

# System Monitor Driver

© 2025 PTC Inc. All Rights Reserved.

# Table of Contents

<b>System Monitor Driver</b>	<b>1</b>
<b>Table of Contents</b>	<b>2</b>
Welcome to the System Monitor Driver Help Center	3
Overview	3
Setup	3
Channel Properties – General	4
Tag Counts	4
Channel Properties – Write Optimizations	4
Channel Properties – Advanced	5
Device Properties – General	7
Operating Mode	7
Tag Counts	8
Device Properties – Scan Mode	8
Device Properties – Import	9
Tag Descriptions	10
Device Properties – Settings	10
<b>Data Types Description</b>	<b>12</b>
<b>Address Descriptions</b>	<b>13</b>
<b>Error Descriptions</b>	<b>14</b>
Counter <counter> contains invalid data	14
Counter <counter> has a negative denominator	14
Counter <counter> has a negative time base	14
Counter <counter> has a negative value	14
Counter <counter> is not valid	15
Counter <counter> was not found	15
Devices of model type N/A cannot be multidropped	15
Failed to retrieve instances for object <object>	15
<b>Index</b>	<b>17</b>

## Welcome to the System Monitor Driver Help Center

---

This help center is the user documentation for Kepware System Monitor Driver. This help center is updated regularly to reflect the latest functionality and information.

### [Overview](#)

What is the System Monitor Driver?

### [Setup](#)

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

### [Data Types Description](#)

What data types does this driver support?

### [Address Descriptions](#)

How do I address a data location with the System Monitor Driver?

### [Importing Items](#)

How can I import items to the System Monitor Driver?

### [Error Descriptions](#)

What messages does the System Monitor Driver produce?

Version 1.037

© 2025 PTC Inc. All Rights Reserved.

## Overview

---

The System Monitor Driver uses the Microsoft Performance Data Helper application interface to provide access to performance information on Processes, Threads and other system components on the user's system. To begin monitoring the PC's performance from the OPC Client application (including HMI, SCADA, Historian, MES, ERP, and countless custom applications) install the OPC server and the System Monitor Driver on each System/PC.

■ **Note:** A redistributable version of the PDH library may be required. It is available from Microsoft®.

## Setup

---

Configuration of the environment includes specifying channel and device properties for communication with the System Monitor Driver.

### Channel and Device Limits

The maximum number of channels supported by this driver is 100. The maximum number of devices supported by this driver is 1 per channel.

## Channel Properties – General

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

Property Groups	<b>General</b>	
	Write Optimizations	
	Advanced	
	<b>Identification</b>	
	Name	
	Description	
	Driver	
	<b>Diagnostics</b>	
	Diagnostics Capture	Disable
	<b>Tag Counts</b>	
Static Tags	10	

### Identification

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

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

**Description:** Specify user-defined information about this channel.

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

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

**Note:** With the server's online full-time operation, these properties can be changed at any time. This includes changing the channel name to prevent clients from registering data with the server. If a client has already acquired an item from the server before the channel name is changed, the items are unaffected. If, after the channel name has been changed, the client application releases the item and attempts to re-acquire using the old channel name, the item is not accepted. Changes to the properties should not be made once a large client application has been developed. Utilize proper user role and privilege management to prevent operators from changing properties or accessing server features.

### Diagnostics

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

**Note:** This property is not available if the driver or operating system does not support diagnostics.

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

### Tag Counts

**Static Tags:** Provides the total number of defined static tags at this level (device or channel). This information can be helpful in troubleshooting and load balancing.

## Channel Properties – Write Optimizations

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

Property Groups	Write Optimizations	
General	Optimization Method	Write Only Latest Value for All Tags
Write Optimizations	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	Non-Normalized Float Handling	
General	Floating-Point Values	Replace with Zero
Write Optimizations	Inter-Device Delay	
Advanced	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 disabled if the driver does not support floating-point values or if it only supports the option that is displayed. According to the channel's float normalization setting, only real-time driver tags (such as values and arrays) are subject to float normalization. For example, EFM data is not affected by this setting.

● *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.

## 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	General	

Identification	
Name	
Description	
Channel Assignment	
Driver	
Model	
ID Format	Decimal
ID	2

### Identification

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

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

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

**Description:** Specify the user-defined information about this device.

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

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

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

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

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

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

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

### Operating Mode

Property Groups	General	

Identification	
Operating 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:** Place the device into or out of Simulation Mode. In this mode, the driver does not attempt to communicate with the physical device, but the server continues to return valid OPC data. Simulated stops physical communications with the device, but allows OPC data to be returned to the OPC client as valid data. While in Simulation Mode, the server treats all device data as reflective: whatever is written to the simulated device is read back and each OPC item is treated individually. The data is not saved if the server removes the item (such as when the server is reinitialized). The default is No.

● **Notes:**

1. Updates are not applied until clients disconnect and reconnect.
2. The 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.
3. 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.
4. When a device is simulated, updates may not appear faster than one (1) second in the client.

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

## Tag Counts

Property Groups	[-] Identification	
General	[-] Operating Mode	
	[-] Tag Counts	
	Static Tags	130

**Static Tags:** Provides the total number of defined static tags at this level (device or channel). This information can be helpful in troubleshooting and load balancing.

## Device Properties – Scan Mode

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

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

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

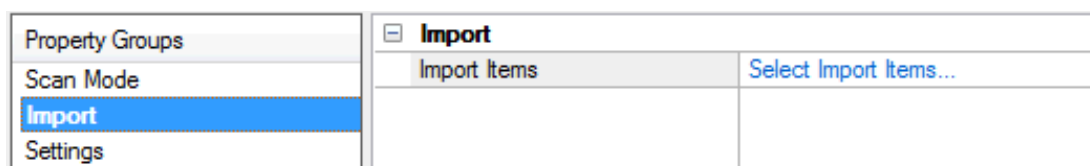
- **Respect Client-Specified Scan Rate:** This mode uses the scan rate requested by the client.
- **Request Data No Faster than Scan Rate:** This mode specifies the value set as the maximum scan rate. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
  - **Note:** When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



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

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

## Device Properties – Import

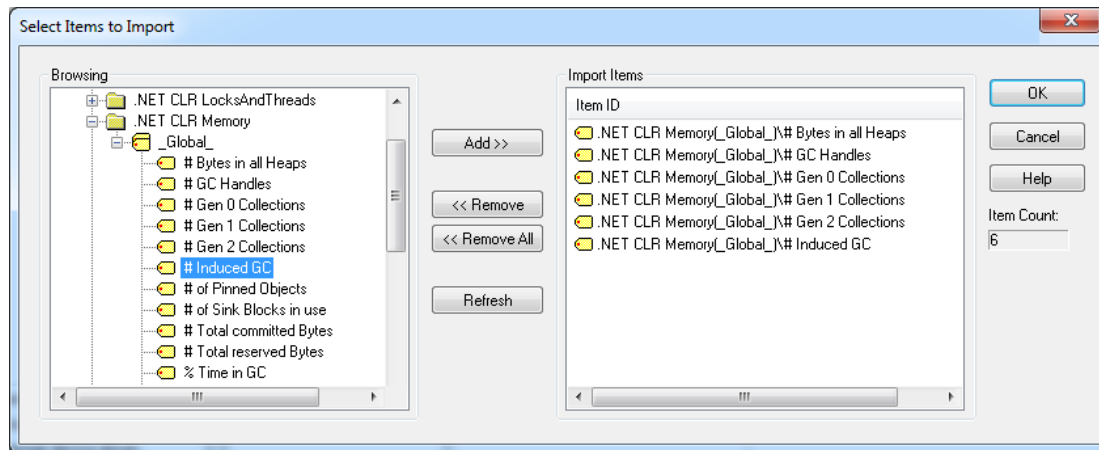


The import property group initiates the process of selecting and importing available Performance Objects, instances, and counters on the machine through the Import Items dialog. From this dialog, individual counters (or groups of counters) can be selected so that tags can be automatically generated. To access this dialog after device creation, click **Select Import Items**.

The Import Items dialog is separated into two sections: the **Item Import Tree** and the **Import Item List**. Items are selected from the Import Item Tree and then added to the Import Item List. The driver will then be able to create tags based on the contents of the Import Item List. The Import Item Tree, found on the left, is a hierarchical tree view that displays all of the computer's current high-level Performance Objects, instances, and counters. Each Performance Object can be expanded to show all its associated instances and/or counters, and each instance can be expanded to display all of its counters. Counters can be added by either double-clicking on the individual counter or by selecting the counter and then clicking **Add**. If an item in the tree is right-clicked, a context menu will appear that has shortcuts for adding counters. If an object or instance is right-clicked and **Add Items** is selected, all the items below that item will be added to the Import Item list. Multiple counters can be selected by left-clicking on the desired counters while using either the **Ctrl** or **Shift** key. The Import Item List can be found on the right side of the dialog.

Once a counter or group of counters has been added to the Import Item list, they can easily be removed. To remove a specific counter, select and then click **Remove**. Alternatively, right-click on a counter and then choose the corresponding menu item in the right-click menu. To remove all counters, click **Remove All**.

To refresh the Item Import Tree, click **Refresh**. Alternatively, right-click on the base item of the tree and then select **Refresh** from the pop-up menu. This will update the entire tree to include any changes that have taken place since the tree was last updated. This is particularly important for items in the Process and Thread groups (which can change often).



The System Monitor Driver creates a tag for each import item and then adds it to the device for which the import operation was performed. The driver also assigns proper tag names and organizes the tags into groups based on the complete item name. The counter name forms the tag name. The tag will be added to a group that derives its name from the object name. If the counter also belongs to an instance, then another group will be created underneath the object group. This group derives its name from the instance name. For example, the following will be created under the device for the counter *?# Induced GC?* shown in the image above.

- + **U\_Net CLR Memory:** This is the group name based on the object.
- + **U\_Global\_:** This is the group name based on the instance.
- + **# Induced GC:** This is the tag name based on the counter.

The tag address is the complete counter name: *.NET CLR Memory(\_Global\_)# Induced GC*

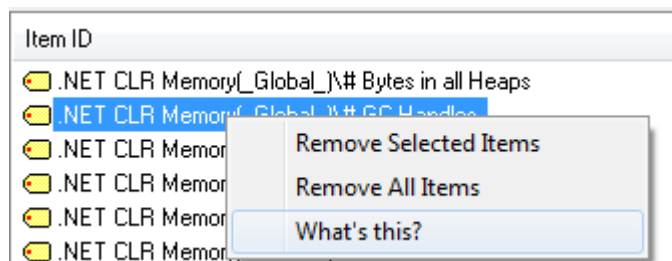
During the import, the driver applies the following changes when creating tag and group names.

1. Dots ('.'), double quotes (""), and underscores ('\_') are not allowed to be the first character in the tag name. They will be replaced with "U\_".
2. Dots and double quotes are not allowed anywhere in the tag name. They will be replaced with underscores.
3. Dots, double quotes, underscores and back slashes are not allowed to be the first character in the group name. They will be replaced with "U\_".
4. Dots, double quotes and back slashes are not allowed anywhere in the group name. They will be replaced with an underscore.

**Note:** The default data type is Double. If a Long data type is needed, change the data type from the GUI.

## Tag Descriptions

To view Tag Descriptions, right-click on the particular counter in the Item Import dialog and then select **"What's this?"** at the bottom of the menu. If more than one Counter is selected, the option will be disabled.



## Device Properties – Settings

Property Groups	[-] <b>Settings</b>	
Import	Query Size	50
<b>Settings</b>		

**Query Size:** Specify how many tags the System Monitor Driver updates at once. Although the Performance Data Helper functions can be CPU intensive for large projects, it can be offset by reducing (or in some cases, increasing) the Query Size. The valid range is between 1 and 100 inclusive.


● **Note:** This driver does not support multidropping devices. Any attempt to multidrop a device can result in the following error message: Devices of model type 'N/A' cannot be multidropped.

## Data Types Description

---

The default data type is Double.

Data Type	Description
Double	64-bit floating point value
Long	Signed 32-bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit

 **Note:** Longs must either be manually created by the user or by altering the data type of an automatically generated tag after it has been created.

## Address Descriptions

Because of the dynamic aspect of many of the counters used by the System Monitor Driver, it is recommended that tags be automatically generated from the Import Items dialog. This ensures that the Performance Objects and instances are available and being used. If the counter address of a specific counter is known, however, tags can be manually created.

The System Monitor Driver uses the same syntax used by the Performance Data Helper library.

*Object(Instance/ObjectInstance#InstanceIndex)\CounterName*

Descriptions are as follows:

- **Object:** The name of the Performance Object that contains the instance or counter that will be monitored (such as Process).
- **Instance (optional):** The name of the Performance Instance that will be monitored. Some objects do not have instances.
- **Object Instance (optional):** The name or number of the object (if any). If there can be multiple instances of a particular Performance Object, this will be the name of a particular instance of that object.
- **Instance Index (optional):** For Object Instances that can have multiple instances at the same time that have the same name, this will be a zero based index specifying a particular instance.
- **Counter Name:** The actual name of the counter. Counters are specific to certain Performance Objects; meaning, a counter that appears under the Thread Object may not appear under any other object.

The Instance, Object Instance and Instance Index are optional and are only used if the instance being monitored allows for multiple instances (and/or if those instances have multiple threads). Some Performance Objects have associated tags instead of instances. For example, the simplest valid counter is of the following form:

*IP\Datagrams Outbound Discarded*

The most complicated addresses would resemble as follows:

*Thread(svchost/7#2)\% Processor Time*

This will monitor the percentage of the processor time used by the third thread of the seventh instance of the svchost thread.

• For more information on the structure of Counter Paths, refer to the MSDN documentation [Specifying a Counter Path](#).

• **Note:** The Wildcard character '\*' is not supported for use in the System Monitor Driver.

## Error Descriptions

---

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

[Counter <counter> contains invalid data](#)

[Counter <counter> has a negative denominator](#)

[Counter <counter> has a negative time base](#)

[Counter <counter> has a negative value](#)

[Counter <counter> is not valid](#)

[Counter <counter> was not found](#)

[Devices of model type N/A cannot be multidropped](#)

[Failed to retrieve instances for object <object>](#)

[Instance <instance> was not found](#)

[Performance <object> was not found](#)

### Counter <counter> contains invalid data

---

#### Error Type:

Warning

#### Possible Cause:

This warning is most likely a problem associated with moving a project from one computer to another.

#### Solution:

Regenerate the tags from within the Import Items dialog.

### Counter <counter> has a negative denominator

---

#### Error Type:

Warning

#### Possible Cause:

This warning is most likely caused by timing errors in the Windows kernel and usually appears when monitoring low-level information. For example, counters within the Process group.

#### Solution:

This can happen intermittently. It can normally be fixed by waiting until the next update or performing a Read on the counter.

### Counter <counter> has a negative time base

---

#### Error Type:

Warning

#### Possible Cause:

This warning is most likely caused by timing errors in the Windows kernel and usually appears when monitoring low-level information. For example, counters within the Process group.

#### Solution:

Regenerate the tags from within the Import Items dialog.

### Counter <counter> has a negative value

---

#### Error Type:

Warning

#### Possible Cause:

This warning is most likely caused by timing errors in the Windows kernel and usually appears when monitoring low-level information. For example, counters within the Process group.

**Solution:**

This can happen intermittently. It can normally be fixed by waiting until the next update or performing a Read on the counter.

**Counter <counter> is not valid**

---

**Error Type:**

Warning

**Possible Cause:**

1. The counter name is spelled wrong.
2. The counter is not valid for the Performance Object being monitored.

**Solution:**

1. Check the spelling of the counter name.
2. Try to find the counter in the Import Items dialog under the Performance Object that is being monitored to make sure it is a valid counter.

**Counter <counter> was not found**

---

**Error Type:**

Warning

**Possible Cause:**

1. The counter name is spelled wrong.
2. The counter is not valid for the Performance Object being monitored.

**Solution:**

1. Ensure that the Counter is valid for the Performance Object.
2. If a project is being used from another computer, regenerate the tags from the Import Items dialog in Device Properties.

**Devices of model type N/A cannot be multidropped**

---

**Error Type:**

Warning

**Possible Cause:**

An attempt was made to multidrop a device.

**Solution:**

Configure a single device for this driver.

**Failed to retrieve instances for object <object>**

---

**Error Type:**

Warning

**Possible Cause:**

The application providing data for the Performance Object was installed or uninstalled incorrectly.

**Solution:**

If this Performance Object is particular to an application or service, try reinstalling that application or service.

**Note:**

This message is from the Import Items dialog.



# Index

## A

Address Descriptions 13

## C

Channel Assignment 7

Channel Properties – Advanced 5

Channel Properties – General 4

Channel Properties – Write Optimizations 5

Counter <counter> contains invalid data 14

Counter <counter> has a negative denominator 14

Counter <counter> has a negative time base 14

Counter <counter> has a negative value 14

Counter <counter> is not valid 15

Counter <counter> was not found 15

## D

Data Collection 8

Data Types Description 12

Device Properties – General 7

Devices of model type N/A cannot be multidropped 15

Diagnostics 4

Do Not Scan, Demand Poll Only 9

Driver 7

Duty Cycle 5

## E

Error Descriptions 14

## F

Failed to retrieve instances for object 15

## G

General 7

## I

ID 7

Identification 4, 7

Importing Items 9

Initial Updates from Cache 9

Instance <instance> was not found 16

Inter-Device Delay 6

## M

Model 7

## N

Name 7

Non-Normalized Float Handling 5

## O

Operating Mode 7

Optimization Method 5

Overview 3

## P

Performance <object> was not found 16

## R

Replace with Zero 5

Respect Tag-Specified Scan Rate 9

## S

Scan Mode 8

Settings 10

Setup 3

Simulated 8

**T**

Tag Counts 4, 8

Tag Descriptions 10

**U**

Unmodified 5

**W**

Write All Values for All Tags 5

Write Only Latest Value for All Tags 5

Write Only Latest Value for Non-Boolean Tags 5