# Enron Modbus Driver

©2016 Kepware, Inc.

# Table of Contents

**Enron Modbus Driver**

Help version 1.037

**CONTENTS**

**Overview**
What is the Enron Modbus Driver?

**Channel Setup**
How do I configure channels for use with this driver?

**Device Setup**
How do I configure devices for use with this driver?

**Data Types Description**
What data types does this driver support?

**Address Descriptions**
How do I address a data location on an Enron Modbus device?

**Automatic Tag Database Generation**
How can I easily configure tags for this driver?

**Error Descriptions**
What error messages are produced by the Enron Modbus Driver?

## Overview

The Enron Modbus Driver provides real-time and EFM data access. The driver configuration retrieves historical and alarm/event archive data, and also maps data in the device to the server's EFM Model (which consists of various EFM attributes such as pressure, temperature, and so forth).

**Important:** EFM functionality is not available in all server versions. To determine whether support is available, refer to the "Server Summary Information" topic located in the server help file.

## Channel Setup

### Communication Serialization

The Enron Modbus Driver supports Communication Serialization, which specifies whether data transmissions should be limited to one channel at a time. For more information, refer to "Channel Properties - Advanced" in the server help file.

## Device Setup

### Supported Communication Parameters

Baud Rate: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 56000, 57600, 115200, 128000, and 256000
Parity: Odd, Even, and None
Data Bits: 5, 6, 7, and 8
Stop Bits: 1 and 2

**Note:** Not all of the listed configurations may be supported in every device.

### Maximum Number of Channels and Devices

The maximum number of channels supported by this driver is 256. The maximum number of devices supported per channel is 1024.

### Models

| Name | Address Bytes | Station ID Range |
|------|---------------|------------------|
| Standard | 1 | 0-255 |
| Extended Station ID | 2 | 0-65535 |
| Enron Modbus[1] | 1 | 0-255 |

*1: Deprecated. Older projects using model Enron Modbus will be converted to Standard when opened.*

### Ethernet Encapsulation

This driver supports Ethernet Encapsulation, which allows the driver to communicate with serial devices attached to an Ethernet network using a terminal server. It may be enabled through the Communications dialog in Channel Properties. *For more information, refer to the main server's help file.*

**See Also: Using Ethernet Encapsulation**

## Data Encoding Settings

This dialog specifies the device's data bit settings.
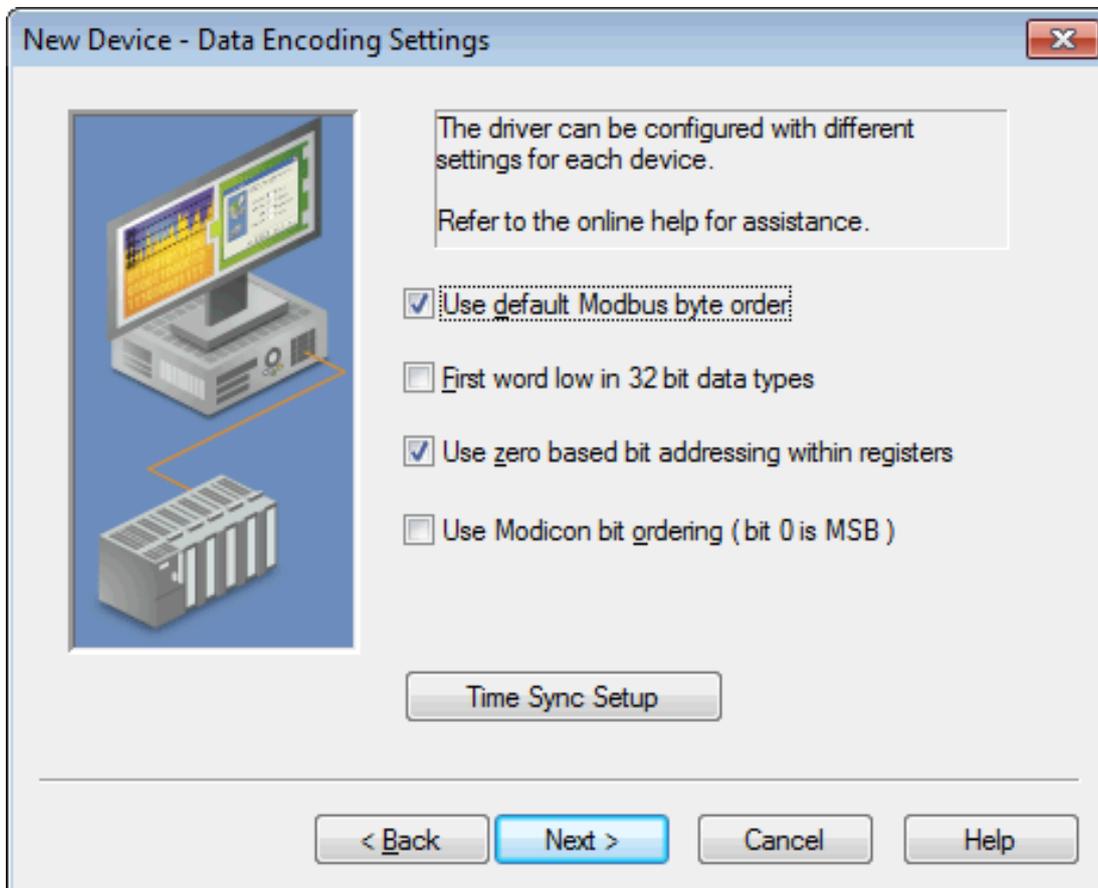
Descriptions of the parameters are as follows:

- **Use default Modbus byte order:** When unchecked, this option allows users to change the driver's byte order from the default Modbus byte ordering to Intel byte ordering. The default setting is checked, which is the normal setting for Modbus compatible devices.

  **Note:** If the device uses Intel byte ordering, unchecking this option will allow the Enron Modbus Driver to read Intel formatted data properly.
- **First word low in 32-bit data types:** Users can specify whether the driver should assume the first word is the low or high word of a 32-bit value. First word low follows the convention of the Modicon Modsoft programming software. The default setting is unchecked.
- **Use zero based bit addressing within registers:** When checked, this option will use zero based bit addressing within registers and start the first bit at 0. The default setting is checked.
- **Use Modicon bit ordering (bit 0 is MSB):** When checked, the driver will reverse the bit order on reads and writes to registers to follow the convention of the Modicon Modsoft programming software. For example, when enabled, a write to address 40001.0/1 will affect bit 15/16 in the device. The default setting is unchecked. For more information, refer to the "Use Modicon Bit Ordering" subtopic below.
- **Time Sync Setup:** When clicked, this button launches a dialog that may be used to assign a register address to the server's time synchronization attributes. For more information, refer to **Time Synchronization Mapping**.

### Zero vs. One Based Addressing Within Registers

Memory types that allow bits within Words can be referenced as a Boolean. The addressing notation for doing this is *<address>.<bit>*, where *<bit>* represents the bit number within the Word. Bit level addressing within registers provides two ways of addressing a bit within a given Word; Zero Based and One Based. Zero Based Bit Addressing within registers simply means that the first bit begins at 0. One Based Bit Addressing within registers means that the first bit begins at 1.

### Use Modicon Bit Ordering

For the following example, the 1st through 16th bit signifies either 0 to 15 bits or 1 to 16 bits, depending on whether the driver is set at zero based addressing within registers. In the tables below, MSB is the Most Significant Bit and LSB is the Least Significant Bit.

| MSB | | | | | | | | | | | | | | | LSB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

**Use Modicon Bit Ordering Unchecked**

| MSB | | | | | | | | | | | | | | | LSB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

### Time Synchronization Mapping

This dialog is a mapping of the device's real-time clock register addresses. It is only used when synchronizing the device clock. For more information, refer to "Device Properties - Time Synchronization" in the server help file.

Time Synchronization Mapping

This is a mapping of the device real-time clock register addresses to the server's time synchronization attributes. Assign a register address to each attribute. See the help file for more information.

| Time Sync Mapping | |
| --- | --- |
| Hour | 3014 |
| Minute | 3015 |
| Second | 3016 |
| Year | 3011 |
| Month | 3012 |
| Day | 3013 |

[ OK ]    [ Cancel ]    [ Help ]
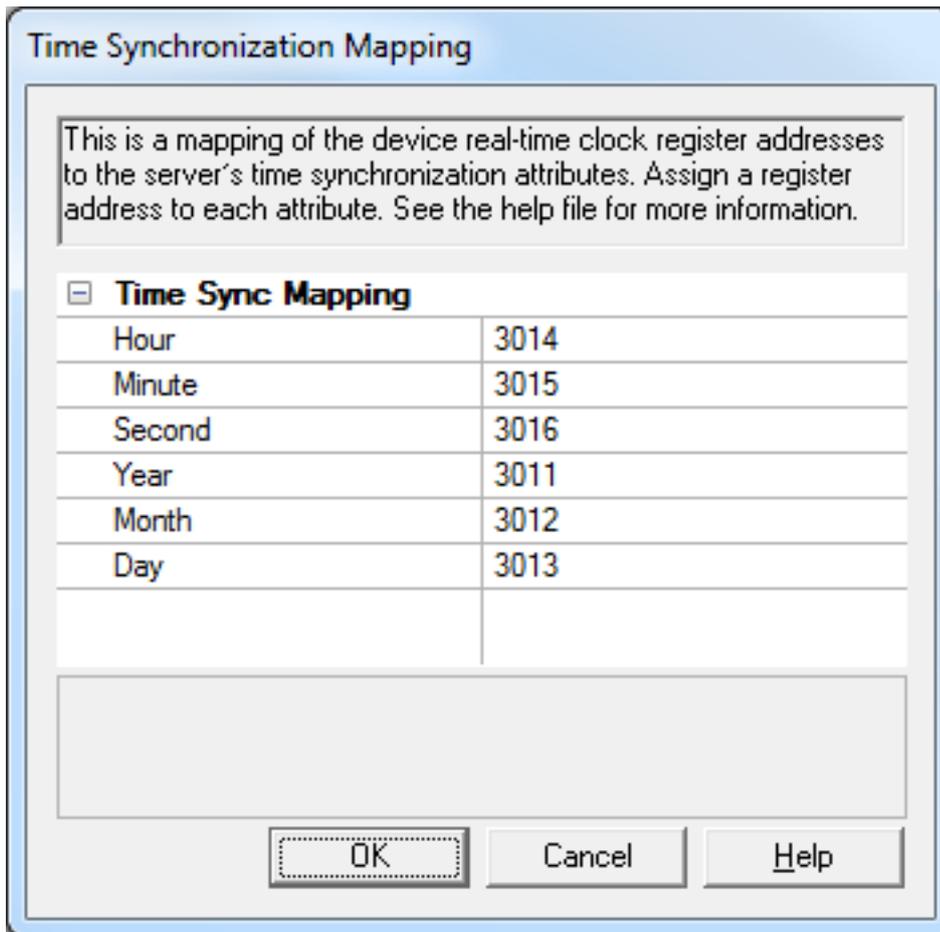
Descriptions of the registers are as follows:

- **Hour:** This is the address of the register containing the RTC hour.
- **Minute:** This is the address of the register containing the RTC minute.
- **Second:** This is the address of the register containing the RTC second.
- **Year:** This is the address of the register containing the RTC year.
- **Month:** This is the address of the register containing the RTC month.
- **Day:** This is the address of the register containing the RTC day.

**Important:** Addresses must fall within their defined ranges. For more information, refer to **Address Ranges**.

## Block Sizes
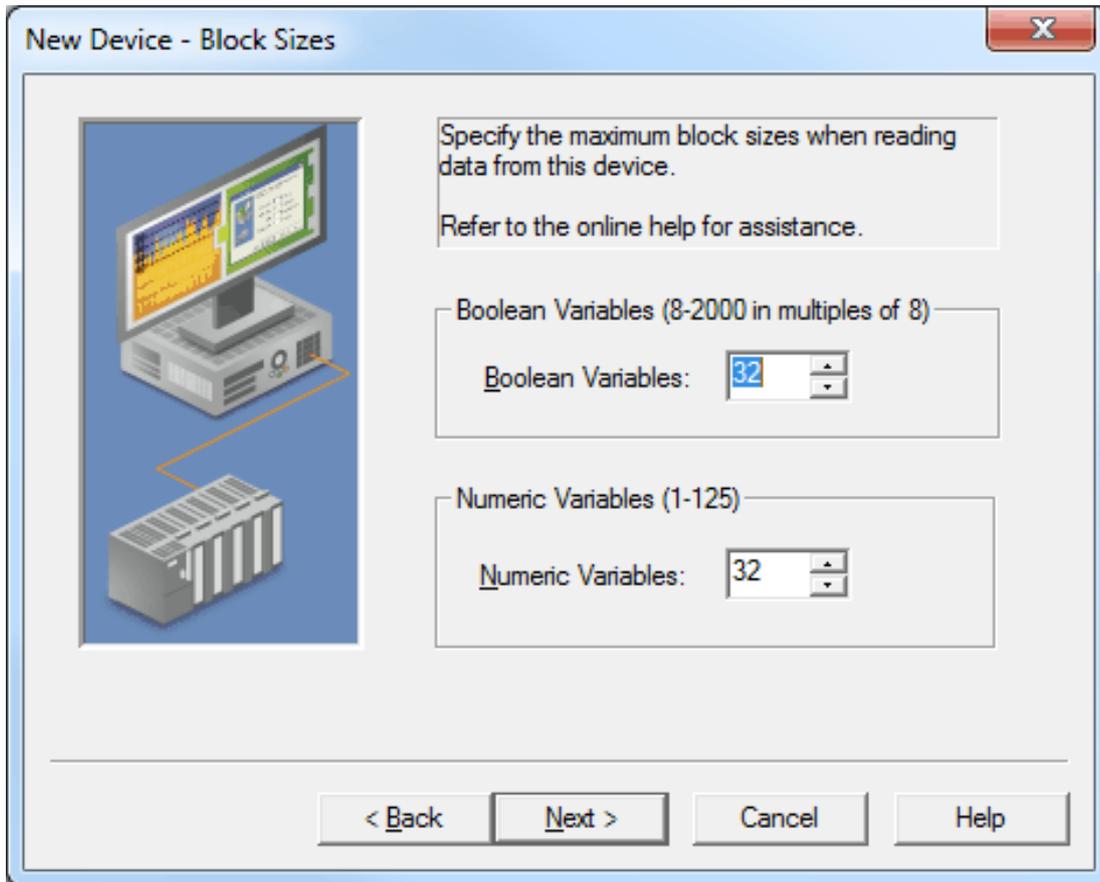


Descriptions of the parameters are as follows:

- **Boolean Variables:** This parameter specifies the output and input coils. Coils can be read from 8 to 2000 points (bits) at a time. A higher block size means more points will be read from the device in a single request. The block size can be reduced in order to read data from non-contiguous locations within the device. The default setting is 32.
- **Numeric Variables:** This parameter specifies the internal and holding registers. Registers can be read from 1 to 125 locations (words) at a time. A higher block size means more register values will be read from the device in a single request. The block size can be reduced in order to read data from non-contiguous locations within the device. The default setting is 32.

## Framing

Because some terminal server devices add additional data to Modbus frames, this dialog may be used to configure the driver to ignore the additional bytes in response messages.
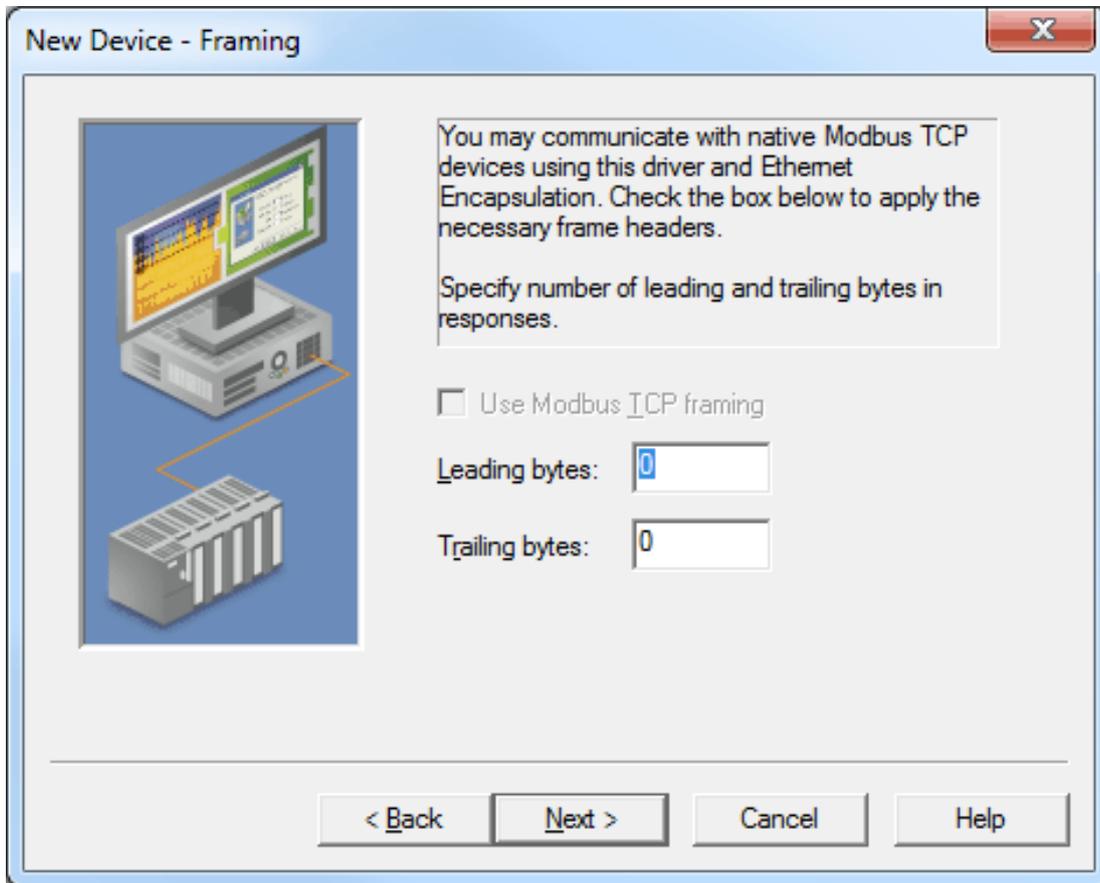


Descriptions of the parameters are as follows:

- **Use Modbus TCP Framing:** When checked, this option communicates with native Modbus TCP devices using Ethernet Encapsulation. For more information, refer to "Using Ethernet Encapsulation" below.
- **Leading bytes:** This parameter specifies the number of bytes that will be attached to the beginning of Modbus responses. The valid range is 0 to 8. The default setting is 0.
- **Trailing bytes:** This parameter specifies the number of bytes that will be attached to the end of Modbus responses. The valid range is 0 to 8. The default setting is 0.

### Using Ethernet Encapsulation

Ethernet Encapsulation must be enabled in order for both Framing and the Use Modbus TCP Framing option to be enabled. For information on enabling Ethernet Encapsulation, refer to the instructions below.

1. To start, right-click on the channel and select **Properties**.

2. Next, open the **Communications** tab. In the **Connection Type** drop-down menu, select **Use Ethernet Encapsulation**. Then, click **Apply** | **Close**.

3. Next, right-click on the device and select **Properties**.

4. Open the **Ethernet Encapsulation** tab.
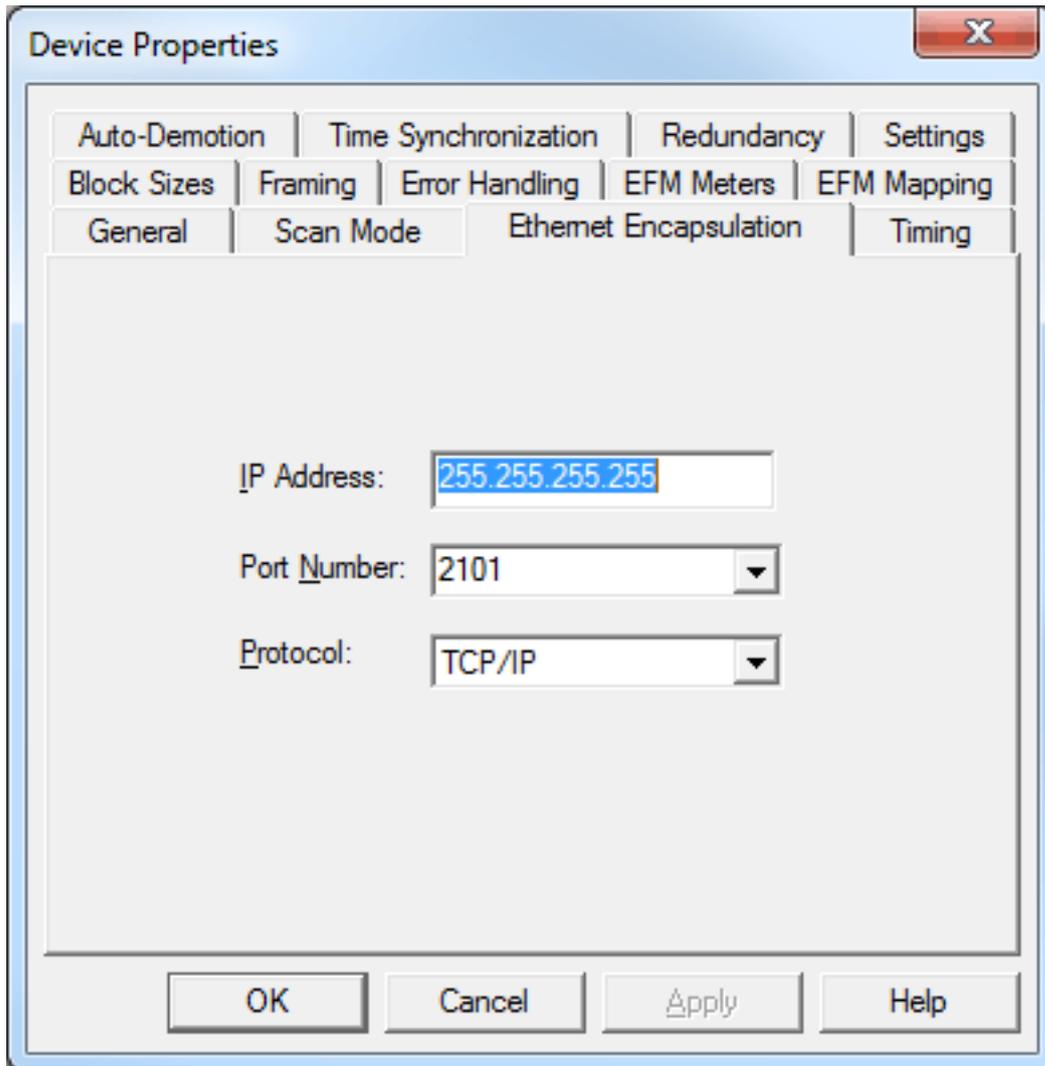
Descriptions of the parameters are as follows:

- **IP Address:** This parameter specifies the device's IP address. The default setting is 255.255.255.255.

- **Port Number:** This parameter specifies the port number. Modbus TCP devices typically use 502.

- **Protocol:** This parameter specifies the protocol. Options include UDP and TCP/IP. The default setting is TCP/IP.

5. Configure the Ethernet Encapsulation settings as desired. Once finished, click **OK**.

## Error Handling

This dialog determines how the Enron Modbus Driver handles errors from the device.
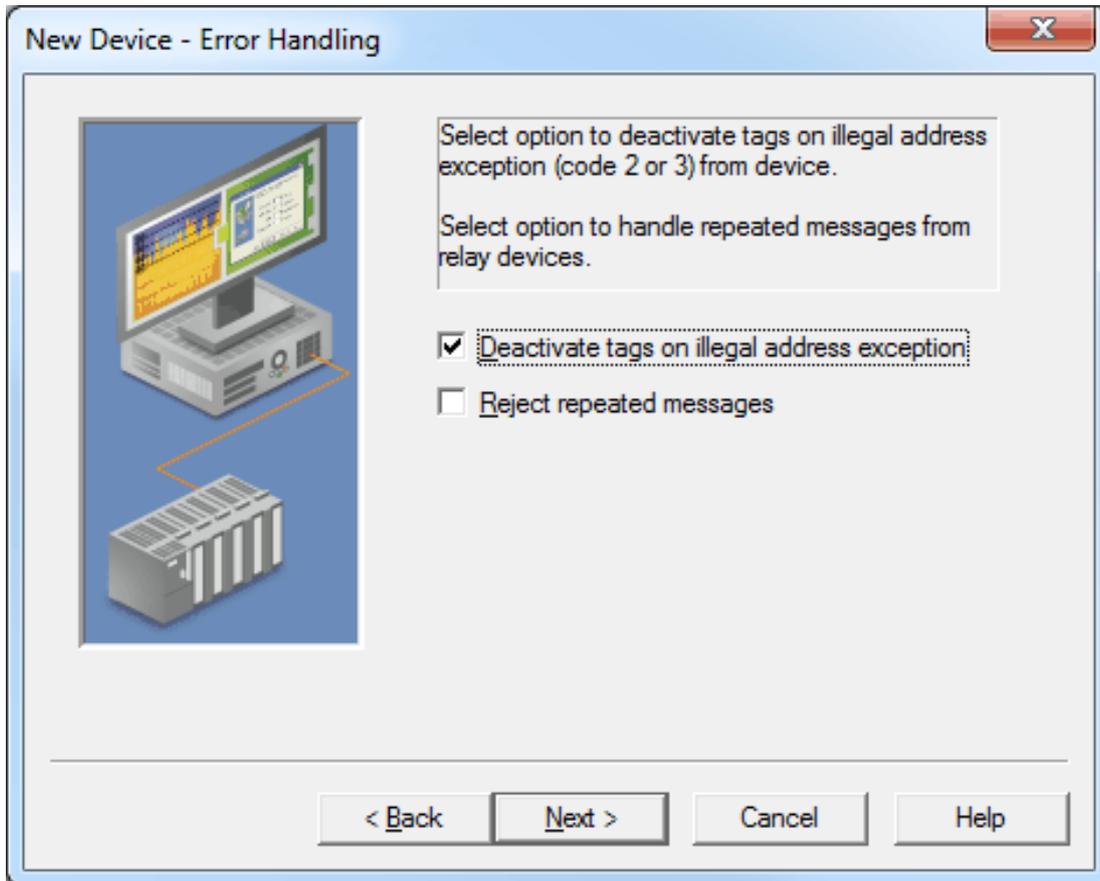


Descriptions of the parameters are as follows:

- **Deactivate tags on illegal address exception:** When checked, this option will cause the driver to stop polling for a block of data if the device returns Modbus exception code 2 (illegal address) or 3 (illegal data, such as number of points) in response to a read of that block. To read addresses that are accessible dynamically in the device, uncheck this option. The default setting is checked.
- **Reject repeated messages:** When checked, the driver will reject repeated messages. When unchecked, the driver will interpret a repeated message as an invalid response and will retry the request. The default setting is unchecked.

**Note:** Some message-relay equipment will echo Modbus requests back to the driver.

## Address Ranges

This dialog specifies the address ranges that are available in the device.
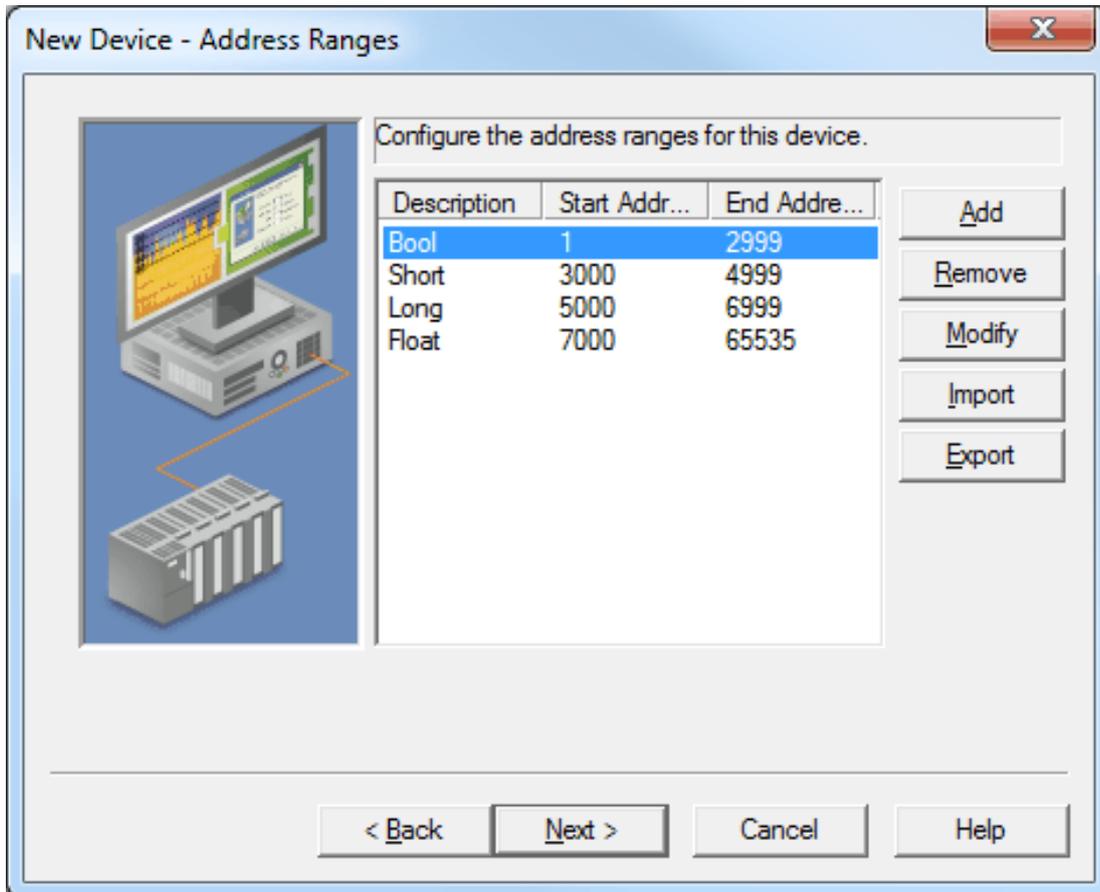


Descriptions of the parameters are as follows:

- **Add:** When clicked, this button launches the Address Range dialog, which is used to define a new address range. For more information, refer to "Address Range" below.
- **Remove:** When clicked, this button removes the selected address range.
- **Modify:** When clicked, this button launches the Address Range dialog, which is used to modify an existing address range. For more information, refer to "Address Range" below.
- **Import:** When clicked, this button launches the Import from CSV dialog, which is used to import address ranges defined in a CSV file.

   **Note:** Imported address ranges can neither overlap nor have the same descriptions as existing address ranges.
- **Export:** When clicked, this button launches the Export to CSV dialog, which is used to export the address ranges to a CSV file (where they can be edited and imported).

**Note:** The Import and Export buttons are available after device creation through the **Device Properties | Address Ranges** tab. For more information, refer to **CSV Import/Export**.

## Address Range

This dialog is used to create a new address range or modify an existing address range.

Descriptions of the parameters are as follows:

- **Description:** This parameter specifies a descriptive name for the address range.
- **Device Data Format:** This parameter specifies the data format of the address range in the device. Valid data types include Boolean, Short, Long, Float, Long(2x16), and Float(2x16). The default setting is Float. For more information, refer to **Data Types Description**.
  **Note:** Float and Long data types read/write to one register; Long(2x16) and Float(2x16) read/write to two registers.
- **Start Address:** This parameter specifies the starting address for the address range. The default setting is 0.
- **End Address:** This parameter specifies the ending address for the address range. The default setting is 0.
- **Base Address:** This parameter specifies an offset that may be applied to the address range to map to registers in the device. This should be used if the device expects the address in read and write requests to be a data address rather than a coil/register number. For example, if the start address is 40001, the end address is 49999, and the device expects a read request for the first register to be address 0, then the base address would be 40001. The default setting is 0.
- **Read:** This parameter specifies the read function code for the address range. The valid values are 1 and 2 for Boolean ranges, and 3 and 4 for all other device data formats. The default setting is 3.
- **Write:** This parameter specifies the write function code for the address range. If the device does not support the single write function code, this parameter may be set to the same value as Multi Write. The valid values are 5 and 15 for Boolean ranges, and 6 and 16 for all other device data formats. The default setting is 6.
- **Multi Write:** This parameter specifies the multi-write function code for the address range. The valid values are 15 for Boolean ranges, and 16 for all other device data formats. The default setting is 16.

## EFM Mapping

EFM data (Configuration, History, Alarms, and Events) in the device must be mapped to the server's EFM attributes. The EFM Mapping dialog is used to configure these mappings, which can shared by all meters, a subset of meters, or uniquely per meter.
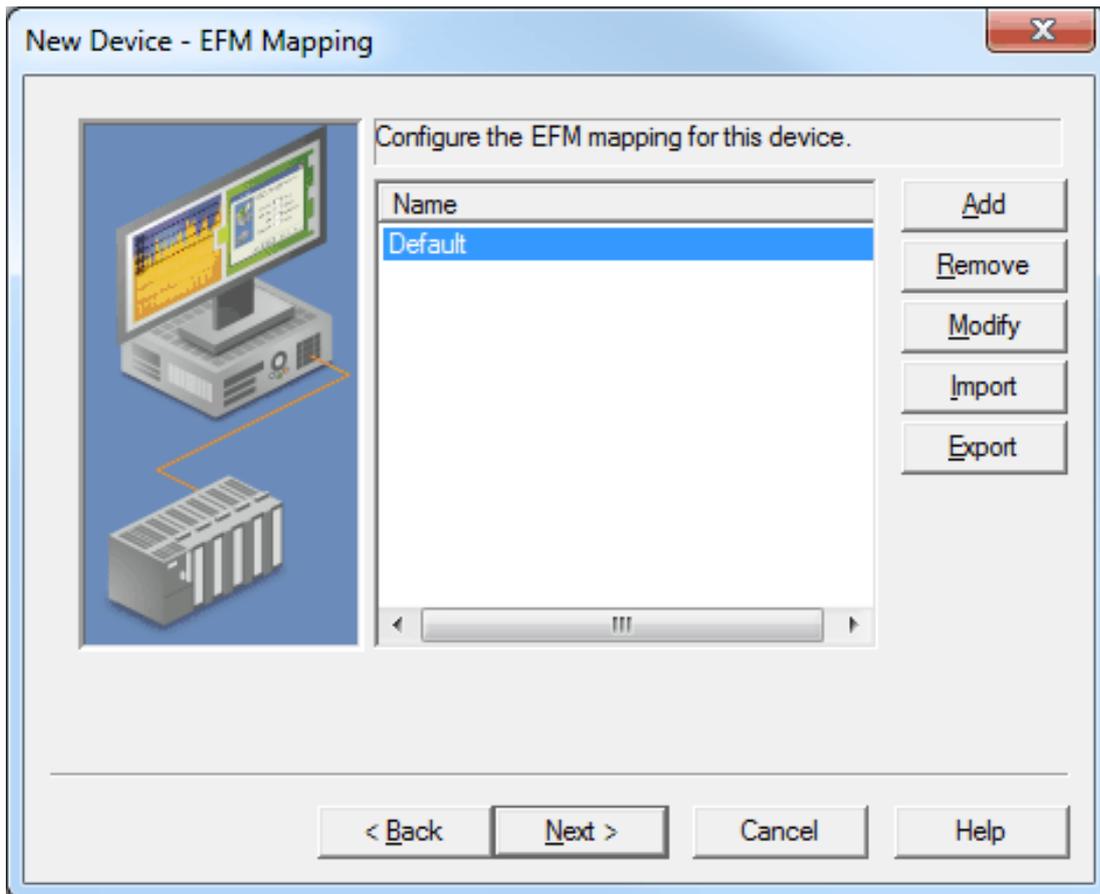
Descriptions of the selections are as follows:

- **Add:** When clicked, this button launches the General dialog, which is the first step in creating a new mapping. For more information, refer to "Creating a New Mapping" below.
- **Remove:** When clicked, this button removes the selected mapping.
- **Modify:** When clicked , this button launches the EFM Mapping dialog, which provides access to the specific mapping's general, Configuration, History, and Alarms record parameters.
- **Import:** When clicked, this button launches the Import from CSV dialog. When a mapping is selected in the list, this option allows users to import a CSV file and replace a mapping configuration.
- **Export:** When clicked, this button launches the Export to CSV dialog. When a mapping is selected in the list, this option exports the mapping configuration to a CSV file, where it can be edited and imported.

**Note:** The Import and Export buttons are available after device creation through the **Device Properties | EFM Mapping** tab. For more information, refer to **CSV Import/Export**.

### Creating a New Mapping
For information on creating a new mapping, refer to the instructions below.

1. To start, right-click on the device and then select **Properties | EFM Mapping**.

2. Next, click **Add**.

3. In **Mapping Name**, specify the new mapping. Mappings are uniquely identified by name.



4. Once finished, click **Next**.

5. Configuration data represents device properties that will be read on every upload. Attributes that are left blank will not be read from the device. For more information on the available attributes and address syntax, refer to **Configuration Mapping**. Once finished, click **Next**.

6. History records read from the device are an array of float data. This mapping allows users to map the data in the array by index to the server's EFM attributes. The mapping starts at Index 0 and extends to Index 60. For more information on the available attributes and index syntax, refer to **History Mapping**. Once finished, click **Next**.
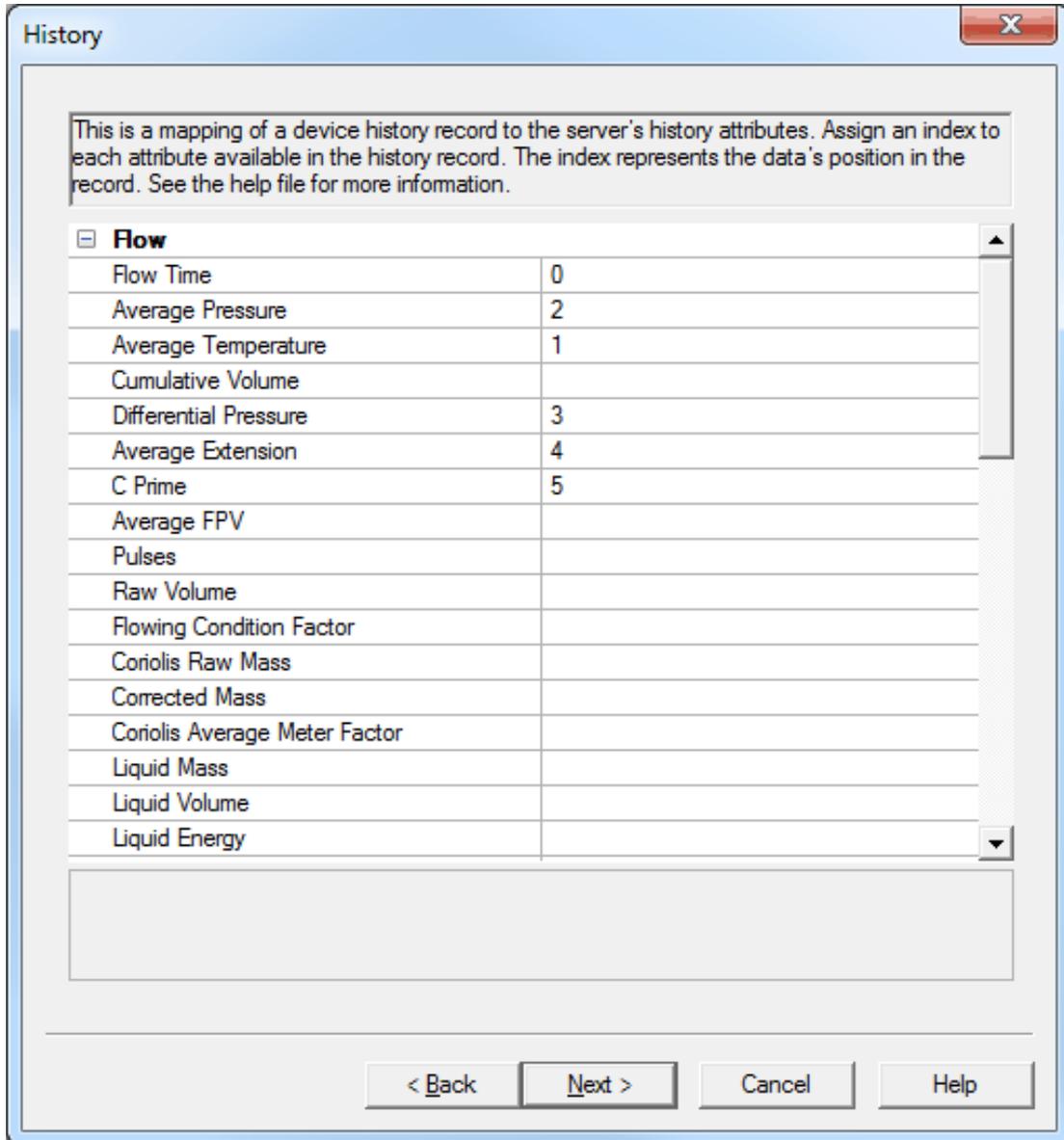


7. Alarms come from a single archive in the device and are mapped by address to the server's alarms. For more information, refer to **Alarm Mapping**.
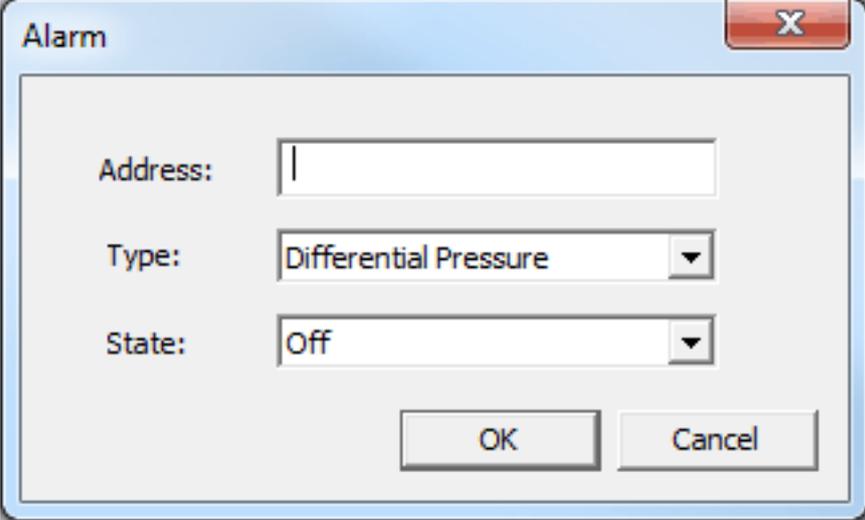
**Note:** To add a new alarm, click **Add**. Then, specify the alarm's parameters and click **OK**.

8.  Once finished, click **Finish**.

## Configuration Mapping

The addresses that are defined in the Configuration Mapping will be read from the device per meter on each EFM poll. Addresses that are left blank or static will not be read from the device.

**Configuration**

This is a mapping of a device configuration to the server's configuration attributes. Assign a register or static value to each attribute available in the configuration record. See the help file for more information.

| ⊟ **Configuration** | |
|---|---|
| Meter ID | B3167[16] |
| Meter Type | |
| Pressure Base | B7149 |
| Temp Base | B7148 |
| Live Analysis | |
| Live BTU | |
| Live Specific Gravity | |
| Live Temp | |
| Calculation Method | B7101(0=0,1=1,5=3,7=4,V=22) |
| Pipe Diameter | B7155 |
| Pipe Reference Temperature | B7156 |
| Meter Tap | |

[ < Back ]  [ Next > ]  [ Cancel ]  [ Help ]

### Configuration Syntax

Dynamic values that are read from the device will use the following syntax: *B1234[LL](E1=x1,E2=x2,...)* where:

- **B:** The base address. The address is the base address used for Meter 1: an offset will be added for each subsequent meter. The offset depends on the data type, and will be defined by the Bool, Short, Long, and Float offsets specified in **EFM Meters**. No offset will be used for the attribute if B is omitted from the address.

  **Note:** The base address syntax allow meters to share a common Configuration Mapping. For example, with a Bool Offset of 100, "B1000" would be "B1000" for Meter 1, "B1100" for Meter 2, and so forth.

- **LL:** The specifier that is used for data that spans more than one register. LL is the number of subsequent registers to span. This is generally only used for string data, and should only be used for Meter ID.

- **(E1=x1,E2=x2,...):** The enumeration mapping. The first value (E1, E2, and so forth) is the server's enumeration, and the second value (x1, x2, and so forth) is the equivalent numeric value in the device.

  **Note:** For example, 4000(O=1,T=2) for Meter Type. If the value at address 4000 is 1, the Meter Type is Orifice. If the value is 2, the Meter Type is Turbine. 'O' and 'T' are defined by the server. For more information on the enumerated types, refer to "Configuration Attributes and Mappings" below.

  **Important:** All configuration addresses must fall within the defined address ranges. Dynamic value

addresses that do not fall within the defined ranges will be skipped when the driver uploads configuration data from the device. For more information, refer to **Address Ranges**.

Static values use the following syntax: *!<static>* where:

- **!:** This character indicates that the subsequent entry is static for the associated attribute.
- **static:** Static can be a string, float, int, or enumeration character depending on the configuration attribute's data type.

**Note:** Static values are not read from the device. If the Configuration Mapping contains all static values, no device communications or polls will be performed when collecting configuration data.

### Configuration Attributes and Mappings

The table below lists all attributes available in the Configuration Mapping, and includes their CSV Name, data type, address syntax, and description.

| Attribute | CSV Name | Data Type | Address Syntax | Description |
|---|---|---|---|---|
| Meter ID | meter_id | String | B1234[10]<br>1234[10]<br>!Meter1 | ID that uniquely identifies the meter. |
| Meter Type | meter_type | Enumeration | B1234(O=1,P=2,...)<br>1234(O=1,P=2,...)<br>!O | Enumeration for the meter type. Server enumerations are as follows:<br><br>O = Orifice<br>P = Positive Displacement<br>T = Turbine<br>U = Ultrasonic<br>L = Liquids<br>V = Vcone<br>C = Coriolis<br>I = Line Pack |
| Pressure Base | pressure_base | Float | B1234<br>1234<br>!1.2 | Pressure base for measured gas volume or gas equivalent volume.** |
| Temp Base | temp_base | Float | B1234<br>1234<br>!1.2 | Temperature base for measured gas volume or gas equivalent volume.* |
| Live Analysis | live_analysis | Bool | B1234<br>1234<br>!1 | I/O connected for live analysis. |
| Live BTU | live_btu | Bool | B1234<br>1234<br>!1 | I/O connected for live heat/energy measurement. |
| Live Gravity | live_gravity | Bool | B1234<br>1234<br>!1 | I/O connected for live gravity measurement. |
| Live Specific Gravity | live_specific_gravity | Bool | B1234<br>1234<br>!1 | I/O connected for live specific gravity measurement. |
| Live Temp | live_temp | Bool | B1234<br>1234<br>!1 | I/O connected for live temperature readings. |
| Calculation Method | calculation_method | Enumeration | B1234(0=2,1=3,...)<br>1234(0=2,1=3,...)<br>!V | The method of flow calculation. Options are as follows:<br><br>0 = AGA3 1985<br>1 = AGA3 1992<br>5= AGA5<br>7 = AGA7<br>V = VCone |

| Attribute | CSV Name | Data Type | Address Syntax | Description |
|---|---|---|---|---|
| | | | | C = AGA11<br>I = Line Pack |
| Pipe Diameter | pipe_diameter | Float | B1234<br>1234<br>!1.2 | Diameter of the pipe.<br><br>Inches for English units, millimeters for metric. |
| Pipe Reference Temperature | pipe_ref_temp | Float | B1234<br>1234<br>!1.2 | Reference temperature of the pipe.* |
| Meter Taps | meter_taps | Enumeration | B1234(F=1,P=2)<br>1234(F=1,P=2)<br>!F | Placement of the meter taps. Options are as follows:<br><br>F = Flange<br>P = Pipe |
| Static Pressure Taps | static_pressure_taps | Enumeration | B1234(U=1,D=2)<br>1234(U=1,D=2)<br>!U | Placement of the static pressure taps. Options are as follows:<br><br>U = Upstream<br>D = Downstream |
| Units | units | Enumeration | B1234(E=1,M=2)<br>1234(E=1,M=2)<br>!E | Default units for most of the properties. More granular options are available for Volume Units and K Factor Units.<br><br>E = English<br>M = Metric<br><br>**Note:** This design assumes a device does not mix English and Metric units. |
| Orifice Plate Size | orifice_plate_size | Float | B1234<br>1234<br>!1.2 | Size of the Orifice plate. This is only used for Orifice meters.<br><br>Inches for English Units and Millimeters for Metric. |
| Orifice Reference Temperature | orifice_ref_temp | Float | B1234<br>1234<br>!1.2 | Reference Temperature for the Orifice plate. Used for Orifice meters only.* |
| DP Low Flow Cutoff | dp_low_flow_cutoff | Float | B1234<br>1234<br>!1.2 | Low flow alarm cutoff.** |
| Atmospheric Pressure | atmospheric_pressure | Float | B1234<br>1234<br>!1.2 | Atmospheric Pressure is in PSI for English and Kilopascal for Metric. |
| BTU | btu | Float | B1234<br>1234<br>!1.2 | Heating value.<br><br>BTU/scfc foot for English and MJ/cubic meter for Metric. |
| Specific Gravity | specific_gravity | Float | B1234<br>1234<br>!1.2 | Specific Gravity of real gas. |
| Viscosity | viscosity | Float | B1234<br>1234<br>!1.2 | lb/(feet * sec) for English and Centipoises for Metric |
| CO2 | co2 | Float | B1234 | % |

| Attribute | CSV Name | Data Type | Address Syntax | Description |
|-----------|----------|-----------|----------------|-------------|
| | | | 1234<br>!90 | |
| N2 | n2 | Float | B1234<br>1234<br>!90 | % |
| C1 | c1 | Float | B1234<br>1234<br>!90 | % |
| C2 | c2 | Float | B1234<br>1234<br>!90 | % |
| C3 | c3 | Float | B1234<br>1234<br>!90 | % |
| ISO C4 | iso_c4 | Float | B1234<br>1234<br>!90 | % |
| NC4 | nc4 | Float | B1234<br>1234<br>!90 | % |
| ISO C5 | iso_c5 | Float | B1234<br>1234<br>!90 | % |
| C5 | c5 | Float | B1234<br>1234<br>!90 | % |
| NEO C5 | neo_c5 | Float | B1234<br>1234<br>!90 | % |
| C6 | c6 | Float | B1234<br>1234<br>!90 | % |
| C7 | c7 | Float | B1234<br>1234<br>!90 | % |
| C8 | c8 | Float | B1234<br>1234<br>!90 | % |
| C9 | c9 | Float | B1234<br>1234<br>!90 | % |
| C10 | c10 | Float | B1234<br>1234<br>!90 | % |
| O2 | o2 | Float | B1234<br>1234<br>!90 | % |
| H2O | h2o | Float | B1234<br>1234<br>!90 | % |
| H2S | h2s | Float | B1234<br>1234<br>!90 | % |
| HE | he | Float | B1234<br>1234<br>!90 | % |
| H2 | h2 | Float | B1234<br>1234<br>!90 | % |
| CO | co | Float | B1234<br>1234 | % |

| Attribute | CSV Name | Data Type | Address Syntax | Description |
|---|---|---|---|---|
| | | | !90 | |
| AR | ar | Float | B1234<br>1234<br>!90 | % |
| Hourly Record Span | hourly_record_span | Enumeration | B1234(L=1,T=2)<br>1234(L=1,T=2)<br>!L | The meaning of the contract hour.<br><br>Leading, or Data Preceding, means a contract hour of 8:00 is from 8:00 to 8:59.<br><br>Trailing, or Data Following, means a contract hour of 8:00 is from 7:01 to 8:00.<br><br>L = Leading<br>T = Trailing |
| Contract Hour | contract_hour | Int | B1234<br>1234<br>!0 | The start of a new billing day. Valid options are 0 to 23. |
| Contract Day | contract_day | Int | B1234<br>1234<br>!0 | The start of a new billing month. Valid options are 0 to 31.<br><br>**Note:** For most fields, this is usually 0. |
| BTU Base | btu_base | Enumeration | B1234(W=1,D=2,...)<br>1234(W=1,D=2,...)<br>!W | Options are as follows:<br><br>W = Wet<br>D = Dry<br>A = As Delivered |
| Factor FA | factor_fa | Bool | B1234<br>1234<br>!1 | Orifice expansion factor due to temperature. |
| Factor FB | factor_fb | Bool | B1234<br>1234<br>!1 | Basic Orifice Factor. True if used. |
| Factor FG | factor_fg | Bool | B1234<br>1234<br>!1 | Specific Gravity Factor SQRT(1/G). True if used. |
| Factor FPV | factor_fpv | Bool | B1234<br>1234<br>!1 | Super compressibility factors. |
| Factor FR | factor_fr | Bool | B1234<br>1234<br>!1 | Reynolds factor. |
| Factor FT | factor_ft | Bool | B1234<br>1234<br>!1 | Temperature factor. |
| Factor FWV | factor_fwv | Bool | B1234<br>1234<br>!1 | Water Vapor Correction factor. |
| Factor FY | factor_fy | Bool | B1234<br>1234<br>!1 | Expansion factor. |
| Fixed Factor | fixed_factor | Float | B1234<br>1234<br>!1.2 | Calibration multiplier. |
| Pipe Material | pipe_material | Enumeration | B1234(S=1,C=2,...) | Possible values are as |

| Attribute | CSV Name | Data Type | Address Syntax | Description |
|---|---|---|---|---|
| | | | 1234(S=1,C=2,...) !S | follows:<br><br>S = Stainless Steel<br>C = Carbon Steel<br>M = Monel |
| Plate Material | plate_material | Enumeration | B1234(S=1,C=2,...) 1234(S=1,C=2,...) !S | Possible values are as follows:<br><br>S = Stainless Steel<br>C = Carbon Steel<br>M = Monel |
| FPV Method | fpv_method | Enumeration | B1234(N=1,A=2,...) 1234(N=1,A=2,...) !N | Super compressibility method. Possible values are as follows:<br><br>N = NX19<br>A = AGA8 Detail<br>1 = AGA8 Gross 1<br>2 = AGA8 Gross 2 |
| Static Pressure Type | static_pressure_ type | Enumeration | B1234(G=1,A=2) 1234(G=1,A=2) !G | Possible values are as follows:<br><br>G = Gauge<br>A = Absolute |
| DP Calibration Range High | dp_calib_range_ high | Float | B1234 1234 !1.2 | High calibration range for dynamic pressure.** |
| DP Transducer Range High | dp_transd_range_ high | Float | B1234 1234 !1.2 | High transducer range for dynamic pressure.** |
| SP Calibration High | sp_calib_high | Float | B1234 1234 !1.2 | High calibration range for static pressure.** |
| SP Calibration Low | sp_calib_low | Float | B1234 1234 !1.2 | Low calibration range for static pressure.** |
| SP Transducer Range High | sp_transd_range_ high | Float | B1234 1234 !1.2 | High transducer range for static pressure.** |
| Temp Calibration Range High | temp_calib_range_ high | Float | B1234 1234 !1.2 | High calibration range for temperature.* |
| Temp Calibration Range Low | temp_calib_range_ low | Float | B1234 1234 !1.2 | Low calibration range for temperature.* |
| Temp Transducer Range High | temp_transd_ range_high | Float | B1234 1234 !1.2 | High transducer range for temperature.* |
| Temp Transducer Range Low | temp_transd_ range_low | Float | B1234 1234 !1.2 | Low transducer range for temperature.* |
| DP Low Alarm | dp_low_alarm | Float | B1234 1234 !1.2 | Dynamic pressure low alarm.** |
| DP Backflow Alarm | dp_backflow_alarm | Float | B1234 1234 !1.2 | Dynamic pressure backflow alarm.** |
| DP High Alarm | dp_high_alarm | Float | B1234 1234 !1.2 | Dynamic pressure high alarm.** |
| SP Low Alarm | sp_low_alarm | Float | B1234 | Static pressure low |

| Attribute | CSV Name | Data Type | Address Syntax | Description |
|-----------|----------|-----------|----------------|-------------|
| | | | 1234<br>!1.2 | alarm.** |
| SP High Alarm | sp_high_alarm | Float | B1234<br>1234<br>!1.2 | Static pressure high alarm.** |
| Temp Low Alarm | temp_low_alarm | Float | B1234<br>1234<br>!1.2 | Temperature low alarm.* |
| Temp High Alarm | temp_high_alarm | Float | B1234<br>1234<br>!1.2 | Temperature high alarm.* |
| K Factor | k_factor | Float | | Units are controlled by K Factor Units Setting. |
| Meter Factor | meter_factor | Float | B1234<br>1234<br>!1.2 | The actual volume of gas passing through the meter over the meter indicated volume. |
| Accumulated Volume | accumulated_ volume | Float | B1234<br>1234<br>!1.2 | Units are controlled by Volume Units setting. |
| Volume Units | unit_volume | Enumeration | B1234(0=1,1=2,...)<br>1234(0=1,1=2,...)<br>!0 | 0= Million Cubic Meters (CM)<br>1 = Thousand CM<br>2= Hundred CM<br>3 = CM<br>4 = Million Cubic Feet (CF)<br>5 = Thousand CF<br>6 = Hundred CF<br>7 = CF |
| K Factor Units | unit_k_factor | Enumeration | B1234(0=1,1=2,...)<br>1234(0=1,1=2,...)<br>!0 | 0 = Million Cubic Meters (CM)<br>1 = Thousand CM<br>2= Hundred CM<br>3 = CM<br>4 = Million Cubic Feet (CF)<br>5 = Thousand CF<br>6 = Hundred CF<br>7 = CF |
| Ratio of Specific Heats | specific_heats | Float | B1234<br>1234<br>!1.2 | The heat capacity ratio, adiabatic index, or ratio of specific heats. |

*Units of Fahrenheit for English and Celsius for Metric.
**Units of Inches of water for English and Kilopascal for Metric.

## History Mapping

History data pulled from the device is in record form, with each record containing an array of four byte floats. Each float has a unique index or position in the array. The valid range is 0 to *N*, where *N* is the maximum number of floats in the record.

The History dialog is used to assign each float to an EFM attribute using the float's unique index. Records that are retrieved from the device are parsed using this mapping, which applies to both Hourly and Daily History data and can include data from the Gas Chromatograph (GC) archives. When data is not included from the GC archives, they will not be uploaded.

**Note:** The mapping assumes that the first float in the record is Date, and that the second float in the record is Time. As a result, Index 0 actually refers to the third value in the record. The Time format is specified in the "History Record Time Stamp Format" parameter. For more information, refer to **EFM Meters**.
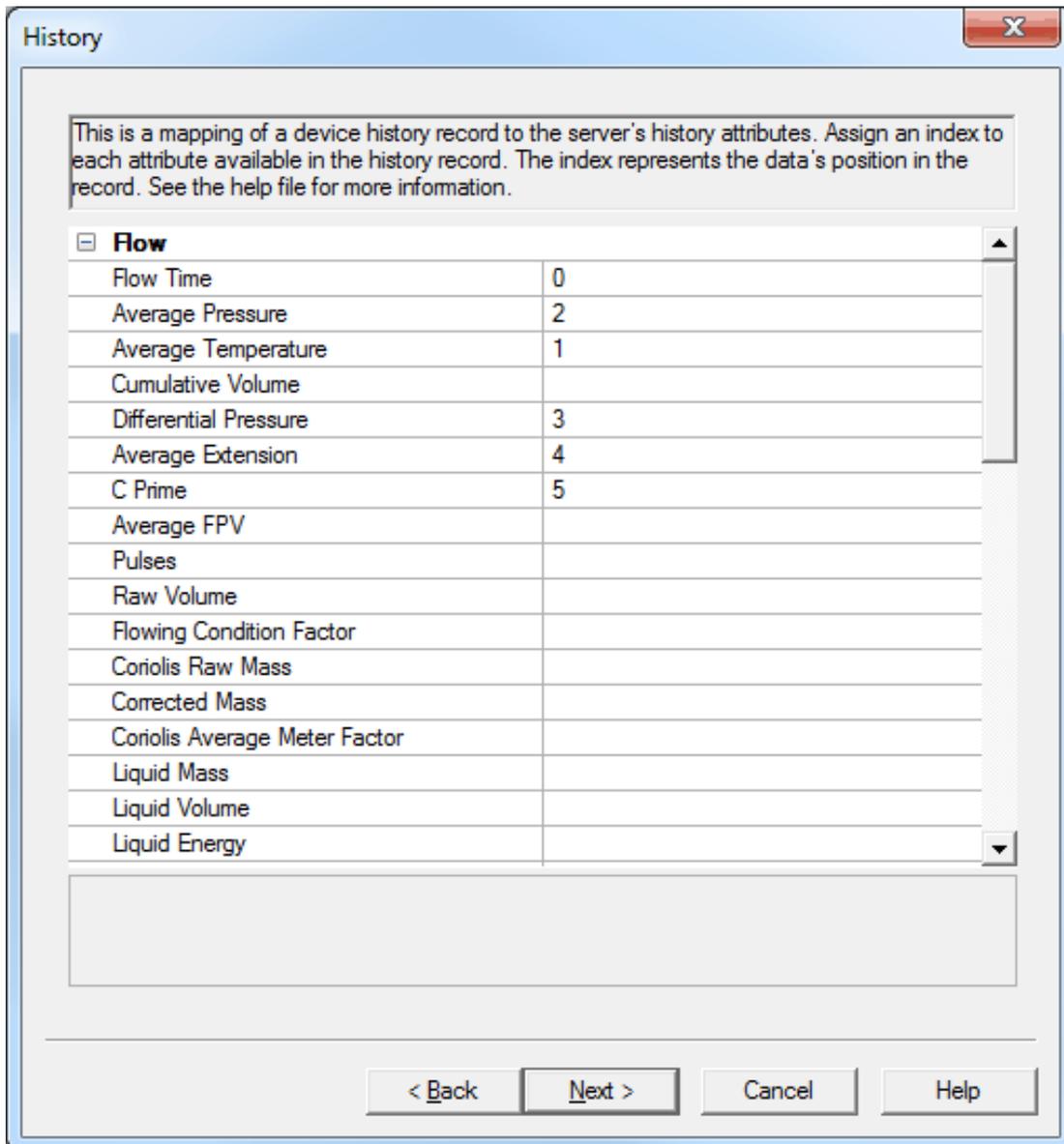
**History Syntax**

A History Index uses the following syntax: *QN,BO=0/1,WO=0/1* where:

- **Q:** This optional index indicates that the element comes from the Gas Chromatograph archive.
- **N:** This index in the record associates with an attribute. The valid range is 0 to 60.
- **BO=0/1:** This optional syntax allows the value's Byte Order to differ from the "History Archive Modbus Byte Order" setting located in EFM Meters. "BO=0" means Modbus Byte Order with Big Endian (or the most significant bit) first. "BO=1" means Little Endian (or the least significant bit) first.
- **WO=0/1:** This optional syntax allows the value's Word Order to differ from the "History Archive First Word Low" setting located in EFM Meters. "WO=0" means the first word is low. "WO=1" means the first word is high.

Static values use the following syntax: *!<static>* where:

- **!:** This character indicates that the subsequent entry is static for the associated attribute.
- **static:** Static values are always considered floats.

**See Also: EFM Meters**

**History Attributes and Mappings**

The table below lists all the attributes available in the History Mapping, and includes their CSV name, data type, and description. Attributes that are left blank will be ignored.

**Note:** The index syntax (*QN,BO=0/1,WO=0/1*) is available for all attributes.

**Flow**

| Attribute | CSV Name | Data Type | Description |
|---|---|---|---|
| Flow Time | flow_time | Float | Flow time for this record in minutes. |
| Average Pressure | avg_pressure | Float | Average pressure.* |
| Average Temperature | avg_temp | Float | Average temperature over the flow time.<br><br>Fahrenheit for English and Celsius for Metric. |
| Cumulative Volume | cumulative_volume | Float | Volume added during this interval for orifice and turbine meters.<br><br>Units are controlled by the Volume Units in the Configuration Mapping. |
| Differential Pressure | diff_pressure | Float | Average differential pressure for orifice meters.* |
| Average Extension | avg_extension | Float | Average extension for orifice meters.* |
| C Prime | c_prime | Float | Orifice flow constant. |
| Average FPV | avg_fpv | Float | Average Super Compressibility Factor. |
| Pulses | pulses | Float | Pulses for turbine meters. |
| Raw Volume | raw_volume | Float | Raw volume for turbine meters.<br><br>Units are controlled by the Volume Units in the Configuration Mapping. |
| Flowing Condition Factor | flowing_condition_factor | Float | Flowing Condition Factor for turbine meters. |
| Coriolis Raw Mass | coriolis_raw_mass | Float | Raw mass for coriolis meters.<br><br>Units are pounds for English and KG for Raw Mass. |
| Corrected Mass | corrected_mass | Float | Corrected mass for coriolis meters.<br><br>Units are pounds for English and KG for Metric. |
| Coriolis Average Meter Factor | coriolis_avg_meter_factor | Float | Average meter factor for coriolis meters. |
| Liquid Mass | liquid_mass | Float | Mass for liquid meters.<br><br>Units are pounds for English and KG for Metric. |
| Liquid Volume | liquid_volume | Float | Volume for liquid meters.<br><br>Units are controlled by the Volume Units in the Configuration Mapping. |
| Liquid Energy | liquid_energy | Float | Energy for liquid meters.<br><br>Units are BTU/cubic foot for English and MJ/cubic meter for Metric. |
| Total Volume | total_volume | Float | Total volume. |

| Attribute | CSV Name | Data Type | Description |
|---|---|---|---|
| | | | Units are controlled by the Volume Units in the Configuration Mapping. |
| Total Energy | total_energy | Float | Total energy.<br><br>Units are BTU/cubic foot for English and MJ/cubic meter for Metric. |

*Inches of Water for English and Kilopascals for Metric.

**Gas Composition (Mole%)**

| Attribute | CSV Name | Data Type | Description |
|---|---|---|---|
| Average BTU | avg_btu | Float | Average heating value.<br><br>Units are Dekatherms for English and Gigajoules for Metric. |
| Average Specific Gravity | avg_specific_gravity | Float | Average specific gravity. |
| Average CO2 | avg_co2 | Float | % |
| Average N2 | avg_n2 | Float | % |
| Average C1 | avg_c1 | Float | % |
| Average C2 | avg_c2 | Float | % |
| Average C3 | avg_c3 | Float | % |
| Average ISOC4 | avg_isoc4 | Float | % |
| Average NC4 | avg_nc4 | Float | % |
| Average ISOC5 | avg_isoc5 | Float | % |
| Average NC5 | avg_nc5 | Float | % |
| Average NEOC5 | avg_neoc5 | Float | % |
| Average C6 | avg_c6 | Float | % |
| Average C7 | avg_c7 | Float | % |
| Average C8 | avg_c8 | Float | % |
| Average C9 | avg_c9 | Float | % |
| Average C10 | avg_c10 | Float | % |
| Average O2 | avg_o2 | Float | % |
| Average H2O | avg_h2o | Float | % |
| Average H2S | avg_h2s | Float | % |
| Average HE | avg_he | Float | % |
| Average H2 | avg_h2 | Float | % |
| Average CO | avg_co | Float | % |
| Average AR | avg_ar | Float | % |
| Specific Heat Ratio | specific_heat_ratio | Float | Ratio of specific heat. |
| Viscosity | viscosity | Float | Viscosity.<br><br>Units of Pounds/Mass per Foot/Second for English and Centipoises for Metric. |

## Alarm Mapping

The Alarms dialog is used to assign alarms received from the device to specific meters, alarm types, and states. Alarms can apply to one or more meters depending on how the alarm address is specified.

**Note:** When an alarm is received from a device that does not match an address in any of the Alarm Mappings being used, the server will check if the alarm matches an address in the Event Mappings. If it does, it will be handled by the Event Mapping. If it does not, the alarm will be logged as a user string event.

**See Also: EFM Mapping**

**Alarms**

This is a mapping of device alarms to server alarm types and state. Assign per-meter alarms to a base address, and flow computer alarms to a static address. See the help file for more information.

| Address | Type | State |
|---------|------|-------|
| B3106 | Temperature | Lo |
| B3107 | Temperature | Hi |
| B3108 | Static Pressure | Lo |
| B3109 | Static Pressure | Hi |
| B3110 | Differential Pressure | Lo |
| B3111 | Differential Pressure | Hi |

Add      Delete      Edit

< Back      Finish      Cancel      Help

Descriptions of the parameters are as follows:

- **Address:** This parameter specifies the Enron address that generates the alarm. The default setting is blank.
- **Type:** This parameter specifies the type of alarm. The default setting is Differential Pressure. Options are as follows:

  - Differential Pressure
  - Static Pressure
  - Temperature
  - Cutoff
  - Backflow
  - Battery
- **State:** This parameter specifies the alarm state. The default setting is Off. Options are as follows:

  - Off
  - On
  - Hi
  - Lo

- **Add:** When clicked, this button launches a new Alarm dialog. For more information, refer to "Adding a New Alarm" below.
- **Delete:** When clicked, this button deletes the selected alarm from the mapping.
- **Edit:** When clicked, this button launches the Alarm dialog that contains the selected alarm's properties.

## Alarm Syntax

An alarm address may use one of the following syntactic forms:

- **B1234:** This is a base address, and makes the alarm meter-specific. The offset depends on the address data type and the Bool, Short, Long, and Float Offsets specified in **EFM Meters**.

  **Note:** For example, an address is "B1234" and the data type is Bool. If the Bool Offset is 10, an alarm received from device address "1234" will be assigned to Meter 1. An alarm received from device address "1244" will be assigned to Meter 2.
- **1234:** This is a static address, and makes the alarm non-meter specific. An alarm that is received from address 1234 will be sent to all meters that use the Alarm Mapping.

**Important:** All alarm addresses must fall within the defined address ranges. Dynamic value addresses that do not fall within the defined ranges will be skipped when the driver uploads alarm data from the device. For more information, refer to **Address Ranges**.

## Adding a New Alarm

1. To start, click **Add**.

2. Next, specify the new alarm's Address, Type, and State.
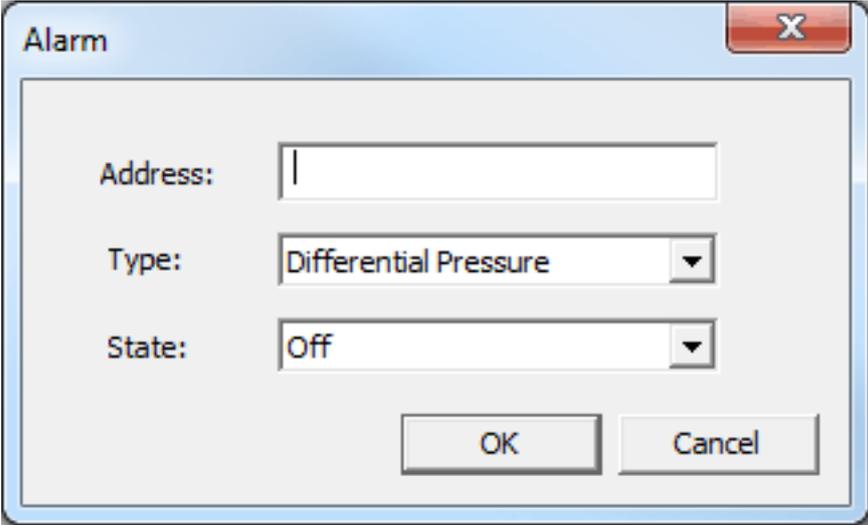


3. Once finished, click **OK**.

## Event Mapping

Event Mappings are not user-configurable: they depend on the Configuration Mapping.

When an event is received from an address that matches an address in any Configuration Mapping, it will be converted to an audit event. For example, if a Configuration Mapping with address "B7500" is set to BTU and Meter 1 is using the mapping, an event will be generated from address "7500" when the user changes the BTU in the device. The event will be converted to an audit event for the BTU field on Meter 1. Both the old and new values will be displayed.

When an event is received from an address that does not match an address in any Configuration Mapping, it will be considered a non-meter event. The event will be converted to a string event, and then handled as defined in the "Non Meter Events" setting located in the EFM Meters dialog.

## EFM Meters

This dialog contains meter-specific configuration options, and includes a grid control with the following device and per-meter settings. Up to twelve meters are supported.

**Archive Configuration**
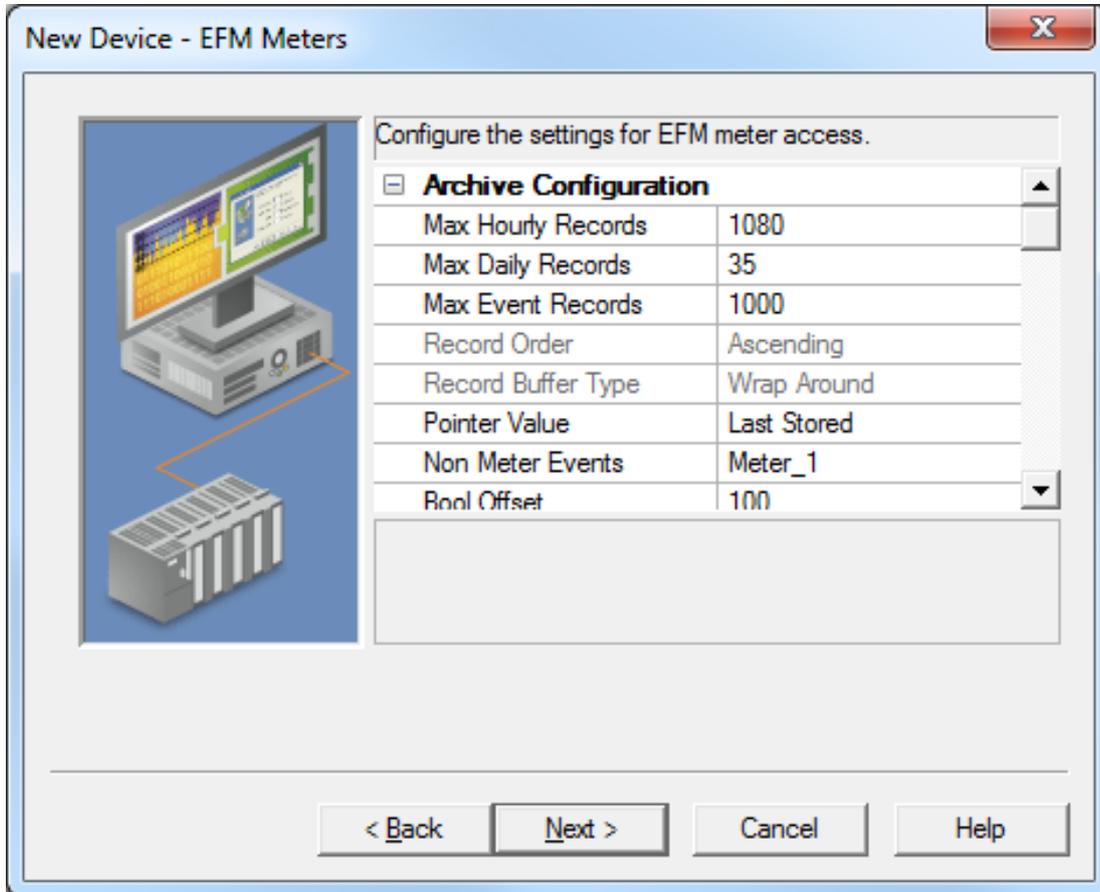


Descriptions of the parameters are as follows:

- **Max Hourly Records:** This parameter specifies the maximum number of hourly records that the device will store. The valid range is 1 to 65535. The default setting is 1080.
- **Max Daily Records:** This parameter specifies the maximum number of daily records that the device will store. The valid range is 1 to 65535. The default setting is 35.
- **Max Event Records:** This parameter specifies the maximum number of event records that the device will store. The valid range is 1 to 65535. The default setting is 1000.
- **Record Order:** This parameter specifies the order that the records will be stored in the device. The default setting is Ascending. This setting is currently not supported.
- **Record Buffer Type:** This parameter specifies the type of buffer that the device will use to store the records in the device. The default setting is Wrap Around. This setting is currently not supported.
- **Pointer Value:** This parameter specifies whether the value in the hourly and daily pointer registers will be the location of the last stored record or the next record that is available. The default setting is Last Stored.
- **Non Meter Events:** This parameter specifies the meter number for non-meter events. Utilize the drop-down menu to ignore non-meter events, log non-meter events to a specific meter, or log non-meter events to all meters. The default setting is Meter_1. For more information, refer to **Event Mapping**.
- **Bool Offset:** This parameter specifies the offset of the Bool configuration register range for each successive meter. The default setting is 100. For more information, refer to **Configuration Mapping**.
- **Short Offset:** This parameter specifies the offset of the Short configuration register range for each successive meter. The default setting is 100. For more information, refer to **Configuration Mapping**.
- **Long Offset:** This parameter specifies the offset of the Long configuration register range for each successive meter. The default setting is 100. For more information, refer to **Configuration Mapping**.

- **Float Offset:** This parameter specifies the offset of the Float configuration register range for each successive meter. The default setting is 250. For more information, refer to **Configuration Mapping**.
- **History Archive Modbus Byte Order:** This parameter specifies whether the data in History archives will be in standard Modbus byte order. The default setting is Yes.

    **Note:** This parameter can be overridden per record element. For more information, refer to **History Mapping**.
- **History Archive First Word Low:** This parameter specifies whether the data in History archives will be stored with the first word low. The default setting is No.

    **Note:** This parameter can be overridden per record element. For more information, refer to **History Mapping**.
- **Event Archive Modbus Byte Order:** This parameter specifies whether data in Alarm/Event archives will be in standard Modbus byte order. The default setting is Yes.
- **Event Archive First Word Low:** This parameter specifies whether the data in the Alarm/Event archives will be stored with the first word low. The default setting is No.
- **Multi-Byte Characters:** This parameter specifies whether string configuration data will be stored in two bytes per character format. The default setting is Yes.
- **History Record Time Stamp Format:** Enron Modbus time stamps for history records are floats. This parameter specifies whether the time stamps will be in HHMMSS or HHMM.SS (with the seconds after the decimal) format. The default setting is HHMMSS.

    **Note:** The driver assumes that the first and second values in a history record are Date and Time, respectively.
- **Clear Cache:** This parameter specifies whether to clear the device's EFM cache, which is maintained by the server and stores history, alarms, and events data for each meter. When enabled, the cache will be cleared on the next poll. This feature will also remove pointer files, which are used to track EFM uploads in order to prevent uploading the same records twice. All EFM data in the device will be requested again on the next poll. Once the cache is cleared, this parameter will automatically be set back to No. The default setting is No.

    **Note:** This option should be used during testing, if the EFM mappings are not configured correctly, or in situations where it is beneficial to re-request all EFM data from the device.

**Meter *N***
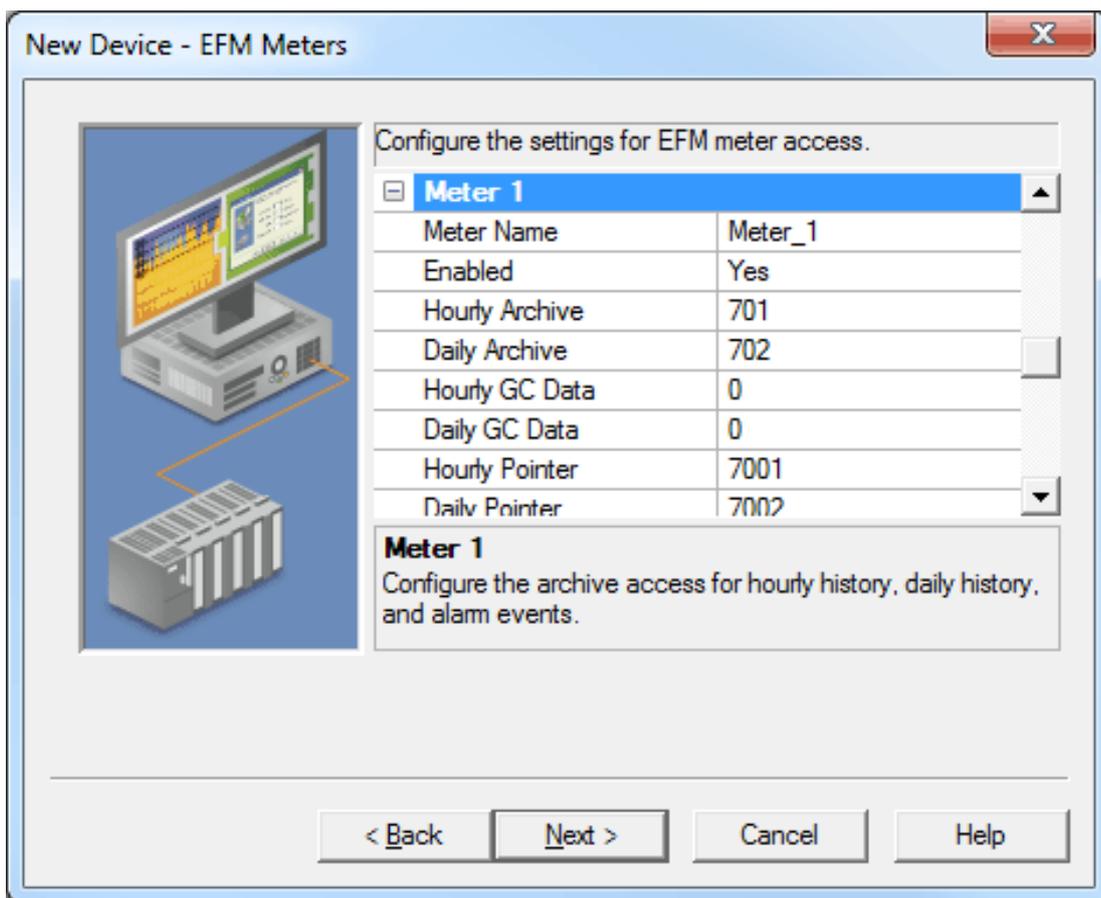
Descriptions of the parameters are as follows:

- **Meter Name:** This parameter specifies a descriptive name for the meter. The valid range is 1 to 128 characters. It can neither begin nor end with a trailing blank space. It also cannot begin with an underscore or contain an '@' character, period, or quotation mark.

  **Note:** Multiple meters may not use the same meter name.
- **Enabled:** This parameter specifies whether the meter is enabled. The default setting for Meter 1 is Yes. The default setting for meters 2 through 12 is No.
- **Hourly Archive:** This parameter specifies the meter/run's hourly archive address. The default setting is 701.
- **Daily Archive:** This parameter specifies the meter/run's daily archive address. The default setting is 702.
- **Hourly GC Data:** This parameter specifies the meter/run's hourly archive address for gas chromatography. The default setting is 0.

  **Note:** When set to 0, GC data will not be requested from the device for hourly archives.
- **Daily GC Data:** This parameter specifies the meter/run's daily archive address for gas chromatography. The default setting is 0.

  **Note:** When set to 0, GC data will not be requested from the device for daily archives.
- **Hourly Pointer:** This parameter specifies the register that the device will use to indicate the position of the current (or last) hourly record in the buffer. The default setting is 7001.
- **Daily Pointer:** This parameter specifies the register that the device will use to indicate the position of the current (or last) daily record in the buffer. The default setting is 7002.
- **Event Counter:** This parameter specifies the register that the device will use to indicate the number of alarm/event records in the buffer. The default setting is 7000.
- **Mapping:** This parameter specifies the type of mapping that will be used for Configuration, History, and Alarm data. The default setting is Default. For information on creating a new mapping, refer to **Creating a New Mapping**.

**Important:** The Hourly Pointer, Daily Pointer, and Event Counter addresses must fall within the defined address ranges. For more information, refer to **Address Ranges**.

## CSV Import/Export

The EFM Mappings support the import and export of data in a Comma Separated Variable (CSV) file. CSV import and export supports the efficient configuration of many devices. For more information on a specific aspect of CSV Import/Export, select a link from the list below.

**Creating a Template**
**Exporting EFM Mappings**
**Importing EFM Mappings**
**Using Other Characters as the Delimiter**

### Creating a Template
The easiest way to create an import CSV file is to create a template. For more information, refer to the instructions below.

1.  To start, create a new device using the default settings. Then, click **OK**.

2.  Next, right-click on the device and select **Properties** | **EFM Mapping**.

3.  Select the **Default** mapping and then click **Export**.

4.  Save the file to an accessible location.

5.  Use the exported template in a spreadsheet application that supports CSV files, and then modify the file as desired.

**Note:** Microsoft Excel is an excellent tool for editing large groups of tags outside the server. Once a template CSV file has been exported, it can be loaded directly into Excel for editing.

### Exporting EFM Mappings
Exporting an EFM Mapping will generate a CSV text file that contains sections for Configuration, History, and Alarms. Each section has a heading record followed by a record for each item. Column names must match those listed; however, columns may be in any order.

#### Configuration Mapping
The required columns are listed in **bold**.

| Column Name | Description |
| --- | --- |
| **Attribute** | This is the name of the Configuration Mapping attribute. Attributes can be in any order. Attributes that are not included in an import will be left blank in the mapping.<br><br>**Note:** All possible attribute names are listed in the Configuration dialog. For more information, refer to **Configuration Mapping**. |
| Value | This is the address syntax for the attribute. It can be blank, static, or take the form *B1234[LL](E1=1,E2=2)*.<br><br>**Note:** For information on each attribute's value limitations, refer to **Configuration Mapping**. |

#### History Mapping
The required columns are listed in **bold**.

| Column Name | Description |
| --- | --- |
| **Attribute** | This is the name of the History Mapping attribute. Attributes can be in any order. Attributes that are not included on an import will be left blank in the mapping.<br><br>**Note:** All possible attribute names are listed in the History dialog. For more information, refer to **History Mapping**. |
| Value | This is the address syntax for the attribute. It can be blank, static, or take the form *QN,BO=0/1,WO=0/1*.<br><br>**Note:** For information on each attribute's value limitations, refer to **History Mapping**. |

**Alarm Mapping**
The required columns are listed in **bold**.

| Column Name | Description |
|---|---|
| **Address** | This is the address of the alarm. It can take the form *B1234.* For more information, refer to **Alarm Mapping**. |
| Alarm Type | This is the type of the alarm. The default setting is Differential Pressure. The valid types are as follows:<br><br>1 = Differential Pressure<br>2 = Static Pressure<br>3 = Temperature<br>4 = Cutoff<br>5 = Backflow<br>6 = Battery |
| Alarm State | This is the state of the alarm. The default setting is Off.<br><br>1 = Off<br>2 = On<br>3 = Hi<br>4 = Lo |

**Importing EFM Mappings**
Once the CSV file has been created and exported, it may be re-imported into an EFM Mapping. To do so, open **EFM Mapping** and then click **Import**.

**Note:** For Configuration, History, and Alarms, importing will replace all existing settings with the settings specified in the CSV file. When the import is complete, the configured mapping should match one for one with the file.

**Using Other Characters as the Delimiter**
When utilizing a CSV file that does not use a comma or semi-colon delimiter, users should do one of the following:

- Save the project in XML. Then, perform mass configuration on the XML file instead of using CSV.
- Perform a search-and-replace on the delimiter in the CSV file and then replace the delimiter with a comma or semicolon. The delimiter being used by the OPC server (either comma or semicolon) must be set to the replacement character.

**Note:** For information on specifying which character to use as the variable (comma or semicolon), refer to "Options - General" in the server help file.

## EFM Cache

The Enron Modbus Driver caches EFM data per device. During polls, the driver will only request new data from the device and then add it to its local cache. This minimizes communication between the physical device and the driver. The cache that is maintained by the driver will be cleared under the following scenarios:

1. The server is reinitialized, restarted, or a new project is loaded.

2. The channel or device is deleted.

3. The cache is cleared manually through the "Clear Cache" setting located in EFM Meters.

4. A meter's Hourly or Daily GC or History archives change.

**See Also: EFM Meters**

## Data Types Description

| Data Type | Description |
|-----------|-------------|
| Boolean | Single bit |
| Word | Unsigned 16-bit value<br><br>bit 0 is the low bit<br>bit 15 is the high bit |
| Short | Signed 16-bit value<br><br>bit 0 is the low bit<br>bit 14 is the high bit<br>bit 15 is the sign bit |
| DWord | Unsigned 32-bit value<br><br>bit 0 is the low bit<br>bit 31 is the high bit |
| Long /<br>Long(2x16) | Signed 32-bit value<br><br>bit 0 is the low bit<br>bit 30 is the high bit<br>bit 31 is the sign bit<br>*See Note 2* |
| BCD | Two byte packed BCD<br><br>Value range is 0-9999. Behavior is undefined for values beyond this range. |
| LBCD | Four byte packed BCD<br><br>Value range is 0-99999999. Behavior is undefined for values beyond this range. |
| Float /<br>Float(2x16) | 32-bit floating point value<br><br>The driver interprets two consecutive registers as a single precision value by making the last register the high word and the first register the low word<br>*See Note 2* |
| Float Example | If register 40001 is specified as a float, bit 0 of register 40001 would be bit 0 of the 32-bit data type, and bit 15 of register 40002 would be bit 31 of the 32-bit data type. |

**Notes:**

1. The descriptions assume the default first word low data handling of 32-bit data types.

2. Float and Long data types can read/write to one register or two registers based on how they are specified by **Address Range** when setting up the device.

## Address Descriptions

The Enron Modbus Driver supports the default address ranges listed in the table below. The default data types are shown in **bold**.

**See Also: Address Ranges**

| Address | Default Range | Data Type | Access |
|---|---|---|---|
| Boolean Variables | 0001-2999 | **Boolean** | Read/Write |
| 16-bit Short Variables | 3001-4999<br><br>3xxx.0/1-4xxx.15/16* | **Short**, Word, BCD<br><br>Boolean | Read/Write |
| 32-bit Long Variables | 5001-6999<br><br>5xxx.0/1-6xxx.31/32* | **Long**, DWord, LBCD, Float<br><br>Boolean | Read/Write |
| 32-bit Float Variables | 7000-65535<br><br>7xxx.0/1-6xxxx.31/32* | **Float**, Long, DWord, LBCD<br><br>Boolean | Read/Write |

**Notes:**

1. 32-bit Float/Long variables can span either one register (1x32) or two registers (2x16), depending on how the address range is defined.

2. *For more information, refer to "Zero vs. One Based Addressing Within Registers" in **Data Encoding Settings**.

### Array Support

Arrays are supported for all data types. There are two methods of addressing an array. The following examples use holding register locations:

7*xxx* [rows] [cols]
7*xxx* [cols] with assumed row count of one.

For arrays, rows multiplied by cols cannot exceed the block size that has been assigned to the device for the register/coil type. For register arrays of 32-bit data types, rows multiplied by cols multiplied by 2 cannot exceed the block size.

## Automatic Tag Database Generation

The Enron Modbus Driver supports the server's Automatic Tag Database Generation feature. When enabled, a list of tags will be built in the server for registers in the device that are mapped to the EFM Configuration Attributes for each enabled meter. To access and configure the Automatic Tag Database Generation settings, right-click on the device and select **Properties**. Then, open the **Database Creation** tab.
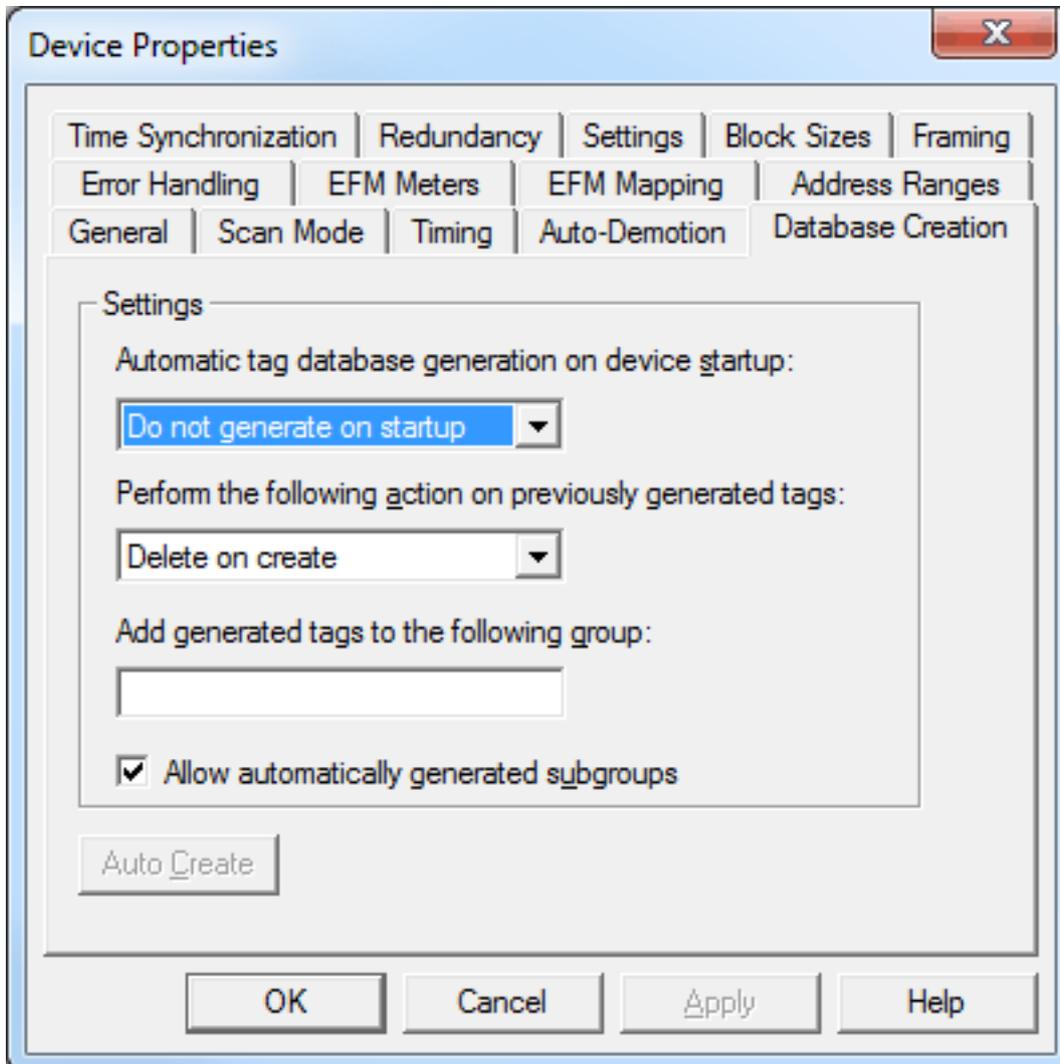


**Note:** For more information on EFM Configuration Attribute Mapping, refer to **Configuration Mapping**.

## Error Descriptions

The following error/warning messages may be generated. Click on the link for a description of the message.

**Address Validation**

**Address <address> is out of range for the specified device or register.**
**Array size is out of range for address <address>.**
**Data Type <type> is not valid for device address <address>.**
**Device address <address> contains a syntax error.**

**Device Status Messages**

**Device <device name> is not responding.**
**Unable to write to <address> on device <device name>.**

**Enron Modbus Specific Messages**

**<Device Name> - Failed to read EFM pointer file. <Extended Error>.**
**<Device Name> - Failed to write EFM pointer file. <Extended Error>.**
**Alarm mapping for address <address> is invalid and will be ignored.**
**Archive address <address> is used in Meter <number> for <archive> archive and in Meter <number> for <archive> archive in device <device name>. Duplicate archive addresses are not allowed.**
**Bad address in block [<start address> to <end address>] on device <device name>.**
**Bad array spanning [<address> to <address>] on device <device>.**
**Block address [<start address> to <end address>] on device <device> responded with exception code <code>.**
**Config attribute <attribute index> is unknown and will be ignored.**
**Config mapping for attribute <attribute name> is invalid and will be ignored.**
**Error parsing alarm/event record. The record size is incorrect.**
**Error parsing history record. History mapping does not match record read from device, record will not be logged.**
**Error reading date and time, alarm/event record will not be logged.**
**Error reading date and time, history record will not be logged.**
**Failure to load <mapping name> mapping from CSV. The header contains a duplicate field name <name>.**
**Failure to load <mapping name> mapping from CSV. The header contains an unrecognized field name <name>.**
**Failure to load <mapping name> mapping from CSV. There is no header in the CSV file.**
**History attribute <attribute index> is unknown and will be ignored.**
**History mapping for attribute <attribute name> is invalid and will be ignored.**
**Meter <number> has an invalid EFM Mapping (<mapping name>). Defaulting the mapping to <mapping name>.**
**Meter name <name> is used in Meter <number> and in Meter <number> in device <device name>. Duplicate meter names are not allowed.**
**Serialization of EFM data to temporary file <file name> failed. Reason: <file I/O error>.**
**The configuration map address <address> for meter <meter name> is beyond the maximum address allowed by the Enron Modbus protocol. This address will be ignored.**
**The EFM Meter Daily GC data value <value> in device <device name>' is not valid. Valid range is 0 or <min> to <max>.**
**The EFM Meter Event Counter value <value> in device <device name> is not valid. Valid range is 0 or <min> to <max>.**
**The EFM Meter Hourly GC data value <value> in device <device name> is not valid. Valid range is 0 or <min> to <max>.**
**Unable to create tag for EFM configuration attribute <attribute> with address <address> on meter <meter name>.**
**Unable to read <address> from device <device name>. The device is configured for broadcast writes only.**
**Unable to read <address> from device <device name>. The device is not responding.**
**Unable to read block address [<start address> to <end address>] on device <device name>. Unexpected characters in response.**
**Unable to read from address <address> on device <device name>: Device responded with exception code <code>.**
**Unable to read from address <address> on device <device name>. Response is not the correct size.**

**Unable to read from address \<address\> on device \<device name\>. Unexpected characters in response.**
**Unable to synchronize time with device \<device name\>. The device is not responding.**
**Unable to write to address \<address\> on device \<device\>. Device responded with exception code \<code\>.**
**Unable to write to address \<address\> on device \<device name\>. Unexpected characters in response.**
**Warning loading \<mapping name\> mapping from CSV. Alarm state for address \<address\> is invalid. Setting the state to off.**
**Warning loading \<mapping name\> mapping from CSV. Alarm type for address \<address\> is invalid. Setting the type to differential pressure alarm.**
**Warning loading \<mapping name\> mapping from CSV. Ignoring alarm with no address.**
**Warning loading \<mapping name\> mapping from CSV. Ignoring record with no address.**
**Warning loading \<mapping name\> mapping from CSV. No records were imported.**
**Warning loading \<mapping name\> mapping from CSV. The attribute \<name\> is unknown, and will be ignored.**

**Serial Communications**
**Communications error on \<channel name\> [\<error mask\>].**
**COMn does not exist.**
**COMn is in use by another application.**
**Error opening COMn.**
**Unable to set comm parameters on COMn.**

**See Also: Modbus Exception Codes**

## Address \<address\> is out of range for the specified device or register.

**Error Type:**
Warning

**Possible Cause:**
A tag address that has been specified statically references a location that is beyond the range of supported locations for the device.

**Solution:**
Verify that the address is correct; if it is not, re-enter it in the client application.

## Array size is out of range for address \<address\>.

**Error Type:**
Warning

**Possible Cause:**
A tag address that has been specified statically is requesting an array size that is too large for the address type or block size of the driver.

**Solution:**
Re-enter the address in the client application to specify a smaller value for the array or a different starting point.

## Data Type \<type\> is not valid for device address \<address\>.

**Error Type:**
Warning

**Possible Cause:**
A tag address that has been specified statically has been assigned an invalid data type.

**Solution:**
Modify the requested data type in the client application.

## Device address \<address\> contains a syntax error.

**Error Type:**

Warning

**Possible Cause:**

A tag address that has been specified statically contains one or more invalid characters.

**Solution:**

Re-enter the address in the client application.

## Communications error on <channel name> [<error mask>].

**Error Type:**
Serious

**Error Mask Definitions:**

**B** = Hardware break detected.
**F** = Framing error.
**E** = I/O error.
**O** = Character buffer overrun.
**R** = RX buffer overrun.
**P** = Received byte parity error.
**T** = TX buffer full.

**Possible Cause:**

1. The serial connection between the device and the Host PC is bad.

2. The communications parameters for the serial connection are incorrect.

**Solution:**

1. Verify the cabling between the PC and the PLC device.

2. Verify that the specified communications parameters match those of the device.

## COMn does not exist.

**Error Type:**
Fatal

**Possible Cause:**

The specified COM port is not present on the target computer.

**Solution:**

Verify that the proper COM port has been selected.

## COMn is in use by another application.

**Error Type:**
Fatal

**Possible Cause:**

The serial port assigned to a device is being used by another application.

**Solution:**

1. Verify that the correct port has been assigned to the channel.

2. Verify that only one copy of the current project is running.

## Error opening COMn.

**Error Type:**
Fatal

**Possible Cause:**
The specified COM port could not be opened due an internal hardware or software problem on the target computer.

**Solution:**
Verify that the COM port is functional and may be accessed by other Windows applications.

## Unable to set comm parameters on COMn.

**Error Type:**
Fatal

**Possible Cause:**
The serial parameters for the specified COM port are not valid.

**Solution:**
Verify the serial parameters and make any necessary changes.

## Device <device name> is not responding.

**Error Type:**
Serious

**Possible Cause:**

1. The serial connection between the device and the Host PC is broken.

2. The communications parameters for the serial connection are incorrect.

3. The named device may have been assigned an incorrect Network ID.

4. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.

**Solution:**

1. Verify the cabling between the PC and the PLC device.

2. Verify the specified communications parameters match those of the device.

3. Verify the Network ID given to the named device matches that of the actual device.

4. Increase the Request Timeout setting so that the entire response can be handled.

## Unable to write to <address> on device <device name>.

**Error Type:**
Serious

**Possible Cause:**

1. The serial connection between the device and the host PC is broken.

2. The communications parameters for the serial connection are incorrect.

3. The named device may have been assigned an incorrect network ID.

**Solution:**

1. Verify the cabling between the PC and the PLC device.

2. Verify the specified communications parameters match those of the device.

3. Verify that the Network ID given to the named device matches that of the actual device.

## <Device Name> - Failed to read EFM pointer file. <Extended Error>.

**Error Type:**
Warning

**Extended Error:**
When supplied by the operating system, this describes the file error that occurred.

**Possible Cause:**

1. A permission error was encountered when the EFM pointer cache was read.

2. The EFM pointer cache file is corrupt.

**Solution:**
The Enron Modbus Driver automatically generates a new EFM pointer file; however, the server re-polls (uploading all EFM data) during the next EFM poll for meters in the device.

**Note:**
For more information, refer to the extended error.

## <Device name> - Failed to write EFM pointer file. <Extended error>.

**Error Type:**
Warning

**Extended Error:**
When supplied by the operating system, this describes the file error that occurred.

**Possible Cause:**

1. The disk is full.

2. A permission error was encountered when the EFM pointer cache was written.

**Solution:**
The server will attempt to update the EFM pointer file periodically, in addition to when the server is shutdown. If the pointer file cannot be written, the server will re-poll (uploading all EFM data) during the next EFM poll for meters in the device.

**Note:**
For more information, refer to the extended error.

## Alarm mapping for address <address> is invalid and will be ignored.

**Error Type:**
Warning

**Possible Cause:**
An invalid Alarm Mapping was imported from a CSV file or loaded from an XML project file.

**Solution:**
Correct the Alarm Mapping in the CSV import file or the XML project file.

**See Also:**
**Alarm Mapping**

## Archive address <address> is used in Meter <number> for <archive> archive and in Meter < number> for <archive> archive in device <device name>. Duplicate archive addresses are not allowed.

**Error Type:**
Serious

**Possible Cause:**

The XML project file contains duplicate archive addresses.

**Solution:**

Ensure that the XML project file does not contain duplicate archive addresses.

**Note:**

Daily and hourly GC archive addresses may be shared across meters.

## Bad address in block [<start address> to <end address>] on device <device name>.

**Error Type:**
Serious

**Possible Cause:**

1. An attempt has been made to reference a nonexistent location in the specified device.

2. An attempt has been made to read more registers than allowed by the protocol.

**Solution:**

1. Verify the tags assigned to addresses in the specified range on the device and eliminate ones that reference invalid locations.

2. Decrease the register **block size** value to 125.

**See Also:**
**Error Handling**
**Block Sizes**

## Bad array spanning [<address> to <address>] on device <device>.

**Error Type:**
Serious

**Possible Cause:**

1. An attempt has been made to reference a nonexistent location in the specified device.

2. An attempt has been made to read more registers than allowed by the protocol.

**Solution:**

1. Verify that all the register addresses requested in the array exist in the device and reduce the array size such that only valid addresses (that exist in the device) are requested by the array.

2. Reduce the array size value to 125.

**See Also:**
**Error Handling**
**Block Sizes**

## Block address [<start address> to <end address>] on device <device> responded with exception code <code>.

**Error Type:**
Warning

**Possible Cause:**

For a description of the exception codes, refer to **Modbus Exception Codes**.

**Solution:**
For a description of the exception codes, refer to **Modbus Exception Codes**.

## Config attribute <attribute index> is unknown and will be ignored.

**Error Type:**
Warning

**Possible Cause:**
An invalid Configuration Mapping was imported from a CSV file or loaded from an XML project file.

**Solution:**
Correct the Configuration Mapping in the CSV import file or the XML project file.

**See Also:**
**Configuration Mapping**

## Config mapping for attribute <attribute name> is invalid and will be ignored.

**Error Type:**
Warning

**Possible Cause:**
An invalid Configuration Mapping was imported from a CSV file or loaded from an XML project file.

**Solution:**
Correct the Configuration Mapping in the CSV import file or the XML project file.

**See Also:**
**Configuration Mapping**

## Error parsing alarm/event record. The record size is incorrect.

**Error Type:**
Warning

**Possible Cause:**
An EFM Alarm/Event archive record returned by the device is not a complete Enron Modbus Historical or Alarm/Event archive record.

**Solution:**
Verify that the EFM Archive settings are correct.

**See Also:**
**EFM Meters**

## Error parsing history record. History mapping does not match record read from device, record will not be logged.

**Error Type:**
Warning

**Possible Cause:**
The History Mapping does not match the History Record that was returned from the device.

**Solution:**
Verify that the History Mapping is correct for the device.

## Error reading date and time, alarm/event record will not be logged.

**Error Type:**
Warning

**Possible Cause:**
The date and time format in the Alarm/Event record returned by the device could not be read.

**Solution:**
Verify that the EFM Meter Event's Word and Byte order are correct.

**See Also:**
**EFM Meters**

## Error reading date and time, history record will not be logged.

**Error Type:**
Warning

**Possible Cause:**
The date and time format in the History Record returned by the device could not be read.

**Solution:**
Verify that the EFM Meter History's Word and Byte order are correct.

**See Also:**
**EFM Meters**

## Failure to load <mapping name> mapping from CSV. The header contains a duplicate field name <name>.

**Error Type:**
Fatal

**Possible Cause:**
The CSV file header contains a duplicate field name.

**Solution:**
Verify that the CSV file is a valid EFM Mapping CSV import file.

## Failure to load <mapping name> mapping from CSV. The header contains an unrecognized field name <name>.

**Error Type:**
Fatal

**Possible Cause:**
The CSV file header contains an invalid field name.

**Solution:**
Verify that the CSV file is a valid EFM Mapping CSV import file.

## Failure to load <mapping name> mapping from CSV. There is no header in the CSV file.

**Error Type:**
Fatal

**Possible Cause:**
The CSV file does not contain a valid header.

**Solution:**
Verify that the CSV file is a valid EFM Mapping CSV import file.

## History attribute <attribute index> is unknown and will be ignored.

**Error Type:**
Warning

**Possible Cause:**

An invalid History Mapping was imported from a CSV file or loaded from an XML project file.

**Solution:**

Correct the History Mapping in the CSV import file or the XML project file.

**See Also:**
**History Mapping**

## History mapping for attribute <attribute name> is invalid and will be ignored.

**Error Type:**

Warning

**Possible Cause:**

An invalid History Mapping was imported from a CSV file or loaded from an XML project file.

**Solution:**

Correct the History Mapping in the CSV import file or the XML project file.

**See Also:**
**History Mapping**

## Meter <number> has an invalid EFM Mapping (<mapping name>). Defaulting the mapping to <mapping name>.

**Error Type:**

Warning

**Possible Cause:**

The EFM Mapping specified for a meter/run in the project file is missing or invalid.

**Solution:**

Verify that the project file contains the specified EFM Mapping.

## Meter name <name> is used in Meter <number> and in Meter <number> in device <device name>. Duplicate meter names are not allowed.

**Error Type:**

Serious

**Possible Cause:**

The XML project file contains duplicate meter names.

**Solution:**

Ensure that the XML project file does not contain duplicate meter names.

## Serialization of EFM data to temporary file <file name> failed. Reason: <file I/O error>.

**Error Type:**

Warning

**Possible Cause:**

1. The driver was unable to create the specified file directory.

2. The driver was unable to access the specified file.

**Solution:**

1. Verify that the disk has sufficient disk space.

2. Verify user permissions for the specified file directory.

## The configuration map address <address> for meter <meter name> is beyond the maximum address allowed by the Enron Modbus protocol. This address will be ignored.

**Error Type:**
Warning

**Possible Cause:**
When the offset is applied for the specified meter, the base address of an EFM configuration attribute extends beyond the maximum address that is allowed by the Enron Modbus Protocol.

**Solution:**
Verify that the specified meter's base address and offset are correct.

**See Also:**
**Configuration Mapping**
**EFM Meters**

## The EFM Meter Daily GC data value <value> in device <device name>' is not valid. Valid range is 0 or <min> to <max>.

**Error Type:**
Serious

**Possible Cause:**
The XML project file contains a Daily GC data value that is out of range.

**Solution:**
Ensure that the Daily GC data value is within the specified range.

## The EFM Meter Event Counter value <value> in device <device name> is not valid. Valid range is 0 or <min> to <max>.

**Error Type:**
Serious

**Possible Cause:**
The XML project file contains an Event Counter value that is out of range.

**Solution:**
Ensure that the Event Counter value is within the specified range.

## The EFM Meter Hourly GC data value <value> in device <device name> is not valid. Valid range is 0 or <min> to <max>.

**Error Type:**
Serious

**Possible Cause:**
The XML project file contains an Hourly GC data value that is out of range.

**Solution:**
Ensure that the Hourly GC data value is within the specified range.

## Unable to create tag for EFM configuration attribute <attribute> with address <address> on meter <meter name>.

**Error Type:**

Warning

**Possible Cause:**
The calculated address for the attribute is out of range when given the meter number.

**Solution:**

1. Ensure that the address mapped to the attribute is correct.

2. Ensure that the offset is correct for the data type.

3. Ensure that the address ranges are properly configured for the device.

**See Also:**
**Configuration Mapping**
**EFM Meters**
**Address Ranges**

## Unable to read <address> from device <device name>. The device is configured for broadcast writes only.

**Error Type:**
Warning

**Possible Cause:**
The device is configured for broadcast writes only, and an EFM Poll was triggered. The Device ID is set to 0.

**Solution:**

1. Disable EFM polling for broadcast devices.

2. Do not use a Device ID of 0 for EFM-enabled devices.

## Unable to read block address [<start address> to <end address>] on device <device name>. Unexpected characters in response.

**Error Type:**
Warning

**Possible Cause:**
The calculated CRC did not match the CRC that was sent by the device.

**Solution:**

1. Verify the cabling between the PC and the PLC device.

2. Verify that the specified communications parameters match those of the device.

## Unable to read from address <address> on device <device>: Device responded with exception code <code>.

**Error Type:**
Warning

**Possible Cause:**
For a description of the exception code, refer to **Modbus Exception Codes**.

**Solution:**
For a description of the exception code, refer to **Modbus Exception Codes**.

## Unable to read from address <address> on device <device name>. Response is not the correct size.

**Error Type:**
Warning

**Possible Cause:**
An EFM upload request response did not contain a complete Enron Modbus Historical or Alarm/Event archive record.

**Solution:**
Verify that the EFM Archive settings are correct.

**See Also:**
**EFM Meters**

## Unable to read from address <address> on device <device name>. Unexpected characters in response.

**Error Type:**
Warning

**Possible Cause:**
The calculated CRC did not match the CRC that was sent by the device.

**Solution:**

1. Verify the cabling between the PC and the PLC device.

2. Verify that the specified communications parameters match those of the device.

## Unable to read <address> from device <device name>. The device is not responding.

**Error Type:**
Warning

**Possible Cause:**
The device is not responding to a read request.

**Solution:**

1. Verify the cabling between the PC and the PLC device.

2. Verify that the specified communications parameters match those of the device.

3. Verify that the Network ID given to the named device matches that of the actual device.

4. Increase the Request Timeout setting so that the entire response can be handled.

## Unable to synchronize time with device <device name>. The device is not responding.

**Error Type:**
Warning

**Possible Cause:**
The device is not responding to a time synchronization write.

**Solution:**

1. Verify the cabling between the PC and the PLC device.

2. Verify that the specified communications parameters match those of the device.

3. Verify that the Network ID given to the named device matches that of the actual device.

4. Increase the Request Timeout setting so that the entire response can be handled.

## Unable to write to address <address> on device <device>: Device responded with exception code <code>.

**Error Type:**
Warning

**Possible Cause:**
For a description of the exception code, refer to **Modbus Exception Codes**.

**Solution:**
For a description of the exception code, refer to **Modbus Exception Codes**.

## Unable to write to address <address> on device <device name>. Unexpected characters in response.

**Error Type:**
Warning

**Possible Cause:**
The calculated CRC did not match the CRC that was sent by the device.

**Solution:**

1. Verify the cabling between the PC and the PLC device.

2. Verify that the specified communications parameters match those of the device.

## Warning loading <mapping name> mapping from CSV. Alarm state for address <address> is invalid. Setting the state to off.

**Error Type:**
Warning

**Possible Cause:**
The alarm state in the CSV import file for the specified address is invalid.

**Solution:**
Verify that the alarm state in the CSV import file is correct.

## Warning loading <mapping name> mapping from CSV. Alarm type for address <address> is invalid. Setting the type to differential pressure alarm.

**Error Type:**
Warning

**Possible Cause:**
The alarm type in the CSV import file for the specified address is invalid.

**Solution:**
Verify that the alarm type in the CSV import file is correct.

## Warning loading <mapping name> mapping from CSV. Ignoring alarm with no address.

**Error Type:**

Warning

**Possible Cause:**
The CSV import file contains an alarm that does not specify an address.

**Solution:**
Verify that an alarm address is present in the CSV import file and is correct.

## Warning loading <mapping name> mapping from CSV. Ignoring record with no address.

**Error Type:**
Warning

**Possible Cause:**
The CSV import file contains a configuration or history attribute that does not specify an address.

**Solution:**
Verify that an attribute address is present in the CSV import file and is correct.

## Warning loading <mapping name> mapping from CSV. No records were imported.

**Error Type:**
Warning

**Possible Cause:**
The CSV import file did not contain any valid records.

**Solution:**
Verify that the CSV import file contains valid records.

## Warning loading <mapping name> mapping from CSV. The attribute <name> is unknown, and will be ignored.

**Error Type:**
Warning

**Possible Cause:**
The specified attribute in the CSV import file is unknown.

**Solution:**
Verify that the attribute in the CSV import file is correct.

## Modbus Exception Codes

The following data is from Modbus Application Protocol Specifications documentation.

| Code Dec/Hex | Name | Meaning |
|---|---|---|
| 01/0x01 | ILLEGAL FUNCTION | The function code received in the query is not an allowable action for the server (or slave). This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server (or slave) is in the wrong state to process a request of this type, for example, because it is unconfigured and is being asked to return register values. |
| 02/0x02 | ILLEGAL DATA ADDRESS | The data address received in the query is not an allowable address for the server (or slave). More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 would succeed. A request with offset 96 and length 5 will generate exception 02. |
| 03/0x03 | ILLEGAL DATA VALUE | A value contained in the query data field is not an allowable value for server (or slave). This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does not mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the MODBUS protocol is unaware of the significance of any particular value of any particular register. |
| 04/0x04 | SLAVE DEVICE FAILURE | An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action. |
| 05/0x05 | ACKNOWLEDGE | The slave has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the master. The master can next issue a Poll Program Complete message to determine if processing is completed. |
| 06/0x06 | SLAVE DEVICE BUSY | The slave is engaged in processing a long-duration program command. The master should retransmit the message later when the slave is free. |
| 07/0x07 | NEGATIVE ACKNOWLEDGE | The slave cannot perform the program function received in the query. This code is returned for an unsuccessful programming request using function code 13 or 14 decimal. The master should request diagnostic or error information from the slave. |
| 08/0x08 | MEMORY PARITY ERROR | The slave attempted to read extended memory, but detected a parity error in the memory. The master can retry the request, but service may be required on the slave device. |
| 10/0x0A | GATEWAY PATH UNAVAILABLE | Specialized use in conjunction with gateways indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. This usually means that the gateway is misconfigured or overloaded. |
| 11/0x0B | GATEWAY TARGET DEVICE FAILED TO RESPOND | Specialized use in conjunction with gateways indicates that no response was obtained from the target device. This usually means that the device is not present on the network. |

**Note:** For this driver, the terms Slave and Unsolicited are used interchangeably.

# Index

## A

## B

## C

## D

# E

# F

# H

# M

# O

## P

## S

## T

## U

## W