OMNI Flow Computer Help

Table of Contents

OMNI Flow Computer Driver Help

Help version 1.013

CONTENTS

Overview

The OMNI Flow Computer Driver provides real-time and EFM data access. In addition to archive, time zone, and device password settings, the driver configuration 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).

Channel Setup

Communication Serialization

The OMNI Flow Computer 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: 1200, 2400, 9600, and 19200.
Parity: Odd, Even, and None.
Data Bits: 8.
Stop Bits: 1 and 2.

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

Supported Firmware Versions

23.xx
27.xx

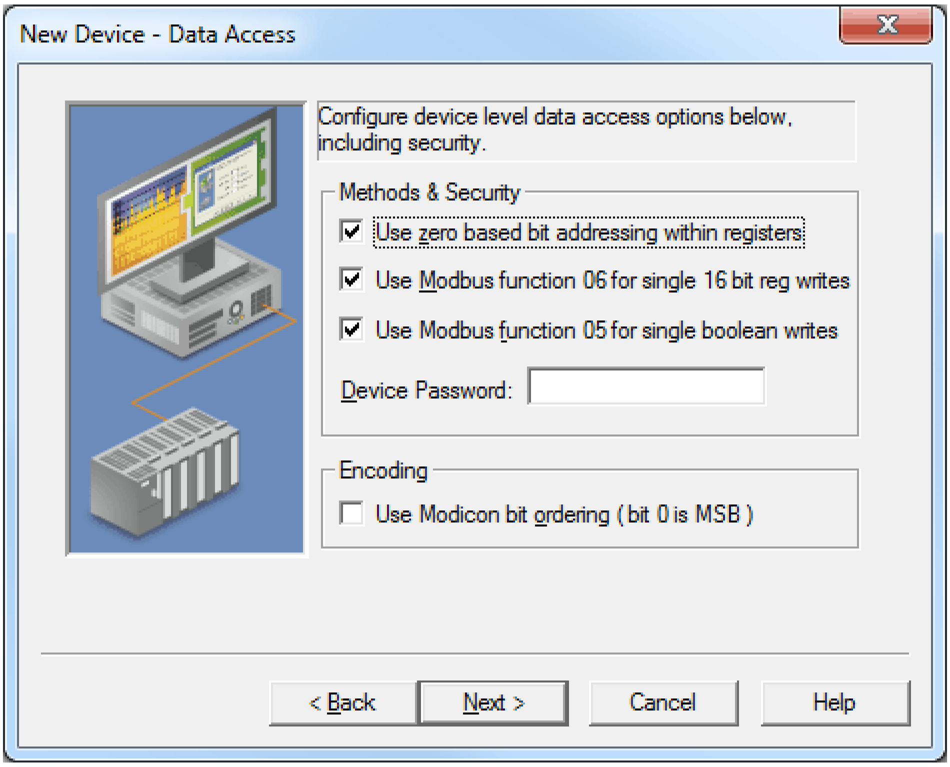
Maximum Number of Channels and Devices

The maximum number of supported channels is 256. The maximum number of devices supported per channel is 255.

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.

Data Access



New Device - Data Access

Configure device level data access options below, including security.

Methods & Security

☑ Use zero based bit addressing within registers

☑ Use Modbus function 06 for single 16 bit reg writes

☑ Use Modbus function 05 for single boolean writes

Device Password:

Encoding

□ Use Modicon bit ordering (bit 0 is MSB)

Descriptions of the parameters are as follows:

- **Use zero based bit addressing within registers:** When checked, this option will use zero based bit addressing within registers and will start the first bit at 0. The default setting is checked. For more information, refer to the "Zero vs. One Based Bit Addressing Within Registers" subtopic below.
- **Use Modbus function 06 for single 16 bit reg writes:** When checked, this option will use Modbus function 06 for single 16 bit register writes. The default setting is checked. For more information, refer to the "Use Modbus Function 06 for Single 16 Bit Reg Writes" subtopic below.
- **Use Modbus function 05 for single boolean writes:** When checked, this option will use Modbus function 05 for single boolean writes. This allows the driver to operate as it has historically, switching between

05 and 15 as needed. When unchecked, all writes will be done using only Modbus function 15. The default setting is checked. For more information, refer to the "Use Modbus Function 05 for Single Boolean Writes" subtopic below.

- **Device Password:** This parameter specifies a password for the device.
- **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.

Zero vs. One Based Bit Addressing Within Registers

Memory types that allow bits within Words can be referenced as a Boolean. The addressing notation for 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 the first bit begins at 0. One Based Bit addressing means that the first bit begins at 1. Descriptions are as follows:

- **Zero Based:** For the Word data type, the bit range is 0 to 15.
- **One Based:** For the Word data type, the bit range is 1 to 16.

Use Modbus Function 06 for Single 16 Bit Reg Writes

Although all OMNI Firmware revisions support Function Code 06 when writing a single 16 bit register, revisions older than xx.44 will not support Function Code 06 when writing a single 32 bit register. As such, Function Code 16 must be used when writing one or more 32 bit registers on the older Firmware revisions. Even though newer Firmware revisions support Function Code 06 for single 32 bit register writes, Function Code 16 will always be used for single 32 bit register writes in order to support legacy installations.

Use Modbus Function 05 for Single Boolean Writes

The OMNI Flow Computer Driver can use two Modbus protocol functions to write output coil data to the target device. In most cases, it will switch between these two functions based on the number of coils being written. When writing a single coil, the driver will use the Modbus function 05. When writing an array of coils, the driver will use Modbus function 15. The standard Modicon PLC can use either of these functions. There are many Third-Party devices that have implemented the Modbus protocol, however, and only support the use of Modbus function 15 to write to output coils (regardless of the number of coils being written).

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.

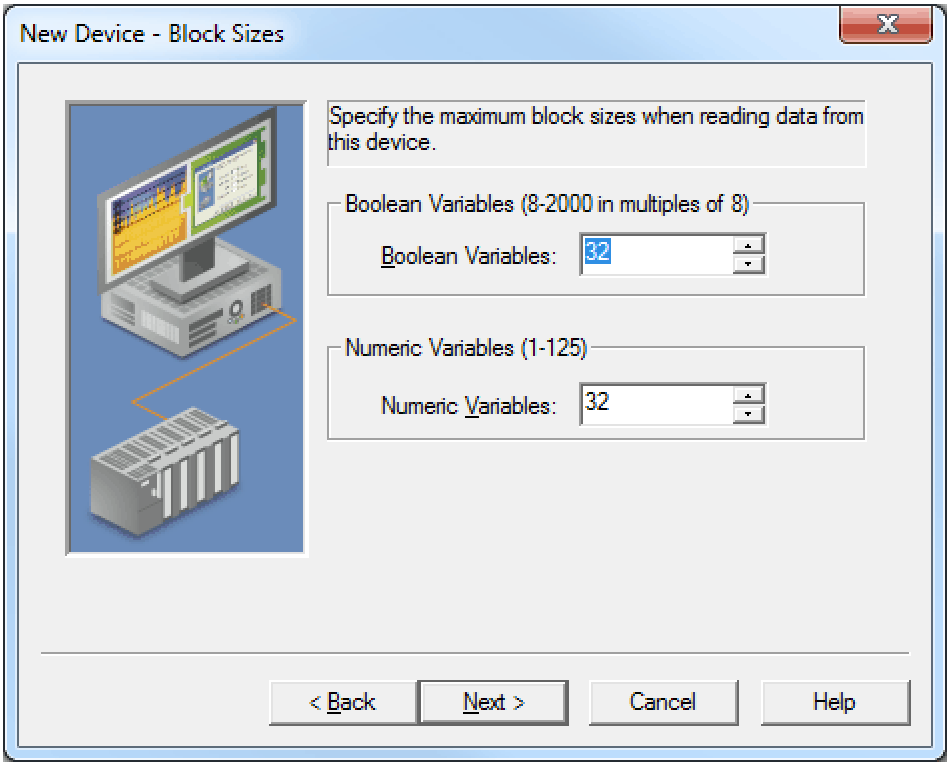
Use Modicon Bit Ordering Checked

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

Block Sizes



New Device - Block Sizes

Specify the maximum block sizes when reading data from this device.

Boolean Variables (8-2000 in multiples of 8)

Boolean Variables: 32

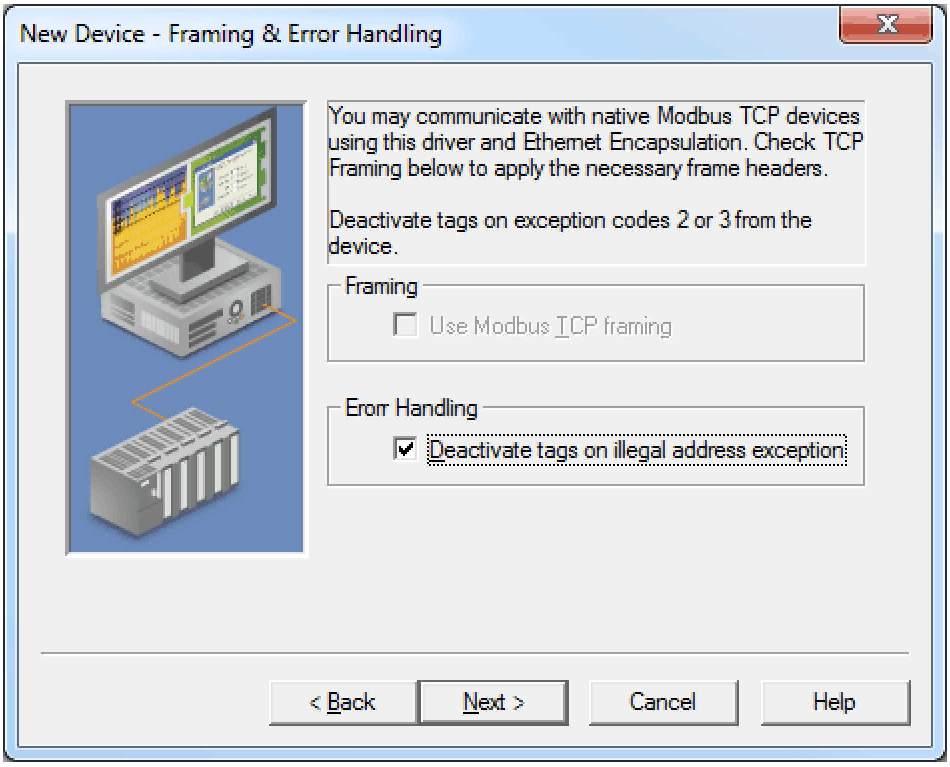
Numeric Variables (1-125)

Numeric Variables: 32

Descriptions of the parameters are as follows:

- **Boolean Variables:** This parameter specifies the Boolean variables. The valid range is 8 to 2000, in multiples of 8. The default setting is 32.
- **Numeric Variables:** This parameter specifies the Numeric variables. The valid range is 1 to 125. The default setting is 32.

Framing & Error Handling



New Device - Framing & Error Handling

You may communicate with native Modbus TCP devices using this driver and Ethernet Encapsulation. Check TCP Framing below to apply the necessary frame headers.

Deactivate tags on exception codes 2 or 3 from the device.

Framing

□ Use Modbus TCP framing

Error Handling

☑ Deactivate tags on illegal address exception

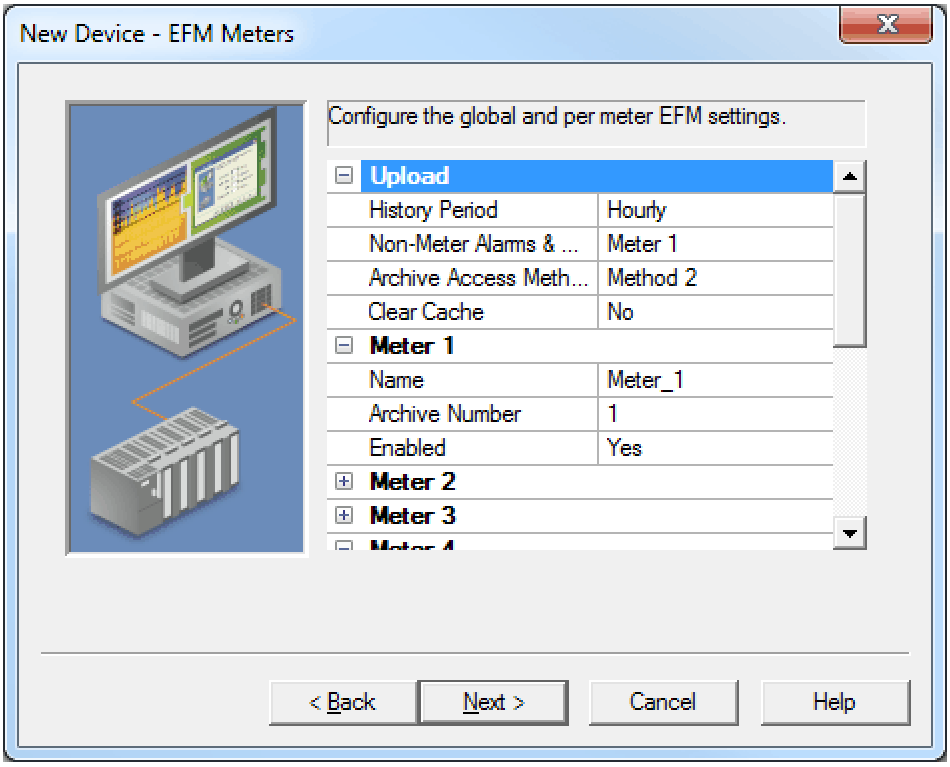
Descriptions of the parameters are as follows:

- **Use Modbus TCP Framing:** When checked, this option enables communications with native Modbus TCP devices using Ethernet Encapsulation. The default setting is unchecked.
- **Deactivate tags on illegal address exception:** When checked, the driver will 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.

EFM Meters

This dialog contains meter-specific EFM configuration and upload settings. The OMNI Flow Computer Driver supports up to four meters.

Upload



New Device - EFM Meters

Configure the global and per meter EFM settings.

Upload	
History Period	Hourly
Non-Meter Alarms & ...	Meter 1
Archive Access Meth...	Method 2
Clear Cache	No
Meter 1	
Name	Meter_1
Archive Number	1
Enabled	Yes
Meter 2	
Meter 3	
Meter 4	

Descriptions of the parameters are as follows:

- **History Period:** The driver assumes that the device is configured to log either Hourly or Daily history data. This property must match the device's configuration. Options include Hourly and Daily. The default setting is Hourly.

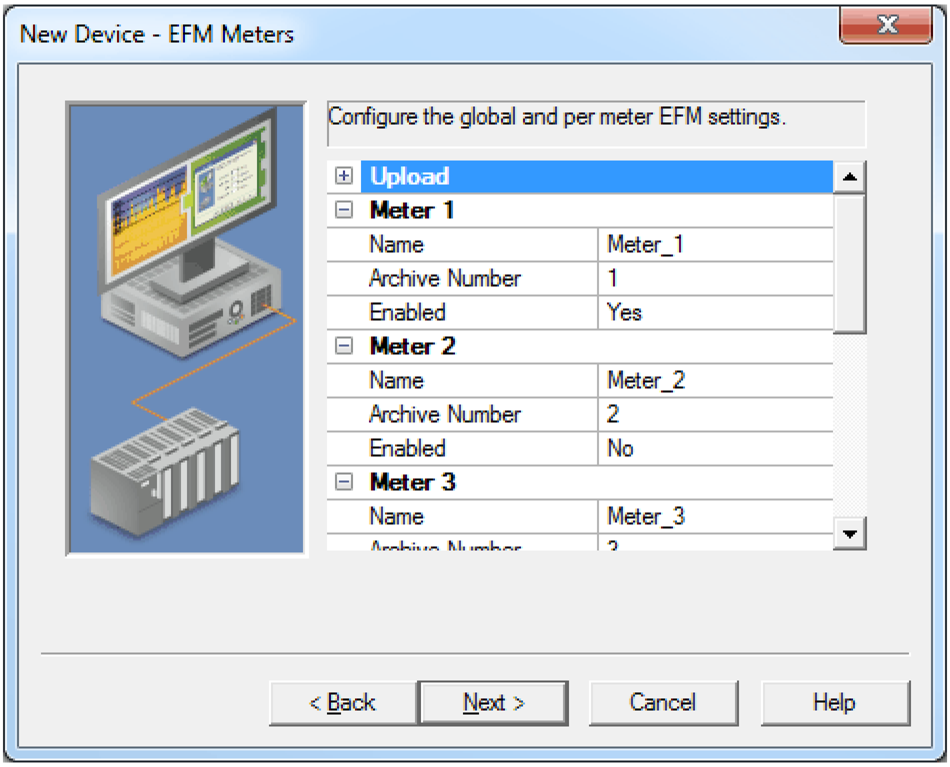
  Note: This setting does not affect the data that is displayed to clients: it only affects how the data is labeled.
- **Non-Meter Alarms & Events:** OMNI devices produce some alarms and events that are not meter specific. This parameter specifies the location where these alarms and events will be stored. Options include Ignore, Meter 1, Meter 2, Meter 3, Meter 4, and All. The default setting is Meter 1. Descriptions of the options are as follows:
  - **Ignore:** The alarms and events will not be sent to any meters.
  - **Meter 1, Meter 2, Meter 3, or Meter 4:** The alarms and events will be sent to one specified meter.
  - **All:** The alarms and events will be sent to all meters.
- **Archive Access Method:** This parameter specifies whether archives will be accessed using Method 1 or Method 2. Method 1 is the legacy method, and Method 2 is more efficient. The default setting is Method 2.

Note: When Method 2 is selected but is not supported by the device, the read will fail and the driver will fall back to Method 1 and issue a warning.

- **Clear Cache:** This parameter specifies whether to clear the EFM cache maintained by the server, which stores history, alarms, and events data for each meter. When enabled, the cache will be cleared on the next poll. All EFM data in the device will be requested again on the next poll that follows. 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*



New Device - EFM Meters

Configure the global and per meter EFM settings.

⊞ **Upload**	
⊟ **Meter 1**	
Name	Meter_1
Archive Number	1
Enabled	Yes
⊟ **Meter 2**	
Name	Meter_2
Archive Number	2
Enabled	No
⊟ **Meter 3**	
Name	Meter_3
Archive Number	3

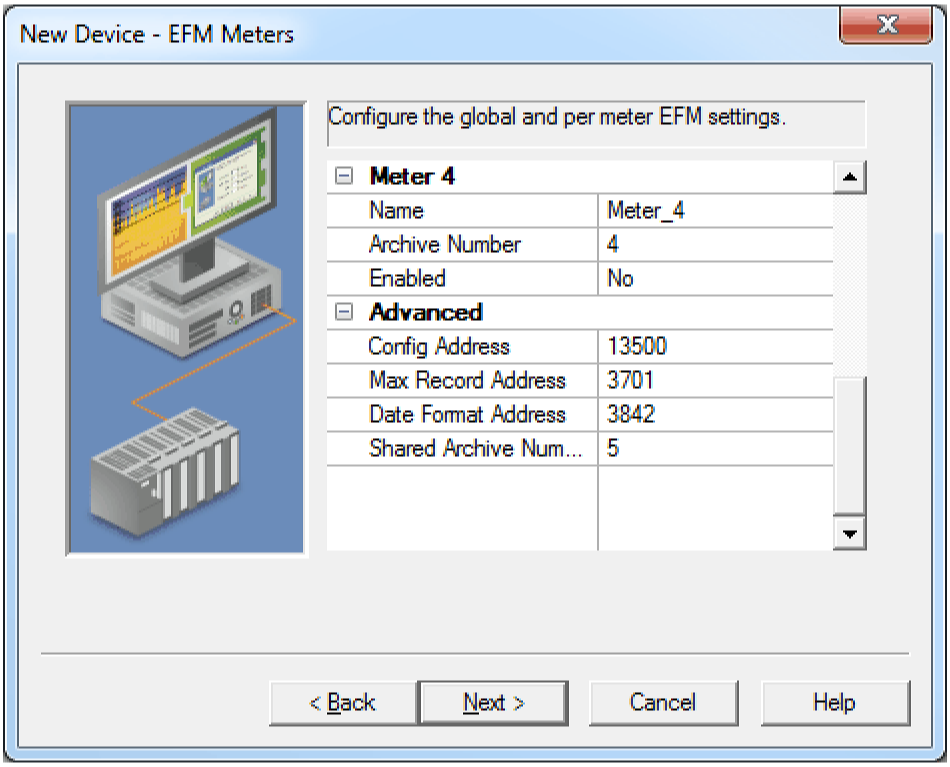
Descriptions of the parameters are as follows:

- **Name:** This parameter specifies a descriptive name for the meter. It must be at least one character, and cannot exceed 128 characters. The '@' character is restricted.

  Note: Multiple meters may not use the same meter name.
- **Archive Number:** This parameter specifies the device archive number that will store the meter's historical data. This number must match the physical device configuration. The valid range is 1 to 10.

  Note: This design assumes that each meter stores historical data in a single unique device archive.
- **Enabled:** This parameter specifies whether the meter is enabled. EFM data will not be retrieved for disabled meters. Options include Yes and No. The default setting for Meter 1 is Yes. The default setting for Meter 2, Meter 3, and Meter 4 is No.

Advanced



Configure the global and per meter EFM settings.

Meter 4	
Name	Meter_4
Archive Number	4
Enabled	No
Advanced	
Config Address	13500
Max Record Address	3701
Date Format Address	3842
Shared Archive Num...	5

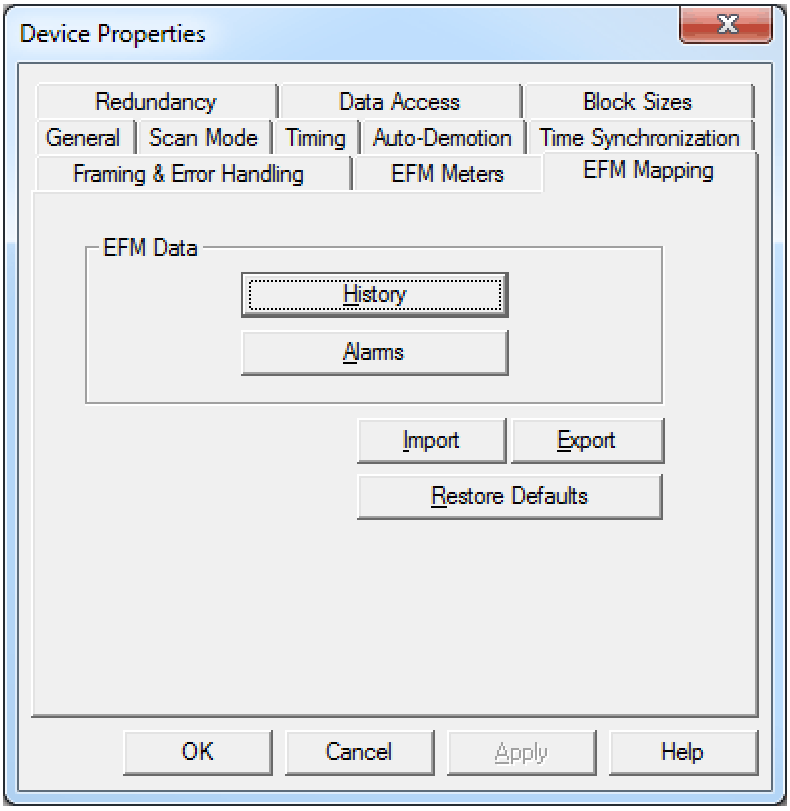
Descriptions of the parameters are as follows:

- **Config Address:** This parameter specifies the group configuration address for archive 1 in the device. The default setting is 13500. This setting should usually not be changed.
- **Max Record Address:** This parameter specifies the device address whose value is the maximum number of records for archive 1 in the device. The default setting is 3701. This setting should usually not be changed.
- **Date Format Address:** This parameter specifies the Date Format address that defines the date format for all archive data. The default setting is 3842. This setting should usually not be changed.
- **Shared Archive Number:** This parameter specifies the device archive number for shared Gas Chromatograph (GC) data. Each meter shares this archive. The valid range is 1 to 10. The default setting is 5.

  Note: The device will not query for GC data if there is no GC data configured in the EFM History Mapping.

EFM Mapping

This dialog configures the mapping of EFM data in the device to the server's EFM data model. The History and Alarms Mappings can be configured. Configuration and Event data is fixed.



Descriptions of the parameters are as follows:

- **History:** When clicked, this button launches the EFM History Mapping dialog, which is used to map history records from the device to EFM attributes. The EFM History Mapping is applied to all enabled meters. For more information, refer to **EFM History Mapping**.
- **Alarms:** When clicked, this button launches the EFM Alarm Mapping dialog, which is used to map alarms to the server's alarms. The EFM Alarm Mapping is applied to all enabled meters. For more information, refer to **EFM Alarm Mapping**.
- **Import:** When clicked, this button launches the Import from CSV dialog, which is used to import EFM History and EFM Alarm Mappings from a CSV file. The CSV file will replace all existing mappings.
- **Export:** When clicked, this button launches the Export to CSV dialog, which is used to export the current EFM History and EFM Alarm Mappings to a CSV file for easy editing.
- **Restore Defaults:** When clicked, this button restores both the EFM History Mapping and the EFM Alarm Mapping to their default settings.

Configuration Mapping

The OMNI Flow Computer Driver does not support configuration of the EFM Configuration Mapping. The driver will request the configuration data from the device on each poll per meter as listed in the table below.

Note: The *N* syntax denotes a base address, where *N* is replaced with the meter number.

Attribute	Address
Meter ID	4N14
Meter Type	3N08
Pressure Base	7894
Temperature Base	7892
Live Analysis	3839
Live BTU	13058
Live Gravity	13053
Live Temperature	13041
AGA8/FPV Method	3N11
Pipe Diameter	7N48

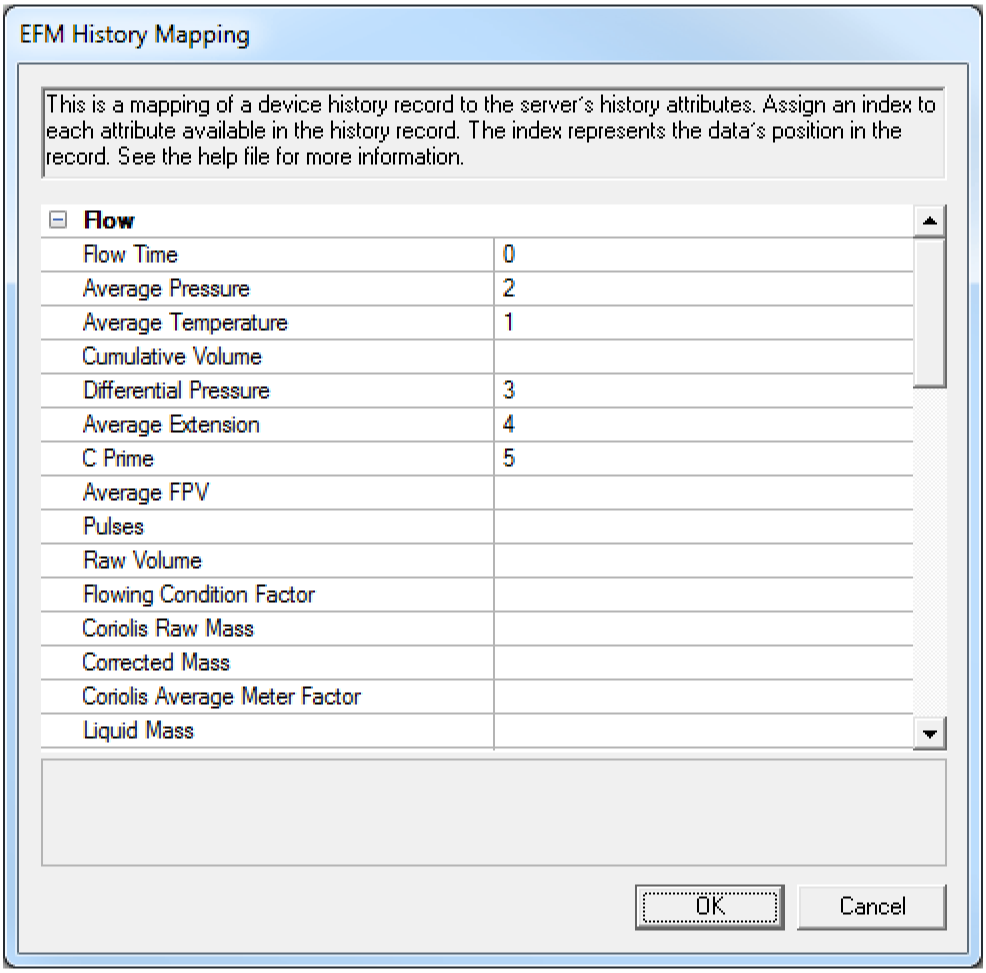
Pipe Material	7N49
Pipe Reference Temperature	7N50
Meter Tap	3N12
Static Pressure Tap	3N10
Orifice Plate Size	7N45
Orifice Plate Material	7N46
Orifice Plate Temperature	7N47
DP Low Flow Cutoff	7N51
Atmospheric Pressure	7891
DP Low Alarm	7N52
DP High Alarm	7N53
Temperature Low Alarm	7N63
Temperature High Alarm	7N64
Static Pressure Low Alarm	7N68
Static Pressure High Alarm	7N69
K Factor	17501
Meter Factor	17179
Specific Heats	17252
BTU	17253
Specific Gravity	17254
Viscosity	17251
Contract Hour	4819
Units	4849
C1	5N21/5N26*
N2	5N22/5N21*
CO2	5N33/5N22*
C2	5N24/5N27*
C3	5N25/5N28*
ISO C4	5N31/5N30*
NC4	5N32/5N29*
ISO C5	5N33/5N32*
C5	5N34/5N31*
NEO C5	5N42
C6	5N35/5N33*
C7	5N36/5N34*
C8	5N37/5N35*
C9	5N38/5N36*
C10	5N39/5N37*
O2	5N30/5N38*
H2O	5N26/5N24*
H2S	5N27/5N23*
HE	5N40/5N25*
H2	5N28/5N40*
CO	5N29/5N39*
AR	5N41

*The address of the Gas Chromatograph data is independent of the Calculation Method. It is the address on the left for AGA8 1992-94 and the address on the right for all others.

EFM History Mapping

History data pulled from the device is in record form, with each record containing an array of data points. Each data point has a unique index or position in the array. The valid range is 0 to 31.

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 will be parsed using this mapping, which can include data from the Gas Chromatograph (GC) archives. When data is not included from the GC archives, they will not be uploaded.



EFM History Mapping

This is a mapping of a device history record to the server's history attributes. Assign an index to each attribute available in the history record. The index represents the data's position in the record. See the help file for more information.

Flow	
Flow Time	0
Average Pressure	2
Average Temperature	1
Cumulative Volume	
Differential Pressure	3
Average Extension	4
C Prime	5
Average FPV	
Pulses	
Raw Volume	
Flowing Condition Factor	
Coriolis Raw Mass	
Corrected Mass	
Coriolis Average Meter Factor	
Liquid Mass	

History Syntax

A History Index uses the following syntax: *N* or *S<N>* where:

- **N:** This index in the record associates with an attribute. The valid range is 0 to 31.
- **S:** This optional index indicates that the element comes from the Shared Archive configured in the EFM Meters dialog.

  Note: GC data will not be requested from the device when S indices are not used.

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**

Meter History Attributes & Mappings

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

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. Fahrenheit for English and Celsius for Metric.
Cumulative Volume	cumulative_volume	Float	Volume added during this interval for orifice and turbine meters.

			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. 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. Units are pounds for English and KG for Raw Mass.
Corrected Mass	corrected_mass	Float	Corrected mass for coriolis meters. 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. Units are pounds for English and KG for Metric.
Liquid Volume	liquid_volume	Float	Volume for liquid meters. Units are controlled by the Volume Units in the Configuration Mapping.
Liquid Energy	liquid_energy	Float	Energy for liquid meters. Units are BTU/cubic foot for English and MJ/cubic meter for Metric.
Total Volume	total_volume	Float	Total volume. Units are controlled by the Volume Units in the Configuration Mapping.
Total Energy	total_energy	Float	Total energy. Units are BTU/cubic foot for English and MJ/cubic meter for Metric.
Average BTU	avg_btu	Float	Average heating value. 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. Units of Pounds/Mass per Foot/Second for English and Centipoises for Metric.

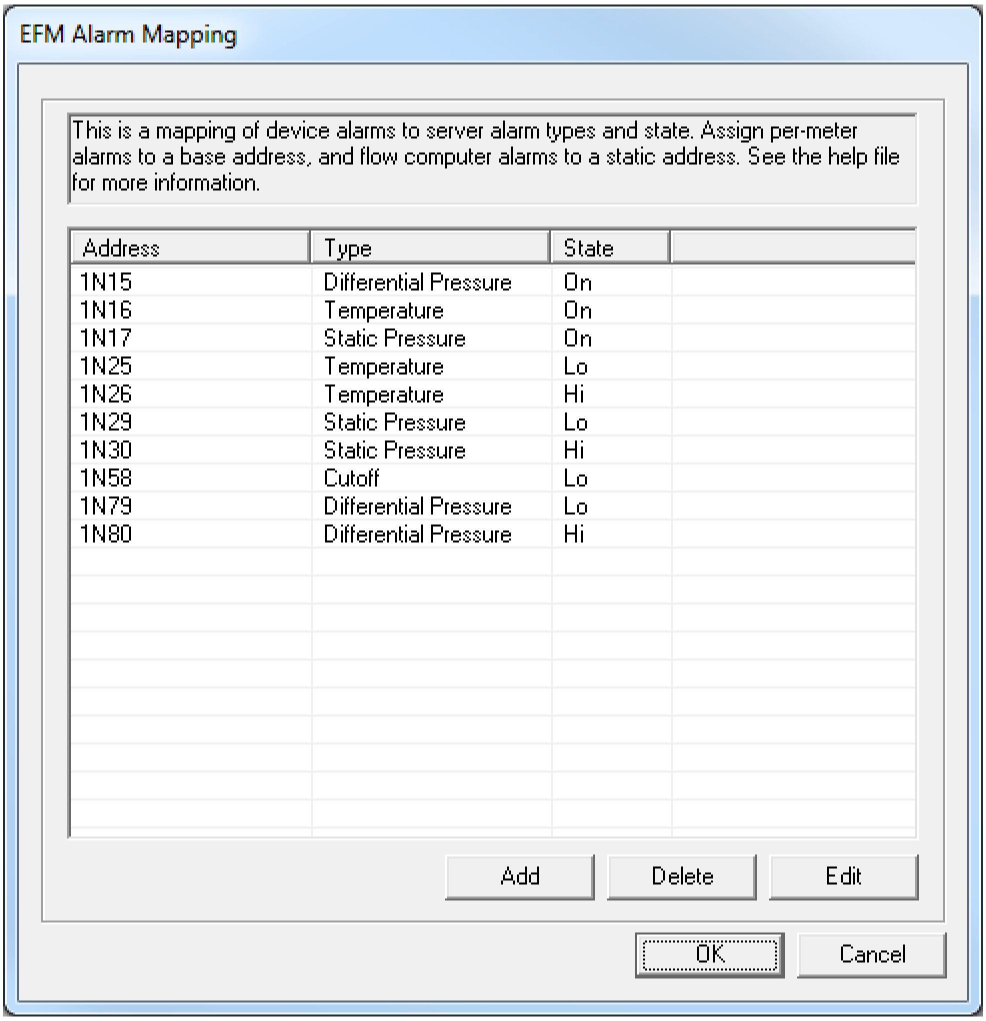
*Inches of Water for English and Kilopascals for Metric.

EFM 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 the device that does not match an address in the Alarm Mappings, the alarm will be logged as a user string alarm. It will be handled according to the "Non-Meter Alarms & Events" setting located in EFM Meters.

See Also: EFM Meters



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
1N15	Differential Pressure	On
1N16	Temperature	On
1N17	Static Pressure	On
1N25	Temperature	Lo
1N26	Temperature	Hi
1N29	Static Pressure	Lo
1N30	Static Pressure	Hi
1N58	Cutoff	Lo
1N79	Differential Pressure	Lo
1N80	Differential Pressure	Hi

Descriptions of the parameters are as follows:

- **Address:** This parameter specifies the OMNI device 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
    - Lo
    - Hi
- **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

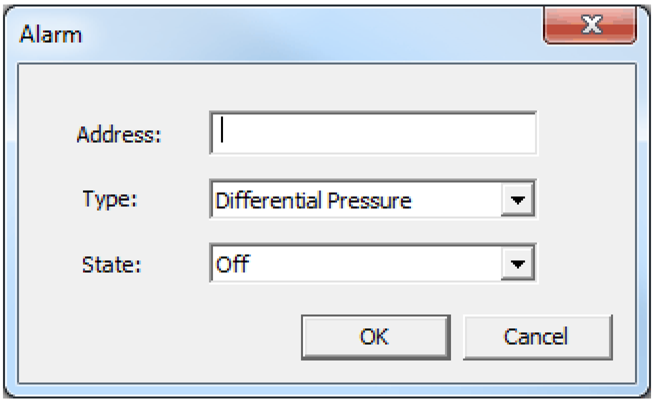
An alarm address may use one of the following syntactic forms:

- **1N34:** This is a base address, and makes the alarm meter-specific. N will be replaced with the meter number.

  Note: For example, an address is "1N34". An alarm received from device address "1034" will be assigned to Meter 1. An alarm received from device address "1134" 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.

Adding a New Alarm

1. To start, click **Add**.
2. Next, specify the new alarm's Address, Type, and State.



Address:

Type: Differential Pressure

State: Off

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

EFM Event Mapping

EFM 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, a Configuration Mapping with address "7N48" is set to Pipe Diameter. An event will be generated from address 7148 when the user changes the Pipe Diameter in the device for Meter 1. The event will be converted to an audit event for the Pipe Diameter 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 Alarms & Events" setting located in the EFM Meters dialog.

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**. Then, click **Export**.
3. Save the file to an accessible location.
4. 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 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.

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. **Note:** All possible attribute names are listed in the History dialog. For more information, refer to **EFM History Mapping**.
Value	This is the address syntax for the attribute. It can be blank, static, or take the form *S<N>*. **Note:** For information on each attribute's value limitations, refer to **EFM 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 *1N34.* For more information, refer to **EFM Alarm Mapping**.
Alarm Type	This is the type of the alarm. The default setting is Differential Pressure. The valid types are as follows: 1 = Differential Pressure 2 = Static Pressure

	3 = Temperature 4 = Cutoff 5 = Backflow 6 = Battery
Alarm State	This is the state of the alarm. The default setting is Off. 1 = Off 2 = On 3 = Hi 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 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 OMNI Flow Computer 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 Archive Number changes.
5. The Max History Archive Size changes.

See Also: EFM Meters

Data Types Descriptions

Data Type	Description
Boolean	Single bit
Word	Unsigned 16 bit value bit 0 is the low bit bit 15 is the high bit
Short	Signed 16 bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32 bit value bit 0 is the low bit bit 31 is the high bit
Long	Signed 32 bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
BCD	Two byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range.
LBCD	Four byte packed BCD Value range is 0-99999999. Behavior is undefined for values beyond this range.
String	Null terminated ASCII string Supports 8 byte and 16 byte string data.
Float*	32 bit floating point value 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.
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.

Address Descriptions

The default data types are shown in **bold**.

Address	Range	Data Type	Access
Digital I/O Point	1001-1024	**Boolean**	Read/Write
Programmable Boolean Point	1025-1088	**Boolean**	Read/Write
Programmable Accumulator Points	1089-1099	**Boolean**	Read/Write
Meter Run Status and Alarm Points	1n01-1n99 n = Number of Meter Run	**Boolean**	Read/Write
User Scratch Pad Boolean Points	1501-1599 1601-1649	**Boolean**	Read/Write
User Scratch Pad One Shot Points	1650-1699	**Boolean**	Read/Write
Command Boolean Points/Variables	1700-1797	**Boolean**	Read/Write
Meter Station Alarm and Status Points	1801-1898	**Boolean**	Read/Write
Prover Alarm and Status Points	1909-1967	**Boolean**	Read/Write
Meter Totalizer Roll-Over Flags	2n01-2n84 n = Number of Meter Run	**Boolean**	Read/Write
Misc. Meter Station Alarm and Status	2601-2660	**Boolean**	Read/Write
Boolean Status Points for Meter Tube Switching	2717-2732	**Boolean**	Read/Write
Archive Trigger Commands	2733-2774	**Boolean**	Read/Write
Station Totalizer Roll-over Flags	2801-2851 2865-2876	**Boolean**	Read/Write
Station Totalizer Decimal Resolution Flags	2858-2861	**Boolean**	Read/Write
Status Booleans Relating to Redundant Flow Computer Systems	2863-2864	**Boolean**	Read/Write
Boolean Command Outputs and Status Points used for Meter Tube Switching	2877-2896	**Boolean**	Read/Write

16 Bit Integer Data Addresses

These addresses support bit level access. For more information, refer to "Zero vs One Based Bit Addressing Within Registers" in **Data Access**.

Address	Range	Data Type	Access
Custom Data Packet #1	3001-3040	**Short**, Word, BCD	Read/Write
Custom Data Packet #2	3041-3056	**Short**, Word, BCD	Read/Write
Custom Data Packet #3	3057-3096	**Short**, Word, BCD	Read/Write
Misc. 16 bit Integer Data	3097-3099 3737-3752 3880-3899	**Short**, Word, BCD	Read/Write
Meter Run 16 Bit Integer Data	3n01-3n99 n = Number of Meter Run	**Short**, Word, BCD	Read/Write
Scratch Pad 16 Bit Integer Data	3501-3599	**Short**, Word, BCD	Read/Write
User Display #1	3601-3608	**Short**, Word, BCD	Read/Write
User Display #2	3609-3616	**Short**, Word, BCD	Read/Write
User Display #3	3617-3624	**Short**, Word, BCD	Read/Write
User Display #4	3625-3632	**Short**, Word, BCD	Read/Write
User Display #5	3633-3640	**Short**, Word, BCD	Read/Write
User Display #6	3641-3648	**Short**, Word, BCD	Read/Write
User Display #7	3649-3656	**Short**, Word, BCD	Read/Write
User Display #8	3657-3664	**Short**, Word, BCD	Read/Write
Access Raw Data Archive Records	3701-3736	**Short**, Word, BCD	Read/Write
Honeywell Multivariable 16 Bit Integer Data	3753-3793	**Short**, Word, BCD	Read/Write
Meter Station 16 Bit Integer Data	3800-3842	**Short**, Word, BCD	Read/Write

Danalyzer Gas Chromatograph Data	3843-3864	**Short**, Word, BCD	Read/Write
Flow Computer Time and Date Variables	3867-3879	**Short**, Word, BCD	Read/Write
Prover 16 Bit Integer Data	3901-3944	**Short**, Word, BCD	Read/Write

8 Character ASCII String Data

Address	Range	Data Type	Access
Meter Run ASCII Data	4n01-4n43 n = Number of Meter Run	**String**	Read/Write
Scatch Pad ASCII Data	4501-4599	**String**	Read/Write
User Display Definition Variables	4601-4640	**String**	Read/Write
Station Auxiliary Input Variables	4707-4714	**String**	Read/Write
Meter Station ASCII Data	4801-4850	**String**	Read/Write
Prover ASCII String Data	4901-4942	**String**	Read/Write

32 Bit Integer Data

These addresses support bit level access. For more information, refer to "Zero vs One Based Bit Addressing Within Registers" in **Data Access**.

Address	Range	Data Type	Access
Meter Run 32 Bit Integer Data	5n01-5n99 n = Number of Meter Run	**Long**, DWord, LBCD, Float	Read/Write
Scratch Pad 32 Bit Integer Data	5501-5599	**Long**, DWord, LBCD, Float	Read/Write
Station 32 Bit Integer Data	5801-5891	**Long**, DWord, LBCD, Float	Read/Write
Prover 32 Bit Integer Data	5930-5994	**Long**, DWord, LBCD, Float	Read/Write
Meter Run Premium Level 32 Bit Integer Data	6n01-6n99	**Long**, DWord, LBCD, Float	Read/Write
Station Premium Level 32 Bit Integer Data	6801-6855	**Long**, DWord, LBCD, Float	Read/Write

32 Bit IEEE Floating Point Data

These addresses support bit level access. For more information, refer to "Zero vs One Based Bit Addressing Within Registers" in **Data Access**.

Address	Range	Data Type	Access
Digital to Analog Outputs	7001-7024	**Float**, Long, DWord, LBCD	Read/Write
User Variables	7025-7088	**Float**, Long, DWord, LBCD	Read/Write
Programmable Accumulator	7089-7099	**Float**, Long, DWord, LBCD	Read/Write
Meter Run Data	7n01 - 7n99 n = Number of Meter Run	**Float**, Long, DWord, LBCD	Read/Write
Scatch Pad Data	7501-7599	**Float**, Long, DWord, LBCD	Read/Write
PID Control Data	7601-7623	**Float**, Long, DWord, LBCD	Read/Write
Miscellaneous Meter Run Data	7624-7683	**Float**, Long, DWord, LBCD	Read/Write
Miscellaneous Variables	7701-7778	**Float**, Long, DWord, LBCD	Read/Write
Meter Station Data	7801-7899	**Float**, Long, DWord, LBCD	Read/Write
Prover Data	7908-8199	**Float**, Long, DWord, LBCD	Read/Write
Miscellaneous Meter Run #1	8501-8599	**Float**, Long, DWord, LBCD	Read/Write
Miscellaneous Meter Run #2	8601-8699	**Float**, Long, DWord, LBCD	Read/Write
Miscellaneous Meter Run #3	8701-8799	**Float**, Long, DWord, LBCD	Read/Write
Miscellaneous Meter Run #4	8801-8899	**Float**, Long, DWord, LBCD	Read/Write
Station Previous Batch Average Data	8949-8989	**Float**, Long, DWord, LBCD	Read/Write

16 Bit Integer Configuration Data

These addresses support bit level access. For more information, refer to "Zero vs One Based Bit Addressing Within Registers" in **Data Access**.

Address	Range	Data Type	Access
Meter Run #1	13001-13013	**Short**, Word, BCD	Read/Write
Meter Run #2	13014-13026	**Short**, Word, BCD	Read/Write
Meter Run #3	13027-13039	**Short**, Word, BCD	Read/Write
Meter Run #4	13040-13052	**Short**, Word, BCD	Read/Write
Meter Run Configuration 16 Bit Integer Data	13053-13073	**Short**, Word, BCD	Read/Write
General Flow Configuration	13074-13084	**Short**, Word, BCD	Read/Write
Serial Port Configuration	13085-13128	**Short**, Word, BCD	Read/Write
PID Configuration	13129-13160	**Short**, Word, BCD	Read/Write
PLC Data	13161-13299	**Short**, Word, BCD	Read/Write
Peer to Peer Setup	13300-13477	**Short**, Word, BCD	Read/Write
Serial Ports #5 and #6 16 Bit Integer Data	13478-13499	**Short**, Word, BCD	Read/Write
Raw Data Archive	13500-13999	**Short**, Word, BCD	Read/Write

16 Character ASCII String Data

Address	Range	Data Type	Access
Flow Computer Configuration	14001-14499	**String**	Read/Write

32 Bit Integer Data

These addresses support bit level access. For more information, refer to "Zero vs One Based Bit Addressing Within Registers" in **Data Access**.

Address	Range	Data Type	Access
Flow Computer Configuration	15001-16999	**Long**, DWord, LBCD, Float	Read/Write

32 Bit IEEE Floating Point Data

These addresses support bit level access. For more information, refer to "Zero vs One Based Bit Addressing Within Registers" in **Data Access**.

Address	Range	Data Type	Access
Flow Computer Configuration	17001-19999	**Float**, Long, DWord, LBCD	Read/Write

Array Support

Arrays are supported for register locations for all data types except for strings. Arrays are also supported for input and output coils (Boolean data types). There are two methods of addressing an array. The following examples use register locations:

3*xxx* [rows] [cols]
3*xxx* [cols] this method assumes rows is equal to one.

For arrays, rows multiplied by cols cannot exceed the maximum number of registers or Booleans (depending on the data type) that can be read by the protocol in a single access.

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>'
Array support is not available for the specified address: '<address>'
Data Type '<type>' is not valid for device address '<address>'
Device address '<address>' contains a syntax error
Device address '<address>' is not supported by model '<model name>'
Device address '<address>' is Read Only
Missing address
Received block length of '<received length>' does not match expected length of '<expected length>' for address '<address>' on device '<device>'

Device Status Messages

Device '<device name>' is not responding
Unable to write to address '<address>' on device '<device>': Device responded with exception code '<code>'
Unable to write to '<address>' on device '<device name>'
Write failed for '<tag name>' on device '<device name>'. Maximum path length of '<number>' characters exceeded

OMNI Flow Computer Error Messages

'<device name>' - Alarm record parse for device failed
'<device name>' - Archive is not configured correctly for address '<address>'. Max number of records is zero
'<device name>' - Archive record contains an invalid address. BOOL, 8-byte strings, and 16-byte strings are not supported
'<device name>' - Archive record for address '<address>' contains an unexpected number of bytes. Expected '<number of bytes>' bytes, received '<number of bytes>' bytes
'<device name>' - Config data attribute for meter tap location read from device address '<address>' is '<value>', and does not map to any valid meter tap locations. Expecting 0 for flange, or 1 for pipe. Defaulting to flange
'<device name>' - Config data attribute for meter type read from device address '<address>' is '<value>', and does not map to any valid meter types. Expected values are 0, 2, or 3 for orifice, 1 for turbine, 4 or 8 for ultra sonic, and 5 for vcone. Defaulting to orifice
'<device name>' - Config data attribute for static pressure tap read from device address '<address>' is '<value>', and does not map to any valid static pressure tap locations. Expecting 0 for up, or 1 for down. Defaulting to up
'<device name>' - Date format for address '<address>' is invalid. Device returned '<value>', valid values are 0 or 1
'<device name>' - Device password write not successful. Value in response is different from the written value
'<device name>' - Event record parse for device failed
'<device name>' - Meter and shared archives are not in sync. Records will only contain flow data (no analysis)
'<device name>' - Meter archive record parse failed
'<device name>' - Shared archive record parse failed
'<device name>' - The history mapping contains more configured attributes than the device. Some historical attributes will not contain valid data
'<device name>' - The max alarm archive size was changed from '<previous size>' to '<current size>'
'<device name>' - The max event archive size was changed from '<previous size>' to '<current size>'
'<device name>' - Time sync write not successful. Value in response is different from the written value
'<device name>' - Unable to read '<number of registers>' registers in config register block at address '<address>'
'<device name>' - Unable to read date format register for address '<address>'. Response is not the correct size
'<device name>' - Unable to read index registers. Response is not the correct size

'<device name>' - Unable to read record format registers for address '<address>'. Response is not the correct size
'<device name>' - Unable to write requested record register for address '<address>'. Response is not the correct size
'<device name>' - Unable to write requested record register for address '<address>'. Wrote '<value>', read back '<value>'
Bad address in block [<start address> to <end address>] on device '<device name>'
Bad array spanning [<address> to <address>] on device '<device>'
CSV alarm mapping for address '<address>' is invalid and will be removed
CSV history mapping for attribute '<attribute>' is invalid and will be removed
Device password invalid for device '<device name>'
Device password write for device '<device name>' was successful
Unable to read '<address>' from device '<device name>'. The device is configured for broadcast writes only
Warning loading '<mapping type>' mapping from CSV. '<warning type>'

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

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 out-

		put 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.

Address Validation

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>'
Array support is not available for the specified address: '<address>'
Data Type '<type>' is not valid for device address '<address>'
Device address '<address>' contains a syntax error
Device address '<address>' is not supported by model '<model name>'
Device address '<address>' is Read Only
Missing address
Received block length of '<received length>' does not match expected length of '<expected length>' for address '<address>' on device '<device>'

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.

Array support is not available for the specified address: '<address>'

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically contains an array reference for an address type that doesn't support arrays.

Solution:
Re-enter the address in the client application to remove the array reference or correct the address type.

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.

Device address '<address>' is not supported by model '<model name>'

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically references a location that is valid for the communications protocol but not supported by the target device.

Solution:
Verify that the address is correct; if it is not, re-enter it in the client application. Also verify that the selected model name for the device is correct.

Device address '<address>' is Read Only

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically has a requested access mode that is not compatible with what the device supports for that address.

Solution:
Change the access mode in the client application.

Missing address

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically has no length.

Solution:
Re-enter the address in the client application.

Received block length of '<received length>' does not match expected length of '<expected length>' for address '<address>' on device '<device>'

Error Type:
Warning

Possible Cause:
The driver attempted to read a block of memory but the PLC did not provide the driver with the requested size of data. No error code was returned.

Solution:
N/A

Device Status Messages

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

Device Status Messages

Device '<device name>' is not responding
Unable to write to address '<address>' on device '<device>': Device responded with exception code '<code>'
Unable to write to '<address>' on device '<device name>'
Write failed for '<tag name>' on device '<device name>'. Maximum path length of '<number>' characters exceeded

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.

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

Error Type:
Warning

Possible Cause:
See **Modbus Exception Codes** for a description of the exception code.

Solution:
See **Modbus Exception Codes**.

Write failed for '<tag name>' on device '<device name>'. Maximum path length of '<number>' exceeded

Error Type:
Warning

Possible Cause:

Path length is limited to the indicated number of characters.

Solution:
Devise a shorter path.

OMNI Flow Computer Specific Messages

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

OMNI Flow Computer Specific Messages

'<device name>' - Alarm record parse for device failed
'<device name>' - Archive is not configured correctly for address '<address>'. Max number of records is zero
'<device name>' - Archive record contains an invalid address. BOOL, 8-byte strings, and 16-byte strings are not supported
'<device name>' - Archive record for address '<address>' contains an unexpected number of bytes. Expected '<number of bytes>' bytes, received '<number of bytes>' bytes
'<device name>' - Config data attribute for meter tap location read from device address '<address>' is '<value>', and does not map to any valid meter tap locations. Expecting 0 for flange, or 1 for pipe. Defaulting to flange
'<device name>' - Config data attribute for meter type read from device address '<address>' is '<value>', and does not map to any valid meter types. Expected values are 0, 2, or 3 for orifice, 1 for turbine, 4 or 8 for ultra sonic, and 5 for vcone. Defaulting to orifice
'<device name>' - Config data attribute for static pressure tap read from device address '<address>' is '<value>', and does not map to any valid static pressure tap locations. Expecting 0 for up, or 1 for down. Defaulting to up
'<device name>' - Date format for address '<address>' is invalid. Device returned '<value>', valid values are 0 or 1
'<device name>' - Device password write not successful. Value in response is different from the written value
'<device name>' - Event record parse for device failed
'<device name>' - Meter and shared archives are not in sync. Records will only contain flow data (no analysis)
'<device name>' - Meter archive record parse failed
'<device name>' - Shared archive record parse failed
'<device name>' - The history mapping contains more configured attributes than the device. Some historical attributes will not contain valid data
'<device name>' - The max alarm archive size was changed from '<previous size>' to '<current size>'
'<device name>' - The max event archive size was changed from '<previous size>' to '<current size>'
'<device name>' - Time sync write not successful. Value in response is different from the written value
'<device name>' - Unable to read '<number of registers>' registers in config register block at address '<address>'
'<device name>' - Unable to read date format register for address '<address>'. Response is not the correct size
'<device name>' - Unable to read index registers. Response is not the correct size
'<device name>' - Unable to read record format registers for address '<address>'. Response is not the correct size
'<device name>' - Unable to write requested record register for address '<address>'. Response is not the correct size
'<device name>' - Unable to write requested record register for address '<address>'. Wrote '<value>', read back '<value>'
Bad address in block [<start address> to <end address>] on device '<device name>'
Bad array spanning [<address> to <address>] on device '<device>'
CSV alarm mapping for address '<address>' is invalid and will be removed
CSV history mapping for attribute '<attribute>' is invalid and will be removed
Device password invalid for device '<device name>'
Device password write for device '<device name>' was successful
Unable to read '<address>' from device '<device name>'. The device is configured for broadcast writes only
Warning loading '<mapping type>' mapping from CSV. '<warning type>'

'<device name>' - Alarm record parse for device failed

Error Type:
Warning

Possible Cause:
1. A failure occurred when parsing the alarm record for EFM attribute data.
2. The device's archive memory may have been cleared since the last upload.

Solution:
1. Ensure that the time and date have been set correctly.
2. Verify the cabling between the PC and the PLC device.
3. Verify that the specified communications parameters match those of the device.

'<device name>' - Archive is not configured correctly for address '<address>'. Max number of records is zero

Error Type:
Warning

Possible Cause:
The meter's archive configuration is invalid.

Solution:
Verify that the configuration of the archive's group and max records are correct.

'<device name>' - Archive record contains an invalid address. BOOL, 8-byte strings, and 16-byte strings are not supported

Error Type:
Warning

Possible Cause:
The meter's archive group configuration is invalid.

Solution:
Ensure that the archive's group configuration does not contain any addresses that correspond to Boolean or String data.

'<device name>' - Archive record for address '<address>' contains an unexpected number of bytes. Expected '<number of bytes>' bytes, received '<number of bytes>' bytes

Error Type:
Warning

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.

'<device name>' - Config data attribute for meter tap location read from device address '<address>' is '<value>', and does not map to any valid meter tap locations. Expecting 0 for flange, or 1 for pipe. Defaulting to flange

Error Type:
Warning

Possible Cause:
The meter tap location register contains an invalid value.

Solution:
Set the value to one of the expected values from the error string.

'<device name>' - Config data attribute for meter type read from device address '<address>' is '<value>', and does not map to any valid meter types. Expected values are 0, 2, or 3 for orifice, 1 for turbine, 4 or 8 for ultra sonic, and 5 for vcone. Defaulting to orifice

Error Type:
Warning

Possible Cause:
The meter type register contains an invalid value.

Solution:
Set the value to one of the expected values from the error string.

'<device name>' - Config data attribute for static pressure tap read from device address '<address>' is '<value>', and does not map to any valid static pressure tap locations. Expecting 0 for up, or 1 for down. Defaulting to up

Error Type:
Warning

Possible Cause:
The static pressure tap location register contains an invalid value.

Solution:
Set the value to one of the expected values from the error string.

'<device name>' - Date format for address '<address>' is invalid. Device returned '<value>', valid values are 0 or 1

Error Type:
Warning

Possible Cause:
The date format register contains an invalid value.

Solution:
Set the value to one of the expected values from the error string.

'<device name>' - Device password write not successful. Value in response is different from the written value

Error Type:
Warning

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.

'<device name>' - Event record parse for device failed

Error Type:
Warning

Possible Cause:
1. A failure occurred when parsing the event record for EFM attribute data.

2. The device's archive memory may have been cleared since the last upload.

Solution:

1. Ensure that the time and date have been set correctly.
2. Verify the cabling between the PC and the PLC device.
3. Verify that the specified communications parameters match those of the device.

'<device name>' - Meter and shared archives are not in sync. Records will only contain flow data (no analysis)

Error Type:

Warning

Possible Cause:

1. The number of records in the meter and shared archives are not equal.
2. The timestamps of the records located at the same index are not equal.

Solution:

1. Ensure that the shared archive and all meter archives are configured to use the same trigger.
2. Ensure that the values in the Current Record Number registers for all meter archives and the shared archive are the same.

'<device name>' - Meter archive record parse failed

Error Type:

Warning

Possible Cause:

1. A failure occurred when parsing the meter history record for EFM attribute data.
2. The device's archive memory may have been cleared since the last upload.

Solution:

1. Verify that the meter's archive configuration is correct, and that it matches the History Mapping.
2. Ensure that the time and date have been set correctly.

'<device name>' - Shared archive record parse failed

Error Type:

Warning

Possible Cause:

1. A failure occurred when parsing the shared data history record for EFM attribute data.
2. The device's archive memory may have been cleared since the last upload.

Solution:

1. Verify that the shared data archive's configuration is correct, and that it matches the Gas Quality mapping.
2. Ensure that the time and date have been set correctly.

'<device name>' - The history mapping contains more configured attributes than the device. Some historical attributes will not contain valid data

Error Type:

Warning

Possible Cause:

The number of registers configured in an archive's group configuration does not equal the number of attributes with a configured index in the History Mapping.

Solution:

Verify that the archive's group configuration matches the History Mapping configuration.

'<device name>' - The max alarm archive size was changed from '<previous size>' to '<current size>'

Error Type:

Warning

Possible Cause:
The size of the alarm archive in the device has been changed since the last alarm archive upload.

Solution:
Change the alarm archive size back to its default value for the Firmware version that is being used.

Note:
Data may be lost if the archive size is not set to the default value.

'<device name>' - The max event archive size was changed from '<previous size>' to '<current size>'

Error Type:
Warning

Possible Cause:
The size of the event archive in the device has been changed since the last event archive upload.

Solution:
Change the event archive size back to its default value for the Firmware version that is being used.

Note:
Data may be lost if the archive size is not set to the default value.

'<device name>' - Time sync write not successful. Value in response is different from the written value

Error Type:
Warning

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.

'<device name>' - Unable to read '<number of registers>' registers in config register block at address '<address>'

Error Type:
Warning

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 that the Firmware version running on the device is supported.
2. Verify the cabling between the PC and the PLC device.
3. Verify that the specified communications parameters match those of the device.

'<device name>' - Unable to read date format register for address '<address>'. Response is not the correct size

Error Type:
Warning

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.

'<device name>' - Unable to read index registers. Response is not the correct size

Error Type:
Warning

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.

'<device name>' - Unable to read record format registers for address '<address>'. Response is not the correct size

Error Type:
Warning

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.

'<device name>' - Unable to write requested record register for address '<address>'. Wrote '<value>', read back '<value>'

Error Type:
Warning

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.

'<device name>' - Unable to write requested record register for address '<address>'. Response is not the correct size

Error Type:
Warning

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.

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:

Framing & 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 the number of addresses that can be read by the protocol in a single access. For example, set the value to 125 for 16 bit registers.

See Also:

Framing & Error Handling
Block Sizes

CSV alarm mapping for address '<address>' is invalid and will be removed

Error Type:

Warning

Possible Cause:

The address in the error message is invalid.

Solution:

Change the address to a valid one.

Note:

The address is located in the Alarm Mapping section of the CSV file.

CSV history mapping for attribute '<attribute>' is invalid and will be removed

Error Type:

Warning

Possible Cause:

The address in the error message is invalid.

Solution:

Change the address to a valid one.

Note:

The address is located in the History Mapping section of the CSV file.

Device password invalid for device '<device name>'

Error Type:

Warning

Possible Cause:
The device responded with the "needs password" exception code, even though the configured password was written to the device.

Solution:
Ensure that the device password that is configured in the Data Access tab of Device Properties matches the password that is configured on the device.

Device password write for device '<device name>' was successful

Error Type:
Information

Possible Cause:
After receiving the "Device password invalid for device '<device name>'" error message, a valid password was written to the device.

Solution:
N/A.

Note:
This message is logged to indicate that the device's password requirements have been met.

See Also:
Device password invalid for device '<device name>'

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.

Warning loading '<mapping type>' mapping from CSV. '<warning type>'

Error Type:
Information

Possible Cause:
A new EFM mapping was imported from the CSV file.

Solution:
N/A.

Serial Communications

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

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

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.

Verify the serial parameters and make any necessary changes.

Index