Honeywell

Experion PKS
Enron Modbus Interface Reference

EPDOC-X233-en-431A
February 2015

Release 431

Honeywell

Document	Release	Issue	Date
EPDOC-X233-en-431A	431	0	February 2015

Disclaimer

Contents

Planning considerations for installing and configuring Enron Modbus controllers

This reference describes how to set up, configure, and test Enron Modbus controllers.

Revision history

Revision	Date	Description
A	February 2015	Initial release of document.

How to use this guide

Complete each step before commencing the next step.

Step	Go to
Set up the controllers and network	Other documentation for Enron Modbus
Use Quick Builder to build Enron Modbus channels	• Enron Modbus channel and controller reference • "Build channels" topic in the *Quick Builder User's Guide*
Use Quick Builder to build Enron Modbus controllers	• Enron Modbus channel and controller reference • "Build controllers" topic in the *Quick Builder User's Guide*
Download channel and controller definitions to the server	"Downloading items" topic in the *Quick Builder User's Guide*
Use Quick Builder to define points	"Building and configuring points" topic in the *Quick Builder User's Guide*
Optional. If you want to configure EFM meters, use Quick Builder to build Electronic Flow Measurement (EFM) meters	• Main properties for an Enron Modbus meter template • "Build meters" topic in the *Quick Builder User's Guide*
Test communications	Testing Enron Modbus to field device communications

Related topics

"About the Enron Modbus protocol" on page 6
"Devices supported by the Enron Modbus interface" on page 7
"Other documentation for Enron Modbus" on page 8
"Architectures for Enron Modbus" on page 9
"Enron Modbus channel and controller reference" on page 17
"Main properties for an Enron Modbus meter template" on page 36
"Testing Enron Modbus to field device communications" on page 44

About the Enron Modbus protocol

The Enron Modbus protocol is based on the standard Modbus protocol with variations specified by Enron Corporation.

The most significant variations to the Modbus protocol are:

- A single address register range from 1–32,767 (although a more restricted range is most common), which is divided up into regions for specific register sizes.
- Support for single bit, 16-bit, and 32-bit register sizes, each within its own designated address range.
- Ability to transmit event logs and historical data, for EFM API21.1 support.
- Modbus address and device register address are not offset.

The Enron Modbus protocol uses the same framing and frame format as standard Modbus. This is described in the Modbus specifications at http://www.modbus.org/specs.php and is not repeated in this reference.

Devices supported by the Enron Modbus interface

A controller is any device, including RTU or Flow Computer, that is communicating using the Enron Modbus protocol.

The server supports the following devices:

- *NuFlo Cameron Scanner 2000 MicroEFM Flow Computer* controller
- Other controllers that use the Enron Modbus protocol, as specified in the document titled *Specifications and Requirements for an Electronic Flow Measurement Remote Terminal Unit for Enron Corp*

The server communicates with Enron Modbus controllers by way of the Enron Modbus RTU, Enron Modbus ASCII, or Enron Modbus TCP protocol.

Other documentation for Enron Modbus

From Enron Corporation

The following Enron Modbus document describes the requirements and specifications of an Electronic Flow Measurement (EFM) Remote Terminal Unit (RTU) suitable for use by Enron Corporation. Reading this document is useful when installing the interface.

- *Specifications and Requirements for an Electronic Flow Measurement Remote Terminal Unit for Enron Corp*

From third-party flow computer vendors

- *Scanner® 2000 microEFM Hardware User Manual*, Cameron International Corporation

From Honeywell

- *Quick Builder User's Guide*

Architectures for Enron Modbus

The server supports both single and redundant communications to Enron Modbus controllers via serial connection, or a Modbus TCP connection.

Serial connection by Terminal Server

The server supports Terminal Server connections to Enron Modbus controllers, with supported device Systech NDS5000 or NDS 6000 series.

- Single or dual link
- Modbus RTU or ASCII

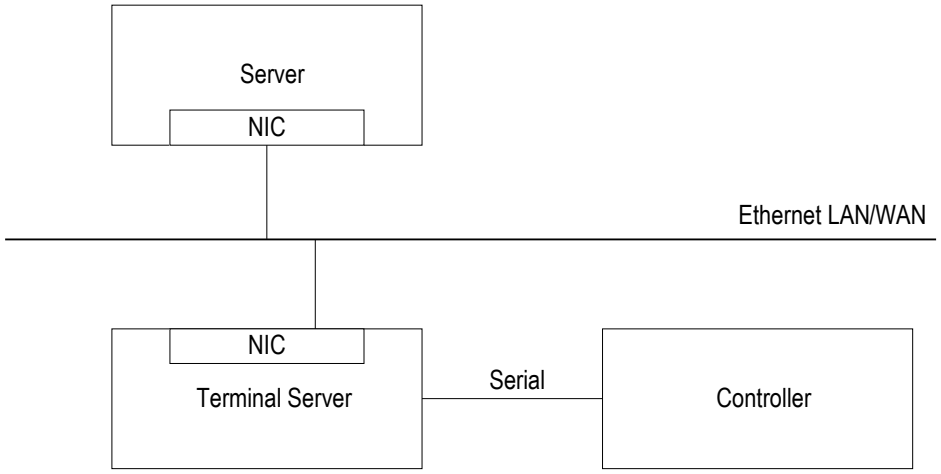

Figure 1: Serial connection by Terminal Server network architecture

Serial connection by Modbus TCP Bridge

The server supports serial connection via Modbus TCP Bridge connections to Enron Modbus controllers, for Lantronix Xpress DR-IAP device server, and Systech NDS5000 or NDS6000 series terminal servers configured to act as a Modbus Bridge.

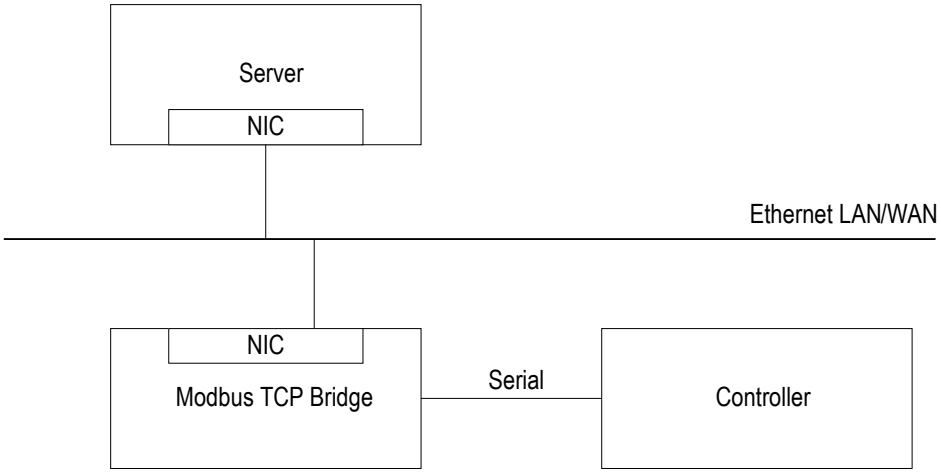

Figure 2: Serial connection by Modbus TCP Bridge network architecture

Enron Modbus TCP connection

To connect Enron Modbus controllers to the server communicating using the Enron Modbus TCP protocol, you are required to have a *network interface card* (NIC) connected to an Ethernet network on both the Server as well as the controller.

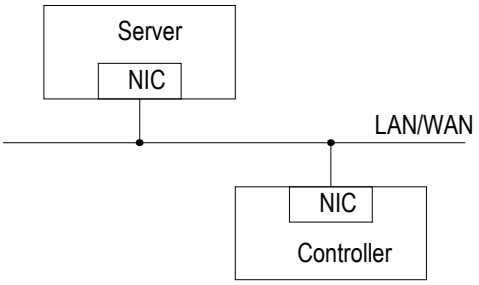

Figure 3: Non-redundant Enron Modbus TCP network architecture

Redundant communication architecture

If you require redundant communications, you must have two separate *network interface cards* (NICs) on both the Server and the controller, which are connected to separate Ethernet subnets.

It is expected that, during normal operation when both networks are available, the controller can respond to requests on either network connection at any time.

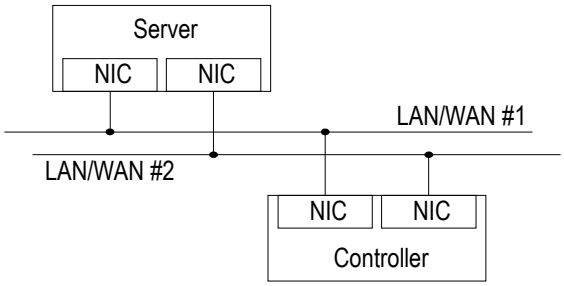

Figure 4: Redundant Enron Modbus TCP network architecture

Attaching a redundant Enron Modbus device to an FTE network

Enron Modbus Serial over FTE

Enron Modbus controllers supporting serial connections can redundantly connect to an FTE network by using either a single terminal server or two separate terminal servers. It does not matter to which switch you connect the terminal servers, although to maximize redundancy, if you're using two terminal servers, you should connect one to the yellow switch and one to the green switch.

Connect the terminal server(s) to the primary and backup serial links on your Enron Modbus device.

Enron Modbus TCP over FTE

If your Enron Modbus TCP devices support redundant links in the same subnet, you should be able to connect the links directly to each switch (for example, primary link to the yellow switch and backup link to the green switch).

If your Enron Modbus TCP device does not support redundant links in the same subnet, you will need to set up a second subnet from your yellow NICs on your servers. To do so, in the Advanced property page for the TCP/IP protocol of the yellow NIC, add a second IP address using a private subnet. You can then connect the primary link on the Enron Modbus device to one of the switches, and the backup link to the other.

Function codes supported by Enron Modbus

The function codes used to read from and write to Enron Modbus register addresses are:

Function Code (Hexadecimal)	Action	Register Type
01	Read	Boolean variables Default = 1xxx
03	Read	Numeric variables Default = 3xxx, 5xxx, 7xxx
05	Write Single	Boolean variables Default = 1xxx
06	Write Single	Numeric variables Default = 3xxx, 5xxx, 7xxx
0F	Write Multiple	Boolean variables Default = 1xxx
10	Write Multiple	Numeric variables Default = 3xxx, 5xxx, 7xxx

Event collection

For Event collection, the following function codes are used:

Function Code (Hexadecimal)	Action
03	Event data request Note that number of registers is ignored.
05	Event Acknowledgement

Interval log and daily log collection

For Interval log collection and Daily log collection, the following function codes are used:

Function Code (Hexadecimal)	Action
03	Log request Record number to read follows address.

Enron Modbus register map

The registers listed in the following tables refer to the addresses configured in Quick Builder. Note that if the Address Offset configuration for the controller is non-zero, the address in the message requests to the controller will be offset from the registers listed in these tables.

Default map

Valid address ranges, with their applicable register sizes, are listed below for a 'default' Enron Modbus implementation. The Experion Enron Modbus Interface will use this register map when the controller Device Type is set to *DEFAULT*.

Table 1: Default map

Register	Description	Default data format
32	Event/Alarm data (EFM support).	See EFM description
700–799	Transaction log data (EFM support).	See EFM description
1000–1999	Boolean Variables. Each address is 1-bit wide.	WORD
3000–3999	Short Integer registers. Each address is 16-bits wide.	WORD
5000–5999	Long Integer registers. Each address is 32-bits wide.	U32B
7000–7999	Floating Point registers. Each address is 32-bits wide.	IEEEFP

Write access to any register is device—and address—specific. No checks are made at configuration time against write access for any registers configured in a scanned parameter destination address. Many device vendors have chosen to deviate from the Modbus and/or Enron Modbus standards, most typically with variations to the register mappings above. The Enron Modbus Interface provides a configurable 'Device Type' for the Controller to enable selection of specific device types that deviate from the Enron Specification.

NuFlo Cameron Scanner 2000 MicroEFM map

Valid address ranges, with their applicable register sizes, are listed below for a *NuFlo Cameron Scanner 2000 MicroEFM* Enron Modbus implementation. The Experion Enron Modbus Interface will use this register map when the controller Device Type is set to *Cameron Scanner 2000*.

Table 2: NuFlo Cameron Scanner 2000 MicroEFM map

Register	Description	Default data format
32	Event/Alarm data (EFM support).	See EFM description
700–701	Transaction log data (EFM support, 700 = Interval, 701 = Daily).	See EFM description
1000–1423	Mixed data types. Each address is 16-bits wide.	WORD
2000–2900	Mixed data types. Each address is 16-bits wide.	WORD
3000–3268	Mixed data types. Each address is 16-bits wide.	WORD
4000–4067	Mixed data types. Each address is 16-bits wide.	WORD
7000–7338	Floating point registers. Each address is 32-bits wide.	IEEEFP
8000–8676	Floating point registers. Each address is 16-bits wide.	WORD
9000–9048	User–defined register configuration. Each address is 16-bits wide.	WORD
9100–9148	User–defined registers. Each address is 16-bits wide.	WORD
9900–9905	Device status U32 values. Each address is 16-bits wide.	WORD

GE and Modicon

Both GE and Modicon controllers can use asynchronous Modbus communications with the Enron Modbus interface.

You can use the information in “Table 3: APL Modicon Combined map” to access a limited range of registers within each data table on a single controller.

The remaining tables provide access to only a single data table. Therefore, if you need to access multiple data types in one controller, you need to configure a separate *logical* Experion controller for each single data table on the *physical* controller.

Any differences in byte/word order are handled by Experion Point Data Formats, and as such, the same Enron Modbus Device Type can be applied to either controller type.

Attention

It is expected that the **Address Offset** on the controller is set to *1* to account for the offset applied to the address that is sent in the Modbus message in standard Modbus protocol (this is not applied with *Enron Modbus* variation, which has offset of *0*).

In each of the tables below, *Low Address* and *High Address* represent the address as it is entered for point configuration in Quick Builder. The *(msg)* shown below each address is the address that is sent in the Modbus message for the expected configuration of controller address offset equal to *1*.

Table 3: APL Modicon Combined map

Low Address (msg)	High Address (msg)	Data Type (Read/Write)	Max Request	Modbus Function Code			Register Size
				Read	Single Write	Multi-Write[1]	
1 (0)	4000 (3999)	Digital Coil (Read/Write)	2,000	1	5	15	1–bit
10001 (0)	14000 (3999)	Digital Input (Read)	2,000	2	N/A	N/A	1–bit
30001 (0)	34000 (3999)	Analog Input Registers (Read)	125	4	N/A	N/A	16–bit
40001 (0)	60000 (19999)	Holding Register (Read/ Write)	125	3	6	16	16–bit

Table 4: APL Modicon Digital Coil map

Low Address (msg)	High Address (msg)	Data Type (Read/Write)	Max Request	Modbus Function Code			Register Size
				Read	Single Write	Multi-Write[1]	
1 (0)	10000 (9999)	Digital Coil (Read/Write)	2,000	1	5	15	1–bit

Table 5: APL Modicon Digital Input map

Low Address (msg)	High Address (msg)	Data Type (Read/Write)	Max Request	Modbus Function Code			Register Size
				Read	Single Write	Multi-Write[1]	
10001 (0)	30000 (19999)	Digital Input (Read)	2,000	2	N/A	N/A	1–bit

[1] **Multi-write function code is used for writes to consecutive registers used for a single *point.parameter* value (for example, consecutive bits 8-state status point, or floating point from two consecutive 16–bit registers), but writes for multiple parameters are treated as individual writes (although each of those may use a multi-write for the individual parameter value).**

Table 6: APL Modicon Input Register map

Low Address (msg)	High Address (msg)	Data Type (Read/Write)	Max Request	Modbus Function Code			Register Size
				Read	Single Write	Multi-Write[1]	
30001 (0)	40000 (9999)	Analog Input Registers (Read)	125	4	N/A	N/A	16–bit

Table 7: APL Modicon Holding Register map

Low Address (msg)	High Address (msg)	Data Type (Read/Write)	Max Request	Modbus Function Code			Register Size
				Read	Single Write	Multi-Write[1]	
40001 (0)	60000 (19999)	Holding Register (Read/ Write)	125	3	6	16	16–bit

Modbus

Modbus controllers can use asynchronous Modbus communications with the Enron Modbus interface.

You can use the information in “Table 8: Modbus Combined map” to access a limited range of registers within each data table on a single controller.

The remaining tables provide access to only a single data table. Therefore, if you need to access multiple data types in one controller, you need to configure a separate *logical* Experion controller for each single data table on the *physical* controller.

Any differences in byte/word order are handled by Experion Point Data Formats.

Attention

It is expected that the **Address Offset** on the controller is set to *1* to account for the offset applied to the address that is sent in the Modbus message in standard Modbus protocol (this is not applied with *Enron Modbus* variation, which has offset of *0*).

In each of the tables below, *Low Address* and *High Address* represent the address as it is entered for point configuration in Quick Builder. The *(msg)* shown below each address is the address that is sent in the Modbus message for the expected configuration of controller address offset equal to *1*.

Table 8: Modbus Combined map

Low Address (msg)	High Address (msg)	Data Type (Read/Write)	Max Request	Modbus Function Code			Register Size
				Read	Single Write	Multi-Write[1]	
1 (0)	8000 (7999)	Digital Coil (Read/Write)	2,000	1	5	15	1–bit
10001 (0)	18000 (7999)	Digital Input (Read)	2,000	2	N/A	N/A	1–bit
30001 (0)	38000 (7999)	Analog Input Registers (Read)	125	4	N/A	N/A	16–bit
40001 (0)	48000 (7999)	Holding Register (Read/ Write)	125	3	6	16	16–bit

Table 9: Modbus Digital Coil map

Low Address (msg)	High Address (msg)	Data Type (Read/Write)	Max Request	Modbus Function Code			Register Size
				Read	Single Write	Multi-Write[1]	
1 (0)	32765 (32764)	Digital Coil (Read/Write)	2,000	1	5	15	1–bit

Table 10: Modbus Digital Input map

Low Address (msg)	High Address (msg)	Data Type (Read/Write)	Max Request	Modbus Function Code			Register Size
				Read	Single Write	Multi-Write[1]	
1 (0)	32765 (32764)	Digital Input (Read)	2,000	2	N/A	N/A	1–bit

Table 11: Modbus Input Register map

Low Address (msg)	High Address (msg)	Data Type (Read/Write)	Max Request	Modbus Function Code			Register Size
				Read	Single Write	Multi-Write[1]	
1 (0)	32765 (32764)	Analog Input Registers (Read)	125	4	N/A	N/A	16–bit

Table 12: Modbus Holding Register map

Low Address (msg)	High Address (msg)	Data Type (Read/Write)	Max Request	Modbus Function Code			Register Size
				Read	Single Write	Multi-Write[1]	
1 (0)	32765 (32764)	Holding Register (Read/ Write)	125	3	6	16	16–bit

Related topics

“Main properties for an Enron Modbus controller” on page 25

Enron Modbus channel and controller reference

This section describes the configuration and addressing information specific to Enron Modbus channels and controllers.

In addition to the information contained in this reference, and for help to build channels and controllers, see the section titled "Building controllers or channels" in the *Quick Builder User's Guide*.

Related topics

“Main properties for an Enron Modbus channel” on page 18

“Port properties for an Enron Modbus channel” on page 20

“Redundant port properties for an Enron Modbus channel” on page 22

“Alternating behavior of redundant Enron Modbus controllers” on page 23

“Main properties for an Enron Modbus controller” on page 25

“About time synchronization on Enron Modbus controllers” on page 28

“Optimizing Enron Modbus scanning performance” on page 29

“Planning considerations for installing and configuring Enron Modbus controllers” on page 5

Main properties for an Enron Modbus channel

The Main tab defines the basic properties for an Enron Modbus channel.

For information about how to create a channel, see the topic titled "Building controllers and channels" in the *Quick Builder User's Guide*.

Property	Description
Name	The unique name of the channel. A maximum of *10* alphanumeric characters (no spaces or double quotes). Note: In Station displays, underscore characters (_) appear as spaces.
Description	(Optional) A description of the channel. A maximum of *132* alphanumeric characters, including spaces.
Associated Asset	The Tag Name of the Asset to be associated with the alarm group.
Marginal Alarm Limit	The communications alarm marginal limit at which the channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one channel, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. A channel barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by two or more, depending on the error, and decrements for each good call. To calculate an acceptable marginal alarm limit, use the formula: Square root of the number of controllers on the channel × Marginal Alarm Limit defined on those controllers (Normally, you specify the same value for all controllers on a channel). For example, if there are 9 controllers on the channel and their Marginal Alarm Limit is set to *25*, the value would be *3* (which is the square root of 9) *x 25 = 75*.
Fail Alarm Limit	The communications alarm fail limit at which the channel is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one channel, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. Set this to double the value specified for the channel Marginal Alarm Limit.
Diagnostic Scan Rate	The period, in seconds, between diagnostic scans that verify communications integrity with the controller. The default value is *60* seconds. The diagnostic scans continue even if a controller is marked as failed, thus enabling the system to detect return-to-normal communications. If there are multiple controllers on a channel, the diagnostic scan rate should be set to 60 seconds multiplied by the number of controllers on that channel. For example, if there are five controllers on the channel, the diagnostic scan rate should be set to 300 seconds. Diagnostic scans are done at the set scan rate per controller. Therefore, with five controllers and a default value of 60 seconds, the diagnostic scans will be done every 12 seconds.
Write Delay	Reserved for future use. (Applicable only to Serial Port) The number of milliseconds that the server waits before writing to the controller. The default value is *10* ms.
Connect Timeout	The time, in seconds, the server attempts to connect to a controller before giving up. For Modbus TCP protocol, it is recommended to use a timeout of *1* or *2* seconds, unless a loaded Ethernet network necessitates a higher timeout value. The default value is *10* seconds.

Property	Description
Read Timeout	The time, in seconds, the server attempts to read data from a controller before giving up. The default value is *2* seconds. For Modbus TCP protocol, it is recommended to use a timeout of *1* second, unless a loaded Ethernet network necessitates a higher timeout value.
Default Modbus Protocol	(Optional) Sets the default Modbus protocol (Modbus TCP, Modbus RTU, Modbus ASCII) for the channel. The default value is *None*. The setting can be overridden on a controller-by-controller basis using the Modbus protocol property on the controller.
Asynchronous Option	Defines what level of asynchronous communications can be used on the channel. This will depend on the communications architecture and end device support. • *Asynchronous* (Default) Full asynchronous communications. There may be multiple open transactions to any or all RTUs across both links. This is applicable only to Modbus TCP connections direct to the end devices. • *Synchronous on RTU* Synchronous communication for requests to an RTU. Only one request can be open on each RTU. The communications remains asynchronous across channel and link. • *Synchronous on Link* Synchronous communications for requests down each link. A total of two requests can be open on the channel, and they must be to different RTUs with one open request on each link. • *Synchronous on Channel* Synchronous communications for all requests across the entire channel. Only one transaction can be open at a time. The default is *Asynchronous*. Modbus ASCII or RTU connections via direct serial, terminal server, and Modbus TCP bridge need to be set either to '*Synchronous on Link*' or '*Synchronous on Channel*.' (That is, if the **Port Type** setting on the **Port** tab or **Redundant Port** tab is either '*Serial*' or '*TerminalServer*,' you cannot choose '*Asynchronous*' or '*Synchronous on RTU*.')
Item Type	The type of channel specified when this item was created.
Last Modified	The date and time the channel properties were modified.
Last Downloaded	The date and time the channel was last downloaded to the server.
Item Number	The unique item number currently assigned to this channel, in the format *CHNcc*, where *cc* is the channel number. You can change the item number if you need to match your current server database configuration. The number must be between *01* and the maximum number of channels allowed for your system. For more information about setting the maximum value, see the topic titled "Adjusting sizing of non-licensed items" in the *Supplementary Installation Tasks Guide*.

Port properties for an Enron Modbus channel

The Port tab defines the communication-related properties for a channel. The properties vary according to the selected **Port Type**:

- *Serial*. This connection method is not fully supported for the Experion system. Select this if you are using an RS-232 serial port. See the section below titled "Serial port properties."
- *TerminalServer*. Select this if you want to connect the controller to the server via a LAN. See the section below titled "TerminalServer port properties."
- *LANVendor*. Select this if you want to connect the controller over a TCP network using Enron Modbus TCP, RTU over TCP, or Enron Modbus RTU via a Modbus Bridge device. You can set the protocol choice in the Enron Modbus channel properties or controller properties.

Attention
Set the port properties to the same values as those you specified when you configured the controller.

Serial port properties

Property	Description
Serial Port Name	The device name of the serial port.
Baud	The number of data bits per second. The default is *9600*.
Number of Data Bits	The number of data bits used for transmission. The default is *8*.
Stop Bits	The number of stop bits used for transmission The default is *1*.
Parity	Defines parity verification of each character and must match configuration on the end device. The default is *NONE*.
Checksum	Specifies any checksum being added to the frame in addition to the LRC/CRC check already included in Modbus ASCII/RTU protocols respectively. Not applicable, leave as *NONE*.
XON/XOFF	The type of XON/XOFF software flow control used to stop a receiver from being overrun with messages from a sender. The types are: • *Input* (use XON/XOFF to control the flow of data on the receive line) • *None* (default) • *Output* (use XON/XOFF to control the flow of data on the transmit line)
RS-232	Applicable only if the controller is connected to the RS-232 COM port. **Enable RTS/CTS flow control**. Select this if you want to use RTS/CTS for flow control to stop a receiver from being overrun with messages from a sender. **Detect DCD**. Select this if the Data Carrier Detect communication status line of the COM port requires monitoring (usually when using modem or microwave linking). When selected, the communications fails if the desired COM status line is not high—for example, on a dial-up link connection for a modem. **Detect DSR**. Select this if the Data Set Ready communication status line of the COM port requires monitoring (usually when using modem or microwave linking). When selected, the communications fails if the desired COM status is not achieved.

Property	Description
RS-485	Applicable only if the controller is connected to the RS-485 COM port. **Enable Stallion RS-485 Half Duplex**. Select if RS-232 to RS-485 uses a Stallion EasyConnection adapter. **Echo (Required for Stallion RS-485 ports)**. Select so that the server expects the messages it sends to the port on the transmit line to be echoed back on the receive line. Select for a Stallion EasyConnection adapter or a Black Box converter.

TerminalServer port properties

Property	Description
Terminal Server TCP Host Name Terminal Server TCP Port No	The name and port number of terminal server to which the channel is connected. You can specify either a TCP host name or an IP address, but it must match the TCP host name used when you installed and internally configured the terminal server.
Idle Timeout	The time, in seconds, the channel waits for a successful connection to the server before closing the connection. A value of *0* indicates that the connection is never closed.
Checksum	The type of checksum error detection used for the port. Not applicable for this channel. Select *NONE*.

LANVendor port properties

Set the protocol choice in the Enron Modbus controller properties.

Redundant port properties for an Enron Modbus channel

The **Redundant Port** tab defines the communication-related properties for a redundant Enron Modbus channel. The **Port Type** selected on this tab must be the same as the port type selected on the **Port** tab.

Property	Description
Port Type	• *None* (Default) • *RedundantSerial* • *RedundantTerminalServer* • *RedundantLANVendor*

Alternating behavior of redundant Enron Modbus controllers

The alternating behavior of redundant Enron Modbus controllers is defined on the controller's **Main** tab in Quick Builder. Note that the active link on the channel or controller has to fail for the other link to be used. You can force a controller to use the other link by disabling the currently active link.

If alternate links are enabled

If the **Disable Alternate Polling** check box is not selected (the default), requests will alternate between link A and B. Whenever a control is performed to a parameter (for example, changing the Set Point in Station), a write request will be performed to one link, followed by a read request for the same address on the alternate link. If the user is not performing any controls, then the read requests will be made at the requested update period and will alternate between links.

For example:

User changes Set Point in Station.

Write the value on link A.

Read the same address for confirmation on link B.

User changes Set Point in Station.

Write the value on link A.

Read the same address for confirmation on link B.

…

Server does a periodic scan:

Read the address on link A.

Server does a periodic scan:

Read the address on link B.

Server does a periodic scan:

Read the address on link A.

…

User changes Set Point in Station.

Write the value on link B.

Read the same address for confirmation on link A.

If alternate links are disabled

If the **Disable Alternate Polling** check box is selected, then all requests will be made on the same link until the link fails.

For example:

User changes Set Point in Station.

Write the value on link A.

Read the same address for confirmation on link A.

User changes Set Point in Station.

Write the value on link A.

Read the same address for confirmation on link A.

…

Server does a periodic scan:

Read the address on link A.

Server does a periodic scan:

Read the address on link A.

Server does a periodic scan:

Read the address on link A.

…

Link A is marked as failed.

…

User changes Set Point in Station.

Write the value on link B.

Read the same address for confirmation on link B.

User changes Set Point in Station.

Write the value on link B.

Read the same address for confirmation on link B.

…

Server does a periodic scan:

Read the address on link B.

Server does a periodic scan:

Read the address on link B.

Server does a periodic scan:

Read the address on link B.

Related topics

“Main properties for an Enron Modbus controller” on page 25

Main properties for an Enron Modbus controller

The **Main** tab defines the basic properties for an Enron Modbus controller.

For information about how to create a controller, see the topic titled "Building controllers and channels" in the *Quick Builder User's Guide*.

Attention
When Electronic Flow Measurement (EFM) is collecting configuration logs for meters associated to the controller, dynamic scanning must be enabled on the controller.

Property	Description
Name	The unique name of the controller. A maximum of 10 alphanumeric characters (no spaces or double quotes). For LAN-connected controllers, the name must not contain underscore characters (_). This name is used to look up the in the Hosts file or DNS if you do not specify an **IP Address** property. In the case of communications redundancy when the IP Addresses are not defined in Quick Builder, the IP Address 1 and 2 must be specified in the Server hosts file. The host name for IP Address 1 is then the Name property with an "A" appended to it and the host name for IP Address 2 is the Name property with a "B" appended to it.
Description	(Optional) A description of the controller. A maximum of *132* alphanumeric characters, including spaces.
Associated Asset	The Tag Name of the Asset to be associated with the alarm group.
Channel Name	The name of the channel on which the controller communicates with the server. (You must have already defined a channel for it to appear in this list.)
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this limit is reached, a high priority alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one controller, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. A controller barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by two or more, depending on the error, and decrements for each good call. The default value is *25*.
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one controller, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. Set this to double the value specified for the controller Marginal Alarm Limit. The default is *50*.
Dynamic Scanning Fastest Scan Period	Select the **Dynamic Scanning** check box to enable dynamic scanning of all point parameters on this controller. The default setting for this check box is selected. Define the fastest possible scan period (in seconds) that dynamic scanning will scan point parameters on this controller. The default is *15* seconds. The dynamic scanning period does not affect the static scanning rate for a parameter. For example, if the scanning rate for a parameter is 10 seconds, and the dynamic scanning rate for the controller is 15 seconds, the parameter will still be scanned at a period of 10 seconds.

Property	Description
Slave ID	ID of the controller, specified when the controller was configured.
Modbus Protocol	The Modbus protocol that will be used to communicate with this controller. *Modbus TCP*, *Modbus RTU*, *Modbus ASCII*, or *Channel default*. The default protocol is *Modbus TCP*. If *Channel default* is selected, the Modbus Protocol for this controller is determined from the Modbus Protocol property for the channel to which controller is assigned. All controllers on the same channel must be configured with the same Modbus protocol.
IP Address 1 and 2	**IP Address 1** and **Port No.** are visible only when the port type of the associated channel is *LANVendor*. **IP Address 2** and **Port No.** are visible only when the redundant port type of the associated channel is *RedundantLANVendor*. Network address of the controller. IP Address 2 is used only when the controller has a redundant network adapter card and the associated channel has been defined as a redundant channel. If the IP Address is not specified, the controller name is used as the TCP host name. For further information, see the **Name** property.
Port No. 1 and 2	The TCP port numbers for communicating with the Ethernet port of the Enron Modbus controller or Modbus TCP Bridge device. The ability to define a specific port enables multiple Modbus devices to be addressed behind a single .
Disable Alternate Polling	Visible only when the associated channel's redundant port type is *RedundantSerial*, *RedundantTerminalServer*, or *RedundantLANVendor* (that is, not *None*). If selected, the failover from active link will only occur when the active link has failed or been disabled. If not selected (default), the scans are alternated between the links. In a redundant controller configuration, it is recommended that you clear this check box so as to enable faster recovery of the master controller connection. Note that a channel or controller has to fail in order for a non-alternating channel to use the other link. You can also force a channel to use the other link by disabling the currently active link. See the topic titled "Alternating behavior of redundant Enron Modbus controllers" for more information.
Diagnostic Address	The valid address in the controller to read for the diagnostic scan.
Device Type	The type of device connected to the controller. Your choice affects which register map will apply. See the topic titled "Enron Modbus register map" for more information. • *DEFAULT* – For all devices that adhere to the *Specifications and Requirements for an Electronic Flow Measurement Remote Terminal Unit for Enron Corp*. This is the default value. • *Cameron Scanner 2000* – NuFlo Cameron Scanner 2000 MicroEFM. • *OTHER* – Support for future device types. (Experion documentation will instruct you when to select this option.) When selected, type the keyword for the device in the **Other device type** box. (Maximum 20 characters. No spaces or '=' characters.) To change the device type, you must first delete any points that are built on the controller.
Time Synchronization Enable at	If selected, time synchronization is enabled for the device. The default is not selected (disabled). Type the time (in 24–hour format: *HH:SS*) at which synchronization will occur. The default value is midnight (*00:00*).
Address Offset	Optional. Type an offset between the address configured for the points (or EFM addresses) and the address that is sent to the device in the Modbus message. Valid range is range *–65535* to *+65535*. For example, an address offset value of *1* means the address entered in Quick Builder will be one greater than the address that is sent in the Modbus message.

Property	Description
Time Format	(Applicable only when time synchronization is enabled.) The format that the time is written to the device at the nominated scheduled time each day. The time format is device-specific. • *Cameron Scanner 2000* – NuFlo Cameron Scanner 2000 MicroEFM. • *Daniels 2500* – Daniels 2500 flow computer. • *OTHER* – Used for devices not listed. When selected, type the keyword for the device in the **Other time format** box. (Maximum 20 characters. No spaces or '=' characters.) See the topic titled "About time synchronization on Enron Modbus controllers" for more information.
Time Zone UTC	(Applicable only when time synchronization is enabled.) The time zone at which the device is located, specified in UTC offset. Type the UTC offset, plus (+) or minus (–), in 24–hour format: *HH:SS*. The default value is UTC (*+00:00*); that is, no UTC offset. For example, for a device located in California, USA, the UTC offset should be set to *–08:00*. For a device located in the United Arab Emirates, the UTC offset should be set to *+04:00*. The offset will always be applied to UTC time regardless of daylight savings settings on the server or the controller.
Item Type	The type of controller specified when this item was created.
Last Modified	The date and time the controller properties were modified.
Last Downloaded	The date and time the controller was last downloaded to the server.
Item Number	The unique item number currently assigned to this controller, in the format *RTUnnnnn*. You can change the item number if you need to match your current server database configuration. The number must be between *01* and the maximum number of controllers allowed for your system. For more information about setting the maximum value, see the topic titled "Adjusting sizing of non-licensed items" in the *Supplementary Installation Tasks Guide*.

Related topics

“Alternating behavior of redundant Enron Modbus controllers” on page 23
“Enron Modbus register map” on page 11
“About time synchronization on Enron Modbus controllers” on page 28

About time synchronization on Enron Modbus controllers

When choosing to enable time synchronization on an Enron Modbus controller, you need to define the correct time format, based on the device type. This is critical when you use EFM and the controller's time source is not synchronized by any other means. For example, some controllers have their own GPS-based time source or use NTP over TCP/IP to synchronize their time, in which case you do not need to synchronize using the Enron Modbus interface.

Enron Modbus controllers do not prescribe any method of time synchronization. Time synchronization is supported at the device. The following sections describe how each supported device uses write formats for setting the realtime clock.

NuFlo Cameron Scanner 2000

The time format for NuFlo Cameron Scanner 2000 writes to a series of integer registers, where:

Register value	Description
Register	Year%100 (that is, real year in device is 2000 + register value)
Register + 1	Month: *1–12*
Register + 2	Day: *1–31*
Register + 3	Hour: *0–23*
Register + 4	Minute: *0–59*
Register + 5	Second: *0–59*

The write is issued in a single write to all registers.

Cameron Scanner Register = *1200*.

Daniels 2500 flow computer

The time format for Daniels 2500 flow computers writes an ASCII date and time string to register 1 in the form *DD-MMM-YYYY hh mm ss*, where:

Item	Description
DD	The two-digit day: 01–31
MMM	The month: *JAN*, *FEB*, *MAR*, *APR*, *MAY*, *JUN*, *JUL*, *AUG*, *SEP*, *OCT*, *NOV*, or *DEC*
YYYY	Year
hh	Hour, in 24-hour format: *00–23*
mm	Minute: *00–59*
ss	Second: *00–59*

For example: *28-NOV-2014 12 30 00*.

Related topics

"Main properties for an Enron Modbus controller" on page 25

Optimizing Enron Modbus scanning performance

An Enron Modbus scan packet:

- Can contain a maximum number of addresses using the following rule: *Number of addresses x Register width <= 250 bytes*.
- Cannot contain addresses with different register widths.
- Must have addresses with the same scan period, unless *dynamic scanning* is enabled.

Two types of scan packet are built for Enron Modbus:

- **Hardware Diagnostic**. One scan per controller at a defined regular interval (the default is 60 seconds) to verify communications integrity with the controller. This scan packet is automatically created to scan the diagnostic address configured for the controller.
- **Periodic Data Acquisition**. A defined regular interval in which the server database acquires information from the addressed registers in the controller and processes the values as point parameters.

  You need to define the scan period for each point parameter source address. The scan period should reflect both the rate at which the value held in memory changes and its importance to the process (critical or non-critical).

  There is one periodic data acquisition scan per scan packet.

Reducing the number of scan packets for Enron Modbus controllers

Attention
This topic is applicable only when *Dynamic scanning* is disabled.

If the number of scan packets becomes too great, scanning performance is impaired.

To minimize the number of scan packets, use a small number of available scan periods for all your point definitions.

Closely block the registers read by the server and ensure that all addresses within a block have the same scan period.

Make each Modbus scan packet as close to the maximum size as possible and ensure there are no small packets being scanned at fast rates.

The scan packets that have been built can be listed by using the utility **lisscn** (list scan). Listing scan packets helps verify the scanning strategy.

For more information about **lisscn**, see the section titled "Command Reference" in the *Server and Client Configuration Guide*.

Enron Modbus points reference

This section describes how to configure points for an Enron Modbus controller using Quick Builder.

In addition to the information contained in this reference, and for help to build points, see the section titled "Building and configuring points" in the *Quick Builder User's Guide*.

Related topics

“Defining an Enron Modbus address for a point parameter” on page 32

Defining an Enron Modbus address for a point parameter

For **PV Source Address**, **Source Address**, and **Destination Address** the format for an Enron Modbus controller address is:

ControllerName FullAddress

Part	Description
ControllerName	The name of the Enron Modbus controller.
FullAddress	The address within the controller where the value is stored. See the topic titled "Address syntax for Enron Modbus controllers" for more information.

For help when defining an address, click ... next to **Address** to display Address Builder.

Address syntax for Enron Modbus controllers

The format for the address is:

Address [*DataFormat* | B:*nn* W:*nn*]

Part	Description
Address	The address (decimal) for the register.
DataFormat	The data format acronym, depending on how you want the value to be read: • Data format for scaling 16-bit integers. • Data format for reading floating point values. • Data format for reading raw values without scaling. See the sections below titled "Data format support," "Unscaled data formats," and "Data formats for scaling." If you want to use a user-defined data format, you must define the format on the server. See the section titled "About user-defined data formats" in the *Server and Client Configuration Guide* for more information.
B:nn W:nn	*B:nn*. (Optional) Bit specifier for bit, or partial word, data format. The *nn* specifies the bit number (0 based) within a word at the start of the partial word. If omitted, the default is assumed to be *B:0*. *W:nn* (Optional) Width specifier for bit or partial word data format. The *nn* specifies the number of bits from the *B:nn* specifier, which is then used to read the bit or partial word value. If omitted, the default is assumed to be width of the register for the given address.

Analog point

PV source: *3001 U4095*

SP destination: *3002 U4095*

Mode destination: *3003 B:1 W:1*

Status point

PV source: *1001*

Accumulator point

PV source: *5001 U32B*

Data format support

The following data types are supported for registers of given size as described below:

- Single-bit, multi-bit (consecutive), and non-consecutive-bit data format for Boolean registers, and for nominated bit(s) within Short Integer and Long Integer registers.
- Word (16-bit) integer scaled and unscaled data formats for Short Integer registers.
- DWord (32-bit) integer scaled and unscaled data formats for Short Integer (2 consecutive 16-bit registers) and Long integer (32-bit) registers.
- 32-bit Floating Point for 2 consecutive Short Integer registers, or single 32-bit Long Integer or Floating Point register (IEEEFP* data formats).

If no data format is supplied for the address, the default data format as listed in the topic titled "Enron Modbus register map" will be applied.

Unscaled data formats

Data Format	Description
WORD	Whole word
HALFWD	Upper half word
REAL	Native real (float)
INT4	Native int4 (signed int)
DBLE	Native dble (double)
C16	16-bit counter
C3BCD	3-digit BCD 0–999 counts
C4BCD	4-digit BCD 0–9,999 counts
C8BCD	8-digit BCD 0–99,999,999 counts
IEEEFP	IEEE Floating Point (Big Endian)
IEEEFPL	IEEE Floating Point (Little Endian)
INT2	Native int2 (signed short)
IEEEFPBB	Byte-swapped Big Endian float
IEEEFPLB	Byte-swapped Little Endian float
S32BB	32-bit signed binary Big Endian
U32BB	32-bit unsigned binary Big Endian
S32B	32-bit signed binary
U32B	32-bit unsigned binary
FENUM	Enumerated integer
REVWD	Reverse word

Data formats for scaling

You can scale point parameter values to the range of the PV with a scaled data format. Select the format that corresponds to the counts that have been set in the controller register.

Data Format[2]	Counts in Controller Register
U1023	0–1,023 (U=unsigned)

[2] **S = Signed**

U = Unsigned

Data Format[2]	Counts in Controller Register
U4095	0–4,095
U9999	0–9,999
S9999	–9,999–9,999 (S=signed)
U999	0–999
U3BCD	3 digit BCD 0–999
U4BCD	4 digit BCD 0–9999
U6BCD	6 digit BCD 0–999,999
U8BCD	8 digit BCD 0–99,999,999
UBCD16	4 digit BCD 0–4,095
UBCD12	3 digit BCD 0–410
U16B	16 bit unsigned binary
S16B	16 bit signed binary
E3BCD	3 digit BCD with error status
U100	0 to 100 percent
U15B	15 bit unsigned binary
U14B	14 bit unsigned binary
SLC_AI	A-B SLC Analog Input 3,277–16,384
SLC_AO	A-B SLC Analog Output 6,242–31,208
U9998	0–9,998 for Square D AI
S8B	8 bit signed binary
U8B	8 bit unsigned binary
D9999	double integer for A-B QCL
S32BS	32 bit signed binary
U32BS	32 bit unsigned binary
S32BSB	32 bit signed binary Big Endian
U32BSB	32 bit unsigned binary Big Endian

Attention

If auxiliary parameters have a data format type that requires scaling (U4095, U999, and so on), they take the same range as the PV.

[2] **S = Signed**

U = Unsigned

Enron Modbus EFM meter templates reference

This section describes the configuration information specific to meter templates on Enron Modbus controllers. See the "Building Electronic Flow Measurement (EFM)" section of the Quick Builder User's Guide for information about how to build EFM meter templates and equipment.

Related topics

"Main properties for an Enron Modbus meter template" on page 36
"Configuration Log properties for an Enron Modbus meter template" on page 37
"Interval Log properties for an Enron Modbus meter template" on page 38
"Daily Log properties for an Enron Modbus meter template" on page 39
"Alarm and Event properties for an Enron Modbus meter template" on page 40
"Data Export properties for an Enron Modbus meter template" on page 41

Main properties for an Enron Modbus meter template

This topic lists the Main tab settings for a meter template on an Enron Modbus controller. Different flow computers require different settings. The **Interval Log**, **Daily Log**, and **Alarm and Event Log** logs are all optional. However, you would configure at least one, if not all, of them. The flow computer vendor documentation should provide addressing information for addressing the Enron Modbus logs, with following information provided for immediate reference for NuFlo Cameron Scanner devices.

See the topic titled "Main properties for an EFM meter template" in the *Quick Builder User's Guide* for information about each field on this tab.

To make configuring meter templates an easier task, samples of Enron Modbus meter templates are included in Experion. See the topic titled "Managing EFM meter templates" in the *Quick Builder User's Guide* for information about how to import sample Enron Modbus meter templates.

NuFlo Cameron Scanner 2000 MicroEFM flow computer

TheNuFlo Cameron Scanner 2000 MicroEFM Flow Computer supports only one run. Use the following settings for **Interval Log**, **Daily Log**, and **Alarm and Event Log** for an Enron Modbus meter template.

Table 13: Interval, daily, and alarm and event settings for NuFlo Cameron Scanner 2000 MicroEFM flow computer runs

Property	Run 1
Interval Log	
Register	700
Pointer	7000
Rollover	2304 (no expansion board) 6392 (with expansion board)
Daily Log	
Register	701
Pointer	7001
Rollover	768
Alarm and Event Log	
Register	32
Pointer	7002
Rollover	1152

Related topics

“Planning considerations for installing and configuring Enron Modbus controllers” on page 5

Configuration Log properties for an Enron Modbus meter template

When creating meter templates for flow computers on Enron Modbus controllers, configuration log properties are specific to the flow computer. See the vendor documentation for the properties supported by the flow computer.

If the configuration log requires compliance with API21.1 then the configuration log properties configured here need to include at least those listed in the API21.1 standard.

See the topic titled "Configuration Log properties for an EFM meter template" in the *Quick Builder User's Guide* for information on how to add configuration properties to this tab.

Interval Log properties for an Enron Modbus meter template

The **Interval Log** tab appears only when the interval log is enabled on the **Main** tab.

When creating meter templates for flow computers on Enron Modbus controllers, the properties you collect for the interval log are specific to the flow computer. See the vendor documentation for the properties supported by the flow computer.

See the topic titled "Interval Log properties for an EFM meter template" in the *Quick Builder User's Guide* for information on how to add interval properties to this tab.

Daily Log properties for an Enron Modbus meter template

The **Daily Log** tab appears only when the daily log is enabled on the **Main** tab.

When creating meter templates for flow computers on Enron Modbus controllers, the properties you collect for the daily log are specific to the flow computer. See the vendor documentation for the properties supported by the flow computer.

See the topic titled "Daily Log properties for an EFM meter template" in the *Quick Builder User's Guide* for information on how to add properties to this tab.

Alarm and Event properties for an Enron Modbus meter template

The **Alarm and Event** tab appears only when the alarm and event log is enabled on the **Main** tab.

When creating meter templates for flow computers on Enron Modbus controllers, the properties you collect for the Alarm and Event Log are specific to the flow computer. See the vendor documentation for the properties supported by the flow computer.

See the topic titled "Alarm and Event properties for an EFM meter template" in the *Quick Builder User's Guide* for information on how to add properties to this tab.

Data Export properties for an Enron Modbus meter template

When creating meter templates for flow computers on Enron Modbus controllers, the data export formats you specify depend on the gas measurement system receiving the exported EFM data.

See the topic titled "Data Export properties for an EFM meter template" in the *Quick Builder User's Guide* for information on how to configure data export formats on this tab.

Troubleshooting Enron Modbus issues

This section describes troubleshooting tasks for Enron Modbus that you can perform either on the server or from any Station.

Related topics

“Testing Enron Modbus to field device communications” on page 44
“Troubleshooting Enron Modbus point configuration errors” on page 46

Testing Enron Modbus to field device communications

The first indication of an issue with communications to a field device using the Enron Modbus interface is that the error count on the controller (and channel) detail display will begin rising with the request count.

Use the following tips to help diagnose the cause of the errors:

1. Confirm that the diagnostic address configured on the controller is a valid address that can be read as a single register from the device.
2. View the Experion PKS Server Log (HLV application) for errors being logged on the channel.

   Adding an include filter for *pascn*rtu*N* where *N* is the controller of interest will filter the diagnostic log for error messages related to the controller of interest.

The following sections contain common examples of messages and their root cause.

Modbus TCP connection being attempted to a device which cannot be reached on the network

```
11-Jul-13  22:58:08.6162 ( -1  8484  5480 T00000000) pascn.exe:eth_open.c:812:
    select fail - timeout connecting to socket 672 port 53694 for rtu 14, link 1, lrn 77
11-Jul-13  22:58:08.6162 Errno 0x110:  Timeout on connect
11-Jul-13  22:58:08.6162 ( -1  8484  5480 T00000000) pascn.exe:scanio.h:364: CScanIO::
    Open(from rtu con).. hsc_eth_open(rtu=14, link=1, lrn=77, port=502) failed, error 0x110
```

To resolve:

1. Confirm that IP address and port configuration on the controller are configured correctly. Note that default Modbus TCP port for most devices is *502*.
2. Confirm that device is powered on, with network configuration matching IP and Port configured on the Experion controller, and any required gateway address is configured.
3. Use the 'ping' command on the server to verify the device is responding on the network.

Wrongly configured Asynchronous option

```
11-Jul-13  23:06:54.1171 ( -1  8484  5480 T00000000) pascn.exe:EnronRtu.cpp:891:
    CEnronRtu[rtu=14]::ProcessResponse(hTransaction=0x00974160, RequestType=3,
    ResponseBuffer[11,0x00490F80], pDecodedResponse=0x04EEFA60, pResponseKey=0x04EEFA78,
    pRequestBuffer=0x04EEFA70) .. ERROR: Response Length [4] not as expected [66]
```

To resolve:

1. Ensure that for the architecture and protocol in use the Asynchronous option has been set correctly. Unexpected response length errors may be a symptom of selecting for example asynchronous option for a terminal server Modbus RTU configuration. If there is any uncertainty about use of the Asynchronous option, select *Synchronous on Channel* to see if the errors clear.

Time in device is not being synchronized

If time synchronization has been configured for the controller, but the time is not being synchronized as expected check the following:

1. Confirm correct Time Format, which defines the register and time format being written, are correct for the device.
2. Ensure that the device is configured to accept writes to the register specified by the time format.
3. Ensure that the time zone offset has been set correctly for the controller.

Device is slow to respond (request timeout)

Log messages similar to the following example indicate that no response was received from the device within the READ timeout configured for the channel.

```
12-Jul-13  01:13:38.1015 ( -1  8484  5480 T00000000) pascn.exe:patransaction.cpp:550:
    CPATransaction::ExecuteState(acqreq<2:SCNTYP_HWARE rtu 14 lrn 0x803c adr 0x1b58 num 31
```

```
fmt 0x0> pTimeSyncQueue=0x0155B674) .. timed out waiting for response (elapsed time 2094ms,
max allowed time 2000ms)
```

To resolve:

1. Set the READ timeout to a longer period and use time data in the trace to determine a more appropriate configuration for the latency in the communications network.
2. Confirm that all addresses configured on points are valid for the device. *Lisscn* output for the controller can help by showing the requests that are going to be made to the device.

Advanced troubleshooting using trace

Any Enron Modbus channel can be traced to provide the raw data being written to and read from the device, including timing of the writes and reads. See trace documentation for use of trace. See Modbus.org documentation for format of Modbus messages.

Diagnostic Address Errors

If the diagnostic address is configured for an address to which the device does not respond, the following message may appear in the server diagnostic log.

```
11-Jul-13  23:10:03.1171 ( -1  8484  5480 T00000000) pascn.exe:patransaction.cpp:671:
    CPATransaction::ProcessResponse(acqreq<4:SCNTYP_DIAG rtu 14 lrn
    0x803c adr 0x0 num 0 fmt 0x0 inital>) .. pProtocolAdapter->ProcessResponse()
    failed with PAStatus=0xffffffff, fatal error
```

Related topics

“Planning considerations for installing and configuring Enron Modbus controllers” on page 5

Troubleshooting Enron Modbus point configuration errors

Point configuration errors appear when you download them from Quick Builder to the server. Most typically, the cause of these errors will be an invalid address.

Confirm that the register map for the chosen device type is correct for the field device. Note that some devices may not accept requests for all addresses in the range, and the controller vendor documentation should be consulted for valid addresses.

Notices

Trademarks

Experion®, PlantScape®, SafeBrowse®, TotalPlant®, and TDC 3000® are registered trademarks of Honeywell International, Inc.

OneWireless™ is a trademark of Honeywell International, Inc.

Other trademarks

Microsoft and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Trademarks that appear in this document are used only to the benefit of the trademark owner, with no intention of trademark infringement.

Third-party licenses

This product may contain or be derived from materials, including software, of third parties. The third party materials may be subject to licenses, notices, restrictions and obligations imposed by the licensor. The licenses, notices, restrictions and obligations, if any, may be found in the materials accompanying the product, in the documents or files accompanying such third party materials, in a file named third_party_licenses on the media containing the product, or at http://www.honeywell.com/ps/thirdpartylicenses.

Documentation feedback

You can find the most up-to-date documents on the Honeywell Process Solutions support website at:

http://www.honeywellprocess.com/support

If you have comments about Honeywell Process Solutions documentation, send your feedback to:

hpsdocs@honeywell.com

Use this email address to provide feedback, or to report errors and omissions in the documentation. For immediate help with a technical problem, contact your local Honeywell Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.

How to report a security vulnerability

For the purpose of submission, a security vulnerability is defined as a software defect or weakness that can be exploited to reduce the operational or security capabilities of the software.

Honeywell investigates all reports of security vulnerabilities affecting Honeywell products and services.

To report a potential security vulnerability against any Honeywell product, please follow the instructions at:

https://honeywell.com/pages/vulnerabilityreporting.aspx

Submit the requested information to Honeywell using one of the following methods:

- Send an email to security@honeywell.com.

  or

- Contact your local Honeywell Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.

Support

For support, contact your local Honeywell Process Solutions Customer Contact Center (CCC). To find your local CCC visit the website, https://www.honeywellprocess.com/en-US/contact-us/customer-support-contacts/Pages/default.aspx.

Training classes

Honeywell holds technical training classes on Experion PKS. These classes are taught by experts in the field of process control systems. For more information about these classes, contact your Honeywell representative, or see http://www.automationcollege.com.

Index

F

I

L

M

N

P

R

S

T

U

W