Fisher ROC Plus Serial Driver

Table of Contents

Fisher ROC Plus Serial Driver

Help version 1.090

CONTENTS

Overview

The Fisher ROC Plus Serial Driver supports both real-time read and write access and historical Electronic Flow Measurement (EFM) data access in ROC Plus Serial controllers. It is ideal for both local and remote communications to RTUs, supporting Ethernet Encapsulation and having the ability to serialize requests between multiple devices on remote serial networks. Like all EFM-enabled drivers, the Fisher ROC Plus Serial Driver also supports time synchronization and the interleaving of real-time and EFM data access. This ensures that no interruptions occur during EFM data collection.

Access real-time data in ROC Plus Serial controllers via OPC client applications, including HMI, SCADA, Historians, MES, ERP systems, and more. Export Gas and Liquid EFM data to Flow-Cal, PGAS, databases, and other custom formats. For more information on scheduling and exporting EFM data from ROC Plus Serial controllers, refer to the EFM Exporter Plug-In help documentation.

For more information on the Opcodes, Point Types, and Parameters available in the ROC Plus protocol, refer to the device's ROC Plus protocol user manual.

Supported Devices

ROC809
ROC827
ROC809L
ROC827L

Tip: The ROC809L and ROC827L models support both Gas and Liquid EFM. The ROC809 and ROC827 models only support Gas EFM.

Supported Protocol

ROC Plus

Liquid EFM Firmware Requirement

ROC800L (W68258) Firmware version 1.30 or later

User Program Requirements

Liquid Calcs version 1.03.00 (W68259) or later
Batching version 1.03.00 (W68260) or later

Ethernet Encapsulation

This driver supports Ethernet Encapsulation, which allows communication with serial devices attached to an Ethernet network using a serial-to-Ethernet server. It can be enabled through the Communications tab in channel properties.
For more information, refer to Channel Properties — Ethernet Encapsulation.

Note: This driver does not support Report by Exception.

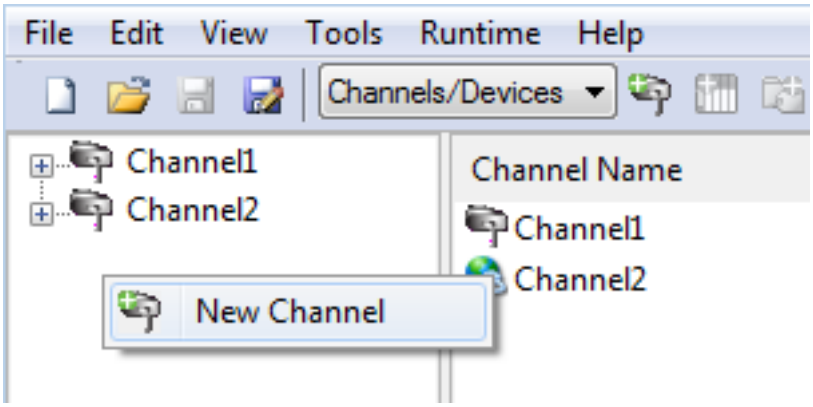
Channel Setup

Communication Serialization

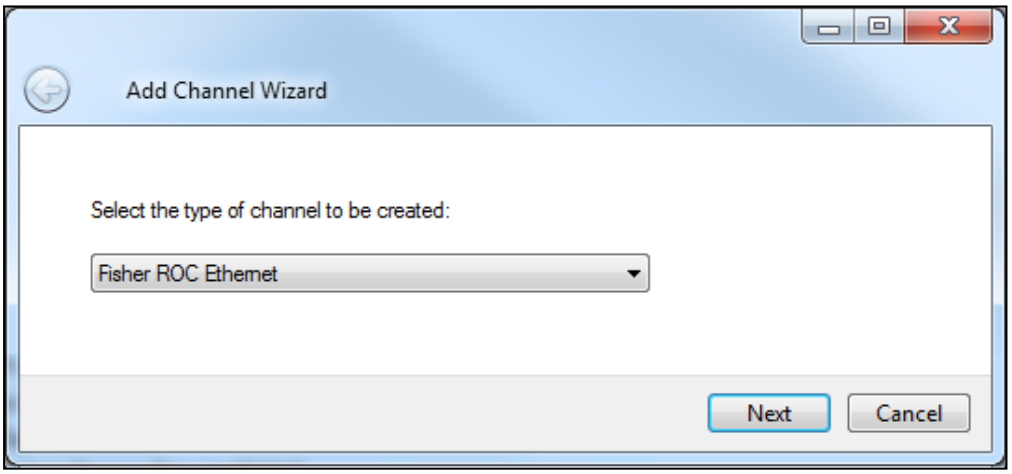
The Fisher ROC Plus Serial Driver supports Communication Serialization, which specifies whether data transmissions should be limited to one channel at a time. The maximum number of supported channels is 1024. *For more information, refer to Channel Properties — Advanced*.

To create a new channel:

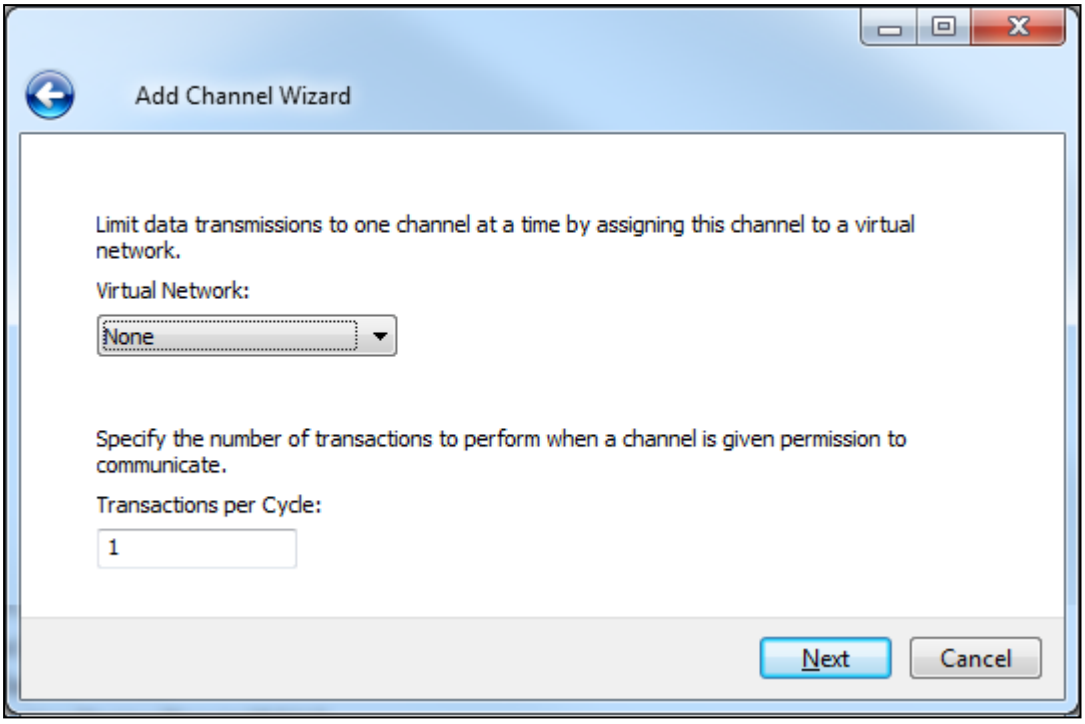
1. In the Project View, right-click and choose **New Channel**.



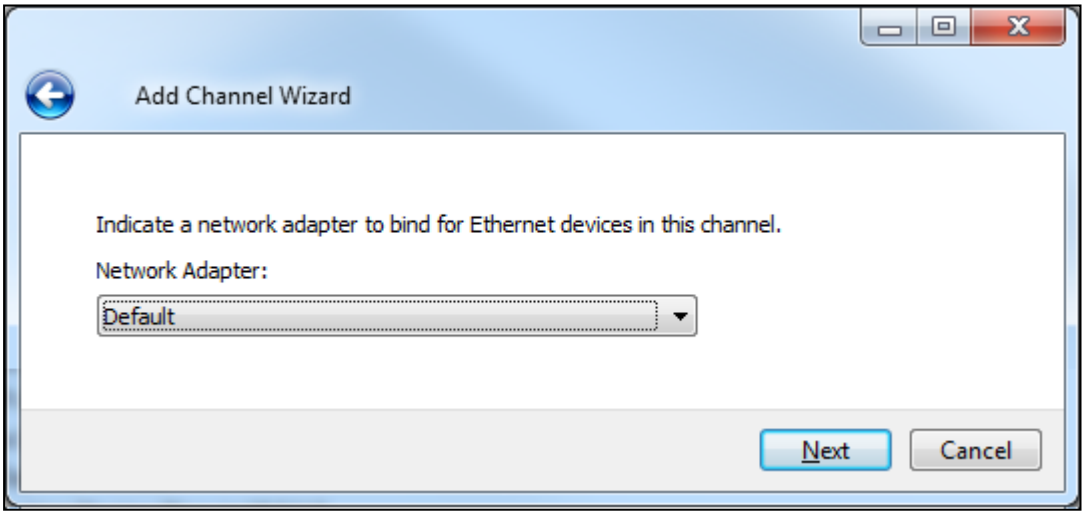
2. From the Device Driver drop-down, select Fisher ROC Plus Serial Driver.



3. Click **Next**.
4. Accept the default channel name or enter a name for the new channel.
5. Click **Next**.
6. Configure the channel.
   Virtual Network - Select the network name or the default (None)
   Transactions per cycle - Enter the target number or accept 1 (default).

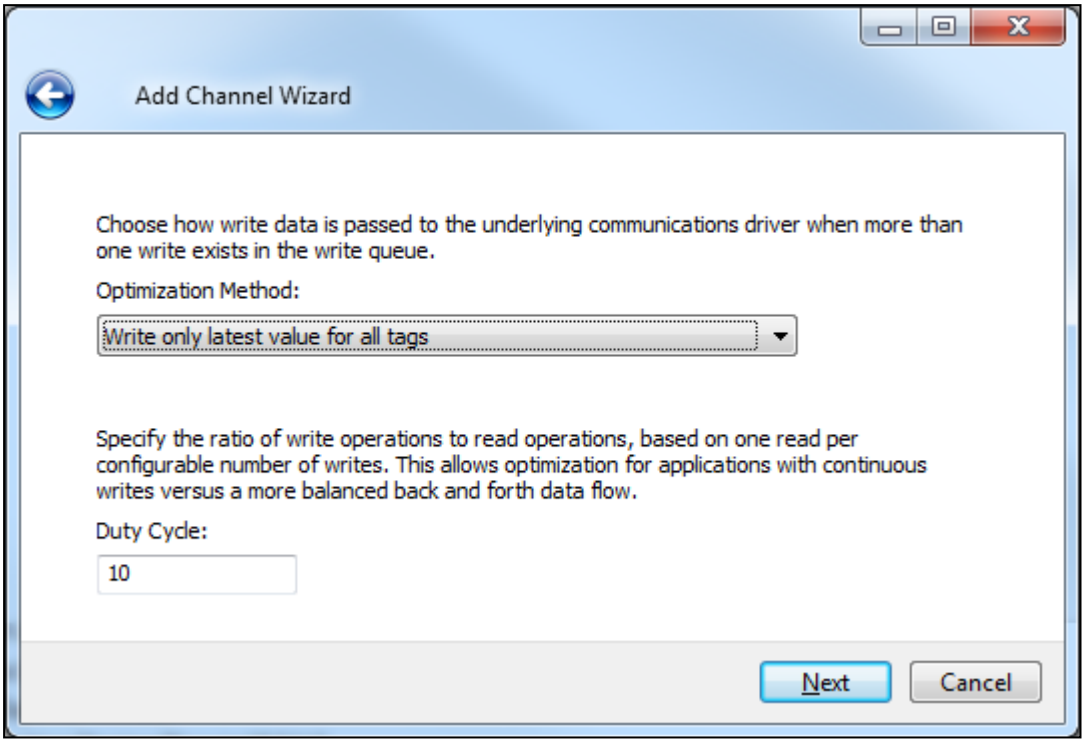


7. Click **Next**.
8. Select **None**, **COM Port**, **Modem**, or **Ethernet Encapsulation** as the hardware device type.

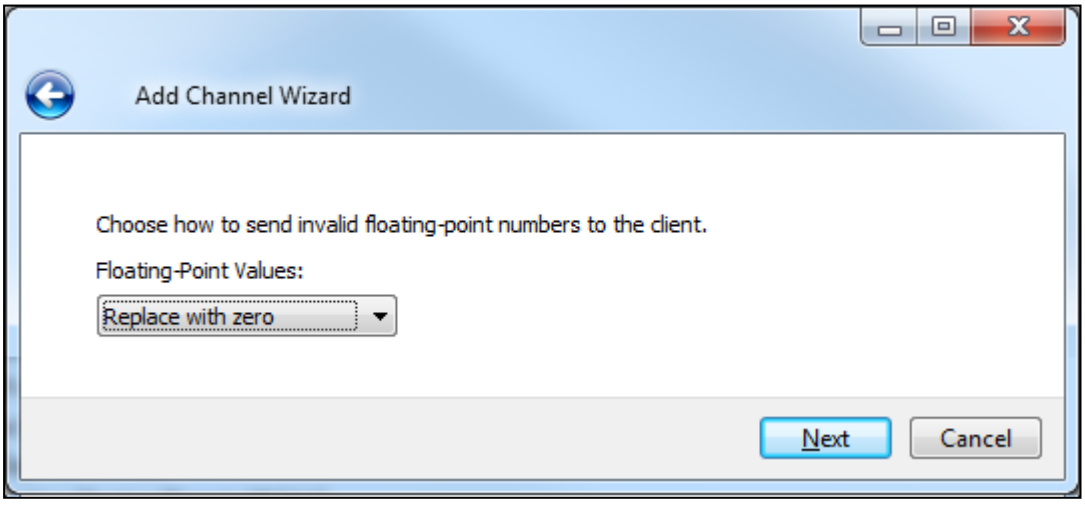


9. Configure communication based on the environment and click **Next**.
10. Select **Close connection when idle** to reduce traffic and specify the number of seconds before the connection is terminated.
11. Click **Next**.
12. In the New Channel - Write Optimizations step, configure channel writes. Optimization Method: Accept the default or change the selection. l Write all values for all tags (most data, most network traffic) l Write only the latest value for non-Boolean tags (least data, least network traffic) l Write

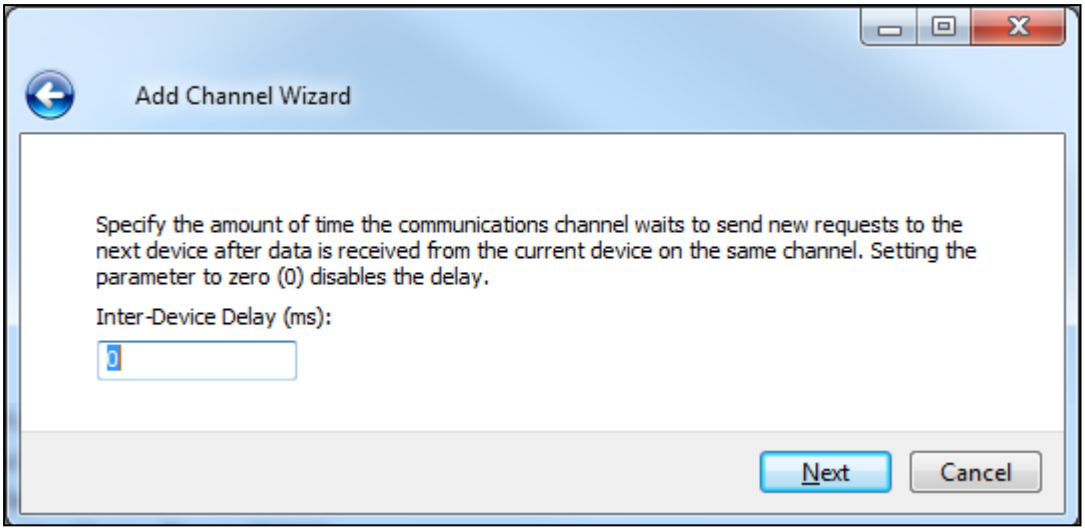
only the latest value for all tags (key data, moderate network traffic) Duty Cycle: Accept 10 (default) writes for every 1 read or adjust as needed. The range is 1-10.



13. Click **Next**.
14. Configure how non-normalized values will be handled. **Replaced with zero** recognizes invalid outliers and eliminates them by replacing the value with integer zero (default). **Unmodified** allows values that are potentially invalid outliers into the data stream.



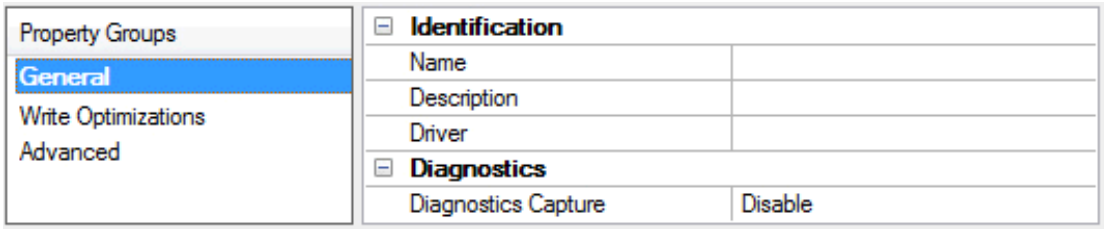
15. Click **Next**.
16. Configure the time, in milliseconds, between requests to devices.
Delay Accept 0 milliseconds (default) or adjust as needed. The range is 0-60000 ms.



17. Click **Next**.
18. Review the configuration in the Summary wizard step.
19. If necessary, use the **Back** button to return to previous steps to make changes.
20. Click **Finish**.

Channel Properties — General

This server supports the use of multiple simultaneous 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.



Identification

Name: Specify the 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. The property is required for creating a channel.

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

Description: Specify user-defined information about this channel.

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

Driver: Specify the protocol / driver for this channel. This property specifies the device driver that was selected during channel creation. It is a disabled setting in the channel properties. The property is required for creating a channel.

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. Changes to the properties should not be made once a large client application has been developed. Utilize proper user role and privilege management to prevent operators from changing properties or accessing 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.

Note: This property is not available if the driver does not support diagnostics.

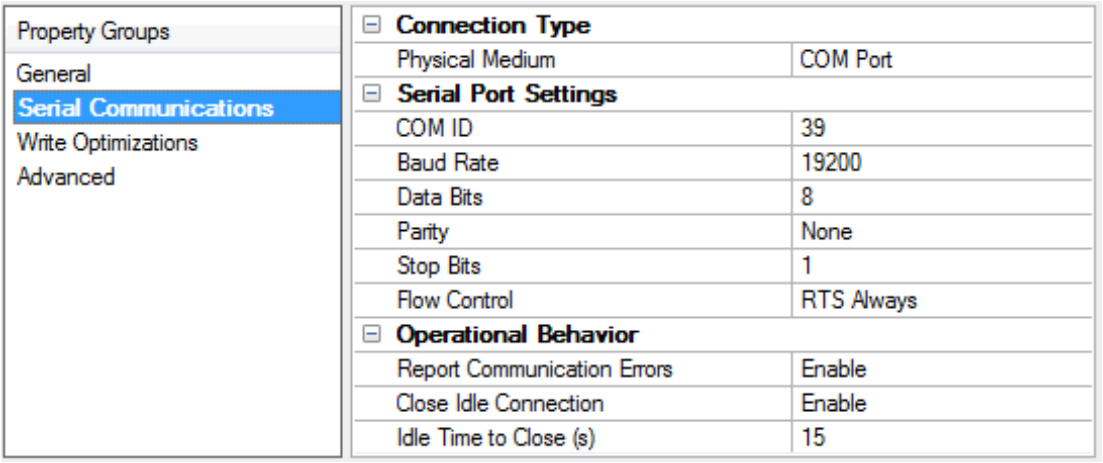
For more information, refer to "Communication Diagnostics" and "Statistics Tags" in the server help.

Channel Properties — Serial Communications

Serial communication properties are available to serial drivers and vary depending on the driver, connection type, and options selected. Below is a superset of the possible properties.

Click to jump to one of the sections: Connection Type, Serial Port Settings or Ethernet Settings, and Operational Behavior.

Note: With the server's online full-time operation, these properties can be changed at any time. Utilize proper user role and privilege management to prevent operators from changing properties or accessing server features.



Connection Type

Physical Medium: Choose the type of hardware device for data communications. Options include COM Port, None, Modem, and Ethernet Encapsulation. The default is COM Port.

- **None**: Select None to indicate there is no physical connection, which displays the Operation with no Communications section.
- **COM Port**: Select Com Port to display and configure the Serial Port Settings section.
- **Modem**: Select Modem if phone lines are used for communications, which are configured in the Modem Settings section.
- **Ethernet Encap.**: Select if Ethernet Encapsulation is used for communications, which displays the Ethernet Settings section.
- **Shared**: Verify the connection is correctly identified as sharing the current configuration with another channel. This is a read-only property.

Serial Port Settings

COM ID: Specify the Communications ID to be used when communicating with devices assigned to the channel. The valid range is 1 to 9991 to 16. The default is 1.

Baud Rate: Specify the baud rate to be used to configure the selected communications port.

Data Bits: Specify the number of data bits per data word. Options include 5, 6, 7, or 8.

Parity: Specify the type of parity for the data. Options include Odd, Even, or None.

Stop Bits: Specify the number of stop bits per data word. Options include 1 or 2.

Flow Control: Select how the RTS and DTR control lines are utilized. Flow control is required to communicate with some serial devices. Options are:

- **None**: This option does not toggle or assert control lines.
- **DTR**: This option asserts the DTR line when the communications port is opened and remains on.
- **RTS**: This option specifies that the RTS line is high if bytes are available for transmission. After all buffered bytes have been sent, the RTS line is low. This is normally used with RS232/RS485 converter hardware.
- **RTS, DTR**: This option is a combination of DTR and RTS.
- **RTS Always**: This option asserts the RTS line when the communication port is opened and remains on.
- **RTS Manual**: This option asserts the RTS line based on the timing properties entered for RTS Line Control. It is only available when the driver supports manual RTS line control (or when the properties are shared and at least one of the channels belongs to a driver that provides this support).
  RTS Manual adds an **RTS Line Control** property with options as follows:
  - **Raise**: This property specifies the amount of time that the RTS line is raised prior to data transmission. The valid range is 0 to 9999 milliseconds. The default is 10 milliseconds.
  - **Drop**: This property specifies the amount of time that the RTS line remains high after data transmission. The valid range is 0 to 9999 milliseconds. The default is 10 milliseconds.
  - **Poll Delay**: This property specifies the amount of time that polling for communications is delayed. The valid range is 0 to 9999. The default is 10 milliseconds.

Tip: When using two-wire RS-485, "echoes" may occur on the communication lines. Since this communication does not support echo suppression, it is recommended that echoes be disabled or a RS-485 converter be used.

Operational Behavior

- **Report Communication Errors**: Enable or disable reporting of low-level communications errors. When enabled, low-level errors are posted to the Event Log as they occur. When disabled, these same errors are not posted even though normal request failures are. The default is Enable.
- **Close Idle Connection**: Choose to close the connection when there are no longer any tags being referenced by a client on the channel. The default is Enable.
- **Idle Time to Close**: Specify the amount of time that the server waits once all tags have been removed before closing the COM port. The default is 15 seconds.

Ethernet Settings

Note: Not all serial drivers support Ethernet Encapsulation. If this group does not appear, the functionality is not supported.

Ethernet Encapsulation provides communication with serial devices connected to terminal servers on the Ethernet network. A terminal server is essentially a virtual serial port that converts TCP/IP messages on the Ethernet network to serial data. Once the message has been converted, users can connect standard devices that support serial communications to the terminal server. The terminal server's serial port must be properly configured to match the requirements of the serial device to which it is attached. *For more information, refer to "Using Ethernet Encapsulation" in the server help.*

- **Network Adapter**: Indicate a network adapter to bind for Ethernet devices in this channel. Choose a network adapter to bind to or allow the OS to select the default.
  Specific drivers may display additional Ethernet Encapsulation properties. For more information, refer to Channel Properties — Ethernet Encapsulation.

Modem Settings

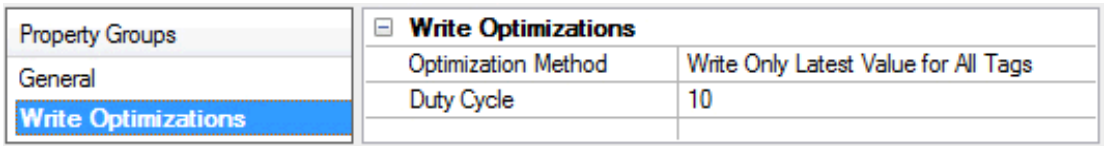
- **Modem**: Specify the installed modem to be used for communications.
- **Connect Timeout**: Specify the amount of time to wait for connections to be established before failing a read or write. The default is 60 seconds.
- **Modem Properties**: Configure the modem hardware. When clicked, it opens vendor-specific modem properties.
- **Auto-Dial**: Enables the automatic dialing of entries in the Phonebook. The default is Disable. *For more information, refer to "Modem Auto-Dial" in the server help.*
- **Report Communication Errors**: Enable or disable reporting of low-level communications errors. When enabled, low-level errors are posted to the Event Log as they occur. When disabled, these same errors are not posted even though normal request failures are. The default is Enable.
- **Close Idle Connection**: Choose to close the modem connection when there are no longer any tags being referenced by a client on the channel. The default is Enable.
- **Idle Time to Close**: Specify the amount of time that the server waits once all tags have been removed before closing the modem connection. The default is 15 seconds.

Operation with no Communications

- **Read Processing**: Select the action to be taken when an explicit device read is requested. Options include Ignore and Fail. Ignore does nothing; Fail provides the client with an update that indicates failure. The default setting is Ignore.

Channel Properties — Write Optimizations

The server must ensure that the data written from the client application gets to the device on time. Given this goal, the server provides optimization properties to meet specific needs or improve application responsiveness.



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.

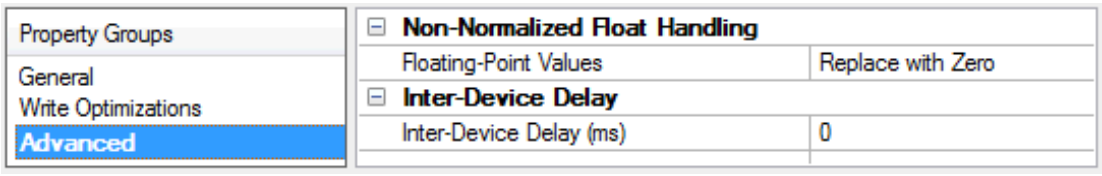
- **Write Only Latest Value for All Tags**: This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write 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: 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. Non-normalized float handling allows users to specify how a driver handles non-normalized IEEE-754 floating point data. 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 not available 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.

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

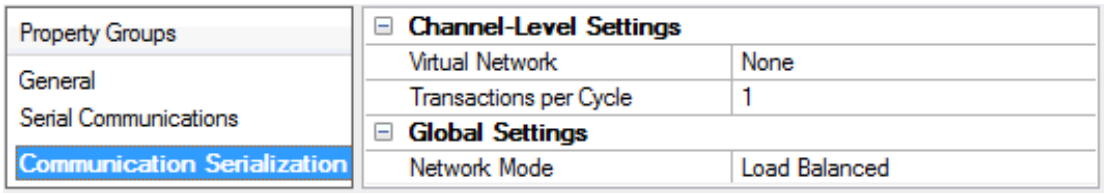
Inter-Device Delay: Specify the amount of time the communications channel waits to send new requests to the next device after data is received from the current device on the same channel. Zero (0) disables the delay.

Note: This property is not available for all drivers, models, and dependent settings.

Channel Properties — Communication Serialization

The server's multi-threading architecture allows channels to communicate with devices in parallel. Although this is efficient, communication can be serialized in cases with physical network restrictions (such as Ethernet radios). Communication serialization limits communication to one channel at a time within a virtual network.

The term "virtual network" describes a collection of channels and associated devices that use the same pipeline for communications. For example, the pipeline of an Ethernet radio is the client radio. All channels using the same client radio associate with the same virtual network. Channels are allowed to communicate each in turn, in a "round-robin" manner. By default, a channel can process one transaction before handing communications off to another channel. A transaction can include one or more tags. If the controlling channel contains a device that is not responding to a request, the channel cannot release control until the transaction times out. This results in data update delays for the other channels in the virtual network.



Channel-Level Settings

Virtual Network: Specify the channel's mode of communication serialization. Options include None and Network 1 - Network 500. The default is None. Descriptions of the options are as follows:

- **None**: This option disables communication serialization for the channel.
- **Network 1 - Network 500**: This option specifies the virtual network to which the channel is assigned.

Transactions per Cycle: Specify the number of single blocked/non-blocked read/write transactions that can occur on the channel. When a channel is given the opportunity to communicate, this is the number of transactions attempted. The valid range is 1 to 99. The default is 1.

Global Settings

Network Mode: This property is used to control how channel communication is delegated. In **Load Balanced** mode, each channel is given the opportunity to communicate in turn, one at a time. In **Priority** mode, channels are given the opportunity to communicate according to the following rules (highest to lowest priority):

1. Channels with pending writes have the highest priority.
2. Channels with pending explicit reads (through internal plug-ins or external client interfaces) are prioritized based on the read's priority.
3. Scanned reads and other periodic events (driver specific).

The default is Load Balanced and affects *all* virtual networks and channels.

Devices that rely on unsolicited responses should not be placed in a virtual network. In situations where communications must be serialized, it is recommended that Auto-Demotion be enabled.

Due to differences in the way that drivers read and write data (such as in single, blocked, or non-blocked transactions); the application's Transactions per cycle property may need to be adjusted. When doing so, consider the following factors:

- How many tags must be read from each channel?
- How often is data written to each channel?
- Is the channel using a serial or Ethernet driver?
- Does the driver read tags in separate requests, or are multiple tags read in a block?
- Have the device's Timing properties (such as Request timeout and Fail after *x* successive timeouts) been optimized for the virtual network's communication medium?

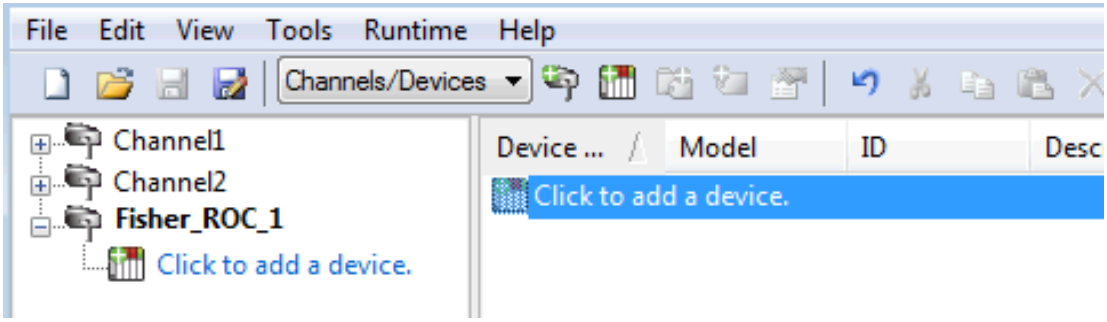
Device Setup

Once at least one channel is configured, devices using the Remote Operation Controllers (ROC) protocol can be added for data collection and monitoring. The maximum number of devices supported on any one channel is 255. Devices should be added to channels organized based on the channel configuration.

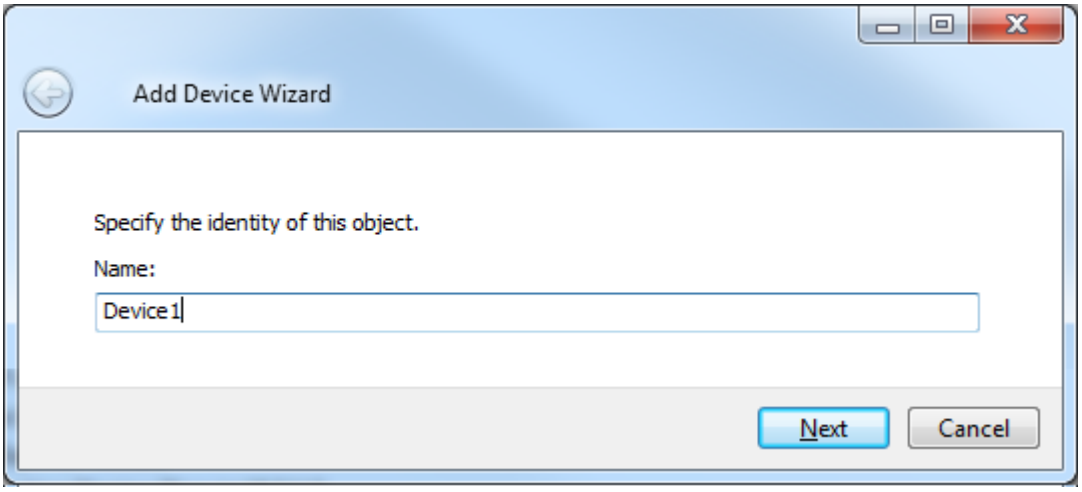
Adding a Device

To add a new device to a channel:

1. In the Project View, select the channel to contain the new device.
2. Select **Click to add a device** or right-click and choose **New Device**.



3. Accept the default channel name or enter a name for the new device.

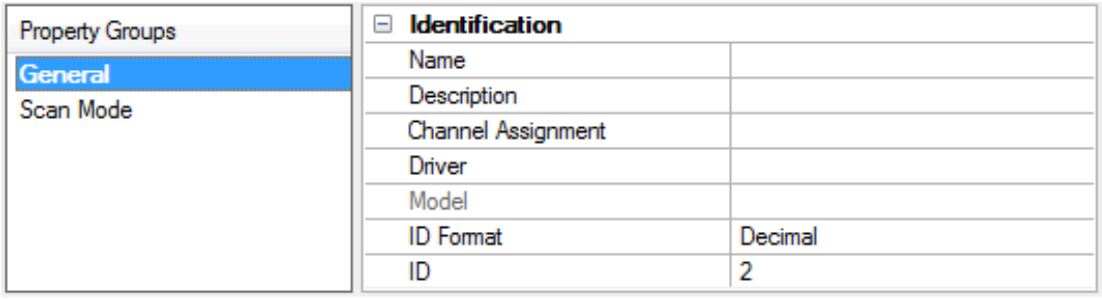


4. Click **Next >**.
5. From the Device Model drop-down, select the correct model for the device.
6. Click **Next >**.
7. Configure the Scan Mode.
8. Click **Next >**.
9. Configure the **Timeouts and Timing**.
10. Click **Next >**.
11. Configure how communication failure is handled.

12. Click **Next >**.
13. Configure Tag Generation.
14. Click **Next >**.
15. Configure timezone, DST, and Time Synchronization *(see **Time Synchronization**)*.
16. Click **Next >**.
17. Identify and locate existing files to be included *(see Tag Import Settings)*
18. Click **Next >**.
19. Identify the source and destination addresses *(see Communication Specification)*.
20. Click **Next >**.
21. Configure the authorized user *(see Operator Identification)*.
22. Click **Next >**.
23. Review the configuration.
24. If necessary, use the **< Back** button to return to previous steps to make changes.
25. Click **Finish >**.

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.



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: Specify the user-defined information about this device.
Many of these properties, including Description, have an associated system tag.

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

Driver: Selected protocol driver for this device.

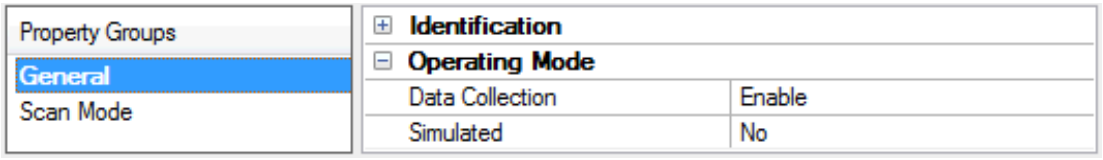
Model: Specify the type of device that is associated with this ID. The contents of the drop-down menu depend on the type of communications driver being used. Models that are not supported by a driver are disabled. If the communications driver supports multiple device models, the model selection can only 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 format is set by the driver by default. Options include 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: Place the device into or out of 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:

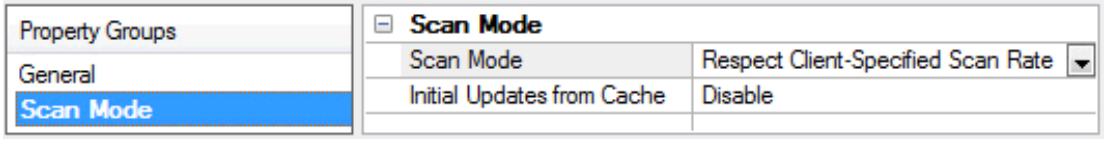
1. This System tag (_Simulated) is read only and cannot be written to for runtime protection. The System tag allows this property to be monitored from the client.

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.



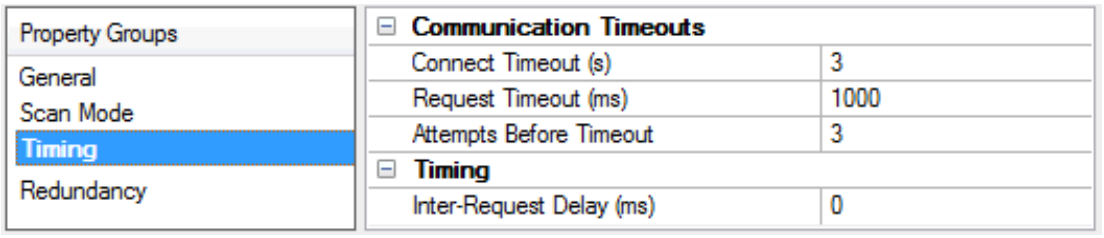
Scan Mode: Specify how tags in the device are scanned for updates sent to subscribing clients. Descriptions of the options are:

- **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 value set as the maximum scan rate. 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 OPC 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 — Timing

The device Timing properties allow the driver's response to error conditions to be tailored to fit the application's needs. In many cases, the environment requires changes to these properties for optimum performance. Factors such as electrically generated noise, modem delays, and poor physical connections can influence how many errors or timeouts a communications driver encounters. Timing properties are specific to each configured device.



Communications Timeouts

Connect Timeout: This property (which is used primarily by Ethernet based drivers) controls the amount of time required to establish a socket connection to a remote device. The device's connection time often takes longer than normal communications requests to that same device. The valid range is 1 to 30 seconds. The default is typically 3 seconds, but can vary depending on the driver's specific nature. If this setting is not supported by the driver, it is disabled.

Note: Due to the nature of UDP connections, the connection timeout setting is not applicable when communicating via UDP.

Request Timeout: Specify an interval used by all drivers to determine how long the driver waits for a response from the target device to complete. The valid range is 50 to 9,999,999 milliseconds (167.6667 minutes). The default is usually 1000 milliseconds, but can vary depending on the driver. The default timeout for most serial drivers is based on a baud rate of 9600 baud or better. When using a driver at lower baud rates, increase the timeout to compensate for the increased time required to acquire data.

Attempts Before Timeout: Specify how many times the driver issues a communications request before considering the request to have failed and the device to be in error. The valid range is 1 to 10. The default is typically 3, but can vary depending on the driver's specific nature. The number of attempts configured for an application depends largely on the communications environment. This property applies to both connection attempts and request attempts.

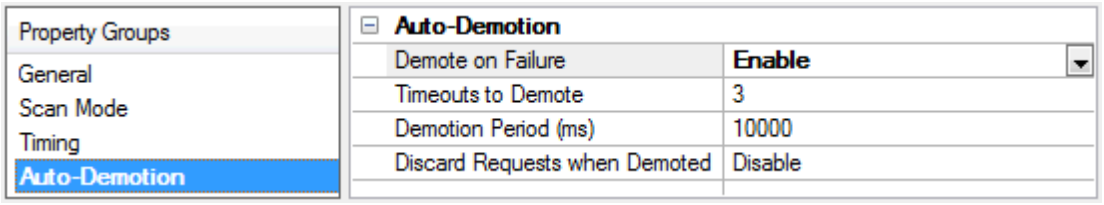
Timing

Inter-Request Delay: Specify how long the driver waits before sending the next request to the target device. It overrides the normal polling frequency of tags associated with the device, as well as one-time reads and writes. This delay can be useful when dealing with devices with slow turnaround times and in cases where network load is a concern. Configuring a delay for a device affects communications with all other devices on the channel. It is recommended that users separate any device that requires an inter-request delay to a separate channel if possible. Other communications properties (such as communication serialization) can extend this delay. The valid range is 0 to 300,000 milliseconds; however, some drivers may limit the maximum value due to a function of their particular design. The default is 0, which indicates no delay between requests with the target device.

Note: Not all drivers support Inter-Request Delay. This setting does not appear if it is not available.

Device Properties — Auto-Demotion

The Auto-Demotion properties can temporarily place a device off-scan in the event that a device is not responding. By placing a non-responsive device offline for a specific time period, the driver can continue to optimize its communications with other devices on the same channel. After the time period has been reached, the driver re-attempts to communicate with the non-responsive device. If the device is responsive, the device is placed on-scan; otherwise, it restarts its off-scan time period.



Demote on Failure: When enabled, the device is automatically taken off-scan until it is responding again.
Tip: Determine when a device is off-scan by monitoring its demoted state using the _AutoDemoted system tag.

Timeouts to Demote: Specify how many successive cycles of request timeouts and retries occur before the device is placed off-scan. The valid range is 1 to 30 successive failures. The default is 3.

Demotion Period: Indicate how long the device should be placed off-scan when the timeouts value is reached. During this period, no read requests are sent to the device and all data associated with the read requests are set to bad quality. When this period expires, the driver places the device on-scan and allows for another attempt at communications. The valid range is 100 to 3600000 milliseconds. The default is 10000 milliseconds.

Discard Requests when Demoted: Select whether or not write requests should be attempted during the off-scan period. Disable to always send write requests regardless of the demotion period. Enable to discard writes; the server automatically fails any write request received from a client and does not post a message to the Event Log.

Device Properties — Tag Generation

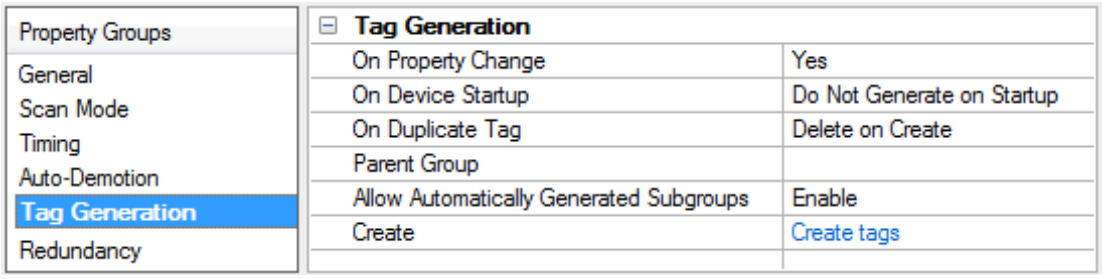
The automatic tag database generation features make setting up an application a plug-and-play operation. Select communications drivers can be configured to automatically build a list of tags that correspond to device-specific data. These automatically generated tags (which depend on the nature of the supporting driver) can be browsed from the clients.

Not all devices and drivers support full automatic tag database generation and not all support the same data types. Consult the data types descriptions or the supported data type lists for each driver for specifics.

If the target device supports its own local tag database, the driver reads the device's tag information and uses the data to generate tags within the server. If the device does not natively support named tags, the driver creates a list of tags based on driver-specific information. An example of these two conditions is as follows:

1. If a data acquisition system supports its own local tag database, the communications driver uses the tag names found in the device to build the server's tags.
2. If an Ethernet I/O system supports detection of its own available I/O module types, the communications driver automatically generates tags in the server that are based on the types of I/O modules plugged into the Ethernet I/O rack.

Note: Automatic tag database generation's mode of operation is completely configurable. *For more information, refer to the property descriptions below.*



On Property Change: If the device supports automatic tag generation when certain properties change, the **On Property Change** option is shown. It is set to **Yes** by default, but it can be set to **No** to control over when tag generation is performed. In this case, the **Create tags** action must be manually invoked to perform tag generation.

On Device Startup: Specify when OPC tags are automatically generated. Descriptions of the options are as follows:

- **Do Not Generate on Startup**: This option prevents the driver from adding any OPC tags to the tag space of the server. This is the default setting.
- **Always Generate on Startup**: This option causes the driver to evaluate the device for tag information. It also adds tags to the tag space of the server every time the server is launched.
- **Generate on First Startup**: This option causes the driver to evaluate the target device for tag information the first time the project is run. It also adds any OPC tags to the server tag space as needed.

Note: When the option to automatically generate OPC tags is selected, any tags that are added to the server's tag space must be saved with the project. Users can configure the project to automatically save from the **Tools** | **Options** menu.

On Duplicate Tag: When automatic tag database generation is enabled, the server needs to know what to do with the tags that it may have previously added or with tags that have been added or modified after the communications driver since their original creation. This setting controls how the server handles OPC tags that were automatically generated and currently exist in the project. It also prevents automatically generated tags from accumulating in the server.

For example, if a user changes the I/O modules in the rack with the server configured to **Always Generate on Startup**, new tags would be added to the server every time the communications driver detected a new I/O module. If the old tags were not removed, many unused tags could accumulate in the server's tag space. The options are:

- **Delete on Create**: This option deletes any tags that were previously added to the tag space before any new tags are added. This is the default setting.
- **Overwrite as Necessary**: This option instructs the server to only remove the tags that the communications driver is replacing with new tags. Any tags that are not being overwritten remain in the server's tag space.
- **Do not Overwrite**: This option prevents the server from removing any tags that were previously generated or already existed in the server. The communications driver can only add tags that are completely new.
- **Do not Overwrite, Log Error**: This option has the same effect as the prior option, and also posts an error message to the server's Event Log when a tag overwrite would have occurred.

Note: Removing OPC tags affects tags that have been automatically generated by the communications driver as well as any tags that have been added using names that match generated tags.

Users should avoid adding tags to the server using names that may match tags that are automatically generated by the driver.

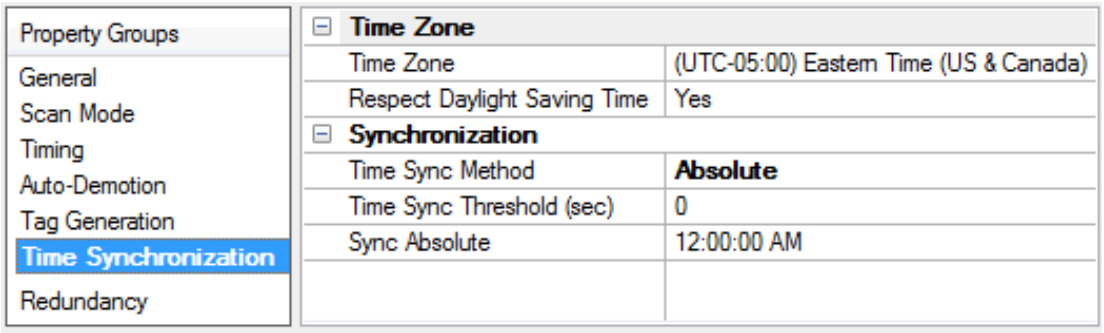
Parent Group: This property keeps automatically generated tags from mixing with tags that have been entered manually by specifying a group to be used for automatically generated tags. The name of the group can be up to 256 characters. This parent group provides a root branch to which all automatically generated tags are added.

Allow Automatically Generated Subgroups: This property controls whether the server automatically creates subgroups for the automatically generated tags. This is the default setting. If disabled, the server generates the device's tags in a flat list without any grouping. In the server project, the resulting tags are named with the address value. For example, the tag names are not retained during the generation process.
Note: If, as the server is generating tags, a tag is assigned the same name as an existing tag, the system automatically increments to the next highest number so that the tag name is not duplicated. For example, if the generation process creates a tag named "AI22" that already exists, it creates the tag as "AI23" instead.

Create: Initiates the creation of automatically generated OPC tags. If the device's configuration has been modified, **Create tags** forces the driver to reevaluate the device for possible tag changes. Its ability to be accessed from the System tags allows a client application to initiate tag database creation.
Note: **Create tags** is disabled if the Configuration edits a project offline.

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.



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: Specify Yes to follow Daylight Saving Time offset when synching the device time. Specify No to ignore Daylight Saving Time. Only time zones that observe Daylight Saving Time will be affected. The default is No (disabled).
Note: When enabled, the time of the device is adjusted by +1 hour for Daylight Saving Time (in the spring), and adjusted by -1 hour after Daylight Saving Time (in the fall).

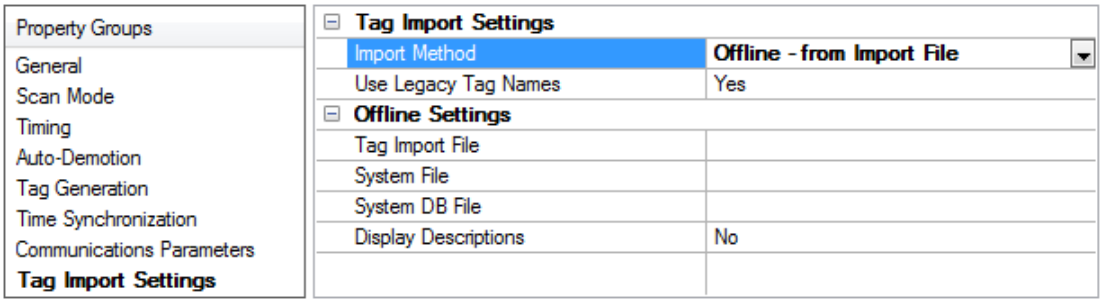
Time Sync Method: Specify the method of synchronization. Options include Disabled, Absolute, and Interval. The default is Disabled. Descriptions of the options are as follows:

- **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.
- **OnPoll**: Synchronizes when poll is completed (applicable only to EFM devices).

Time Sync Threshold: Specify the maximum allowable difference, in seconds, between the device time and the system time before syncing the device time to the system time. If the threshold is set to 0, a time synchronization occurs every time. The default is 0 seconds. The maximum allowable threshold is 600 seconds.

Device Properties — Tag Import Settings

A tag database can be created based on the device's configuration file or a ROCLINK 800 project file by bringing those tags into the project with an import. Tag Import settings are defined as a device is added and configured through the New Device wizard and can also be modified after the device has been added. To define the Tag Import settings for a new device, follow the steps for **defining a new device**.



Import Method: specifies the import method. Options include Online - from Device and Offline - from Import File. The default setting is Online - from Device. Descriptions of the options are as follows:

- **Online - From Device**: This method automatically creates tags by polling the device for its configuration and I/O data.
- **Offline - From Import File**: This method automatically creates tags from a project file created in ROCLINK 800.

Use Legacy Tag Names: When set to Yes, Automatic Tag Database Generation creates tags with names consistent with the tags created in prior versions of the server. When unchecked, Automatic Tag Database Generation creates tags with names consistent with the current version of the server. The default setting is set to Yes. *For more information, refer to "Legacy vs. Non-Legacy Tag Names" below.*

Tag Import File: specifies the * .800 file and path that was created using the ROCLINK800 software.

ROC System File: specifies the * .mdb file and path. This file is usually named "ROC.mdb" and resides in the same folder where the ROCLINK 800 software is installed.

System DB File: specifies the * .mdw file and path. This file is usually named "ROCLINK.mdw" and resides in the same folder where the ROCLINK 800 software is installed.

Display Descriptions: When set to Yes, this option includes the tag descriptions from the ROCLINK 800 database.

Legacy vs. Non-Legacy Tag Names

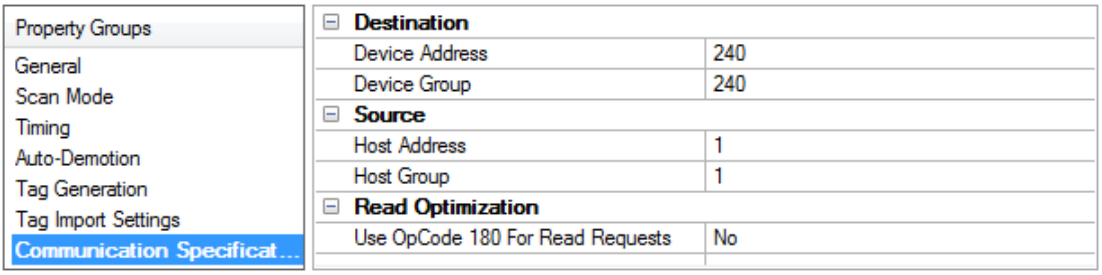
For information on how legacy and non-legacy tag names are automatically generated based on the "Use legacy tag names" option, refer to the table below.

Tag Type	Mode	Tag Name	Tag Address
Non-Boolean	Legacy Non-Legacy	IPAddress_137_0 IP Address-137 (T137,L0,P1)	137-0.1 137-0.1
Boolean (.Bit)	Legacy Non-Legacy	HighAlarm_41_0 High Alarm-41 (T41,L0,P16) Bit 2	41-0.16:2 41-0.16:2

***See Also**: Tag Generation*

Device Properties — Communication Specification

Communication Specification settings are defined as a device is added and configured through the New Device wizard and can also be modified after the device has been added. To define the Communication Specification settings for a new device, follow the steps for **defining a new device**. To modify settings on a defined device; select the device, right-click, select **Properties**, and select the **Communication Specification** property group.



Device Address: specifies the device number of the remote ROC device. The valid range is 1 to 255. The default setting is 240.

Device Group: specifies the group number of the remote ROC device. The valid range is 1 to 255. The default setting is 240.

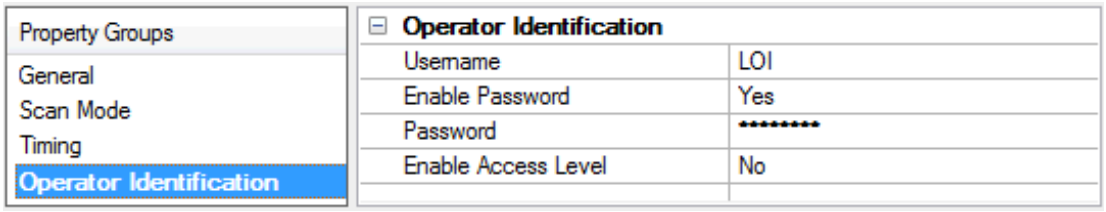
Host Address: specifies the ROC unit number of the server. The valid range is 1 to 255. The default setting is 1.

Host Group: specifies the ROC group number of the server. The valid range is 1 to 255. The default setting is 1.

Use Opcode 180 for Read Requests: This option should be used if few parameters from each point type and logical address are typically used, as it yields more efficient communication. If set to No, the driver uses Opcode 167 to read entire point type logical addresses in one transaction. The default is No.

Device Properties — Operator Identification

Operator Identification settings are defined as a device is added and configured and can also be modified after the device has been added. To define the Operator Identification settings for a new device through the wizard, follow the steps for **defining a new device**.



Username: specifies the authorized account identity. Three characters (as set in the device) are required.

Enable Password: When set to yes, this option specifies that the device has a password defined for the operator ID. The default setting is Yes.

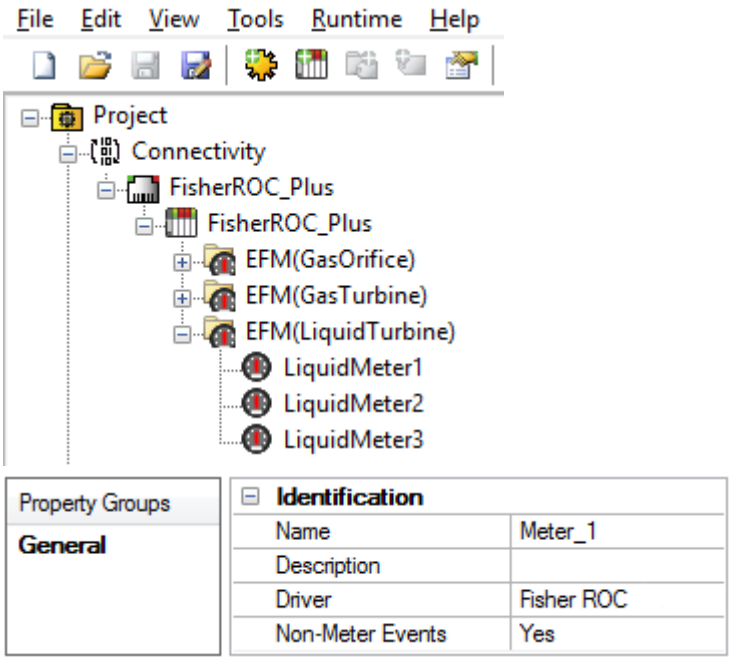
Password: Four numeric characters can be entered. The valid range is 0000 to 9999. This parameter is not available unless the Enable Password parameter is set to Yes.

Enable Access Level: When set to Yes, this option specifies that the device has defined access levels. The default setting is No.

Access Level: sets the level of access, where the valid range is 0 to 5. This parameter is not available unless the Enable Access Level parameter is set to Yes.

Device Properties — EFM Meters

EFM Meters can be added to an existing device. The Fisher ROC Plus Serial Driver supports Gas and Liquid Meters. A meter group for each type and model of meter is automatically created with each device. To add meters to a device; click the + symbol to expand the meter groups below the device. Right-click the meter group and select **New Meter**. A new meter is added below the selected meter group. To edit the meter; right-click and select **Properties**.



Name: This parameter specifies the meter name. Each meter must be assigned a unique name. The default setting is Meter_n (where n increments up from 1 for the first meter).

Description: This parameter can be used to provide a short description of the meter.

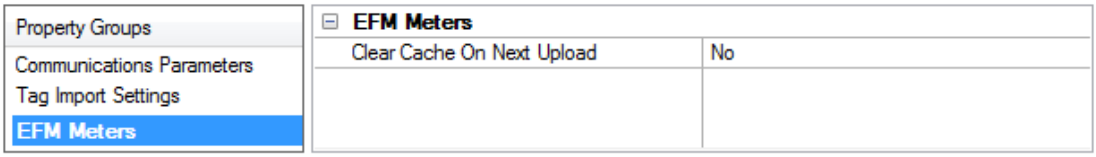
Driver: This parameter indicates the current driver. It cannot be modified.

Non-Meter Events: This parameter specifies whether non-meter EFM events are provided to EFM Exporters for this meter. The default setting is Yes.

Important!

- This driver supports Gas Orifice, Gas Turbine, and Liquid Turbine (809L and 827L models only) meters. Meter groups for each meter type and model are automatically created for each device. A maximum of six liquid meters can be added.
- Changes to settings in the hardware meter do NOT affect the data returned until the server runtime service is shutdown and restarted (through the Administrator or by stopping the service and restarting it).
- The meter order in ROCLink800 corresponds to the meter order in the EFM meter list.
- Extended History data is not supported.
- The Identification parameters of the meter groups cannot be edited and are provided for informational purposes only.

Clear Cache on Next Upload: Users have the option to clear any cached EFM data from the device during the next upload. This feature also removes pointer files, which are used to track EFM uploads to prevent uploading the same records twice. All EFM data is re-uploaded. Once the cache is cleared, this parameter is automatically disabled. To enable this option, open **Device Properties | EFM Meters** and select Yes in the Clear cache on next upload property. The default setting is **No**.



Liquid Meter Record Timestamp Tolerance

Some ROC+ Liquid EFM meter configurations store station and meter data in separate history segments. This driver uploads the data from both segments and merges it into a single historical record based on timestamp.

Important: Data from each segment is not merged if the timestamps from each record are more than two seconds apart.

***See Also**: EFM Attribute Mapping*

EFM Attribute Mapping

The tables below describe the mapping of TLPs to attributes used by the EFM Exporter Plug-in. This mapping is not user-configurable and provided for reference only.

Gas Configuration Mapping

n = orifice run number or turbine run number
s = station number
h = history segment number

EFM Config Attribute	ROC+ Orifice TLP	ROC+ Turbine TLP
meter_id	113-n.0	115-n.0
pressure_base	112-s.13	
temp_base	112-s.14	112-s.14
live_analysis	112-s.24	112-s.24
calculation_method	112-s.1	112-s.1
pipe_diameter	113-n.12	
pipe_ref_temp	113-n.13	
static_pressure_tap	113-n.3	
unit	112-s.4	
orifice_plate_size	113-n.15	
orifice_ref_temp	113-n.16	
dp_low_flow_cutoff	113-n.20	115-n.22
atmospheric_pressure	112-s.16	112-s.16
btu	112-s.22	112-s.22
specific_gravity	112-s.23	112-s.23
viscosity	113-n.18	
specific_heats	113-n.19	
pipe_material	113-n.14	
orifice_material	113-n.17	
btu_base	112-n.21	112-n.21
compressibility_calc	112-n.3	
static_pressure_type	113-n.2	115-n.2
static_pressure_taps	113-n.3	
k_factor		115-n.11
fixed_factor	113-n.34	115-n.21
co2	112-s.27	112-s.27
n2	112-s.26	112-s.26
c1	112-s.28	112-s.28
c2	112-s.29	112-s.29
c3	112-s.30	112-s.30
isoc4	112-s.32	112-s.32
nc4	112-s.31	112-s.31
isoc5	112-s.34	112-s.34
c5	112-s.33	112-s.33
c6	112-s.35	112-s.35
c7	112-s.36	112-s.36
c8	112-s.37	112-s.37
c9	112-s.38	112-s.38
c10	112-s.39	112-s.39

EFM Config Attribute	ROC+ Orifice TLP	ROC+ Turbine TLP
o2	112-s.43	112-s.43
h2o	112-s.41	112-s.41
h2s	112-s.40	112-s.40
he	112-s.42	112-s.42
h2	112-s.45	112-s.45
co	112-s.44	112-s.44
ar	112-s.62	112-s.62
contract_hour	124-h.8	124-h.8

Gas History Mapping

n = meter number

EFM Attribute	Orifice TLP	Turbine TLP
flow_time	114-n.28	116-n.21
avg_diff_pressure	113-n.26	113-n.26
avg_pressure	113-n.28	115-n.16
avg_temp	113-n.30	115-n.18
avg_extension	114-n.4	114-n.4
c_prime	114-n.12	114-n.12
pulses	116-n.9	116-n.9
raw_volume	116-n.31	116-n.31
flowing_condition_factor	116-n.8	116-n.8
total_volume (accumulated)	114-n.23	116-n.16
total_volume (daily)	114-n.0	116-n.0
total_volume (hourly)	114-n.2	116-n.2
total_energy (accumulated)	114-n.33	116-n.26
total_energy (daily)	114-n.1	116-n.1
total_energy (hourly)	114-n.3	116-n.3
avg_btu	112-n.22	112-n.22
avg_specific_gravity	112-n.23	112-n.23
avg_co2	112-n.27	112-n.27
avg_n2	112-n.26	112-n.26
avg_c1	112-n.28	112-n.28
avg_c2	112-n.29	112-n.29
avg_c3	112-n.30	112-n.30
avg_isoc4	112-n.32	112-n.32
avg_nc4	112-n.31	112-n.31
avg_isoc5	112-n.34	112-n.34
avg_neoc5	112-n.33	112-n.33
avg_c6	112-n.35	112-n.35

EFM Attribute	Orifice TLP	Turbine TLP
avg_c7	112-n.36	112-n.36
avg_o2	112-n.43	112-n.43
avg_h2o	112-n.41	112-n.41
avg_h2s	112-n.42	112-n.42
specific_heat_ratio	113-n.19	113-n.19
viscosity	113-n.18	113-n.18

Liquid Mapping

n = meter number
l = logical number is dependent on input TLP

EFM Attribute	TLP
meter_id	204-n.0
meter_serial_number	204-n.3
liquid_calculation_method	204-n.7
liquid_product_name	201-n.0
liquid_density_meter_factor	202-n.9
temp_low_alarm_setpoint (Analog Input)	103-l.24
temp_high_alarm_setpoint (Analog Input)	103-l.25
temp_calibrated_range_low (Analog Input)	103-l.13
temp_calibrated_range_high (Analog Input)	103-l.17
temp_low_alarm_setpoint (RTD Input)	106-l.25
temp_high_alarm_setpoint (RTD Input)	106-l.26
temp_calibrated_range_low (RTD Input)	106-l.14
temp_calibrated_range_high (RTD Input)	106-l.18
temp_high_alarm_setpoint (Thermocouple Input)	107-l.11
temp_calibrated_range_low (Thermocouple Input)	107-l.13
temp_high_alarm_setpoint (MVS Input)	103-l.25
temp_calibrated_range_low (MVS Input)	103-l.13
low_flow_cutoff_set_point	204-n.36
atmospheric_pressure	200-0.16
liquid_totalizer_digits	91-0.56
static_pressure_units	200-0.4
temp_units	200-0.5
density_units	200-0.6
volume_units	200-0.10
mass_units	200-0.11
meter_mf_kf_curve_type	204-n.76
linear_meter_factor	204-n.75
linear_meter_k_factor	204-n.74

EFM Attribute	TLP
contract_hour	200-0.89

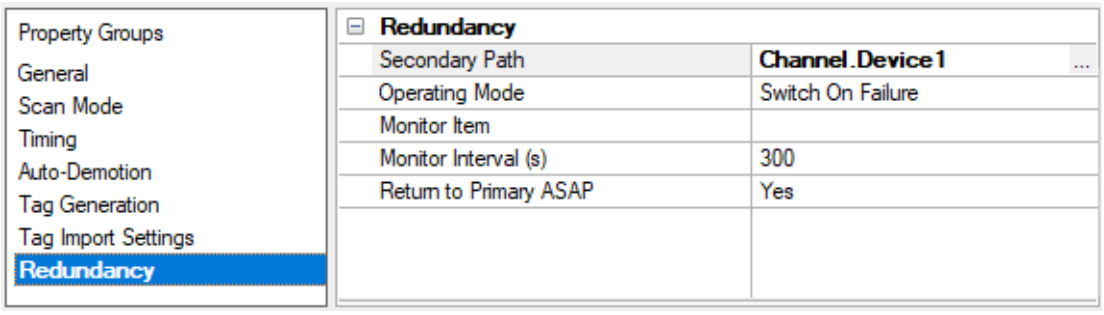
Liquid History Mapping

n = meter number

EFM Attribute	TLP
flow_time	204-n.245
liquid_observed_density	204-n.22
liquid_density_temperature	202-n.6
liquid_density_pressure	202-n.8
liquid_uncorrected_density	202-n.3
liquid_meter_flowing_density	204-n.21
liquid_meter_temperature	204-n.34
liquid_meter_pressure	204-n.33
liquid_indicated_volume	204-n.145
liquid_iv_index_end	204-n.145
liquid_gross_volume	204-n.146
liquid_gv_index_end	204-n.146
liquid_gross_standard_volume	204-n.147
liquid_gsv_index_end	204-n.147
liquid_mass	204-n.150
liquid_mass_index_end	204-n.150
liquid_net_standard_volume	204-n.148
liquid_nsv_index_end	204-n.148
liquid_sw_volume	204-n.149
liquid_sw_index_end	204-n.149
liquid_sw_percent	204-n.35
liquid_sw_correction	204-n.40
liquid_pulses	204-n.184
liquid_pulse_index_end	204-n.184
liquid_densitometer_factor	202-n.9
meter_factor	204-n.75
k_factor	204-n.74
liquid_equilibrium_vapor_pressure	204-n.243
ctl	204-n.44
cpl	204-n.45
ctpl	204-n.46
ccf	204-n.39
co2	201-n.119

EFM Attribute	TLP
n2	201-n.117
c1	201-n.105
c2	201-n.106
c3	201-n.107
ic4	201-n.109
nc4	201-n.108
ic5	201-n.111
nc5	201-n.110
neoc5	201-n.104
c6	201-n.112
c7	201-n.113
c8	201-n.124
c9	201-n.125
c10	201-n.126
ethylene	201-n.114
propylene	201-n.115
o2	201-n.118
h2o	201-n.128
h2s	201-n.120
he	201-n.127

Device Properties — Redundancy



Redundancy is available with the Media-Level Redundancy Plug-In.

Consult the website, a sales representative, or the user manual for more information.

PLUS Data Type Descriptions

Data Types	Description
Boolean	Single bit
Byte	Unsigned 8-bit value bit 0 is the low bit bit 7 is the high bit
Char	Signed 8-bit value bit 0 is the low bit bit 6 is the high bit bit 7 is the sign bit
Date	The number of seconds since Jan 1 1970 @ 00:00:00 Example Date format:YYYY-MM-DDTHH:MM:SS.000 2000-01-01T12:30:45.000
DWord	Unsigned 32-bit value bit 0 is the low bit bit 31 is the high
DWord TLP	32-bit value: Point 'T'ype, 'L'ogical (or point number), and 'P'arameter number* Three bytes are used, but the top byte is not. • The type refers to the point type number.** • The location/logical number refers to individual points. • The parameter is a number assigned to each piece of data contained in a given point type. Example 557158: BIN = 00001000 10000000 01100110. 01100110 = Point Type 102, 10000000 = Location 128, 00001000 = Parameter 8. The resulting TLP is 102-128.8
Float	32-bit floating point value bit 0 is the low bit bit 31 is the high bit
LongLong	Signed 64-bit value Bit 0 is the low bit bit 62 is the high bit bit 63 is the sign bit
QWord	Unsigned 64-bit value bit 0 is the low bit bit 63 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
String	A linear group of ASCII characters with preserved spaces (1 byte per character)

Data Types	Description
Word	Unsigned 16-bit value bit 0 is the low bit bit 15 is the high bit
Word HOURMINUTE	Point Type 100, parameters 3-5 are HOURMINUTE. This driver represents this data type as Word. Length: 2 Bytes. Time is listed as a decimal based number. The first two digits represent the hour and the last two digits represent the minute. Range: 9999, 0-23 for 2 MS Digits; 0-59 for 2 LS Digits Special Meanings: 9999 = Disabled

See Also:
Logical Point Number Details
ROC Plus Point Types
TLP Data Type Conversion Process and Examples

TLP Data Type Conversion Process and Examples

The Fisher ROC TLP data type is represented by the driver as a DWord. TLP values can be written and read from device tags; however some data conversion must be applied for the value to be correct when received by the device. The TLP data type is used frequently throughout the various point type parameters and primarily used within Point Type 99. The following examples are useful for converting a decimal tag value communicated with the device.

DWord TLP Example:

- Decimal Tag Value = 557158
- Binary tag value = 0000 1000 1000 0000 0110 0110

TLP Value Equivalent:

- 0110 0110 = Point Type 102
- 1000 0000 = Logical/Location 128
- 0000 1000 = Parameter 8

DWord TLP Example:

- Decimal Tag Value = 2162786
- Binary tag value = 0010 0001 0000 0000 0110 0010

TLP Value Equivalent:

- 0110 0010 = Point Type 98
- 0000 0000 = Logical/Location 0
- 0010 0001 = Parameter 33

DWord TLP Example:

- Decimal Tag Value = 264905
- Binary tag value = 0000 0100 0000 1010 1100 1001

TLP Value Equivalent:

- 1100 1001 = Point Type 201
- 0000 1010 = Logical/Location 10
- 0000 0100 = Parameter 4

See Also:
Data Type Descriptions
ROC Plus Point Types
Logical / Point Number Details
User-Defined Point Types
Bit Assignments

Address Descriptions

ROC addresses are divided first by point type, logical address, and then by parameter index within the point type. The general format is *T-L.P*, where:

- **T:** The Point Type
- **L:** The Logical Address
- **P:** The Parameter Index

Parameters are blocked together on point type and logical address to a size up to 230 bytes. Some parameters are broken down into individual bits. Those parameters are addressed as *T-L.P:B*, where:

- **B:** The Bit Offset

For example, the address 1-50.3:2 indicates the following:

- **Point Type:** 1
- **Logical Address:** 50
- **Parameter:** 3
- **Bit Offset:** 2

For a detailed listing of all point type parameters, access, data type, length, and description, refer to the ROC device's ROC Protocol User Manual. For more information on ROC addressing, select a link from the list below.

See Also:
Data Type Descriptions
ROC Plus Point Types
Logical / Point Number Details
User-Defined Point Types
Bit Assignments

Logical / Location Details

Within each point type, individual points are referenced by a logical number or a location. The location used by the ROC Plus protocol for point types 101 to 109 is based on a physical input or output (I/O) module and point location. All other point types use a logical number and are numbered in sequence.

Note: The "L" in "TLP" references the logical / location scheme.

Physical Point Numbers 1 to 160

Point types 101 through 109 have location numbers for the field I/O. For diagnostic inputs, the scheme is as follows:

- Location numbers 16 to 160 are assigned to field I/O. For example, if there was an I/O module in slot 1 with 4 points on it, they would be points 16 through 19.
- Location numbers 0 to 15 are assigned to the system I/O. For example, the five diagnostic points in a ROC800-Series would be 0 through 4.

Logical Point Numbers 0 to 127

For all other point types (except 101-109), the logical number is 0 to *x*, where *x* is one less than the total number of points that exist for that point type. For example, the 16 PIDs would be logical numbers 0 through 15.

For a detailed listing of all point types' access, data type, length, and description, refer to the device's ROC Plus protocol user manual.

ROC Plus Point Types

For a detailed listing of all point types' parameters, access, data type, length, and description; refer to the device's ROC Plus protocol user manual.

Point Type	Description
82	Virtual Discrete Outputs
85	HART
91	System Variables
92	Login Parameters
93	License Key Information
94	User C++ Configuration
95	Communication Ports
96	FST Parameters
97	FST Register Tags
98	Soft Point Parameters
99	Configurable Opcode Table
100	Power Control Parameters
101	Discrete Inputs
102	Discrete Outputs
103	Analog Inputs
104	Analog Outputs
105	Pulse Inputs
106	RTD
107	Thermocouple
108	Multi-Variable Sensor
109	System Analog Inputs
110	PID Control Parameters
111	Sampler/Odorizer Parameters
112	Station Parameters
113	Orifice Meter Run Configuration
114	Orifice Meter Run Values
115	Turbine Meter Run Configuration
116	Turbine Meter Run Values
117	Modbus Configuration Parameters
118	Modbus Register to TLP Mapping
119	Modbus Event, Alarm and History Table
120	Modbus Client Modem Configuration
121	Modbus Client Table
122	DS800 Configuration
123	Security -- Group Configuration

Point Type	Description
124	History Segment Configuration
125	History Segment 0 Point Configuration
126	History Segment 1 Point Configuration
127	History Segment 2 Point Configuration
128	History Segment 3 Point Configuration
129	History Segment 4 Point Configuration
130	History Segment 5 Point Configuration
131	History Segment 6 Point Configuration
132	History Segment 7 Point Configuration
133	History Segment 8 Point Configuration
134	History Segment 9 Point Configuration
135	History Segment 10 Point Configuration
136	ROC Clock
137	Internet Configuration Parameters
138	User C++ Host Parameters
139	Smart I/O Module Information
140	Alternating Current Input / Output
141	Advanced Pulse Module
142	History Segment 11
143	History Segment 12
144	Transactional History Configuration
145	Transactional History Point Configuration
177	IEC62591 Commissioned List

User-Defined Point Types

User-Defined Points (UDP) make user program data available to ROCLINK and OPC clients. They are generally used for configuration purposes. When creating a UDP in the server, the server Configuration always sets the data type to default. The data type is later read live from the device.

Important: Users must reinitialize the server after upgrading the user program on a device; otherwise, the server cannot access the new points available in the upgraded user program.

Supported Device Models

All ROC800 Series devices.

Supported User-Defined Point Range

60 to 78
196 to 254

Troubleshooting

To avoid potential issues, users should do the following:

- Verify that the point type is within the supported UDP range.
- If a client attempts to write to a UDP type when no UDP type tags have been read since the server started, the write may fail with a Type Mismatch error. Always complete a read on UDP type tags before a write is attempted.
- Verify that the point type exists in one of the user programs installed on the device.
- Check the Event Log for the following error message, which occurs if the server fails to parse the UDP configuration: **Unable to parse the user-defined point configuration information for point type <point type> on device <device name>**.

User Table Points

User tables, also called Opcode tables, provide the ability to map any Point Type parameters to tables in the device. This driver has the ability to read and write data points in the user tables using Opcodes 10 and 11. The syntax for user table tags is:
user_table-n.m where *n* is the user table number and *m* is the data point or location within that table.

The user table number and location number are zero-based.
For example, the first location in the first user table is: user_table-0.0.

Important: Users must increment the version number of the user table when making changes to the table configuration. Failure to do so when making changes to the table while the server is actively reading user table tags results in bad quality tags or erroneous data.

Supported User Table Point Range

user_table-0.0 to user_table-15.43

Binary Field (BIN) Example

The table below shows an example alarm code from an Analog Input Point Type. This is used to demonstrate how a binary parameter is returned. A "1" in any bit indicates that it is active or enabled.

	Response Code	Bit
Low Alarm	0	0
Low Low Alarm	0	1
High Alarm	0	2
High High Alarm	0	3
Rate Alarm	0	4
Not Used	0	5
Point Fail Alarm	0	6
Scanning Disabled Alarm	1	7

Statistics Items

Statistical items use data collected through additional diagnostics information, which is not collected by default. To use statistical items, Communication Diagnostics must be enabled. To enable Communication Diagnostics, right-click on the channel in the Project View and click **Properties | Enable Diagnostics**. Alternatively, double-click on the channel and select **Enable Diagnostics**.

Channel-Level Statistics Items

The syntax for channel-level statistics items is *<channel>._Statistics*.

Note: Statistics at the channel level are the sum of those same items at the device level.

Item	Data Type	Access	Description
_CommFailures	DWord	Read/Write	The total number of times communication has failed (or has run out of retries).
_ErrorResponses	DWord	Read/Write	The total number of valid error responses received.
_ExpectedResponses	DWord	Read/Write	The total number of expected responses received.
_LastResponseTime	String	Read Only	The time at which the last valid response was received.
_LateData	DWord	Read/Write	The total number of times that a tag is read later than expected (based on the specified scan rate). This value does not increase due to a DNR error state. A tag is not counted as late (even if it was) on the initial read after a communications loss. This is by design.
_MsgResent	DWord	Read/Write	The total number of messages sent as a retry.
_MsgSent	DWord	Read/Write	The total number of messages sent initially.
_MsgTotal	DWord	Read Only	The total number of messages sent (both _MsgSent + _MsgResent).
_PercentReturn	Float	Read Only	The proportion of expected responses (Received) to initial sends (Sent) as a percentage.
_PercentValid	Float	Read Only	The proportion of total valid responses received (_TotalResponses) to total requests sent (_MsgTotal) as a percentage.
_Reset	Bool	Read/Write	Resets all diagnostic counters. Writing to the _Reset Tag causes all diagnostic counters to be reset at this level.
_RespBadChecksum	DWord	Read/Write	The total number of responses with checksum errors.
_RespTimeouts	DWord	Read/Write	The total number of messages that failed to receive any kind of response.
_RespTruncated	DWord	Read/Write	The total number of messages that received only a partial response.
_TotalResponses	DWord	Read Only	The total number of valid responses received (_ErrorResponses + _ExpectedResponses).

Statistical items are not updated in simulation mode *(see device general properties)*.

Device-Level Statistics Items

The syntax for device-level statistics items is *<channel>.<device>._Statistics*.

Item	Data Type	Access	Description
_CommFailures	DWord	Read/Write	The total number of times communication has failed (or has

Item	Data Type	Access	Description
			run out of retries).
_ErrorResponses	DWord	Read/Write	The total number of valid error responses received.
_ExpectedResponses	DWord	Read/Write	The total number of expected responses received.
_LastResponseTime	String	Read Only	The time at which the last valid response was received.
_LateData	DWord	Read/Write	The total number of times that a tag is read later than expected (based on the specified scan rate). This value does not increase due to a DNR error state. A tag is not counted as late (even if it was) on the initial read after a communications loss. This is by design.
_MsgResent	DWord	Read/Write	The total number of messages sent as a retry.
_MsgSent	DWord	Read/Write	The total number of messages sent initially.
_MsgTotal	DWord	Read Only	The total number of messages sent (both _MsgSent + _MsgResent).
_PercentReturn	Float	Read Only	The proportion of expected responses (Received) to initial sends (Sent) as a percentage.
_PercentValid	Float	Read Only	The proportion of total valid responses received (_TotalResponses) to total requests sent (_MsgTotal) as a percentage.
_Reset	Bool	Read/Write	Resets all diagnostic counters. Writing to the _Reset Tag causes all diagnostic counters to be reset at this level.
_RespBadChecksum	DWord	Read/Write	The total number of responses with checksum errors.
_RespTimeouts	DWord	Read/Write	The total number of messages that failed to receive any kind of response.
_RespTruncated	DWord	Read/Write	The total number of messages that received only a partial response.
_TotalResponses	DWord	Read Only	The total number of valid responses received (_ErrorResponses + _ExpectedResponses).

Statistical items are not updated in simulation mode *(see device general properties)*.

EFM Pointer Rollback

EFM Pointer rollback allows the user to manually change the EFM pointer value to an earlier position in the archive using system tags provided by the EFM Exporter. Cached EFM pointer rollback functionality is supported for the Hourly History, Daily History, Alarm, and Event archives.

Notes:

- If the driver's cached EFM pointer is invalid, as is the case if records have never been uploaded or the **Clear Cache on Next Upload** setting is enabled prior to the EFM upload, the rollback value is applied to the physical device's current archive pointer.
- When uploading EFM data for all meters in a device and rollback is required for the alarm and/or event archives, it is recommended to only set the _AlarmsRollback and/or _EventsRollback system tags for one meter under the device and initiate the poll via the _Poll system tag for just that meter using the meter-level system tag, not the poll-group-level system tag. Setting the _AlarmsRollback and _EventsRollback system tags for all meters under the device or initiating the poll via the poll group level system tag can result in duplicate alarm and event records in the EFM output files for the device's meters. This behavior is the result of the driver employing a more efficient method of uploading EFM alarm and event data from the device. The device stores alarms for all meters in a single alarm archive and events for all meters in a single event archive. When the driver uploads alarms from the alarm archive, it stores the alarms for each meter in separate caches to avoid uploading the same records multiple times. Setting the _AlarmsRollback system tag for multiple meters under a device causes the driver to read the same records multiple times, each time storing those records in the per meter record caches which could result in duplicate entries. ROC800L devices have an additional weights and measures event log that is uploaded along with the standard event log when the server is configured to upload EFM events for a ROC800L liquid meter. Writing a rollback value to the _EventsRollback tag prior to initiating an EFM poll causes the rollback value to be applied when uploading the standard event log as well as when uploading the weights and measures event log. This can result in a larger number of event records in the EFM output file than the value that was written to the _EventsRollback tag. For example, if the standard event log and the weights and measures event log both have 100 records and a value of three is written to the _EventsRollback tag; the driver uploads the newest three records from the standard event log, then uploads the newest three records from the weights and measures event log, resulting in six new event records in the EFM output file.
- Non-meter events are events logged by an EFM device that are not specific to a meter. The number of event records returned when using the EFM Pointer Rollback feature varies depending on the Non-Meter Events configuration setting and the presence of non-meter events in the event log.

  Examples

  - If the newest three records in a log with 50 records are non-meter events, an EFM poll is done with the value written to the _EventsRollback tag of five and Non-Meter Events configuration setting is set to Yes; the result includes five new records in the EFM output file.
  - If the newest three records in a log with 50 records are non-meter events, an EFM poll is done with the value written to the _EventsRollback tag of five and Non-Meter Events configuration setting is set to No; the result has two new records in the EFM output file. A rollback value of five causes the driver to read the three newest non-meter event records, but they are not added to the EFM output file because the meter is configured to not accept non-meter events.

For more information, refer to Non-Meter Events in the EFM Meter properties.

For more information, refer to the EFM Exporter help file.

Event Log Messages

The following information concerns messages posted to the Event Log pane in the main user interface. Consult the OPC server help on filtering and sorting the Event Log detail view. Server help contains many common messages, so should also be searched. Generally, the type of message (informational, warning) and troubleshooting information is provided whenever possible.

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 server 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 server application.

Error importing CSV tag record <record number>: Address <address> is out of range for the specified device or register.

Error Type:
Warning

Possible Cause:
An imported tag address specifies 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 file being imported.

Unable to generate a tag database for device <device>. Reason: <Error reason>.

Error Type:
Warning

Possible Cause:
The error occurred due to the specified error reason.

Solution:
The solution depends on the specified error reason.

See Also:
Error Reasons

Unable to generate a tag database for device <device>. Reason: Auto tag generation.

Error Type:
Serious

Possible Cause:

1. The connection between the device and the host PC is intermittent.
2. The communication parameters for the connection are incorrect.

Solution:

1. Verify the cabling between the PC and the device.
2. Verify that the specified communication parameters match those of the device.

Unable to generate a tag database for device <device>. Reason: Error while reading from import file.

Error Type:
Warning

Possible Cause:

1. The tag import file (*.800) is corrupt.
2. The specified file was not created using the ROCLINK 800 software.

Solution:

1. Ensure that the project is pointing to the correct import file.
2. Re-create the import file using the ROCLINK 800 software and then re-try the import.

Unable to generate a tag database for device <device>. Reason: Error while reading from ROC system file.

Error Type:
Warning

Possible Cause:

1. The ROC system file (*.mdb) is corrupt.
2. The specified file was not created using the ROCLINK 800 software.

Solution:

1. Ensure that the project is pointing to the correct ROC system file.
2. Re-install the ROCLINK 800 software to re-install the system file. Then re-try the import.

Unable to generate a tag database for device <device>. Reason: Failed to open record set.

Error Type:
Warning

Possible Cause:

1. The project file is corrupt or does not exist.
2. The location of the ROC.MDB and/or ROCLINK.MDW files have been specified incorrectly.

Solution:
In the server project, right-click on the device and select **Properties**. Open the **Tag Import Settings** tab to check the name of the project file to import.

See Also: Tag Generation and Tag Import Settings

Unable to generate a tag database for device <device>. Reason: Import file <file name> not found.

Error Type:
Warning

Possible Cause:
The import file cannot be found.

Solution:
Ensure that the tag import file (*.800) is present in the location specified in the Tag Import Settings tab of device properties. This file must be accessible to the server runtime.

Unable to generate a tag database for device <device>. Reason: Input file is corrupt.

Error Type:
Warning

Possible Cause:
The import file is corrupt.

Solution:
In the server project, right-click on the device and select **Properties**. Open the **Tag Import Settings** tab to review the settings and check the import file. If necessary, re-export the project file from within ROCLINK800.

See Also: Tag Generation and Tag Import Settings

Unable to generate a tag database for device <device>. Reason: Input file not found.

Error Type:
Warning

Possible Cause:
The import file cannot be located.

Solution:
In the server project, right-click on the device and select **Properties**. Open the **Tag Import Settings** tab to review the name of the project file to import.

See Also: Tag Generation and Tag Import Settings

Unable to generate a tag database for device <device>. Reason: Low memory resources.

Error Type:
Warning

Possible Cause:
The memory required for Automatic Tag Generation could not be allocated. The process is cancelled.

Solution:
Close any unused applications and/or increase the amount of virtual memory. Try again.

Unable to generate a tag database for device <device>. Reason: ROC system file <file name> not found.

Error Type:
Warning

Possible Cause:
The ROC system file cannot be found.

Solution:
Ensure that the ROC system file (* .mdb) is present in the location specified in the Tag Import Settings tab of device properties. This file must be accessible to the server runtime.

Unable to generate a tag database for device <device>. Reason: System DB file <file name> not found.

Error Type:
Warning

Possible Cause:
The system database (DB) file cannot be found.

Solution:
Ensure that the system database (DB) file (* .mdw) is present in the location specified in the Tag Import Settings tab of device properties. This file must be accessible to the server runtime.

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

Error Type:
Warning

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

Possible Cause:

1. A permission error was encountered when the EFM pointer cache was read.
2. The EFM pointer cache file is corrupt.

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

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

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

Error Type:
Warning

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

Possible Cause:
1. The disk is full.
2. A permission error was encountered when the EFM pointer cache was written.

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

For more information, refer to the extended error.

Block read for point type <point type>, logical address <logical address>, parameter range <start parameter - end parameter> of device <device name> failed. <Error reason>.

Error Type:
Serious

Possible Cause:
The error occurred due to the specified error reason.

Solution:
The solution depends on the specified error reason.

See Also:
Error Reasons

Block read for point type <point type>, logical address <logical address>, parameter range <start parameter - end parameter> of device <device name> failed. Parameters are not in the loaded UDP configuration.

Error Type:
Serious

Possible Cause:
The user program that is associated with the specified parameters has been upgraded to a newer version.

Solution:
Reinitialize the server to access the new parameters available in the upgraded user program.

Device <device> responded with error. (Tag <tag address>) - Details: <error code>.

Error Type:
Serious

Possible Cause:
1. The connection between the device and the host PC is intermittent.
2. The communication parameters for the connection are incorrect.
3. The value written is out of range.
4. The write was performed while in an incorrect setup area.

Solution:
1. Check the cabling between the PC and the device.
2. Verify that the specified communication parameters match those of the device.

See Also: Setup

Failed to obtain data block for point type = <point type>, logical address = <address>, starting parameter = <starting parameter>, ending parameter <ending parameter> for device <device>. Error = <ROC error code>.

Error Type:
Serious

Possible Cause:
1. An invalid tag address is used for the point in block.
2. The device is not responding.

Solution:
1. Consult the ROC error code reference for further information regarding the error code.
2. Verify the cabling between the PC and the device.

3. Confirm that all tags within this block exist on the device.

See Also:
ROC Plus Error Codes

Failed to write data for point type = <point type>, logical address = <address>, parameter = <parameter> for device <device>. Error = <error code>.

Error Type:
Serious

Possible Cause:

1. The address is incorrect.
2. The unit does not support the particular address point.
3. The privileges for the logged-in user do not permit this operation.

Solution:

1. Consult the ROC error code reference for further information regarding the error code.
2. Correct the address.
3. Confirm that the address is supported by the controller in use.
4. Supply an operator identification with sufficient privileges.

See Also:
ROC Plus Error Codes

Multiple batches completed since the previous batch history poll for meter <meter> on device <device>. The last uploaded batch ticket number is <last ticket number> and the current batch ticket number is <current ticket number>.

Error Type:
Warning

Possible Cause:
Multiple batches have completed since the last time EFM batch data was polled for the specified meter.

Solution:
Fisher ROC+ devices only store the latest complete batch and current batch data. The meter should be polled at a rate such that there is only a single batch completed between polls.

Operator identification failed for device <device name>. <Error reason>.

Error Type:
Serious

Possible Cause:
The error occurred due to the specified error reason.

Solution:
The solution depends on the specified error reason.

See Also:
Error Reasons
Operator Identification

Read for point type <point type>, logical address <logical address>, parameter number <parameter number> of device <device name> failed. <Error reason>.

Error Type:
Serious

Possible Cause:
The error occurred due to the specified error reason.

Solution:
The solution depends on the specified error reason.

See Also:
Error Reasons

ROC initialization error: Unable to read general configuration.

Error Type:
Serious

Possible Cause:
The driver may not be receiving a response from the device.

Solution:
1. Verify the device is physically connected and powered on.
2. Check that the COM port is working and configured properly at the channel level (in the server).
3. Check the device-level operator identification and address specification settings and verify that they are correct.

See Also:
Operator Identification
Communication Specification

ROC initialization error: Unable to retrieve I/O map.

Error Type:
Serious

Possible Cause:
Access to the I/O map has been restricted for the current user.

Solution:
Check the operator identification settings (such as, username, password, and access level) and verify that they are correct.

See Also:
Operator Identification

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

Error Type:
Warning

Possible Cause:
1. The driver was unable to create the specified file directory.
2. The driver was unable to access the specified file.

Solution:
1. Verify that the disk has sufficient disk space.
2. Verify user permissions for the specified file directory.

The rollback value has been adjusted to match the archive size because it exceeded the size of the archive | Adjusted Rollback = <size>, Archive = <name>.

Error Type:
Warning

Possible Cause:
The rollback value entered is larger than the archive size.

Solution:
Enter a value less than or equal to the archive size to prevent this message from being displayed.

The username or password supplied was not accepted. Error = 6.

Error Type:
Serious

Possible Cause:
An access level has been enabled on the device but not in the driver.

Solution:
Check the operator identification settings and verify the Enable Access Level checkbox is yes/enabled.

See Also:
Operator Identification

The username or password supplied was not accepted. Error = 63.

Error Type:
Serious

Possible Cause:
The access level that has been enabled on the device is lower than the operator's access level.

Solution:
Check the operator identification settings and verify the operator's access level is less than or equal to the access level enabled in the device.

See Also:
Operator Identification

Time synchronization with device <device name> failed. <Error reason>.

Error Type:
Serious

Possible Cause:
The error occurred due to the specified error reason.

Solution:
The solution depends on the specified error reason.

See Also:
Error Reasons

Write failed with error code <error code> for the following tag(s) in device <device name>:<tag list>.

Error Type:
Serious

Possible Cause:
The ROC device responded with an error code.

Solution:
Consult the error code reference for further information regarding the error code.

See Also:
ROC Plus Error Codes

Write for the following tags of device <device name> failed: <tag list>. <Error reason>.

Error Type:
Serious

Possible Cause:
The error occurred due to the specified error reason.

Solution:
The solution depends on the specified error reason.

See Also:
Error Reasons

Write request rejected on read-only item reference <channel name> <device name> <address>.

Error Type:
Warning

Possible Cause:
The driver attempted to write to read-only data in the ROC controller.

Solution:
Do not attempt to write to read-only points.

Note:
In some situations, the Automatic Tag Generation process identifies read-only data as read/write, based on the configuration that the driver retrieved from the ROC controller and the ROC specification. Nonetheless, the ROC controller itself is the final authority on whether data is writable. *For more information, refer to the controller's documentation.*

<Device> may have incomplete history configured for meter <meter>.

Error Type:
Warning

Possible Cause:
The EFM History that was uploaded for the meter is missing one or more fields of data.

Solution:
Check the EFM output for missing data. If necessary, configure the RTU's EFM History using ROCLINK 800.

Device <device name> is not responding.

Error Type:
Serious

Possible Cause:

1. The connection between the device and the host PC is intermittent.
2. The communication parameters for the connection are incorrect.
3. 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 device.
2. Verify that the specified communication parameters match those of the device.
3. Increase the Request Timeout setting so that the entire response can be handled.

EFM <type> upload for device <device name> meter <meter name> failed. Framing error.

Error Type:
Warning

Possible Cause:
An EFM upload of the specified type could not be completed due to the specified reason.

Solution:
Resolve the issue. Then re-attempt the EFM upload.

EFM <type> upload for meter <meter name> on device <device name>. Device responded with error code <error code>.

Error Type:
Error

Possible Cause:
If the error code is '1' and the <type> is event, then the wrong model is selected.

Solution:
This occurs if the liquid model is selected when it is in fact a gas model. Correct the model in the device properties group to be a gas model. Then re-attempt the EFM upload.

Resetting the EFM cache for device <device>.

Error Type:
Informational

Possible Cause:
The EFM cache was successfully cleared for the specified device.

Solution:
N/A

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 communication parameters for the serial connection are incorrect.

Solution:
1. Verify the cabling between the PC and the device.
2. Verify that the specified communication 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 in the Channel Properties.

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

Error opening COMn.

Error Type:
Fatal

Possible Cause:
The specified COM port could not be opened due to 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.

Block read for user table <table number>, location range <start location> - <end location> of device <device name> failed. Device returned error code <error code>.

Error Type:
Serious

Possible Cause:
The error occurred for the reason specified by the error code.

Solution:
The solution depends on the specified error code.

See Also:
ROC Plus Error Codes

Block read for user table <table number>, location range <start location> - <end location> of device <device name> failed. Framing error.

Error Type:
Serious

Possible Cause:
There may be an error in the device configuration or the server received a malformed packet.

Solution:
Troubleshoot the device configuration.

Block read for user table <table number>, location range <start location> - <end location> of device <device name> failed. Locations are not configured in the user table.

Error Type:
Serious

Possible Cause:
There is an error in the device configuration. At least one location in the specified range is undefined.

Solution:
Define missing location(s) in the specified user table.

Error parsing user table configuration on device <device name>. User table <table number> contains an invalid point type, location, or parameter in table location <location address>.

Error Type:
Serious

Possible Cause:
There is an error in the device configuration. The specified user table is configured with at least one invalid TLP.

Solution:
Configure the specified user table with valid TLP entries.

Error parsing user table configuration on device <device name>. User table <table number> contains an invalid user-defined point type, location, or parameter in table location <location address>.

Error Type:
Serious

Possible Cause:
There is an error in the device configuration. The specified user table is configured with at least one invalid user-defined point.

Solution:
Configure the specified user table with valid TLP and/or UDP entries.

Read for user table <table number>, location <location address> of device <device name> failed. Device returned error code <error code>.

Error Type:
Serious

Possible Cause:
The error occurred for the reason specified by the error code.

Solution:
The solution depends on the specified error code.

See Also:
ROC Plus Error Codes

Read for user table <table number>, location <location address> of device <device name> failed. Framing error.

Error Type:
Serious

Possible Cause:
There may be an error in the device configuration or the server received a malformed packet.

Solution:
Troubleshoot the device configuration.

Read for user table <table number>, location <location address> of device <device name> failed. Location is not configured in the user table.

Error Type:
Serious

Possible Cause:
There is an error in the device configuration. The specified user table location is undefined.

Solution:
Define the missing table location in the device.

User table configuration upload on device <device name> failed. Device not responding.

Error Type:
Serious

Possible Cause:
1. The connection between the device and the host PC is intermittent.
2. The communication parameters for the Ethernet connection are incorrect.
3. 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 network between the PC and the device.
2. Verify that the specified communication parameters match those of the device.

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

User table configuration upload on device <device name> failed. Device responded with error code <error code>.

Error Type:
Serious

Possible Cause:
The error occurred for the reason specified by the error code.

Solution:
The solution depends on the specified error code.

See Also:
ROC Plus Error Codes

User table configuration upload on device <device name> failed. Framing error.

Error Type:
Serious

Possible Cause:
There may be an error in the device configuration or the server received a malformed packet.

Solution:
Troubleshoot the device configuration.

User table configuration upload on device <device name> failed. Internal error.

Error Type:
Serious

Possible Cause:
Inadequate system resources.

Solution:
Free system resources and reinitialize the server. If trouble persists, please contact Technical Support.

Block read for point type <point type>, logical address <logical address>, parameter range <start parameter - end parameter> of device <device name> failed. Parameters are not in the loaded UDP configuration.

Error Type:
Serious

Possible Cause:

The user program that is associated with the specified parameters has been upgraded to a newer version.

Solution:
Reinitialize the server to access the new parameters available in the upgraded user program.

Read for point type <point type>, logical address <logical address>, parameter number <parameter> of device <device name> failed. Parameter is not in the loaded UDP configuration.

Error Type:
Serious

Possible Cause
The user program that is associated with this parameter has been upgraded to a newer version.

Solution:
Reinitialize the server to access the new parameters available in the upgraded user program.

Unable to parse the user-defined point configuration information for point type <point type> on device <device name>.

Error Type:
Serious

Possible Cause
There was unexpected data in the UDP configuration read from the device.

Solution:
This error requires further troubleshooting. Please contact Technical Support.

User-defined point configuration upload for point type <point type>, logical address <logical address>, parameter number <parameter> of device <device name> failed. <Error reason>.

Error Type:
Serious

Possible Cause
The error occurred due to the specified reason.

Solution:
The solution depends on the specified error reason.

See Also:
Error Reasons

ROC Plus Error Codes

Note: OpCode 255 is an error message indicator that returns an error code.

Error Code	Description
1	Invalid Opcode request
2	Invalid parameter number
3	Invalid logical number
4	Invalid point type
5	Received too many data bytes
6	Received too few data bytes
12	Obsolete (reserved, but not used)
13	Outside valid address range
14	Invalid history request
16	Invalid event entry
17	Requested too many alarms
18	Requested too many events
19	Write to read-only parameter*
20	Security error
21	Invalid security login
22	Invalid store and forward path
24	History configuration in progress
25	Invalid parameter range
29	Invalid 1 day history index request
30	Invalid history point
31	Invalid min./max. request
32	Invalid TLP
33	Invalid time
34	Illegal Modbus range
53	ROC file system error - possibly the point type does not exist
63	Requested access level too high

* Exception for Opcode 166, which can have multiple parameters. Some parameters may be read only.

Error Reasons

Error Reason	Possible Cause	Solution
Device not responding	*For more information, see* ***Device <device name> is not responding***.	*For more information, see* ***Device <device name> is not responding***.
Device responded with error code	The device responded with an error code.	*For more information, see* ***ROC Plus Error Codes***.
Framing error	The response packet from the device has data fields that are not as per the protocol.	This error is very rare. If encountered, users should check with the manufacturer to ensure that the device is consistent with the protocol.
Operator identification error	The operator identification login (with user ID and password) failed.	Refer to the Event Log message that corresponds to the operator identification failure.

Index

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

W