SNMP Driver

Table of Contents

SNMP Driver 1
Table of Contents 2
SNMP Driver 6
Overview 6
Channel Setup 7
Channel Properties — General 7
Channel Properties — Ethernet Communications 8
Channel Properties — Write Optimizations 8
Channel Properties — Advanced 9
Channel Properties — Communication Serialization 10
Device Discovery Procedure 11
Device Setup 13
Device Properties — General 13
Device Properties — Scan Mode 15
Device Properties — Timing 15
Device Properties — Auto-Demotion 16
Device Properties — Tag Generation 17
Device Properties — Communications 19
Device Properties — v3 Security 21
Device Properties — MIB Import 22
Device Properties — Trap / Inform Notifications 22
Device Properties — Network Analyst 24
Device Properties — Redundancy 24
Data Types Description 25
Historical Data Attributes 27
Previous Value 27
Delta Time 27
Moving Average 27
Address Descriptions 29
About SNMP Addresses 29
About MIB Modules 31
About Network Analyst Tags 32
Trap Tags 32
Trap Events Queue 35
Auto-Created Trap Tags 36

Message Descriptions **37**
Address Validation 37
Address <address> is out of range for the specified device or register. 37
Data Type <type> is not valid for device address <address>. 37
Device address <address> contains a syntax error. 38
Device address <address> is read only. 38
The remote device reports that the requested name <OID> does not exist on <device name>. 38
Runtime Messages 38
<Channel name>.<device name>: unable to open a SNMP session to host <host> on port <port>, using protocol <protocol>. 39
<Channel name>.<device name>: Unable to establish a trap listener on port <port>, using protocol <protocol>. No trap events will be received. 40
Access to address <address> on <channel name>.<device name> is not permitted. 40
Address <address> on <channel name>.<device name> is not writable. 40
Address <address> on <channel name>.<device name> is unavailable. 41
Device <device name> does not support the necessary information required to perform network analysis. Network Analyst tags will be disabled for this device. 41
Device <device name> does not support the number of ports currently configured in this application. Network Analyst tags will be disabled for this device. 41
Device <device name> is not responding. 41
Device Discovery has exceeded <max devices> maximum allowed devices. 42
High-capacity counters for network analysis are not available for device <device name>. Attempting to use low capacity counters. 42
The remote device reports that the requested name <name> does not exist on <channel name>.<device name>. 42
The response message for the current transaction on <channel name>.<device name> would have been too large, and has been discarded by the remote device. 43
Unable to bind trap socket on binding address <address>, port <port>, and protocol <protocol> for device <device>. 43
Unable to bind trap socket on binding address <IP address>, port <port number> and protocol <protocol> for device <device name>. 43
Unable to create communications thread on trap socket for binding address <IP address>, port <port number>, and protocol <protocol> for device <device name>. 44
Unable to create listener on trap socket for binding address <IP address>, port <port number> and protocol <protocol> for device <device name>. 44
Unable to create trap socket on binding address <IP address>, port <port number> and protocol <protocol> for device <device name>. 44
Unable to load authentication and privacy passphrases for device <device name>. Please specify an authentication and privacy passphrase in the SNMP V3 Security property group of Device Properties. 45
Unable to load authentication passphrase for device <device name>. Please specify an authentication passphrase in the SNMP V3 Security property group of Device Properties. 45

Unable to load username for device <device name>. Please specify a username in the SNMP V3 Security property group of Device Properties. 45
Unable to resolve host address <IP address> on device <device name> for trap processing. 46
Unable to send transaction: <reason>. 46
SNMP Agent Error Messages 47
Data for address <address> on <channel name>.<device name> has an inconsistent value. 47
Data for address <address>on <channel name>. <device name> has the wrong encoding. 47
Data for address <address>on <channel name>.<device name> has the wrong length. 47
Data for address <address>on <channel name>. <device name> has the wrong value. 48
XML Messages 48
Invalid XML document [Reason: The excluded port list is invalid for device <device name>]. 48
Invalid XML document [Reason: Port Status 0 limit must be less than Port Status 1 limit for device <device name>]. 48
Communications Messages 49
Unable to bind to adapter: <adapter address>. Connect failed. Winsock Err # n. 49
Winsock initialization failed (OS Error = n). 49
Winsock shut down failed (OS Error = n). 50
Winsock V1.1 or higher must be installed to use the SNMP device driver. 50
Authentication Messages 50
The authentication passphrase fields do not match. Please retype the passphrase identically in both fields. 50
The privacy passphrase fields do not match. Please retype the passphrase identically in both fields. 51
MIB Parser Messages 51
Cannot redefine macro name. 52
Cannot redefine primitive type. 52
Close IMPORTS statement with a ';'. 52
Could not add object: <object name>; parent object: <parent object name> undefined. 52
Could not find module: <module name> to import. 53
Could not obtain MIB module information. 53
DEFINITIONS must directly follow MIB module name. 53
End one module definition before beginning another. 53
Failed to open file: <file path>. 54
Invalid assignment value. 54
Invalid DESCRIPTION value. 54
Invalid ENTERPRISE value. 54
Invalid MAX-ACCESS value. 55
Invalid module name. 55
Invalid NOTIFICATION-TYPE clause. 55

Invalid object assignment. 55
Invalid OBJECT-IDENTITY clause. 55
Invalid OBJECT-TYPE clause. 56
Invalid OBJECTS value. 56
Invalid octet or bit string. 56
Invalid parent object name. 56
Invalid STATUS value. 57
Invalid SYNTAX value. 57
Invalid TRAP-TYPE assignment. 57
Invalid TRAP-TYPE clause. 57
Open bracket not closed. 58
Open parenthesis not closed. 58
Sub-identifier out of range: 0 to 4294967295. 58
Syntax Error. 58
Undefined identifier: <identifier name>. 59
Security Related Messages 59
<channel name>.<device name> reports a decryption error. Check the privacy passphrase. 59
<channel name>.<device name> reports the authentication digest is incorrect. Check the authentication passphrase. 59
<Channel name>.<device name> reports the request was not within the time window. 60
<channel name>.<device name> reports the specified security level is not supported. 60
<channel name>.<device name> reports the specified user is unknown. 60
<channel name>.<device name> responded to a request with a Report-PDU containing no valid data. 60
Index **62**

SNMP Driver

Help version 1.076

CONTENTS

Overview
What is the SNMP Driver?

Channel Setup
How do I configure the driver to search for devices on the network?

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

Data Types Description
What data types does the SNMP Driver support?

Address Descriptions
How do I reference a data location in an SNMP device?

Event Log Messages
What messages does the SNMP Driver produce?

Overview

The SNMP Driver provides a reliable way to connect managed and unmanaged Ethernet network devices to OPC client applications; including HMI, SCADA, Historian, MES, ERP, and countless custom applications. It is intended to work with all devices supporting the SNMP protocol (versions 1, 2c, and 3).

Channel Setup

Communication Serialization

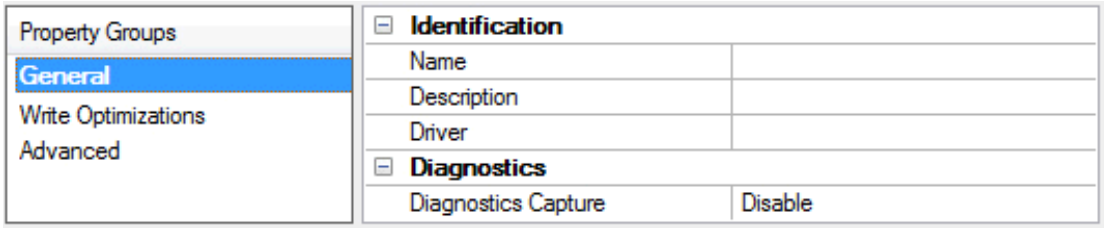
The SNMP Driver supports Communication Serialization, which specifies whether data transmissions should be limited to one channel at a time. For more information, refer to "Channel Properties - Advanced" in the server help file.

Maximum Number of Channels

The maximum number of channels is 1024.

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.



Identification

Name: User-defined identity of this channel. In each server project, each channel name must be unique. Although names can be up to 256 characters, some client applications have a limited display window when browsing the OPC server's tag space. The channel name is part of the OPC browser information.
For information on reserved characters, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in the server help.

Description: User-defined information about this channel.
Many of these properties, including Description, have an associated system tag.

Driver: Selected protocol / driver for this channel. This property specifies the device driver that was selected during channel creation. It is a disabled setting in the channel properties.
Note: With the server's online full-time operation, these properties can be changed at any time. This includes changing the channel name to prevent clients from registering data with the server. If a client has already acquired an item from the server before the channel name is changed, the items are unaffected. If, after the channel name has been changed, the client application releases the item and attempts to re-acquire using the old channel name, the item is not accepted. With this in mind, changes to the properties should not be made once a large client application has been developed. Utilize the User Manager to prevent operators from changing properties and restrict access rights to server features.

Diagnostics

Diagnostics Capture: When enabled, this option makes the channel's diagnostic information available to OPC applications. Because the server's diagnostic features require a minimal amount of overhead

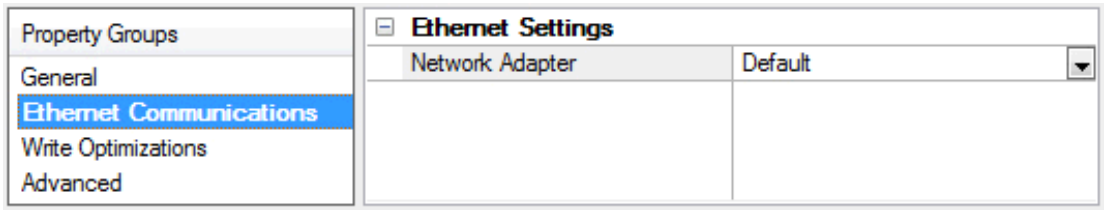
processing, it is recommended that they be utilized when needed and disabled when not. The default is disabled.

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 — Ethernet Communications

Ethernet Communication can be used to communicate with devices.

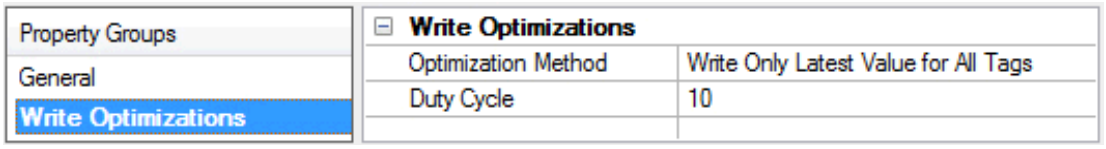


Ethernet Settings

Network Adapter: Specify the network adapter to bind. When Default is selected, the operating system selects the default adapter.

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.



Write Optimizations

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

- **Write All Values for All Tags**: This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing data to the device as quickly as possible. This mode ensures that everything written from the client applications is sent to the target device. This mode should be selected if the write operation order or the write item's content must uniquely be seen at the target device.
- **Write Only Latest Value for Non-Boolean Tags**: Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the server's internal write queue and sent to the device at the next possible opportunity. This can greatly

improve the application performance.

Note: This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.

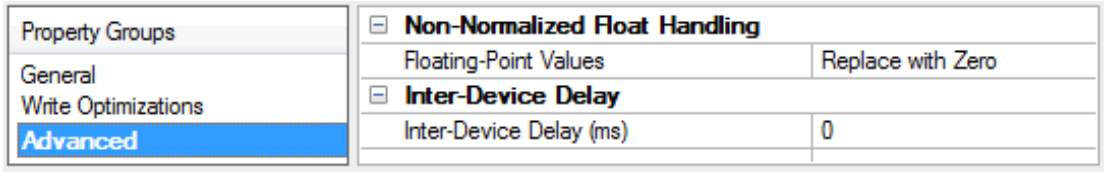
- **Write Only Latest Value for All Tags**: This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write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

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

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

Channel Properties — Advanced

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



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

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

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

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

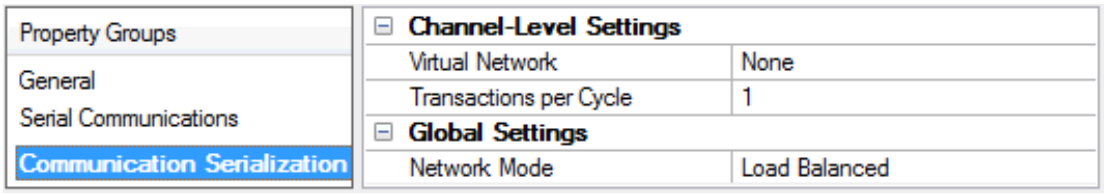
Inter-Device Delay: Specify the amount of time the communications channel waits to send new requests to the next device after data is received from the current device on the same channel. Zero (0) disables the delay.

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

Channel Properties — Communication Serialization

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

The term "virtual network" describes a collection of channels and associated devices that use the same pipeline for communications. For example, the pipeline of an Ethernet radio is the 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.



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 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.

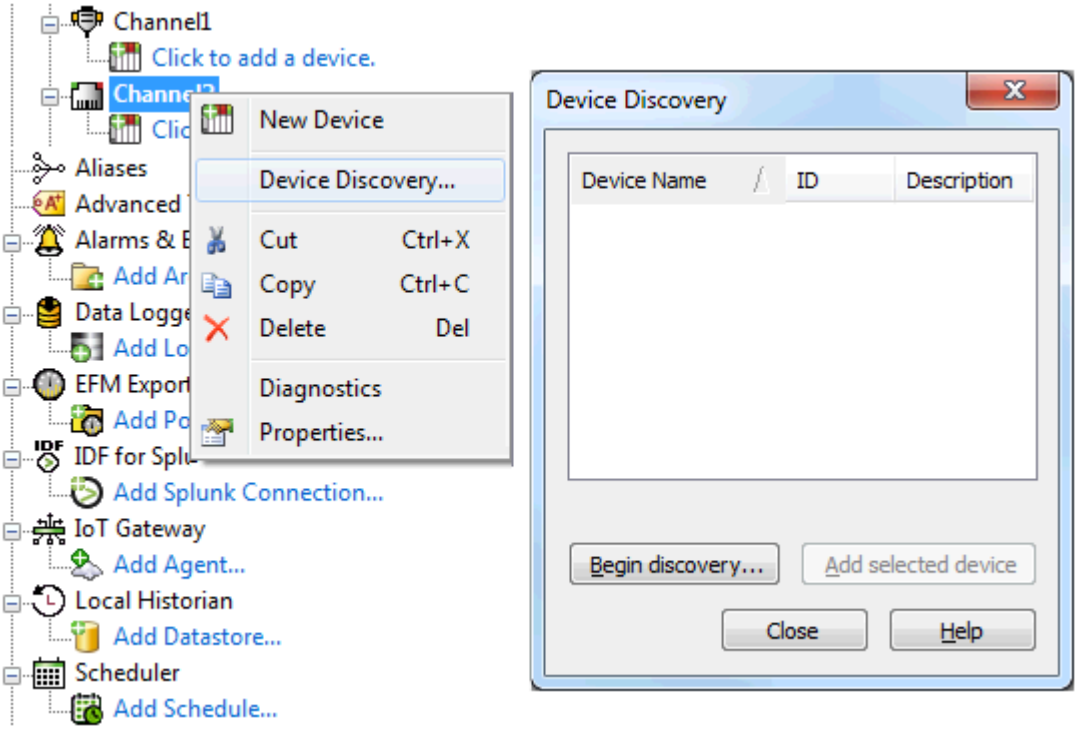
Devices that rely on unsolicited responses should not be placed in a virtual network. In situations where communications must be serialized, it is recommended that Auto-Demotion be enabled.

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 Discovery Procedure

Device Discovery is available for drivers that support locating devices on the network. Once devices are found, they may be added to a channel. The maximum number of devices that can be discovered at once is 65535.



1. Select the channel in which devices should be discovered and added.
2. Right click on the channel node and select **Device Discovery...**
3. Click the **Begin discovery...** button to start the discovery process.
4. Specify the discovery properties, which are driver-specific, such as address range, timeout, discovery scope.
5. Click **OK**.
6. Devices discovered populate the dialog with the following information / headings **Name**, **ID**, **Description**.

7. If any discovered device is of interest, select that device and click **Add selected device...**.
8. Click **Close**.

Device Setup

Supported Devices

The SNMP Driver is designed to work with any SNMP Agent (typically in a device) that supports the Simple Network Management Protocol (SNMP) versions 1, 2c, and 3. The driver works with a broad range of SNMP managed devices, such as the following:

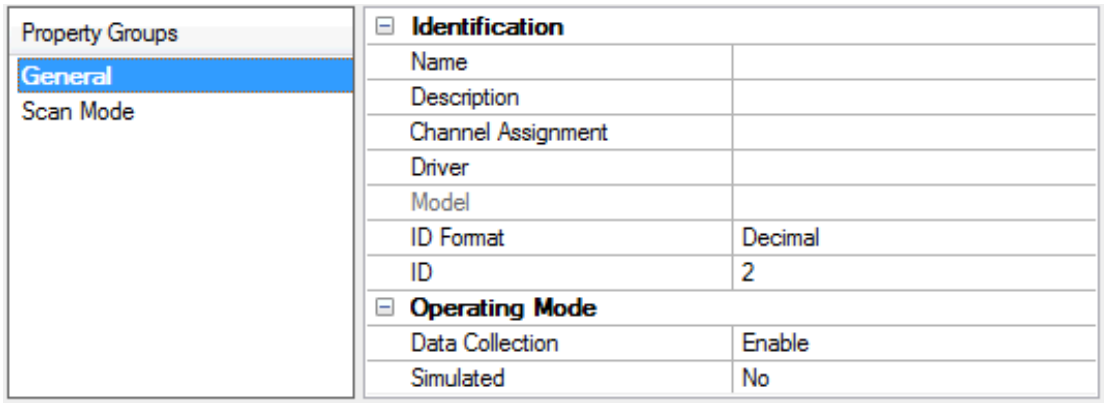
- Alarm Management RTUs
- Device Servers
- Environment Monitoring Equipment for Server Rooms
- Managed Industrial Ethernet Switches
- Net-SNMP Software Version: 5.4.1
- Printers
- Routers
- Uninterruptible Power Supplies (UPS)
- Unix-based Servers
- Windows-based PCs and Servers

Maximum Number of Devices

The maximum number of devices supported per channel is 100.

Device Properties — General

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



Identification

Name: 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. This property specifies the driver selected during channel creation. It is disabled in the channel properties.

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

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.

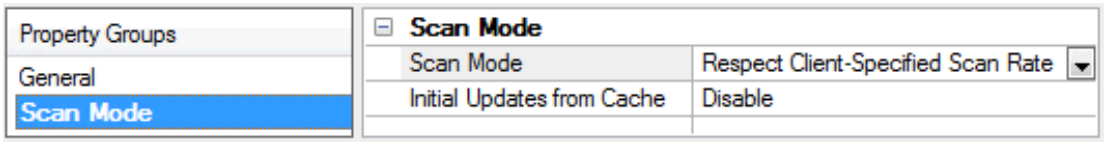
Notes:

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.



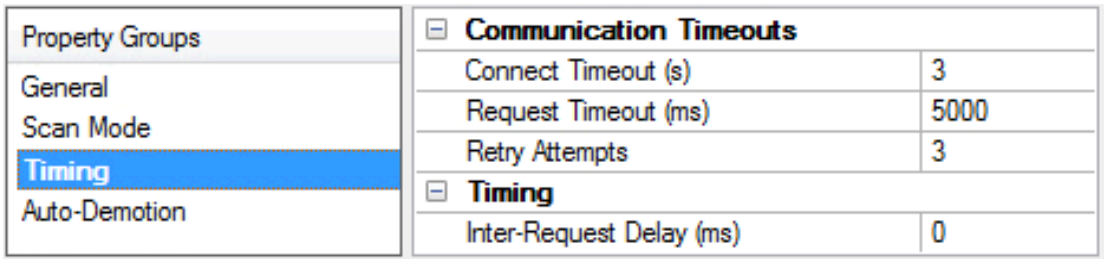
Scan Mode: specifies how tags in the device are scanned for updates sent to subscribing clients. Descriptions of the options are:

- **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 maximum scan rate to be used. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
  Note: When the server has an active client and items for the device and the scan rate value is increased, the changes take effect immediately. When the scan rate value is decreased, the changes do not take effect until all client applications have been disconnected.
- **Request All Data at Scan Rate**: This mode forces tags to be scanned at the specified rate for subscribed clients. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
- **Do Not Scan, Demand Poll Only**: This mode does not periodically poll tags that belong to the device nor perform a read to get an item's initial value once it becomes active. It is the client's responsibility to poll for updates, either by writing to the _DemandPoll tag or by issuing explicit device reads for individual items. *For more information, refer to "Device Demand Poll" in server help*.
- **Respect Tag-Specified Scan Rate**: This mode forces static tags to be scanned at the rate specified in their static configuration tag properties. Dynamic tags are scanned at the client-specified scan rate.

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

Device Properties — Timing

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



Communications Timeouts

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

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

Request Timeout: 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

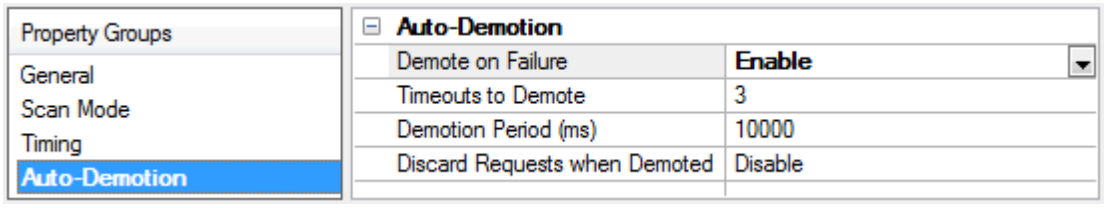
Inter-Request Delay: This property specifies how long the driver waits before sending the next request to the target device. It overrides the normal polling frequency of tags associated with the device, as well as one-time reads and writes. This delay can be useful when dealing with devices with slow 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.



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

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

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

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

Device Properties — Tag Generation

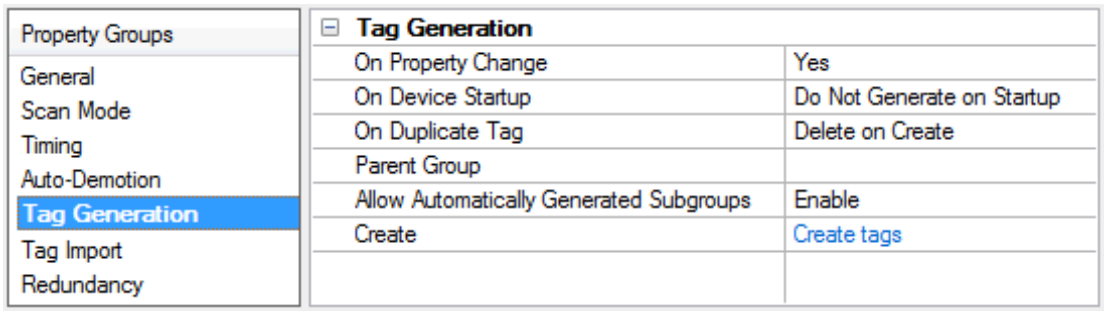
The automatic tag database generation features make setting up an application a plug-and-play operation. Select communications drivers can be configured to automatically build a list of tags that correspond to device-specific data. These automatically generated tags (which depend on the nature of the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

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

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

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

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



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

On Device Startup: 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.

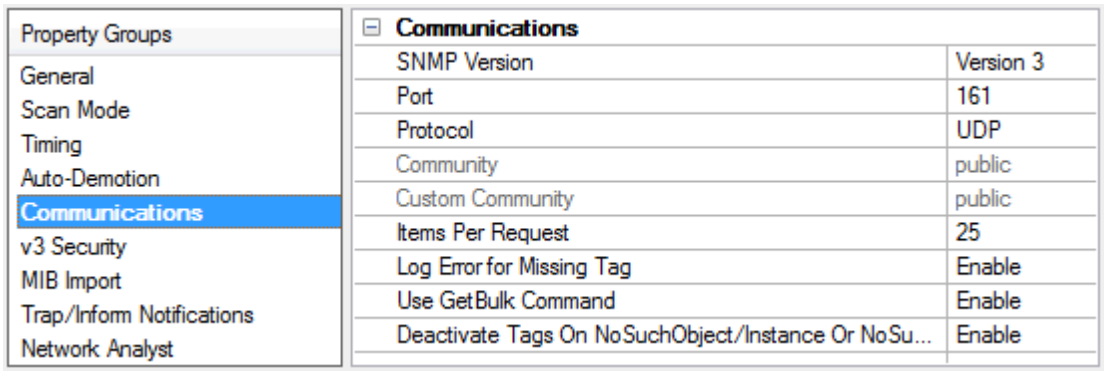
Allow Automatically Generated Subgroups: This property controls whether the server automatically creates subgroups for the automatically generated tags. This is the default setting. If disabled, the server generates the device's tags in a flat list without any grouping. In the server project, the resulting 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 — Communications



SNMP Version: Specify the version that will be used by the remote device. Options include Version 1, Version 2c, and Version 3. The default setting is Version 2c.

Port: Specify the port. The valid range is 1 to 65535. The default setting is 161.

Protocol: Specify the protocol. Options include UDP and TCP. The default setting is UDP.

Community: This property is used when accessing the remote SNMP device. The community name can be defined by the user and depends entirely on the configuration of the remote device. Common options include "public" and "private". The "public" community is usually used for reading data, whereas the "private" community is used for writing data to an Agent. This field is limited by the driver to 256 characters.
For information on determining the correct community name, refer to the device manufacturer documentation.

Items per Request: This property controls how many SNMP data items will be bundled together in each read request. For Agents or devices supporting SNMP v1, this may need to be set to a value as low as 1. SNMP version 2c devices can typically handle the maximum items per request. The valid range is 1 to 25. The default setting is 25.

Log Error for Missing Tag: An SNMP Agent or device is dynamic and may change during operation. When enabled, this property has the OPC server display an error notice when a specified OID address does not exist on the target device. When unchecked, the messages will be suppressed. The default setting is enabled.

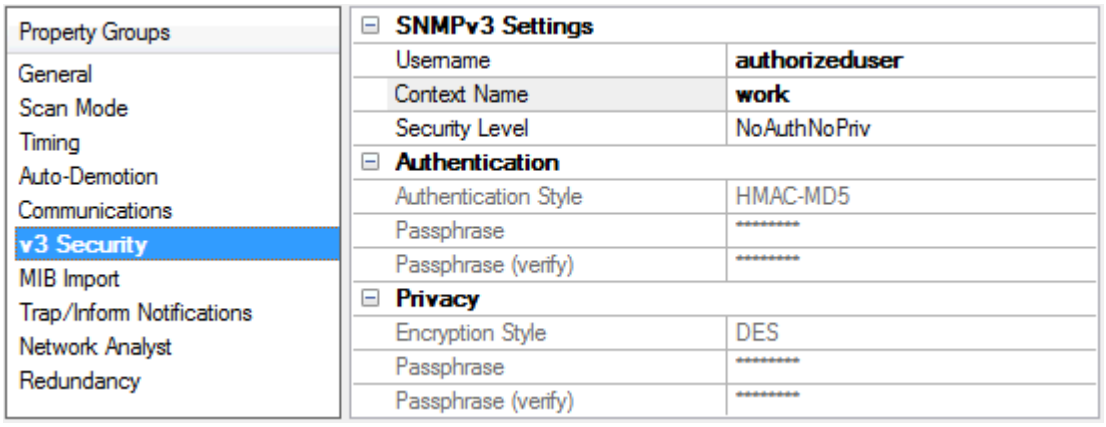
Use GetBulk Command: This command applies to OID addresses ending with the *[1-n]* table offset notation. When enabled, the SNMP GetBulk command will obtain table data from the device (Agent MIB) by packaging multiple Get-Next commands in a single request to the Agent. The GetBulk command is more efficient than individual Get-Next commands.
Note: The GetBulk command is not supported in the SNMP Version 1 specification. The driver uses individual Get-Next commands to retrieve table data from Version 1 Agents *(see the table below)*.

Agent Version	Table Data	SNMP Command	# Requests Sent to Agent
1	.1.3.6.1.4.1.30144.1.1.2[1] .1.3.6.1.4.1.30144.1.1.2[2] .1.3.6.1.4.1.30144.1.1.2[3] .1.3.6.1.4.1.30144.1.1.2[4]	SNMP Get-Next	4
2c/3	.1.3.6.1.4.1.30144.1.1.2[1] .1.3.6.1.4.1.30144.1.1.2[2] .1.3.6.1.4.1.30144.1.1.2[3] .1.3.6.1.4.1.30144.1.1.2[4]	SNMP GetBulk	1
2c/3	.1.3.6.1.4.1.30144.1.1.2[1] .1.3.6.1.4.1.30144.1.1.2[2] .1.3.6.1.4.1.30144.1.1.2[3] .1.3.6.1.4.1.30144.1.1.2[4] .1.3.6.1.4.1.30144.1.1.3[1] .1.3.6.1.4.1.30144.1.1.3[2] .1.3.6.1.4.1.30144.1.1.3[3] .1.3.6.1.4.1.30144.1.1.3[4]	SNMP GetBulk	2

Deactivate Tags on NoSuchObject/Instance or NoSuchName errors: When enabled, this property will deactivate tags on NoSuchObject, NoSuchInstance, or NoSuchName errors. The default setting is disabled.
Note: This property is not always desirable. For example, a device may provide a NoSuchObject error for one condition but provide valid data for another. This property applies to normal SNMP OID polling and polling that occurs for Network Analyst tags. If there are many tags for SNMP OIDs that continuously result in NoSuchName errors, disabling this setting may significantly affect the performance of SNMP Driver.

Device Properties — v3 Security

The SNMP V3 Security settings are only available when Version 3 is selected as the SNMP version in **Communications**.



SNMPv3 Settings

Username: Specify the username that will be associated with the authorization and privacy keys. It is blank by default.

Note: If a device sending SNMP version 3 traps uses a different username, a second device with the user credentials for the trap receiver can be used to receive the traps. This means that each device in the server can only be associated with one set of user credentials. Users can have multiple devices with the same credentials; however, one set of credentials has no effect on another because user credentials are tied to the device.

Context Name: Specify a contextual name for the SNMP message request. It is blank by default.

Security Level: Specify the security level. Options include NoAuthNoPriv, AuthNoPriv, and AuthPriv. The default setting is NoAuthNoPriv. Descriptions of the options are as follows:

- **NoAuthNoPriv**: This level includes neither authentication nor encryption.
- **AuthNoPriv**: This level includes authentication, but not encryption.
- **AuthPriv**: This level includes both authentication and encryption.

  Note: When the Security Level is set to AuthNoPriv or AuthPriv, the following properties will be available for configuration.

Authentication

Authentication Style: Specify the style of authentication. Options include HMAC-MD5 and HMAC-SHA1. The default setting is MHMAC-MD5.

Passphrase: This property generates a localized key that is used to authenticate the SNMP data frames.

Passphrase (Verify): This property is used to verify the previously entered passphrase.

Privacy

Encryption Style: Specify the style of encryption. Options include DES, AES 128, AES 192, and AES 256. The default setting is DES.

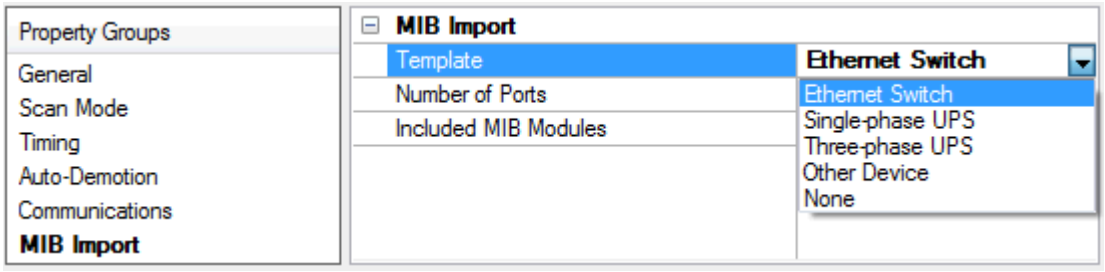
Note: AES 192 and AES 256 are non-standard extensions of the SNMP User Security Model (USM) and are not supported by all SNMP V3 Agents.

For more information on the key expansion algorithms for AES 192 and AES 256, refer to the SNMP V3 Working Group's Internet-Draft ***Extension to the User-Based Security Model (USM) to Support Triple-DES EDE in "Outside" CBC Mode***.

Passphrase: This property generates a localized key that is used to encrypt/decrypt the data in an SNMP frame.

Passphrase (Verify): This property is used to verify the previously entered passphrase.

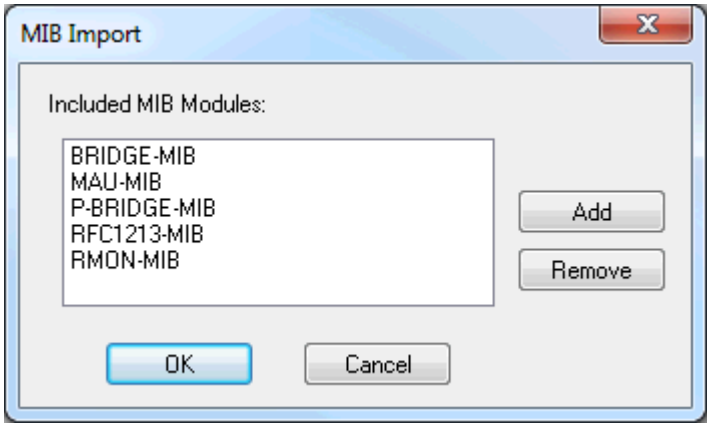
Device Properties — MIB Import



Template: Specify the template to guide the automatic creation of tags for the new device. Options include Ethernet Switch, Single-phase UPS, Three-phase UPS, Other Device, and None. Other Device creates a generic set of tags for a multi-port SNMP-enabled device. None has no associated preset tag set.

Number of Ports: All templates (except for UPS) must enter the number of Ethernet ports on the device. Tags will be generated for each port present. The valid range is 0 to 99. The default setting is 0.

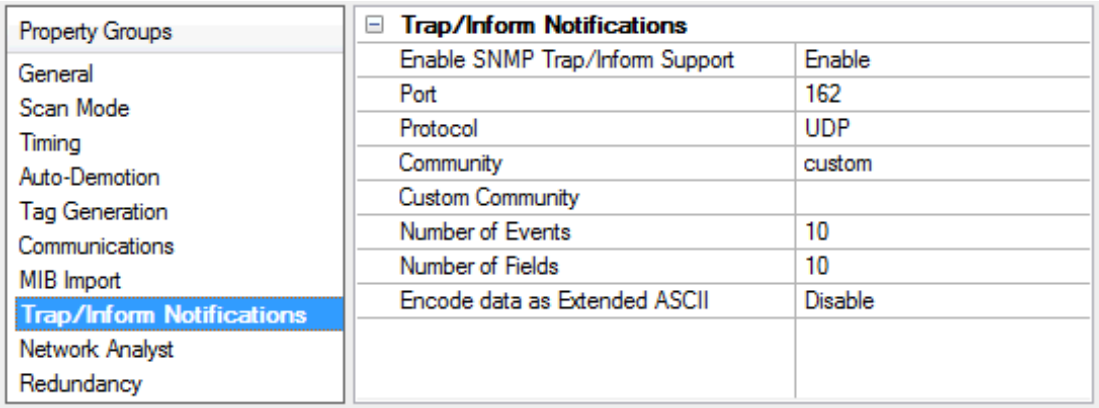
Included MIB Modules: The list of included MIB Modules is displayed as a semi-colon delimited string. To modify this list, click the browse (**...**) button to launch the MIB Import dialog with the option to Add or Remove MIBs. *For more information, refer to* ***About MIB Modules***.



Device Properties — Trap / Inform Notifications

SNMP managed devices can be configured to send unsolicited messages (known as traps, informs, or notifications) to host systems or managers.

Note: The SNMP Driver supports Trap-PDU (SNMPv1 only), SNMPV2-Trap-PDU (SNMPv2c/V3 only), and the Inform-Request-PDU (SNMPv2c/V3 only).



Enable SNMP Trap/Inform Support: Choose Enable for the SNMP Driver to receive traps sent from SNMP managed devices or systems. The default setting is enabled.
Note: Trap support cannot be enabled when the SNMP channel is part of a virtual network. *For more information on communication serialization, refer to the server help file.*

Port: Specify the port on which the device will listen for notifications. The valid range is 1 to 65535. The default setting is 162, which is the most commonly used port for sending and receiving traps.

Protocol: The protocol may be UDP or TCP. The default is UDP.

Community: Leave this property set to custom to specify a **Custom Community**.

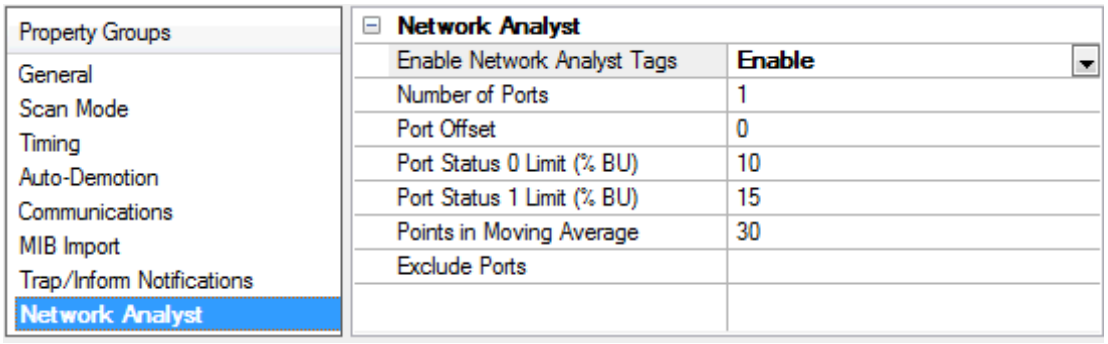
Custom Community: This is an optional setting. If a community name is entered, the SNMP Driver will only accept trap messages addressed to that community. In addition, traps will only be accepted from the IP address configured in the OPC server device. Entering no community information will allow trap messages to be received that are addressed to any community (or none at all). The community is limited to 256 characters.
Note: For SNMP version 3, the specified username and passphrase for normal communications are used to authenticate, encrypt, and validate the SNMP message. Messages for a different user are ignored.

Number of Events: Trap messages are provided to client applications via an event queue in the driver. The queue is a FIFO stack that displays several trap messages that were received last. Specify the amount of trap messages to retain in the queue. The driver allows between 1 and 100 events to be collected. The default setting is 10.

Number of Fields: Each trap message may carry additional variables, which are then parsed into a number of individual tag fields. The default setting is 10. It is recommended that users choose the maximum number to allow extra fields for the server-generated timestamp and a generic trap description (which is only for SNMP version 1). The driver allows between 1 and 20 fields. For more information on trap message addressing, refer to **Trap Events Queue** and **Trap Tags**.

Encode data as Extended ASCII: Enable for non-printable ASCII characters to be encoded and displayed as extended ASCII.

Device Properties — Network Analyst



Enable Network Analyst Tags: When enabled, network analyst tags are made available with the Ethernet Switch and Other Device profiles. For more information, refer to **About Network Analyst Tags**.

Number of Ports: Specify the number of ports for the switch device. This is separate from the port number setting in Profile Selection. The valid range is 1 to 99.

Port Offset: Specify the offset that will be added to the Network Analyst port when polling the special OIDs. The valid range is 0 to 65436. The default setting is 0.

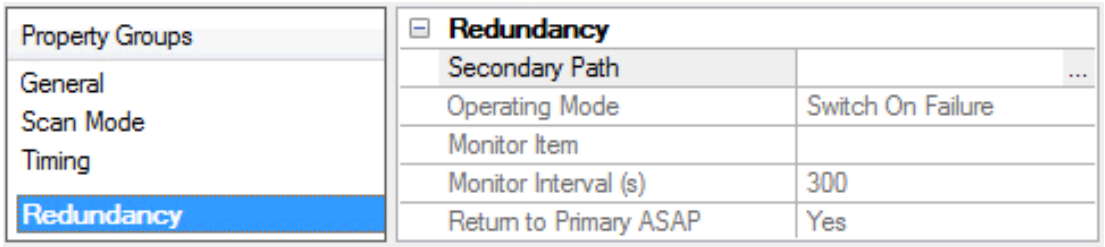
Port Status 0 limit and **Port Status 1 limit**: These properties specify the threshold settings for each switch port's buStat tags. The buStat tags are a three-state indicator of the rough class of utilization for incoming bandwidth. When the buPctIn for a port rises above the Port Status 0 limit, that buStat tag will change from 0 to 1. Similarly, when the buPctIn rises above the Port Status 1 limit, the buStat tag will change from 1 to 2. The valid range is 0 to 100. The Port Status 0 limit should not be greater than Port Status 1 limit.

Points in Moving Average: Specify how many sample values will be used when calculating the buPctIn and buPctOut values. The data points' average is taken to smooth the Ethernet traffic's inherently erratic behavior. The number of points in the moving average can be from 1 to 200. The default setting is 30.

Exclude Ports: This property allows the switchBUStat tag to ignore some ports when calculating the highest buStat value. This is a list (1, 3, 6, 8) that can also contain ranges (1, 3-7, 9-11).

***See Also**: About Network Analyst Tags*

Device Properties — Redundancy



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

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

Data Types Description

The SNMP Driver supports the below data types.

Data Type	Description
Boolean	Single bit
DWord	Unsigned 32-bit value bit 0 is the low-bit bit 31 is the high bit
DWord Example	The driver interprets two consecutive registers as a single precision value.
Long	Signed 32-bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
Long Example	The driver interprets two consecutive registers as a single precision value.
String	ASCII text string
Float	32-bit floating point value bit 0 is the low bit bit 31 is the high bit
Float Example	The driver interprets two consecutive registers as a single precision value.
Double	64-bit floating point value bit 0 is the low bit bit 63 is the high bit
Double Example	The driver interprets four consecutive registers as a double precision value.

Each tag used in the driver has a fixed data type when there is MIB information for the address. Therefore, it is recommended that the driver be allowed to use the default data type for the point.

In a few cases, SNMP-centric data types do not exist in standard OPC. These items should be mapped or correlated to a valid OPC data type to be read. Extensive testing has been performed to assure that SNMP-centric data types can be served to and written from correctly with OPC client applications.

SNMP Centric	OPC Data Type
Integer32	Long
UInteger32	DWord
Counter64	NS*
Octet String	String
Bits	NS**
Object Identifier	String
Sequence	NS***
IPAddress	DWord

SNMP Centric	OPC Data Type
Counter32	DWord
Guage32	DWord
Timeticks	DWord
Opaque	NS****
Trap/Notification	String

*This is a 64-bit integer.
**Bit string.
***A sequence is a list of data. Complex data is currently not supported in OPC.
****Opaque data is a memory BLOB.

Note: There is no corresponding data type in OPC to handle these data types.

Historical Data Attributes

Addresses may be accompanied by one of three modifiers to access historical attributes. Historical values are generated by the SNMP Driver (not the remote Agent or device) when valid historical modifiers append to an OID. For more information, select a link from the list below.

Previous Value (PV)
Delta Time (DT)
Moving Average (MA5)

Previous Value

The Previous Value historical attribute returns the value of the SNMP address from the previous read cycle. This is not the previous differing value. If the address data has not changed, the previous value will be the same as the current value.

(Module::Object notation)
RFC1213-MIB::ifOutOctets.1(PV)

(Numeric notation)
.1.3.6.1.2.1.2.2.1.16.1(PV)

(Verbose notation)
.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifOutOctets.1(PV)

Delta Time

The Delta Time historical attribute returns the time difference between the current and previous read cycle, and is expressed in whole seconds for compatibility with legacy projects. Delta values of less than 1 second will report as 0.

(Module::Object notation)
RFC1213-MIB::ifOutOctets.1(DT)

(Numeric notation)
.1.3.6.1.2.1.2.2.1.16.1(DT)

(Verbose notation)
.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifOutOctets.1(DT)

Moving Average

The Moving Average historical attribute returns the average of the last *n* readings, as specified in the address modifier. The modifier form is Max, where *x* is the number of points to use in calculating the moving average. Values for *x* may be anything larger than 1. If the *x* value is left out, the moving average calculation defaults to 5 points.

(Module::Object notation)
RFC1213-MIB::ifOutOctets.1(MA5)

(Numeric notation)
.1.3.6.1.2.1.2.2.1.16.1(MA5)

(Verbose notation)

.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifOutOctets.1(MA5)

Address Descriptions

Addresses in the SNMP Driver are specified by the Object Identifier (OID) followed by an instance number. The OID can be defined in one of several forms and as follows:

Object Identifier	Description
SNMPv2-MIB::sysDescr.0	(Module::Object notation)
.1.3.6.1.2.1.1.1.0	(Numeric notation)
.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0	(Verbose notation)

Note: For more information about address structure, refer to **About SNMP Addresses**.

Table Offsets

To address an SNMP Table, specify the OID of the table head followed by the table offset (in brackets).

IF-MIB::tcpConnState[1]

Note: All SNMP table offsets begin at 1. Tags addressed to table offsets beyond the end of the table will be reported with bad quality until the table grows to that offset or beyond.

Historical Data

Each SNMP address has one or more historical data options available. Historical values are generated by the SNMP Driver, not the remote Agent or device.

***See Also**: Previous Value, Delta Time, Moving Average*

String Data

Strings that contain non-printable characters will be displayed as hexadecimal by default. Any character outside the ASCII range of 0x20 to 0x7E is considered non-printable. To keep strings from being converted to hexadecimal, add "(EncExtAsc)" to the end of the address description (without the quotation marks).

Unsolicited Data

SNMP-enabled devices may be configured to send unsolicited messages, called traps (or notifications). For more information, refer to **Trap Events Queue** and **Trap Tags**.

Scan Rate Floor

The scan rate can be set in milliseconds for each SNMP device. The _ScanRateFloor Tag will display the setting's current value. When it is set greater than zero, the SNMP Driver will not allow tags to be scanned faster than specified. The device can also be set to lock the scan rate at this value, prohibiting any change by the OPC client. The _ScanRateFloorLock Tag will show the lock option's status. The tags are Read Only.

Note: Setting this feature to zero will disable it.

About SNMP Addresses

The Simple Network Management Protocol accesses information in a Management Information Base (MIB). The MIB is a tree structure whose origin is at the top, which is a node labeled ".1" or ".iso." Although many discussions of SNMP refer to MIBs as a plural, there is only one. The plural references actually refer to MIB modules, which describe portions of the MIB tree.

The SNMP address is known as an Object Identifier (OID) and consists of a series of elements that describes its location in the MIB tree. The elements are separated by a character referred to as dots ('.'). Most addresses of interest will begin with *.iso.org.dod.internet.mgmt* (or .1.3.6.1.2). From that point, the address extends into particular modules that describe related sets of information. For example, consider the IF-MIB module: it contains a variety of objects' definitions that access data about the network interfaces of the remote device. These include port Status, traffic counters, and so forth.

The *Module::Object* syntax of SNMP addresses means that "IF-MIB::" can be written instead of ".iso.org.dod.internet.mgmt.mib-2.interfaces" (or ".1.3.6.1.2.1.2.2"). The address "IF-MIB::ifOutOctets.1" refers to the number of octets (bytes) sent out of interface 1 on the target device. That form is easier to write than ".iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifOutOctets.1" or ".1.3.6.1.2.1.2.2.1.16.1". The SNMP Driver will accept all three of these address notations.

Enterprise or Private MIB Modules

Much of the SNMP address space is defined by Internet RFC standards. Individuals are not permitted to change or extend these module definitions. For that purpose, the SNMP standard provides an extension area of the address space under ".iso.org.dod.internet.private.enterprises". The value following this base is known as a Private Enterprise Number (PEN) and every address below that point is defined by the PEN owner. Manufacturers that need to provide unique information not otherwise described in standard MIB modules will need to define them in their own Enterprise space and typically supply a MIB module definition with their equipment. The SNMP Driver uses these supplied MIB definitions to correctly access the unique information in remote device.

Instances

The OID "IF-MIB::ifOutOctets.1" above provides an example of SNMP instances. A managed switch will have a set of "IF-MIB::ifOutOctets" OIDs, one for each network interface. They will use a trailing digit (or digits) to index into the set of instances. Instances may be numbered beginning at 1 for groups that map to physical attributes, such as "IF-MIB::ifOutOctets.1," "IF-MIB::ifOutOctets.2," "IF-MIB::ifOutOctets.3" and so forth. The number of instances for a given OID is typically fixed. Other OIDs may have multiple instances, such as "SNMPv2-MIB::sysLocation". Although the first instance will be "SNMPv2-MIB::sysLocation.0," an agent may optionally provide "SNMPv2-MIB::sysLocation.1" and so on.

Note: Instances should not be confused with tables.

Tables

The SNMP address space is dynamic. The SNMP Agent on the remote device may add and remove OIDs as necessary. The most frequent occurrence of this is in SNMP Tables. An SNMP Table is a grouping of logically related data into conceptual rows. The rows are conceptual because the SNMP protocol does not have a facility to retrieve a full row at a time. Table access is accomplished by enumerating a table's columns. The SNMP Driver uses an array-like notation for table access, as in "RFC1213-MIB::tcpConnState[1]". That OID is part of the "tcpConnTable". Tables differ from instances in the following two ways:

1. Tables may grow or shrink during operation. An SNMP Driver tag that references a table column element will lose data quality if the table shrinks to less than the referenced element (offset).
2. The OIDs representing table column elements are not necessarily consecutive. The OIDs for individual column elements may not be predictable, and may change from moment to moment in the Agent or device.

Device Implementation RFC-Standard Modules

SNMP has defined a large and rich set of data that may or may not be implemented in SNMP-enabled devices. Although many device manufacturers implement the complete MIB module definition, others do

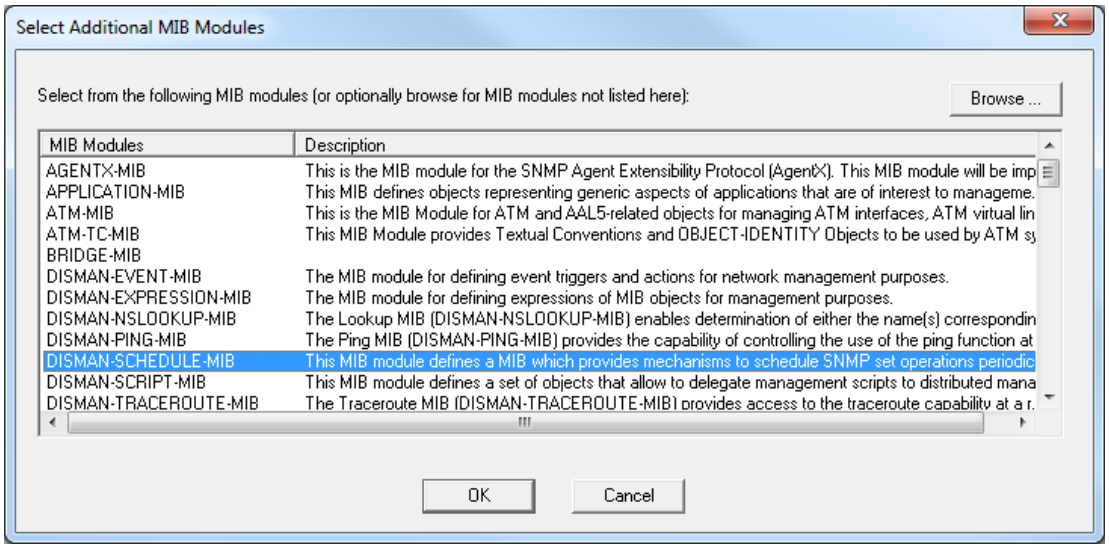
not. If the SNMP Driver is able to poll some but not all of the OIDs defined in the server project, users should start by verifying what OIDs are fully supported in the remote device.

Community Credentials

There is also the question of the credentials used to connect to the SNMP device (the community name), and whether those credentials have permission to access certain data. The final authority for the presence and accessibility of an OID lies with the remote device. For more information, refer to the device's help documentation.

About MIB Modules

Much of the SNMP address space is defined by Internet RFC Standards. These standards break up the address space into modules, many of which are drawn from the RFC standards. Selecting a device template also selects a number of MIB modules to be referenced. Additional MIB modules may be associated with a device to support specialized capabilities. The SNMP Driver ships with a number of MIB modules pre-installed. To access these MIB definitions, click **Add** on the **SNMP MIB Import Settings** wizard page. Then, click **OK**.



Adding New MIB Modules

New MIB definitions, such as MIB modules supplied by a manufacturer, may be installed by clicking **Browse...** to import. Navigate to the MIB definition file and then click **Open**. The MIB definition will be checked for correctness and its description will be displayed if present. To accept the file for import, click **OK**. The module will be added to the current project and tags will be created for the objects that are defined.

Notes:

1. If the selected MIB module is already present in the repository, the relative dates of the two versions will be displayed. The user will be given the option to replace the module.
2. If a MIB module contains errors, it cannot be imported. The import process automatically considers all MIB definition files in the same folder with the import candidate, and will bring in additional files if needed. Be sure that all MIB files associated with the device are present in the folder.
3. Adding or importing a MIB module does not guarantee that new tags will be created. Some MIB modules (including those supplied by manufacturers) do not define any accessible objects.

About Network Analyst Tags

Ethernet switches carry traffic around networks. The SNMP Driver features a set of Network Analyst tags to easily keep track of a switch's capacity and utilization. These tags track the percentage of bandwidth in use on each switch's ports at any given time.

The buPctIn and buPctOut tags show the usage of each port in percent, averaged over a number of sample periods. The OPC client's scan rate is the sample period. For best results, the scan rate should be at least 1000 milliseconds. Longer periods are acceptable, whereas shorter periods may cause network congestion (because a number of SNMP data points must be read on each sample). The readings are averaged to smooth out the Ethernet traffic's inherently erratic behavior and make the values more useful for alarming.

The buStat tags utilize the threshold settings Port Status 0 limit and Port Status 1 limit to present a basic three-state "health" indicator. When a given port's buPctIn tag rises above the 0 limit, the buStat changes from 0 to 1. Likewise, when buStat rises above the 1 limit, buStat changes to 2. This provides a basic "traffic light" style, indicating the available capacity.

The switchBUStat tag assumes the highest value of the buStat tags, giving a single indication of the device's available capacity. The switchBUStat tag's behavior may be altered through the use of a list of ports to exclude. For example, a switch may have two ports that always run at or near capacity. By excluding these two ports, switchBUStat can indicate when the rest of the switch's capacity is nearing exhaustion without the known high-capacity activity causing false alerts.

Note: When enabled, the SNMP Driver will automatically create Network Analyst tags for a switch device.

Trap Tags

Trap tags are a notification mechanism for incoming trap messages, which may be generic or Enterprise-specific.

Version 1 Trap Tags

The syntax for a generic SNMP Version 1 trap tag is as follows:

TRAP_V1:.1.3.6.1.2.1.11:Gx

All V1 generic traps use this same OID. The ':Gx' field specifies the generic trap to which it is subscribed. Valid values for x are as follows:

coldStart: 0
warmStart: 1
linkDown: 2
linkUp: 3
authenticationFailure: 4
egpNeighborLoss: 5

For Enterprise-specific traps, the Enterprise OID is used in place of the generic OID in addition to a ':G6' field. Trap type 6 also requires a specific trap type, using the notation ':Sx' where x is the specific trap number. For example, an Enterprise-specific address may appear as follows:

TRAP_V1:.1.3.6.1.2.1.17:G6:S2

Note: For information on which Enterprise-specific traps may be sent, refer to the device manufacturer's help documentation.

To reset Boolean tags that transition to 1 on trap reception, users can write 0. To reset the notification tag for OPC clients who receive OnDataChange events for subsequent trap messages, users can write a 0 or a FALSE value.

Additionally, linkUp, linkDown and Enterprise traps may use the ':Px' field to specify which port will be monitored on the switch device. Enterprise traps must provide an "ifIndex" varbind for this to be useful. An incoming trap will populate both the port specific tag and the base tag. For example, a tag that monitors for linkDown on port 3 is as follows:

TRAP_V1:.1.3.6.1.2.1.11:G2:P3

Version 2c Trap Tags

The syntax for a generic SNMP Version 2C trap uses a set of OIDs in place of the ':Gx' field.

coldStart: .1.3.6.1.6.3.1.1.5.1
warmStart: .1.3.6.1.6.3.1.1.5.2
linkDown: .1.3.6.1.6.3.1.1.5.3
linkUp: .1.3.6.1.6.3.1.1.5.4
authenticationFailure: .1.3.6.1.6.3.1.1.5.5

Note: egpNeighborLoss generic traps are not implemented in SNMP Version 2C.

For example, a tag to monitor for linkDown on port 3 is as follows:

TRAP_V2C:.1.3.6.1.6.3.1.1.5.3:P3

Version 2C Enterprise-specific traps use the OID that the remote device places in the snmpTrapOID.0 varbind field. No specific trap field is used. Version 2C doesn't use the specific trap property. For information on which Enterprise-specific traps may be sent, refer to the device manufacturer's documentation.

SNMP Version 2c and Version 3 Informs

Some SNMPv2c and SNMPV3 devices may support Informs (confirmed traps). For convenience, all SNMP Version 2c Trap Tag definitions can be used for both traps and Inform-requests: this does not require that the SNMP Driver be configured to receive SNMP informs.

Additional Functionality

All trap tags may use a table-like syntax for accessing additional trap information. The virtual table fields are as follows:

[1] Local time stamp, generated on trap arrival (string).
[2] Enterprise OID (string).
[3] Generic trap type (int).
[4] Specific trap type (int, 0 unless the generic type is 6).
[5] SysUpTime (in timeticks, not a time stamp).
[6] Number of varbind items.
[7] First varbind OID (as string).
[8] First varbind value (as string).
[9]..[n] Successive varbinds.

All the virtual table tags are Read Only. Automatic Tag Generation provides a number of virtual table tags by default.

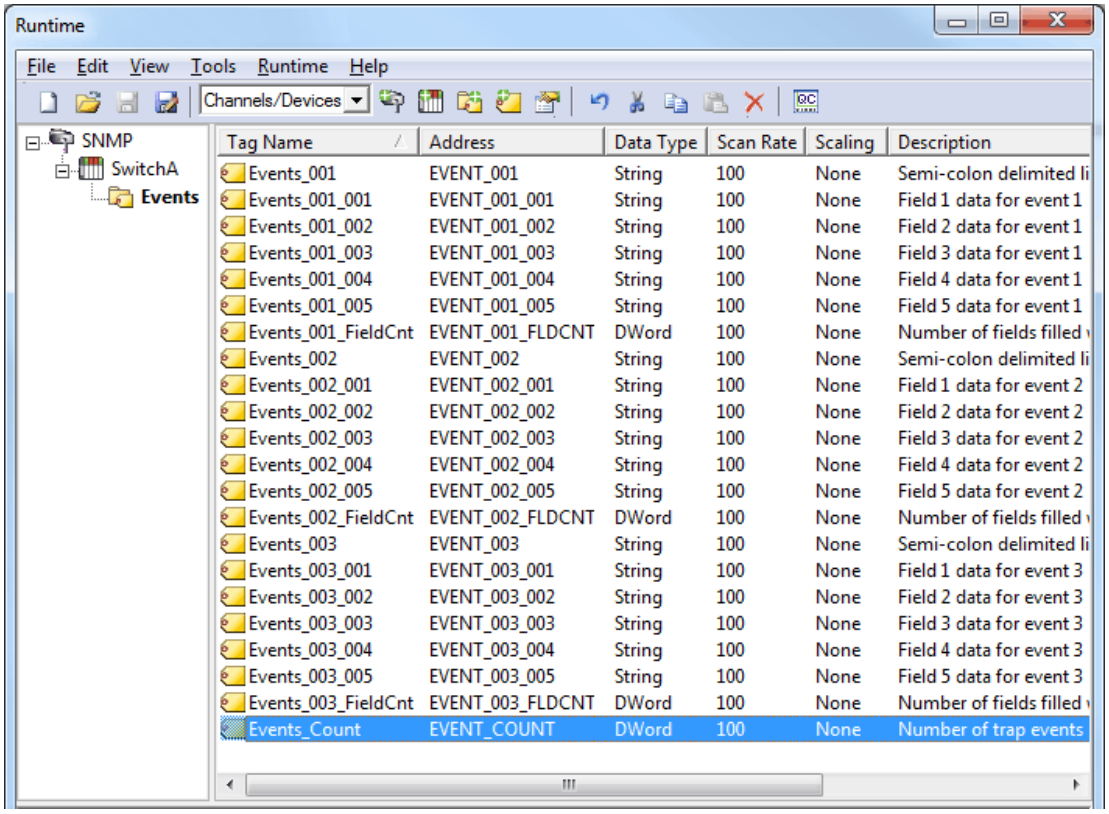
Notes:

1. Virtual table entry [5], sysUpTime, refers to the trap event's time-of-occurrence. This is expressed as the number of timeticks beginning when the remote SNMP agent started. It does not represent any specific wall/clock time.
2. Although the older trap syntax (which is the OID to be monitored followed by a (T) modifier) is deprecated, it is still supported. The older syntax does not support the virtual table information.

Trap Events Queue

SNMP remote devices may be configured to send unsolicited messages back to the SNMP Driver. To configure traps, users must login to the device to check the SNMP settings and then enable the traps. This includes defining Host IP(s) to receive the trap notifications. Since configuration changes usually require warm or cold restart of the device, users should check related network dependencies before performing a restart. Description of the messages are as follows:

- **Receiving Trap Messages**: These messages are configured during SNMP Driver setup. They may also be referred to as Notification messages. *For more information, refer to* ***Communications Properties***.
- **Incoming Trap Messages**: These messages are placed into an Events queue. The most recent message is placed at position 1.



Tag Name	Address	Data Type	Scan Rate	Scaling	Description
Events_001	EVENT_001	String	100	None	Semi-colon delimited li
Events_001_001	EVENT_001_001	String	100	None	Field 1 data for event 1
Events_001_002	EVENT_001_002	String	100	None	Field 2 data for event 1
Events_001_003	EVENT_001_003	String	100	None	Field 3 data for event 1
Events_001_004	EVENT_001_004	String	100	None	Field 4 data for event 1
Events_001_005	EVENT_001_005	String	100	None	Field 5 data for event 1
Events_001_FieldCnt	EVENT_001_FLDCNT	DWord	100	None	Number of fields filled
Events_002	EVENT_002	String	100	None	Semi-colon delimited li
Events_002_001	EVENT_002_001	String	100	None	Field 1 data for event 2
Events_002_002	EVENT_002_002	String	100	None	Field 2 data for event 2
Events_002_003	EVENT_002_003	String	100	None	Field 3 data for event 2
Events_002_004	EVENT_002_004	String	100	None	Field 4 data for event 2
Events_002_005	EVENT_002_005	String	100	None	Field 5 data for event 2
Events_002_FieldCnt	EVENT_002_FLDCNT	DWord	100	None	Number of fields filled
Events_003	EVENT_003	String	100	None	Semi-colon delimited li
Events_003_001	EVENT_003_001	String	100	None	Field 1 data for event 3
Events_003_002	EVENT_003_002	String	100	None	Field 2 data for event 3
Events_003_003	EVENT_003_003	String	100	None	Field 3 data for event 3
Events_003_004	EVENT_003_004	String	100	None	Field 4 data for event 3
Events_003_005	EVENT_003_005	String	100	None	Field 5 data for event 3
Events_003_FieldCnt	EVENT_003_FLDCNT	DWord	100	None	Number of fields filled
Events_Count	EVENT_COUNT	DWord	100	None	Number of trap events

Trap messages may carry several variables or components of information. These variables are placed into the Event field tags. When a new trap is received, the entire message is placed into address EVENTS_001 as a semicolon-delimited string. Each component is broken into EVENTS_001_001, EVENTS_001_002, EVENTS_001_003 and so forth. The EVENTS_001_FLDCNT address contains the number of fields found in the trap message.

Some SNMPv2c and SNMPV3 devices support Inform-requests. Informs are a more reliable way for SNMP devices to send unsolicited messages to an SNMP manager. When the SNMP Driver receives an Inform, a response message containing the OIDs contained within the Inform is returned to the device. This provides a way for SNMP managers to verify the receipt of these unsolicited messages. For SNMPV3, this means that the SNMP device is required to authenticate and encrypt the Inform (which may require additional device configuration). For more information on SNMP Inform and/or SNMPV3 configuration, refer the device manufacturer's manual.

Note: The address EVENTS_COUNT increments with each incoming trap message. To reset the counter, users can write a new value. To reset the EVENTS_COUNT address from client applications, users can write a zero.

Auto-Created Trap Tags

If traps are enabled, a set of trap tags will be created for the trap OIDs present in the device profile. For the Ethernet Switch and Other Device profiles, these will be coldStart, warmStart, linkUp, and linkDown. A base tag is created for each of these, along with 20 table entries representing the first 20 rows of the virtual trap message table. For more information on table entries, refer to **Trap Tags**.

Note: Trap OIDs defined in any included MIB modules will also have a similar set of trap tags created.

Message Descriptions

The following categories of messages may be generated. Click on the link for a list of messages.

Address Validation
Runtime Messages
SNMP Agent Messages
XML Messages
Communications Messages
Authentication Messages
MIB Parser Messages
Security Related Messages

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 read only.
The remote device reports that the requested name <OID> does not exist on <device name>.

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

Error Type:

Warning

Possible Cause:

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

Solution:

Verify 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:
An invalid tag address has been specified in a dynamic request.

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

Device address <address> is read only.

Error Type:
Warning

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

Solution:
Change the access mode in the client application.

The remote device reports that the requested name <OID> does not exist on <device name>.

Error Type:
Warning

Possible Cause:
An object in the project is not available in the physical device. It has been deactivated.

Solution:

1. Remove the object from the project.
2. It is possible that the process the object is referring to is disabled in the physical device. Make sure it is enabled. The error should not occur in the next request.

Runtime Messages

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

<Channel name>.<device name>: unable to open a SNMP session to host <host> on port <port>, using protocol <protocol>.
<Channel name>.<device name>: Unable to establish a trap listener on port <port>, using protocol <protocol>. No trap events will be received.
Access to address <address> on <channel name>.<device name> is not permitted.
Address <address> on <channel name>.<device name> is not writable.
Address <address> on <channel name>.<device name> is unavailable.

Device <device name> does not support the necessary information required to perform network analysis. Network Analyst tags will be disabled for this device.
Device <device name> does not support the number of ports currently configured in this application. Network Analyst tags will be disabled for this device.
Device <device name> is not responding.
Device discovery has exceeded <max devices> maximum allowed devices.
High capacity counters for network analysis are not available for device <device name>. Attempting to use low capacity counters.
The remote device reports that the requested name <name> does not exist on <channel name>.<device name>.
The response message for the current transaction on <channel name>.<device name> would have been too large, and has been discarded by the remote device.
Unable to bind trap socket on binding address <address>, port <port> and protocol <protocol> for device <device>.
Unable to bind trap socket on binding address <IP address>, port <port number> and protocol <protocol> for device <device name>.
Unable to create communications thread on trap socket for binding address <IP address>, port <port number> and protocol <protocol> for device <device name>.
Unable to create listener on trap socket for binding address <IP address>, port <port number> and protocol <protocol> for device <device name>.
Unable to create trap socket on binding address <IP address>, port <port number> and protocol <protocol> for device <device name>.
Unable to load authentication and privacy passphrases for device <device name>. Please specify an authentication and privacy passphrase in the SNMP V3 Security property group of Device Properties.
Unable to load authentication passphrase for device <device name>. Please specify an authentication passphrase in the SNMP V3 Security property group of Device Properties.
Unable to load username for device <device name>. Please specify a username in the SNMP V3 Security property group of Device Properties.
Unable to resolve host address <IP address> on device <device name> for trap processing.
Unable to send transaction: <reason>.

<Channel name>.<device name>: unable to open a SNMP session to host <host> on port <port>, using protocol <protocol>.

Error Type:

Warning

Possible Cause:

1. The device ID contains a bad IP address or hostname.
2. The port specified is incorrect for the remote device.
3. The protocol specified is incorrect for the remote device.

Solution:

Check the Device Properties and ensure that the device ID and port and protocol are correct.

See Also:
Communication Properties

<Channel name>.<device name>: Unable to establish a trap listener on port <port>, using protocol <protocol>. No trap events will be received.

Error Type:
Warning

Possible Cause:
The specified port is unavailable for listening.

Solution:

1. Check for other applications listening for IP traffic on the chosen port.
2. Ensure that the Windows SNMP Trap Service is not running on the OPC server host machine.

Access to address <address> on <channel name>.<device name> is not permitted.

Error Type:
Warning

Possible Cause:
The remote SNMP does not permit access to the requested SNMP OID.

Solution:
Verify that the community name is correct and permits access to the address.

See Also:
About SNMP Addresses
Communication Properties

Address <address> on <channel name>.<device name> is not writable.

Error Type:
Warning

Possible Cause:
The configured community name does not have write privileges for this address.

Solution:
Verify that the community name is correct and permits write access to the address.

See Also:
About SNMP Addresses
Communication Properties

Address <address> on <channel name>.<device name> is unavailable.

Error Type:
Warning

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

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

Device <device name> does not support the necessary information required to perform network analysis. Network Analyst tags will be disabled for this device.

Error Type:
Warning

Possible Cause:
Although Network Analyst functions were selected, the device does not support the OIDs required by this function.

Solution:
Disable the device's Network Analyst functions.

Device <device name> does not support the number of ports currently configured in this application. Network Analyst tags will be disabled for this device.

Error Type:
Warning

Possible Cause:
The number of ports specified in the Network Analyst settings exceeds the number of ports available in the device.

Solution:
Verify the number of ports in the device. Then, edit the Network Analyst property group in Device properties to regenerate the project tags with the correct number of ports specified.

Device <device name> is not responding.

Error Type:
Serious

Possible Cause:

1. The Ethernet connection between the device and the Host PC is broken.
2. The named device may have been assigned an incorrect IP address.
3. The requested address is not available in the device.
4. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.

Solution:

1. Verify the cabling between the PC and the device network.
2. Verify that the IP address given to the named device matches that of the actual device.
3. Verify that the device supports the requested address.
4. Increase the Request Timeout setting so that the entire response can be handled.

Device Discovery has exceeded <max devices> maximum allowed devices.

Error Type:
Warning

Possible Cause:
The Device Discovery has exceeded the maximum number of allowed devices.

Solution:
Limit the discovery range and then try again.

High-capacity counters for network analysis are not available for device <device name>. Attempting to use low capacity counters.

Error Type:
Warning

Possible Cause:
The device does not support the 64-bit counters that the project is created with. The server is attempting to use low capacity 32-bit counters instead.

Solution:

1. Verify that the supplied MIB is correct.
2. Edit the MIB to reflect the correct counter type and then import again.

The remote device reports that the requested name <name> does not exist on <channel name>.<device name>.

Error Type:
Warning

Possible Cause:

The remote SNMP Agent has not implemented the requested SNMP OID.

Solution:
Remove the tag referring to the address.

See Also:
About SNMP Addresses

The response message for the current transaction on <channel name>.<device name> would have been too large, and has been discarded by the remote device.

Error Type:
Warning

Possible Cause:
The remote SNMP Agent was unable to fit the requested data into a single SNMP reply.

Solution:
Reduce the number of items per request. For older SNMP V1 Agents, this may need to be as low as 1.

See Also:
Communication Properties

Unable to bind trap socket on binding address <address>, port <port>, and protocol <protocol> for device <device>.

Error Type:
Fatal

Possible Cause:
More than one channel has been assigned the same IP address, with SNMP Trap Support enabled.

Solution:

1. The trap socket is only allowed to bind to one IP address: ensure that the correct IP address is the one assigned to the PC.
2. Ensure that SNMP Trap Support is not enabled on more than one channel using the same address.

Unable to bind trap socket on binding address <IP address>, port <port number> and protocol <protocol> for device <device name>.

Error Type:
Warning

Possible Cause:
Unable to bind the trap socket to the specified network card.

Solution:
Some other application has already bound a socket to the binding address/port pair.

Unable to create communications thread on trap socket for binding address <IP address>, port <port number>, and protocol <protocol> for device <device name>.

Error Type:
Warning

Possible Cause:
A thread that handles unsolicited communications for the specified socket/port and protocol could not be created.

Solution:
1. Check the operating system's event log for resource errors.
2. Check the number of process threads being used by the OPC server. Some older operating systems will limit the number of process threads to 1024 per process. For newer operating systems, this is limited by available memory.

Unable to create listener on trap socket for binding address <IP address>, port <port number> and protocol <protocol> for device <device name>.

Error Type:
Warning

Possible Cause:
An incoming connection request (TCP/IP only) could not be listened for.

Solution:
1. Verify that there is not a resource conflict.
2. Verify that the remote device is able to establish a connection to the trap socket.

Unable to create trap socket on binding address <IP address>, port <port number> and protocol <protocol> for device <device name>.

Error Type:
Warning

Possible Cause:
The server was unable to create the specified trap socket on the bound network card.

Solution:

1. Check for other applications listening for IP traffic on the chosen port and IP address.
2. Ensure that the Windows SNMP Trap Service is not running on the OPC server host machine.

Unable to load authentication and privacy passphrases for device <device name>. Please specify an authentication and privacy passphrase in the SNMP V3 Security property group of Device Properties.

Error Type:
Warning

Possible Cause:
The authentication and privacy passphrases failed to load from the XML project file.

Solution:
Specify both an authentication and privacy passphrase in the **SNMP V3 Security** property group located in **Device Properties**.

See Also:
SNMP V3 Security

Unable to load authentication passphrase for device <device name>. Please specify an authentication passphrase in the SNMP V3 Security property group of Device Properties.

Error Type:
Warning

Possible Cause:
The authentication passphrase failed to load from the XML project.

Solution:
Specify an authentication passphrase in the **SNMP V3 Security** property group located in **Device Properties**.

See Also:
SNMP V3 Security

Unable to load username for device <device name>. Please specify a username in the SNMP V3 Security property group of Device Properties.

Error Type:
Warning

Possible Cause:

1. An OPF or XML project file was saved without a username specified in the **SNMP V3 Security** property group located in **Device Properties**.

2. An XML project file was manually edited to remove the username.

Solution:
Specify a username in SNMP V3 Security.

See Also:
SNMP V3 Security

Unable to resolve host address <IP address> on device <device name> for trap processing.

Error Type:
Warning

Possible Cause:
The server's Hostname Resolver is unable to resolve the hostname string for the device to an IP address.

Solution:
1. Verify the spelling of the hostname.
2. If the connection was working before, verify the Cache Lifetime settings in the Server Runtime Hostname Resolution settings.

Unable to send transaction: <reason>.
The following error/warning messages concern transaction transmission to the remote device.

Reason	Possible Cause	Solution
Generic error	The protocol subsystem has reported a non-specific error.	N/A
Invalid local port	The local port may be restricted or in use.	Select an available port.
Unknown host	The remote hostname did not resolve.	Check the device ID.
Unknown session	The SNMP session terminated unexpectedly.	Disconnect and reconnect the client to refresh the session.
Too long	The SNMP message was too long.	Reduce the number of items per request.
No socket	The local port may be restricted or in use.	Select an available port.
Failure in send to	Unable to send the transaction.	Check the device ID and port.
Bad community specified	Bad community specified.	Check the community name.
Authentication failure	Incorrect password, community or key.	Check the community name.
MIB not initialized	MIB module file is not installed.	Check that the MIB module file is installed.

SNMP Agent Error Messages

The following errors reflect problems with the data received from the remote SNMP Agent. They are advisory and no local action is indicated.

Data for address <address> on <channel name>.<device name> has an inconsistent value.
Data for address <address> on <channel name>.<device name> has the wrong encoding.
Data for address <address> on <channel name>.<device name> has the wrong length.
Data for address <address> on <channel name>.<device name> has the wrong value.

Data for address <address> on <channel name>.<device name> has an inconsistent value.

Error Type:

Advisory

Possible Cause:

Problem with the data received from the remote SNMP Agent. Data for address has an inconsistent value.

Solution:

Check configuration of the remote SNMP Agent.

Data for address <address>on <channel name>. <device name> has the wrong encoding.

Error Type:

Advisory

Possible Cause:

Problem with the data received from the remote SNMP Agent. Data for address has the wrong encoding.

Solution:

Check configuration of the remote SNMP Agent.

Data for address <address>on <channel name>.<device name> has the wrong length.

Error Type:

Advisory

Possible Cause:

Problem with the data received from the remote SNMP Agent. Data for address has the wrong length.

Solution:

Check configuration of the remote SNMP Agent.

Data for address <address>on <channel name>. <device name> has the wrong value.

Error Type:
Advisory

Possible Cause:
Problem with the data received from the remote SNMP Agent. Data for address has the wrong value.

Solution:
Check configuration of the remote SNMP Agent.

XML Messages

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

Invalid XML document [Reason: The excluded port list is invalid for device <device name>].
Invalid XML document [Reason: Port Status 0 limit must be less than port Status 1 limit for device <device name>].

Invalid XML document [Reason: The excluded port list is invalid for device <device name>].

Error Type:
Fatal

Possible Cause:
The XML project file was edited such that the ExcludePorts element for the device is invalid.

Solution:
Search the XML project file for the ExcludePorts element of the device and make sure that the string value complies with the following guidelines:

1. Port numbers are in ascending order.
2. Port numbers are separated by a comma. For example, 1,3,10.
3. A hyphen may be used for consecutive ports to indicate a range. For example, 2, 5-7, 15-18.
4. Port numbers are in the range 1-'Number of Ports' setting.

See Also:
Network Analyst Tags

Invalid XML document [Reason: Port Status 0 limit must be less than Port Status 1 limit for device <device name>].

Error Type:
Fatal

Possible Cause:

The XML project file was edited such that the PortStatusLimit0 element for the device has an integer value that is greater than or equal to the integer value of the corresponding PortStatusLimit1 element.

Solution:
Search the XML project file for the PortStatusLimit0 element of the device and make sure that the integer value is less than the integer value of the corresponding PortStatusLimit1 element.

See Also:
Network Analyst Tags

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

Unable to bind to adapter: <adapter address>. Connect failed. Winsock Err # n.
Winsock initialization failed (OS error = n).
Winsock shutdown failed (OS error = n).
Winsock V1.1 or higher must be installed to use the SNMP device driver.

Unable to bind to adapter: <adapter address>. Connect failed. Winsock Err # n.

Error Type:
Fatal

Possible Cause:
The driver was unable to bind to the specified network adapter, which is necessary for communications with the device. This may have occurred because of the following:

1. The adapter is disabled or no longer exists
2. There was a network system failure (such as Winsock or network adapter failure).
3. There are no more available ports.

Solution:

1. Check the Network Adapter list in the communications server application for network adapters available on the system. If <adapter> is not in this list, steps should be taken to make it available to the system. This includes verifying that the network connection is enabled and connected in the PC's Network Connections.
2. Determine how many channels are using the same <adapter> in the communications server application. Reduce this number so that only one channel is referencing <adapter>. If the error still occurs, check to see if other applications are using that adapter and then shut down those applications.

Winsock initialization failed (OS Error = n).

Error Type:

Fatal

OS Error	Indication	Possible Solution
10091	Indicates that the underlying network subsystem is not ready for network communication.	Wait a few seconds and restart the driver.
10067	Limit on the number of tasks supported by the Windows Sockets implementation has been reached.	Close one or more applications that may be using Winsock and restart the driver.

Winsock shut down failed (OS Error = n).

Error Type:
Fatal

Possible Cause:
The network was unable to disable or shut down a network connection.

Solution:
N/A

Winsock V1.1 or higher must be installed to use the SNMP device driver.

Error Type:
Fatal

Possible Cause:
The version number of the Winsock DLL found on the system is less than 1.1.

Solution:
Upgrade Winsock to version 1.1 or higher.

Authentication Messages

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

The authentication passphrase fields do not match. Please retype the passphrase identically in both fields.
The privacy passphrase fields do not match. Please retype the passphrase identically in both fields.

The authentication passphrase fields do not match. Please retype the passphrase identically in both fields.

Error Type:
Information

Possible Cause:

The authentication passphrase entered in the server does not match the passphrase entered into the remote device.

Solution:
Enter the correct passphrase.

The privacy passphrase fields do not match. Please retype the passphrase identically in both fields.

Error Type:
Information

Possible Cause:
The privacy passphrase entered in the server does not match the passphrase entered into the remote device.

Solution:
Enter the correct passphrase.

MIB Parser Messages

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

Cannot redefine macro name.
Cannot redefine primitive type.
Close IMPORTS statement with a ';'.
Could not add object: <object name>; parent object: <parent object name> undefined.
Could not find module: <module name> to import.
Could not obtain MIB module information.
DEFINITIONS must directly follow MIB module name.
End one module definition before beginning another.
Failed to open file: <file path>.
Invalid assignment value.
Invalid DESCRIPTION value.
Invalid ENTERPRISE value.
Invalid MAX-ACCESS value.
Invalid module name.
Invalid NOTIFICATION-TYPE clause.
Invalid object assignment.
Invalid OBJECT-IDENTITY clause.
Invalid OBJECT-TYPE clause.
Invalid OBJECTS value.
Invalid octet or bit string.
Invalid parent object name.
Invalid STATUS value.
Invalid SYNTAX value.

Invalid TRAP-TYPE assignment.
Invalid TRAP-TYPE clause.
Open bracket not closed.
Open parenthesis not closed.
Sub-identifier out of range: 0 to 4294967295.
Syntax error.
Undefined identifier: <identifier name>.

Cannot redefine macro name.

Error Type:
Warning

Possible Cause:
An object's name is the same as a macro's name.

Solution:
Change the object's name, in addition to any references made to the object. Then, re-import the MIB file.

Cannot redefine primitive type.

Error Type:
Warning

Possible Cause:
An object's name is the same as a primitive data type.

Solution:
Change the object's name, in addition to any references made to the object. Then, re-import the MIB file.

Close IMPORTS statement with a ';'.

Error Type:
Error

Possible Cause:
The semicolon was excluded from the end of the MIB's IMPORTS section.

Solution:
Correct the error and then re-import the MIB file.

Could not add object: <object name>; parent object: <parent object name> undefined.

Error Type:
Warning

Possible Cause:
The parent object referenced in an object's definition is either misspelled or undefined.

Solution:
Correct the error and then re-import the MIB file.

Could not find module: <module name> to import.

Error Type:
Warning

Possible Cause:
The module referenced in the MIB's IMPORTS section is not in the same directory as the module being imported.

Solution:
Add the MIB file to the same directory as the dependent MIB file, and then re-import.

Could not obtain MIB module information.

Error Type:
Error

Possible Cause:

1. The selected file is not a MIB file.
2. The MIB file is not defined correctly.

Solution:
Verify that the MIB file begins with "<module name> DEFINITIONS". If it does not, correct the error and then re-import the MIB file.

DEFINITIONS must directly follow MIB module name.

Error Type:
Error

Possible Cause:
The token preceding DEFINITIONS is not a valid identifier.

Solution:
Correct the error and then re-import the MIB file.

End one module definition before beginning another.

Error Type:
Warning

Possible Cause:
The MIB file defined a new module before the 'END' token in the previous module.

Solution:
Signify the end of the previous module with 'END' and then re-import the MIB file.

Failed to open file: <file path>.

Error Type:
Error

Possible Cause:
The driver was not able to load the MIB file, which may be locked by another process.

Solution:
Try to re-import the MIB file.

Invalid assignment value.

Error Type:
Warning

Possible Cause:
The right half of an assignment is not a primitive type, an identifier that resolves to a primitive type, or a TEXTUAL-CONVENTION.

Solution:
Correct the error and then re-import the MIB file.

Invalid DESCRIPTION value.

Error Type:
Warning

Possible Cause:
The object's DESCRIPTION value is not a quoted string.

Solution:
Correct the error and then re-import the MIB file.

Invalid ENTERPRISE value.

Error Type:
Warning

Possible Cause:
The TRAP-TYPE ENTERPRISE value is not an identifier or an OID.

Solution:
Correct the error and then re-import the MIB file.

Invalid MAX-ACCESS value.

Error Type:
Warning

Possible Cause:
The object's ACCESS/MAX-ACCESS value is not valid.

Solution:
Correct the error and then re-import the MIB file.

Invalid module name.

Error Type:
Error

Possible Cause:
A reserved word was used as a module name.

Solution:
Change the module's name, in addition to any references made to the module. Then, re-import the MIB file.

Invalid NOTIFICATION-TYPE clause.

Error Type:
Warning

Possible Cause:
The NOTIFICATION-TYPE clause is either misspelled or undefined.

Solution:
Correct the error and then re-import the MIB file.

Invalid object assignment.

Error Type:
Warning

Possible Cause:
The object's value is not a valid OID.

Solution:
Correct the error and then re-import the MIB file.

Invalid OBJECT-IDENTITY clause.

Error Type:

Warning

Possible Cause:
The OBJECT-IDENTITY clause is either misspelled or undefined.

Solution:
Correct the error and then re-import the MIB file.

Invalid OBJECT-TYPE clause.

Error Type:
Warning

Possible Cause:
The OBJECT-TYPE clause is either misspelled or undefined.

Solution:
Correct the error and then re-import the MIB file.

Invalid OBJECTS value.

Error Type:
Warning

Possible Cause:
The value of an OBJECT or VARIABLE begins does not begin with an open curly brace.

Solution:
Correct the error and then re-import the MIB file.

Invalid octet or bit string.

Error Type:
Error

Possible Cause:

1. A character besides 0-F was included within an octet string.
2. The character 'h' or 'b' was excluded from the end of a string.

Solution:
Correct the error and then re-import the MIB file.

Invalid parent object name.

Error Type:
Warning

Possible Cause:
The parent object referenced in an object's definition is not an identifier.

Solution:
Correct the error and then re-import the MIB file.

Invalid STATUS value.

Error Type:
Warning

Possible Cause:
The object's STATUS value is not valid.

Solution:
Correct the error and then re-import the MIB file.

Invalid SYNTAX value.

Error Type:
Warning

Possible Cause:
The object's SYNTAX is neither a primitive type nor an identifier that resolves to a primitive type.

Solution:
Correct the error and then re-import the MIB file.

Invalid TRAP-TYPE assignment.

Error Type:
Warning

Possible Cause:
The TRAP-TYPE's value is not a number.

Solution:
Correct the error and then re-import the MIB file.

Invalid TRAP-TYPE clause.

Error Type:
Warning

Possible Cause:
The TRAP-TYPE clause is either misspelled or undefined.

Solution:
Correct the error and then re-import the MIB file.

Open bracket not closed.

Error Type:
Error

Possible Cause:
A closing bracket was inadvertently omitted from the selected MIB file.

Solution:
Correct the error and then re-import the MIB file.

Open parenthesis not closed.

Error Type:
Error

Possible Cause:
A closing parenthesis was inadvertently omitted from the selected MIB file.

Solution:
Correct the error and then re-import the MIB file.

Sub-identifier out of range: 0 to 4294967295.

Error Type:
Error

Possible Cause:
An object's sub-identifier is out of the valid range of 0 to 4294967295.

Solution:
Correct the error and then re-import the MIB file.

Syntax Error.

Error Type:
Warning

Possible Cause:
An unexpected token was encountered during parsing of the MIB file.

Solution:
Correct the error and then re-import the MIB file.

Undefined identifier: <identifier name>.

Error Type:
Warning

Possible Cause:
An identifier referenced in an object's SYNTAX clause (or as the right half of an assignment) is undefined.

Solution:
Correct the error and then re-import the MIB file.

Security Related Messages

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

<channel name>.<device name> reports a decryption error. Check the privacy passphrase.
<channel name>.<device name> reports the authentication digest is incorrect. Check the authentication passphrase.
<channel name>.<device name> reports the request was not within the time window.
<channel name>.<device name> reports the specified security level is not supported.
<channel name>.<device name> reports the specified user is unknown.
<channel name>.<device name> responded to a request with a Report-PDU containing no valid data.

<channel name>.<device name> reports a decryption error. Check the privacy passphrase.

Error Type:
Warning

Possible Cause:
The SNMP device was unable to decrypt the SNMP V3 Read/Write request because the encryption passphrase and/or authentication styles do not match.

Solution:
Verify that the encryption passphrases and authentication styles set in the SNMP device configuration match those specified in the SNMP Driver Device Properties.

<channel name>.<device name> reports the authentication digest is incorrect. Check the authentication passphrase.

Error Type:
Warning

Possible Cause:
The authentication passphrase and/or authentication style does not match the authentication passphrase and/or authentication style specified in the SNMP device configuration.

Solution:

Verify that the authentication passphrase and authentication style set in the SNMP device configuration matches those specified in the SNMP Driver Device Properties.

<Channel name>.<device name> reports the request was not within the time window.

Error Type:
Warning

Possible Cause:
The device rejected the SNMP Read/Write request from the driver due to one of the following reasons:

1. The message was not received within 150 seconds of sending.
2. The SNMP Driver time properties are not synchronized with the SNMP device.

Solution:
In most cases, the SNMP Driver will synchronize the SNMP time properties with the device, and then communicate with the device successfully.

<channel name>.<device name> reports the specified security level is not supported.

Error Type:
Warning

Possible Cause:
The device does not support the specified SNMP security level.

Solution:
Verify that the security level set in the SNMP device matches the security level specified in the SNMP Driver Device Properties.

See Also:
SNMP V3 Security

<channel name>.<device name> reports the specified user is unknown.

Error Type:
Warning

Possible Cause:
The username specified in the SNMP Driver does not match the username configured in the SNMP device.

Solution:
Verify that the username set in the SNMP device configuration matches the username specified in the SNMP Driver Device Properties.

<channel name>.<device name> responded to a request with a Report-PDU containing no valid data.

Error Type:

Warning

Possible Cause:

The SNMP device/agent has responded with a report PDU that does not contain an OID (and is not supported by the driver).

Solution:

For more information on the report PDU, refer to the device manual.

Index

<

<Channel name>.<device name> reports a decryption error. Check the privacy passphrase. 59
<Channel name>.<device name> reports the authentication digest is incorrect. Check the authentication passphrase. 59
<Channel name>.<device name> reports the request was not within the time window. 60
<Channel name>.<device name> reports the specified security level is not supported. 60
<Channel name>.<device name> reports the specified user is unknown. 60
<Channel name>.<device name> responded to a request with a Report-PDU containing no valid data. 60
<Channel name>.<device name>: Unable to establish a trap listener on port <port>, using protocol <protocol>.No trap events will be received. 40
<Channel name>.<device name>: unable toopen a SNMP session to host <host> on port <port>_ using protocol <protocol>. 39

A

About MIB Modules 31
About Network Analyst Tags 32
About SNMP Addresses 29
Access to address <address> on <channel name>.<device name> is not permitted. 40
Address <address> is out of range for the specified device or register. 37
Address <address> on <channel name>.<device name> is unavailable. 41
Address <address> on <channel name>.<devicename> is not writable. 40
Address Descriptions 29
Address Validation 37
Allow Sub Groups 19
Attempts Before Timeout 16
Authentication 21
Authentication Messages 50
AuthNoPriv 21
AuthPriv 21
Auto Created Trap Tags 36

C

Cannot redefine macro name. 52
Cannot redefine primitive type. 52

Channel Assignment 14
Channel Setup 7
Close IMPORTS statement with a ';'. 52
Communication Properties 19
Communications Messages 49
Communications Timeouts 15-16
Community Credentials 31
Connect Timeout 16
Context 21
Could not add object: <object name>; parent object: <parent object name> undefined. 52
Could not find module: <module name> to import. 53
Could not obtain MIB module information. 53
Create 19

D

Data Collection 14
Data for address <address> on <channel name>.<device name> has an inconsistent value. 47
Data for address <address>on <channel name>.<device name> has the wrong encoding. 47
Data for address <address>on <channel name>.<device name> has the wrong length. 47
Data for address <address>on <channel name>.<device name> has the wrong value. 48
Data Type <type> is not valid for device address<address>. 37
Data Types Description 25
DEFINITIONS must directly follow MIB module name. 53
Delete 18
Delta Time 27
Demote on Failure 17
Demotion Period 17
Description 14
Device <device name> does not support the necessary information required to perform network analysis. Network Analyst tags will be disabled for this device. 41
Device <device name> does not support the number of ports currently configured in this application. Network Analyst tags will be disabled for this device. 41
Device <device name> is not responding. 41
Device address <address> contains a syntax error. 38
Device address <address> is read only. 38
Device Discovery has exceeded <max devices> maximum allowed devices. 42
Device Properties — Auto-Demotion 16
Device Properties — General 13

Device Properties — Tag Generation 17
Device Setup 13
Discard Requests when Demoted 17
Do Not Scan, Demand Poll Only 15
Driver 14

E

End one module definition before beginning another. 53
Enterprise 30

F

Failed to open file: <file path>. 54

G

Generate 18

H

Help Contents 6
High-capacity counters for network analysis are not available for device <device name>. Attempting to use low capacity counters. 42
Historical 29
Historical Data Attributes 27

I

ID 14
Initial Updates from Cache 15
Instances 30
Inter-Request Delay 16
Invalid assignment value. 54
Invalid DESCRIPTION value. 54
Invalid ENTERPRISE value. 54
Invalid MAX-ACCESS value. 55
Invalid module name. 55

Invalid NOTIFICATION-TYPE clause. 55
Invalid OBJECT-IDENTITY clause. 55
Invalid OBJECT-TYPE clause. 56
Invalid object assignment. 55
Invalid OBJECTS value. 56
Invalid octet or bit string. 56
Invalid parent object name. 56
Invalid STATUS value. 57
Invalid SYNTAX value. 57
Invalid TRAP-TYPE assignment. 57
Invalid TRAP-TYPE clause. 57
Invalid XML document [Reason: Port Status 0 limit must be less than port Status 1 limit for device <device name>]. 48
Invalid XML document [Reason: The excluded port list is invalid for device <device name>]. 48

M

Message Descriptions 37
MIB Import Settings 22
MIB Parser Messages 51
Model 14
Moving Average 27

N

Name 13
Network Analyst Tags 24
NoAuthNoPriv 21

O

On Device Startup 18
On Duplicate Tag 18
On Property Change 18
Open bracket not closed. 58
Open parenthesis not closed. 58
Overview 6
Overwrite 18

P

Parent Group 19
Passphrase 22
Previous Value 27
Privacy 21

R

Redundancy 24
Request All Data at Scan Rate 15
Request Data No Faster than Scan Rate 15
Request Timeout 16
Respect Client-Specified Scan Rate 15
Respect Tag-Specified Scan Rate 15
Runtime Messages 38

S

Scan Mode 15
Scan Rate Floor 29
Security Related Messages 59
Simulated 14
SNMP Agent Errors 47
SNMPv3 Settings 21
Sub-identifier out of range: 0 to 4294967295. 58
Syntax Error. 58

T

Tag Generation 17
The authentication passphrase fields do not match. Please retype the passphrase identically in both fields. 50
The privacy passphrase fields do not match. Please retype the passphrase identically in both fields. 51
The remote device reports that the requested name <name>does not exist on <channel name>.<device name>. 42
The remote device reports that the requested name <OID> does not exist on <device name>. 38
The response message for the current transaction on <channel name>.<device name> would have been

too large and has been discarded by the remote device. 43

Timeouts to Demote 17

Trap Event Queue 35

Trap Tags 32

Trap/Inform Notifications 22

U

Unable to bind to adapter: <adapter address>. Connect failed. Winsock Err # n. 49

Unable to bind trap socket on binding address <address>,port <port>, and protocol <protocol> for device <device>. 43

Unable to bind trap socket on binding address <IP address>,port <port number> and protocol <protocol> for device <device name>. 43

Unable to create communications thread on trap socket for binding address <IP address>,port <port number>, and protocol <protocol> for device <device name>. 44

Unable to create listener on trap socket for binding address <IP address>,port <port number> and protocol <protocol> for device <device name>. 44

Unable to create trap socket on binding address <IP address>,port <port number> and protocol <protocol> for device <device name>. 44

Unable to load authentication and privacy passphrases for device <device name>. Please specify an authentication and privacy passphrase in the SNMP v3 Security property group of Device Properties. 45

Unable to load authentication passphrase for device <device name>. Please specify an authentication passphrase in the SNMP v3 Security proeprty group of Device Properties. 45

Unable to load username for device <device name>. Please specify a username in the SNMP v3 Security property group of Device Properties. 45

Unable to resolve host address <IP address> on device <device name> for trap processing. 46

Unable to send transaction: <reason>. 46

Undefined identifier: <identifier name>. 59

Unsolicited 29

W

Winsock initialization failed (OS Error = n). 49

Winsock shut down failed (OS Error = n). 50

Winsock V1.1 or higher must be installed to use the SNMP device driver. 50

X

XML Messages 48