

EFM Simulator Driver

© 2016 PTC Inc. All Rights Reserved.

Table of Contents

EFM Simulator Driver	1
Table of Contents	2
EFM Simulator Driver	3
Overview	3
Setup	4
Channel Properties - General	4
Channel Properties - Write Optimizations	5
Channel Properties - Advanced	6
Device Properties - General	6
Device Properties - Scan Mode	8
Device Properties - Time Synchronization	9
Device Properties - Simulated Meter Data	9
EFM Meter Properties	10
Using the EFM Simulator Driver	12
Liquid Product Data	13
Data Types Description	14
Address Descriptions	15
Error Descriptions	16
Address <address> is out of range for the specified device or register.	17
Data type <type> is not valid for device address <address>.	17
Device address <address> contains a syntax error.	17
Device address <address> is read only.	17
Missing Address.	17
Index	19

EFM Simulator Driver

Help version 1.013

CONTENTS

[Overview](#)

What is the EFM Simulator Driver?

[Device Setup](#)

How do I configure a device for use with this driver?

[Using the EFM Simulator Driver](#)

How do I configure and utilize an EFM Simulator Driver project?

[Data Types Description](#)

What data types can be used with an EFM simulated device?

[Address Descriptions](#)

How are addresses specified on an EFM simulated device?

[Error Descriptions](#)

What error messages does the EFM Simulator Driver produce?

Overview

The EFM Simulator Driver provides a reliable way to simulate EFM data for testing the EFM Exporter Plug-In without an external device.

Setup

Supported EFM Output Types

Alarms
Configuration
Events
History

Supported Device Models

Gas EFM
Liquid EFM

Maximum Number of Channels and Devices

The maximum number of supported channels is 100. The maximum number of devices supported per channel is 999.

Device ID

The EFM Simulator Driver supports decimal Device IDs in the range of 1 to 999. Unique Device IDs are required when creating multiple devices on the same channel. The default setting is 1.

Channel Properties - General

This server supports the use of simultaneous multiple communications drivers. Each protocol or driver used in a server project is called a channel. A server project may consist of many channels with the same communications driver or with unique communications drivers. A channel acts as the basic building block of an OPC link. This group is used to specify general channel properties, such as the identification attributes and operating mode.

Property Groups	<input type="checkbox"/> Identification	
General	Name	
Ethernet Communications	Description	
Write Optimizations	Driver	
Advanced	<input type="checkbox"/> Diagnostics	
	Diagnostics Capture	Disable

Identification

Name: User-defined identity of this channel. In each server project, each channel name must be unique. Although names can be up to 256 characters, some client applications have a limited display window when browsing the OPC server's tag space. The channel name is part of the OPC browser information.

● For information on reserved characters, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in the server help.

Description: User-defined information about this channel.

● Many of these properties, including Description, have an associated system tag.

Driver: Selected protocol / driver for this channel. Specify the device driver that was selected during channel creation. It is a disabled setting in the channel properties.

● **Note:** With the server's online full-time operation, these properties can be changed at any time. This includes changing the channel name to prevent clients from registering data with the server. If a client has already acquired an item from the server before the channel name is changed, the items are unaffected. If,

after the channel name has been changed, the client application releases the item and attempts to re-acquire using the old channel name, the item is not accepted. With this in mind, changes to the properties should not be made once a large client application has been developed. Utilize the User Manager to prevent operators from changing properties and restrict access rights to server features.

Diagnostics

Diagnostics Capture: When enabled, this option makes the channel's diagnostic information available to OPC applications. Because the server's diagnostic features require a minimal amount of overhead processing, it is recommended that they be utilized when needed and disabled when not. The default is disabled.

● *For more information, refer to "Communication Diagnostics" in the server help.*

● **Note:** Not all drivers support diagnostics. To determine whether diagnostics are available for a particular driver, open the driver information and locate the "Supports device level diagnostics" statement.

Channel Properties - Write Optimizations

As with any OPC server, writing data to the device may be the application's most important aspect. The server intends to ensure that the data written from the client application gets to the device on time. Given this goal, the server provides optimization properties that can be used to meet specific needs or improve application responsiveness.

Property Groups	[-] Write Optimizations	
General	Optimization Method	Write Only Latest Value for All Tags
Ethernet Communications	Duty Cycle	10
Write Optimizations		

Write Optimizations

Optimization Method: controls how write data is passed to the underlying communications driver. The options are:

- **Write All Values for All Tags:** This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing data to the device as quickly as possible. This mode ensures that everything written from the client applications is sent to the target device. This mode should be selected if the write operation order or the write item's content must uniquely be seen at the target device.
- **Write Only Latest Value for Non-Boolean Tags:** Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the server's internal write queue and sent to the device at the next possible opportunity. This can greatly improve the application performance.
 - **Note:** This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.

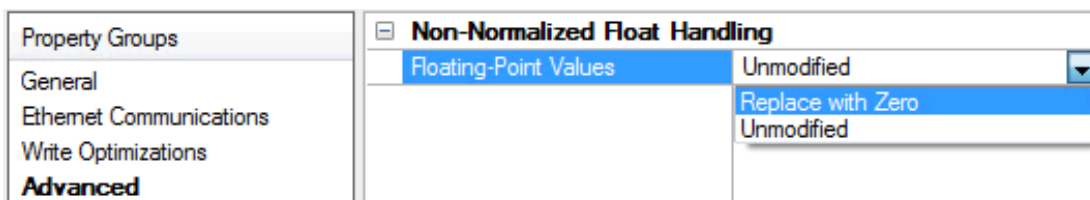
- **Write Only Latest Value for All Tags:** This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write queue before they are sent. This is the default mode.

Duty Cycle: is used to control the ratio of write to read operations. The ratio is always based on one read for every one to ten writes. The duty cycle is set to ten by default, meaning that ten writes occur for each read operation. Although the application is performing a large number of continuous writes, it must be ensured that read data is still given time to process. A setting of one results in one read operation for every write operation. If there are no write operations to perform, reads are processed continuously. This allows optimization for applications with continuous writes versus a more balanced back and forth data flow.

● **Note:** It is recommended that the application be characterized for compatibility with the write optimization enhancements before being used in a production environment.

Channel Properties - Advanced

This group is used to specify advanced channel properties. Not all drivers support all properties; so the Advanced group does not appear for those devices.



Non-Normalized Float Handling: Non-normalized float handling allows users to specify how a driver handles non-normalized IEEE-754 floating point data. A non-normalized value is defined as Infinity, Not-a-Number (NaN), or as a Denormalized Number. The default is Replace with Zero. Drivers that have native float handling may default to Unmodified. Descriptions of the options are as follows:

- **Replace with Zero:** This option allows a driver to replace non-normalized IEEE-754 floating point values with zero before being transferred to clients.
- **Unmodified:** This option allows a driver to transfer IEEE-754 denormalized, normalized, non-number, and infinity values to clients without any conversion or changes.

● **Note:** This property is disabled if the driver does not support floating point values or if it only supports the option that is displayed. According to the channel's float normalization setting, only real-time driver tags (such as values and arrays) are subject to float normalization. For example, EFM data is not affected by this setting.lin

● *For more information on the floating point values, refer to "How To ... Work with Non-Normalized Floating Point Values" in the server help.*

Device Properties - General

A device represents a single target on a communications channel. If the driver supports multiple controllers, users must enter a device ID for each controller.

Property Groups	Identification	
General	Name	
Scan Mode	Description	
Ethernet Encapsulation	Channel Assignment	
Timing	Driver	
Auto-Demotion	Model	
Redundancy	ID Format	Decimal
	ID	2
	Operating Mode	
	Data Collection	Enable
	Simulated	No

Identification

Name: Specify the name of the device. It is a logical user-defined name that can be up to 256 characters long, and may be used on multiple channels.

- **Note:** Although descriptive names are generally a good idea, some OPC client applications may have a limited display window when browsing the OPC server's tag space. The device name and channel name become part of the browse tree information as well. Within an OPC client, the combination of channel name and device name would appear as "ChannelName.DeviceName".

- *For more information, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in server help.*

Description: User-defined information about this device.

- Many of these properties, including Description, have an associated system tag.

Channel Assignment: User-defined name of the channel to which this device currently belongs.

Driver: Selected protocol driver for this device.

Model: Specify the specific type of device that is associated with this ID. The contents of the drop-down menu depends on the type of communications driver being used. Models that are not supported by a driver are disabled. If the communications driver supports multiple device models, the model selection can only be changed when there are no client applications connected to the device.

- **Note:** If the communication driver supports multiple models, users should try to match the model selection to the physical device. If the device is not represented in the drop-down menu, select a model that conforms closest to the target device. Some drivers support a model selection called "Open," which allows users to communicate without knowing the specific details of the target device. For more information, refer to the driver help documentation.

ID: Specify the device's driver-specific station or node. The type of ID entered depends on the communications driver being used. For many communication drivers, the ID is a numeric value. Drivers that support a Numeric ID provide users with the option to enter a numeric value whose format can be changed to suit the needs of the application or the characteristics of the selected communications driver. The ID format can be Decimal, Octal, and Hexadecimal.

- **Note:** If the driver is Ethernet-based or supports an unconventional station or node name, the device's TCP/IP address may be used as the device ID. TCP/IP addresses consist of four values that are separated by periods, with each value in the range of 0 to 255. Some device IDs are string based. There may be additional

properties to configure within the ID field, depending on the driver. For more information, refer to the driver's help documentation.

Operating Mode

Data Collection: This property controls the device's active state. Although device communications are enabled by default, this property can be used to disable a physical device. Communications are not attempted when a device is disabled. From a client standpoint, the data is marked as invalid and write operations are not accepted. This property can be changed at any time through this property or the device system tags.

Simulated: This option places the device into Simulation Mode. In this mode, the driver does not attempt to communicate with the physical device, but the server continues to return valid OPC data. Simulated stops physical communications with the device, but allows OPC data to be returned to the OPC client as valid data. While in Simulation Mode, the server treats all device data as reflective: whatever is written to the simulated device is read back and each OPC item is treated individually. The item's memory map is based on the group Update Rate. The data is not saved if the server removes the item (such as when the server is reinitialized). The default is No.

● **Notes:**

1. This System tag (_Simulated) is read only and cannot be written to for runtime protection. The System tag allows this property to be monitored from the client.
2. In Simulation mode, the item's memory map is based on client update rate(s) (Group Update Rate for OPC clients or Scan Rate for native and DDE interfaces). This means that two clients that reference the same item with different update rates return different data.

● Simulation Mode is for test and simulation purposes only. It should never be used in a production environment.

Device Properties - Scan Mode

The Scan Mode specifies the subscribed-client requested scan rate for tags that require device communications. Synchronous and asynchronous device reads and writes are processed as soon as possible; unaffected by the Scan Mode properties.

Property Groups	[-] Scan Mode	
General	Scan Mode	Respect Client-Specified Scan Rate ▼
Scan Mode	Initial Updates from Cache	Disable

Scan Mode: specifies how tags in the device are scanned for updates sent to subscribed clients.

Descriptions of the options are:

- **Respect Client-Specified Scan Rate:** This mode uses the scan rate requested by the client.
- **Request Data No Faster than Scan Rate:** This mode specifies the maximum scan rate to be used. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
 - **Note:** When the server has an active client and items for the device and the scan rate value is increased, the changes take effect immediately. When the scan rate value is decreased, the changes do not take effect until all client applications have been disconnected.
- **Request All Data at Scan Rate:** This mode forces tags to be scanned at the specified rate for subscribed clients. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.

- **Do Not Scan, Demand Poll Only:** This mode does not periodically poll tags that belong to the device nor perform a read to get an item's initial value once it becomes active. It is the client's responsibility to poll for updates, either by writing to the `_DemandPoll` tag or by issuing explicit device reads for individual items. *For more information, refer to "Device Demand Poll" in server help.*
- **Respect Tag-Specified Scan Rate:** This mode forces static tags to be scanned at the rate specified in their static configuration tag properties. Dynamic tags are scanned at the client-specified scan rate.

Initial Updates from Cache: When enabled, this option allows the server to provide the first updates for newly activated tag references from stored (cached) data. Cache updates can only be provided when the new item reference shares the same address, scan rate, data type, client access, and scaling properties. A device read is used for the initial update for the first client reference only. The default is disabled; any time a client activates a tag reference the server attempts to read the initial value from the device.

Device Properties - Time Synchronization

This group is used to specify the device's time zone and time synchronization properties. It primarily applies to time stamped data or information from battery-powered devices at remote locations where the device time may deviate (causing issues with the time-stamped data). To prevent this problem from occurring, users can specify that the server synchronize the device time.

Property Groups	<input checked="" type="checkbox"/> Time Zone	
General	Time Zone	(UTC-05:00) Eastern Time (US & Canada)
Scan Mode	Respect Daylight Saving Time	No
Timing	<input checked="" type="checkbox"/> Synchronization	
Auto-Demotion	Time Sync Method	Absolute
Time Synchronization	Sync Absolute	12:00:00 AM

- **Note:** Not all drivers and models support all options.

Time Zone: Specify the device's time zone. To ignore the time zone, select one of the first four options in the list (which do not have an offset). The default is the time zone of the local system.

- **Note:** The driver uses this property both when synching the device time and when converting EFM timestamps from the device to UTC time.

Respect Daylight Saving Time: Select Yes to follow Daylight Saving Time offset when synching the device time. Select No to ignore Daylight Saving Time. The default is disabled / No.

Method: Specify the method of synchronization. Options include Disabled, Absolute, and Interval. The default is Disabled. Descriptions of the options are as follows:

- **Disabled:** No synchronization.
- **Absolute:** Synchronizes to an absolute time of day specified through the Time property (appears only when Absolute is selected).
- **Interval:** Synchronizes on startup and every number of minutes specified through the Sync Interval property (appears only when Interval is selected). The default is 60 minutes.

Device Properties - Simulated Meter Data

To access these properties, right-click on the device and click **Properties | Simulated Meter Data**.

Property Groups	<input type="checkbox"/> Test Data	
General	Daily Records Per File	100
Scan Mode	Records Per File	100
Time Synchronization	Record Interval (min)	60
Simulated Meter Data	Batch Records Per File	100
	Batch Interval (min)	60
	Start Date/Time	10/12/2016 3:25:14 PM
	Simulate Configuration Upload Failure	Disable
	<input type="checkbox"/> Scheduling	
	Autogenerate Test Data	Disable
	Period	60
	Period Units	Minutes
	<input type="checkbox"/> Simulated Meter Data	
	File Path	C:\ProgramData\V6\EFM Simulator\
	Generate Test Data	Create

Descriptions of the properties are as follows:

- **Daily Records Per File:** Specify how many records will be included in the Daily History file. The valid range is 1 to 1000. The default setting is 100.
- **Records Per File:** Specify how many records will be included per file. The valid range is 1 to 1000. The default setting is 100.
- **Record Interval (min):** Specify the amount of time between each record (in minutes). The valid range is 1 to 50,400. The default setting is 60 minutes.
- **Batch Records Per File:** Specify how many batch records will be included per file. The valid range is 1 to 1000. The default setting is 100.
Note: This property is only supported by the Liquid EFM model.
- **Batch Interval (min):** Specify the amount of time between each record (in minutes). The valid range is 1 to 50,400. The default setting is 60 minutes.
Note: This property is only supported by the Liquid EFM model.
- **Start Date/Time:** Specify the date and time of the first generated record.
- **Simulate Config Upload Failure:** When enabled, this property will cause subsequent polls to simulate configuration file read errors. The default setting is Disable.
- **Autogenerate Test Data:** When enabled, this property specifies that test data will be automatically generated at the demand of the client. The default setting is Disable.
- **Period:** Specify the units for the period at which test data will be automatically generated. The valid range is 1 to 129,600 and may be specified in units of minutes, hours, or days. The default setting is 60.
- **Period Units:** Specify the units to use for the period at which test data will be automatically generated. The default setting is Minutes.
- **File Path:** This field specifies the location where the test data will be placed.
- **Generate Test Data:** When clicked, this button will generate test XML.
Note: For more information on the contents of the generated file, refer to the EFM Exporter Plug-In help file.

EFM Meter Properties

EFM Meters gather record line pressure, temperature, and differential pressure in pipes to determine total gas or liquid flow.

Note: The EFM Simulator Driver supports a maximum of four meters per device.

Property Groups	Identification	
General	Name	GasMeter1
	Description	
	Driver	EFM Simulator
	Enable	Yes
	EFM	
	Hourly History	Yes
	Daily History	Yes
	Alarms	Yes
	Events	Yes

Property Groups	Identification	
General	Name	LiquidMeter1
	Description	
	Driver	EFM Simulator
	Enable	Yes
	EFM	
	Hourly History	Yes
	Daily History	Yes
	Alarms	Yes
	Events	Yes
	Batch	Yes

Name: Specify the name of the meter. The default settings are *<device model><type>Meter1* through *<device model><type>Meter4*.

Description: This property allows the user to provide a brief summary for the meter. The default setting is empty.

Driver: Specify the driver for which the meter is configured. This property cannot be modified.

Enable: Specify whether the meter is enabled. The default setting is Yes.

Hourly History: Specify whether the meter's Hourly History output is enabled. The default setting is Yes.

Daily History: Specify whether the meter's Daily History output is enabled. The default setting is Yes.

Alarms: Specify whether the meter's Alarms output is enabled. The default setting is Yes.

Events: Specify whether the meter's Events output is enabled. The default setting is Yes.

Batch: Specify whether the meter's batch (group) output is enabled. The default setting is Yes. Batch is only available for liquid meters.

Using the EFM Simulator Driver

1. To start, create a channel using the EFM Simulator Driver. Select the default values, and then click **Finish**.
2. Next, create a device. Select the default values, and then click **Finish**.
3. Under the newly-created device right-click on a meter and then select **Properties**.

Property Groups	[-] Identification	
General	Name	GasMeter1
	Description	
	Driver	EFM Simulator
	Enable	Yes
	[-] EFM	
	Hourly History	Yes
	Daily History	Yes
	Alarms	Yes
	Events	Yes

4. The General property group displays the properties for the selected meter. Users can assign a unique name, turn the meter on or off, and specify the type of data that the meter will provide. To do so, simply click on a field and select the desired value from the drop-down menu located in the right-hand column.
5. Once finished, click **Apply**.
6. Next, right-click on the device and select **Properties**. Navigate to the **Simulated Meter Data** property group. This property group contains information about the data that users can simulate, including the amount, the time interval between data records, and when the data starts. The EFM Simulator Driver can also be configured to automatically generate new data every few minutes, hours, or days.

Property Groups	[-] Test Data	
General	Daily Records Per File	100
Scan Mode	Records Per File	100
Time Synchronization	Record Interval (min)	60
Simulated Meter Data	Batch Records Per File	100
	Batch Interval (min)	60
	Start Date/Time	10/12/2016 3:25:14 PM
	Simulate Configuration Upload Failure	Disable
	[-] Scheduling	
	Autogenerate Test Data	Disable
	Period	60
	Period Units	Minutes
	[-] Simulated Meter Data	
	File Path	C:\ProgramData\V6\EFM Simulator\
	Generate Test Data	Create

Note: Users also have the ability to simulate a configuration upload failure, which provides the opportunity to observe what will happen if the device configuration files are corrupt and/or missing.

7. To view the location of the stored simulated data, locate **File Path**.

Note: When configuring automatic test data generation, users must have a client connected in order for it to work. Data will not be created without a client. Users must also be aware of both how many records are being generated and the interval between records. Many files will be created if the default values are used while polling for data on a short interval. Although this depends on the size of the project, testing even with just a few meters can cause this to happen.

8. After all the parameters have been configured, click **Create** for the Generate Test Data property. A confirmation message will be posted to the Event Log if the generation was successful.

Note: This step is not necessary if the device is configured for automatic test data generation.

9. Once finished, select **Apply | OK**.

Liquid Product Data

The EFM Simulator Driver generates a Liquid Product Record for each of the following products:

- Light Sweet Crude
- Gasoline
- Fuel Oil
- Jet Fuel
- Natural Gas Liquids
- Cyclohexane
- Benzene
- Cumene
- Styrene
- Toluene

Only the liquid product name will be generated. When Liquid Periodic/Daily historical data is generated, records will be randomly associated with one liquid product. The liquid product name will be included in the Configuration Data. When Liquid Batch historical data is generated, each batch record can be associated with a liquid product. The liquid product name will be included in each batch record.

Data Types Description

The EFM Simulator Driver only supports the String data type, which is a null terminated ASCII character array.

Address Descriptions

The syntax below is supported for EFM Simulator Driver tags. Each of these tags represents one of the EFM types that can be read from an EFM meter. Each output type can be enabled for up to four meters per device.

Gas Meter Tags

<Channel Name>.<Device Name>._EFM.<Meter Name>._alarms
<Channel Name>.<Device Name>._EFM.<Meter Name>._configuration
<Channel Name>.<Device Name>._EFM.<Meter Name>._events
<Channel Name>.<Device Name>._EFM.<Meter Name>._historydaily
<Channel Name>.<Device Name>._EFM.<Meter Name>._historyhourly

Liquid Meter Tags

<Channel Name>.<Device Name>._LiquidEFM.<Meter Name>._alarms
<Channel Name>.<Device Name>._LiquidEFM.<Meter Name>._batch
<Channel Name>.<Device Name>._LiquidEFM.<Meter Name>._configuration
<Channel Name>.<Device Name>._LiquidEFM.<Meter Name>._events
<Channel Name>.<Device Name>._LiquidEFM.<Meter Name>._historydaily
<Channel Name>.<Device Name>._LiquidEFM.<Meter Name>._historyhourly
<Channel Name>.<Device Name>._LiquidEFM.<Meter Name>._liquidproduct

Important: This driver is not intended to expose real-time data.

Error Descriptions

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

[Address <address> is out of range for the specified device or register.](#)

[Data Type <type> is not valid for device address <address>.](#)

[Device address <address> contains a syntax error.](#)

[Device address <address> is read only.](#)

[Missing address.](#)

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.

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 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.

Index

A

Absolute 9
Address <address> is out of range for the specified device or register. 17
Address Descriptions 15
Advanced Channel Properties 6
Alarms 4, 11
ASCII character array 14
Autogenerate 10

B

Batch 10

C

Channel 4, 15
Channel Assignment 7
Channel Properties - General 4
Channel Properties - Write Optimizations 5
Configuration 4

D

Data Collection 8
Data type <type> is not valid for device address <address>. 17
Data Types Description 14
Daylight Saving Time 9
Description 7
Device 4
Device address <address> contains a syntax error. 17
Device address <address> is read only. 17
Device Properties - General 6
Diagnostics 5
Do Not Scan, Demand Poll Only 9
Driver 4, 7

Duty Cycle 6

E

EFM Meters 10

Error Descriptions 16

Events 4, 11

G

Gas Meter 15

H

Help Contents 3

History 4, 11

I

ID 7

IEEE-754 floating point 6

Initial Updates from Cache 9

Interval 9-10

invalid data type 17

L

Liquid Meter 15

Liquid Product Data 13

M

Method 9

Missing address. 17

Model 7

N

Name 7

Non-Normalized Float Handling 6

O

Optimization Method 5

Overview 3

R

range 17

Records 10

register 17

Request All Data at Scan Rate 8

Request Data No Faster than Scan Rate 8

Respect Client-Specified Scan Rate 8

Respect Tag-Specified Scan Rate 9

S

Scan Mode 8

Setup 4

simulate 3

Simulated 8

Simulated Meter Data 9

Start Date/Time 10

syntax error 17

T

Time Synchronization 9

Time Zone 9

U

Using the EFM Simulator 12

W

Write All Values for All Tags 5

Write Only Latest Value for All Tags 6

Write Only Latest Value for Non-Boolean Tags 5

Write Optimizations 5