border	0 to 32
Border Color	The color for the graphic object’s border	

Rotary Selector

A Rotary Selector allows a user to make a selection 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.

These are the properties for the Rotary Selector.

Property	Description	Limits
Name	The name for the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1,*Notes, p. 152*
Data Type	The data type of the direct address or symbol (variable)	See Note 2,*Notes, p. 152*
Background	The background color for the graphic object	
Choices	The choices for the selector. Each choice is given as a ‘label=value’ entry (when a user selects ‘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 for the label	
Label Font	The font for the label	
Scale Visible	Indicates whether a "scale," labeled with the choices, is to be shown	
Scale Color	The color for the scale and its labels	
Scale Font	The font 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 for the knob	
Border Width	The width (in pixels) for the graphic object’s border	0 to 32
Border Color	The color for the graphic object’s border	

Trend Recorder

A Trend Recorder provides a continuous, time-based charting of the value of up to six symbol (variable)s 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. The rate at which the chart is updated, and the appearance of the chart can be set.

In order to make it easier to set this object's many properties, a Customizer is provided. All properties (except Name) are set with its Customizer, not with the **Graphic Editor's** Property Sheet.

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

Property	Description	Limits
Name	The name for the graphic object	
Background	The background color for the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color for the label	
Label Font	The font 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 for the scale and its labels	
Scale Font	The font 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 for 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

Property	Description	Limits
Border Width	The width (in pixels) for the graphic object’s border	0 to 32
Border Color	The color for 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, p. 152*
Data Type	The data type of the direct address or symbol (variable).	See Note 2,*Notes, p. 152*
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC.	See Note 3,*Notes, p. 152*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC.	See Note 3,*Notes, p. 152*
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 the user to switch to another graphic display by clicking on it with a mouse. To indicate to the user 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 its **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 the user presses the SHIFT key while they click 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.

These are the properties for the Display Link.

Property	Description	Limits
Label	The label for 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 for the label	
Label Font	The font for the label	

Notes

These are the notes for the chapter.

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 (variable)'s actual data type.

If the Address property is a direct address for a discrete PLC reference (Quantum 0x/1x reference), the Data Type property must be set to BOOL. The Data Type property may be set to BOOL only for a discrete PLC reference.

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

Data Type	Meaning
UNDEFINED	no data type specified
BOOL	1-bit discrete (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 a REAL with respect to its limit values.

4. For a Push Button, a minimum of one value must be provided. 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 will be sent to the PLC as an array of values starting at the specified direct address.

Description of the TSX WMY 100 and 140 NWM 100 00 modules

II

At a Glance

Subject of this Part

This part describes the Premium TSX WMY 100 module and the Quantum 140 NWM 100 00 module.

What's in this Part?

This part contains the following chapters:

Compatibility with the different software workshops

6

FactoryCast HMI module compatibility with different software workshops

At a Glance

To implement FactoryCast HMI applications, you need to configure the module in the appropriate software workshop (i.e. Unity Pro, Concept or PL7).

Compatibility

The table below shows which FactoryCast HMI modules are compatible with which software workshop versions:

	Unity Pro	**Concept**	**PL7**
TSX WMY 100	V1.0		V4.3 with TSX WMY 100 update
140 NWM 100 00	V1.0	V2.6 + update on FactoryCast HMI CD	

Note: Earlier software versions are always upwardly compatible with later versions.

TSX WMY 100 module

7

At a Glance

Subject of this Chapter

This chapter covers the implementation of hardware for a **TSX WMY 100** HMI module on a Premium PLC.

What's in this Chapter?

This chapter contains the following sections:

7.1 At a Glance

At a Glance

General

The **TSX WMY 100** HMI module is used by FactoryCast HMI software. It is made up of a communication channel whose main features are as follows:

- Connection to a TCP/IP network.
- Communication in Half and Full Duplex mode by automatic recognition.
- Transmission speed from 10 or 100 Mbits/s by automatic recognition.
- Connection to network by copper cable via an RJ45 connector.

This module is used to carry out the following functions:

- X-WAY UNI-TE and Modbus messaging service on TCP/IP.
- SNMP Service,
- Web server,
- E-mail service,
- Database.

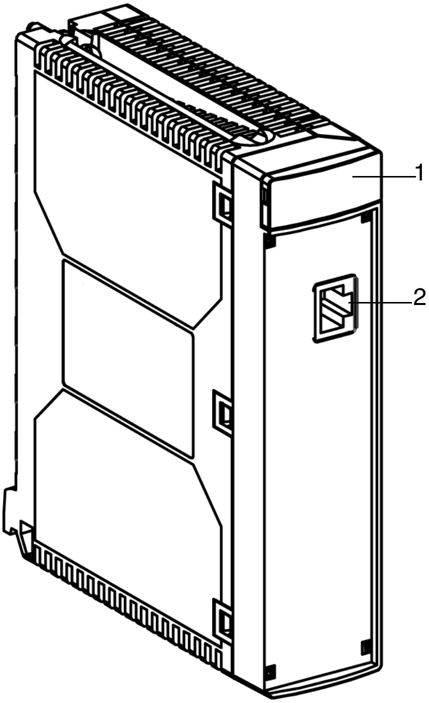
7.2 Description

Description

General

The **TSX WMY 100** HMI module is a standard format module which is inserted in a slot on the main or extension rack of a Premium PLC station.

Description:

1. Display panel indicating module status: ● One green **RUN** LED. ● Two red LEDs: **ERR** and **COL**. ● Three yellow LEDs: **STS**, **TX** and **RX**. **2.** Standard connector for 10baseT (RJ45) interface.	Illustration: 

7.3 Ethernet Channel Characteristics

Communication Channel Characteristics

General

The **TSX WMY 100** HMI module has a standard interface for connecting to a 10/100baseT network and on the front panel there is a RJ45 connector for a point-to-point link via a link cable comprising two independent twisted pairs.

Services and functions supported by the modules:

Support	Service	Protocol	Functions
TCP-IP Services	Messaging	UNI-TE	● Client/server mode. ● Synchronous requests of 256 bytes. ● Asynchronous requests of 1 Kbyte.
		Modbus	● Data exchange.
	Network management	SNMP	● SNMP agent, (MIB II, Schneider MIB).
	Web	HTTP	● Website which can be modified and increased by increments within the limit of 7.5Mb.

Maximum Capacity of the Module

The maximum frame size depends on the type of transaction:

- In synchronous messaging, the maximum frame size is 256 bytes.
- In asynchronous messaging, the maximum frame size is 1 KByte.

The **TSX WMY 100** module is used:

- to manage 64 TCP connections using the port 502 messaging service,

In conjunction with a processor dedicated to this sort of processing, the module can be used:

- for synchronous X-WAY messaging on TCP/IP (UNI-TE server):
  - with a TSX P57-453 processor: 800 messages per second,
  - with a TSX P57-303 processor: 490 messages per second,
- in the case of asynchronous X-WAY messaging on TCP/IP:
  - between 600 and 1200 messages per second (number varies depending on the size of the messages, the number of clients and the application execution time).

Faulty Device Replacement

The module reserves a 512 KBytes zone for this function.

Compatibility

The **TSX WMY 100** module can only be configured in a TSX Premium with a V5.1 or higher processor.
The functional level of the **TSXWMY 100** module requires version V4.4 of the PL7 software.

7.4 Installing the Module

At a Glance

Subject of this Section

This section covers the installation of a **TSX WMY 100** HMI module on a Premium PLC.

What's in this Section?

This section contains the following topics:

At a Glance

General

The **TSX WMY 100** HMI module is mounted in the rack slot of a Premium/Atrium PLC station. It can be installed in any available slot (except in an offset X Bus rack), on condition that the supply constraints of the rack are observed.

Selecting the Type of Processor

Selection Guide

Selecting the processor to control the PLC station will depend on the number of network connections required.

Processors	Number of network connections	Number of TSX WMY 100 modules
TSX P57 1••/2•• PCX 57 253	1	1
TSX P57 3•• PCX 57 353	3	3
TSX P57 4••	4	4
TSX P57 2•23/2634	1	0
TSX P57 3623/3634	3	2
TSX P57 4823	4	3
TSX P57 5634	5	4
(*) assuming that the power consumption breakdown on the 5V supply is compatible with the power supply selected.		

Connection/Disconnection with Power Switched On

The Module

The **TSX WMY 100** HMI module can be connected or disconnected when switched on without disrupting the operation of the station.
The modules do not have an internal RAM backup memory function: this will be erased when power is switched off.
The modules reset when switched on. A communication break can be expected during these interventions.

The link

The 10/100baseT interface RJ45 connector can be connected or disconnected when power is on. A communication break can therefore be expected in the application in progress.

7.5 10/100baseT Interface

10/100baseT Interface

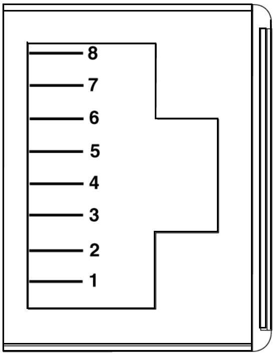
General

This interface has a standard type RJ45 connector.

Refer to the ETHERNET reference manual for the connection accessories that comply with environmental circumstances the PLC requires in an industrial setting.

Pin Assignment

Illustration:



Reminder of pin assignment:

Pin	Signal
1	TD+
2	TD-
3	RD+
4	not connected
5	not connected
6	RD-
7	not connected
8	not connected

Note: If there is a connection via a shielded cable, the connector casing on the module is linked up to the ground connection.

Speed Line

The choice of different line speeds for the TSX WMY 100 module are as follows:

- 100 Mb in Half Duplex
- 10 Mb in Half Duplex
- 10 Mb in Half Duplex

Speed Adaptation

The speed line cannot be configured by the user. The process of self adaptation is as follows:

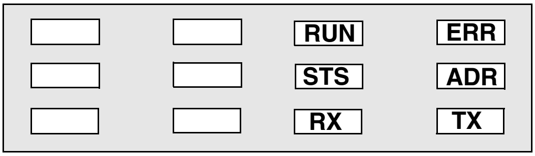
1	Each entity broadcasts its possibilities on the line.
2	The chosen speed is the fastest of the entity possibilities on the line. In other words, speed is limited by the slowest entity on the line of which the speed possibility is the weakest.

7.6 Display, Diagnostics

Display Panel, Diagnostics

General

The display panel conforms to the Premium standard



The COL, RX and TX LEDs are managed by the line's electronics; they indicate:

- COL: a collision.
- RX: a reception
- TX: a transmission.

Diagnostics

Meaning of the diagnostics LEDs:

RUN	ERR	STS	COL	TX	RX	Meaning
OFF	OFF	OFF	ns	ns	ns	No supply to module.
OFF	OFF	ON	ns	ns	ns	Module running self-test.
ON	OFF	OFF	ns	ns	ns	Module ready.
OFF	ON	OFF	ns	ns	ns	Module not operational.
OFF	ON	ON	ns	ns	ns	Software operation error. Temporary state causing module reinitialization.
OFF	F	ON, F	ns	ns	ns	Module not configured or configuration in progress.
ON	OFF	ON	ns	ns	ns	Module configured, operational.
ns	ns	F	ns	ns	ns	Module configured. Diagnostics according to how the LEDs are flashing: ● 2 flashes: module has no MAC address. ● 3 flashes: ETHERNET cable not connected on the module or Hub side ● 4 flashes: the module IP address is duplicated by another IP address on the network. Conflicting remote device flashing in the same way. ● 5 flashes: module configured as a BOOTP client and is waiting for a BOOTP server response.
ON	OFF	ON	OFF	F	OFF	Sending ETHERNET communication.
ON	OFF	ON	OFF	OFF	F	Receiving ETHERNET communication.
ON	OFF	ON	OFF	F	F	Sending/Receiving ETHERNET communication.
ON	OFF	ON	F	F	OFF	Module has detected collision.
ON = Permanently on, F = flashing, OFF = Off, ns = not significant						

7.7 Electrical Characteristics

Electrical Characteristics

General

The **TSX WMY 100** HMI module can be inserted in any rack slot of a Premium/ Atrium station (except in the X Bus offset rack).

Table of consumption:

Voltage	**Power consumption**		**Power dissipation**	
5 volts	Typical	Maximum	Typical	Maximum
TSX WMY 100	360 mA	400 mA	1.8 W	2.1 W

7.8 Standards

Norms and Standards

Compliance with Standards

The **TSX WMY 100** HMI module complies with the following standards:

- UL 508
- CSA
- IEC 1121-2
- Marine classification

7.9 Operating Conditions

Operating Conditions

Applicable Conditions

- **Configuration software:**
  - **PL7 version** ≥ **V4.4**: recognizes processors including TSX WMY 100 module.
- **Conditions of use:**
  - Temperature from 0 to +60° C
  - Relative humidity between 10% and 95% (without condensation) at 60° C
  - Altitude of between 0 and 4500 meters
  - Immunity to vibrations complies with the IEC 68-2-6 standard, Fc test
  - Immunity to shocks complies with the IEC 68-2-27 standard, Ea test
  - Immunity to free fall, hardware packaged as per IEC 68-2-32 standard, method 1
  - IP 20 protection rating
- **Storage conditions:**
  - Temperature from -40° C to +85° C
  - Relative humidity between 0% and 95% (without condensation) at 60° C

Quantum module 140 NWM 100 00

8

At a Glance

Subject of this Chapter

This chapter covers the implementation of hardware for a 140 NWM 100 00 Quantum HMI module.

What's in this Chapter?

This chapter contains the following sections:

8.1 140 NWM 100 00 Product Description

Purpose

Overview

This section provides an overview of the 140 NWM 100 00 primary features and LED indicators.

What's in this Section?

This section contains the following topics:

140 NWM 100 00 Module Overview

Overview

The following information provides an overview of the Quantum 140 NWM 100 00 module.

General Description

The Qauntum 140 NWM 100 00, 10/100 Ethernet module, is the latest model in a line of Quantum Ethernet TCP/IP modules. The module is designed to make it possible for a Quantum Programmable Logic Controller (PLC) to communicate with devices over an Ethernet network. The electronics for the 140 NWM 100 00 module are contained in a standard Quantum single width case that takes up one slot in a Quantum backplane.

The 140 NWM 100 00 provides real-time peer-to-peer communications and a MODBUS/TCP server. The included HTTP services provide maintenance and configuration utilities to the module.

The following table shows the different features of the 140 NWM 100 00 Ethernet module. A key new feature of this module is that it supports embedded Java Virtual Machine.

Key Features

This is a module that provides a PLC with access to an Ethernet network. The module can plug into any available slot with a Quantum backplane and is capable of being hot swapped. The key features are listed below.

Feature	140 NWM 100 00
Java Virtual Machine	yes
FTP Server	yes
Flash File System	yes
HTTP Server	yes
SNMP v1	yes
Schneider Private MIB	yes
FactoryCast HMI	yes
User Programmable Web Pages	yes
Modbus Messaging	yes

Front Panel Components

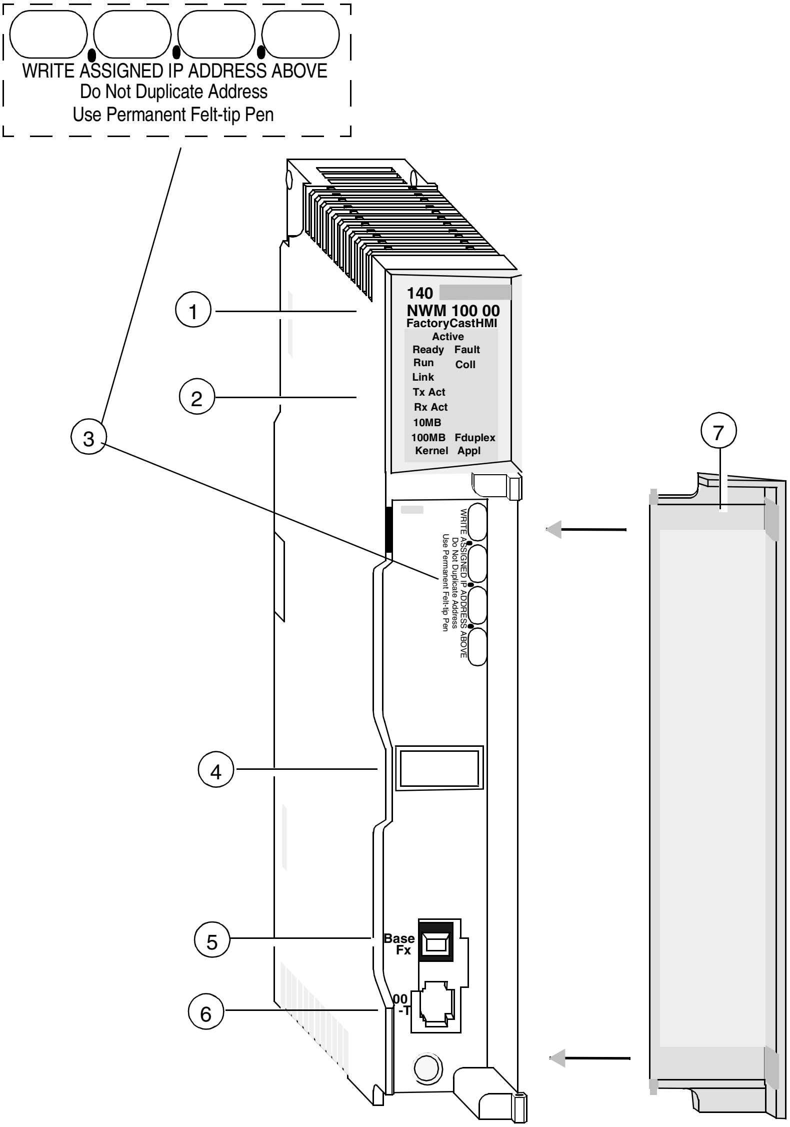
The front panel of the 140 NWM 100 00 module contains identification markings, color codes, and LED displays. A writable area for an IP address, a global address label, and two Ethernet cable connectors are located behind the removable front panel door.

The following table provides a description of the front panel components, which are shown in front view figure.

Component	**Description**
LED indicator Panel	Indicates the operating status of the module and the fiber optic or Ethernet communications network to which it is connected. (See LED Indicators in this chapter.)
IP Address Area	Provides a writable area to record the module's assigned IP address.
Global Address Label	Indicates the module's global Ethernet MAC address assigned at the factory.
100 BASE-FX Connector	Provides an MT-RJ socket for connection to a 100 megabit fiber optic Ethernet cable.
10/100BASE-T Connector	Provides an RJ-45 socket for connection to a shielded, twisted pair Ethernet cable. Connects to both 10 and 100 mbps networks.

Front View

The following figure shows the front of the NWM 100 00 Ethernet module.



1 Model number, module description, color code

2 LED display

3 IP Address writable area

4 Global address label

5 100 Base Fx MT-RJ cable connector

6 10/100 Base-T RJ-45 cable connector

7 Removable door

LED Indicators

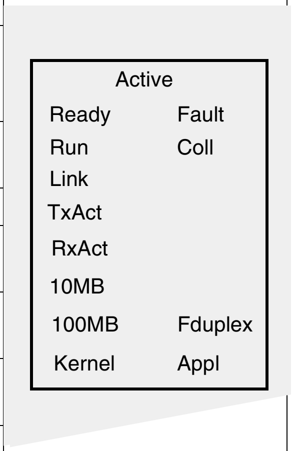
Overview

The following information describes the LED indicator panel.

LED Indicator Panel

The LED indicator panel provides continuous operating information about the 140 NWM 100 00 module and its connection to the network.
The following table describes the function of each LED indicator on the LED indicator panel.

LED	Color	Description
Active	Green	Indicates the backplane is operating.
Ready	Green	Indicates module is healthy.
Fault	Red	Indicates when the NWM is in a fault state.
Run	Green	Flashes to indicate diagnostic code, as described in "Run LED Status" (below).
Coll	Red	Flashes when Ethernet collisions occur.
Link	Green	On when Ethernet link is active.
TxAct	Green	Flashes to indicate Ethernet transmission.
RxAct	Green	Flashes to indicate Ethernet reception.
10MB	Green	On when the module is connected to a 10 Megabit network.
100MB	Green	On when the module is connected to a 100 Megabit network.
Fduplex	Green	On when Ethernet is operating in the full duplex mode.
Kernel	Amber	On when in Kernel Mode. Flashing while in download mode.
Appl	Green	On when crash log entry exists.



Run LED Status

The following table lists each available state of the *Run LED* indicator.

Indicator State	**Status**
On (steady)	Normal operation: The NWM module is ready for network communication.
Number of flashes in sequence	
one	Not used
two	Not used
three	No Link: the network cable is not connected or is defective
four	Duplicate IP address: The module will stay offline.
five	No IP address: The module is attempting to obtain an IP address from a BOOTP server.
six	Using default IP address
seven	No valid executive NWM present
eight	Invalid IP configuration (Likely cause: Default gateway is not on the same subnet mask.)
nine	Flash file system is corrupted.

8.2 140 NWM 100 00 Installation

Installing the Module

Overview

The following information describes how to install the 140 NWM 100 00 module.

Before You Begin

Locate the backplane where you will mount the140 NWM 100 00 module. Ensure that an open slot is available to mount the module.

Note: The 140 NWM 100 00 module can be installed only in a local backplane.

Note: Power requirements

- When installing the 140 NWM 100 00, ensure that it does not exceed the Quantum backplane requirements. See the specifications section.

Backplane Slot Placement

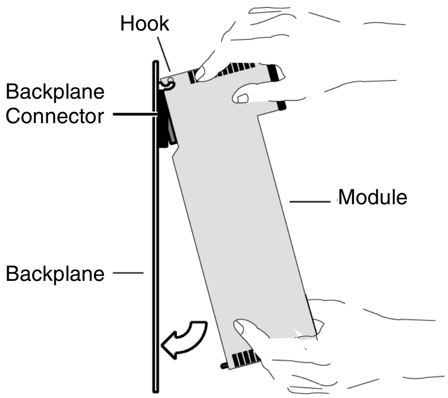
The modules may be placed in any slot on the backplane. They do not have to be placed next to each other.

Tools Required

You will need one medium-size, Phillips-head screw driver.

Mounting the Module in the Backplane

Perform the following steps to mount the 140 NWM 100 00 module on a Quantum backplane.

Step	Action
1	Holding the module at an angle, mount it on the two hooks located near the top of the backplane. The following figure shows the correct way to hold the module. 
2	Swing the module down so its connector engages the backplane connector.
3	Using a Phillips-head screw driver, tighten the screw at the bottom of the module between 2 and 4 in-lbs or between .22 and .45 Newton meters of torque.

8.3 140 NWM 100 00 Ethernet and Communications

Purpose

Overview

This section contains information pertaining to the Ethernet and communications aspects of the 140 NWM 100 00 module.

What's in this Section?

This section contains the following topics:

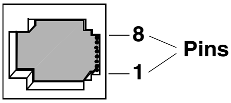
10/100 Base T and 100 BASE-FX Ethernet

Overview

The following information describes the 10/100 BASE-T and 100 BASE-FX connectors.

10/100 BASE-T Twisted Pair Connector

The 140 NWM 100 00 modules' 10/100 BASE-T connector (shown below) is a standard RJ-45 twisted pair socket.
The following figure shows the 10/100 BASE-T connector.



Schneider Electric recommends that you use Category 5 STP cabling, which is rated to 100 Mbps, with an RJ-45 connector.
The eight pins are arranged vertically and numbered in order from the bottom to the top. The RJ-45 pinout used by this module is:

- Receive Data (+)3
- Receive Data (-)6
- Transmit Data (+)1
- Transmit Data (-)2

100 BASE-FX

The140 NWM 100 00 modules' 100 BASE-FX connector consists of a MT-RJ socket, or uses a mating fiber optic cable connector.
For the140 NWM 100 00, you may need an MT-RJ to SC (Duplex) multimode fiber optic cable assembly 62.5/125mm. Schneider Electric recommends cable number 490NOC00005 to connect to fiber hubs/switches.

Note: The 140 NWM 100 00 is a one channel device. It is capable of communicating over either a 10/100BASE-T or a 100BASE-FX Ethernet network at any given time, <u>but not over both at the same time</u>.

FTP and HTTP Server

Overview

The following information describes the services provided by the FTP and HTTP Servers.

FTP Server

The NWM's File Transfer Protocol (FTP) Server is available as soon as the module receives an IP address. An FTP client can log on to the module, if the client uses the correct user name and password.

The FTP Server provides the following services.

- Update the NWM's firmware by downloading a new Exec
- Provides error log visibility by uploading error log files
- Upload/download SNMP configuration files

The default user name is USER, and the default password is USERUSER. Both the user name and password are case sensitive.

HTTP Server

The NWM's HyperText Transport Protocol (HTTP) Server is available as soon as the module receives an IP address. It can be used with version 4.0 or greater of either the Internet Explorer or Netscape browser.

The NWM HyperText Transport Protocol (HTTP) Server allows you to view the following information.

- Module's Ethernet statistics
- Controller and I/O information

The HTTP Server's HTML pages allow you to configure the module's SNMP Agent.

The HTTP Server is protected with a default name and password. The default name and password are both USER, and both are case sensitive.

For the NWM module, the user name and password can be changed via the FactoryCast Configurator.

The NWM supports a maximum of 32 HTTP simultaneous connections.

Note: Browsers may open multiple connections so 32 HTTP connections does not indicate 32 simultaneous users.

Establishing the FTP Password

Overview

The following information describes how to set the FTP Password.

Establishing the FTP Password

The FTP Password is established using the Embedded Web Server. This section contains information about initially accessing the web server. The first thing the system administrator should do upon accessing the web server is change the FTP password. Doing this restricts access to the web server functionality to the system administrator.

This section contains information on how to access the web server for purposes of changing the FTP and HTTP passwords.

Introduction to Accessing the Web Server

Each 140 NWM 100 00 module contains an Embedded Web Server, which allows you to access diagnostics and online configurations for the module and its associated controller (PLC).

For information about the additional functionality provided by the FactoryCast system in the 140 NWM 100 00 modules, see the *FactoryCast Manual*, 890 USE 152 00.

How to Access the Web Server

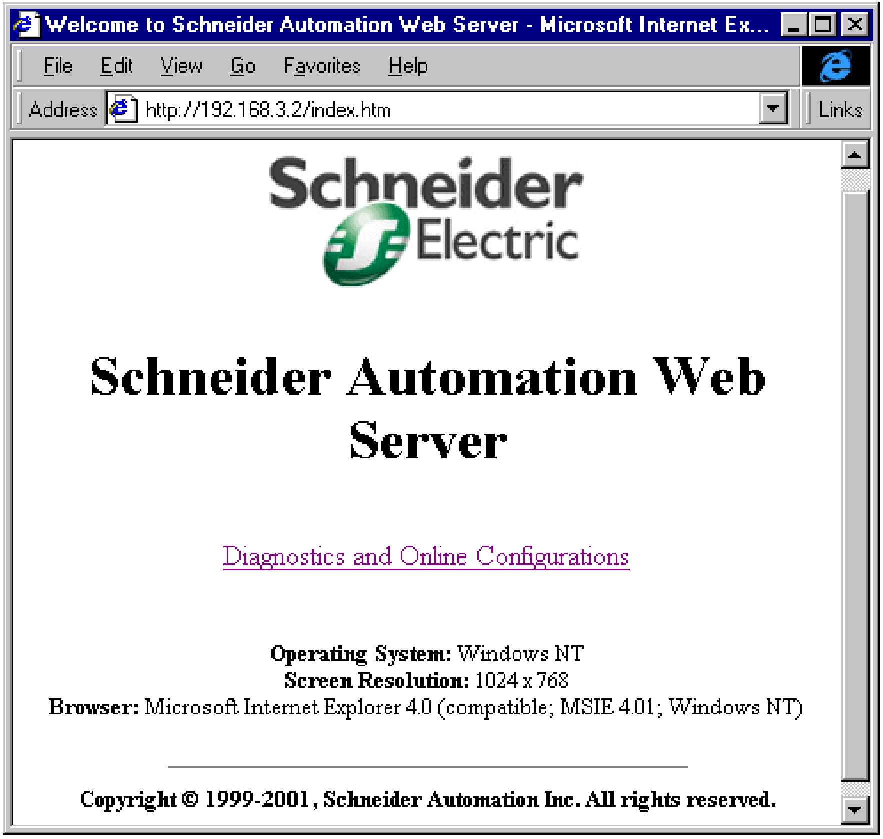
Before you can access the module's home page, you must enter the full IP address or URL in the Address or Location box in the browser window.

For example: *http://hostname* (*hostname* is full IP address or DNS host name.)

After you do this, the Schneider Automation Web Utility home page displays.

Schneider Web Utility Home Page

The following figure shows the Schneider Automation Web Utility home page



Welcome to Schneider Automation Web Server - Microsoft Internet Ex...
File
Edit
View
Go
Favorites
Help
Address
http://192.168.3.2/index.htm
Links
Schneider Electric

Schneider Automation Web Server

Diagnostics and Online Configurations

Operating System: Windows NT
Screen Resolution: 1024 x 768
Browser: Microsoft Internet Explorer 4.0 (compatible; MSIE 4.01; Windows NT)

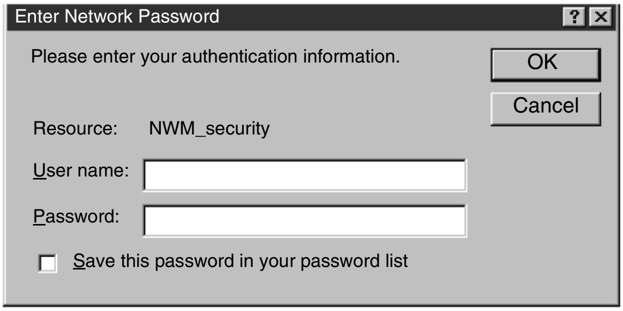
Copyright © 1999-2001 , Schneider Automation Inc. All rights reserved.

From this page, you can access the following pages.

- Access the pages to change the FTP password
- Access the pages to change the HTTP password
- Access the pages for diagnostic and configuration information.

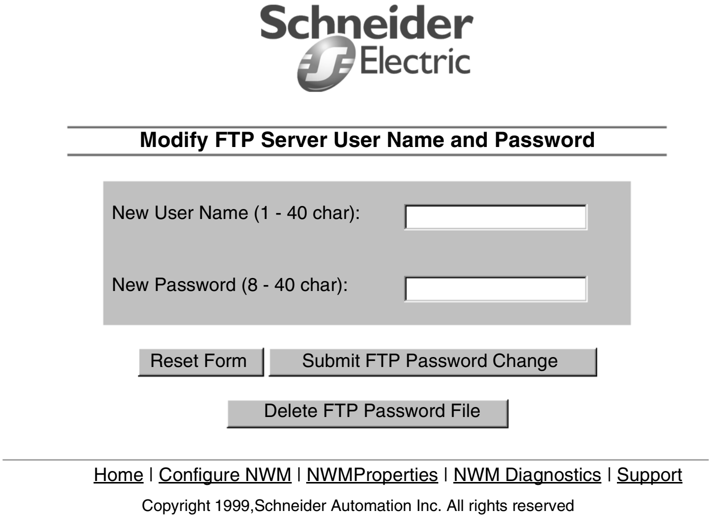
Modifying the FTP Server Password

The following steps detail how to link to the correct web page for modifying the FTP Password

Step	Action
1	Enter the URL, for example, *http://hostname/secure/embedded/ftp_passwd_config.htm*
2	The result of step 1 is that the user is requested to supply a user name and password, as shown in the figure that follows. The following figure shows the Enter Network Password dialog box.  Enter Network Password Please enter your authentication information. Resource: NWM_security User name: Password: □ Save this password in your password list
3	Upon supplying the user name, password, and clicking the <OK> button, the Modify FTP Server User Name and Password Page displays. **NOTE:** The default User Name is USER, and the default Password is USERUSER. Both should be changed by the system administrator during module installation.

FTP Username and Password Modify Page Overview

The following figure shows the page used for modifying the FTP user name and password.



Modify FTP Server User Name and Password

New User Name (1 - 40 char):

New Password (8 - 40 char):

Reset Form
Submit FTP Password Change
Delete FTP Password File
Home | Configure NWM | NWMProperties | NWM Diagnostics | Support

Copyright 1999,Schneider Automation Inc. All rights reserved

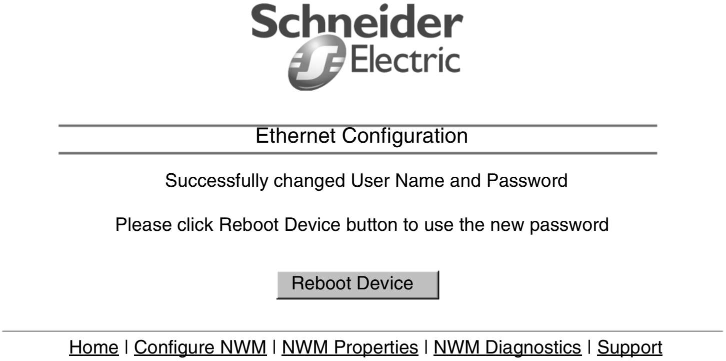
Change the Username and Password

At this point, the system administrator should change the Username and Password to restrict future access to the system. The following steps should be used.

Step	Action
1	Type in the new Username in the New User Name block
2	Type in the new Password in the New Password block
3	Click on the **Submit FTP Password Change** button.

Modify FTP Server User Name and Password Message

The following figure shows the message that is generated if you click on the Submit FTP Password Change button



Ethernet Configuration

Successfully changed User Name and Password

Please click Reboot Device button to use the new password

Schneider Electric
Reboot Device
Home | Configure NWM | NWM Properties | NWM Diagnostics | Support

Clicking the **Reboot Device** button will reset the Username and Password for the 140 NWM 100 00 board.

Note: The Reboot requires about 40 seconds (with large applications reboot may require up to one (1) minute).
During the reboot all services provided by the 140 NWM 100 00 are not available.

MODBUS Messaging

Overview

The following information describes the functionality of the MODBUS/TCP Client and Server.

Introduction - Client

The 140 NWM 100 00 Quantum Ethernet TCP/IP module provides the user with the capability of transferring data to and from nodes on a TCP/IP network through the use of a communication instruction. All PLCs that support networking communication capabilities over Ethernet can use the MSTR Ladder Logic instruction to read or write controller information or can also use IEC communication blocks.

Introduction - Server

The 140 NWM 100 00 Quantum Ethernet TCP/IP module provides the user with the ability to access data from the controller using the standard MODBUS/TCP protocol. Any device: PC, HMI package, another PLC, or any MODBUS/TCP compliant device can access data from the PLC. The MODBUS/TCP Server also allows programming panels to log into the controller over Ethernet.

Limitations

The 140 NWM 100 00 supports up to 64 simultaneous MODBUS/TCP Server connections. The 140 NWM 100 00 allows only one Programming Panel to be logged in at a time to guarantee consistency of changes to the controller configuration.
The following MODBUS/TCP commands are supported by the 140 NWM 100 00.

- Read Data
- Write Data
- Read/Write Data
- Get Remote Statistics
- Clear Remote Statistics
- MODBUS 125 Commands (used by programming panels to download a new Exec to the NWM)

Performance

The following table shows the performance characteristics of the NWM's MODBUS/TCP Server.

Parameter	Value
Typical Response Time (ms)	0.6
Number of MODBUS connections (Client and Server) There can be a maximum of 64 connections of which there can be a maximum of 16 client connections.	64
Number of simultaneous login channels	1

Note: 140 NWM 100 00 MODBUS/TCP performance measurements are made with Quantum 140 CPU 534 14 PLC.

8.4 140 NWM 100 00 Specifications

Specifications

Overview

The following information describes the main specifications for the Quantum 140 NWM 100 00 Ethernet Module.

Specification Table

The main specifications for the Quantum 140 NWM 100 00 Ethernet Module are described in the following table.

Item	Specification
Communication Ports	One auto-sensing 10/100Base-T shielded twisted pair (RJ-45 connector) port and one 100Base-FX (MT-RJ connector) port. Both ports transmit and receive Modbus commands encapsulated in TCP/IP protocol
Bus Current Required	900 mA
Power Dissipation	4.5 W
Fuse	None
Programming Software	
Type and version	Concept, Ver. 2.6
Firmware	
CPU Type and Executive version	Quantum CPU113/213 Ver 2.4 Quantum CPU113/213 Ver 2.31 (stripped) Quantum 424 Firmware Ver 2.19 Quantum 486/586 Firmware Ver 1.15 Quantum 486A/586A Firmware Ver 1.2
NWM Upgradeable	Field Upgradeable via FTP or Programming Panel.
Operating Conditions	
Temperature	0 to +60° C
Humidity	0 to 95% Rh non condensing @ 60° C
Altitude	6500 ft (2000 m)
Vibration	10-57 Hz @ 0.0075 mm d.a
	57-150 Hz @ 1 g
Processor	
Processor Speed	80 MHz
Expanded RAM size	32 MB

Item	**Specification**
Flash	16 MB)
Storage Conditions	
Temperature	-40 to +85°C
Humidity	0 to 95% Rh non condensing @ 60°C
Free Fall	1 m unpackaged
Shock	3 shocks / axis, 15 g, 11 ms
International Standard	ANSI/IEEE Std
US Standard	UL508, CEI 1131-2 for Immunity and 50082-1 for Emissions.
Canadian Standard	CSA C22.2 / 142
European Standard	Conformity to FCC-B for Emission (50082-1 CE Mark IEC 61131-2, EN66631-2
Agency Approvals	UL 508, 94 Specifications CSA 22.2-142 IEC 1131-2 CE Factory Mutual Class 1 Division 2

Configuration of HMI modules using Unity Pro, Concept and PL7

III

At a Glance

Subject of this Part

This part describes configuration of Quantum and Premium modules using Unity Pro, Concept and PL7 software workshops.

What's in this Part?

This part contains the following chapters:

Configuration of the TSX WMY 100 and 140 NWM 100 00 modules using Unity Pro

9

At a Glance

Subject of this Chapter

This chapter describes installation of the Premium **TSX WMY 100** and Quantum **140 NWM 100 00** HMI modules in the Unity Pro software workshop.

Note: The IO scanning, address server, and Global Data services can be configured using Unity Pro. But project generation will fail because these services are not available on the HMI TSX WMY 100 and 140 NWM 100 00 modules.

Note: Configuration of the HMI TSX WMY 100 and 140 NWM 100 00 modules is identical in Unity Pro.

What's in this Chapter?

This chapter contains the following sections:

9.1 General

At a Glance

Subject of this Section

This section introduces ETHERNET communication from a Premium TSX WMY 100 module or a Quantum 140 NWM 100 00 module.

What's in this Section?

This section contains the following topics:

General

At a Glance

The communication channel of the TSX WMY 100 or the 140 NWM 100 00 module offers connection to a TCP/IP network supporting the UNI-TE and Modbus messaging service on a TCP/IP profile.

The module also offers the following services:

- Thanks to their SNMP agent functionality, the modules can be supervised by one or two SNMP managers.
- They are able to perform the role of DHCP and BOOTP client.
- They integrate an embedded HTTP server.
- The Replace faulty device service, which allows the device configuration to be saved in the module. In case of breakdown, another installed blank module can be restarted with the configuration parameters of the previous one.

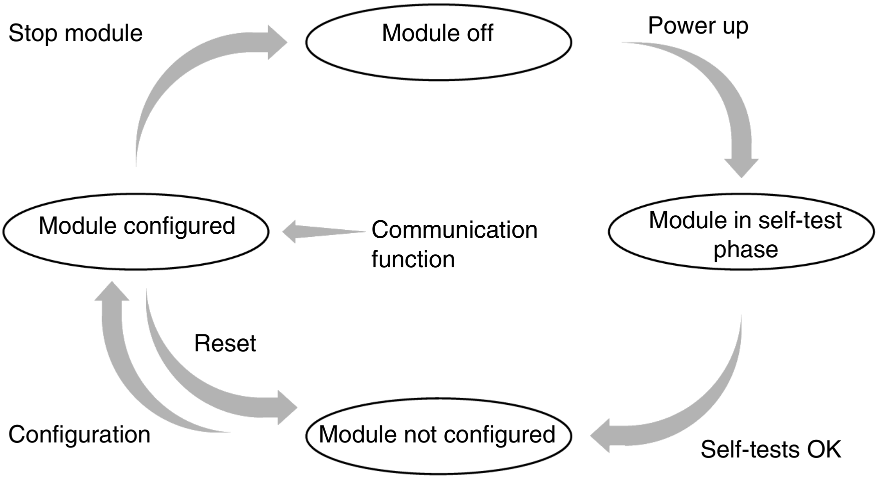
The module offers the option of creating user Web and TCP Open pages.

Operating modes of the HMI modules

At a Glance

The following diagram describes the operating modes of the Premium and Quantum HMI modules.

General Diagram



Operation

- After power-up, the module carries out self-testing. During this phase, the STS LED is lit. After the self tests, the RUN LED lights up.
- The module does not operate with a default configuration. It must be sent this configuration by the application of the local PLC.
  The configuration values are given in the list of language objects %KW and in particular provide the IP address and the X-Way address of the module.
  The configuration is retransmitted upon every PLC restart (warm or cold).
- When the configuration is received, the module resets the current communication to zero before configuring itself (terminates current exchanges, shuts down TCP connections).
  The module is now operational. The RUN and STS LEDs are lit up.

Special Cases

If a module is not configured in the Unity Pro application (RUN LEDs go out and ERR LEDs flash), it takes the IP address constructed from its MAC address: 085.016.xxx.yyy where xxx and yyy are the last two numbers of the MAC address.

Example

The module’s MAC address is (in hexadecimal): 00 80 F4 01 12 20
In this case the default IP address is (in decimal): 085.016.018.032

The module also transmits BOOTP/DHCP requests to obtain another IP address. These requests are sent until a BOOTP/DHCP server responds or until configuration by Unity Pro.

If the module detects a duplicated IP address, the STS LED flashes 4 times during a minute (as does that of the module having the same IP address).

9.2 Configuration of TSX WMY 100 & 140 NWM 100 modules running Unity Pro

At a Glance

Subject of this Section

This section describes configuration of the TSX WMY 100 and 140 NWM 100 00 modules running Unity Pro.

What's in this Section?

This section contains the following topics:

Module configuration screen

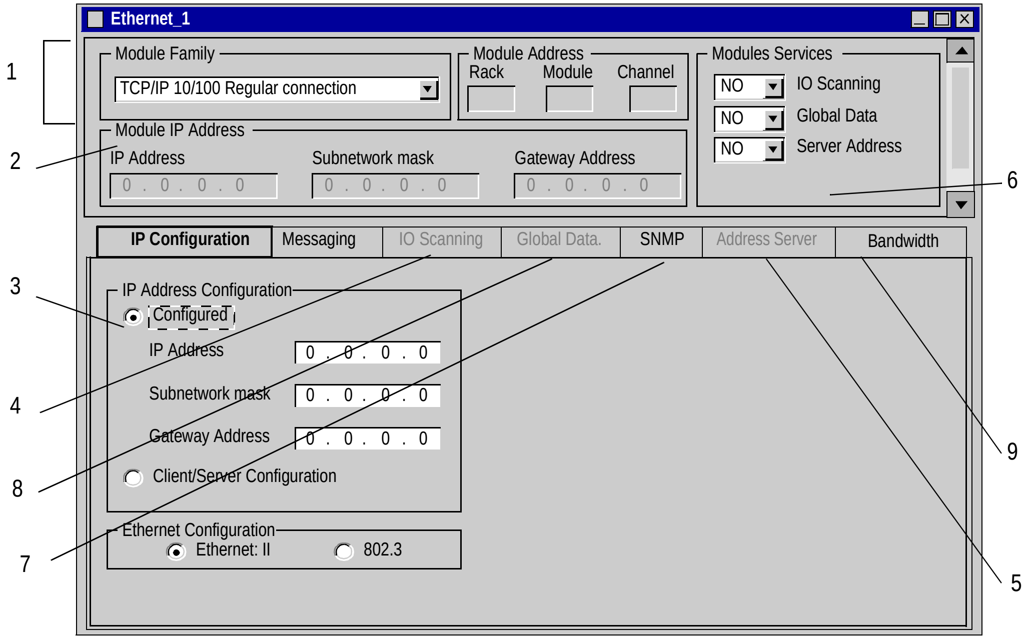
At a Glance

This screen, separated into two zones, is used to declare the communication channel and to configure the necessary parameters for an Ethernet link.

Note: Configuration of the HMI TSX WMY 100 and 140 NWM 100 00 modules is identical in Unity Pro.

Illustration

The screen dedicated to Ethernet communication is displayed as follows:



Ethernet_1

Module Family: TCP/IP 10/100 Regular connection

Module Address: Rack, Module, Channel

Modules Services:
- NO IO Scanning
- NO Global Data
- NO Server Address

Module IP Address:
- IP Address: 0 . 0 . 0 . 0
- Subnetwork mask: 0 . 0 . 0 . 0
- Gateway Address: 0 . 0 . 0 . 0

Tabs: IP Configuration | Messaging | IO Scanning | Global Data. | SNMP | Address Server | Bandwidth

IP Address Configuration:
- ● Configured
  - IP Address: 0 . 0 . 0 . 0
  - Subnetwork mask: 0 . 0 . 0 . 0
  - Gateway Address: 0 . 0 . 0 . 0
- ○ Client/Server Configuration

Ethernet Configuration: ● Ethernet: II ○ 802.3

Elements and Functions

This table describes the various zones that make up the configuration screen:

Zone	Number	Function
common	1	common part of the communication configuration screens.
specific	2	allows the display of the module address.
	3	allows configuration of the TCP/IP services. See *Configuration Parameters Linked to TCP/IP Services, p. 345*.
	4	IO scanning service, see note below.
	5	Address server service, see note below.
	6	allows selection of the services used by the module.
	7	allows configuration of the SNMP service. See *Configuration Parameters Linked to the SNMP Service, p. 352*.
	8	Global Data service, see note below.
	9	allows confirmation that the services configured are compatible with the Ethernet channel processing capacity.

Note: The IO scanning, address server and Global Data services can be configured using Unity Pro. But project generation will fail because these services are not available on the HMI TSX WMY 100 and 140 NWM 100 00 modules.

Type of Communication According to Connection Configuration

At a Glance

Depending on the connection configuration of the HMI modules, you can carry out messaging:

- in mono-connection
- in multi-connection.

According to the type of protocol and access control management, multi-connection mode requires a specific configuration of the correspondence table.

Note: In the following, only examples of multi-connection will be dealt with. Furthermore, the emphasis is placed on communication between the module and a remote PC device containing several applications.

Configuration Rules

Several connections can be configured with the same IP address. They must be defined with the same protocol, the same access rights, and the same connection mode.

In the case of a Modbus connection, only two connections can be configured with the same IP address. Addresses associated with these connections are:

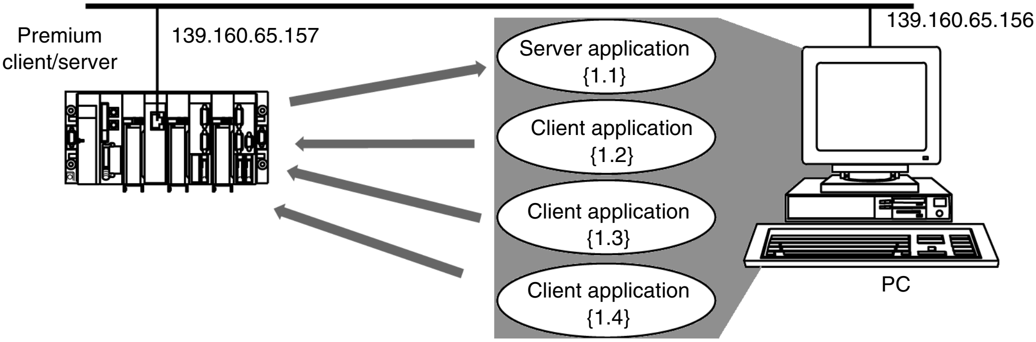
- an X-Way address (for example: {2.103}) for the Modbus client connection,
- an X-Way address equal to {x.x} for Modbus server connections.

Multi-connection in UNI-TE Protocol with Access Management

In this example, the PC contains four applications with a unique X-Way address and one single IP address:

- The module opens a connection and communicates with the server application with the address {1.1}.
- Each of the other client applications with the address {1.2}, {1.3}, {1.4} is authorized to open a connection and communicate with the module.

Exchanges are carried out according to the UNI-TE protocol and the module's access control service is enabled.



In order to implement this example, you must:

- activate connection management by checking the module's **Access control** box,
- configure communication from the module to the PC's server application in the table
  - **X-Way address**: 1.1
  - **IP address**: 139.160.65.156
  - **Protocol**: UNI-TE
  - **Access**: authorized (cell checked)
  - **Mode**: Multi
- configure communication of each of the PC's client applications to the server module in the table:
  - **X-Way address**: 1.2 and 1.3 and 1.4
  - **IP address**: 139.160.65.156
  - **Protocol**: UNI-TE
  - **Access**: authorized (cell checked)
  - **Mode**: Multi

Multi-connection in UNI-TE protocol without access management

This example is the same as the previous one, except that access control is not configured.

In this case, if the application {1.1} is the only server application, the module's correspondence table only contains the following line:

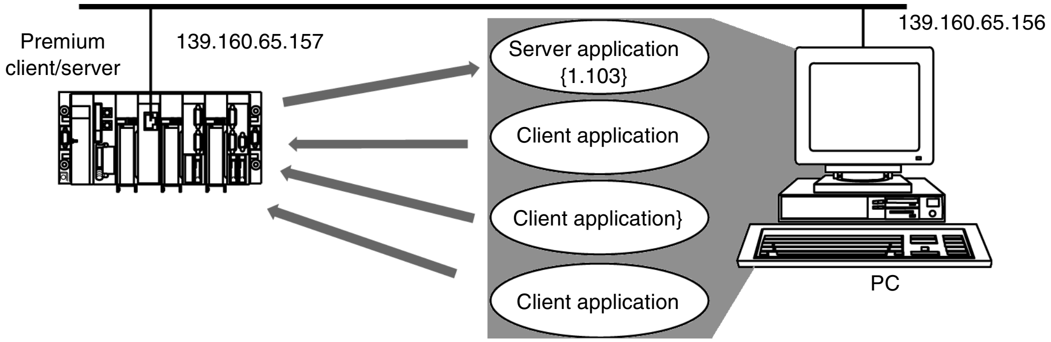
- **X-Way address**: 1.1
- **IP address**: 139.160.65.156
- **Protocol**: UNI-TE
- **Access**: the cell is grayed out
- **Mode**: Multi

Multi-connection in Modbus Protocol with Access Management

In this example, the PC contains four applications. The server application has an imaginary X-Way address and the other client applications do not need an X-WAY address:

- The module opens a connection and communicates with the server application with the imaginary address {1.103}.
- Each of the client applications has no address but is authorized to open a connection and communicates with the module.

Exchanges are carried out according to the Modbus protocol and the module's access control service is enabled.



In order to implement this example, you must:

- activate connection management by checking the module's **Access control** box,
- configure communication from the module to the PC's server application in the table
  - **X-Way address**: 1.103 (address between 100 and 163)
  - **IP address**: 139.160.65.156
  - **Protocol**: Modbus
  - **Access**: the cell is grayed out
  - **Mode**: Multi

- configure communication of all the PC's client applications to the server module in the table:
  - **X-Way address**: x.x (this address allows you to specify that other applications exist for the same IP address)
  - **IP address**: 139.160.65.156
  - **Protocol**: Modbus
  - **Access**: authorized (allows you to specify that other applications are authorized to connect and communicate)
  - **Mode**: Multi

Note: If you want to prevent connection by client applications you must configure
- **X-Way address**: x.x (this address allows you to specify that other applications exist for the same IP address)
- **IP address**: 139.160.65.156
- **Protocol**: Modbus
- **Access**: not authorized (cell unchecked)
- **Mode**: Multi

Multi-connection in Modbus protocol without access management

This example is the same as the previous one, except that access control is not configured.

In this case, if the application {1.103} is the only server application, the module's correspondence table only contains the following line:
- **X-Way address**: 1.103
- **IP address**: 139.160.65.156
- **Protocol**: Modbus
- **Access**: the cell is grayed out
- **Mode**: Multi

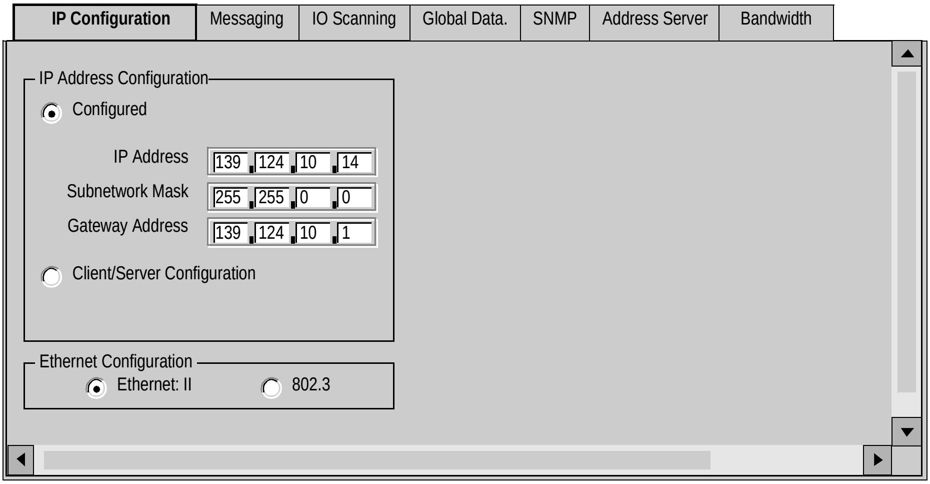
Configuration of TCP/IP Messaging

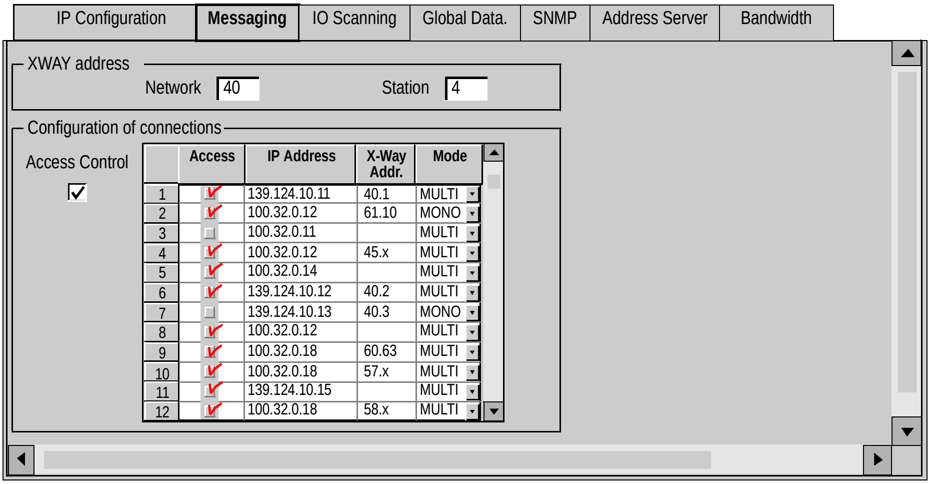
At a Glance

In order to use the TSX WMY 100 and 140 NWM 100 00 modules to communicate on Ethernet, it is necessary to adjust the messaging configuration parameters. Messaging is configured in the two windows accessible by the tabs: **IP Configuration** and **Messaging**.

How to Access the Messaging

The procedure for accessing the configuration parameters for TCP/IP services is as follows.

Step	Action
1	Access the module configuration screen.
2	Select the **IP Configuration** tab. **Result**  IP Configuration Messaging IO Scanning Global Data. SNMP Address Server Bandwidth IP Address Configuration ● Configured IP Address: 139.124.10.14 Subnetwork Mask: 255.255.0.0 Gateway Address: 139.124.10.1 ○ Client/Server Configuration Ethernet Configuration ● Ethernet: II ○ 802.3

Step	Action
3	Select the **Messaging** tab. **Result** 

XWAY address

Network 40 Station 4

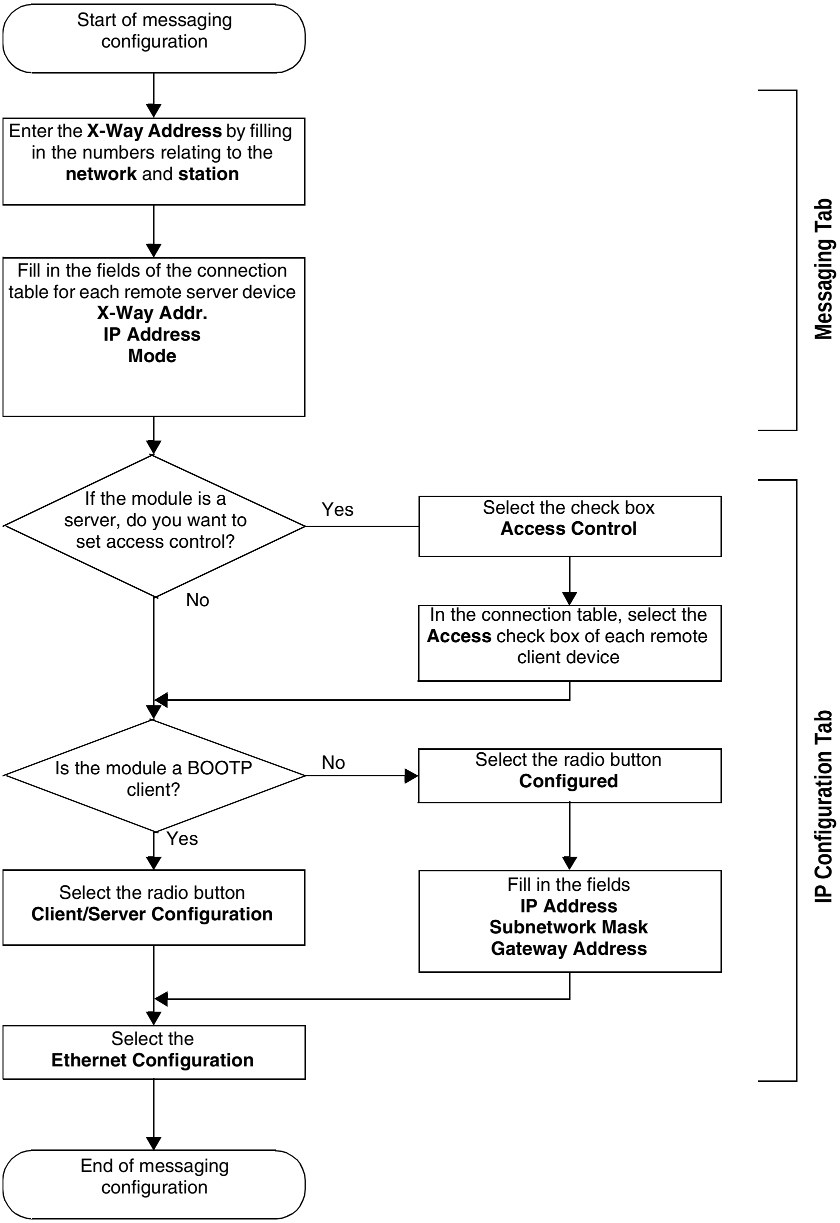
Configuration of connections

Access Control ☑

	Access	IP Address	X-Way Addr.	Mode
1	✓	139.124.10.11	40.1	MULTI
2	✓	100.32.0.12	61.10	MONO
3	□	100.32.0.11		MULTI
4	✓	100.32.0.12	45.x	MULTI
5	✓	100.32.0.14		MULTI
6	✓	139.124.10.12	40.2	MULTI
7	□	139.124.10.13	40.3	MONO
8	✓	100.32.0.12		MULTI
9	✓	100.32.0.18	60.63	MULTI
10	✓	100.32.0.18	57.x	MULTI
11	✓	139.124.10.15		MULTI
12	✓	100.32.0.18	58.x	MULTI

How to Configure the Messaging

The following procedure gives the configuration principle.


Start of messaging configuration
Enter the X-Way Address by filling in the numbers relating to the network and station
Fill in the fields of the connection table for each remote server device X-Way Addr. IP Address Mode
If the module is a server, do you want to set access control?
Yes
Select the check box Access Control
In the connection table, select the Access check box of each remote client device
No
Is the module a BOOTP client?
No
Select the radio button Configured
Fill in the fields IP Address Subnetwork Mask Gateway Address
Yes
Select the radio button Client/Server Configuration
Select the Ethernet Configuration
End of messaging configuration
Messaging Tab
IP Configuration Tab

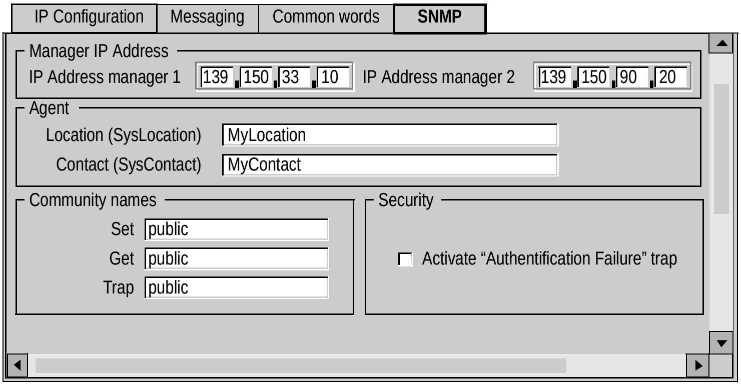
Configuration of the SNMP Service

At a Glance

In order to use the TSX WMY 100 and 140 NWM 100 00 module as an SNMP agent, it is necessary to adjust the configuration parameters of the SNMP service.

How to Access the SNMP Service

The procedure for accessing the configuration parameters of the SNMP service is as follows.

Step	Action
1	Access the module configuration screen.
2	Select the **SNMP** tab. **Result**  Manager IP Address IP Address manager 1: 139.150.33.10 IP Address manager 2: 139.150.90.20 Agent Location (SysLocation): MyLocation Contact (SysContact): MyContact Community names Set: public Get: public Trap: public Security ☐ Activate “Authentification Failure” trap

How to Configure SNMP

The following procedure gives the configuration principle for the SNMP service.

Step	Action
1	Enter the SNMP manager addresses: ● **Manager 1 IP addresses** ● **Manager 2 IP addresses**
2	Fill in the fields: ● **Location (SysLocation)** ● **Contact (SysLocation)**. Or alternatively check the **SNMP Manager** box to indicate that the information will be completed by the SNMP Manager.
3	If you want to define access rights, fill in the community names: ● **Set** ● **Get** ● **Trap**
4	If you want to activate transmission of an event to the module, check the **Activate "Authentication Failure" trap** box.

Configuration of the Bridge Function

At a Glance

The TSX WMY 100 and 140 NWM 100 00 modules may be used as an X-WAY bridge station (See *Configuration of Properties, p. 29*). This guarantees transparent communication between various networks.

9.3 Debugging HMI modules in Unity Pro

At a Glance

Subject of this Section

This section provides information about debugging Premium and Quantum HMI modules in the Unity Pro software workshop.

What's in this Section?

This section contains the following topics:

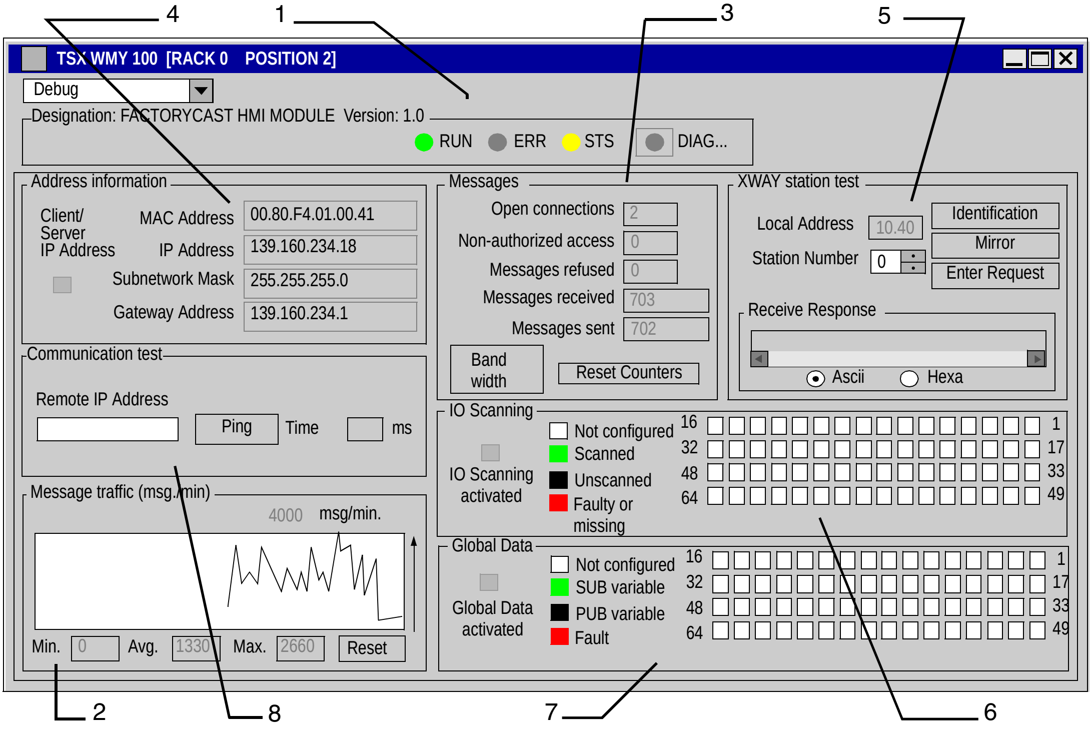
Module debug screen

At a Glance

This screen, separated into eight zones, allows debugging via an Ethernet link.

Illustration

The screen dedicated to Ethernet communication is displayed as follows:



TSX WMY 100 [RACK 0 POSITION 2]

Debug

Designation: FACTORYCAST HMI MODULE Version: 1.0

Address information

Client/ Server IP Address

MAC Address 00.80.F4.01.00.41

IP Address 139.160.234.18

Subnetwork Mask 255.255.255.0

Gateway Address 139.160.234.1

Messages

Open connections 2

Non-authorized access 0

Messages refused 0

Messages received 703

Messages sent 702

XWAY station test

Local Address 10.40

Station Number 0

Receive Response

● Ascii ○ Hexa

Communication test

Remote IP Address

Time ms

IO Scanning

Message traffic (msg./min)

4000 msg/min.

Min. 0 Avg. 1330 Max. 2660

Global Data

Elements and Functions

This table describes the different zones that make up the debug screen:

Number	Zone	Function
1	common	common part of the communication debug screens
2	Message traffic	allows the graphical display of the number of messages processed by the module:
3	Messages	allows the display of the number of connections and unacknowledged or refused messages. The counter values can be reinitialized using the **Reset Counters** button. A **Bandwidth** button is used to access bandwidth diagnostics.
4	TCP/IP services	allows: ● display of TCP/IP services configuration, ● communication testing of the TCP/IP profile.
5	X-WAY test station	allows UNI-TE communication testing on the TCP/IP profile.
6	IO Scanning	unavailable.
7	Global Data.	unavailable.
8	Communication test	is used to perform a communication test

General Debugging Parameters

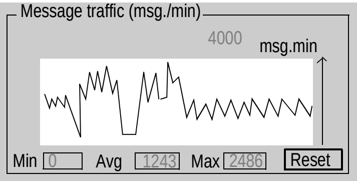
At a Glance

The general debugging parameters are grouped together in two windows:

- the **Message traffic** window
- the **Messages** window.

Message traffic

The window is displayed as below:



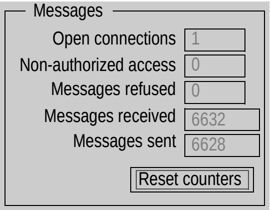
Min 0 Avg 1243 Max 2486

It shows the number of messages processed by the module per minute (transmission and reception) in graphic form. To provide a better display, the scale is automatically adapted according to change in flow.

The **Reset** button sets the three counters **Min.**, **Avg.** and **Max.** to zero.

Messages

The window is displayed as below:



Open connections 1
Non-authorized access 0
Messages refused 0
Messages received 6632
Messages sent 6628

This window is related to TCP/IP messaging:

- number of TCP connections open
- number of non-authorized accesses
- number of messages refused, received and sent

You can reset these counters to zero by clicking on the **Reset counter** button.

Debugging parameters for TCP/IP services

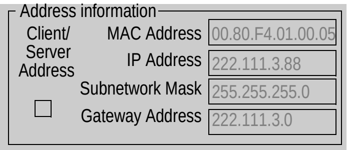
At a Glance

The debugging parameters for the TCP/IP services are grouped together in two windows:

- the **Address information** window,
- the **Communication test** window.

Address information

The window is displayed as below:



Address information

Client/ Server Address □

MAC Address: 00.80.F4.01.00.05

IP Address: 222.111.3.88

Subnetwork Mask: 255.255.255.0

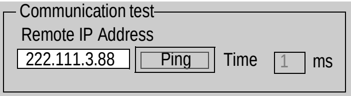
Gateway Address: 222.111.3.0

It specifies the configuration data of the IP address:

- **MAC address**: unique fixed address for a module
- **IP Address**
- **Subnetwork mask**
- **Gateway Address**: address of the gateway

Communication test

Once the **Ping** button has been pressed, the window is displayed as follows:



Communication test

Remote IP Address: 222.111.3.88

Time: 1 ms

This window is used to test IP communication towards another station.

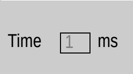
How to Test TCP/IP communication with the Ping request

At a Glance

This page indicates the procedure for testing TCP/IP communication from a **Ping** request.

Procedure

The following procedure is used to send the Ping request and therefore to test the correct routing of information between two devices.

Step	Action
1	From the main screen, click on the **Ping** button.
2	Enter the address of the station to be queried using the **Remote IP address** field.
3	Press the **Ping** button. **Result** The response appears in the **Time** field:  Time 1 ms The returned time corresponds to the time it takes the message to be sent and return in ms.

Response type

The following table groups together the various types of possible responses to the Ping request.

If the response is	then
positive	The windows records the time it takes the message to be sent and return in ms. a window appears with the message **Exchange successful**.
negative	a window with the message **Exchange incorrect** specifies that the remote device was not reached in the network architecture.

Available requests for testing a communication channel

At a Glance

This page describes the various options for testing a communication channel from the debug screen.

Test conditions

When a request is sent to a station that is not connected, an error message appears. The test applies to stations belonging to the same local area network.

Available requests

The **X-Way station test** window allows the following requests:

- **Identification**: causes the Identification request to be sent to the designated remote station,
- **Mirror**: is used to send a Mirror request to the designated station. If this function is chosen, a screen appears allowing you to select the length of the string of characters to be sent (maximum of 80 characters). The PLC then sends this string of characters (ABCD, etc.) to the destination device. This device automatically returns the string of characters received to the sender,
- **Enter Request** : is used to send a UNI-TE request, other than those sent by the command buttons, to the designated station. If this function is chosen, a screen appears allowing you to select the parameters specific to the request (the request code must be coded in hexadecimal).

Note: The first two requests are sent to the UNITE server of the TSX WMY 100 module. The third is sent to the server of the processor.

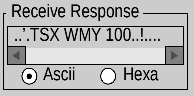
How to Test a Channel with the Identification and Mirror requests

At a Glance

This page indicates the procedure for testing a communication channel using the Identification and Mirror requests.

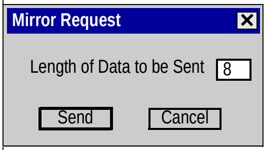
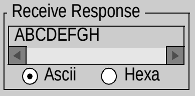
How to Identify a Station

The following procedure is used to identify a designated station.

Step	Actions
1	Select the address of the station to be queried using the **Station number** field.
2	Press the **Identification** button. **Result** The response appears in the **Receive Response** window:  ..’.TSX WMY 100..!.... ● Ascii ○ Hexa

How to Send the Mirror Request

The following procedure is used to send the Mirror request and therefore to test the correct routing of information between two devices.

Step	Action
1	Select the address of the station to be queried using the **Station** field.
2	Press the **Mirror** button. **Result** The following window appears:  Mirror Request Length of Data to be Sent 8
3	Enter the length of the data to be sent (maximum 80 characters).
4	Press the **Send** button. **Result** The response appears in the **Receive Response** window:  Receive Response ABCDEFGH ● Ascii ○ Hexa The response contains: - the character string ABCDEFGH corresponding to a length of data sent of 8.

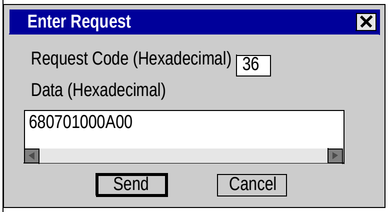
How to test a channel with requests

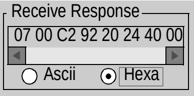
At a Glance

This page indicates the procedure for testing a communication channel from the debug screen through different requests.

How to Send a Request

The following procedure is used to send a request, other than those sent by the command buttons, to a designated station. In this example, the sent request enables 10 words to be read (from %MW1 to %MW10).

Step	Action
1	Select the address of the station to be queried using the **Station** field.
2	Press the **Enter Request** button. **Result** The following window appears:  Enter Request Request Code (Hexadecimal) 36 Data (Hexadecimal) 680701000A00 The data transmitted in this example is coded on 6 bytes.
3	Enter the function code (coded in hexadecimal on a byte), corresponding to the request that you want to send. In this example, the read request code is 16#36.
4	Enter the data to be transmitted by coding each data item in hexadecimal. The data is then entered one after the other with no spaces between data items. When the data is coded on a word, the most significant bytes and the least significant bytes are inverted. In this example, the data is as follows: ● 16#68: on a byte, define the segment (internal data), ● 16#07: on a byte, define the type of object (words), ● 16#0100: on a word, define the first word to be read, ● 16#0A00: on a word, define the number of words to be read.

Step	Action
5	Press the **Send** button. **Result** The response appears in the **Receive Response** window:  Receive Response 07 00 C2 92 20 24 40 00 ○ Ascii ● Hexa The response for the example contains the data on 21 bytes. ● 16#07: corresponds to the type of object (words), ● 16#00C2: corresponds to the value of the first word (the most significant bytes and the least significant bytes are inverted; its value is 16#C200), ● ...

9.4 Language objects associated with HMI modules

At a Glance

Subject of this Section

This section describes the language objects associated with the Premium and Quantum HMI modules, as well as the IODDTs.

What's in this Section?

This section contains the following topics:

Creating an IODDT Type Data Instance

At a Glance

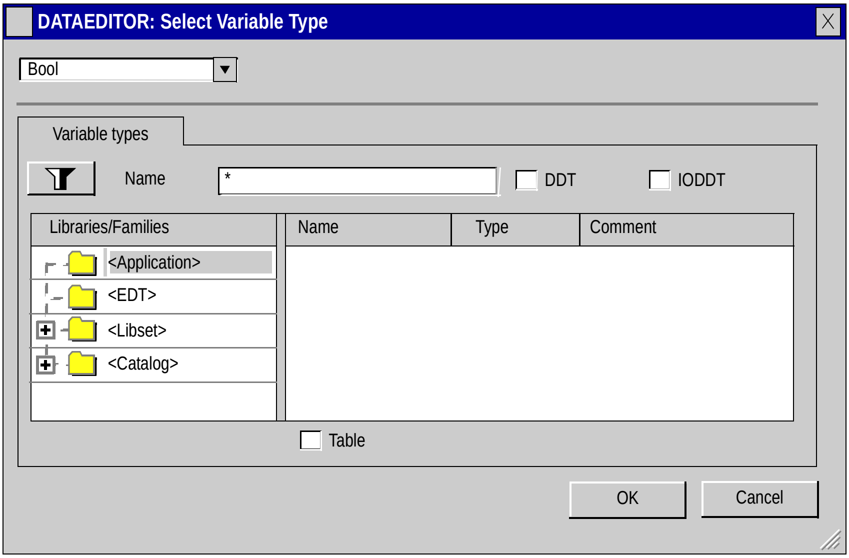
In the software installation principle the following must be carried out in order:

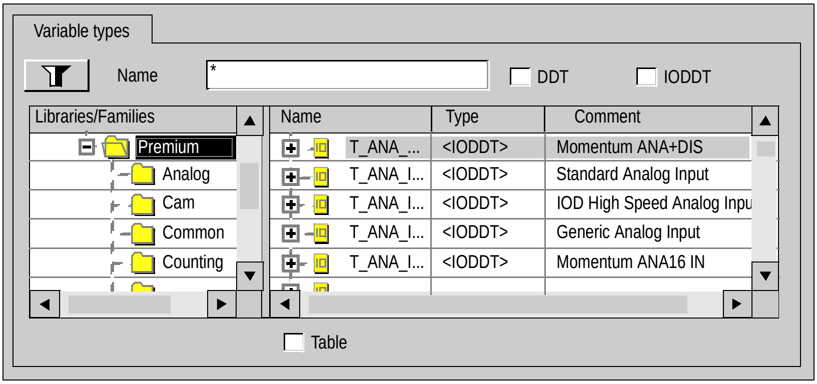
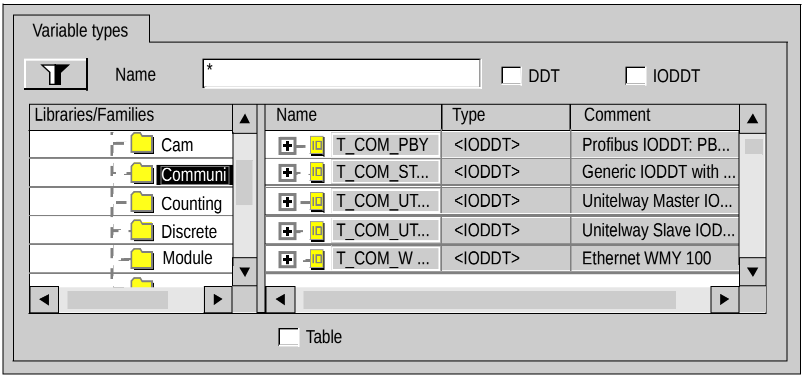
- create an IODDT type instance,
- associate the IODDT instance with the module,
- generate the project.

The following examples present the creation and association with a channel of an IODDT instance of the type **T_COM_WMY_100**.

Creating an IODDT Type Instance

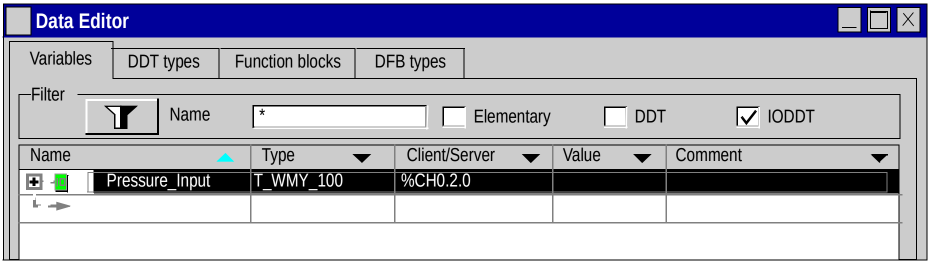
To create an IODDT type instance, carry out the following actions:

Step	Action
1	Using the variable editor, select the **Variables** tab.
2	Double-click on an empty **Name** cell, and enter the name of the instance.
3	Double-click on the corresponding **Type** cell, and press on the button ... . The following screen is displayed:  DATAEDITOR: Select Variable Type Bool Name: * □ DDT □ IODDT Variable types Libraries/Families <Application> <EDT> <Libset> <Catalog> Name Type Comment □ Table OK Cancel

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>In the zone <b>Libraries/Families</b> deploy the <b>Premium</b> subdirectory of the <b>Catalog</b> directory.
<b>Result</b>: the following screen is displayed.

Variable types
Name * ☐ DDT ☐ IODDT
<table>
<tr><th>Libraries/Families</th></tr>
<tr><td>Premium</td></tr>
<tr><td>Analog</td></tr>
<tr><td>Cam</td></tr>
<tr><td>Common</td></tr>
<tr><td>Counting</td></tr>
</table>
<table>
<tr><th>Name</th><th>Type</th><th>Comment</th></tr>
<tr><td>T_ANA_...</td><td><IODDT></td><td>Momentum ANA+DIS</td></tr>
<tr><td>T_ANA_I...</td><td><IODDT></td><td>Standard Analog Input</td></tr>
<tr><td>T_ANA_I...</td><td><IODDT></td><td>IOD High Speed Analog Inpu</td></tr>
<tr><td>T_ANA_I...</td><td><IODDT></td><td>Generic Analog Input</td></tr>
<tr><td>T_ANA_I...</td><td><IODDT></td><td>Momentum ANA16 IN</td></tr>
</table>
☐ Table</td></tr>
<tr><td>5</td><td>Select the <b>Module</b> directory in the zone <b>Libraries/Families</b>.
<b>Result</b>: the IODDT type appears.

Variable types
Name * ☐ DDT ☐ IODDT
<table>
<tr><th>Libraries/Families</th></tr>
<tr><td>Cam</td></tr>
<tr><td>Communi</td></tr>
<tr><td>Counting</td></tr>
<tr><td>Discrete</td></tr>
<tr><td>Module</td></tr>
</table>
<table>
<tr><th>Name</th><th>Type</th><th>Comment</th></tr>
<tr><td>T_COM_PBY</td><td><IODDT></td><td>Profibus IODDT: PB...</td></tr>
<tr><td>T_COM_ST...</td><td><IODDT></td><td>Generic IODDT with ...</td></tr>
<tr><td>T_COM_UT...</td><td><IODDT></td><td>Unitelway Master IO...</td></tr>
<tr><td>T_COM_UT...</td><td><IODDT></td><td>Unitelway Slave IOD...</td></tr>
<tr><td>T_COM_W ...</td><td><IODDT></td><td>Ethernet WMY 100</td></tr>
</table>
☐ Table</td></tr>
<tr><td></td><td>In the <b>Type</b> column, select the <b>IODDT type</b> desired (in our example T_WMY_100).
Validate with <b>OK</b>.</td></tr>
</table>

IODDT instance link with the application-specific module

To link an IODDT type instance to the channel of an application-specific module, carry out the following actions:

Step	Action
1	Using the variable editor, select the **Variables** tab.
2	Check the **IO Derived** box, so that only IODDT type instances are displayed.
3	Select the IODDT instance from which the link is to be created.
4	In the **Address** column, enter the address corresponding to the module or to its future slot (the module need not be configured at this stage). Example:  Data Editor Variables | DDT types | Function blocks | DFB types Filter: Name * ☐ Elementary ☐ DDT ☑ IODDT Name | Type | Client/Server | Value | Comment Pressure_Input | T_WMY_100 | %CH0.2.0 | |

Rules to observe

The rules are as follows:

- an IODDT data type cannot be nested in a DDT data type,
- an IODDT data type cannot be nested in another IODDT data type,
- the public or private variable of a DFB cannot be of the type IODDT,
- the input/output parameter of a DFB can be of the type IODDT,
- it is not possible to access an IODDT type instance from a DFB section.

Implicit Exchange Language Objects Associated with the Application-Specific Function

At a Glance

An integrated application-specific interface or the addition of a module automatically enhances the language objects application used to program this interface or module.

These objects correspond to the input/output images and software data of the module or integrated application-specific interface.

Reminders

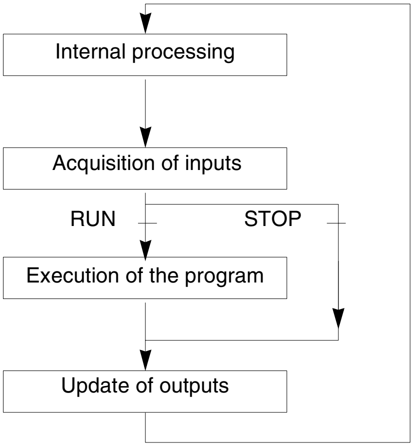
The module inputs (%I and %IW) are updated in the PLC memory at the start of the task, the PLC being in RUN or STOP mode.

The outputs (%Q and %QW) are updated at the end of the task, only when the PLC is in RUN mode.

Note: When the task occurs in STOP mode, either of the following are possible, depending on the configuration selected:
- Outputs are set to fallback position (fallback mode),
- Outputs are maintained at their last value (maintain mode).

Illustration

The following diagram shows the operating cycle of a PLC task (cyclical execution).



Explicit exchange language objects associated with the application-specific function

At a Glance

Explicit exchanges are exchanges performed at the user program's request, and using instructions:

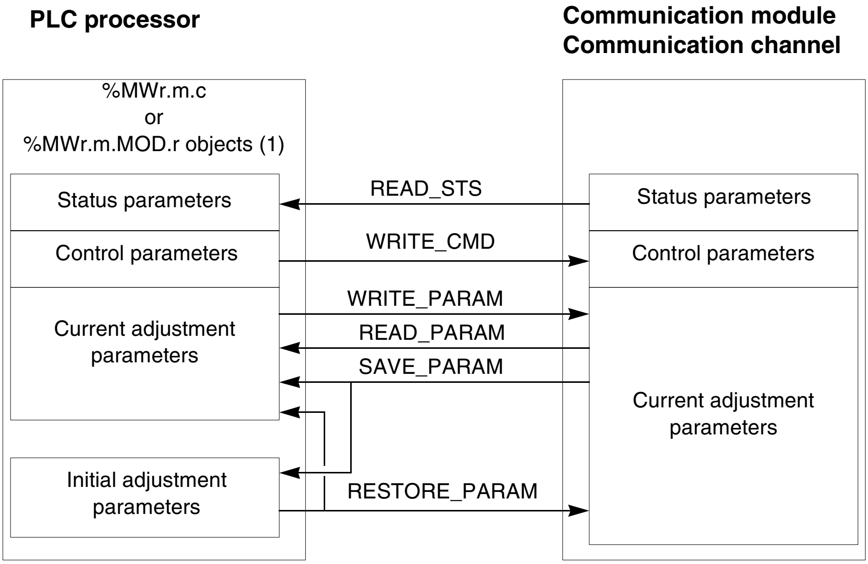
- READ_STS (read status words),
- WRITE_CMD (write command words),
- WRITE_PARAM (write adjustment parameters),
- READ_PARAM (read adjustment parameters),
- SAVE_PARAM (save adjustment parameters),
- RESTORE_PARAM (restore adjustment parameters).

These exchanges apply to a set of %MW objects of the same type (status, commands or parameters) that belong to a channel.

Note: These objects provide information about the module (e.g.: type of channel fault, etc.), can be used to command them (e.g.: switch command) and to define their operating modes (save and restore adjustment parameters in the process of application).

General principle for using explicit instructions

The diagram below shows the different types of explicit exchanges that can be made between the processor and module.



(1) Only with the instructions READ_STS and WRITE_CMD.

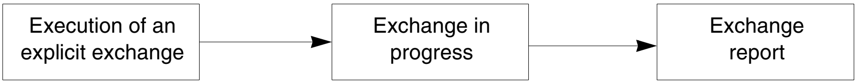
Managing exchanges

During an explicit exchange, it is necessary to check its performance in order that data is only taken into account when the exchange has been correctly executed.

To do this, two types of information is available:

- information concerning the exchange in progress,
- the exchange report.

The following diagram describes the management principle for an exchange



Language Objects Associated with Configuration

At a Glance

This page describes all the configuration language objects for Ethernet communication with the TSX ETY 4103/PORT/5103 module and the Ethernet channel of the TSX 57 564, which can be displayed by the application program.

Internal Constants

The following table describes the internal constants:

Object	Function	Meaning
%KWxy.i.0	Type	Byte 0 = 11 for Ethernet communication Byte 1: reserved
%KWxy.i.1	Physical layer	Byte 0: fixed at 16#01 (corresponds to GPX2) Byte 1: reserved
%KWxy.i.2	Reserved	-
%KWxy.i.3	Reserved	-
%KWxy.i.4	Reserved	-
%KWxy.i.5	X-Way network address	Byte 0: network number (0 by default) Byte 1: reserved
%KWxy.i.6	Type of Ethernet driver for TCP/IP	Byte 0: fixed at 16#01 (corresponds to RJ45) ● = 16#01: RJ45 Byte 1: ● = 16#00: Ethernet II (default value) ● = 16#01: 802.3
%KWxy.i.7	TCP/IP configuration: address type	Byte 0: reserved Byte 1: inherited address ● = 16#01: from the application-specific function ● = 16#03: from a server
%KWxy.i.8 and %KWxy.i.9	Local IP address	Example with the address 139.160.650.109 Byte 0 = 109 Byte 1 = 65 Byte 2 = 160 Byte 3 = 139
%KWxy.i.10 and %KWxy.i.11	IP address of gateway	Example with the address 139.160.65.1 Byte 0 = 1 Byte 1 = 65 Byte 2 = 160 Byte 3 = 139

Object	Function	Meaning
%KWxy.i.12 and %KWxy.i.13	Subnetwork mask	Example with the address 255.255.255.0 Byte 0 = 0 (least significant) Byte 1 = 255 Byte 2 = 255 Byte 3 = 255 (most significant)
%KWxy.i.14	Services used	Byte 0: ● x0 = 1: the IOScanner service is used ● x1 = 1: BOOTP server service is used ● x2 = 1: the Global Data service is used ● x3 = 1: the service to adjust bandwidth is used Byte 1: reserved
%KWxy.i.15	TCP services used	Byte 0: ● x0 = 1: Modbus communication on TCP/IP is used ● x1 = 1: access control is activated Byte 1: reserved

Details of Explicit Exchange Objects of the IODDT of type T_COM_WMY_100

At a Glance

This part introduces explicit exchange objects of the T_COM_WMY_100 IODDT which apply to the TSX WMY 100 module (there are no implicit exchange objects in this IODDT). It groups word-type objects whose bits have a particular meaning. These objects are described in detail below.

Example of declaring a variable: IODDT_VAR1 of type T_COM_WMY_100

Notes

- The meaning of a bit is generally given for the status of the bit when set to 1. In specific cases an explanation is given for each status of the bit.
- Not all bits are used.

Execution flags of an explicit exchange: EXCH_STS

The table below presents the exchange control bits of the channel EXCH_STS (%MWr.m.c.0).

Standard symbol	Type	Acces s	Meaning	Address
STS_IN_PROGR	BOOL	R	Reading of status words of the channel in progress.	%MWr.m.c.0.X0
CMD_IN_PROGR	BOOL	R	Exchange of parameters of command in progress	%MWr.m.c.0.X1
ADJ_IN_PROGR	BOOL	R	Exchange of parameters of adjustment in progress	%MWr.m.c.0.X2

Explicit exchange report: EXCH_RPT

The table below presents the meaning of the exchange report bits EXCH_RPT (%MWr.m.c.1).

Standard symbol	Type	Acces s	Meaning	Address
STS_ERR	BOOL	R	Error in reading status words of the channel	%MWr.m.c.1.X0
CMD_ERR	BOOL	R	Error when exchanging command parameters	%MWr.m.c.1.X1
ADJ_ERR	BOOL	R	Error when exchanging adjustment parameters	%MWr.m.c.1.X2

Standard channel faults, CH_FLT

The table below presents the meaning of the status word CH_FLT (%MWr.m.c.2). Reading is carried out by a READ_STS(IODDT_VAR1).

Standard symbol	Type	Acces s	Meaning	Address
INTERNAL_FLT	BOOL	R	Internal fault or channel self-test	%MWr.m.c.2.4
APPLI_FLT	BOOL	R	Application fault (adjustment or configuration fault)	%MWr.m.c.2.7

Network diagnostics

The table below shows the status words (or double words) CH_FLT (%MWr.m.c.3 to 6, %MWr.m.c.11 to 15 and %MDr.m.c.7 and 9) used for diagnosing the network. Reading is carried out by a READ_STS(IODDT_VAR1).

Standard symbol	Type	Acces s	Meaning	Address
NB_P502_CNX	INT	R	Number of Port 502 connections and bridge configuration information	%MWr.m.c.3
NB_DENIED_CNX	INT	R	Number of denied Port 502 connections)	%MWr.m.c.4
NB_P502_REF	INT	R	Number of refused messages on Port 502	%MWr.m.c.5
XWAY_ADDR	INT	R	X-Way address (Network, station)	%MWr.m.c.6
NB_SENT_MSG	DINT	R	Number of sent messages on Port 502	%MDr.m.c.7
NB_RCV_MSG	DINT	R	Number of received messages on Port 502	%MDr.m.c.9
BW_OTHER_MSG	INT	R	Loading of messaging service and other services	%MWr.m.c.15

9.5 Title of Section

Questions/Answers

General

Below you will find a list of the most frequently asked questions and answers regarding communication by Ethernet network.

Can a Momentum be searched by several ENT at a time?
Yes, in read only. Only one IO Scanner at a time should access the ENT in write mode,

Do I have to configure the IO Scanner connections in the connections table ("Messaging" tab)?
No.

What should I do if the TSX ETY 4103/PORT5103 module displays the LED status "Not configured" (Flashing ERR)?

- Check that it is configured in Unity Pro.
- Check that there is no other module with the same X-Way network number in the PLC configuration.
- If the module is configured in "Client/Server address", make sure that a BOOTP server is operational and accessible.

What should I do if the module displays a correct LED status (STS=RUN=ON) but does not respond to requests?
Try a Ping command to this IP address, from a device on the same network:

- No response:
  - Check the network configuration: wiring, "hubs" status, IP addresses, masks and gateways if there are routers in the configuration.
- Correct response to the Ping:
  - Check the module's X-Way address.
  - Check whether access control is activated or not.
  - Try to connect to the Rack Viewer page by http. If this connection works, check the X-Way address of the client application.

Can I change the MAC address of the module?
No. This is fixed for every Ethernet device to guarantee its unicity.

How can I find out the speed with which the module communicates?
Via an internet browser, by connecting to the Ethernet Module Statistics page.

What should I do if the messaging EFs do not get a response?

- Check that the X-Way destination address is part of the IP/X-Wayconnections table.
- Check that the configuration of the destination module does not prohibit this connection (access control).
- In debug mode, try the "Ping" command towards the IP address of the destination PLC, then try an "Identification" request towards the destination address, if it is part of the same X-Way network.

What should I do if I receive the 0xFF12 refusal code on the messaging EFs?
This code indicates that the module is full. Reduce the transition rate.

What should I do if the READ_ASYNC, WRITE_ASYN messaging EFs do not get a response (time out)?
Check that the "mast" tasks of the client and server PLCs are configured to "periodic", with a sufficient period to ensure that the PLCs are not overrun.

When should I configure my connections to "Mono-Connection"?
When this connection has to be established to a TSX ETY 110 using Modbus protocol.

The destination PLC of my messages is a bridge. Which X-Way destination address should I configure in the connections table?
The X-Way address of the first communication module of the PLC bridge (if present, it is the processor PCMCIA card).

I have changed the IP addresses in Address Server and IO Scanner configuration, should I restart my client/server devices?
Yes. The devices only receive their IP address when starting up. This restart should be carried out once the TSX ETY 4103/5103 module has been reconfigured (fixed RUN and STS LEDs).

In the module configuration screens, the last entry inserted into one of the grids has not been taken into account at the validation. Why?
You need to remove the focus (the scale indicator) from the last cell of the grid in which new data has been entered in order to take this modification into account.

Why is the Bridge tab grayed out when I use a processor of type ???????????
These processors do not manage bridge data.

Why is the Bridge tab grayed out when I have already changed the position of the processor?
After this type of modification, you have to validate the new hardware configuration to access and modify the bridge data managed globally at the level of the processor.

The momentum 170 ENT modules do not manage to obtain their client/server address from the address server of the TSX ETY 4103, TSX ETY PORT or TSX ETY 5103 modules.

- Check in the TSX ETY 4103 or TSX ETY PORT module configuration that the subnetwork mask is right for the IP address class.
- Check the software version of the 170 ENT module.

The Global Data service does not work. Why?
Make sure that the network switches are not configured to "multicast filtering".

Configuration of 140 NWM 100 00 module using Concept

10

At a Glance

Subject of this Chapter

This chapter describes configuration of the 140 NWM 100 00 module using Concept.

What's in this Chapter?

This chapter contains the following sections:

10.1 Configuration With Concept

Purpose

Overview

This section explains the procedure in configuring your module using Concept software.

What's in this Section?

This section contains the following topics:

Selecting Your PLC

Overview

The following information describes how to start to configure the 140 NWM 100 00 using Concept.

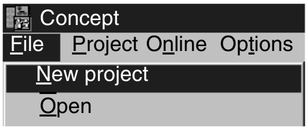
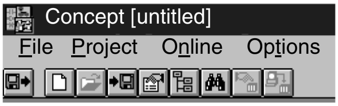
Initial Setup

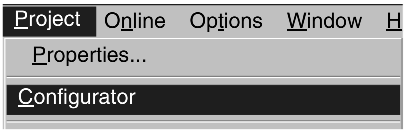
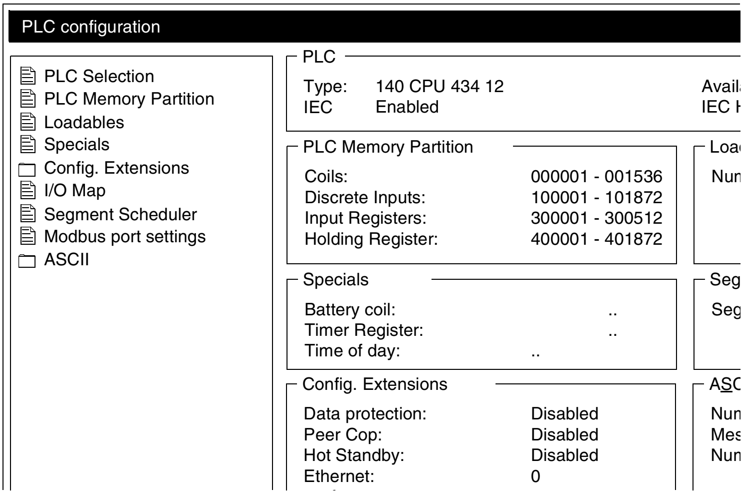
After the NWM module has been installed in a Quantum backplane (refer to the section *Installing the Module*), you can configure it using Concept. To configure the NWM, first select your CPU (PLC).

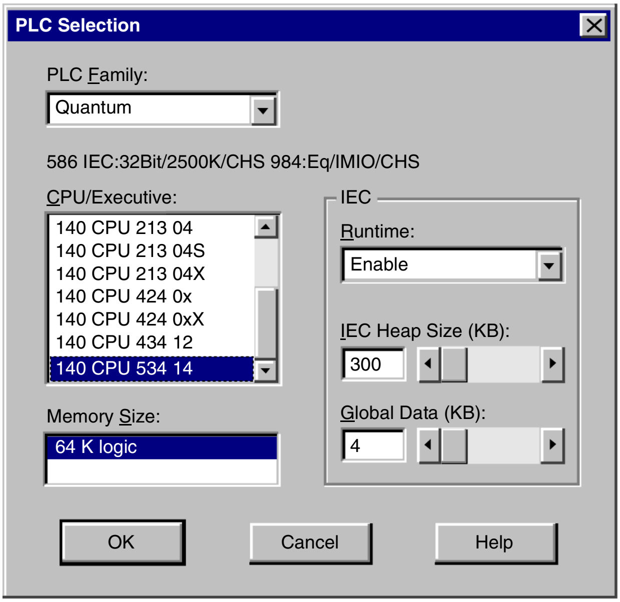
Note: For detailed information about how to use Concept, refer to the set of manuals shipped with that software.

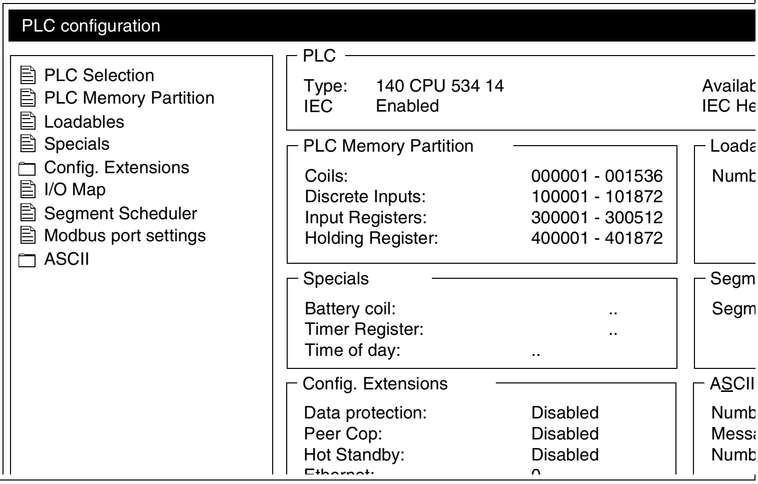
Procedure for Selecting a CPU

Perform the following steps to select a CPU.

Step	Action
1	Open Concept on your programming panel (PC).
2	As shown in the following figure, from the File menu, select **New project**.  Concept File Project Online Options New project Open **Result**: As shown in the following figure, a new project is opened and the file name (untitled) appears over the menu bar. 

Step	Action
3	Or, as shown in the following figure, from the Project menu, select **Configurator**.  Project Online Options Window H Properties... Configurator **Result**: As shown in the following figure, the PLC Configuration screen displays.  PLC configuration PLC Selection PLC Memory Partition Loadables Specials Config. Extensions I/O Map Segment Scheduler Modbus port settings ASCII PLC Type: 140 CPU 434 12 Avail IEC Enabled IEC I PLC Memory Partition Coils: 000001 - 001536 Discrete Inputs: 100001 - 101872 Input Registers: 300001 - 300512 Holding Register: 400001 - 401872 Loa Nun Specials Battery coil: .. Timer Register: .. Time of day: .. Seg Seg Config. Extensions Data protection: Disabled Peer Cop: Disabled Hot Standby: Disabled Ethernet: 0 ASC Nun Mes Nun

Step	Action
4	Double click on the **Type** field in the PLC section of the Configure menu. **Result**: As shown in the following figure, the PLC Selection dialog box displays. The default selection is Quantum.  PLC Selection PLC Family: Quantum 586 IEC:32Bit/2500K/CHS 984:Eq/IMIO/CHS CPU/Executive: 140 CPU 213 04 140 CPU 213 04S 140 CPU 213 04X 140 CPU 424 0x 140 CPU 424 0xX 140 CPU 434 12 140 CPU 534 14 Memory Size: 64 K logic IEC Runtime: Enable IEC Heap Size (KB): 300 Global Data (KB): 4
5	From the **CPU/Executive** scroll box, select the CPU that is installed in your Quantum backplane NOTE: Depending on the CPU selected, you may need to select the correct memory size applicable to it in the **Memory Size** dialog box.

Step	Action
6	Click the **<OK>** button. **Result:** As shown in the following figure, your PLC type and default configuration parameters are displayed in the PLC Configuration screen.  PLC configuration PLC Selection PLC Memory Partition Loadables Specials Config. Extensions I/O Map Segment Scheduler Modbus port settings ASCII PLC Type: 140 CPU 534 14 IEC Enabled Availab IEC He PLC Memory Partition Coils: 000001 - 001536 Discrete Inputs: 100001 - 101872 Input Registers: 300001 - 300512 Holding Register: 400001 - 401872 Loada Numb Specials Battery coil: .. Timer Register: .. Time of day: .. Segm Segm Config. Extensions Data protection: Disabled Peer Cop: Disabled Hot Standby: Disabled Ethernet: 0 ASCII Numb Mess Numb

Next

Next, you must configure the number of Ethernet modules that your system will contain, as shown in the *Cabling Schemes* procedure.

Setting the Number of NWMs

Overview

The following information describes how many NWMs you can configure in a single controller and how to configure that number.

Introduction

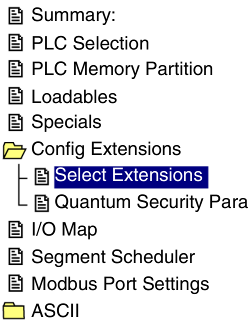
You may configure a mamimum of two to six Ethernet modules in a single controller, depending on the model. A 140 CPU 113 or 213 will accept a total of two network option modules, including NWM, NOE, NOM, NOP, and CRP 811. A 140 CPU 424, 434, 534. 434A, or 534A will accept six.

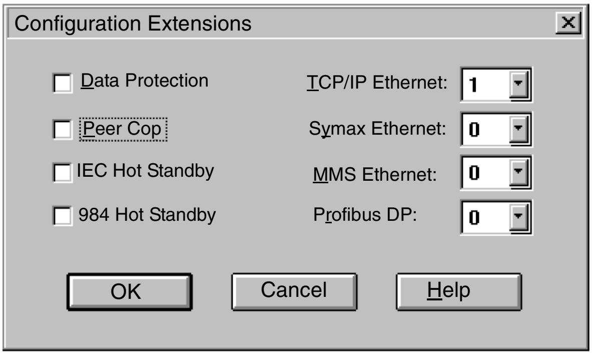
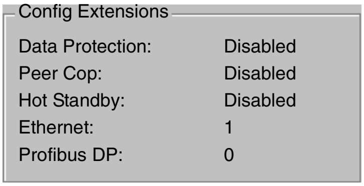
Memory Requirements

The first Ethernet TCP/IP module configured requires 20 words of memory. Each additional module requires an additional 16 words of memory.

Procedure for Setting the Number of NWMs

From the PLC Configuration screen, follow the steps below to select the number of 140 NWM 100 00 modules.

Step	Action
1	As shown in the following figure, from the Configure menu, select Config Extensions, or double-click anywhere in the **Config Extensions** region of the screen. Next, select the Select Extensions prompt.  Summary: PLC Selection PLC Memory Partition Loadables Specials Config Extensions Select Extensions Quantum Security Para I/O Map Segment Scheduler Modbus Port Settings ASCII **Result**: The Configuration Extension dialog box displays.

Step	Action
2	As shown in the following figure, in the TCP/IP Ethernet scroll box, select the number of 140 NWM 100 00 modules to be configured.  Configuration Extensions ☐ Data Protection ☐ Peer Cop ☐ IEC Hot Standby ☐ 984 Hot Standby TCP/IP Ethernet: 1 Symax Ethernet: 0 MMS Ethernet: 0 Profibus DP: 0
3	Click on the <**OK**> button **Result**: As shown in the following figure, the Ethernet status in the Config Extensions section of the PLC Configuration screen, changes from 0 to the number selected in Step 2.  Config Extensions Data Protection: Disabled Peer Cop: Disabled Hot Standby: Disabled Ethernet: 1 Profibus DP: 0

Next

Next, you need to create an I/O map for the NWMs in your configuration, as shown in the *Accessing and Editing the I/O Map* (See *Accessing and Editing the I/O Map, p. 250*) section.

Accessing and Editing the I/O Map

Overview

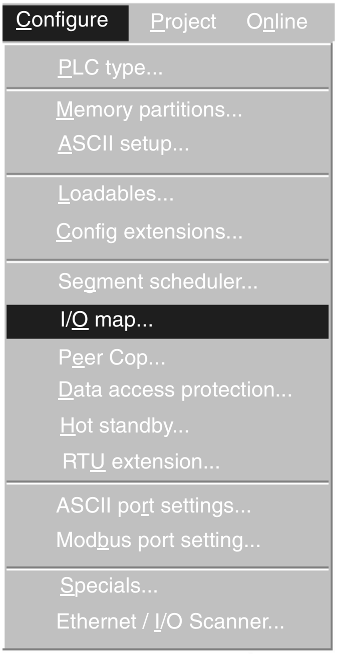
The following information describes how to create an I/O map for the NWMs in your system.

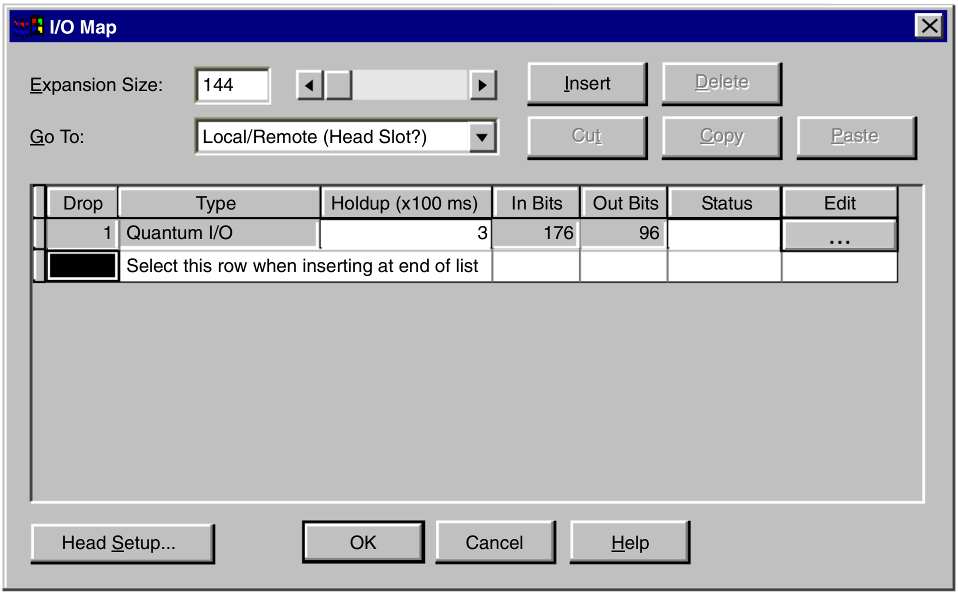
Introduction

This procedure is required to determine the slot location of NWMs in the system. As part of the configuration process, you need to create an I/O Map for the local backplane including the 140 NWM 100 00 module.

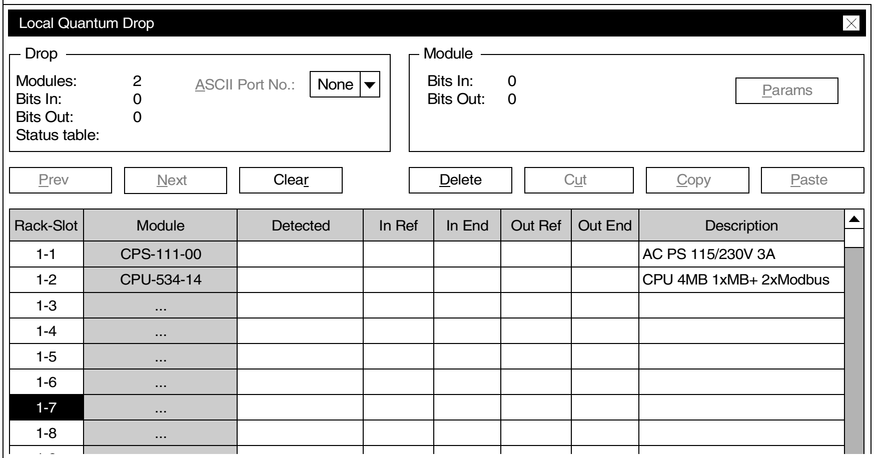
Procedure for Accessing and Editing an I/O Map

Perform the following steps to access and edit an I/O Map from the PLC Configuration screen.

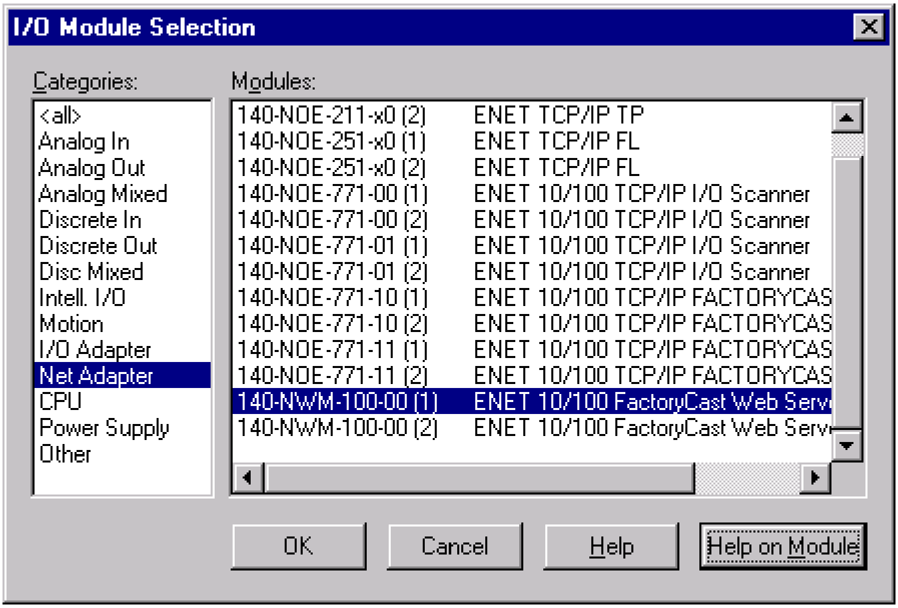
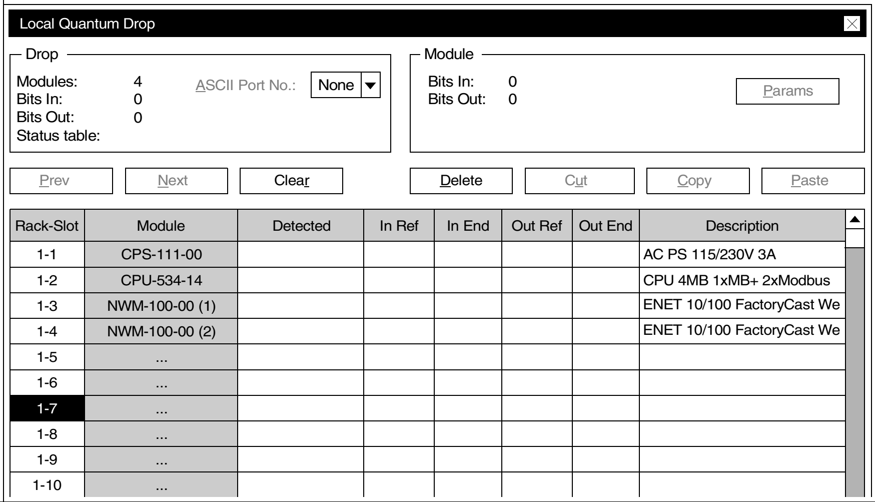
Step	Action
1	As shown in the following figure, from the Configure menu, select **I/O map**.  Configure Project Online PLC type... Memory partitions... ASCII setup... Loadables... Config extensions... Segment scheduler... I/O map... Peer Cop... Data access protection... Hot standby... RTU extension... ASCII port settings... Modbus port setting... Specials... Ethernet / I/O Scanner... **Result**: As shown in the following figure, the I/O Map dialog box displays.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>
Click the &lt;<b>Edit</b>&gt; button at the end of the Quantum I/O row.


Expansion Size: 144
Go To: Local/Remote (Head Slot?)
<table>
<tr><th>Drop</th><th>Type</th><th>Holdup (x100 ms)</th><th>In Bits</th><th>Out Bits</th><th>Status</th><th>Edit</th></tr>
<tr><td>1</td><td>Quantum I/O</td><td>3</td><td>176</td><td>96</td><td></td><td>...</td></tr>
<tr><td></td><td>Select this row when inserting at end of list</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

The Local Quantum Drop I/O box is displayed. Note: there may be no modules yet listed under the Module column. Continue to the next step.


Drop
Modules: 2
Bits In: 0
Bits Out: 0
Status table:
ASCII Port No.: None
Module
Bits In: 0
Bits Out: 0
<table>
<tr><th>Rack-Slot</th><th>Module</th><th>Detected</th><th>In Ref</th><th>In End</th><th>Out Ref</th><th>Out End</th><th>Description</th></tr>
<tr><td>1-1</td><td>CPS-111-00</td><td></td><td></td><td></td><td></td><td></td><td>AC PS 115/230V 3A</td></tr>
<tr><td>1-2</td><td>CPU-534-14</td><td></td><td></td><td></td><td></td><td></td><td>CPU 4MB 1xMB+ 2xModbus</td></tr>
<tr><td>1-3</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1-4</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1-5</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1-6</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1-7</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1-8</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>3</td><td>Click on the <b>...</b> button under <b>Module</b>.
<b>Result</b>: As shown in the following figure, the I/O Module Selection drop-down menu displays.

I/O Module Selection
Categories:
&lt;all&gt;
Analog In
Analog Out
Analog Mixed
Discrete In
Discrete Out
Disc Mixed
Intell. I/O
Motion
I/O Adapter
Net Adapter
CPU
Power Supply
Other
Modules:
<table>
<tr><td>140-NOE-211-x0 (2)</td><td>ENET TCP/IP TP</td></tr>
<tr><td>140-NOE-251-x0 (1)</td><td>ENET TCP/IP FL</td></tr>
<tr><td>140-NOE-251-x0 (2)</td><td>ENET TCP/IP FL</td></tr>
<tr><td>140-NOE-771-00 (1)</td><td>ENET 10/100 TCP/IP I/O Scanner</td></tr>
<tr><td>140-NOE-771-00 (2)</td><td>ENET 10/100 TCP/IP I/O Scanner</td></tr>
<tr><td>140-NOE-771-01 (1)</td><td>ENET 10/100 TCP/IP I/O Scanner</td></tr>
<tr><td>140-NOE-771-01 (2)</td><td>ENET 10/100 TCP/IP I/O Scanner</td></tr>
<tr><td>140-NOE-771-10 (1)</td><td>ENET 10/100 TCP/IP FACTORYCAS</td></tr>
<tr><td>140-NOE-771-10 (2)</td><td>ENET 10/100 TCP/IP FACTORYCAS</td></tr>
<tr><td>140-NOE-771-11 (1)</td><td>ENET 10/100 TCP/IP FACTORYCAS</td></tr>
<tr><td>140-NOE-771-11 (2)</td><td>ENET 10/100 TCP/IP FACTORYCAS</td></tr>
<tr><td>140-NWM-100-00 (1)</td><td>ENET 10/100 FactoryCast Web Serv</td></tr>
<tr><td>140-NWM-100-00 (2)</td><td>ENET 10/100 FactoryCast Web Serv</td></tr>
</table>
</td></tr>
<tr><td>4</td><td>Click on <b>140 NWM 100 00</b>, and then click the &lt;<b>OK</b>&gt; button.
<b>Result</b>: The Local Quantum Drop I/O dialog box updates and the <b>NWM</b> is now listed under Module and described in the Description column.

Local Quantum Drop
Drop
Modules: 4
Bits In: 0
Bits Out: 0
Status table:
ASCII Port No.: None
Module
Bits In: 0
Bits Out: 0

<table>
<tr><th>Rack-Slot</th><th>Module</th><th>Detected</th><th>In Ref</th><th>In End</th><th>Out Ref</th><th>Out End</th><th>Description</th></tr>
<tr><td>1-1</td><td>CPS-111-00</td><td></td><td></td><td></td><td></td><td></td><td>AC PS 115/230V 3A</td></tr>
<tr><td>1-2</td><td>CPU-534-14</td><td></td><td></td><td></td><td></td><td></td><td>CPU 4MB 1xMB+ 2xModbus</td></tr>
<tr><td>1-3</td><td>NWM-100-00 (1)</td><td></td><td></td><td></td><td></td><td></td><td>ENET 10/100 FactoryCast We</td></tr>
<tr><td>1-4</td><td>NWM-100-00 (2)</td><td></td><td></td><td></td><td></td><td></td><td>ENET 10/100 FactoryCast We</td></tr>
<tr><td>1-5</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1-6</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1-7</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1-8</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1-9</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1-10</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

Step	Action
5	Repeat Steps 3 and 4 if other modules need to be added to the I/O map.
6	Click the <**OK**> buttons to return to the PLC Configuration screen.

Next

Next, you will configure the Ethernet address parameters from the Ethernet/ I/O Scanner screen as shown in the *Configuring the Ethernet Address Parameters* (See *Configuring the Ethernet Address Parameters, p. 254*) section.

Configuring the Ethernet Address Parameters

Overview

The following information describes how to configure Ethernet address parameters for the 140 NWM 100 00 with Concept.

Introduction

The 140 NWM 100 00 module’s Ethernet address parameters, consisting of IP address, Subnet mask, and Gateway addresses, are accessible from the Ethernet/I/O Scanner dialog box. Prior to performing the following procedure, consult your system administrator to determine if you must configure new Ethernet address parameters, or whether the module will obtain them from the BOOTP server.

Note: Concept has to be disconnected from the CPU to configure the 140 NWM 100 00 module.

⚠ **CAUTION**

UNINTENTIONAL OPERATION — DUPLICATE IP ADDRESS

Having two devices with the same IP address can cause unpredictable operation of your network.

- Ensure that this device will receive a unique IP address.
- Always obtain your IP addresses from your system administrator to avoid the possibility of duplicate addresses.

Failure to follow this precaution can result in injury or equipment damage.

Procedure for Configuring Ethernet Address Parameters

Perform the following steps to configure the Ethernet Address Parameters.

Step	Action
1	From the Config Extensions menu, select Ethernet I/O Scanner. **Result**: The Ethernet/ I/O Scanner dialog box displays.
2	Click on the **Specify IP Address** radio button.
3	Type in the new IP Address, Subnet Mask, and Gateway address in the applicable text boxes.
4	Select the correct Internet frame type from the **Frame Type** scroll box.
5	If the module's BOOTP server will assign Ethernet address parameters, click on the Use BOOTP Server radio button. Note that if you select this option, the address parameter text boxes will be grayed out and will not display the addresses.

How the Module Derives Its IP Address

During initialization, the 140 NWM 100 00 module attempts to read the address parameter information from the PLC and determines its IP Address in the following fashion.

- If the PLC has the IP Address and the BOOTP server is not selected, the module will use the configured IP address that you assigned in Step 2 of the above procedure.
- If the BOOTP server was selected in Step 5 of the above procedure, the module will send BOOTP requests to receive its IP Address.
- If no Configuration Extension exists, the 140 NWM 100 00 sends out BOOTP requests. If the module does not receive its IP Address from the BOOTP server after 2 minutes, it will then use the IP Address derived from its MAC address.

Note: The MAC address is assigned at the factory and is recorded on a label on the front panel, above the cable connector. This is a unique 48-bit global assigned address. It is set in PROM. The Ethernet address is recorded on the label in hexadecimal, in the form 00.00.54.xx.xx.xx.

10.2 Transfer Data Using 984 LL

Purpose

Overview

This section explains how to transfer data using 984 Ladder Logic.

What's in this Section?

This section contains the following topics:

MSTR Description

Overview

The following information describes MSTR operations.

Introduction

All 140 NWM 100 00 Quantum Ethernet TCP/IP modules provide the user with the capability of transferring data to and from nodes on a TCP/IP network through the use of a special MSTR (master instruction). All PLCs that support networking communication capabilities over Modbus Plus and Ethernet can use the MSTR ladder logic instruction to read or write controller information.

MSTR Operations

The following table lists each of the possible MSTR network communications operations and indicates whether a TCP/IP Ethernet network supports it. Each operation is designated by a code. The following table lists the operations and indicates those that are supported on an Ethernet TCP/IP network.

MSTR Operation	Operation Type	TCP/IP Ethernet Support
Write data	1	supported
Read Data	2	supported
Get local statistics	3	supported
Clear local statistics	4	supported
Get remote statistics	5	supported
Clear remote statistics	6	supported
Reset Option Module	7	supported
Read CTE (config extension)	8	supported
Write CTE (config extension)	9	supported

Number of MSTR Instructions Allowed

Up to16 MSTR instructions can be simultaneously serviced in a ladder logic program per NWM. More than 16 MSTRs may be programmed to be enabled by the logic flow as one active MSTR block releases the resources it has been using and becomes deactivated, the next MSTR operation encountered in logic can be activated.

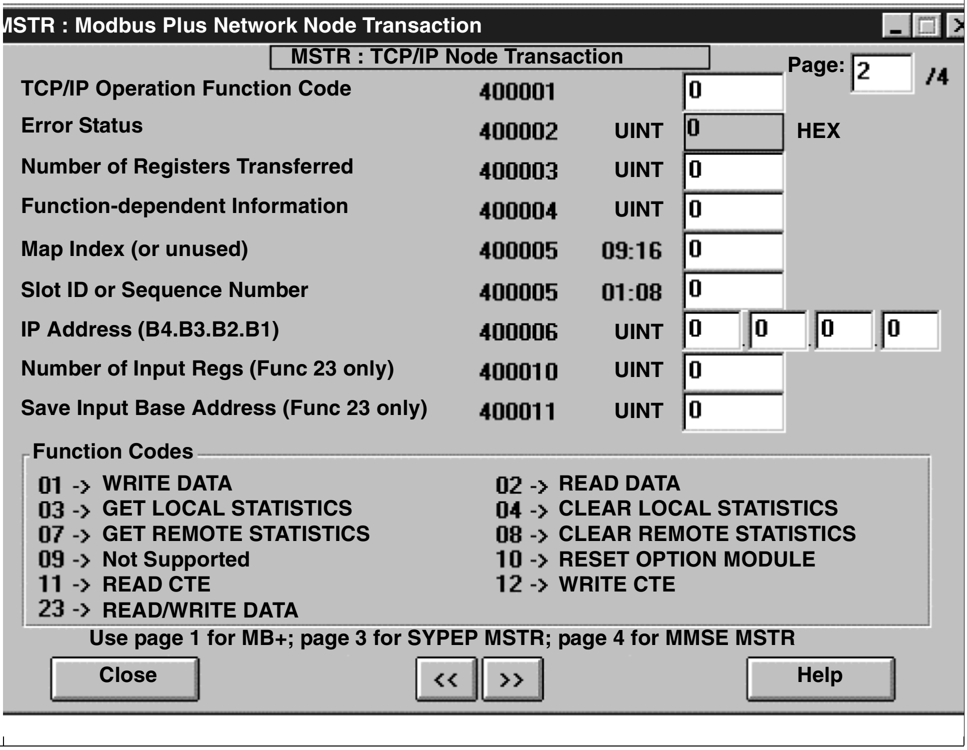
MSTR Block for TCP/IP in Concept

Overview

This following information describes how to complete installing the MSTR block in a TCP/IP network for the 140 NWM 100 00 module.

Procedure for Installing the MSTR Block in TCP/IP

This is the MSTR Block as used for TCP/IP in Concept Ladder Logic. After the MSTR Block is inserted in the network, perform the following steps.

Step	Action
1	Place cursor on the MSTR block.
2	Configure the control block and data area by a double click
3	Press <Ctrl D> or **Edit** → **DX Zoom** (DX Zoom screen)
4	Go to page 2 for TCP/IP specific MSTR block and complete the required information. 

ISTR : Modbus Plus Network Node Transaction

MSTR : TCP/IP Node Transaction

Page: 2 /4

TCP/IP Operation Function Code	400001		0	
Error Status	400002	UINT	0	HEX
Number of Registers Transferred	400003	UINT	0	
Function-dependent Information	400004	UINT	0	
Map Index (or unused)	400005	09:16	0	
Slot ID or Sequence Number	400005	01:08	0	
IP Address (B4.B3.B2.B1)	400006	UINT	0 . 0 . 0 . 0	
Number of Input Regs (Func 23 only)	400010	UINT	0	
Save Input Base Address (Func 23 only)	400011	UINT	0	

Function Codes

01 -> WRITE DATA	02 -> READ DATA
03 -> GET LOCAL STATISTICS	04 -> CLEAR LOCAL STATISTICS
07 -> GET REMOTE STATISTICS	08 -> CLEAR REMOTE STATISTICS
09 -> Not Supported	10 -> RESET OPTION MODULE
11 -> READ CTE	12 -> WRITE CTE
23 -> READ/WRITE DATA	

Use page 1 for MB+; page 3 for SYPEP MSTR; page 4 for MMSE MSTR

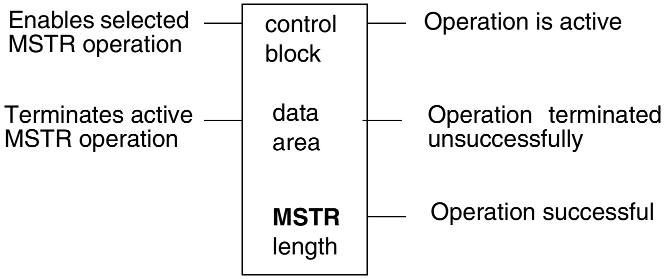
MSTR Ladder Logic Representation

Overview

The following information describes the Ladder Logic representation for MSTR.

Ladder Logic Diagram

As the following figure shows, the MSTR Block is represented in Ladder Logic diagrams.


Enables selected MSTR operation
Terminates active MSTR operation
control block
data area
MSTR length
Operation is active
Operation terminated unsuccessfully
Operation successful

Inputs

The MSTR instruction has the following two control inputs.

- the input to the top node enables the instruction if it is ON
- the input to the middle node terminates the active operation if it is ON

Outputs

The MSTR instruction can produce the following three possible outputs.

- the output from the top node echoes the state of the top input - it goes ON while the instruction is active
- the output from the middle node echoes the state of the middle input - it goes ON if the MSTR operation is terminated prior to completion or if an error occurs in completing the operation
- the output from the bottom node goes ON if an MSTR operation has been completed successfully

All outputs are zero is an indication that four MSTR instructions are already in progress.

Top Node Content

The 4x register entered in the top node is the first of several (network dependent) holding registers that comprise the network *control block*. The *control block* structure differs according to the network in use.
In the case of the Ethernet Read and Write CTE operations the middle node stores the contents of the Ethernet configuration extension table in a series of registers.
The following table shows the *control block* structure for the TCP/IP Ethernet network.

Register	Content
Displayed	Identifies one of ten MSTR operations legal for TCP/IP (1... 4 and 7... 12).
First implied	Displays error status.
Second implied	Displays length (number of registers transferred).
Third implied	Displays MSTR operation-dependent information.
Fourth implied	Low byte: Destination index.
	High byte: Quantum backplane slot address of the NWM module.
Fifth implied	Byte 4 (MSB) of the 32-bit destination IP Address.
Sixth implied	Byte 3 of the 32-bit destination IP Address.
Seventh implied	Byte 2 of the 32-bit destination IP Address
Eight implied	Byte 1 (LSB) of the 32-bit destination IP Address

Middle Node Content

The 4x register entered in the middle node is the first in a group of contiguous holding registers that comprise the *data area*. For operations that provide the communication processor with data such as a Write operation, the *data area* is the source of the data. For operations, such as Read, that get data from the communication processor, the *data area* is the destination for the data
In the case of the Ethernet Read and Write CTE operations, the middle node stores the contents of the Ethernet configuration extension table in a series of registers.

Bottom Node Content

The integer value entered in the bottom node specifies the *length* - the maximum number of registers in the *data area*. The *length* must be in the range 1... 100.

MSTR Function Error Codes

Overview

The following information describes the error codes for MSTR operations for the NWM module.

Where Displayed

If an error occurs during an MSTR operation, a hexadecimal error code is displayed in the first implied register in the *control block* (the top node). Function error codes are network-specific.

TCP/IP Ethernet Error Codes

The following table describes the errors that can occur in the MSTR *control block* if the MSTR routine runs over TCP/IP Ethernet.

Hex Error Code	Meaning
1001	User has aborted the MSTR element.
2001	An unsupported operation type has been specified in the *control block*.
2002	One or more *control block* parameters has been changed while the MSTR element is active (applies only to operations that take multiple scans to complete). *Control block* parameters may be changed only when the MSTR element is not active.
2003	Invalid value in the length field of the *control block*.
2004	Invalid value in the offset field of the *control block*.
2005	Invalid values in the length and offset fields of the *control block*.
2006	Invalid slave device data area.
2008	Invalid slave device network routing
3000	Generic MODBUS fail code.
30ss*	MODBUS slave exception response.
4001	Inconsistent MODBUS slave response.
F001	Option Module not responding
* ss = subfield	

The following table lists the ss subfield values in error code 30ss.

ss Hex value	Meaning
01	Slave device does not support the requested operation.
02	Nonexistent slave device registers requested.
03	Invalid data value requested.
04	Reserved
05	Slave has accepted long-duration program command.

06	Function cannot be performed now; a long-duration command is in effect.
07	Slave rejected long-duration program command.

TCP/IP Ethernet Network Errors

The following table describes the errors that can occur in the MSTR *control block* as a result of an error on the TCP/IP Ethernet network.

Hex Error Code	**Meaning**
5004	Interrupted system call
5005	I/O error
5006	No such address
5009	The socket descriptor is invalid
500C	Not enough memory
500D	Permission denied
5011	Entry exists
5016	An argument is invalid
5017	An internal table has run out of space
5020	The connection is broken
5028	Destination address required
5029	Protocol wrong type for socket
502A	Protocol not available
502B	Protocol not supported
502C	Socket type not supported
502D	Operation not supported on a socket
502E	Protocol family not supported
502F	Address family not supported
5030	Address already in use
5031	Cannot assign requested address
5032	Socket operation on a non-socket
5033	Network is unreachable
5034	Network dropped connection on reset
5035	Network caused connection abort
5036	Connection reset by peer
5037	No buffer space available
5038	Socket is already connected
5039	Socket is not connected

503A	Cannot send after socket shutdown
503B	Too many references, cannot splice
503C	Connection timed-out (See note below.)
503D	Connection refused
503E	Network is down
503F	Text file busy
5040	Too many levels of links
5041	No route to host
5042	Block device required
5043	Host is down
5044	Operation now in progress
5045	Operation already in progress
5046	Operation would block
5047	Function not implemented
5050	No Network Resource
5051	Length Error
5052	Addressing Error
5053	Application Error
5054	Client in Bad State for Request
5055	No Remote Resource (Note: May indicate no path to Remote Device) (See note below.)
5056	Non-Operational TCP Connection
5057	Incoherent Configuration

Note: Error Codes 503C and 5055

- Error 5055 can occur before a 503C error.

No remote device takes precedence over a timeout.

CTE Error Codes

The following table lists the error codes that are returned if there is a problem with the Ethernet configuration extension table (CTE) in your program configuration.

Hex Error Code	Meaning
7001	There is no Ethernet configuration extension.
7002	The CTE is not available for access.
7003	The offset is invalid.

7004	The offset + length is invalid.
7005	Bad data field in the CTE.

Read and Write MSTR Operations

Overview

This section describes the MSTR Read and Write operations.

Introduction

The MSTR Write operation (operation type 1 in the displayed register of the top node) transfers data from a master source device to a specified slave destination device on the network. The MSTR Read operation (operation type 2 in the displayed register of the top node) transfers data from a specified slave source device to a master destination device on the network. Read and Write use one data master transaction path and may be completed over multiple scans.

Note: TCP/IP Ethernet routing must be accomplished via standard third-party Ethernet IP router products.

Control Block Utilization

The following table describes the registers in the MSTR *control block* (the top node) that contain the Read or Write information.

<table>
<tr><th>Register</th><th>Function</th><th colspan="2">Content</th></tr>
<tr><td>Displayed</td><td>Operation Type</td><td colspan="2">1 = Write, 2 = Read</td></tr>
<tr><td rowspan="4">First implied</td><td rowspan="4">Error status</td><td colspan="2">Displays a hex value indicating an MSTR error.</td></tr>
<tr><td>Exception response, where response size is incorrect.</td><td>Exception code +3000</td></tr>
<tr><td>Exception response where response size is incorrect.</td><td rowspan="2">4001</td></tr>
<tr><td>Read Write</td></tr>
<tr><td>Second implied</td><td>Length</td><td colspan="2">Write = number of registers to be sent to slave. Read = number of registers to be read from slave.</td></tr>
<tr><td>Third implied</td><td>Slave device data area</td><td colspan="2">Specifies starting 4x register in the slave to be read from or written to (1 = 4001, 49 =40049).</td></tr>
<tr><td>Fourth implied</td><td>Low byte</td><td colspan="2">Quantum backplane slot address of the NWM module.</td></tr>
<tr><td>Fifth ... Eighth implied</td><td>Destination</td><td colspan="2">Each register contains one byte of the 32-bit IP address.</td></tr>
</table>

Get Local Statistics MSTR Operation

Overview

This section describes the Get Local Statistics operation.

Introduction

The Get Local Statistics operation (operation type 3 in the display register of the top node) obtains information related to the local node where the MSTR has been programmed. Refer to TCP/IP Ethernet Statistics for information recieved in the Get Local Statistics operation.

Control Block Utilization

The following table describes the registers in the MSTR *control block* (the top node). These registers contain the Get Local Statistics information.

Register	Function	Content
Displayed	Operation Type	3
First implied	Error status	Displays a hex value indicating an MSTR error, if relevant.
Second implied	Length	Starting from *offset*, the number of words of statistics from the local processor's statistics table; the *length* must be > 0 < *data area*.
Third implied	Offset	An offset value relative to the first available word in the local processor's statistics table. If the offset is specified as 1, the function obtains statistics starting with the second word in the table.
Fourth implied	Low byte	Quantum backplane slot address of the NWM module.
Fifth ... Eighth implied	Not applicable	

Clear Local Statistics MSTR Operation

Overview

The following information describes the Local Statistics operation.

Introduction

The Clear Local Statistics operation (operation type 4 in the displayed register of the top node) clears statistics relative to the local node where the MSTR has been programmed.

Control Block Utilization

The following table describes the registers in the MSTR *control block* (the top node). These registers contain the Clear Local Statistics information.

Register	Function	Content
Displayed	Operation Type	4
First implied	Error status	Displays a hex value indicating an MSTR error, if relevant.
Second implied	Not applicable	
Third implied	Not applicable	
Fourth implied	Low byte	Quantum backplane slot address of the NWM module.
Fifth ... Eighth implied	Not applicable	

Get Remote Statistics MSTR Operation

Overview

The following information describes the Get Remote Statistics operation.

Introduction

The Get Remote Statistics operation (operation type 7 in the displayed register of the top node) obtains information relative to remote nodes on the network. This operation may require multiple scans to complete and does not require a master data transaction path.

The remote Ethernet module always returns its complete statistics table if a request is made, even if the request is for less than the full table. The MSTR instruction then copies only the amount of words you have requested to the designated 4x registers.

Note: TCP/IP Ethernet routing must be accomplished via standard third-party Ethernet IP router products.

Control Block Utilization

The following table describes the registers in the MSTR *control block* (the top node). These registers contain the Get Remote Statistics information.

Register	Function	Content
Displayed	Operation Type	7
First implied	Error status	Displays a hex value indicating an MSTR error, if relevant.
Second implied	Length	Starting from an *offset*, the number of words of statistics from the local processor's statistics table; the length must be > 0 < *data area*.
Third implied	Offset	Specifies an offset value relative to the first available word in the local processor's statistics table. If the *offset* is specified as 1, the function obtains statistics starting with the second word in the table.
Fourth implied	High byte	Destination index
Fifth ... Eighth implied	Destination	Each register contains one byte of the 32-bit IP address.

Clear Remote Statistics MSTR Operation

Overview

The following information describes the Clear Remote Statistics operation.

Introduction

The Clear Remote Statistics operation (operation type 8 in the displayed register of the top node) clears statistics relative to a remote network node from the *data area* in the local node. This operation may require multiple scans to complete and uses a single data master transaction path.

Control Block Utilization

The following table describes the registers in the MSTR *control block* (the top node). These registers contain the Clear Remote Statistics information.

Register	Function	Content
Displayed	Operation Type	8
First implied	Error status	Displays a hex value indicating an MSTR error, if relevant.
Second implied	Not applicable	
Third implied	Not applicable	
Fourth implied	High byte	Destination index
Fifth ... Eighth implied	Destination	Each register contains one byte of the 32-bit IP address.

Reset Option Module MSTR Operation

Overview

The following information describes the Reset Option Module operation.

Introduction

The Reset Option Module operation (operation type 10 in the displayed register of the top node) causes a Quantum 140 NWM 100 00 option module to enter a reset cycle to reset its operational environment.

Control Block Utilization

The following table describes the registers in the MSTR *control block* (the top node). These registers contain the Reset Option Module information.

Register	Function	Content
Displayed	Operation Type	10
First implied	Error status	Displays a hex value indicating a MSTR error, if relevant.
Second implied	Not applicable	
Third implied	Not applicable	
Fourth implied	Low byte	Quantum backplane slot address of the NWM module.
Fifth ... Eighth implied	Not applicable	

Read CTE (Config Extension Table) MSTR Operation

Overview

The following information describes the Read CTE operation.

Introduction

The Read CTE operation (operation type 11 in the displayed register of the top node) reads a given number of bytes from the Ethernet configuration extension table to the indicated buffer in PLC memory. The bytes to be read begin at a byte offset from the beginning of the CTE. The content of the Ethernet CTE table is displayed in the middle node of the MSTR block.

Control Block Utilization

The following table describes the registers in the MSTR *control block* (the top node). These registers contain the Read CTE information.

Register	Function	Content
Displayed	Operation Type	11
First implied	Error status	Displays a hex value indicating a MSTR error, when relevant.
Second implied	Not applicable	
Third implied	Not applicable	
Fourth implied	Low byte	Quantum backplane slot address of the NWM module.
Fifth ... Eighth implied	Not applicable	

CTE Display Implementation

The values in the Ethernet configuration extension table (CTE) are displayed in a series of registers in the middle node of the MSTR instruction when a Read CTE operation is implemented. The middle node contains the first of 11 contiguous 4x registers.

The following table describes the CTE data contained in the registers.

<table>
<tr><th>Parameter</th><th>Register</th><th colspan="2">Content</th></tr>
<tr><td rowspan="2">Frame type</td><td rowspan="2">Displayed</td><td colspan="2">1 = 802.3</td></tr>
<tr><td colspan="2">2 = Ethernet</td></tr>
<tr><td rowspan="4">IP Address</td><td>First implied</td><td colspan="2">Byte 4 (MSB) of the 32-bit IP address</td></tr>
<tr><td>Second implied</td><td colspan="2">Byte 3 of the 32-bit IP address</td></tr>
<tr><td>Third implied</td><td colspan="2">Byte 2 of the 32-bit IP address</td></tr>
<tr><td>Fourth implied</td><td colspan="2">Byte 1 (LSB) of the 32-bit IP address</td></tr>
<tr><td rowspan="2">Subnetwork mask</td><td>Fifth implied</td><td colspan="2">Hi word</td></tr>
<tr><td>Sixth implied</td><td colspan="2">Low word</td></tr>
<tr><td rowspan="4">Gateway</td><td>Seventh implied</td><td colspan="2">Byte 4 (MSB) of the 32-bit gateway address</td></tr>
<tr><td>Eighth implied</td><td colspan="2">Byte 3 of the 32-bit gateway address</td></tr>
<tr><td>Ninth implied</td><td colspan="2">Byte 2 of the 32-bit gateway address</td></tr>
<tr><td>Tenth implied</td><td colspan="2">Byte 1 (LSB) of the 32-bit gateway address</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">Eleventh implied</td><td><b>High byte</b></td><td><b>Low byte</b></td></tr>
<tr><td></td><td></td></tr>
</table>

Note: Module type only used by the p-unit during an upload to determine module.

Write CTE (Config Extension Table) MSTR Operation

Overview

The following information describes the Write CTE operation.

CTE Write Implementation

The Write CTE operation writes an indicated number of bytes from PLC memory, starting at a specified byte address, to an indicated Ethernet configuration extension table at a specified offset. The content of the Ethernet CTE table is contained in the middle node of the MSTR block.

Network Implementation

The Write CTE operation (type 12 in the displayed register of the top node) can be implemented for TCP/IP Ethernet networks via the appropriate network adapter.

Note: Modbus Plus networks do not use this operation.

Control Block Utilization

In a Write CTE operation, the registers in the MSTR *control block* (the top node) differ according to the network in user.
The following table describes the registers in the MSTR *control block* (the top node). These registers contain the Write CTE information.

Register	Function	Content
Displayed	Operation Type	12
First implied	Error status	Displays a hex value indicating a MSTR error, if relevant.
Second implied	Not applicable	
Third implied		
Fourth implied	Low byte	Either a value displayed in the high byte of the register or not used.
	Slot Index	Number displayed in the low byte, in a range 1 ... 16 indicating the slot in the local backplane where the option resides.
Fifth ... Eighth implied	Not applicable	

CTE Display Implementation

The values in the Ethernet configuration extension table (CTE) are displayed in a series of registers in the middle node of the MSTR instruction if a Write CTE operation is implemented. The middle node contains the first of 11 contiguous 4x registers.
The following table describes the CTE data contained in the registers.

<table>
<tr><th>Parameter</th><th>Register</th><th colspan="2">Content</th></tr>
<tr><td rowspan="2">Frame type</td><td rowspan="2">Displayed</td><td colspan="2">1 = 802.3</td></tr>
<tr><td colspan="2">2 = Ethernet</td></tr>
<tr><td rowspan="4">IP Address</td><td>First implied</td><td colspan="2">First byte of the IP address</td></tr>
<tr><td>Second implied</td><td colspan="2">Second byte of the IP address</td></tr>
<tr><td>Third implied</td><td colspan="2">Third byte of the IP address</td></tr>
<tr><td>Fourth implied</td><td colspan="2">Fourth byte of the IP address</td></tr>
<tr><td rowspan="2">Subnetwork mask</td><td>Fifth implied</td><td colspan="2">Hi word</td></tr>
<tr><td>Sixth implied</td><td colspan="2">Low word</td></tr>
<tr><td rowspan="4">Gateway</td><td>Seventh implied</td><td colspan="2">First byte of the gateway</td></tr>
<tr><td>Eighth implied</td><td colspan="2">Second byte of the gateway</td></tr>
<tr><td>Ninth implied</td><td colspan="2">Third byte of the gateway</td></tr>
<tr><td>Tenth implied</td><td colspan="2">Fourth byte of the gateway</td></tr>
<tr><td></td><td>Eleventh implied</td><td><b>High byte</b></td><td><b>Low byte</b></td></tr>
</table>

TCP/IP Ethernet Statistics

Overview

The following information describes the available TCP/IP Ethernet Statistics.

Introduction

A TCP/IP Ethernet board responds to the "Get Local Statistics" and "Set Local Statistics" commands with the following information.

Word	Meaning
00 ... 02	MAC address
03	Board Status (see the Board Status Bit Definition table in this map)
04 and 05	Number of receiver interrupts
06 and 07	Number of transmitter interrupts
08 and 09	Transmit _ timeout error count
10 and 11	Collision_detect error count
12 and 13	Missed packets
14 and 15	Memory error
16 and 17	Number of times driver has restarted
18 and 19	Receive framing error
20 and 21	Receiver overflow error
22 and 23	Receive CRC error
24 and 25	Receive buffer error
26 and 27	Transmit buffer error
28 and 29	Transmit silo underflow
30 and 31	Late collision
32 and 33	Lost carrier
34 and 35	Number of retries
36 and 37	IP address

Board Status Word Bit Definition

The following table describes the word bit definitions for Board Status.

Bit #	Definition
15	NWM Link On=1, Off=0
14	NWM Appl LED On-1, Off=0
13	NWM Fiber/TP)Fiber=1, TP=0)
12	NWM 100Mbit (100=1, 10=0
11-4	Module Type (see table below)
3	UNUSED
2	NWM Full Duplex (0=Half, 1=Full)
1	NWM Configured
0	NWM Runningt

Board Status Word Bit Definition by Module Type

The following table describes the values of the module types.

Value of Bits 11-4	Module Type
0	NOE 2x1
1	170 ENT 110 00
2	M1E
3	NOE 771 00
4	ETY 410
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	NOE 771 10
11	NOE 771 01
12	NOE 771 11
13	NWM 100 00
14	170 ENT 111 01

For bit level detail for the Momentum 170ENT11001 and Momentum 170ENT11000 see the *Momentum Ethernet Communication Adapter 170ENT11001 and 170ENT11000 User Guide*, 870USE11400.

10.3 Transfer Data Using IEC

Purpose

Overview

This section explains how to transfer data blocks using IEC.

What's in this Section?

This section contains the following topics:

CREAD_REG

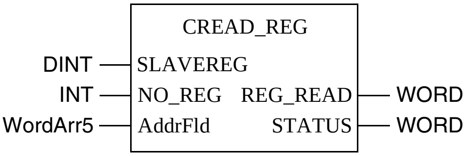
Function Description

This function block reads the register area continuously. It reads data from an addressed node via TCP/IP-Ethernet.
EN and ENO can be projected as additional parameters.

Note: When programming a CREAD_REG function, you must be familiar with the routing procedures used by your network.

Note: For technical reasons, this function block does not allow the use of programming languages ST and IL.

Representation



Parameter Description

Parameter	Data Type	Meaning
SLAVEREG	DINT	Offset address of the first 4x register in the slave to be read from
NO_REG	INT	Number of registers to be read from slave
AddrFld	WordArr5	Data structure describing the TCI/IP address
REG_READ	WORD	First 4x area register for read values
STATUS	WORD	Error code, see Runtime errors

Elementary Description for WordArr5 with TCP/IP Ethernet

Element	Data Type	Meaning
WordArr5[1]	WORD	Low value byte: MBP on Ethernet Transporter (MET) mapping index High value byte: Slot of the NWM module
WordArr5[2]	WORD	Byte 4 (MSB) of the 32-bit destination IP address
WordArr5[3]	WORD	Byte 3 of the 32-bit destination IP address
WordArr5[4]	WORD	Byte 2 of the 32-bit destination IP address
WordArr5[5]	WORD	Byte 1 (LSB) of the 32-bit destination IP address

Function Mode of the CREAD_REG Block

Although a large number of CREAD_REG function blocks can be programmed, only four read operations may be active at the same time. In such a case it is insignificant whether they are the result of this function block or others (e.g. MBP_MSTR, MSTR, READ_REG). All function blocks use one data transaction path and require multiple cycles to complete a job.

Note: A TCP/IP communication between a Quantum PLC (NWM) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only one read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

The entire routing information is contained in data structure WordArr5 of input AddrFld. The type of function block connected to this input and thus the contents of the data structure depends on the network used.
Please use:

- TCP/IP Ethernet the function block TCP_IP_ADDR

Note: For experts: The WordArr5 data structure can be used with constants as well.

Note: This function block puts a heavy load on the network; therefore the network load must be carefully monitored. If the network load is too high, the program logic should be reorganized in order to work with the READ_REG function block, a variation of this function block that does not operate in a continuous mode, but under command control.

SLAVEREG

Start of the area in the addressed slave from which the source data is read. The source area always resides within the 4x register area. SLAVEREG expects the source reference as offset within the 4x area. The leading "4" must be omitted (e.g. 59 (contents of the variables or value of the literal) = 40059).
The parameter can be specified as direct address, located variable, unlocated variable, or literal.

NO_REG

Number of registers to be read from the addressed slave (1 ... 100). The parameter can be entered as a Direct address, Located variable, Unlocated variable, or Literal. The parameter can be entered as a Direct address, Located variable, or Unlocated variable.

REG_READ

This word parameter addresses the first register in a series of NO_REG registers, listed one after the other, which are used as a destination data area. The parameter must be entered as a Direct address or located Variable.

STATUS

Error code, see Runtime errors.
The parameter can be specified as direct address, located variable or unlocated variable.

CWRITE_REG

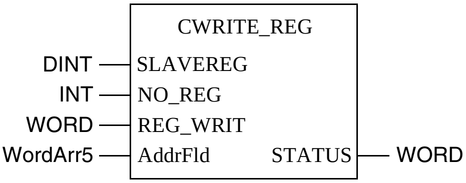
Function Description

The purpose of this function block is to write the register area continuously. It transfers data from the PLC via TCP/IP Ethernet to an addressed slave.
EN and ENO can be configured as additional parameters.

Note: You must be familiar with the routing procedures of the network when programming a CWRITE_REG function.

Note: For technical reasons, this function block does not allow the use of ST and IL programming languages.

Symbol



Parameter Description

Parameter	Data Type	Meaning
SLAVEREG	DINT	Offset address of the first 4x register in the slave to be written to
NO_REG	INT	Number of registers to be written to slave
REG_WRIT	WORD	First 4x register of the source data area
AddrFld	WordArr5	Data structure for transferring the TCI/IP address
STATUS	WORD	MSTR error code, see Runtime errors

Elementary Description for WordArr5 with TCP/IP Ethernet

Element	Data Type	Meaning
WordArr5[1]	WORD	Low value byte: MBP on Ethernet Transporter (MET) mapping index High value byte: Slots of the NWM module
WordArr5[2]	WORD	Byte 4 (MSB) of the 32-bit destination IP address
WordArr5[3]	WORD	Byte 3 of the 32-bit destination IP address
WordArr5[4]	WORD	Byte 2 of the 32-bit destination IP address
WordArr5[5]	WORD	Byte 1 (LSB) of the 32-bit destination IP address

CWRITE_REG Block Function Mode

Although a large number of CWRITE_REG function blocks can be programmed, only four write operations may be active at the same time. It makes no difference whether these operations are performed using this function block or others (e.g. MBP_MSTR, MSTR, WRITE_REG). All function blocks use one data transaction path and require multiple cycles to complete a job.
If several CWRITE_REG function blocks are used within an application, they must at least differ in the values of their NO_REG or REG_WRITE parameters.

Note: A TCP/IP communication between a Quantum PLC (NWM) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only one read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

The entire routing information is contained in data structure WordArr5 of input AddrFld. The type of function block connected to this input and thus the contents of the data structure depend on the network used.
Please use:

- TCP/IP Ethernet: the function block TCP_IP_ADDR

Note: For experts: The WordArr5 data structure can also be used with constants.

Note: This function block puts a heavy load on the network. The network load must therefore be carefully monitored. If the network load is too high, the program logic should be reorganized to work with the WRITE_REG function block, which is a variant of this function block that does not operate in continuous mode but is command driven.

SLAVEREG

Start of the area in the addressed slave to which the source data are written. The destination area always resides within the 4x register area. SLAVEREG expects the destination address as offset within the 4x area. The initial "4" must be omitted (e.g. 59 (contents of the variables or value of the literal) = 40059).
The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

NO_REG

Number of registers to be written to slave processor (1 ... 100). The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

STATUS

Error code, see Runtime errors.
The parameter can be specified as direct address, located variable or unlocated variable.

REG_WRIT

This word parameter addresses the first register in a series of NO_REG Successive registers used as source data area.
The parameter must be entered as a direct address or located variable.

READ_REG

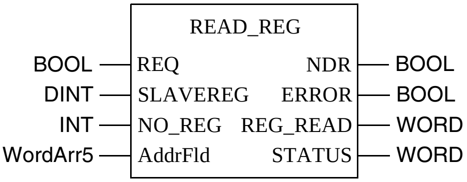
Function Description

If requested, this function block will read a register area once (rising edge of the REQ input). It reads data from an addressed slave via TCP/IP-Ethernet.
EN and ENO can be projected as additional parameters.

Note: You must be familiar with the routing procedures of your network when programming a READ_REG function.

Note: Note: For technical reasons, this function block does not allow use of the programming languages ST and IL.

Symbol



Parameter Description

Parameter	Data Type	Meaning
REQ	BOOL	Start read operation once
SLAVEREG	DINT	Offset address of the first 4x register in the slave to be read from
NO_REG	INT	Number of registers to be read from slave
AddrFld	WordArr5	Data structure describing the TCP/IP address
NDR	BOOL	Set to "1" for one cycle after reading new data
ERROR	BOOL	Set to "1" for one scan in case of error
STATUS	WORD	Error code, see Runtime errors
REG_READ	WORD	First 4x area register for read values

Elementary Description for WordArr5 with TCP/IP Ethernet

Element	Data Type	Meaning
WordArr5[1]	WORD	Low value byte: MBP on Ethernet Transporter (MET) mapping index High value byte: Slot of the NWM module
WordArr5[2]	WORD	Byte 4 (MSB) of the 32-bit destination IP address
WordArr5[3]	WORD	Byte 3 of the 32-bit destination IP address
WordArr5[4]	WORD	Byte 2 of the 32-bit destination IP address
WordArr5[5]	WORD	Byte 1 (LSB) of the 32-bit destination IP address

Function Mode of READ_REG Blocks

Although a large number of READ_REG function blocks can be programmed, only four read operations may be active at the same time. In such a case it is insignificant whether they are the result of this function block or of other read operations (e.g. MBP_MSTR, MSTR, CREAD_REG). All function blocks use one data transaction path and require multiple cycles to complete a job.

Note: A TCP/IP communication between a Quantum PLC (NWM) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only one read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

The entire routing information is contained in data structure WordArr5 of input AddrFld. The type of function block connected to this input and thus the contents of the data structure depends on the network used.
Please use:

- TCP/IP Ethernet the function block TCP_IP_ADDR

Note: For experts: The WordArr5 data structure can be used with constants as well.

REQ

A rising edge triggers the read transaction.
The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

SLAVEREG

Start of the area in the addressed slave from which the source data is read. The source area always resides within the 4x register area. SLAVEREG expects the source reference as offset within the 4x area. The leading "4" must be omitted (e.g. 59 (contents of the variables or value of the literal) = 40059).
The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

NO_REG

Number of registers to be read from the addressed slave (1 ... 100).
The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

NDR

Transition to ON state for one program cycle signifies receipt of new data ready to be processed.
The parameter can be specified as direct address, located variable, or unlocated variable.

ERROR

Transition to ON state for one program cycle signifies detection of a new error.
The parameter can be specified as direct address, located variable, or unlocated variable.

REG_READ

This word parameter addresses the first register in a series of NO_REG registers lying in series used as destination data area.
The parameter must be entered as a direct address or located variable.

STATUS

Error code, see Runtime errors.
The parameter can be specified as direct address, located variable or unlocated variable.

WRITE_REG

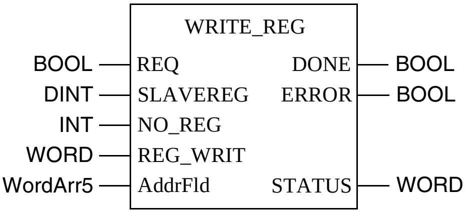
Function Description

If requested, this function block will write a register area once (rising edge of the REQ input). It transfers data from the PLC via TCP/IP Ethernet to an addressed slave.
EN and ENO can be configured as additional parameters.

> **Note:** You must be familiar with the routing procedures of your network when programming a WRITE_REG function.

> **Note:** For technical reasons, this function block does not allow the use of ST and IL programming languages.

Symbol



Parameter Description

Parameter	Data Type	Meaning
REQ	BOOL	Start write operation once
SLAVEREG	DINT	Offset address of the first 4x register in the slave to be written to
NO_REG	INT	Number of registers to be written from slave
AddrFld	WordArr5	Data structure transferring the TCP/IP address
REG_WRIT	WORD	First 4x register of the source data area
DONE	BOOL	Set to "1" for one scan after writing data
ERROR	BOOL	Set to "1" for one scan in case of error
STATUS	WORD	Error code, see Runtime errors

Elementary Description for WordArr5 with TCP/IP Ethernet

Element	Data Type	Meaning
WordArr5[1]	WORD	High value byte: Slot of the NWM module Low value byte: MBP on Ethernet Transporter (MET) mapping index
WordArr5[2]	WORD	Byte 4 (MSB) of the 32-bit destination IP address
WordArr5[3]	WORD	Byte 3 of the 32-bit destination IP address
WordArr5[4]	WORD	Byte 2 of the 32-bit destination IP address
WordArr5[5]	WORD	Byte 1 (LSB) of the 32-bit destination IP address

Function Mode of the WRITE_REG Module

Although a large number of WRITE_REG function blocks can be programmed, only four write operations may be active at the same time. In such a case it is insignificant whether they are the result of this function block or of other write operations (e.g. MBP_MSTR, MSTR, CWRITE_REG). All function blocks use one data transaction path and require multiple cycles to complete a job.
If several WRITE_REG function blocks are used within an application, they must at least differ in the values of their NO_REG or REG_WRITE parameters.

Note: A TCP/IP communication between a Quantum PLC (NWM) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only one read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

The status signals DONE and ERROR report the function block state to the user program.
The entire routing information is contained in data structure WordArr5 of input AddrFld. The type of function block connected to this input and thus the contents of the data structure depend on the network used.
Please use:

- TCP/IP Ethernet: the function block TCP_IP_ADDR

Note: For experts: The WordArr5 data structure can also be used with constants.

REQ

A rising edge triggers the read transaction.
The parameter can be specified as direct address, located variable or unlocated variable.

SLAVEREG

Start of the area in the addressed slave from which the source data is read. The source area always resides within the 4x register area. SLAVEREG expects the source reference as offset within the 4x area. The leading "4" must be omitted (e.g. 59 (contents of the variables or value of the literal) = 40059).
The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

NO_REG

Number of registers to be read from the addressed slave (1 ... 100).
The parameter can be specified as direct address, located variable, unlocated variable, or Literal.

REG_WRIT

This word parameter addresses the first register in a series of NO_REG registers used as source data area.
The parameter must be entered as a direct address or located variable.

DONE

Transition to ON state for one program scan signifies data have been transferred.
The parameter can be specified as direct address, located variable or unlocated variable.

ERROR

Transition to ON state for one program scan signifies detection of a new error.
The parameter can be specified as direct address, located variable or unlocated variable.

STATUS

Error code, see Runtime errors.
The parameter can be specified as direct address, located variable, or unlocated variable.

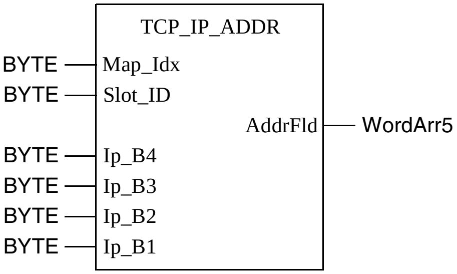
TCP_IP_ADDR

Function Description

This function block enables the input of TCP/IP addresses for the READ_REG, CREAD_REG, WRITE_REG and CWRITE_REG Function Blocks. The address is transferred in the form of a data structure.
The parameters EN and ENO can additionally be projected.

> **Note:** You must be familiar with your network when programming the TCP_IP_ADDR Function Block.

Symbol



Parameter Description

Parameter	Data Type	Meaning
Map_Idx	BYTE	Map-Index MBP on Ethernet Transporter (MET) mapping index
Slot_ID	BYTE	Slot ID Slot of the NWM module
Ip_B4	BYTE	Byte 4 (MSB) of the 32-bit destination IP address
Ip_B3	BYTE	Byte 3 of the 32-bit destination IP address
Ip_B2	BYTE	Byte 2 of the 32-bit destination IP address
Ip_B1	BYTE	Byte 1 (LSB) of the 32-bit destination IP address
AddrFld	WordArr5	Data structure used to transfer the TCP/IP address

Elementary Description for WordArr5

Element	Data Type	Meaning
WordArr5[1]	WORD	High value byte: Slot of the NWM module Low value byte: MBP on Ethernet Transporter (MET) mapping index
WordArr5[2]	WORD	Byte 4 (MSB) of the 32-bit destination IP address
WordArr5[3]	WORD	Byte 3 of the 32-bit destination IP address
WordArr5[4]	WORD	Byte 2 of the 32-bit destination IP address
WordArr5[5]	WORD	Byte 1 (LSB) of the 32-bit destination IP address

Map_Idx

The MBP on Ethernet Transporter (MET) mapping index is given at the Map_Idx input, i.e. if MET is 6, the value appears as follows.

0	0	0	0	0	1	1	0

Slot_ID

If an NWM in the rack of a Quantum controller is addressed as destination node, the value at the Slot_ID input represents the physical NWM slot, i.e. if the NWM is plugged in at Slot 7 of the rack, the value appears as follows.

0	0	0	0	0	1	1	1

AddrFld

If an NWM in the rack of a Quantum controller is addressed as a destination node, the value in the High value byte represents the physical slot of the NWM and the Low value byte represents the MBP on Ethernet Transporter (MET) mapping index, i.e. if the NWM is inserted in slot 7 of the rack and the MET mapping index is 6, the first element of the data structure looks as follows.

High value byte								Low value byte							
0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0

High value byte Slots 1 ... 16
Low value byte MBP on Ethernet Transporter (MET) mapping index

MBP_MSTR

Function Block

With this function block, it is possible to select one of 12 available network communication operations.
Although a large number of MBP_MSTR function blocks can be programmed, only four of them can be active at the same time. All function blocks use one data transaction path and require multiple cycles to complete a job.

Note: A TCP/IP communication between a Quantum PLC (NWM) and a Momentum PLC (all TCP/IP CPUs and all TCP/IP I/O modules) is only possible, when only one read or write job is carried out in every cycle. If several jobs are sent per PLC cycle, the communication stops without generating an error message in the status register of the function block.

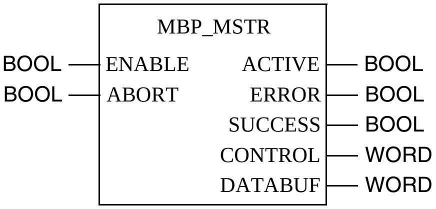
Note: In FBD and LD sections, the function block can only be used on the program level, i.e. not in Derived Function Blocks (DFBs).

Note: For technical reasons, this function block does not allow the use of programming languages ST and IL.

EN and ENO can be configured as additional parameters.

Note: You must be familiar with the routing procedures of your network when programming an MSTR function.

Symbol



Parameter Description

Parameter	Data Type	Meaning
ENABLE	BOOL	Enable MSTR function
ABORT	BOOL	Cancel active MSTR operation
ACTIVE	BOOL	Operation is active
ERROR	BOOL	Faulty operation
SUCCESS	BOOL	Operation completed successfully
CONTROL	WORD	First 4x register of the MSTR control block
DATABUF	WORD	First 4x register of the data field

Function Mode of MBP_MSTR Blocks

Using the MBP_MSTR block, one of 12 available network communication operations can be triggered via the network. Each operation receives a code. Whether the operations are available depends on the type of network used.

Valid Function Codes

Code	Function	TCP/IP Ethernet
1	Write Data	X
2	Read Data	X
3	Get Local Statistics	X
4	Clear Local Statistics	X
7	Get Remote Statistics	X
8	Clear Remote Statistics	X
10	Reset optional module	X
11	Read CTE (Config extension)	X
12	Read CTE (Config extension	X
23	Read/Write register	X

Legend

X	Yes
-	No

ENABLE

When ON, the operation specified in the first CONTROL register is enabled.

ABORT

When ON, the currently active operation is aborted.

ACTIVE

ON, if the operation is active.

ERROR

ON, if the operation was aborted without success.

SUCCESS

ON, if the operation concluded successfully.

DATABUF

The 4x register specified is the first in a group of successive output/marker words, making up the data field. For operations providing data, e.g. the write operation, the data field is the data source. For operations receiving data, e.g. the read operation, the data field is the data sink.
In the case of Ethernet CTE Read and Write operations, the middle input stores the contents of the Ethernet configuration extension table in a series of registers.

CONTROL

This word parameter addresses the first of several successive 4x registers. The control block is contained in these registers. The first register displayed contains a number from 1 to 12, which provides the operation code of the MODBUS operation to be performed. The contents of the sequence registers are determined by the operation.
The structure of the control block differs according to the network used:

- TCP/IP Ethernet

Control Block for TCP/IP Ethernet

Register	Contents
4x	Indicates one of the Operations which are valid for TCP/IP
4x + 1	Indicates the Error status
4x + 2	Indicates the length (number of registers transferred)
4x + 3	Indicates MSTR operation-dependent information
4x + 4	Routing register Low value byte: MBP on Ethernet Transporter (MET) mapping index High value byte: Slot of the NWM module
4x + 5	Byte 4 (MSB) of the 32-bit destination IP address
4x + 6	Byte 3 of the 32-bit destination IP address
4x + 7	Byte 2 of the 32-bit destination IP address
4x + 8	Byte 1 (LSB) of the 32-bit destination IP address

Routing Register (4x + 4) in TCP/IP Ethernet

If a NWM in the rack of a Quantum controller is addressed as destination node, the value in the high value byte represents the physical NWM slot and the value in the low value byte represents the MBP on Ethernet (MET) mapping index, i.e. if the NWM is plugged in at slot 7 of the rack and MET mapping index is 6, the first element of the data structure appears as follows.

High value byte								Low value byte							
0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0

High value byte Slots 1 to 16
Low value byte MBP on Ethernet Transporter (MET) mapping index

10.4 SNMP

Purpose

Overview

This section explains how to set up SNMP service.

What's in this Section?

This section contains the following topics:

SNMP

Overview

This following information describes the Simple Network Management Protocol (SNMP), which is configured on your NWM.

Introduction

Network management software allows a network manager to

- Monitor and control network components
- Isolate problems and find their causes
- Query devices such as a host computer, routers, switches, and bridges to determine their status
- Obtain statistics about the networks to which they attach

Manager/Agent Paradigm

Network management software follows the conventional client-server model.
To avoid confusion with other network communication protocols that use the client/server terminology, network management software uses the following terms:

- *Manager*
  For the client application that runs on the manager's computer
- *Agent*
  For the application that runs on a network device

The manager uses conventional transport protocols (e.g., TCP or UDP) to establish communication with the agent. Managers and agents then exchange requests and responses according to the network management protocol.

Simple Network Management Protocol

Your 140 NWM 100 00 module is configured with the Simple Network Management Protocol (SNMP), which is the standard protocol used to manage a local area network (LAN). SNMP defines exactly how a *manager* communicates with an *agent*. The SNMP defines the format of the requests that a manager sends to an agent and the format of the replies that the agent returns to the manager.

The MIB

Each object SNMP has access to has to be defined and given a unique name. Both the manager and agent program must agree on the names and the meanings of the fetch and store operations. The set of all objects SNMP can access is known as a *Management Information Base (MIB)*.

The Private MIB

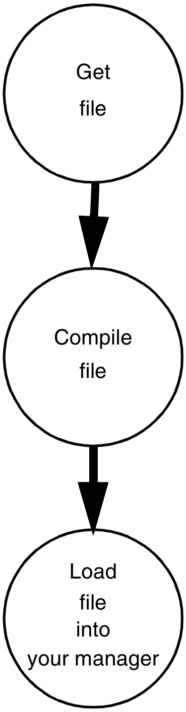
Schneider obtained a private MIB, Groupe_Schneider (3833). Under the Groupe Schneider private MIB is a Transparent Factory Ethernet (TFE) private MIB. The Transparent Factory SNMP embedded component controls the Schneider private MIB function.

Choosing a SNMP Manager

If you have a SNMP Manager already working, you may continue to use that SNMP Manager. If you are selecting a SNMP Manager, there are many SNMP Managers on the market, and you may use any of these managers. You must use a SNMP Version 1 compliant manager.

Using a SNMP Manager

Step	Action
1	Get Schneider `.mib` file from the NWM Web page. You are going to find the `.mib` file as a packed file under **/wwwroot/SchneiderTFE.zip** on your NWM module.
2	Compile `.mib` file in the compiler that comes with the NWM.
3	Load compiled `.mib` file to the SNMP manager.
4	When you are done, you will see the Schneider private MIB manager in your manager.



More SNMP Information

SNMP and related subjects are well documented on Web sites and in many books.

- As of this writing, a useful description appears on Microsoft's *Technet* pages. Browse to *http://www.microsoft.com/technet*. Use the **Search** function to find "Network Management for Microsoft Networks Using SNMP."
- Use an Internet search engine to search for a SNMP introduction, a SNMP tutorial, and other topics on SNMP.
- The SNMP FAQ from the news group `comp.protocols.snmp` appear on many `.com` and `.org` Web pages. Search for the combination of "`comp.protocols.snmp`" and "`FAQ`."
- A list of print books about SNMP appears in the SNMP FAQs. In addition, a search of most online retail book sites will yield a substantial list of titles.

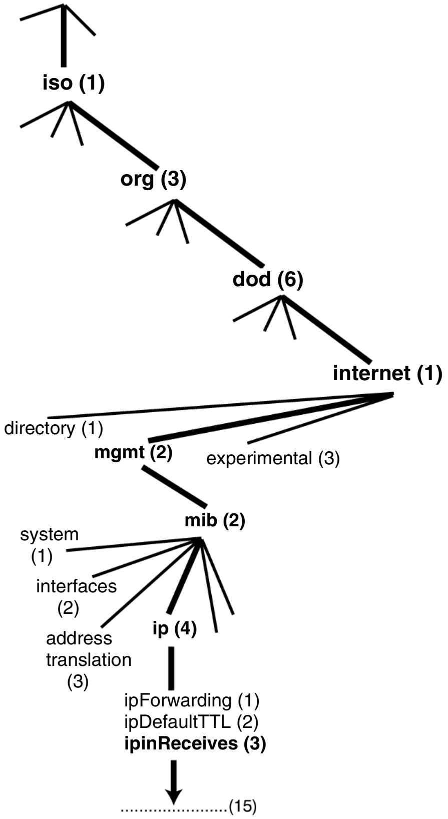
ASN.1 Naming Scheme

ASN.1 Overview

Abstract Syntax Notation One (ASN.1) is a formal language for abstractly describing messages to be exchanged between distributed computer systems.

An Example

Objects in a MIB are defined with the ASN.1 naming scheme that assigns each object a long prefix that guarantees that the name will be unique. For example, an integer that counts the number of IP datagrams that a device has received is named: *iso.org.dod.internet.mgmt.mib.ip.ipinReceives*.
The following figure depicts the ASN.1 Naming Scheme example.



This object name is represented in an SNMP message by assigning each part an integer. So, the above message would appear as 1.3.6.1.2.2.4.3.
Each integer has the following meaning.

- 1 = ISO (International Organization for Standardization)
- 3 = identified organization — one of branches under the ISO root

- 6 = U. S. Department of Defense (DOD) — one of the children under branch1.3
- 1 = the Internet subtree under 1.3.6
- 2 = the mgmt branch — (one of seven) of the Internet subtree. It is managed by the Internet Assigned Numbers Authority, and includes the standard MIBs
- 2 = mib-2(1) group of managed objects
- 4 = ip — the mib-2(1) IP group (one of 11)
- 3 = ipinReceives — the MIB object

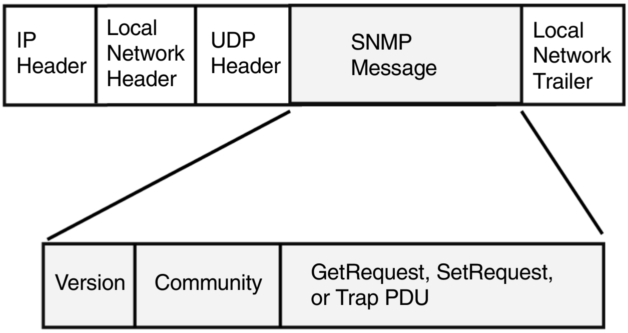
Configuring the NWM with SNMP

The Object Identifier (OID)

In the ASN.1 Naming Scheme example, the MIB object identified by the notation 1.3.6.1.2.2.4.3 is referred to as the Object Identifier or OID. All OIDs can be envisioned as part of a tree structure which begins at the root (ISO) and branches out with each subtree identified by an integer.

SNMP Protocol Data Units

SNMP uses Protocol Data Units (PDUs) to carry the requests and responses, between the manager and the agents, for the information contained in an OID.
As the following figure shows, the SNMP message is the innermost part of a typical network transmission frame.



The PDUs within the SNMP initiate the communication between the manager and the agents.
The SNMP installed on your NWM module uses the following three PDUs.

- GetRequest
- SetRequest
- Trap

GetRequest PDU

The GetRequest (shortened to Get) PDU is used by the SNMP manager to retrieve the value of one or more objects (OIDs) from an agent.

SetRequest PDU

The SetRequest (shortened to Set) PDU is used by the SNMP manager to assign a value to one or more objects (OIDs) residing in an agent.

Trap PDU

The Trap PDU is used by the agent to alert the manager that a predefined event has occurred.

Version & Community Identifiers

The version identifies the version number of the SNMP software being used by the manager and the agent. Your NWM supports Version 1 of the SNMP. The community is an identifier that you assign to your SNMP network. If community names for the manager and the agent do not agree, the agent will send an authentication failure trap message to the manager. If the community names and version number agree, the SNMP PDU will be processed.

What Can Be Configured

Your NWM module can be configured to send an authentication trap to two SNMP managers if it receives a community name in a Get/Set request that does not match the configured name. Also, you can configure the SysContact and SysLocation via the configuration page in the module's Embedded Web pages. After making changes in the SNMP Configuration Web page and to set those changes, reboot the module using hot swap.

Configuring the NWM with TFE Private MIB

Introduction

An MIB, Management Information Base, is an element used in network management. Network management services are based on the need to monitor and manage the following.

- Performance
- Fault occurrences
- Security

Each MIB contains a finite number of objects. Manage your MIB with a management station running an SNMP management application. The management application uses **GETs** and **SETs** to retrieve system information and to set system environment variables.

Schneider Private MIB

Schneider Automation obtained a Private Enterprise Number (PEN) from the Internet Assigned Numbers Authority (IANA). That number represents a subtree in the SNMP MIB, a number that is a unique identifier used for Groupe Schneider. The object identifier for the root of the Groupe Schneider subtree is `1.3.6.1.4.1.3833` and represents a path to the subtree as follows.

```
iso(1)
        org(3)
                dod(6)
                        internet(1)
                               private(4)
                                      enterprise(1)
                                             GroupeSchneider(3833)
                                                        Transparent_Factory_Ethernet(1)
```

Under the GroupeSchneider private MIB is a Transparent Factory Ethernet (TFE) private MIB, **Transparent_Factory_Ethernet(1)**.

TFE Private MIB

The Transparent Factory SNMP-embedded component controls the Schneider private MIB function. The Schneider private MIB, and associated services, perform Network Management on all components of the system. The Transparent Factory private MIB provides the data to manage the main Transparent Factory communication services for all the communication components of the Transparent Factory architecture. The Transparent Factory private MIB does not define the specific management applications and policies.

The **Groupe_Schneider (3833)** subtree is the root of Groupe Schneider's private MIB in the Structure of Management Information (SMI) used by SNMP and defined in RFC-1155, which is a specification that defines the structure and identification of management information for TCP/IP-based Internets.

Transparent Factory Ethernet Subtree

The **Transparent_Factory_Ethernet** subtree defines groups that support the Transparent Factory Ethernet services and devices.

Service	Description
Port502_Messaging (2)	Subtree defines objects for managing explicit client / server communications supporting applications, such as HMI, SCADA, or programming tools
Web (5)	Subtree defines objects for managing the activity of the embedded Web servers
Equipment_Profiles)	Subtree identifies objects for each type of device in Transparent Factory Ethernet's product portfolio

Port502 Messaging Subtree

The Port502_Messaging subtree, or group, provides connection management and data flow services. The following list describes the function of each object.

Service	Description
port502Status(1)	Indicates the status of the service (Idle, Operational)
port502SupportedProtocol(2)	Indicates the supported protocols (MODBUS, Xway)
port502IpSecurity(3):	Indicates the status of the Port502 IP Security service (enabled/disabled)
port502MaxConn(4)	Indicates the maximum TCP connection number supported by the Port502 entity
port502LocalConn(5)	Indicates the TCP connection number currently opened by the local Port502 entity
port502RemConn(6)	Indicates the TCP connection number currently opened by the remote entity to the local Port502 entity
port502IpSecurityTable(7)	Indicates a table containing the number of unsuccessful TCP connection open tries from a remote TCP entity
port502ConnTable(8)	Indicates a table containing Port502 TCP specific information (MsgIn, MsgOut)
port502MsgIn(9)	Indicates the total number of Port502 messages received from the network
port502MsgOut(10)	Indicates the total number of Port502 messages sent from the network
port502MsgOutErr(11)	Indicates the total number of error messages built by the Port502 messaging entity and sent to the network
port502AddStackStat(12)	Indicates the support of additional port502 stack statistics 1 - Disabled 2 - Enabled
port502AddStackStatTable(13)	Indicates additional stack statistics for Port502 (optional)

Web Subtree

The Web subtree, or group, contains the objects related to the Web server service.

Service	Description
webStatus(1)	Indicates the global status of the Web service 1 - Idle 2 - Operational
webPassword (2)	Indicates a switch to enable or disable the use of Web passwords 1 - Disabled 2 - Enabled
webSuccessfullAccess (3)	Indicates the total number of successful accesses to the Web site
webFailedAttempts (4)	Indicates the total number of unsuccessful accesses to the Web site

Equipment Profile Subtree

The Equipment_Profiles subtree contains a set of common objects.

Service	Description
profileProductName(1)	Displays the commercial name of the NWM 100 00
profileVersion(2)	Displays the software version of the communication product in a string form (for example: Vx.y or V1.1)
profileCommunicationServices(3)	Displays a list of the communication services supported by the profile (Port502Messaging and Web.)
profileConfigMode(5)	Indicates the IP configuration mode of the communication module 1 - Local: The IP configuration is created locally 2 - dhcpServed: The IP configuration is created by a remote DHCP server
profileRoleName(6)	Indicates the role name for the IP address management if it exists (Empty string if there is none)
profileLedDisplayTable(9)	Displays a table giving the name and the state of each module's LEDs
profileSlot(10)	Indicates the position of the communication module inside the rack if there is one. If there is no rack, the profileSlot value will be zero
profileCPUType(11)	Indicates that if the CPU type exists, this variable identifies the host for which that communication module is a part. If there is no host, the string is empty
profileTrapTableEntriesMax(12)	Indicates the maximum numbers of entries in the Trap Table. This entry equals the number of possible remote managers
profileTrapTable(13)	Displays a table allowing you to enable or disable the private traps for each of the communication services
profileSpecificId(14)	Indicates a unique Profile Specific Identification inside the equipmentProfile object of the Schneider Transparent Factory MIB. (For example the PLC Premium family is 100)
profileIpAddress(15)	Indicates the IP address of the SNMP agent
profileIpNetMask(16)	Indicates the subnet mask associated with the IP address of the SNMP agent. The value of the mask is an IP address with all the network bits set to 1 and all the host bits set to 0
profileIpGateway(17)	Indicates the default Gateway IP address of the SNMP agent

Service	Description
profileMacAddress(18)	Indicates the Ethernet media-dependent address of the SNMP agent

Private Traps and MIB Files

Traps are used to signal Status Changes to the manager. Using traps helps to avoid unnecessary network traffic.

The four status changes signaled by the trap are for:

- LEDs
- Communication Ports

The following list describes the characteristics of private traps, which means that they can:

- Send messages to the two managers whose IP addresses are configured in the SNMP configuration (either the PL7, Unity Pro, or the Web page)
- Use the community name given to this configuration
- Enable or disable each of the Transparent Factory Ethernet Private MIB groups: Switch (1), Port502_Messaging (2), Web (3), Equipment_Profiles(4)

Private traps are described in the MIB ASN.1 description, which is contained in a **`.mib`** text file.

10.5 Maintenance

Purpose

Overview

This section deals with performing maintenance on the NWM.

What's in this Section?

This section contains the following topics:

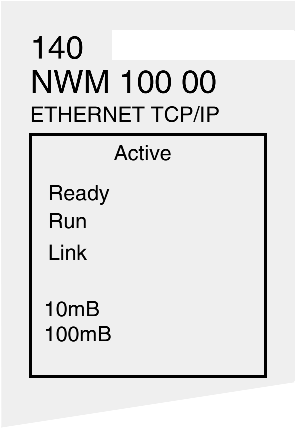
Responding to Errors

Overview

The following information describes how to respond to errors on the 140 NWM 100 00.

Detecting Errors

When faults occur, the 140 NWM 100 00 LED display can help you determine what went wrong. The following figure shows the pattern that the LEDs should display during normal operation.



The **Run** indicator will be solid. The **Coll** LED may flash, indicating that collisions are occurring on the Ethernet network. Such collisions are normal.

If a fault occurs, the normal LEDs may be extinguished or other indicators may light. This section will discuss errors reported by the **Active**, **Ready**, **Coll**, **Link**, **Kernel**, **Appl** and **Fault** indicators.

For each type of error, try the suggested remedies in the order given. If no remedy suggested here overcomes the error, call your local service representative or Schneider Electric customer service for further directions.

Procedure for Responding to an Active LED Error Indicator

If the Active LED fails to light, the 140 NWM 100 00 module is not communicating with the backplane. The following procedure describes the steps to perform to respond to an Active LED error.

Step	Action
1	Make sure the NNWM module and the controller are installed properly.
2	Verify that the controller is working; if it is not, replace it.
3	If neither the new controller nor the NWM module functions, replace the backplane.
4	Make sure that no more than two network option modules -- including NWM, NOE, NOP, and CRP 811 modules -- have been installed in the backplane with a 140 CPU 113 or 213; not more than six network option modules with a 140 CPU 424, 534 or 140 CPU 6xx.
5	Check the version of the controller executive. You must have version 2.0 or greater to support the Ethernet module. Earlier versions do not recognize the module.
6	If steps 4 and 5 above do not resolve the problem, replace the NWM module.

Procedure for Responding to a Ready LED Error Indicator

If the **Ready** LED fails to light, the 140 NWM 100 00 module has failed internal diagnostic tests. The following procedure describes the steps to perform.

Step	Action
1	Make sure that power has been applied to the backplane.
2	If step 1 checks out ok, replace the NWM module.

Procedure for Responding to a Link LED Error Indicator

If the **Link** LED fails to light, the 140 NWM 100 00 module is not communicating with the Ethernet hub/switch. The following procedure describes the steps to perform to respond to a **Link** LED error.

Step	Action
1	Make sure that the cable has been installed correctly and the module is functioning properly.
2	Verify that the hub/switch is working properly.
3	If steps 1 and 2 check ok, replace the NWM module.

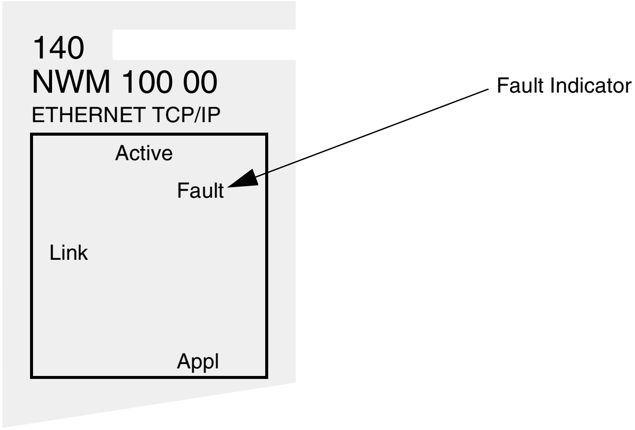
Kernel LED Error

Kernel LED errors that may occur and how to respond to them.

If	Then
The **Ready** LED is on and the **Kernel** LED is flashing	the module has detected an invalid software image.
The **Ready** LED is on and the **Kernel** LED is shining steadily,	the module is not in kernal mode; it is in a fault mode and customer support should be contacted.
Either of the above conditions exists.	download ftp and a new NWM Exec.

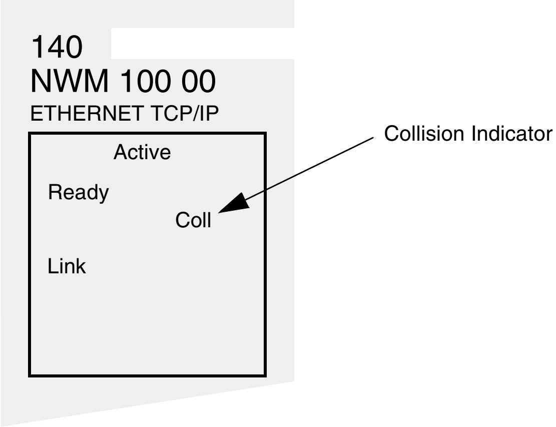
Fault LED

The **Fault** LED can flash briefly following an error as the module attempts to recover. The following figure shows the **Fault** LED.



Collision LED Error

The following figure shows the Collision LED.



Procedure for Responding to a Collision LED Error

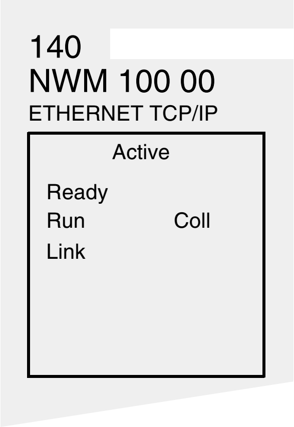
If the Collision LED fails to light, use the following procedure.

Step	Action
1	Make sure that the cable has been installed and is working properly.
2	Verify that the Ethernet Hub/Switch is working properly.

Collision LED Normal Condition

If the **Coll** LED is flashing, the module is reporting collisions on the Ethernet network. While such collisions are normal, the frequency of the flashes is an indication of the volume of traffic on the network. The flashes may be so frequent that the LED appears to be shining steadily. Heavy collisions will slow communications. If response time is important to your application, you should consider segmenting your network to reduce the frequency of collisions.

The following figure shows the Collision LED under normal conditions.



Run LED

The following table describes the action to be taken if the **Run** LED is flashing. The action depends on the number of flashes in sequence.

# of Flashes in Sequence	Action
One	Not used
Two	Not used
Three	No Link:The network cable is not connected or is defective
Four	Duplicate IP address
Five	No IP address: The module is attempting to obtain an IP address from the BOOTP server.
Six	Using default IP address
Seven	No valid executive NWM present
Eight	Invalid IP configuration (Likely cause; Default gateway is not on the same subnet mask as the module.
Nine	Flash file system is corrupted

Application LED

If the module crashes, it will note the reason in a log. If the module is able to recover, the **Appl** LED will light, indicating that an entry has been made in the crash log.

Reading and Clearing the Crash Log

Overview

The following information describes the crash log.

Introduction

The crash log provides you with the ability to capture conditions that lead to an anomalous condition. By providing the crash log to Schneider Electric technical support, you can facilitate their assistance in resolving your problems.

> **Note:** The crash log is provided with the understanding that, with a complex product in thousands of customer applications, there may be conditions that require advance diagnostics. The crash log is one of the tools used to solve complex problems.

The Crash Log

If the **Appl** indicator is on, entries have been made in the crash log. The log may hold up to 64K of entries.

Reading the Crash Log

The crash log can be read from the Embedded Web Pages or via FTP.

Procedure for Reading the Crash Log via FTP

The following procedure describes the steps to perform to access the crash log via FTP.

Step	Action
1	Log into the module’s FTP server with a login ID and default password.
2	Change the directory to *wwwroot/conf/diag*
3	Perform an FTP to get the crash log file: `get crash.log`

Clearing the Crash Log

The crash log can be cleared from the Embedded Web Pages or via FTP.

Procedure for Clearing the Crash Log via FTP

The following procedure describes the steps to perform to access the crash log via FTP.

Step	Action
1	Log into the module’s FTP Server.
2	Change the directory to *wwwroot/conf/diag.*
3	Perform an FTP to delete the crash log file: `rm crash.log`

Downloading a New NWM Exec

Introduction

The following tools can be used to download a new 140 NWM 100 00 Exec.

- Schneider Electric programming packages (see 840USE49300 Concept User Manual)
- FTP

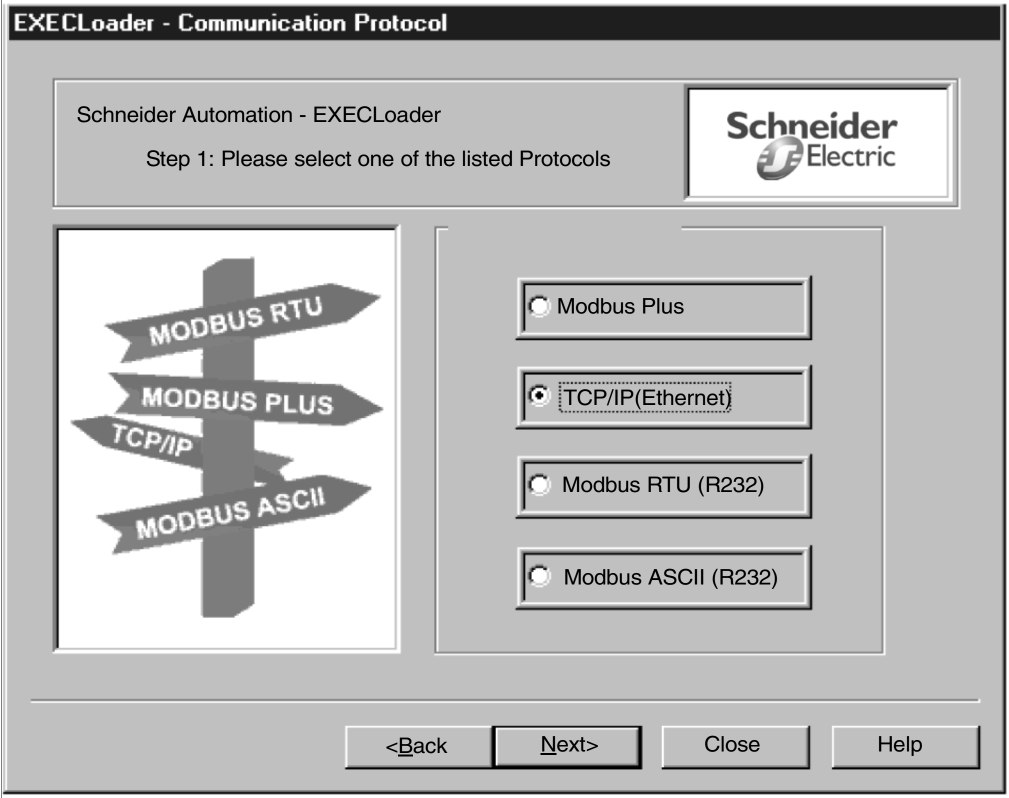
The Concept EXECLoader

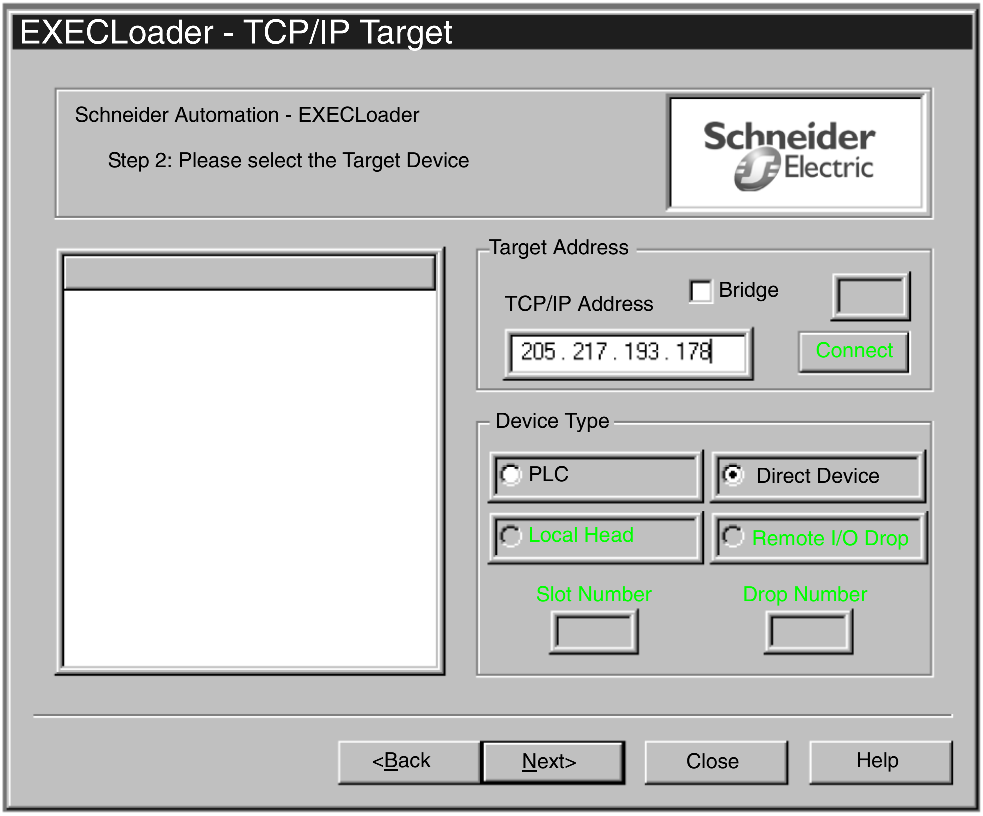
Overview

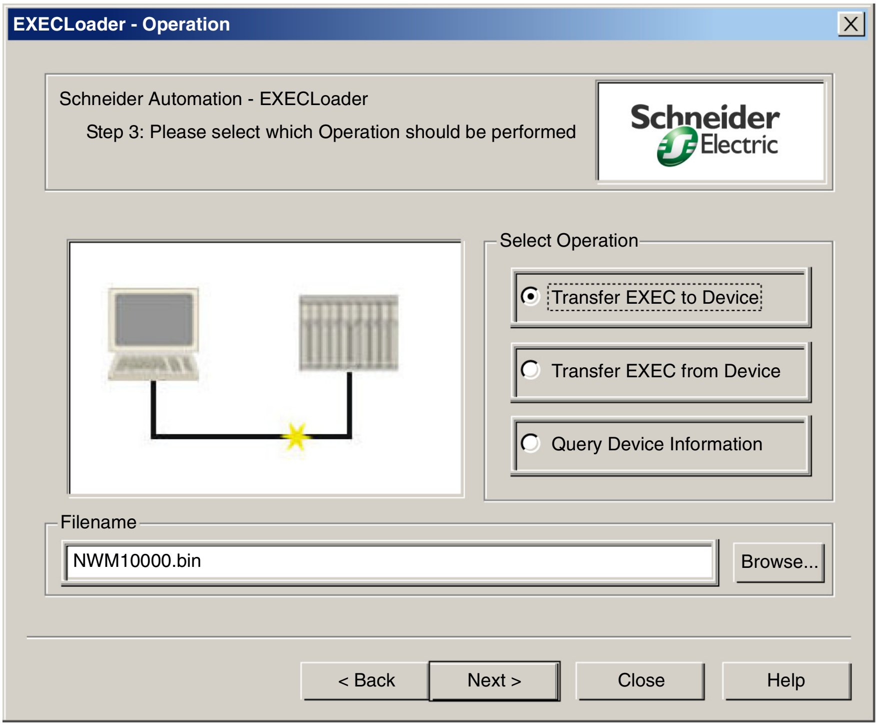
This section describes the use of the Concept EXECLoader, for downloading a new 140 NWM 100 000 Exec.

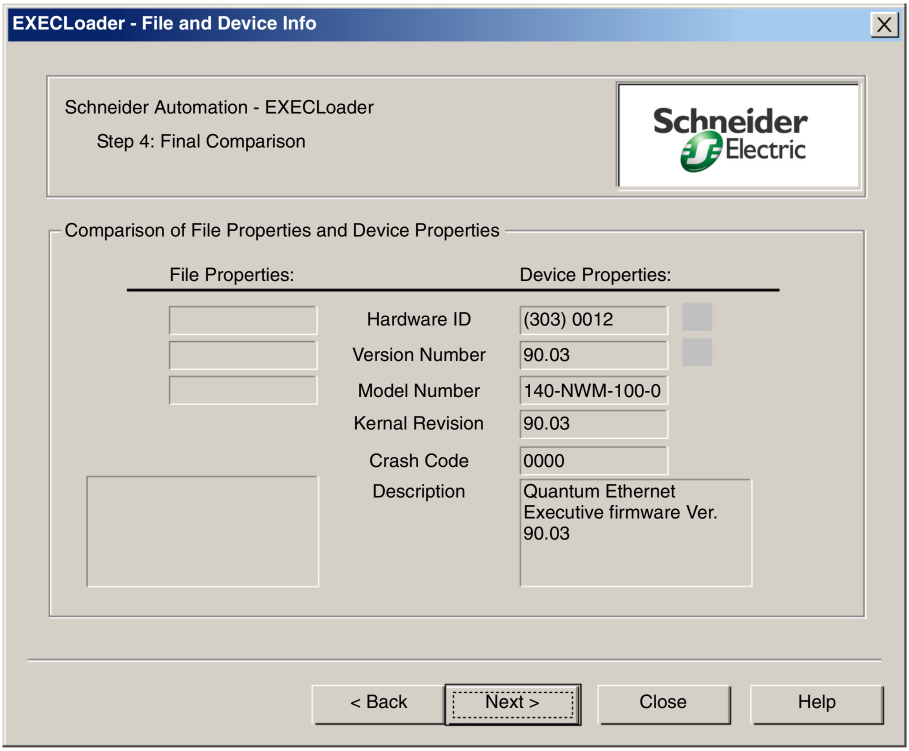
Procedure for Downloading NWM Exec

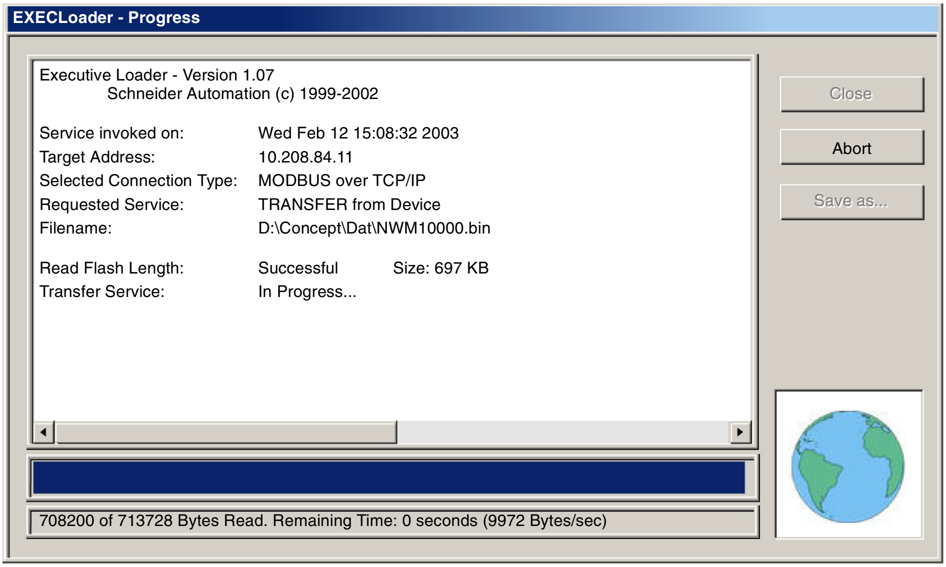
The following procedure provides the steps involved in downloading a new 140 NWM 100 00 Exec, using the Concept EXECLoader.

Step	Action
1	Activate the Exec Loader program.
2	Click on the **Next** button to start the Exec Loader process. **Result:** As shown in the following figure, the EXECLoader - Communication Protocol screen is displayed.  EXECLoader - Communication Protocol Schneider Automation - EXECLoader Step 1: Please select one of the listed Protocols ○ Modbus Plus ● TCP/IP(Ethernet) ○ Modbus RTU (R232) ○ Modbus ASCII (R232) MODBUS RTU MODBUS PLUS TCP/IP MODBUS ASCII <Back Next> Close Help Click on **TCP/IP [Ethernet**.}

Step	Action
3	Click on the **Next** button **Result**: As shown in the following figure, the EXECLoader - TCP/IP Target screen is displayed.  EXECLoader - TCP/IP Target Schneider Automation - EXECLoader Step 2: Please select the Target Device Target Address TCP/IP Address: 205 . 217 . 193 . 178 □ Bridge Device Type ○ PLC ● Direct Device ○ Local Head ○ Remote I/O Drop Slot Number Drop Number Schneider Electric Connect <Back Next> Close Help Click on **Direct Device**.

Step	Action
4	Click on the **Next** button. **Result**: As shown in the following figure, the EXECLoader - Operation screen is displayed.  EXECLoader - Operation Schneider Automation - EXECLoader Step 3: Please select which Operation should be performed Select Operation ● Transfer EXEC to Device ○ Transfer EXEC from Device ○ Query Device Information Filename: NWM10000.bin Schneider Electric Browse... < Back Next > Close Help Click on **Transfer EXEC to Device**.
5	Use Browser to select the file name.

Step	Action
6	Click on the **Next** button. **Result**: The EXECLoader - File and Device Info screen is displayed.  EXECLoader - File and Device Info Schneider Automation - EXECLoader Step 4: Final Comparison Comparison of File Properties and Device Properties File Properties: Device Properties: Hardware ID (303) 0012 Version Number 90.03 Model Number 140-NWM-100-0 Kernal Revision 90.03 Crash Code 0000 Description Quantum Ethernet Executive firmware Ver. 90.03

Step	Action
7	Click on the **Next** button. **Result**: As shown in the following figure, The EXECLoader - Progress screen is displayed.  EXECLoader - Progress Executive Loader - Version 1.07 Schneider Automation (c) 1999-2002 Service invoked on: Wed Feb 12 15:08:32 2003 Target Address: 10.208.84.11 Selected Connection Type: MODBUS over TCP/IP Requested Service: TRANSFER from Device Filename: D:\Concept\Dat\NWM10000.bin Read Flash Length: Successful Size: 697 KB Transfer Service: In Progress... 708200 of 713728 Bytes Read. Remaining Time: 0 seconds (9972 Bytes/sec) When the process is completed, you can click on the **Close** button.

Downloading a New NWM Exec via FTP

Exec Version

Please check the current 140 NWM 100 00 Exec file version on the **NWM Properties** Web page.
Follow these links:| **Web Server** | **Diagnostics and Online Configuration** | **NWM Properties** |

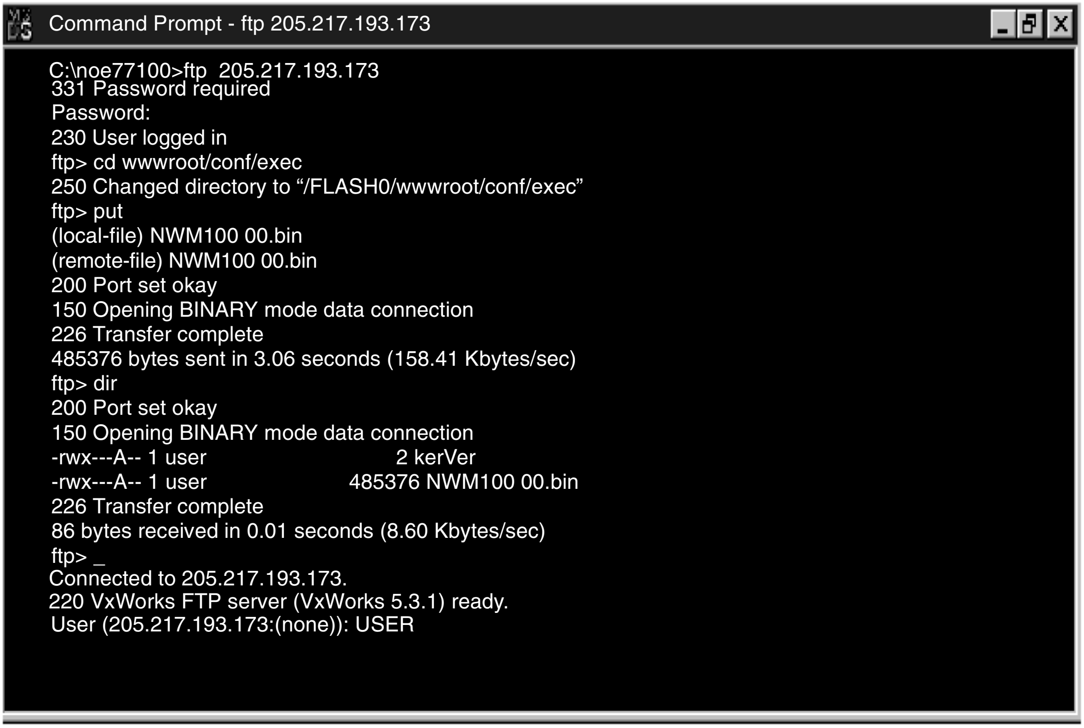
Procedure

The following procedure describes the steps to use to download a new 140 NWM 100 00 Exec via FTP. An example follows the procedure.

Step	Action
1	At the DOS prompt, type `FTP`, followed by the IP Address of the 140 NWM 100 00 and press **Enter**.
2	At the User prompt, type: `USER` and press **Enter**.
3	At the password prompt, enter your FTP Password and press **Enter**.
4	At the FTP prompt, type `cd wwwroot/conf/exec` and press **Enter**.
5	At the FTP prompt, type `put` and press **Enter**. **Note:** Pay attention that the NWM bin has to be to the local path on your PC (default path: `c:\`).
6	At the local file prompt, type `NWM100 00.bin` and press **Enter**.
7	At the remote file prompt, type `NWM100 00.bin` and press **Enter**.
8	After the transfer is complete you must reboot the NWM to allow the new EXEC to become operational. **Note:** The file name is case sensitive and must be entered with the name in uppercase and the extenion in lowercase as shown in the figure below. For example: `NWM100 00.bin`

Sample FTP Session

The following FTP session was used to download an 140 NWM 100 00 Exec. If you wish to use a graphical FTP client, Schneider Electric recommends WSFTP from Ipswitch Inc.



```
C:\noe77100>ftp  205.217.193.173
331 Password required
Password:
230 User logged in
ftp> cd wwwroot/conf/exec
250 Changed directory to “/FLASH0/wwwroot/conf/exec”
ftp> put
(local-file) NWM100 00.bin
(remote-file) NWM100 00.bin
200 Port set okay
150 Opening BINARY mode data connection
226 Transfer complete
485376 bytes sent in 3.06 seconds (158.41 Kbytes/sec)
ftp> dir
200 Port set okay
150 Opening BINARY mode data connection
-rwx---A-- 1 user                      2 kerVer
-rwx---A-- 1 user                 485376 NWM100 00.bin
226 Transfer complete
86 bytes received in 0.01 seconds (8.60 Kbytes/sec)
ftp> _
Connected to 205.217.193.173.
220 VxWorks FTP server (VxWorks 5.3.1) ready.
User (205.217.193.173:(none)): USER
```

Reboot Information after FTP

CAUTION

You may wish to stop your controller since communication with the NWM will be lost during the reboot cycle not to exceed 90 seconds.

Be sure to reboot the module after downloading by FTP.

Failure to follow this precaution can result in injury or equipment damage.

Downloading a New 140 NWM 100 00 Kernel

Overview

The 140 NWM 100 00 Executive provides the ability to allow the updating of the low level Kernel within the140 NWM 100 00's firmware. The following procedure should be followed to assure the proper installation of new Kernel firmware.

Note: The 140 NWM 100 00 Kernel can not be downloaded via FTP.

CAUTION

NWM Operation

Failure to perform this operation will render the NWM inoperable.

Failure to follow this precaution can result in injury or equipment damage.

Kernel Version

Please check the current 140 NWM 100 00 Kernel version on the **NWM Properties** Web page.
Follow these links:| **Web Server** | **Diagnostics and Online Configuration** | **NOE Properties** |

Procedure

Follow these steps to download a 140 NWM 100 00 Kernel.

⚠ **WARNING**

Interrupting the download of the kernel or power cycling of the 140 NWM 100 00 before the completion of the new kernel burn will result in the need to return the module to Schneider Electric customer support for repair.

Failure to follow this precaution can result in death, serious injury, or equipment damage.

Step	Action
1	Check the current version of the NWM's Executive firmware (Exec file), using the Exec Loader.
2	Loading the Kernel firmware is performed using the EXECLoader in the same manner as the Executive firmware.
3	After the transfer comes up as successful, the NWM needs approximately two minutes to burn the new Kernel into the NWM's flash and will then go through a reboot sequence.

Configuration of the TSX WMY 100 module running PL7

11

At a Glance

Subject of this Chapter

This chapter describes installation of the TSX WMY 100 module in the PL7 software workshop.

Note: The functional level of the TSXWMY 100 module requires version V4.4 of the PL7 software.

What's in this Chapter?

This chapter contains the following sections:

11.1 General

At a Glance

Subject of this Section

This section introduces ETHERNET communication from a **TSX WMY 100** HMI module and its properties.

What's in this Section?

This section contains the following topics:

General

At a Glance

The ETHERNET communication channel of the **TSX WMY 100** HMI module provides connection to a TCP/IP network supporting the UNI-TE messaging service and Modbus on a TCP/IP profile.

It also offers the following services:

- FTP and SNMP messaging
- Embedded HTTP server
- DHCP Client
- The Replace faulty device service, which allows the device configuration to be saved in the module. In case of breakdown, another installed blank module can be restarted with the configuration parameters of the previous one.

In addition, the **TSX WMY 100** module offers the option of creating user Web pages.

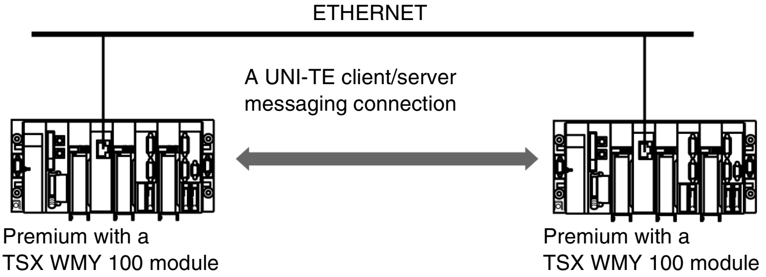
Type of supported connections

At a Glance

The **TSX WMY 100** HMI module authorizes a maximum of 16 parallel HTTP connections and a maximum of 64 parallel TCP/IP connections.

Connection to a TSX Premium

Between two TSX Premium PLCs using UNI-TE messaging on the TCP/IP profile, the **TSX WMY 100** module only allows one connection in client/server mode.

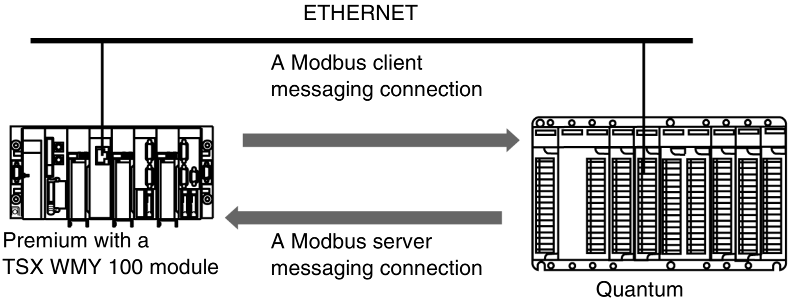

ETHERNET
A UNI-TE client/server messaging connection
Premium with a TSX WMY 100 module
Premium with a TSX WMY 100 module

Connection to a TSX Quantum

Between a TSX Premium PLC and a TSX Quantum PLC using Modbus messaging on the TCP/IP profile, the **TSX WMY 100** module allows:

- a single connection in Modbus client mode,
- a single connection in Modbus server mode.

Example

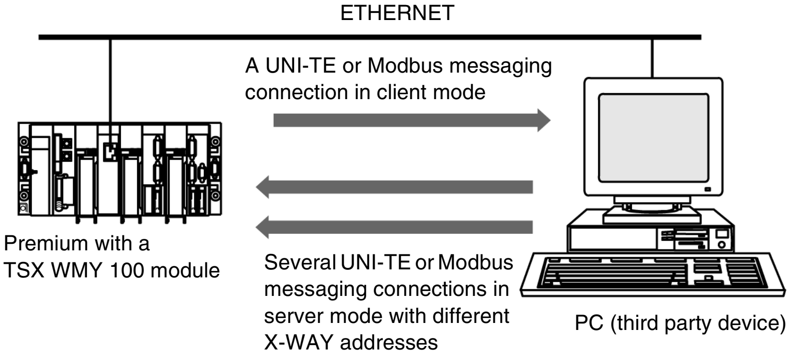

ETHERNET
A Modbus client messaging connection
Premium with a TSX WMY 100 module
A Modbus server messaging connection
Quantum

Connection to a Third Party Device

Between a TSX Premium PLC and a third party device, the **TSX WMY 100** module allows:

- a single connection in UNI-TE or Modbus client mode,
- several connections in UNI-TE or Modbus server mode.

Example

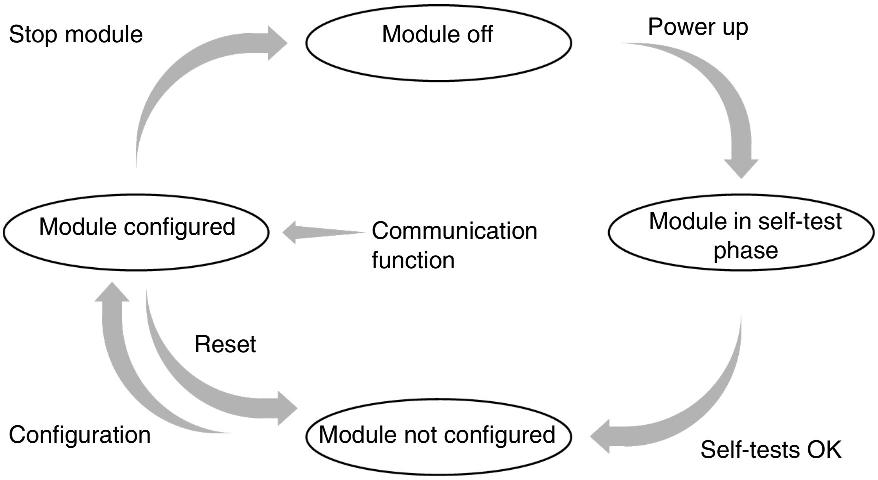

ETHERNET
A UNI-TE or Modbus messaging connection in client mode
Premium with a TSX WMY 100 module
Several UNI-TE or Modbus messaging connections in server mode with different X-WAY addresses
PC (third party device)

Operating modes of the TSX WMY 100 module

At a Glance

The following diagram describes the **TSX WMY 100** HMI module operating modes.

General Diagram



Operation

- After power-up, the module carries out self-testing. During this phase, the STS LED is lit. After the self tests, the RUN LED lights up.
- The module does not operate with a default configuration. This has to be transmitted to it by the local PLC's PL7 application.
  The configuration values are given in the %KW language object list and in particular provide the IP address and the X-WAY address of the module.
  The configuration is retransmitted upon every PLC restart (warm or cold).
- When the configuration is received, the module resets the current communication to zero before configuring itself (terminates current exchanges, shuts down TCP connections).
  The module is now operational. The RUN and STS LEDs are lit up.
- The IO Scanning function starts when the PLC transfers to RUN mode. It stops when it transfers to STOP mode.

Special Cases

If the module is not configured in the PL7 application (RUN LEDs go out and ERR LEDs flash), it takes the IP address constructed from its MAC address: 085.016.xxx.yyy where xxx and yyy are the last two numbers of the MAC address.

Example

The module's MAC address is (in hexadecimal): 00 80 F4 01 12 20
In this case the default IP address is (in decimal): 085.016.018.032

The module also transmits BOOTP/DHCP requests to obtain another IP address. These requests are sent until a BOOTP/DHCP server responds or until configured via PL7.

If the module detects a duplicated IP address, the STS LED flashes 4 times during a minute (as does that of the module having the same IP address).

11.2 Configuration

At a Glance

Subject of this Section

This section describes installation of the **TSX WMY 100** HMI module during its configuration using PL7.

What's in this Section?

This section contains the following topics:

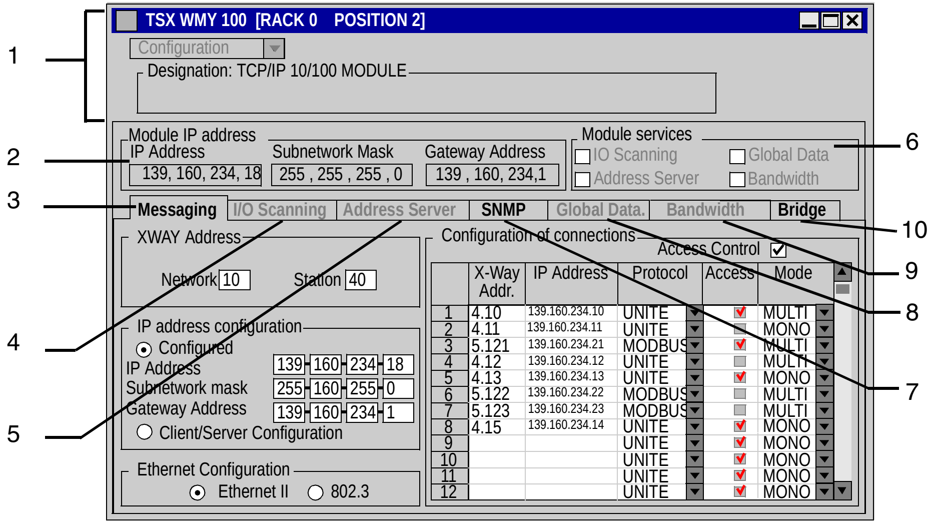
Module configuration screen

At a Glance

This screen, separated into two zones, allows the declaration of the communication channel and the configuration of the necessary parameters for an ETHERNET link.

Illustration

The screen dedicated to ETHERNET communication is displayed as follows:



TSX WMY 100 [RACK 0 POSITION 2]

Configuration

Designation: TCP/IP 10/100 MODULE

Module IP address

IP Address	Subnetwork Mask	Gateway Address
139, 160, 234, 18	255 , 255 , 255 , 0	139 , 160, 234,1

Module services

□ IO Scanning □ Global Data
□ Address Server □ Bandwidth

Messaging | I/O Scanning | Address Server | SNMP | Global Data. | Bandwidth | Bridge

XWAY Address

Network 10 Station 40

IP address configuration

● Configured

IP Address 139.160.234.18
Subnetwork mask 255.160.255.0
Gateway Address 139.160.234.1

○ Client/Server Configuration

Ethernet Configuration

● Ethernet II ○ 802.3

Configuration of connections

Access Control ☑

	X-Way Addr.	IP Address	Protocol	Access	Mode
1	4.10	139.160.234.10	UNITE	☑	MULTI
2	4.11	139.160.234.11	UNITE	□	MONO
3	5.121	139.160.234.21	MODBUS	☑	MULTI
4	4.12	139.160.234.12	UNITE	□	MULTI
5	4.13	139.160.234.13	UNITE	☑	MONO
6	5.122	139.160.234.22	MODBUS	□	MULTI
7	5.123	139.160.234.23	MODBUS	□	MULTI
8	4.15	139.160.234.14	UNITE	☑	MONO
9			UNITE	☑	MONO
10			UNITE	☑	MONO
11			UNITE	☑	MONO
12			UNITE	☑	MONO

Elements and Functions

This table describes the various zones that make up the configuration screen:

Zone	Number	Function
common	1	common part of the communication configuration screens.
specific	2	used to display the module address.
	3	used to configure the TCP/IP services. See *Configuration of TCP/IP Messaging, p. 343*.
	4	not available on the TSX WMY 100 module.
	5	not available on the TSX WMY 100 module.
	6	used to select the services used by the module.
	7	used to configure of the SNMP service. See *Configuring the SNMP Service, p. 350*.
	8	not available on the TSX WMY 100 module.
	9	not available on the TSX WMY 100 module.
	10	allows configuration of the module in Bridge.

Type of Communication According to Connection Configuration

At a Glance

Depending on the connection configuration of the **TSX WMY 100** HMI module, you can carry out messaging:

- in mono-connection
- in multi-connection.

According to the type of protocol and access control management, multi-connection mode requires a specific configuration of the correspondence table.

Note: In the following, only examples of multi-connection will be dealt with. Furthermore, the emphasis is placed on communication between the module and a remote PC device containing several applications.

Configuration Rules

Several connections can be configured with the same IP address. They must be defined with the same protocol, the same access rights, and the same connection mode.

In the case of a Modbus connection, only two connections can be configured with the same IP address. Addresses associated with these connections are:

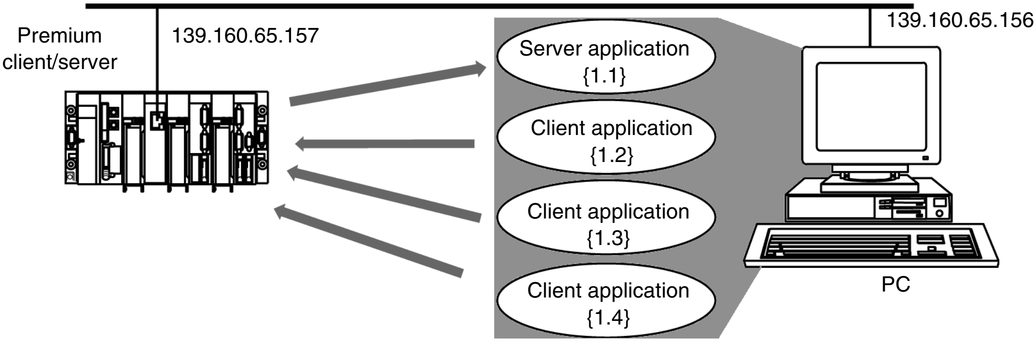
- an X-WAY address (for example: {2.103}) for the Modbus client connection,
- an X-WAY address equal to {x.x} for Modbus server connections.

Multi-connection in UNI-TE Protocol with Access Management

In this example, the PC contains four applications with a unique X-WAY address and one single IP address:

- The module opens a connection and communicates with the server application with the address {1.1}.
- Each of the other client applications with the address {1.2}, {1.3}, {1.4} is authorized to open a connection and communicate with the module.

Exchanges are carried out according to the UNI-TE protocol and the module's access control service is enabled.



In order to implement this example, you must:

- activate connection management by checking the module's **Access control** box,
- configure communication from the module to the PC's server application in the table
  - **X-Way address**: 1.1
  - **IP address**: 139.160.65.156
  - **Protocol**: UNI-TE
  - **Access**: authorized (cell checked)
  - **Mode**: Multi
- configure communication of each of the PC's client applications to the server module in the table:
  - **X-Way address**: 1.2 and 1.3 and 1.4
  - **IP address**: 139.160.65.156
  - **Protocol**: UNI-TE
  - **Access**: authorized (cell checked)
  - **Mode**: Multi

Multi-connection in UNI-TE protocol without access management

This example is the same as the previous one, except that access control is not configured.

In this case, if the application {1.1} is the only server application, the module's correspondence table only contains the following line:

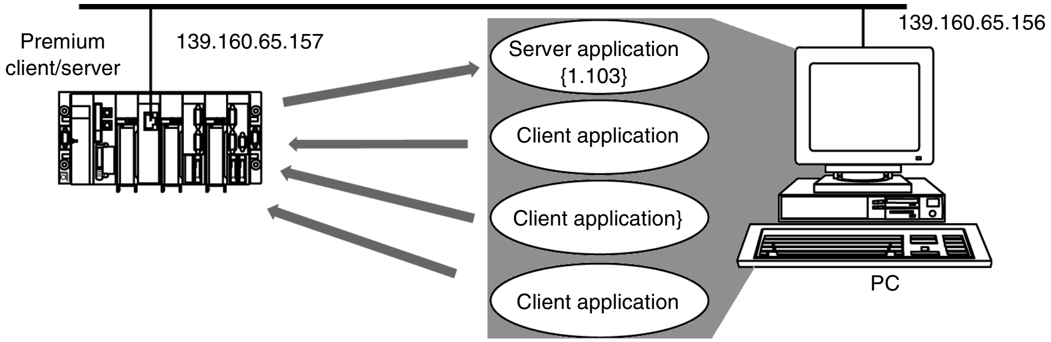
- **X-Way address**: 1.1
- **IP address**: 139.160.65.156
- **Protocol**: UNI-TE
- **Access**: the cell is grayed out
- **Mode**: Multi

Multi-connection in Modbus Protocol with Access Management

In this example, the PC contains four applications. The server application has an imaginary X-Way address and the other client applications do not need an X-Way address:

- The module opens a connection and communicates with the server application with the imaginary address {1.103}.
- Each of the client applications has no address but is authorized to open a connection and communicates with the module.

Exchanges are carried out according to the Modbus protocol and the module's access control service is enabled.



In order to implement this example, you must:

- activate connection management by checking the module's **Access control** box,
- configure communication from the module to the PC's server application in the table
  - **X-Way address**: 1.103 (address between 100 and 163)
  - **IP address**: 139.160.65.156
  - **Protocol**: Modbus
  - **Access**: the cell is grayed out
  - **Mode**: Multi
- configure communication of all the PC's client applications to the server module in the table:

- **X-Way address**: x.x (this address allows you to specify that other applications exist for the same IP address)
- **IP address**: 139.160.65.156
- **Protocol**: Modbus
- **Access**: authorized (allows you to specify that other applications are authorized to connect and communicate)
- **Mode**: Multi

Note: If you want to prevent connection by client applications you must configure
- **X-Way address**: x.x (this address allows you to specify that other applications exist for the same IP address)
- **IP address**: 139.160.65.156
- **Protocol**: Modbus
- **Access**: not authorized (cell unchecked)
- **Mode**: Multi

Multi-connection in Modbus protocol without access management

This example is the same as the previous one, except that access control is not configured.

In this case, if the application {1,103} is the only server application, the module's correspondence table only contains the following line:
- **X-Way address**: 1.103
- **IP address**: 139.160.65.156
- **Protocol**: Modbus
- **Access**: the cell is grayed out
- **Mode**: Multi

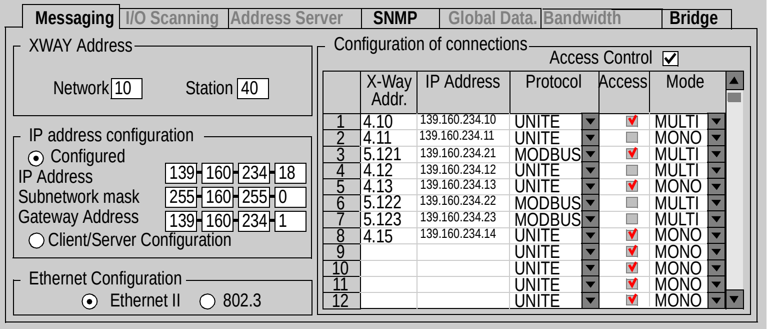
Configuration of TCP/IP Messaging

At a Glance

In order to use the **TSX WMY 100** HMI module to communicate on ETHERNET, it is necessary to adjust the messaging configuration parameters (See *Configuration Parameters Linked to TCP/IP Services, p. 345*).

How to Access the Messaging

The procedure for accessing the configuration parameters for TCP/IP services is as follows.

Step	Action
1	Access the module configuration screen.
2	Select the **Messaging** tab. **Result** 

XWAY Address

Network 10 Station 40

IP address configuration

● Configured

IP Address 139.160.234.18

Subnetwork mask 255.160.255.0

Gateway Address 139.160.234.1

○ Client/Server Configuration

Ethernet Configuration

● Ethernet II ○ 802.3

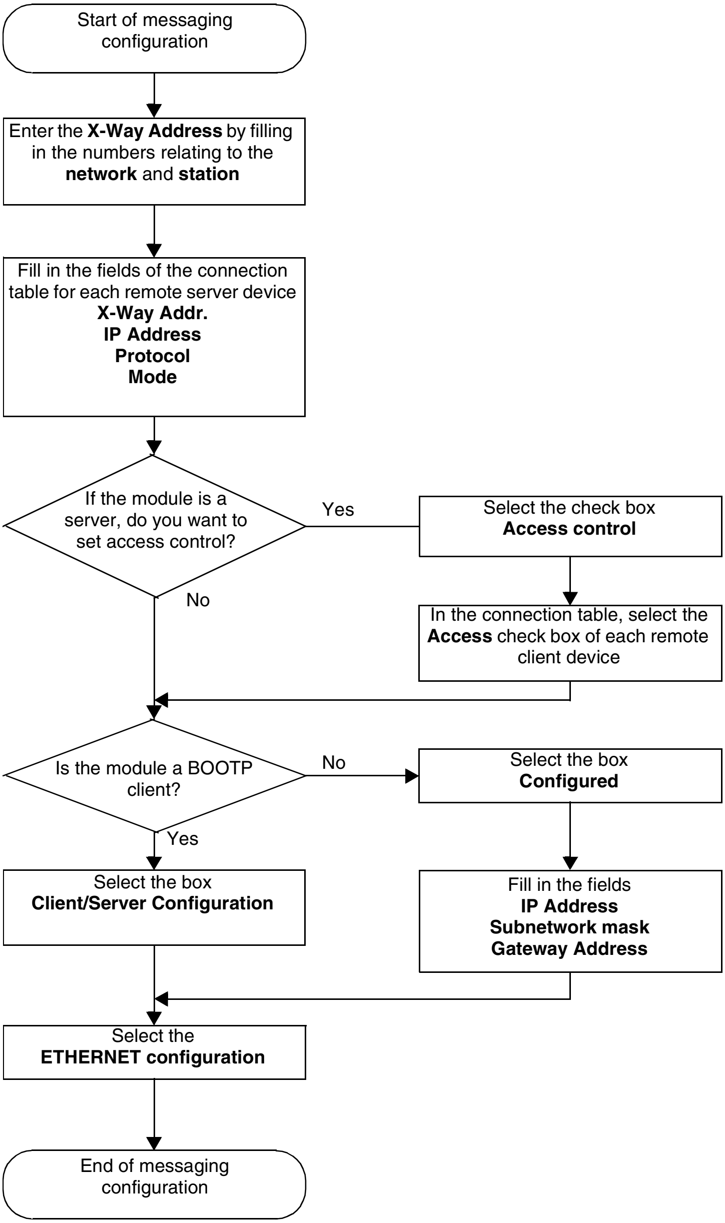
Configuration of connections

Access Control ☑

	X-Way Addr.	IP Address	Protocol	Access	Mode
1	4.10	139.160.234.10	UNITE	☑	MULTI
2	4.11	139.160.234.11	UNITE	□	MONO
3	5.121	139.160.234.21	MODBUS	☑	MULTI
4	4.12	139.160.234.12	UNITE	□	MULTI
5	4.13	139.160.234.13	UNITE	☑	MONO
6	5.122	139.160.234.22	MODBUS	□	MULTI
7	5.123	139.160.234.23	MODBUS	□	MULTI
8	4.15	139.160.234.14	UNITE	☑	MONO
9			UNITE	☑	MONO
10			UNITE	☑	MONO
11			UNITE	☑	MONO
12			UNITE	☑	MONO

How to Configure the Messaging

The following procedure gives the configuration principle.


Start of messaging configuration
Enter the **X-Way Address** by filling in the numbers relating to the **network** and **station**
Fill in the fields of the connection table for each remote server device **X-Way Addr.** **IP Address** **Protocol** **Mode**
If the module is a server, do you want to set access control?
Yes
Select the check box **Access control**
In the connection table, select the **Access** check box of each remote client device
No
Is the module a BOOTP client?
No
Select the box **Configured**
Fill in the fields **IP Address** **Subnetwork mask** **Gateway Address**
Yes
Select the box **Client/Server Configuration**
Select the **ETHERNET configuration**
End of messaging configuration

Configuration Parameters Linked to TCP/IP Services

At a Glance

The **TSX WMY 100** HMI module has configuration parameters linked to TCP/IP services.

Availability of Parameters

The following table shows, for the**TSX WMY 100** HMI module, the parameters that can be accessed from the configuration screen and the values that they can be assigned.

Parameters			**TSX WMY 100**
X-WAY Address			By PL7
IP Address	Default IP address		X
	Configured	IP Address	X
		Subnetwork mask	X
		Gateway address	X
	Client/Server Configuration		X
Connections	Connections which can be opened		-
	Access control		X
	Function table	X-WAY Address	X
		IP Address	X
		Protocol	UNI-TE or Modbus
		Access	X
		Mode	Mono- or Multi-connection
ETHERNET	ETHERNET II		X
	802.3		X
Legends:			
X	Accessible		
-	Not accessible		

Configuration Parameters for IP Addresses

At a Glance

This zone allows the IP address of a module to be defined in three distinct ways:
- by manual configuration,
- or by using a configuration supplied by a BOOTP/DHCP server device.

Configured

Selecting the **Configured** field allows manual configuration according to your own requirements:
- module IP address,
- the subnetwork mask, which defines the part allocated to the subnetwork identifier in the IP address,
- the IP address of the default gateway, to which messages for other networks are transmitted.

Note: If the module is connected to an existing TCP/IP network, the IP addresses are administered globally, therefore the IP parameters must be configured. Otherwise there is a risk of disruption on the existing network caused by possible double allocation of the IP addresses.

Client/Server Configuration

Selecting the **Client/Server configuration** field allows the module's IP address to be configured from a remote device acting as a BOOTP server. See .

In this instance, the **IP address**, **Subnetwork mask** and **Gateway Address** fields are not significant and are grayed out.

Connection Configuration Parameters

At a Glance

This field enables you to :

- configure the number of connections that can be opened by the module,
- activate an access check service,
- list the remote devices which can be connected to the module according to the communication protocol.

Connections which can be opened

The **Connections that can be opened** field gives the maximum number of devices that can be connected in parallel to the module:

- The default value is 8 connections.
- The value is between 1 and 32 connections.

Note: You are recommended to set this parameter to the useful value in order to optimize the communication resources for each connection point.

Access control

The **Access check box** is used to activate or deactivate the check on remote devices that want to open a TCP connection to the module.

- If the box is checked, access check management is activated and the **Access column** in the table is de-grayed (active).
  If the module is operating in server mode, only the remote devices selected by the **Access box** in the table are authorized to be connected as a client and then to communicate.
- If the box is unchecked, check management does not operate and the **Access column** in the table is grayed out (not active).
  If the module is operating in server mode, third party remote devices can be connected as a client and then communicate with the module without having to be declared in the table.

Note: The access check only works on the TCP/IP profile and is used for the module to operate in server mode.

Function table

This table enables you:

- to list the remote devices with which the local module wishes to open a TCP connection and then transmit (module operating in client mode),
- to provide a cross reference between the **X-WAY address** {network, station} and the **IP address**,
  - the network number must be less than or equal to 127,
  - the station number is between 0 and 63 for a connection in UNITE or between 100 and 163 for a connection in Modbus.
- to specify the communication protocol for each remote device during the connection using a scrolling menu:
  - **UNI-TE** (default value),
  - **Modbus**,
- when managing access checks, to nominate the remote devices authorized to open a TCP connection and then to send to the local module (module operating in server mode),
- for each remote IP address, to choose the **Connection mode** using a scrolling menu :
  - **Monoconnection** : the module only allows one connection with the same remote IP address,
  - **Multiconnection** : the module allows only one connection with the same remote IP address in client mode and several connections with the same remote IP address in server mode.

Ethernet Configuration Parameters

At a Glance

This zone enables you to define the format of the Ethernet frame via TCP/IP:

- Ethernet II format: as per RFC 894 (Most commonly used format),
- 802.3 format: as per RFC 1042. This format should be selected when remote devices use this format.

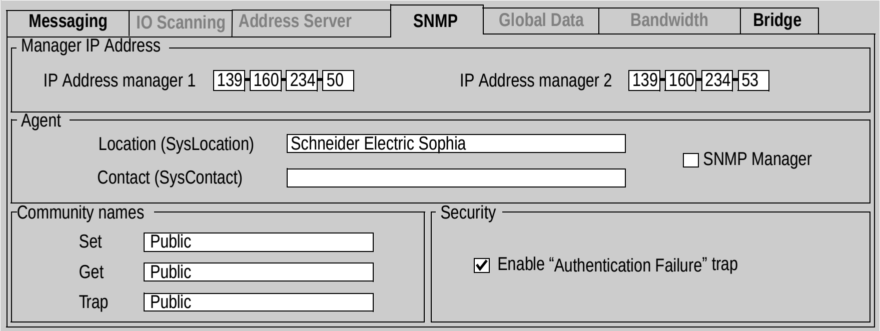
Configuring the SNMP Service

At a Glance

In order to use the **TSX WMY 100** HMI module as an SNMP agent, it is necessary to adjust the configuration parameters of the SNMP service (See *Description of SNMP Service, p. 353*).

How to Access the SNMP Service

The procedure for accessing the configuration parameters of the SNMP service is as follows.

Step	Action
1	Access the module configuration screen.
2	Select the **SNMP** tab. **Result**  Messaging IO Scanning Address Server SNMP Global Data Bandwidth Bridge Manager IP Address IP Address manager 1 139.160.234.50 IP Address manager 2 139.160.234.53 Agent Location (SysLocation) Schneider Electric Sophia Contact (SysContact) □ SNMP Manager Community names Set Public Get Public Trap Public Security ☑ Enable "Authentication Failure" trap

How to Configure SNMP

The following procedure gives the configuration principle for the SNMP service.

Step	Action
1	Enter the SNMP manager addresses: ● **Manager 1 IP addresses** ● **Manager 2 IP addresses**
2	Fill in the fields: ● **Location (SysLocation)** ● **Contact (SysLocation)**. Or alternatively check the **SNMP Manager** box to indicate that the information will be completed by the SNMP Manager.
3	If you want to define access rights, fill in the community names: ● **Set** ● **Get** ● **Trap**
4	If you want to activate transmission of an event to the module, check the **Activate "Authentication Failure" trap** box.

Configuration Parameters Linked to the SNMP Service

At a Glance

The **TSX WMY 100** HMI module has configuration parameters linked to the SNMP service.

Availability of Parameters

The following table shows, for each ETHERNET module, the parameters that can be accessed from the configuration screen.

Parameters		**TSX WMY 100**
Manager IP Address		X
Agent	Location (SysLocation)	X
	Contact (SysContact)	X
	SNMP Manager	X
Community names	Set	X
	Get	X
	Trap	X
Security	Enable "Authentication Failure" trap	X
Legends:		
X	Accessible	
-	Not accessible	

Description of SNMP Service

At a Glance

Parameters linked to the SNMP module are divided into 4 categories:

- the IP addresses of the SNMP module administrative devices,
- the agents,
- the community names,
- security.

Note: Only 7-bit ASCII characters can be used in the character string entry fields.

Administrators IP Addresses

This zone allows the completion of the SNMP administrators IP addresses. The modules authorize a maximum of two administrators.

These addresses are used during possible transmission of events (TRAP).

Agent

This zone allows the localization and identification of an agent from the SNMP administrator.

It comprises two fields:

- The **Location (SysLocation)** field: indicates the physical location of the device (32 characters maximum).
- The **Contact (SysLocation)** field: indicates the personnel to contact for device management and the method of contact (32 characters maximum).
- It is possible to select information assignment by a "SNMP Manager" tool for network management, rather than by PL7 software. To do this, check the **SNMP Manager** box.

Community Name

This zone allows the definition of a community name by service family Set, Get and Trap.

It comprises three fields:

- The **Set** field defines the community name for the Set service (16 characters maximum). The default value of the field is **Public**.
- The **Get** field defines the community name for the Get service (16 characters maximum). The default value of the field is **Public**.
- The **Trap** field defines the community name for the Trap service (16 characters maximum). The default value of the field is **Public**.

The purpose of these fields is to define the access rights for SNMP agent MIB objects (local module) in relation to requests transmitted by the administrator.

Example

If the administrator transmits a SetRequest request with the community name **Test** and the module has the community name **Public**, the request will not be executed.

Security

This zone contains the **Enable "Authentication Failure" trap** check box.

Enabling this box allows the transmission of an event (TRAP) of an authentication failure to be confirmed from the SNMP agent to the administrator that originally posted the request.

In this way, the agent warns the administrator that the request has been refused following an identification error (community name configured in the administrator is different to the one configured in the agent).

CAUTION

Modifications to the SNMP parameters

The SNMP administrator is able to modify the value of certain parameters, which can be configured by PL7 (Enabling "Authentication failure", Location, Contact. etc.)

When there is a cold start, warm restart or application download, the values initially configured in PL7 are then restored.

Failure to follow this precaution can result in injury or equipment damage.

Configuration of the Bridge Function

At a Glance

The**TSX WMY 100**HMI module may be used as an X-Way bridge station (See *Configuration of Properties, p. 29*). This guarantees transparent communication between various networks.

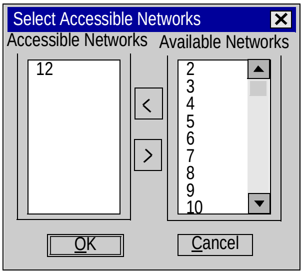
How to Access the Bridge Function

The procedure for accessing the configuration parameters for the bridge function is as follows.

Step	Action
1	Access the module configuration screen.
2	Select the **Bridge** tab.

Configuring the Bridge Function

The following procedure allows you to configure the **TSX WMY 100** HMI module of the station as a bridge.

Step	Action
1	Double click on the zone in reverse video. **Result** The **Select Accessible Networks** window appears.  Select Accessible Networks Accessible Networks Available Networks 12 2 3 4 5 6 7 8 9 10 OK Cancel If the list of accessible networks for the module is empty, the window appears automatically (without double clicking).
2	Double click on the network number in the **Available Networks** scroll down list. **Result** The network number is assigned in the **Accessible Networks** scroll down list.
3	Repeat operation 3 as many times as needed to define all networks that can be accessed by the module. Then go to step 5.
4	Confirm the selection by clicking on the **OK** button.
5	Confirm the bridge configuration by clicking on the **Save** button.

How to Delete the Bridge Function

The module bridge function can be deleted.

Step	Action
1	Access the **Bridge Configuration** screen.
2	Click on the **Delete** button.
3	Confirm the deletion by clicking on the **Save** button.

How to Delete Access to a Network

Access to a single network can be deleted.

Step	Action
1	Access the **Bridge Configuration** screen.
2	Double click on the zone in reverse video.
3	Double click on the network number in the **Accessible Networks** scroll down list. **Result** The network number is reassigned in the **Available Networks** scroll down list.
4	Confirm the selection by clicking on the **OK** button.
5	Confirm the bridge configuration by clicking on the **Save** button.

11.3 Debugging

At a Glance

Subject of this Section

This section describes installation of the **TSX WMY 100** HMI module running PL7 during debugging.

What's in this Section?

This section contains the following topics:

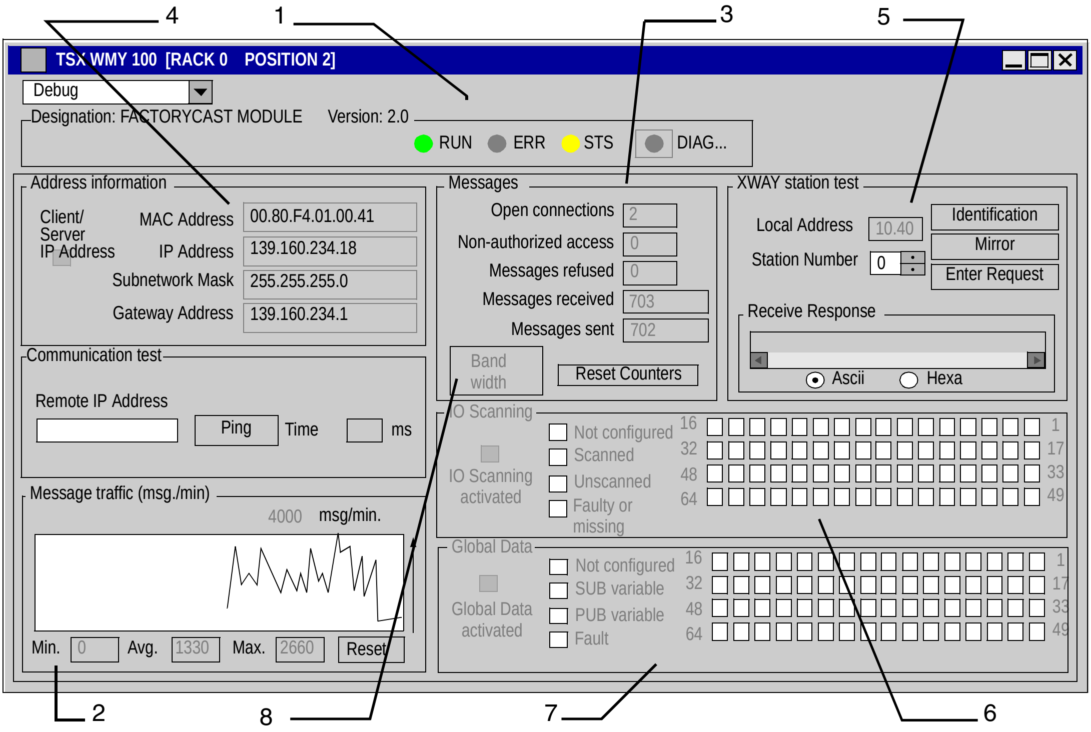
Module Debug Screen

At a Glance

This screen, separated into six zones, is used for debugging through an ETHERNET link.

Illustration

The screen dedicated to ETHERNET communication is displayed as follows:



Designation: FACTORYCAST MODULE   Version: 2.0

● RUN ● ERR ● STS

MAC Address	00.80.F4.01.00.41
IP Address	139.160.234.18
Subnetwork Mask	255.255.255.0
Gateway Address	139.160.234.1

Open connections	2
Non-authorized access	0
Messages refused	0
Messages received	703
Messages sent	702

Local Address 10.40

Station Number 0

(● Ascii ○ Hexa)

4000 msg/min.

Min. 0 Avg. 1330 Max. 2660

Elements and Functions

This table describes the different zones that make up the debug screen:

Number	Zone	Function
1	common	common part of the communication debug screens.
2	Message traffic	allows the graphical display of the number of messages processed by the module:
3	Messages	allows the display of the number of unacknowledged or refused messages and connections.
4	TCP/IP services	allows: ● display of TCP/IP services configuration, ● communication testing of the TCP/IP profile.
5	X-WAY test station	allows UNI-TE communication testing on the TCP/IP profile.
6	IO Scanning	not accessible on this module.
7	Global Data.	not accessible on this module.
8	Bandwidth	not accessible on this module.

General debugging parameters

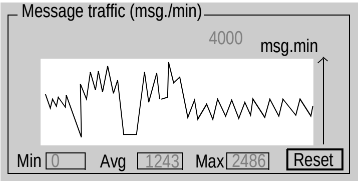
At a Glance

The general debugging parameters are grouped together in two windows:

- the **Message traffic** window
- the **Messages** window.

Message traffic

The window is displayed as below:



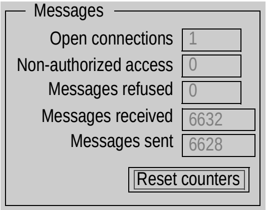
Min 0 Avg 1243 Max 2486

It shows the number of messages processed by the module per minute (transmission and reception) in graphic form. To provide a better display, the scale is automatically adapted according to change in flow.

The **Reset** button sets the three counters **Min.**, **Avg.** and **Max.** to zero.

Messages

The window is displayed as below:



Open connections 1
Non-authorized access 0
Messages refused 0
Messages received 6632
Messages sent 6628

This window is related to TCP/IP messaging:

- number of TCP connections open
- number of non-authorized accesses
- number of messages refused, received and sent

You can reset these counters to zero by clicking on the **Reset counter** button.

Debugging parameters for TCP/IP services

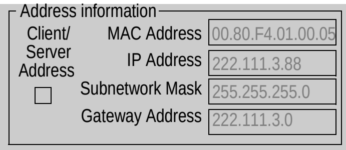
At a Glance

The debugging parameters for the TCP/IP services are grouped together in two windows:

- the **Address information** window,
- the **Communication test** window.

Address information

The window is displayed as below:



Address information

Client/ Server Address □

MAC Address 00.80.F4.01.00.05

IP Address 222.111.3.88

Subnetwork Mask 255.255.255.0

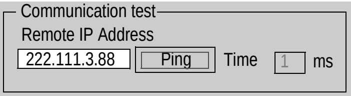
Gateway Address 222.111.3.0

It specifies the configuration data of the IP address:

- **MAC address**: unique fixed address for a module
- **IP Address**
- **Subnetwork mask**
- **Gateway Address**: address of the gateway

Communication test

The window is displayed as below:



Communication test

Remote IP Address 222.111.3.88

Time 1 ms

This window is used to test IP communication towards another station.

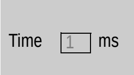
How to Test TCP/IP communication with the Ping request

At a Glance

This page indicates the procedure for testing TCP/IP communication from a **Ping** request.

Procedure

The following procedure is used to send the Ping request and therefore to test the correct routing of information between two devices.

Step	Action
1	Enter the address of the station to be queried using the **Remote IP address** field.
2	Press the **Ping** button. **Result** The response appears in the **Time** field:  Time 1 ms The returned time corresponds to the time it takes the message to be sent and return in ms.

Response type

The following table groups together the various types of possible responses to the Ping request.

If the response is	then
positive	The windows records the time it takes the message to be sent and return in ms.
negative	a window with the message **Exchange incorrect** specifies that the remote device was not reached in the network architecture.

Available requests for testing a communication channel

At a Glance

This page describes the various options for testing a communication channel from the debug screen.

Test conditions

When a request is sent to a station that is not connected, an error message appears. The test applies to stations belonging to the same local area network.

Available requests

The **X-Way station test** window allows the following requests:

- **Identification**: causes the Identification request to be sent to the designated remote station,
- **Mirror**: is used to send a Mirror request to the designated station. If this function is chosen, a screen appears allowing you to select the length of the string of characters to be sent (maximum of 80 characters). The PLC then sends this string of characters (ABCD, etc.) to the destination device. This device automatically returns the string of characters received to the sender,
- **Enter Request** : is used to send a UNI-TE request, other than those sent by the command buttons, to the designated station. If this function is chosen, a screen appears allowing you to select the parameters specific to the request (the request code must be coded in hexadecimal).

Note: The first two requests are sent to the UNI-TE server of the TSX WMY module. The third is sent to the server of the processor.

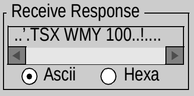
How to Test a Channel with the Identification and Mirror requests

At a Glance

This page indicates the procedure for testing a communication channel using the Identification and Mirror requests.

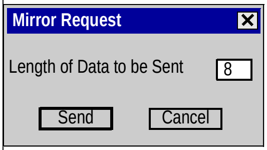
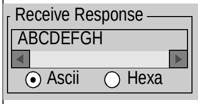
How to Identify a Station

The following procedure is used to identify a designated station.

Step	Actions
1	Select the address of the station to be queried using the **Station number** field.
2	Press the **Identification** button. **Result** The response appears in the **Receive Response** window:  Receive Response ..’.TSX WMY 100..!.... ● Ascii ○ Hexa

How to Send the Mirror Request

The following procedure is used to send the Mirror request and therefore to test the correct routing of information between two devices.

Step	Action
1	Select the address of the station to be queried using the **Station** field.
2	Press the **Mirror** button. **Result** The following window appears:  Mirror Request Length of Data to be Sent: 8
3	Enter the length of the data to be sent (maximum 80 characters).
4	Press the **Send** button. **Result** The response appears in the **Receive Response** window:  Receive Response ABCDEFGH ⦿ Ascii ○ Hexa The response contains: ● the character string ABCDEFGH corresponding to a length of data sent of 8.

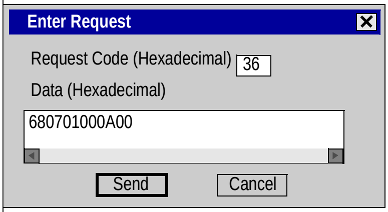
How to test a channel with requests

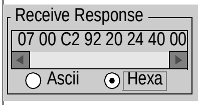
At a Glance

This page indicates the procedure for testing a communication channel from the debug screen through different requests.

How to Send a Request

The following procedure is used to send a request, other than those sent by the command buttons, to a designated station. In this example, the sent request enables 10 words to be read (from %MW1 to %MW10).

Step	Action
1	Select the address of the station to be queried using the **Station** field.
2	Press the **Enter Request** button. **Result** The following window appears:  Enter Request Request Code (Hexadecimal) 36 Data (Hexadecimal) 680701000A00 The data transmitted in this example is coded on 6 bytes.
3	Enter the function code (coded in hexadecimal on a byte), corresponding to the request that you want to send. In this example, the read request code is 16#36.
4	Enter the data to be transmitted by coding each data item in hexadecimal. The data is then entered one after the other with no spaces between data items. When the data is coded on a word, the most significant bytes and the least significant bytes are inverted. In this example, the data is as follows: ● 16#68: on a byte, define the segment (internal data), ● 16#07: on a byte, define the type of object (words), ● 16#0100: on a word, define the first word to be read, ● 16#0A00: on a word, define the number of words to be read.

Step	Action
5	Press the **Send** button. **Result** The response appears in the **Receive Response** window:  Receive Response 07 00 C2 92 20 24 40 00 ○ Ascii ● Hexa The response for the example contains the data on 21 bytes. ● 16#07: corresponds to the type of object (words), ● 16#00C2: corresponds to the value of the first word (the most significant bytes and the least significant bytes are inverted; its value is 16#C200), ● ...

11.4 Language objects associated with the TSX WMY 100 module

At a Glance

Subject of this Section

This section presents the different language objects specific to the **TSX WMY 100** HMI module.

What's in this Section?

This section contains the following topics:

Implicit Exchange Language Objects

At a Glance

This page describes all the implicit exchange language objects for ETHERNET communication with the TSX WMY 100 module that can be displayed or modified by the application program.

Bit Objects

The table below shows the various implicit exchange bit objects.

Object (1)	Function	Meaning
%Ixy.MOD.ERR	Module error bit	This bit set to 1 indicates a module error (at least one of the channels has an error, for instance)
%Ixy.i.ERR	Channel error bit	This bit set to 1 indicates a line fault.
Legend		
(1)	Address xy.i ● x : corresponds to the rack number ● y : corresponds to the module number ● i : corresponds to the channel number	

Word Objects

The table below shows the various implicit exchange word objects.

Object (1)	Function	Meaning
%IWxy.i.0 to %IWxy.i.9	Reserved	-
Legend		
(1)	Address xy.i ● x : corresponds to the rack number ● y : corresponds to the module number ● i : corresponds to the channel number	

Explicit Exchange Language Objects

At a Glance

This page describes all the explicit exchange language objects for ETHERNET communication with the TSX WMY 100 module that can be displayed or modified by the application program.

Internal Words

The following table describes the internal words:

Object (1)	Function	Meaning
%MWxy.MOD.2	Module status	● x0 = 1: defective module ● x1 = 1: functional error (error between the processor and the module, adjustment or configuration error, etc.) ● x2 = 1: terminal block fault (not connected) ● x3 = 1: self-tests running ● x4 = 1: reserved ● x5 = 1: error in hardware or software configuration (the module present is not that declared in the configuration, the sub-modules are not compatible) ● x6 = 1: module missing ● x7 = 1: error in one of the sub-modules
%MWxy.i.2	Channel standard status	● x0 to x3 = 0: reserved ● x4 = 1: module error or self-tests running ● x5 and x6 = 0: reserved ● x7 = 1: application error (error in the configuration)
%MWxy.i.3	Specific channel status	Byte 0: ● x0 = 1: if X-WAY bridge (0 if not bridge) Byte 1: number of TCP connections open
%MWxy.i.4	Error counters	Number of connection requests refused
%MWxy.i.5	Error counters	Number of messages refused
%MWxy.i.6	Network/station address	Byte 0: station number Byte 1: network number
%MWxy.i.7 and %Mwxy.i.8	Message counters on the link layer	Number of messages sent on the link layer
%MWxy.i.9 and %Mwxy.i.10	Message counters on the link layer	Number of messages received on the link layer
%MWxy.i.11	reserved	-
%MWxy.i.12	reserved	-
%MWxy.i.13	reserved	-

Object (1)	Function	Meaning
%MWxy.i.14	reserved	-
%Mwxy.i.15	Module loading rate	Byte 0: ● Messaging loading rate Byte 1: ● Other services loading rate
Legend		
(1)	Address xy.i ● x : corresponds to the rack number ● y : corresponds to the module number ● i : corresponds to the channel number	

Explicit exchange management and report

At a Glance

This page describes all the language objects that manage explicit exchanges.

Word objects

The table below shows the different word objects for the management of explicit exchanges.

Object (1)	Function	Meaning
%MWxy.MOD.0	Module exchanges in progress	● x0 = 1: status reading in progress ● x1 = 1: sending of command parameters to the communication module ● x2 = 1: sending of adjustment parameters to the communication module
%MWxy.MOD.1	Module report	● x1 = 0: command parameters received and accepted by the module ● x2 = 0: adjustment parameters received and accepted by the module
%MWxy.i.0	Channel exchanges in progress	● x0 = 1: status reading in progress ● x1 = 1: sending of command parameters to the communication channel ● x2 = 1: sending of adjustment parameters to the communication channel
%MWxy.i.1	Channel report	● x1 = 0: command parameters received and accepted by the communication channel ● x2 = 0: adjustment parameters received and accepted by the communication channel
Legend		
(1)	Address xy.i ● x : corresponds to the rack number ● y : corresponds to the module number ● i : corresponds to the channel number	

Language Objects Associated with Configuration

At a Glance

This page describes all the configuration language objects for ETHERNET communication with the TSX WMY 100 module that can be displayed by the application program.

Internal Constants

The following table describes the internal constants:

Object	Function	Meaning
%KWxy.i.0	Type	Byte 0 = 11 for ETHERNET communication Byte 1: reserved
%KWxy.i.1	Physical layer	Byte 0: fixed at 16#01 (corresponds to GPX2) Byte 1: reserved
%KWxy.i.2	Reserved	-
%KWxy.i.3	Reserved	-
%KWxy.i.4	Reserved	-
%KWxy.i.5	X-Way network address	Byte 0: network number (0 by default) Byte 1: reserved
%KWxy.i.6	Type of ETHERNET driver for TCP/IP	Byte 0: fixed at 16#01 (corresponds to RJ45) ● = 16#01: RJ45 Byte 1: ● = 16#00: ETHERNET II (default value) ● = 16#01: 802.3
%KWxy.i.7	TCP/IP configuration: address type	Byte 0: reserved Byte 1: inherited address ● = 16#01: from the application-specific function ● = 16#03: from a server
%KWxy.i.8 and %KWxy.i.9	Local IP address	Example with the address 139.160.65.109 Byte 0 = 109 Byte 1 = 65 Byte 2 = 160 Byte 3 = 139
%KWxy.i.10 and %KWxy.i.11	IP address of gateway	Example with the address 139.160.65.1 Byte 0 = 1 Byte 1 = 65 Byte 2 = 160 Byte 3 = 139

Object	Function	Meaning
%KWxy.i.12 and %KWxy.i.13	Subnetwork mask	Example with the address 255.255.255.0 Byte 0 = 0 (least significant) Byte 1 = 255 Byte 2 = 255 Byte 3 = 255 (most significant)
%KWxy.i.14	Services used	Byte 0: ● x0 = 1: IO Scanning service is used ● x1 = 1: BOOTP server service is used ● x2 = 1: global data service is used ● x3 = 1: the bandwidth adjustment service is used Byte 1: reserved
%KWxy.i.15	TCP services used	Byte 0: ● x0 = 1: Modbus communication on TCP/IP is used ● x1 = 1: access control is activated Byte 1: reserved

11.5 Questions/Answers

Questions/Answers

General

Below you will find a list of the most frequently asked questions and answers regarding communication by Ethernet network.

What should I do if the TSX WMY 100 module displays the LED status "Not configured" (Flashing ERR)?

- Check that it is configured in PL7.
- Check that there is no other module with the same X-WAY network number in the PLC configuration.
- If the module is configured in "Client/Server address", make sure that a BOOTP server is operational and accessible.

What should I do if the module displays a correct LED status (STS=RUN=ON) but does not respond to requests?

Try a Ping command to this IP address, from a device on the same network:

- No response:
  - Check the network configuration: wiring, "hubs" status, IP addresses, masks and gateways if there are routers in the configuration.
- Correct response to the Ping:
  - Check the module's X_WAY address.
  - Check whether access control is activated or not.
  - Try to connect to the Rack Viewer page by http. If this connection works, check the X-WAY address of the client application.

Can I change the MAC address of the module?

No. This is fixed for every Ethernet device to guarantee its uniqueness.

How can I find out the speed with which the module communicates?

Via an internet browser, by connecting to the Ethernet Module Statistics page.

What should I do if the messaging EFs do not get a response?

- Check that the X-WAY destination address is part of the IP/X-WAY connections table.
- Check that the configuration of the destination module does not prohibit this connection (access control).
- In debug mode, try the "Ping" command towards the IP address of the destination PLC, then try an "Identification" request towards the destination address, if it is part of the same X-Way network.

What should I do if I receive the 0xFF12 refusal code on the messaging EFs?
This code indicates that the module is full. Reduce the transition rate.

What should I do if the READ_ASYNC, WRITE_ASYN messaging EFs do not get a response (time out)?
Check that the "mast" tasks of the client and server PLCs are configured to "periodic", with a sufficient period to ensure that the PLCs are not overrun.

When should I configure my connections to "Mono-Connection"?
When this connection has to be established to a TSX ETY 110, MODBUS protocol.

The destination PLC of my messages is a bridge. Which Xway destination address should I configure in the connections table?
The X-WAY address of the first communication module of the PLC bridge (it is the address of the processor's PCMCIA card, if there is one present).

In the module configuration screens, the last entry inserted into one of the grids has not been taken into account at the validation. Why?
You need to remove the focus (the scale indicator) from the last cell of the grid in which new data has been entered in order to take this modification into account.

Why is the Bridge tab grayed out when I use a TSX / PCX processor of type 571xx and 572xx?
These processors do not manage bridge data.

Why is the Bridge tab grayed out when I have already changed the position of the processor?
After this type of modification, you have to validate the new hardware configuration to access and modify the bridge data managed globally at the level of the processor.

Index