

# Fisher ROC Serial Driver

© 2019 PTC Inc. All Rights Reserved.

# Table of Contents

<b>Fisher ROC Serial Driver</b> .....	<b>1</b>
<b>Table of Contents</b> .....	<b>2</b>
Fisher ROC Serial Driver .....	6
Overview .....	7
<b>Channel Setup</b> .....	<b>7</b>
Channel Properties — General .....	15
Channel Properties — Serial Communications .....	16
Channel Properties — Write Optimizations .....	18
Channel Properties — Advanced .....	19
Channel Properties — Communication Serialization .....	20
<b>Device Setup</b> .....	<b>21</b>
Device Properties — General .....	23
Operating Mode .....	24
Device Properties — Scan Mode .....	25
Device Properties — Timing .....	25
Device Properties — Auto-Demotion .....	26
Device Properties — Tag Generation .....	27
Device Properties — Time Synchronization .....	29
Device Properties — Tag Import Settings .....	30
Device Properties — Communication Specification .....	31
Device Properties — Operator Identification .....	32
Device Properties — EFM Meters .....	32
EFM Attribute Mapping .....	33
Device Properties — Redundancy .....	36
<b>Data Types Description</b> .....	<b>36</b>
<b>TLP Data Type Conversion Process and Examples</b> .....	<b>37</b>
<b>Address Descriptions</b> .....	<b>38</b>
ROC Point Types .....	39
Logical / Point Number Details .....	43
User-Defined Point Types .....	44
User Table Points .....	44
Bit Assignments .....	45
Statistics Items .....	45
<b>Error Descriptions</b> .....	<b>48</b>
Address Validation .....	48

Address <address> is out of range for the specified device or register. ....	48
Data type <type> is not valid for device address <address>. ....	48
Device address <address> contains a syntax error. ....	49
Device address <address> is not supported by model <model name>. ....	49
Device address <address> is read only. ....	49
Missing address. ....	49
Automatic Tag Database Generation Messages .....	50
Error importing CSV tag record <record number>: Address <Address> is out of range for the specified device or register. ....	50
Unable to generate a tag database for device <device>. Reason: <error reason>. ....	50
Unable to generate a tag database for device <device>. Reason: Auto tag generation failed. Device is not responding. ....	51
Unable to generate a tag database for device <device>. Reason: Error while reading from import file. ....	51
Unable to generate a tag database for device <device>. Reason: Error while reading from ROC system file. ....	51
Unable to generate a tag database for device <device>. Reason: Failed to open record set. ....	52
Unable to generate a tag database for device <device>. Reason: Import file <file name> not found. ....	52
Unable to generate a tag database for device <device>. Reason: Input file is corrupt. ....	52
Unable to generate a tag database for device <device>. Reason: Input file not found. ....	53
Unable to generate a tag database for device <device>. Reason: Low memory resources. ....	53
Unable to generate a tag database for device <device>. Reason: ROC system file <file name> not found. ....	53
Unable to generate a tag database for device <device>. Reason: System DB file <file name> not found. ....	54
Device-Specific Messages .....	54
<Device name> - Failed to read EFM pointer file. <Extended error>. ....	54
<Device name> - Failed to write EFM pointer file. <Extended error>. ....	55
Block read for point type <point type>, logical address <logical address>, parameter range <start parameter - end parameter> of device <device name> failed. <Error reason>. ....	55
Device <device> responded with error. (Tag <tag address>)-Details: <error code>. ....	56
EFM hourly history upload failed for device <device>. Device responded with error code 14. ....	56
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>. ....	56
Failed to write data for point type = <point type>, logical address = <address>, parameter = <parameter> for device <device>. Error = <error code>. ....	57
Operator identification failed for device <device name>. <Error reason>. ....	57
Read for point type <point type>, logical address <logical address>, parameter number <parameter number> of device <device name> failed. <Error reason>. ....	58
ROC initialization error: Unable to read general configuration. ....	58

ROC initialization error: Unable to retrieve I/O map. ....	58
Serialization of EFM data to temporary file <file name> failed. Reason: <file I/O error>. ....	59
The username or password supplied was not accepted. Error = 6. ....	59
The username or password supplied was not accepted. Error = 63. ....	59
Time synchronization with device <device name> failed. <Error reason>. ....	59
Write for the following tags of device <device name> failed: <tag list>. <Error reason>. ....	60
Write request rejected on read only item reference <channel name> <device name> <address>. .	60
Device Status Messages .....	60
Device <device name> is not responding. ....	61
EFM <type> upload failed for device <device>. <Reason>. ....	61
Resetting the EFM cache for device <device>. ....	61
Serial Communications .....	61
Communications error on <channel name> [<error mask>]. ....	62
COMn does not exist. ....	62
COMn is in use by another application. ....	62
Error opening COMn. ....	63
Unable to set comm parameters on COMn. ....	63
User Configurable Table Messages .....	63
Block read for user table <table number>, location range <start location> - <end location> of device <device name> failed. Device returned error code <error code>. ....	64
Block read for user table <table number>, location range <start location> - <end location> of device <device name> failed. Framing error. ....	64
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. ....	64
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>. ....	65
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>. ....	65
Read for user table <table number>, location <location address> of device <device name> failed. Device returned error code <error code>. ....	65
Read for user table <table number>, location <location address> of device <device name> failed. Framing error. ....	65
Read for user table <table number>, location <location address> of device <device name> failed. Location is not configured in the user table. ....	66
User table configuration upload on device <device name> failed. Device not responding. ....	66
User table configuration upload on device <device name> failed. Device responded with error code <error code>. ....	66
User table configuration upload on device <device name> failed. Framing error. ....	67
User table configuration upload on device <device name> failed. Internal error. ....	67
User-Defined Point Messages .....	67

---

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. ....	68
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. ....	68
Unable to load user-defined point configuration for point type <point type>, logical address <logical address>, parameter number <parameter> of device <device name>. The number of parameters for this point is 0. ....	68
Unable to parse the user-defined point configuration information for point type <point type> on device <device name>. ....	68
User-defined point configuration upload for point type <point type>, logical address <logical address>, parameter number <parameter> of device <device name> failed. <Error reason>. ....	69
ROC Error Codes .....	70
Error Reasons .....	74
<b>Index</b> .....	<b>76</b>

## Fisher ROC Serial Driver

---

Help version 1.090

### CONTENTS

#### Overview

What is the Fisher ROC Serial Driver?

#### Channel Setup

How do I configure channels for use with this driver?

#### Device Setup

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

#### Data Types Description

What data types does this driver support?

#### Tag Generation

How can tags be automatically created for this driver?

#### Address Descriptions

How do I address a data location on a Fisher ROC Serial device?


#### Error Descriptions


What error messages does the Fisher ROC Serial Driver produce?

---

## Overview

The Fisher ROC Serial Driver provides a reliable way to connect Fisher ROC Serial controllers to OPC client applications; including HMI, SCADA, Historian, MES, ERP, and countless custom applications.

 For more information on available Opcodes, Point Types, Parameters, and Opcode 255 Error Codes, refer to the ROC Protocol User Manual.

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

## Supported Devices

FloBoss 100 Series  
FloBoss 407  
FloBoss 500 Series  
ROC 300 Series-ROCPAC  
ROC 300 Series-FlashPAC  
RegFlo

## Supported Protocol


Remote Operation Controllers (ROC)


## Maximum Number of Channels and Devices

The maximum number of supported channels is 1024. The maximum number of devices supported per channel is 255.

## Ethernet Encapsulation

This driver supports Ethernet Encapsulation, which allows communications with serial devices attached to an Ethernet network using a Serial-to-Ethernet server. It may be enabled through the Communications tab in Channel Properties. For more information, refer to "Channel Properties — Ethernet Encapsulation" in the server's help documentation.

 **Important:** Some FloBoss 100 Series devices (such as FloBoss 107) fail to respond to TCP requests that use Explicit Congestion Notification (ECN). To communicate with these devices, the global TCP/IP ECN parameter must be disabled.

 **Note:** This driver does not support Report by Exception.

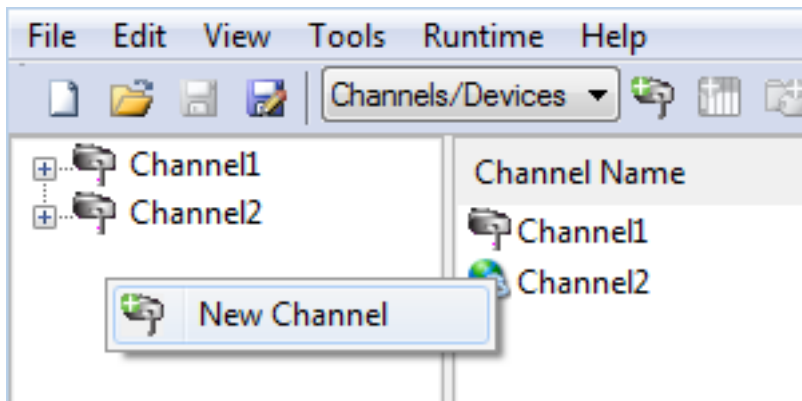
---

## Channel Setup

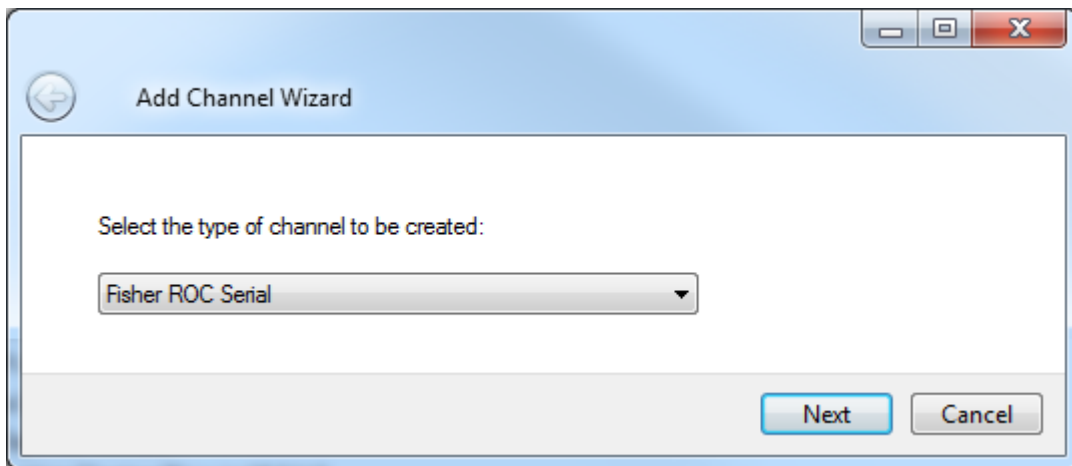
The Fisher ROC Serial Driver supports Communication Serialization, which specifies whether data transmissions should be limited to one channel at a time. For more information, refer to "Channel Properties — Advanced" in the server help file. The maximum number of supported channels is 1024.

To create a new channel:

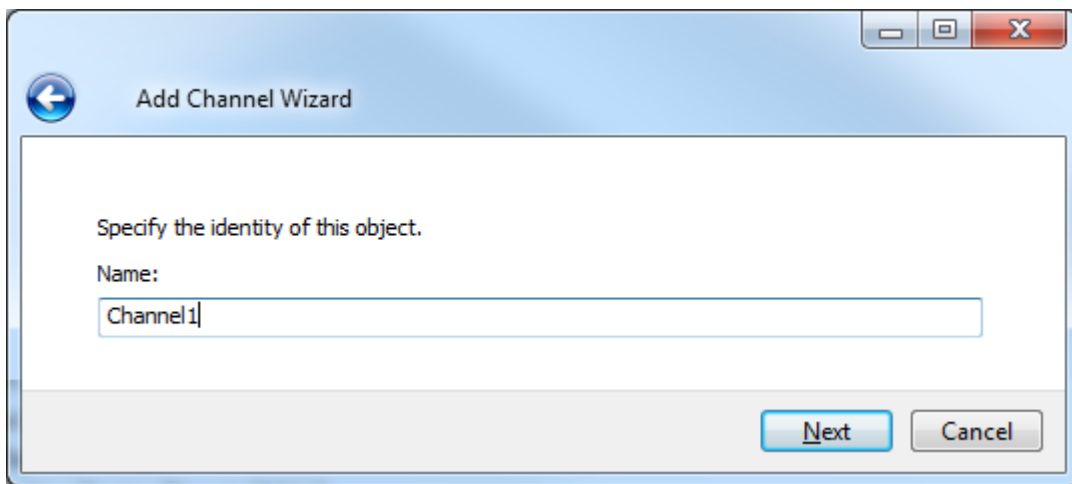
1. In the Project View, right-click and choose **New Channel**.



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



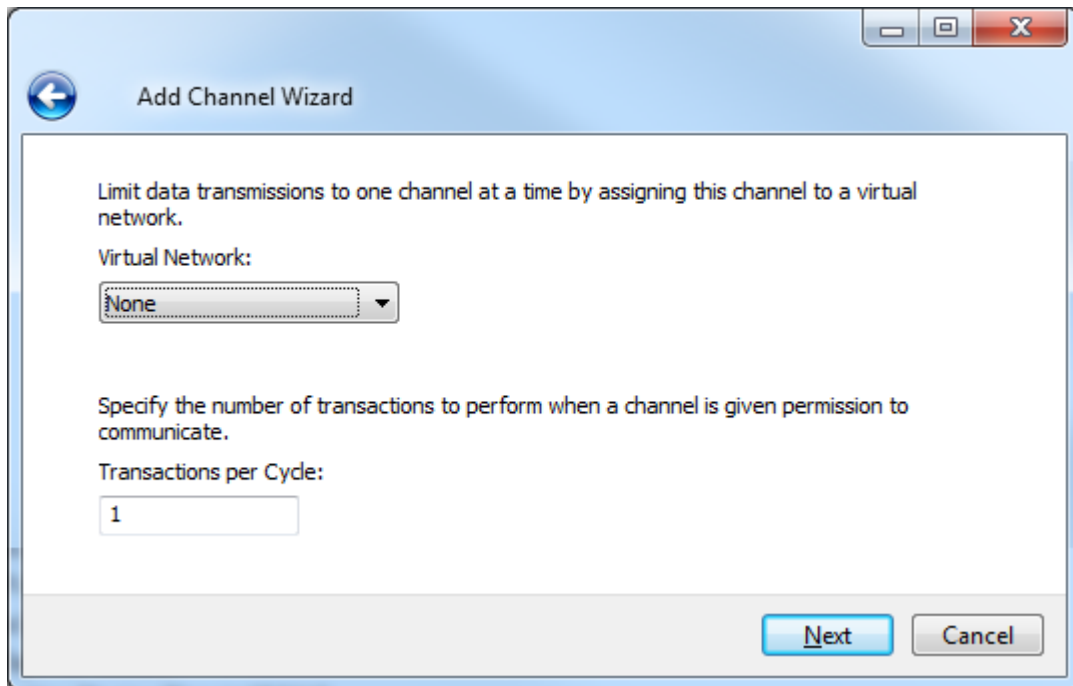
3. Click **Next** >.
4. Accept the default channel name or enter a name for the new channel.



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**.

Select the hardware device type for data communications.

Physical Medium:  
COM Port

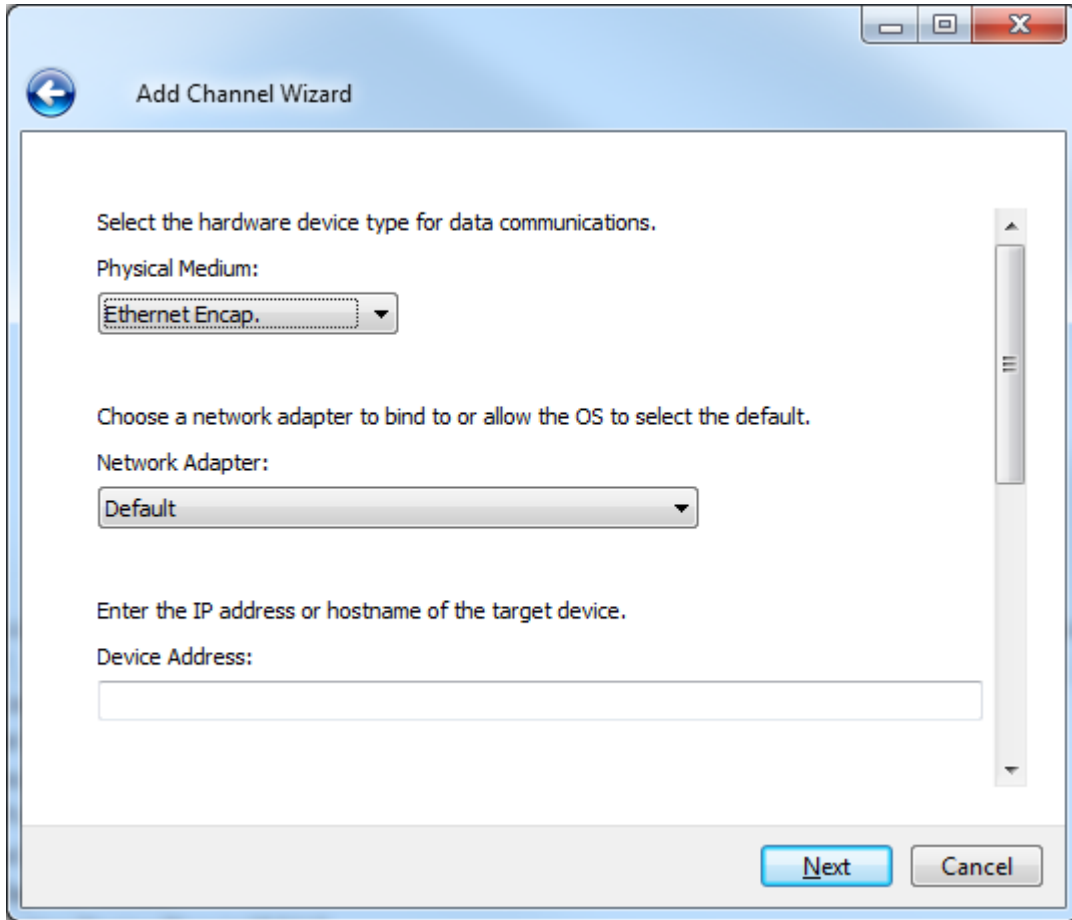
Specify the physical port number.

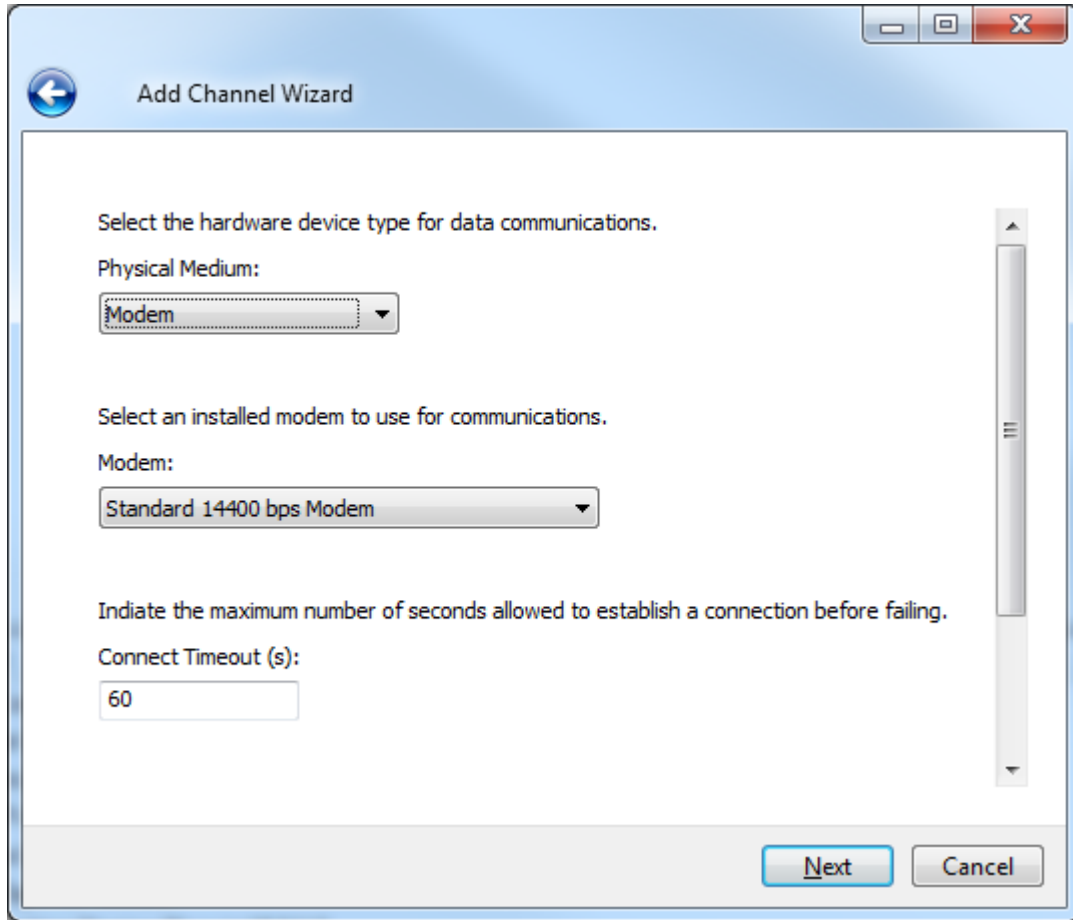
COM ID:  
1

Select the communications speed of the hardware in bits per second.

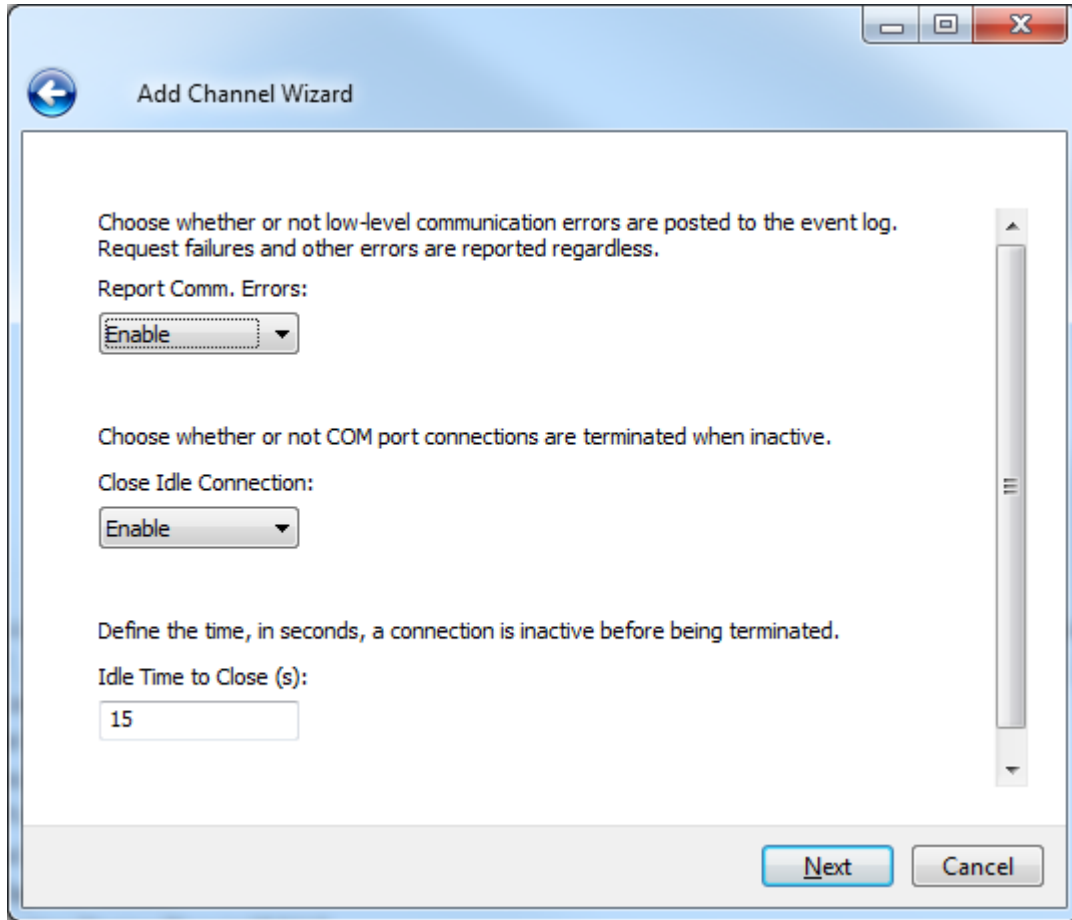
Baud Rate:  
9600

Next Cancel

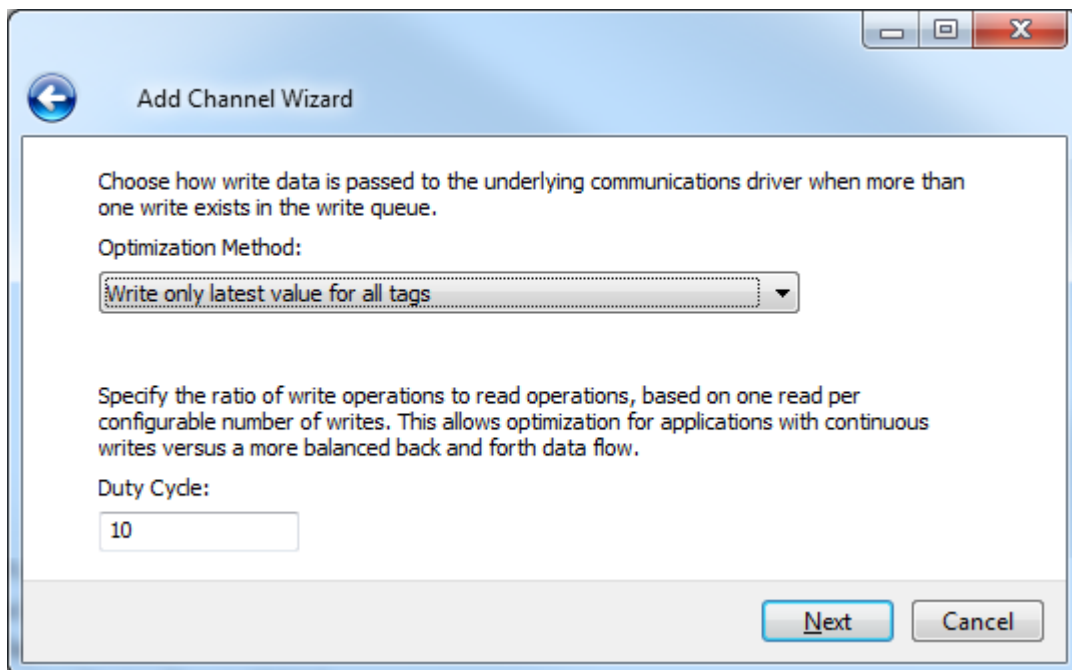




9. Configure communication based on the environment and click **Next >**.
10. Select **Close connection when idle** to reduce traffic and specify the number of seconds before the connection is terminated.



11. Click **Next** >.
12. Configure channel writes.

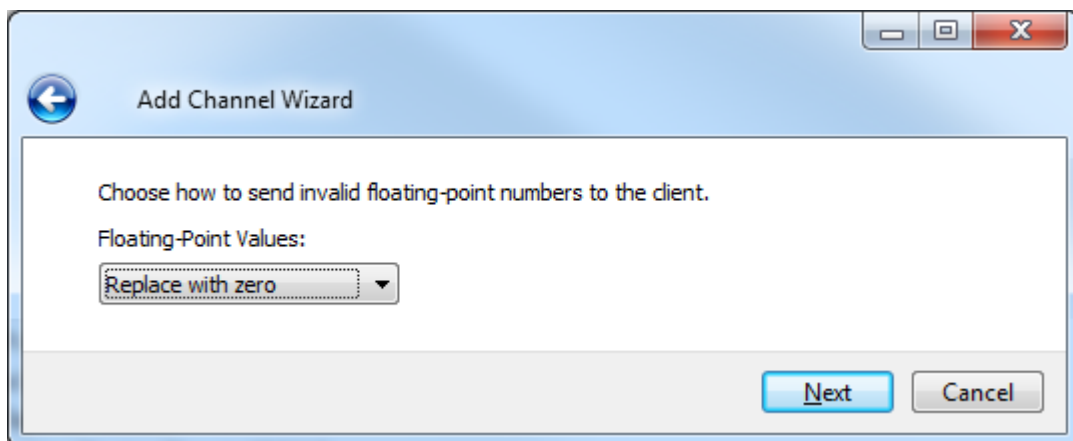


**Optimization Method** - Accept the default or change the selection.

- Write all values for all tags (most data, most network traffic)
- Write only the latest value for non-Boolean tags (least data, least network traffic)
- 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. The range is 1-10.

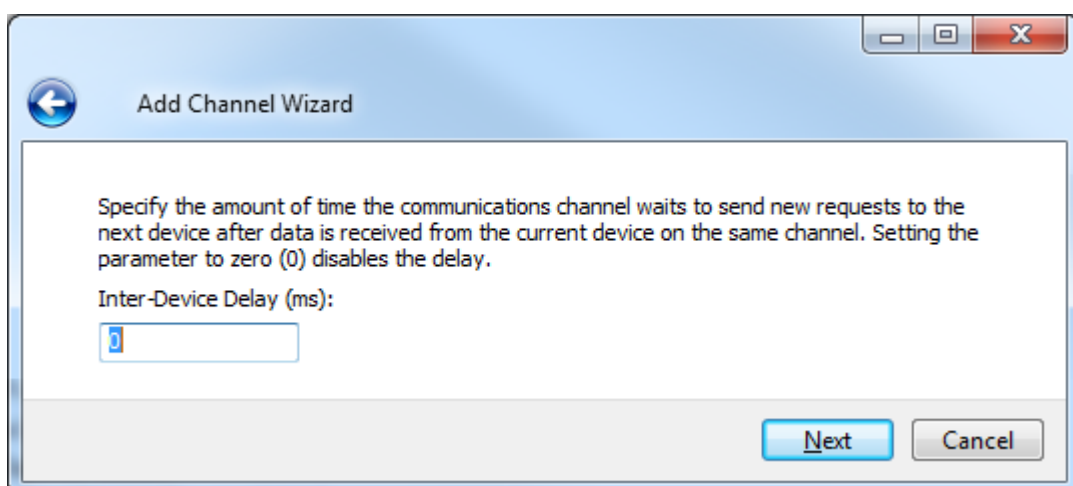
13. Click **Next** >.
14. Configure how non-normalized values will be handled.



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

Property Groups <b>General</b> Write Optimizations Advanced	<table border="1"> <tr> <td colspan="2">[-] <b>Identification</b></td> </tr> <tr> <td>Name</td> <td></td> </tr> <tr> <td>Description</td> <td></td> </tr> <tr> <td>Driver</td> <td></td> </tr> <tr> <td colspan="2">[-] <b>Diagnostics</b></td> </tr> <tr> <td>Diagnostics Capture</td> <td>Disable</td> </tr> </table>	[-] <b>Identification</b>		Name		Description		Driver		[-] <b>Diagnostics</b>		Diagnostics Capture	Disable
[-] <b>Identification</b>													
Name													
Description													
Driver													
[-] <b>Diagnostics</b>													
Diagnostics Capture	Disable												

### Identification

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

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

**Driver:** Selected 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. With this in mind, changes to the properties should not be made once a large client application has been developed. Utilize the User Manager to prevent operators from changing properties and restrict access rights to server features.

### Diagnostics

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

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

For more information, refer to "Communication Diagnostics" 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 the User Manager to restrict access rights to server features, as changes made to these properties can temporarily disrupt communications.

Property Groups		
General		
<b>Serial Communications</b>		
Write Optimizations		
Advanced		
	<b>Connection Type</b>	
	Physical Medium	COM Port
	<b>Serial Port Settings</b>	
	COM ID	39
	Baud Rate	19200
	Data Bits	8
	Parity	None
	Stop Bits	1
	Flow Control	RTS Always
	<b>Operational Behavior</b>	
	Report Communication Errors	Enable
	Close Idle Connection	Enable
	Idle Time to Close (s)	15

### 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 "How To... Use 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

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

Property Groups	<input checked="" type="checkbox"/> <b>Write Optimizations</b>	
General	Optimization Method	Write Only Latest Value for All Tags
<b>Write Optimizations</b>	Duty Cycle	10

## 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.

Property Groups	<input type="checkbox"/> <b>Non-Normalized Float Handling</b>	
General	Floating-Point Values	Replace with Zero
Write Optimizations	<input type="checkbox"/> <b>Inter-Device Delay</b>	
<b>Advanced</b>	Inter-Device Delay (ms)	0

**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 master radio. All channels using the same master 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.

Property Groups	<input type="checkbox"/> <b>Channel-Level Settings</b>	
General	Virtual Network	None
Serial Communications	Transactions per Cycle	1
<b>Communication Serialization</b>	<input type="checkbox"/> <b>Global Settings</b>	
	Network Mode	Load Balanced

### Channel-Level Settings

**Virtual Network** This property specifies 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** This property specifies 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):

- Channels with pending writes have the highest priority.
- Channels with pending explicit reads (through internal plug-ins or external client interfaces) are prioritized based on the read's priority.
- 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.

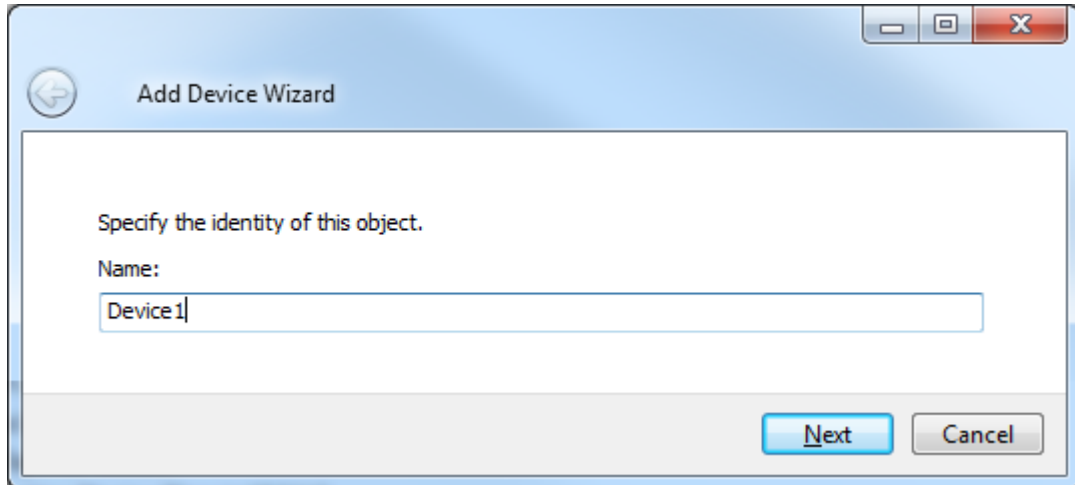
### Adding a Device

To add a new device to a channel:

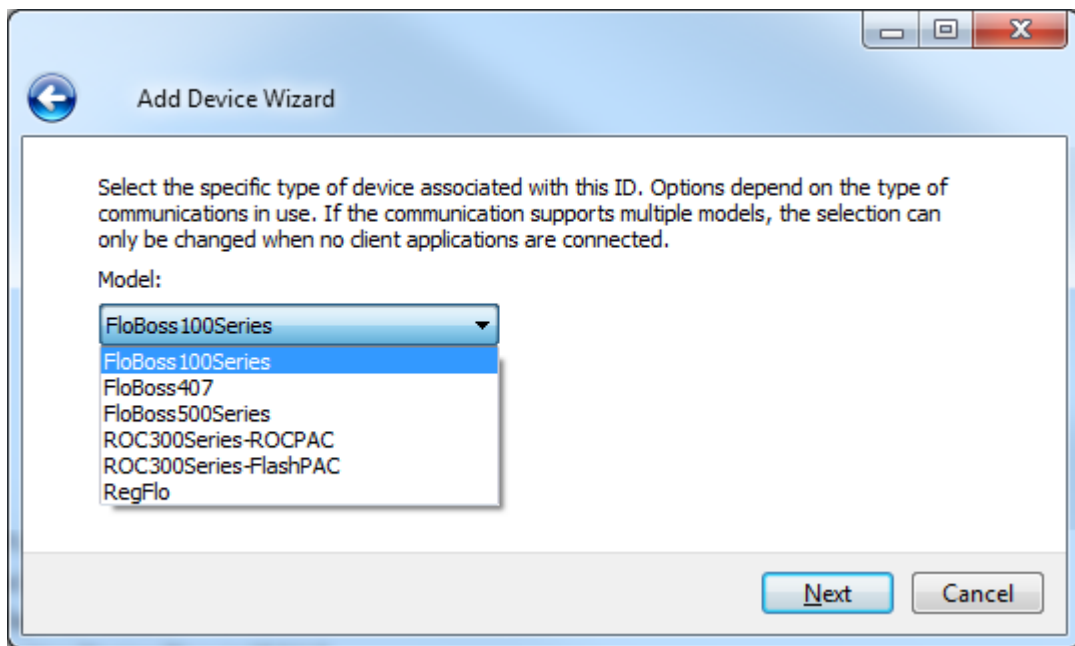
1. In the Project View, select the channel to contain the new device.
2. Select **Click to add a device** or right-click and choose **New Device**.



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.

Property Groups	Identification	
General	Name	
Scan Mode	Description	
	Channel Assignment	
	Driver	
	Model	
	ID Format	Decimal
	ID	2

### Identification

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

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

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

**Description:** User-defined information about this device.

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

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

**Driver:** Selected protocol driver for this device.

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

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

**ID:** This property specifies 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

Property Groups	+ Identification	
General	- Operating Mode	
Scan Mode	Data Collection	Enable
	Simulated	No

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

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

● **Notes:**

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



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

## Device Properties — Scan Mode

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

Property Groups	<input type="checkbox"/> <b>Scan Mode</b>	
General	Scan Mode	Respect Client-Specified Scan Rate ▼
Scan Mode	Initial Updates from Cache	Disable

**Scan Mode:** Specifies how tags in the device are scanned for updates sent to 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 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.

Property Groups	[-] <b>Communication Timeouts</b>	
General	Connect Timeout (s)	3
Scan Mode	Request Timeout (ms)	1000
<b>Timing</b>	Attempts Before Timeout	3
Redundancy	[-] <b>Timing</b>	
	Inter-Request Delay (ms)	0

## 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:** This property specifies 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:** This property specifies 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:** This property specifies 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.

Property Groups	[-] <b>Auto-Demotion</b>	
General	Demote on Failure	Enable
Scan Mode	Timeouts to Demote	3
Timing	Demotion Period (ms)	10000
<b>Auto-Demotion</b>	Discard Requests when Demoted	Disable

**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.

**Note:** *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.*

Property Groups	<input type="checkbox"/> <b>Tag Generation</b>	
General	On Property Change	Yes
Scan Mode	On Device Startup	Do Not Generate on Startup
Timing	On Duplicate Tag	Delete on Create
Auto-Demotion	Parent Group	
<b>Tag Generation</b>	Allow Automatically Generated Subgroups	Enable
Redundancy	Create	Create tags

**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:** This property specifies 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.

Property Groups	Time Zone	
General	Time Zone	(UTC-05:00) Eastern Time (US & Canada)
Scan Mode	Respect Daylight Saving Time	Yes
Timing	Synchronization	
Auto-Demotion	Time Sync Method	<b>Absolute</b>
Tag Generation	Time Sync Threshold (sec)	0
<b>Time Synchronization</b>	Sync Absolute	12:00:00 AM
Redundancy		

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

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

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

**Respect Daylight Saving Time:** Select Yes to follow Daylight Saving Time offset when synching the device time. Select No to ignore Daylight Saving Time. 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:** This property specifies 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:** This property specifies 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

Users can create a tag database based on either the device's configuration file or a ROCLINK 800 project file. To view or change the Tag Import settings after the device has been added, right-click on the device and select **Properties | Tag Import Settings**.

Property Groups General Scan Mode <b>Tag Import Settings</b> Communication Specification Operator Identification EFM Meters Redundancy	<table border="1"> <tr> <td colspan="2">[-] <b>Tag Import Settings</b></td> </tr> <tr> <td>Import Method</td> <td>Offline - from Import File</td> </tr> <tr> <td>Use Legacy Tag Names</td> <td>Yes</td> </tr> <tr> <td colspan="2">[-] <b>Offline Settings</b></td> </tr> <tr> <td>Tag Import File</td> <td></td> </tr> <tr> <td>ROC System File</td> <td></td> </tr> <tr> <td>System DB File</td> <td></td> </tr> <tr> <td>Display Descriptions</td> <td>No</td> </tr> <tr> <td></td> <td></td> </tr> </table>	[-] <b>Tag Import Settings</b>		Import Method	Offline - from Import File	Use Legacy Tag Names	Yes	[-] <b>Offline Settings</b>		Tag Import File		ROC System File		System DB File		Display Descriptions	No		
[-] <b>Tag Import Settings</b>																			
Import Method	Offline - from Import File																		
Use Legacy Tag Names	Yes																		
[-] <b>Offline Settings</b>																			
Tag Import File																			
ROC System File																			
System DB File																			
Display Descriptions	No																		

Descriptions of the parameters are as follows:

- **Import Method:** This parameter 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 Yes.
  - *For more information, refer to "Legacy vs. Non-Legacy Tag Names" below.*
- **Tag Import File:** When pressed, this button invokes a dialog for locating the \*.800 file that was created using the ROCLINK800 software.
- **ROC System File:** When pressed, this button invokes a dialog for locating the \*.mdb file. This file is usually named "ROC.mdb," and resides in the same folder where the ROCLINK 800 software is installed.
- **System DB File:** When pressed, this button invokes a dialog for locating the \*.mdw file. 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 master 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	IPAddress_12_0	12-0.1
	Non-Legacy	IP Address-12 (T12,L0,P1)	12-0.1
Boolean (.Bit)	Legacy	HighAlarm_3_0	3-0.16:2
	Non-Legacy	High Alarm-3 (T3,L0,P16) Bit 2	3-0.16:2

• 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 [Adding a Device](#). To modify settings on a defined device; select the device, right-click, select **Properties**, and select the **Communication Specification** property group.

Property Groups	<input type="checkbox"/> <b>Destination</b> Device Address      240 Device Group        240	
General	<input type="checkbox"/> <b>Source</b> Host Address         1 Host Group            1	
Scan Mode	<input type="checkbox"/> <b>Read Optimization</b> Use OpCode 180 For Read Requests      No	
Timing		
Auto-Demotion		
Tag Generation		
Time Synchronization		
Communication Specificat...		

Descriptions of the parameters are as follows:

- **Device address:** This parameter specifies the device number of the remote ROC device. The valid range is 1 to 255. The default setting is 240.
- **Device group:** This parameter specifies the group number of the remote ROC device. The valid range is 1 to 255. The default setting is 240.
- **Host address:** This parameter specifies the ROC unit number of the server. The valid range is 1 to 255. The default setting is 1.
- **Host group:** This parameter 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

This dialog is used to specify the operator identification values used when logging into the ROC device during initialization. To view or change these settings after the device has been added, right-click on the device and select **Properties** | **Operator Identification** property group.

Property Groups	[-] <b>Operator Identification</b>	
Time Synchronization	Username	LOI
Tag Import Settings	Enable Password	Yes
Communication Specification	Password	*****
<b>Operator Identification</b>	Enable Access Level	No

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

**Enable Password:** When set to Yes, this specifies that the ROC device has a password defined for the operator ID. The default 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 ROC device has defined access levels. The default is No.

**Access Level:** 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 Meter settings can be added to a device. The Fisher ROC Serial driver supports Gas Meters. A meter group for this type of meter is automatically created with the 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. A maximum of six meters can be added. To edit the meter; right-click and select **Properties**.

Property Groups	[-] <b>EFM Meters</b>	
General	Clear EFM Cache on Next Upload	Disable
Scan Mode	Meter Count	0
Timing		
Auto-Demotion		
Tag Generation		
Time Synchronization		
Settings		
<b>EFM Meters</b>		
Trend		
Redundancy		



Property Groups <b>General</b>	[-] <b>Identification</b>	
	Name	Meter_1
	Description	
	Driver	Fisher ROC Serial
	Non-Meter Events	Yes

Descriptions of the parameters are as follows:

- **Name:** This parameter specifies the meter name. Each meter must be assigned a unique name. The default setting is Meter\_1.
- **Description:** This parameter can be used to provide a short description of the meter.
- **Driver:** This parameter indicates the current driver (Fisher ROC Serial). It cannot be modified.
- **Non-Meter Events:** This parameter specifies whether non-meter EFM events are provided to EFM Exporters. The default setting is Yes.

To modify the meter order, select a meter from the project view and use the arrow icons in the toolbar accordingly.

Property Groups <b>General</b>	[-] <b>Identification</b>	
	Name	EFM(Gas)
	Description	
	[-] <b>Meter Order</b>	
	Meter Slot 1	Meter_1
	Meter Slot 2	Meter_2

**Important:**

- The meter order in the EFM Meter List should match the order of the meters in ROCLINK 800.
- Only Hourly and Daily History data may be uploaded from the FloBoss 100 Series and RegFlo devices.
- Extended History data is not supported for those devices.

**Note:** The Identification parameters of the meter group 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 click **Clear cache on next upload**. The default setting is No.

Property Groups Communications Parameters Tag Import Settings <b>EFM Meters</b>	[-] <b>EFM Meters</b>	
	Clear Cache On Next Upload	No

[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

<b>EFM Config Attribute</b>	<b>ROC TLP</b>
meter_id	7-n.0, 46-n.0
pressure_base	7-n.19, 46-n.13
temp_base	7-n.20, 46-n.14
calculation_method	7-n.3:7, 46-n.2
calculation_method_2	41-n.2, 46-n.3
pipe_diameter	7-n.9, 46-n.7
pipe_ref_temp	41-n.4, 46-n.8
meter_tap	7-n.4:0, 46-n.4:0
static_pressure_tap	7-n.4:2, 46-n.4:2
unit	7-n.3:3, 46-n.2:3
orifice_plate_size	7-n.10, 46-n.10
orifice_ref_temp	7-n.11, 46-n.11
dp_low_flow_cutoff	7-n.21, 46-n.43
atmospheric_pressure	41-n.1, 46-n.15
btu	7-n.6, 46-n.17
specific_gravity	7-n.5, 46-n.16
viscosity	7-n.17, 46-n.18
specific_heats	7-n.18, 46-n.19
aga_config	7-n.4, 46-n.4
pipe_material	41-n.5, 46-n.12
orifice_material	7-n.12, 46-n.12
co2	7-n.24, 46-n.24
n2	7-n.23, 46-n.23
c1	7-n.28, 46-n.28
c2	7-n.29, 46-n.29
c3	7-n.30, 46-n.30
isoc4	7-n.32, 46-n.32
nc4	7-n.31, 46-n.31
isoc5	7-n.34, 46-n.34
c5	7-n.33, 46-n.33
c6	7-n.35, 46-n.35
c7	7-n.36, 46-n.36

<b>EFM Config Attribute</b>	<b>ROC TLP</b>
c8	7-n.37, 46-n.37
c9	7-n.38, 46-n.38
c10	7-n.39, 46-n.39
o2	7-n.40, 46-n.40
h2o	7-n.26, 46-n.26
h2s	7-n.25, 46-n.25
he	7-n.27, 46-n.27
h2	7-n.42, 46-n.42
co	7-n.41, 46-n.41
contract_hour	15-0.10, 46-n.5

## History Mapping

n = meter number

l = pulse input number

<b>EFM Attribute</b>	<b>FloBoss 103, 104, 300, 500 Orifice TLP</b>	<b>FloBoss 107 Orifice TLP</b>	<b>FloBoss 103, 104, 300, 500 Turbine TLP</b>	<b>FloBoss 107 Turbine TLP</b>
flow_time	42-n.6	47-n.41	42-n.6	47-n.41
pulses			5-l.16	47-n.59
avg_diff_pressure	10-n.0	46-n.51		
avg_pressure	10-n.1	46-n.52	10-n.1	46-n.52
avg_temp	10-n.2	46-n.53	10-n.2	46-n.53
avg_extension	10-n.9	47-n.4		
c_prime	10-n.10	47-n.16		
raw_volume			42-n.16 (104, 500 series) 5-l.18 (300 series)	47-n.51
flowing_condition_factor			10-n.10	47-n.16
total_volume	10-n.3	47-n.0	10-n.3	47-n.0
total_energy	10-n.4	47-n.1	10-n.4	47-n.1
avg_btu	7-n.6	46-n.17	7-n.6	46-n.17
avg_specific_gravity	7-n.5	46-n.6	7-n.5	46-n.6
avg_co2	7-n.24	46-n.24	7-n.24	46-n.24
avg_n2	7-n.23	46-n.23	7-n.23	46-n.23
avg_c1	7-n.28	46-n.28	7-n.28	46-n.28
avg_c2	7-n.29	46-n.29	7-n.29	46-n.29
avg_c3	7-n.30	46-n.30	7-n.30	46-n.30

EFM Attribute	FloBoss 103, 104, 300, 500 Orifice TLP	FloBoss 107 Orifice TLP	FloBoss 103, 104, 300, 500 Turbine TLP	FloBoss 107 Turbine TLP
avg_isoc4	7-n.32	46-n.32	7-n.32	46-n.32
avg_nc4	7-n.31	46-n.31	7-n.31	46-n.31
avg_isoc5	7-n.34	46-n.34	7-n.34	46-n.34
avg_neoc5	7-n.33	46-n.33	7-n.33	46-n.33
avg_c6	7-n.35	46-n.35	7-n.35	46-n.35
avg_c7	7-n.36	46-n.36	7-n.36	46-n.36
avg_o2	7-n.40	46-n.40	7-n.40	46-n.40
avg_c8	7-n.37	46-n.37	7-n.37	46-n.37
avg_c9	7-n.38	46-n.38	7-n.38	46-n.38
avg_c10	7-n.39	46-n.39	7-n.39	46-n.39
avg_h2o	7-n.26	46-n.26	7-n.26	46-n.26
avg_h2s	7-n.25	46-n.25	7-n.25	46-n.25
avg_h2	7-n.42	46-n.42	7-n.42	46-n.42
avg_he	7-n.27	46-n.27	7-n.27	46-n.27
avg_co	7-n.41	46-n.41	7-n.41	46-n.41
specific_heat_ratio	46-n.19	46-n.19	46-n.19	46-n.19
viscosity	46-n.18	46-n.18	46-n.18	46-n.18

## Device Properties — Redundancy

Property Groups	<input checked="" type="checkbox"/> <b>Redundancy</b>	
General	Secondary Path	...
Scan Mode	Operating Mode	Switch On Failure
Timing	Monitor Item	
<b>Redundancy</b>	Monitor Interval (s)	300
	Return to Primary ASAP	Yes

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

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

## Data Types Description

Data Type	Description
Boolean	Single bit
Char	Signed 8-bit value bit 0 is the low bit bit 6 is the high bit bit 7 is the sign bit

Data Type	Description
Byte	Unsigned 8-bit value bit 0 is the low bit bit 7 is the high bit
Short	Signed 16-bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
Word	Unsigned 16-bit value bit 0 is the low bit bit 15 is the high bit
DWord	Unsigned 32-bit value bit 0 is the low bit bit 31 is the high bit
Float	32-bit floating point value bit 0 is the low bit bit 31 is the high bit
TLP	32-bit value Point type: Logical or point number and parameter number use three bytes. The top byte is not used.
String	A linear group of ASCII characters with preserved whitespace (1 byte per character).
Date	64-bit floating point value
Date Example	Date format:YYYY-MM-DDTHH:MM:SS.000 2000-01-01T12:30:45.000

**See Also:**[Logical Point Number Details](#)[ROC 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 0. 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

## Data Type Descriptions

### ROC 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 236 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.

[Logical / Point Number Details](#)

[ROC Point Types](#)

[User-Defined Point Types](#)

[Bit Assignments](#)

[TLP Data Type Conversion Process and Examples](#)

## ROC Point Types

The availability of point types varies by model and by how the particular unit is equipped. Points may be physical I/O or internal I/O.

### Physical I/O Points

Physical I/O points are one of Discrete Inputs (type 1), Discrete Outputs (type 2), Analog Inputs (type 3), Analog Outputs (type 4), and Pulse Inputs (type 5). The logical address for physical I/O points is based on their position in the unit, known as point number. There are four slots for each rack. The I/O field is not typically fully populated.

**Note:** Rack and slot are ROC nomenclature; however, many ROC devices have no physical racks or slots.

Logical addresses are calculated from the Point Number reference. For information on reconciling the Point Number to the Logical Address, refer to the table below.

ROCLINK Point Number	Server Logical Address
A1-A16	0-15
B1-B16	16-31
C1-C16	32-47
D1-D16	48-63
E1-E16	64-79
...	...

For example, an I/O point with ROCLINK 800 point number "A5" would be logical address "4". An I/O point with ROCLINK 800 point number "C5" would be logical address "36" by the function  $A + B + C5$  (or  $15+16+5-5=36$ ).

### Internal I/O Points

Internal I/O points consist of PID settings, system flags, communications parameters and other internal information. The internal I/O logical address begins at 0 for each point type. A ROC unit may have one or several of a given internal point type. Each point type has a specific set of parameters. ROC drivers have no array types.

## Point Type Tables

### [ROC Point Types for ROC300-Series](#)

### [ROC Point Types for FloBoss 100-Series, FloBoss 407, and FloBoss 500-Series](#)

### [ROC Point Types for RegFlo](#)

## ROC Point Types for ROC300-Series

For a detailed listing of all point type parameters' access, data type, length and description, refer to the device's ROC Protocol User Manual.

Point Types	Description	ROC300-Series FlashPAC	ROC300-Series ROCPAC
0	Configurable Opcode	Yes	Yes
1	Discrete Inputs (DI)	Yes	Yes
2	Discrete Outputs (DO)	Yes	Yes
3	Analog Inputs (AI)	Yes	Yes
4	Analog Outputs (AO)	Yes	Yes
5	Pulse Inputs (PI)	Yes	Yes
6	Proportional, Integral and Derivative (PID) Control	Yes	Yes
7	American Gas Association (AGA) Flow Parameters	Yes	Yes
9	Local Display Panel	Yes	Yes
10	AGA Flow Values	Yes	Yes
11	Tank Parameters	No	Yes
12	ROC Clock	Yes	Yes
13	System Flags	Yes	Yes
14	Communication Ports	Yes	Yes
15	System Variables (ROC Information)	Yes	Yes
16	Function Sequence Table (FST) Parameters	Yes	Yes
17	Soft Points	Yes	Yes
18	AI Calibration	No	Yes
19	Database Setup	Yes	Yes
20	ROC Tasks	Yes	Yes
21	Information for User Defined Points	Yes	Yes
22-31	User Defined Points	No	No
32	User Defined Typically Modem Config for COM1	Yes	Yes
33	User Defined Typically Modem Config for LOI and COM2	Yes	Yes
34	User Defined Typically Modbus Config for COM1	Yes	Yes



Point Types	Description	ROC300-Series FlashPAC	ROC300-Series ROCPAC
35	User Defined Typically Function Config for COM1	Yes	Yes
36	User Defined Typically Host Config for COM1	Yes	Yes
37	User Defined Typically Modbus Config for LOI and COM2	Yes	Yes
38	User Defined Typically Function Config for LOI and COM2	Yes	Yes
39	User Defined Typically Host Config for COM1	Yes	Yes
40	Multi-Variable Sensor (MVS) Parameters	Yes*	No
41	AGA Run Parameters	Yes	No
42	Extra Run Parameters	Yes	No
44	Power Control	Yes	No
49	Upload to Disk	No	No
50	Download to ROC	No	No
56	AI Calibration	Yes	No
57	Keypad / Login Security Parameters	Yes	No
59	Program Flash Control Parameters	Yes	No

\*Added via a user program.

### ROC Point Types for FloBoss 100-Series, FloBoss 407, and FloBoss 500-Series

Point Types	Description	FloBoss 100-Series	FloBoss 407-Series	FloBoss 500-Series
0	Configurable Opcode	Yes	Yes	Yes
1	Discrete Inputs	Yes	Yes	Yes
2	Discrete Outputs	Yes	Yes	Yes
3	Analog Inputs	Yes	Yes	Yes
4	Analog Outputs	Yes	Yes	Yes
5	Pulse Inputs	Yes	Yes	Yes
6	PID Control*	Yes	Yes	Yes
7	AGA Flow Parameters*	Yes	Yes	Yes
8	History Parameters	Yes	No	Yes
10	AGA Flow Values*	Yes	Yes	Yes
12	ROC Clock	Yes	Yes	Yes
13	System Flags	Yes	Yes	Yes
14	Communication Ports	Yes	Yes	Yes
15	System Variables (ROC Information)	Yes	Yes	Yes
16	FST Parameters	Yes	Yes	Yes
17	Soft Points	Yes	Yes	Yes

Point Types	Description	FloBoss 100-Series	FloBoss 407-Series	FloBoss 500-Series
19	Database Setup	Yes	Yes	Yes
20	ROC Tasks Diagnostics	No Yes	Yes No	No No
21	Information for User Defined Points	Yes	Yes	No
22-23	User Defined Points	No	No	No
24	Reserved	N/A	N/A	N/A
25-31	User Defined Points	No	No	No
32	User Defined - Typically Modem Config for COM1	Yes	Yes	No
33	User Defined - Typically Modem Config for LOI and COM2	Yes	Yes	No
34	User Defined - Typically Modbus Config for COM1	Yes	Yes	No
35	User Defined - Typically Function Config for COM1	Yes	Yes	No
36	User Defined - Typically Host Config for COM1	Yes	Yes	No
37	User Defined - Typically Modbus Config for LOI and COM2	Yes	Yes	No
38	User Defined: Typically Function Config for LOI and COM2	Yes	Yes	No
39	User Defined - Typically Host Config for COM1	Yes	Yes	No
40	Multi-Variable Sensor (MVS) Parameters	Yes	Yes	No
41	AGA Run Parameters*	Yes	Yes	Yes
42	Extra Run Parameters*	Yes	Yes	Yes
43	User Lists	Yes	Yes	Yes
44	Power Control	Yes	Yes	Yes
45	Meter Calibration and Sampler	Yes	No	Yes
46	Meter Configuration Parameters	Yes	No	Yes
47	Meter Flow Values	Yes	No	Yes
48	PID Control Parameters	Yes	No	Yes
49	Upload to Disk	No	No	No
50	Download to ROC	No	No	No
52	Battery Parameters	No	No	Yes
53	Modbus Configuration Parameters	Yes	No	Yes
54	Modbus Function Tables	Yes	No	Yes
55	Modbus Special Function Table	Yes	No	Yes
56	AI Calibration	Yes	Yes	Yes
57	Keypad / Login Security Parameters	Yes	Yes	Yes

Point Types	Description	FloBoss 100-Series	FloBoss 407-Series	FloBoss 500-Series
58	Revision Information	Yes	No	Yes
59	Program Flash Control Parameters	Yes	Yes	Yes
60-77	SAM User Defined Parameters	No	No	No
80	Enhanced Communication (ECM) Parameters	Yes	No	No
85	HART Parameters	Yes	No	No
86	Extended History Parameters	Yes	No	No
88	BLM User Lists	Yes	No	No
89	Chart User List Parameters	Yes	No	No
93	License Key Information Parameters	Yes	No	No
94	User C Program Parameters	Yes	No	No
98	Extended Soft Point Parameters	Yes**	No	No
117	Modbus Configuration Parameters	Yes	No	No
118	Modbus Register Mapping Parameters	Yes	No	No
120	Modbus Master Modem Configuration	Yes	No	No
121	Modbus Master Polling Table Configuration Parameters	Yes	No	No
122	DS800 Configuration Parameters	Yes	No	No

\*FloBoss 100-Series and FloBoss 500-Series Backward Compatibility.

\*\*This point type is only supported by FloBoss 107.

### ROC Point Types for RegFlo

Point Types	Description	Point Types	Description
0	Configurable Opcode	17	Soft Points
1	Discrete Inputs	19	Database Setup
2	Discrete Outputs	56	AI Calibration
3	Analog Inputs	57	Keypad / Login Parameters
4	Analog Outputs	80	Regulator Parameters
8	History Parameters	81	Logic Alarm Parameters
13	System Flags	84	User Discrete Values
14	Communication Ports	86	Extended History Parameters
15	System Variables	N/A	N/A

### Logical / Point Number Details

Within each point type, individual points are referenced by a logical number or a point number. The point numbers used by the ROC protocol for point types 1 to 5 are based on a physical input or output (I/O) with a rack and module location. All other point types use a logical number and are numbered in sequence.

### Physical Point Numbers 0 to 69

For point types 1 through 5, there are point numbers for the field I/O and for the diagnostic inputs. They are as follows:

- Point numbers 0 to 63 are assigned to field I/O (built-in or modular, 64 maximum). For example, if there were ten I/O modules in a ROC364, they would be points 0 through 9. The ROC I/O point database would reference these points by rack and module location, such as A1 through A10.
- Point numbers 64 to 69 are assigned to the diagnostic (system) I/O. For example, the five diagnostic points in a ROC364 would be 64 through 68. The ROC I/O point database would reference these points by rack and module (namely, E1 to E5).

### Logical Point Numbers 0 to 127

For all other point types (0 and 6–59), the point 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 four MVS points in a FloBoss 407 would be logical numbers 0 through 3.

**Note:** All parameters are 0-based for each point type.

### 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 FloBoss 100 Series devices.

### Supported User-Defined Point Range

22 to 23

25 to 39

178 to 189

### 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.

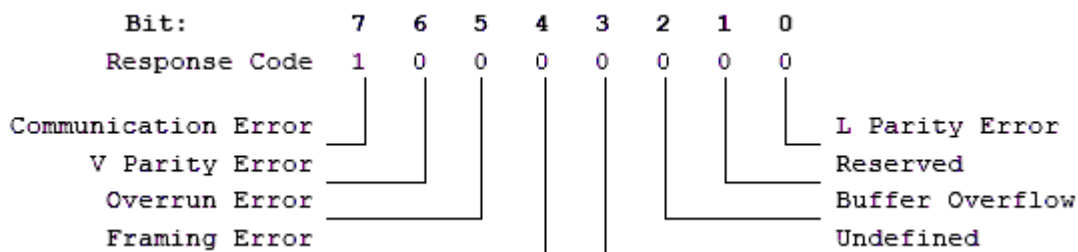
**Important:** Users must increment the version number of the user table when making changes to the table configuration. Failure to do so when making changes to the table while the server is actively reading user table tags results in bad quality tags or erroneous data.

### Supported User Table Point Range

user\_table-0.0 to user\_table-15.43

### Bit Assignments

The graphic below shows a sample bit assignment. The bits in each byte are numbered 0 to 7, right to left, with Bit 7 shown the furthest to the left. A 1 in any bit indicates that it is active or enabled.



### 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 driver tag's data update occurred later than expected (based on the specified scan

Item	Data Type	Access	Description
			rate).
_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 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 driver tag's data update occurred later than expected (based on the specified scan rate).
_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 (_TotalRe-

Item	Data Type	Access	Description
			sponses) 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*).

## Error Descriptions

---

The following categories of messages may be generated. Click on a link to see a list of related messages.

[Address Validation](#)

[Automatic Tag Database Generation](#)

[Device-Specific Messages](#)

[Device Status Messages](#)

[Serial Communications](#)

[User Configurable Table Messages](#)

[User-Defined Point Error Messages](#)

**See Also:**

[ROC Error Codes](#)

[Error Reasons](#)

## Address Validation

---

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

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

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

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

[Device address <address> is not supported by model <model name>.](#)

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

[Missing address.](#)

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

---

#### Error Type:

Warning

#### Possible Cause:

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

#### Solution:

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

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

---

#### Error Type:

Warning

#### Possible Cause:

A tag address that has been specified statically has been assigned an invalid data type.

#### Solution:

Modify the requested data type in the client application.



---

**Device address <address> contains a syntax error.**

---

**Error Type:**

Warning

**Possible Cause:**

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

**Solution:**

Re-enter the address in the client application.

---

**Device address <address> is not supported by model <model name>.**

---

**Error Type:**

Warning

**Possible Cause:**

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

**Solution**

1. Verify that the address is correct; if it is not, re-enter it in the client application.
2. Verify that the selected model name for the device is correct.

---

**Device address <address> is read only.**

---

**Error Type:**

Warning

**Possible Cause:**

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

**Solution:**

Change the access mode in the 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.

## Automatic Tag Database Generation Messages

---

The following messages may be generated. Click on a link for a description of that message.

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

[Unable to generate a tag database for device <device>. Reason: <Error reason>.](#)

[Unable to generate a tag database for device <device>. Reason: Auto tag generation failed: device is not responding.](#)

[Unable to generate a tag database for device <device>. Reason: Error while reading from import file.](#)

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

[Unable to generate a tag database for device <device>. Reason: Failed to open record set.](#)

[Unable to generate a tag database for device <device>. Reason: Import file <file name> not found.](#)

[Unable to generate a tag database for device <device>. Reason: Input file is corrupt.](#)

[Unable to generate a tag database for device <device>. Reason: Input file not found.](#)

[Unable to generate a tag database for device <device>. Reason: Low memory resources.](#)

[Unable to generate a tag database for device <device>. Reason: ROC system file <file name> not found.](#)

[Unable to generate a tag database for device <device>. Reason: System DB file <file name> not found.](#)

### **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 failed. Device 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 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.

**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 Import Settings](#)

[Tag Generation](#)

---

**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 Import Settings](#)[Tag Generation](#)

---

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

---

**Error Type:**

Warning

**Possible Cause:**

The import file cannot be found.

**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. This file must be accessible to the server runtime.

**See Also:**[Tag Import Settings](#)[Tag Generation](#)

---

**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 aborted.

**Solution:**

Close unused applications and/or increase the amount of virtual memory and 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 DB file cannot be found.

### Solution:

Ensure that the System 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-Specific Messages

---

The following error / warning messages may be generated. The messages are listed here in alphabetical order.

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

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

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

EFM hourly history upload failed for device <device>. Device responded with error code 14.

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>.

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

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

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

ROC initialization error: Unable to read general configuration.

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

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

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

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

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

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

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

<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.

 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](#)

---

**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 serial connection are incorrect.
3. Value written is out of range or 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:**

[Device Setup](#)

---

**EFM hourly history upload failed for device <device>. Device responded with error code 14.**

---

**Error Type:**

Error

**Possible Cause:**

The daily history size is set incorrectly causing error code 14 to be returned from the device.

**Solution:**

Set the daily history size to 35 instead of 60.

**See Also:**

[ROC Error Codes](#)

---

**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. Invalid tag address for point in block.
2. Device not responding.

**Solution:**



1. Verify the cabling between the PC and the device.
2. Confirm that all tags within this block exist on the device.

**See Also:**

[ROC 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 Error Codes](#)

**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:**

Fatal

**Possible Cause:**

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

**Solution:**

1. Make sure 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 Communication 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:**

Fatal

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

---

**Error Type:**

Fatal

**Possible Cause:**

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

**Solution:**

Check the Operator Identification settings and make sure the Enable Access Level checkbox is checked.

**See Also:**

[Operator Identification](#)

---

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

---

**Error Type:**

Fatal

**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 make sure 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 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 was attempting 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 a point is writable.

 For more information, refer to the controller's documentation.

**Device Status Messages**

The following error/warning messages may be generated. The messages are listed here in alphabetical order.

Device <device name> is not responding.  
EFM <type> upload failed for device <device>. <Reason>.  
Resetting the EFM cache for device <device>.

---

### 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 serial 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 failed for device <device>. <Reason>.

**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.

---

### 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

---

## Serial Communications

The following error/warning messages may be generated. The messages are listed here in alphabetical order.

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

COMn does not exist.

COMn is in use by another application.

Error opening COMn.

Unable to set comm parameters on COMn.

---

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

**Error Type:**

Serious

**Error Mask Definitions:**

**B** = Hardware break detected.

**F** = Framing error.

**E** = I/O error.

**O** = Character buffer overrun.

**R** = RX buffer overrun.

**P** = Received byte parity error.

**T** = TX buffer full.

**Possible Cause:**

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

---

### **User Configurable Table Messages**

---

The following messages may be generated. The messages are listed here in alphabetical order.

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

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.

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 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 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>.

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

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

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

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

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

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

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

**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 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 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.

---

**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.

---

**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 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 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.

**User-Defined Point Messages**

---

The following messages may be generated. Click on a link for a description of that message.

[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.](#)

[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.](#)

[Unable to load user-defined point configuration for point type <point type>, logical address <logical address>, parameter number <parameter> of device <device name>. The number of parameters for this point is 0.](#)

[Unable to parse the user-defined point configuration information for point type <point type> on device <device name>.](#)

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

**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 load user-defined point configuration for point type <point type>, logical address <logical address>, parameter number <parameter> of device <device name>. The number of parameters for this point is 0.**

---

**Error Type:**

Serious

**Possible Cause:**

The user program that supports the point type is not installed on the device.

**Solution:**

Verify that the user program that supports the point type is installed on the device.

**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 Error Codes

### Error Codes Returned by Opcode 255

Opcode 255 is an error message indicator that returns an error code.

Always consult with the [manufacturer documentation](#) for the official and latest information on protocol standards and codes.

### FlashPACs, FloBoss 500-Series, FloBoss 100-Series, and RegFlo Models

Error Code	Description
1	Invalid Opcode request
2	Invalid parameter number
3	Invalid logical number / point number
4	Invalid point type
5	Received too many data bytes
6	Received too few data bytes
7	Did not receive 1 data byte
8	Did not receive 2 data byte
9	Did not receive 3 data byte
10	Did not receive 4 data byte
11	Did not receive 5 data byte
12	Did not receive 16 data byte
13	Outside valid address range
14	Invalid history request
15	Invalid FST 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
23	Flash programming error
24	History configuration in progress
63	Requested security level too high

### ROCPACs and FloBoss 407 Models Only

Opcode Request	Error Returned	Description
0	2	Received too many data bytes.
2	3	Number of data bytes > 0.
3	4	Number of data bytes > 0.

Opcode Request	Error Returned	Description
6	5	Number of data bytes > 0.
7	6	Number of data bytes > 0.
8	7	Less than 6, or more than 8 data bytes received.
8	251	Industry Canada audit log full.
10	7	One of the following: <ul style="list-style-type: none"> <li>1. Incorrect number of data bytes received.</li> <li>2. Not a valid configurable Opcode point.</li> <li>3. Starting table location greater than 43.</li> <li>4. Number of table locations greater than 44.</li> </ul>
10	8	One of the following: <ul style="list-style-type: none"> <li>1. Starting table location plus the number of table locations greater than 44.</li> <li>2. The length of data is greater than 234 bytes.</li> </ul>
10	9	Error in configuration. One of the parameter definitions is no longer valid.
11	7	One of the following: <ul style="list-style-type: none"> <li>1. Incorrect number of data bytes received.</li> <li>2. Not a valid configurable Opcode point.</li> <li>3. Starting table location greater than 43.</li> <li>4. Number of table locations greater than 44.</li> </ul>
11	8	One of the following: <ul style="list-style-type: none"> <li>1. Starting table location plus the number of table locations greater than 44.</li> <li>2. Expecting more data than was received.</li> </ul>
11	9	Error in configuration. One of the parameter definitions is no longer valid.
11	251	Industry Canada audit log full.
17	6	Too little data.
17	8	Too much data.
17	20	Security error.
17	21	Not in security user list.
17	63	Requested security level too high.
17	251	Industry Canada audit log full.
18	9	Number of data bytes $\neq$ 16.
24	50	Number of data bytes < 12.
102	50	Number of data bytes $\neq$ 30.

Opcode Request	Error Returned	Description
102	251	Industry Canada audit log full.
103	50	Number of data bytes > 0.
105	52	One of the following conditions: <ul style="list-style-type: none"> <li>1. Number of data bytes &gt; 2.</li> <li>2. Invalid Point Number for requested module.</li> <li>3. Invalid RAM area number.</li> </ul>
107	54	One of the following conditions: <ul style="list-style-type: none"> <li>1. Number of data bytes &gt; 22.</li> <li>2. Requested more than 20 tags.</li> <li>3. Invalid RAM area number.</li> </ul>
	55	Invalid Point Number for requested RAM area.
120	56	Number of data bytes > 0.
121	57	One of the following: <ul style="list-style-type: none"> <li>1. Number of data bytes ≠ 3.</li> <li>2. Starting alarm pointer &gt; 239.</li> </ul>
122	58	One of the following: <ul style="list-style-type: none"> <li>1. Number of data bytes ≠ 3.</li> <li>2. Starting event pointer &gt; 239.</li> </ul>
126	59	One of the following conditions: <ul style="list-style-type: none"> <li>1. Number of data bytes &gt; 2.</li> <li>2. Invalid Point Number for requested RAM area.</li> <li>3. Invalid RAM area number.</li> </ul>
128	60	One of the following conditions: <ul style="list-style-type: none"> <li>1. The history Point Number requested exceeds the number of points defined for that history RAM area.</li> <li>2. The data portion of the message received did not consist solely of 3 bytes.</li> <li>3. The history point is not defined for periodic, hourly archival.</li> </ul>
128	61	One of the following conditions: <ul style="list-style-type: none"> <li>1. Not a valid day specified; the day specified must be between 1 and 31, inclusive.</li> <li>2. Not a valid month specified; the month specified must be between 1 and 12, inclusive.</li> </ul>
128	62	Specified day and/or month requested does not match the day and/or month in the timestamp associated with the first history value for the day.



Opcode Request	Error Returned	Description
128	63	The Point Number exceeds the number of possible history points for that RAM area.
130	62	One of the following conditions: <ol style="list-style-type: none"> <li>1. The number of data values requested exceeds the number of data values defined for that history RAM area.</li> <li>2. The data portion of the message received did not consist solely of 5 bytes.</li> <li>3. The module number exceeds or is equal to the maximum number of modules supported by the ROC.</li> </ol>
130	63	One of the following conditions: <ol style="list-style-type: none"> <li>1. The Point Number exceeds the number in the requested module.</li> <li>2. The requested Point Number has an invalid archival type.</li> </ol>
131	103	Industry Canada audit log retrieval error.
132	104	Industry Canada clear audit log error.
133	105	Industry Canada audit log retrieval error.
150	73	One of the following conditions: <ol style="list-style-type: none"> <li>1. Number of data bytes <math>\neq</math> 3.</li> <li>2. Display number <math>&gt;</math> 1.</li> </ol>
151	74	One of the following conditions: <ol style="list-style-type: none"> <li>1. Display number <math>&gt;</math> 1.</li> <li>2. Number lines requested <math>&gt;</math> 3.</li> <li>3. Starting row <math>&gt;</math> 20.</li> <li>4. Starting row + Number of rows <math>&gt;</math> 21.</li> </ol>
151	75	Number of data bytes incorrect.
158	77	Number of data bytes $>$ 0.
162	251	Industry Canada audit log full.
165	85	One of the following conditions: <ol style="list-style-type: none"> <li>1. Number of data bytes <math>&lt;</math> 4.</li> <li>2. Set/Send byte <math>&lt;</math> 4.</li> <li>3. Invalid Point Number for RAM area.</li> </ol>
165	86	Number of data bytes $\neq$ 7 for "Set Operation."
165	87	Number of data bytes $\neq$ 4 for "Send Operation."
166	88	Received 4 or fewer data bytes, or invalid Point Type.
166	91	Point does not exist.
166	251	Industry Canada audit log full.

Opcode Request	Error Returned	Description
167	8	More than 250 data bytes in response.
167	9	Invalid parameter.
167	90	One of the following conditions: <ol style="list-style-type: none"> <li>1. Did not receive 4 data bytes.</li> <li>2. Invalid Point Type.</li> </ol>
167	91	Point does not exist.
167	92	Point does not exist.
167	93	Invalid range of parameters asked for.
167	94	Too many data bytes to send (more than 240).
170	93	One of the following conditions: <ol style="list-style-type: none"> <li>1. Number of data bytes <math>\neq</math> 2.</li> <li>2. Number requested <math>&gt;</math> 32.</li> <li>3. Invalid starting Point Number.</li> <li>4. Too many points requested for requested starting Point Number.</li> </ol>
171	94	One of the following conditions: <ol style="list-style-type: none"> <li>1. Number of data bytes <math>\neq</math> 4.</li> <li>2. Invalid Point Type.</li> </ol>
	95	Invalid point or type.
171	251	Industry Canada audit log full.
180	XXX	The parameter in which the error was detected.
181	103	Received less than 4 data bytes.
181	104	Point type out of range (1 - 24 are valid).
181	105	Point does not exist, or invalid parameter.
181	106	Not enough data bytes received.
181	251	Industry Canada audit log full.
255	102	Number of data bytes $\neq$ 2.
XXX	1	Invalid Opcode requested.

## Error Reasons

Error Reason	Possible Cause	Solution
Device not responding	<i>For more information, refer to <a href="#">Device &lt;device name&gt; is not responding.</a></i>	<i>For more information, refer to <a href="#">Device &lt;device name&gt; is not responding.</a></i>
Device responded	The ROC device responded with an error code.	<i>For more information, refer to <a href="#">ROC Error Codes.</a></i>

<b>Error Reason</b>	<b>Possible Cause</b>	<b>Solution</b>
with error code		
Framing error	The response packet from the ROC 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 ROC 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

- Absolute 30
- Address <address> is out of range for the specified device or register. 48
- Address Descriptions 38
- Address Validation 48
- Allow Sub Groups 29
- Attempts Before Timeout 26
- Auto-Demotion 26
- Automatic Tag Database Generation Error Messages 50

## B

- Bit Assignments 45
- Block read for point type <point type>, logical address <logical address>, parameter range <start parameter - end parameter> of device <device name> failed. <Error reason>. 55
- 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. 68
- Block read for user table <table number>, location range <start location> - <end location> of device <device name> failed. Device returned error code <error code>. 64
- Block read for user table <table number>, location range <start location> - <end location> of device <device name> failed. Framing error. 64
- 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. 64

## C

- Channel Assignment 23
- Channel Setup 7
- Communication Specification 31
- Communications error on <channel name> [<error mask>]. 62
- Communications Timeouts 25-26
- COMn does not exist. 62
- COMn is in use by another application. 62
- Connect Timeout 26
- Create 29

**D**

Data Collection 24

Data type <type> is not valid for device address <address>. 48

Data Types Description 36

Daylight Saving Time 29

Delete 28

Demote on Failure 27

Demotion Period 27

Device <device> responded with error. (Tag <tag address>) - Details: <error code>. 56

Device address <address> contains a syntax error. 49

Device address <address> is not supported by model <model name>. 49

Device address <address> is read only. 49

Device name - Failed to read EFM pointer file. <Extended error>. 54

Device name - Failed to write EFM pointer file. <Extended error>. 55

Device name <device name> is not responding. 61

Device Properties — EFM Meters 32

Device Properties — Tag Generation 27

Device Setup 21

Device Specific Messages 54

Device Status Messages 60

Discard Requests when Demoted 27

Do Not Scan, Demand Poll Only 25

Driver 24

**E**

EFM <type> upload failed for device <device>. <Reason>. 61

EFM Attribute Mapping 33

EFM hourly history upload failed for device <device>. Device responded with error code 14. 56

Error Descriptions 48

Error importing CSV tag record <record number>: Address <Address> is out of range for the specified device or register. 50

Error opening COMn. 63

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>. 65

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>. 65

Error Reasons 74

**F**

Failed to obtain data block for PointType = <point type>, Logical Address = <address>, Starting Parameter = <starting parameter>, Ending Parameter <ending parameter> for device <device>. Error = <ROC error code>. 56

Failed to write data for point type = <point type>, logical address = <address>, parameter = <parameter> for device <device>. Error = <error code>. 57

**G**

General 23

Generate 28

**H**

Help Contents 6

**I**

ID 24

Identification 23

Initial Updates from Cache 25

Inter-Request Delay 26

Interval 30

**M**

Method 29

Missing address. 49

Model 24

**N**

Name 23

**O**

On Device Startup 28

On Duplicate Tag 28  
On Property Change 28  
OnPoll 30  
Operating Mode 24  
Operator Identification 32  
Operator identification failed for device <device name>. <Error reason>. 57  
Overview 7  
Overwrite 28

## **P**

Parent Group 29  
Point Number Details 43

## **R**

Read for point type <point type>, logical address <logical address>, parameter number <parameter number> of device <device name> failed. <Error reason>. 58  
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. 68  
Read for user table <table number>, location <location address> of device <device name> failed. Device returned error code <error code>. 65  
Read for user table <table number>, location <location address> of device <device name> failed. Framing error. 65  
Read for user table <table number>, location <location address> of device <device name> failed. Location is not configured in the user table. 66  
Redundancy 36  
Request Timeout 26  
Resetting the EFM cache for device <device>. 61  
Respect Tag-Specified Scan Rate 25  
ROC Error Codes 70  
ROC initialization error: Unable to read general configuration. 58  
ROC initialization error: Unable to retrieve I/O map. 58  
ROC Point Types 39

## **S**

Scan Mode 25  
Serial Communications 61

Serialization of EFM data to temporary file <file name> failed. Reason: <file I/O error>. 59

Simulated 24

Statistics Items 45

## T

Tag Generation 27

Tag Import Settings 30

The username or password supplied was not accepted. Error = 6. 59

The username or password supplied was not accepted. Error = 63. 59

Time Sync Threshold 30

Time Synchronization 29

Time synchronization with device <device name> failed. <Error reason>. 59

Time Zone 29

Timeouts to Demote 27

TLP Data Type Conversion Process and Examples 37

## U

Unable to generate a tag database for device <device>. Reason: <error reason>. 50

Unable to generate a tag database for device <device>. Reason: Auto tag generation failed. Device is not responding. 51

Unable to generate a tag database for device <device>. Reason: Error while reading from import file. 51

Unable to generate a tag database for device <device>. Reason: Error while reading from ROC system file. 51

Unable to generate a tag database for device <device>. Reason: Failed to open recordset. 52

Unable to generate a tag database for device <device>. Reason: Import file <file name> not found. 52

Unable to generate a tag database for device <device>. Reason: Input file is corrupt. 52

Unable to generate a tag database for device <device>. Reason: Input file not found. 53

Unable to generate a tag database for device <device>. Reason: Low memory resources. 53

Unable to generate a tag database for device <device>. Reason: ROC system file <file name> not found. 53

Unable to generate a tag database for device <device>. Reason: System DB file <file name> not found. 54

Unable to load user-defined point configuration for point type <point type>, logical address <logical address>, parameter number <parameter> of device <device name>. The number of parameters for this point is 0. 68

Unable to parse the user-defined point configuration information for point type <point type> on device <device name>. 68

Unable to set comm parameters on COMn. 63



User-defined point configuration upload for point type <point type>, logical address <logical address>, parameter number <parameter> of device <device name> failed. <Error reason>. 69

User-Defined Point Error Messages 67

User-Defined Point Types 44

User Configurable Table Messages 63

User table configuration upload on device <device name> failed. Device not responding. 66

User table configuration upload on device <device name> failed. Device responded with error code <error code>. 66

User table configuration upload on device <device name> failed. Framing error. 67

User table configuration upload on device <device name> failed. Internal error. 67

User Table Points 44

## W

Write for the following tags of device <device name> failed: <tag list>. <Error reason>. 60

Write request rejected on read only item reference <channel name> <device name> <address>. 60