SNMP Communication Driver

Driver for TCP/IP Ethernet Communication
with SNMP Protocol

Contents

Introduction

The SNMP driver enables communication between the Studio system and remote devices using the SNMP protocol, according to the specifications discussed in this document.

This document will help you to select, configure and execute the SNMP driver, and it is organized as follows:

- **Introduction**: This section, which provides an overview of the document.
- **General Information**: Identifies all of the hardware and software components required to implement communication between the Studio system and the target device.
- **Selecting the Driver**: Explains how to select the SNMP driver in the Studio system.
- **Configuring the Device**: Describes how the target device must be configured to receive communication from the SNMP driver.
- **Configuring the Driver**: Explains how to configure the SNMP driver in the Studio system, including how to associate database tags with device OIDs.
- **Executing the Driver**: Explains how to execute the SNMP driver during application runtime.
- **Troubleshooting**: Lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.
- **Revision History**: Provides a log of all changes made to the driver and this documentation.

✍ **Notes:**

- This document assumes that you have read the “Development Environment” chapter in Studio’s *Technical Reference Manual*.
- This document also assumes that you are familiar with the Microsoft Windows environment. If you are not familiar with Windows, then we suggest using the **Help** feature as you work through this guide.

General Information

This chapter identifies all of the hardware and software components required to implement communication between the SNMP driver in Studio and a remote device using the SNMP protocol.

The information is organized into the following sections:

- Device Specifications
- Network Specifications
- Driver Characteristics
- Conformance Testing

Device Specifications

To establish communication, your target device must meet the following specifications:

Manufacturer: any

Compatible Equipment: SNMP devices

Device Programming Software: not required

For a description of the device(s) used to test driver conformance, see “Conformance Testing”.

Network Specifications

To establish communication, your device network must meet the following specifications:

- **Device Communication Port**: 161
- **Physical Protocol**: Ethernet
- **Logic Protocol**: SNMP Protocol
- **Device Runtime Software**: None
- **Specific PC Board**: Ethernet port
- **Cable Wiring Scheme:** Regular Ethernet cable

Driver Characteristics

The SNMP driver package consists of the following files, which are automatically installed in the **/DRV** subdirectory of Studio:

- **SNMP.INI:** Internal driver file. *You must not modify this file*.
- **SNMP.MSG:** Internal driver file containing messages for each error code. *You must not modify this file*.
- **SNMP.PDF:** This document, which provides detailed information about the SNMP driver.
- **SNMP.DLL:** Compiled driver

You can use the SNMP driver on the following operating systems:

- Windows 7
- Windows 8
- Windows 10
- Windows Server 2008
- Windows Server 2012

For a description of the operating systems used to test driver conformance, see "Conformance Testing" below.

Conformance Testing

The following hardware/software configuration was used to test driver conformance:

- **Driver Configuration**:
  - **Protocol**: CIP over Ethernet TCP/IP

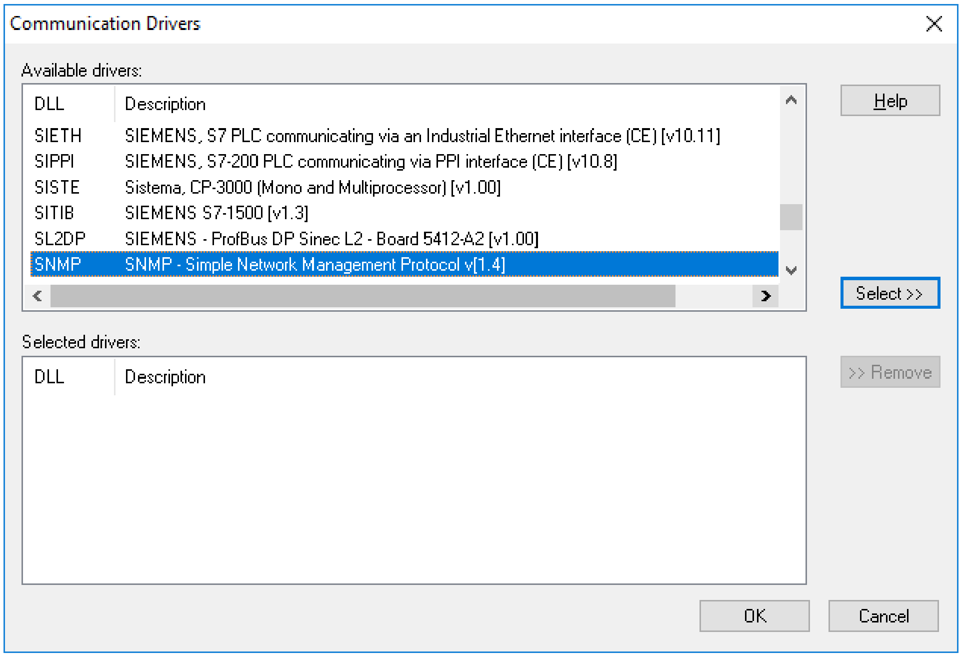
  Cable: Regular Ethernet cables

Driver Version	Studio Version	Operating System (development)	Operating System (target)	Equipment
1.6	8.1 SP1 + P1	Windows 10	▪ Windows 8 ▪ Windows 10	• SNMP version 1 conformance tested on ○ Schneider M340 PLC, ○ Siemens S7-315-2DP, Allen Bradley ○ 1756-L75 ControlLogix 5575 Controller + 1756EN2T Module • SNMP version 2 conformance tested on ○ Net-SNMP extended agent version 5.7.3 ○ Windows SNMP service • SNMP version 3 conformance tested on ○ iReasoning agent version 4.2 • Cable: regular Ethernet cable

Selecting the Driver

When you install Studio, all of the communication drivers are automatically installed in the **\DRV** subdirectory but they remain dormant until manually selected for specific applications. To select the SNMP driver for your Studio application:

1. From the main menu bar, select **Insert** → **Driver** to open the *Communication Drivers* dialog.
2. Select the **SNMP** driver from the *Available Drivers* list, and then click the **Select** button.



Communication Drivers

Available drivers:

DLL	Description
SIETH	SIEMENS, S7 PLC communicating via an Industrial Ethernet interface (CE) [v10.11]
SIPPI	SIEMENS, S7-200 PLC communicating via PPI interface (CE) [v10.8]
SISTE	Sistema, CP-3000 (Mono and Multiprocessor) [v1.00]
SITIB	SIEMENS S7-1500 [v1.3]
SL2DP	SIEMENS - ProfBus DP Sinec L2 - Board 5412-A2 [v1.00]
SNMP	SNMP - Simple Network Management Protocol v[1.4]

Selected drivers:

DLL	Description

Communication Drivers Dialog

3. When the **SNMP** driver is displayed in the **Selected Drivers** list, click the **OK** button to close the dialog. The driver is added to the *Drivers* folder, in the *Comm* tab of the Workspace.

Note:
It is not necessary to install any other software on your computer to enable communication between Studio and your target device.

Attention:
For safety reasons, you must take special precautions when installing any physical hardware. Please consult the manufacturer’s documentation for specific instructions.

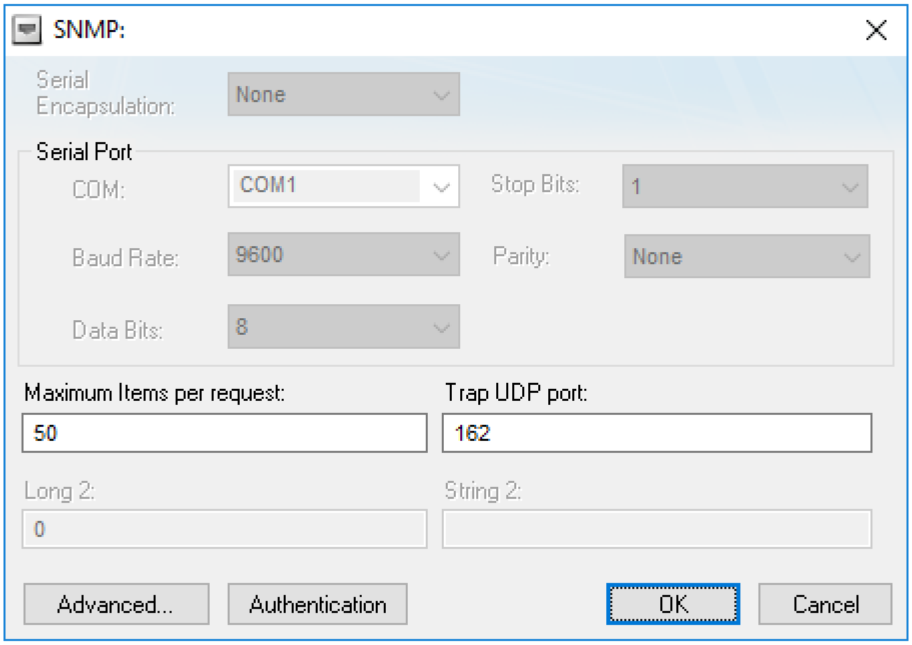
Configuring the Driver

Configuring the Communication Settings

The communication settings are described in detail in the “Communication” chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only SNMP driver-specific settings and procedures will be discussed here. To configure the communication settings for the SNMP driver:

1. In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The SNMP driver is listed here as a subfolder.
2. Right-click on the *SNMP* subfolder and then select the **Settings** option from the pop-up menu:



SNMP:

Serial Encapsulation: None

Serial Port

COM: COM1

Stop Bits: 1

Baud Rate: 9600

Parity: None

Data Bits: 8

Maximum Items per request: 50

Trap UDP port: 162

Long 2: 0

String 2:

SNMP: Communication Settings Dialog

3. In the *Communication Settings* dialog, configure the driver settings to enable communication with your target device. The communication settings and their possible values are described in the following table:

Parameters	Default Values	Valid Values	Description
Maximum items per request	50	**1 to 200**	Maximum number of OIDs included in one SNMP GET command. This limit can vary depending on the capability of the remote devices.
Trap UDP Port	162	**0 to 65535**	IP port number to be open to receive TRAP messages. Zero means no port should be open and TRAPs are not received.

Configuring the Driver Worksheets

Each selected driver includes a Main Driver Sheet and one or more Standard Driver Worksheets. The Main Driver Sheet is used to define tag/register associations and driver parameters that are in effect at all times, regardless of application behavior. In contrast, Standard Driver Worksheets can be inserted to define additional tag/register associations that are triggered by specific application behaviors.

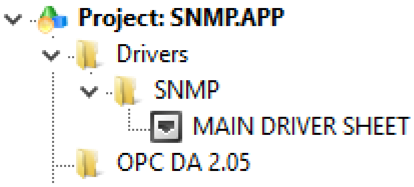
The configuration of these worksheets is described in detail in the “Communication” chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only SNMP driver-specific parameters and procedures will be discussed here.

MAIN DRIVER SHEET

When you select the SNMP driver and add it to your application, Studio automatically inserts the *Main Driver Sheet* in the *SNMP* driver subfolder. To configure the Main Driver Sheet:

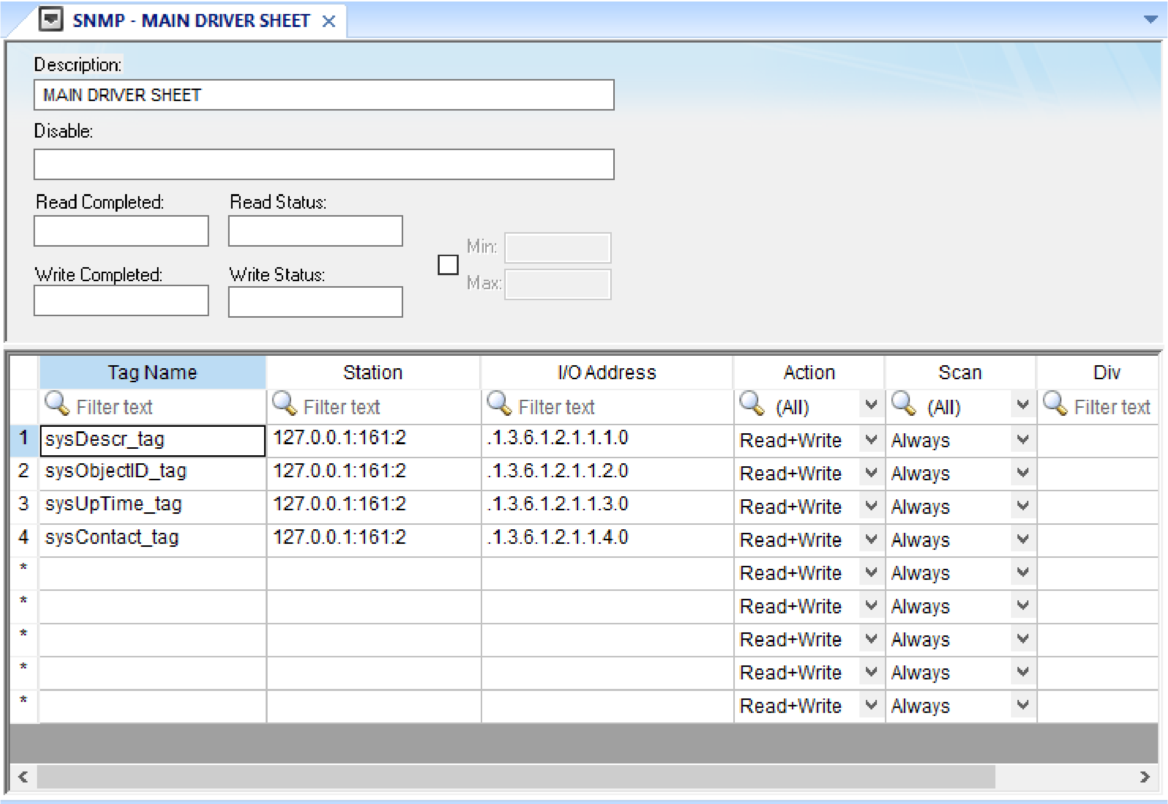
1. Select the *Comm* tab in the *Workspace* pane.
2. Open the *Drivers* folder, and then open the *SNMP* subfolder:



- Project: SNMP.APP
  - Drivers
    - SNMP
      - MAIN DRIVER SHEET
    - OPC DA 2.05

Main Driver Sheet in the SNMP Subfolder

3. Double-click on the **MAIN DRIVER SHEET** icon to open the following worksheet:



Description: MAIN DRIVER SHEET

Disable:

Read Completed:

Read Status:

Write Completed:

Write Status:

□ Min: Max:

	Tag Name	Station	I/O Address	Action	Scan	Div
	Filter text	Filter text	Filter text	(All)	(All)	Filter text
1	sysDescr_tag	127.0.0.1:161:2	.1.3.6.1.2.1.1.1.0	Read+Write	Always	
2	sysObjectID_tag	127.0.0.1:161:2	.1.3.6.1.2.1.1.2.0	Read+Write	Always	
3	sysUpTime_tag	127.0.0.1:161:2	.1.3.6.1.2.1.1.3.0	Read+Write	Always	
4	sysContact_tag	127.0.0.1:161:2	.1.3.6.1.2.1.1.4.0	Read+Write	Always	
*				Read+Write	Always	
*				Read+Write	Always	
*				Read+Write	Always	
*				Read+Write	Always	
*				Read+Write	Always	

Main Driver Sheet

Most of the fields on this sheet are standard for all drivers; see the “Communication” chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station** and **I/O Address** fields use syntax that is specific to the SNMP driver.

4. For each table row (i.e. each tag/register association), configure the **Station** and **I/O Address** fields as follows:
   - **Station** field — Specify the station using the following syntax:

     For SNMP version 1 and 2

     ```
     <IP Address>:<Port>:<Version>:<Read Community>:<Write Community>
     ```

For SNMP version 3

```
<IP Address>:<Port>:<Version>
```

Where:

Parameter	Default Value	Description
IP Address	none	The IP network address of the agent configured.
Port	161	The port number used by the driver for performing GET and SET operations to the OIDs.
Version	1	The version of SNMP used. Possible values are `1`, `2` and `3`.
Read Community	public	The read community name.
Write Community	public	The write and read community name.

`IP Address` is required, but the rest of the parameters are optional. If they are not specified, the default values are used.

The **Station** field cannot be empty. If `IP Address` is not specified, an error will be returned during driver run time.

To change the station during driver run time, type a tag name in curly brackets (e.g. `{MyStation}`). The value of the specified tag will be used, even when the value changes, as long as it uses the syntax described above.

Examples of valid stations:

- `192.186.0.1`
- `192.186.0.1:161`
- `192.186.0.1:161:1`
- `192.186.0.1:166:2:public:private`

- **I/O Address** field

For general SNMP OIDs

Specify the address of the OID (used for SET and GET operations) with the following syntax:

`<Datatype>:<OID>`

OID is the object identifier of the variable (when performing GET and SET operations). This parameter is required.

When performing GET and SET operations, **`Datatype`** is the data type of the OID. This parameter is optional when performing GET operations, and it is required when performing SET operations.

The supported values for **`Datatype`** are shown in the following table:

Value	Description
`INT`	When performing read and write operations with integers.
`OCTETSTRING`	When performing read and write operations with octetstrings.

`OCTETSTRINGHEX`	When performing read and write operations with octetstrings. Sometimes the values are read in hexadecimal, use this header to change to string.
`OID`	When performing read and write operations with OID values.
`IPADDRESS`	When performing read and write operations with IP address datatypes.
`COUNTER32`	When performing read and write operations with 32-bit counter datatypes.
`GAUGE`	When performing read and write operations with gauge datatypes.
`TIMETICKS`	When performing read and write operations with timeticks datatypes.
`OPAQUE`	When performing read and write operations with opaque datatypes.
`COUNTER64`	When performing read and write operations with 64-bit counter datatypes.
`UINT`	When performing read and write operations with unsigned integers datatypes.

For SNMP Traps

Specify the address of the Trap with the following syntax

`TRAP:<OID>/<Variable Binding>`

When using traps, the required header is TRAP. The header is blank when performing SET and GET operations.

When using traps, OID can be either the object identifier of a specific trap (including Enterprise traps) or it can be written as one of the following generic traps:

- ***`COLDSTART`***
- ***`WARMSTART`***
- ***`LINKDOWN`***
- ***`LINKUP`***
- ***`AUTHFAIL`***
- ***`EGPLOSS`***

`Variable Binding` is the variable binding for the particular trap OID that is specified in the OID parameter. This parameter is optional, and it is used only for traps.

✍ **Note:**

If *Variable Binding* is specified, the project tag that is associated with the address will receive the value of the trap. If Variable Binding is not specified, the value of the associated project tag will be incremented each time a trap is received.

For SNMP Tables

Specify the address of the table element (used for SET and GET operations) with the following syntax:

<***Datatype>:<Column OID>[<Row index>]***

Datatype is the data type of the table element. This parameter is optional when performing GET operations, and it is required when performing SET operations.

Column OID is the identifier of the table column where the element sits.

Row index is the table row of the element

Consider the following table as an example:

	Column 1	**Column 2**	**Column 3**	**Column 4**	**Column 5**
	ifIndex	**ifDescr**	**ifType**	**ifMtu**	**ifSpeed**
Row 1	0.0.0.0	lo0	8	32768	0
Row 2	127.0.0.1	lo1	6	1600	1500
Row 3	127.0.1.2	hi0	4	6456	10000000

In the above table, the actual OID's and the corresponding address in the SNMP driver of some table cells are given below:

Row#	**Column#**	**Column OID**	**Element OID**	**Driver address**
1	**1**	.1.2.3.1.2.1	.1.2.3.1.2.1.0.0.0.0	.1.2.3.1.2.1[1]
1	**2**	.1.2.3.1.2.2	.1.2.3.1.2.2.0.0.0.0	.1.2.3.1.2.2[1]
3	**4**	.1.2.3.1.2.1	.1.2.3.1.2.4.127.0.1.2	.1.2.3.1.2.4[3]

In the above case, each cell of the table has the OID of the table along with the column index and the row index value. Here the table OID is '.1.2.3.1.2' and rest of the OID is the corresponding column number and row index value. In the driver worksheet you can use the row number in '[ ]' instead of the row index value as shown above. The SNMP driver internally validates the row number from the item name and translates it with the Index value as required by the SNMP 'Get' and 'Set' command accordingly. The row number always starts with 1. If you enter an invalid row number, the item returns a bad quality.

Examples of valid addresses when performing GET and SET operations:

- `INT:.1.3.6.1.2.1.2.2.1.1.6`
- `OCTETSTRING:.1.3.6.1.2.1.1.1.0`
- `OID:.1.3.6.1.2.1.2.2.1.22.6`
- `IPADDRESS:.1.3.6.1.2.1.4.20.1.2.127.0.0.1`
- `COUNTER32:.1.3.6.1.2.1.2.2.1.11.14`
- `GAUGE:.1.3.6.1.2.1.2.2.1.5.5`
- `TIMETICKS:.1.3.6.1.2.1.1.3.0`
- `OPAQUE:.1.3.6.1.1.1.1.5`
- `COUNTER64:.1.3.6.1.2.1.1.1.6`
- `UINT:.1.3.6.1.3.1.1.4`
- `.1.3.6.1.2.1.1.1.0`

Examples of valid addresses when using traps:

- `TRAP:.1.3.1.1.0`
- `TRAP:COLDSTART`
- `TRAP:LINKUP/.1.3.6.1.2.1.2.2.1.1.19`
- `TRAP:LINKDOWN/.1.3.6.1.2.1.2.2.1.1.19`
- `TRAP:.1.3.6.1.1.1.0/.3.3.3.2`
- `TRAP:LINKUP`
- `TRAP:AUTHFAIL`
- `TRAP:EGPLOSS`
- `TRAP:WARMSTART`
- `TRAP:LINKDOWN`
- `TRAP:COLDSTART`

Examples of valid addresses when performing GET and SET operations on tables:

- `.1.3.6.1.2.1.2.2.1.1[1]`
- `INT:.1.3.6.1.2.1.2.2.1.1[2]`
- `OCTETSTRING: .1.3.6.1.2.1.2.2.1.2[1]`

Note:

The Main Driver Sheet can have up to 32767 rows. If you need to configure more than 32767 communication addresses, then either configure additional Standard Driver Sheets or create additional instances of the driver.

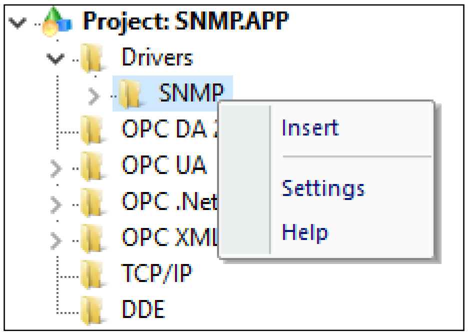
5. Save and close the worksheet.

STANDARD DRIVER WORKSHEET

When you select the SNMP driver and add it to your application, it has only a Main Driver Sheet by default (see previous section). However, you may insert additional Standard Driver Worksheets to define tag/register associations that are triggered by specific application behaviors. Doing this will optimize communication and improve system performance by ensuring that tags/registers are scanned only when necessary – that is, only when the application is performing an action that requires reading or writing to those specific tags/registers.

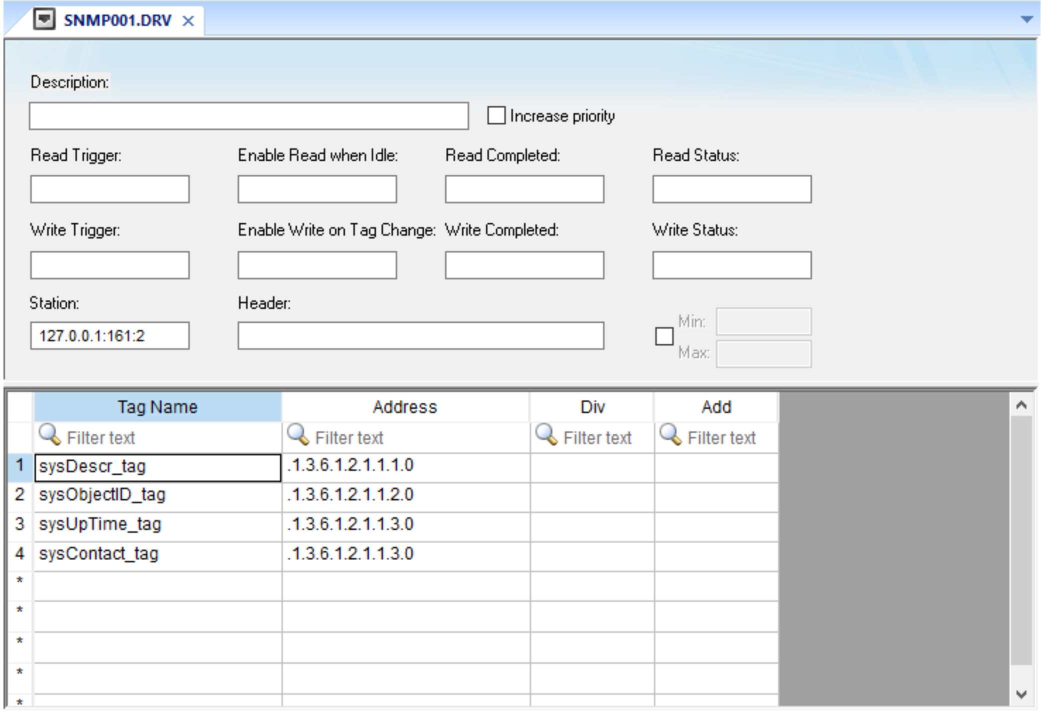
To insert a new Standard Driver Worksheet:

1. In the *Comm* tab, open the *Drivers* folder and locate the *SNMP* subfolder.
2. Right-click the *SNMP* subfolder, and then select **Insert** from the pop-up menu:



Inserting a New Worksheet

A new SNMP driver worksheet is inserted into the SNMP subfolder, and the worksheet is opened for configuration:



Description:
□ Increase priority
Read Trigger:
Enable Read when Idle:
Read Completed:
Read Status:
Write Trigger:
Enable Write on Tag Change:
Write Completed:
Write Status:
Station: 127.0.0.1:161:2
Header:
□ Min: Max:

	Tag Name	Address	Div	Add
	Filter text	Filter text	Filter text	Filter text
1	sysDescr_tag	.1.3.6.1.2.1.1.1.0		
2	sysObjectID_tag	.1.3.6.1.2.1.1.2.0		
3	sysUpTime_tag	.1.3.6.1.2.1.1.3.0		
4	sysContact_tag	.1.3.6.1.2.1.1.3.0		
*				
*				
*				
*				
*				

SNMP Standard Driver Worksheet

Note:

Worksheets are numbered in order of creation, so the first worksheet is **`SNMP001.drv`**.

Most of the fields on this worksheet are standard for all drivers; see the "Communication" chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station**, **Header**, and **Address** fields use syntax that is specific to the SNMP driver.

3. Configure the **Station** and **Header** fields as follows:
   - **Station** field — Specify the station using the following syntax:

     For SNMP version 1 and 2

     ```
     IP Address:Port:Version:Read Community:Write Community
     ```

     For SNMP version 3

     ```
     IP Address:Port:Version
     ```

Where:

Parameter	Default Value	Description
IP Address	none	The IP network address of the agent configured.
Port	`161`	The port number used by the driver for performing GET and SET operations to the OIDs.
Version	`1`	The version of SNMP used. Possible values are `1`, `2` and `3`.
Read Community	`public`	The read community name.
Write Community	`public`	The write and read community name.

IP Address is required, but the rest of the parameters are optional. If they are not specified, the default values are used.

The **Station** field cannot be empty. If IP Address is not specified, an error will be returned during driver run time.

To change the station during driver run time, type a tag name in curly brackets (e.g. `{MyStation}`). The value of the specified tag will be used, even when the value changes, as long as it uses the syntax described above.

Examples of valid stations:

- `192.186.0.1`
- `192.186.0.1:161`
- `192.186.0.1:161:1`
- `192.186.0.1:166:2:public:private`

- **Header** field

  This field is used only to configure the driver to listen to traps from an agent. If that is the case, type `TRAP`. Otherwise, if the driver sheet is configured to perform SET and GET operations on the OIDs, leave this field blank.
  To change the header during project run time, type a tag name in curly brackets (e.g. `{MyHeader}`). The value of the specified tag will be used, even when the value changes, as long as it uses the syntax described above.

4. For each table row (i.e. each tag/register association), configure the **Address** as follows:

For general SNMP OIDs

Specify the address of the OID (used for SET and GET operations) with the following syntax:

<Datatype>:<OID>

OID is the object identifier of the variable (when performing GET and SET operations). This parameter is required.

When performing GET and SET operations, ***Datatype*** is the data type of the OID. This parameter is optional when performing GET operations, and it is required when performing SET operations.

The supported values for ***Datatype*** are shown in the following table:

Value	Description
`INT`	When performing read and write operations with integers.
`OCTETSTRING`	When performing read and write operations with octetstrings.
`OCTETSTRINGHEX`	When performing read and write operations with octetstrings. Sometimes the values are read in hexadecimal, use this header to change to string.
`OID`	When performing read and write operations with OID values.
`IPADDRESS`	When performing read and write operations with IP address datatypes.
`COUNTER32`	When performing read and write operations with 32-bit counter datatypes.
`GAUGE`	When performing read and write operations with gauge datatypes.
`TIMETICKS`	When performing read and write operations with timeticks datatypes.
`OPAQUE`	When performing read and write operations with opaque datatypes.
`COUNTER64`	When performing read and write operations with 64-bit counter datatypes.
`UINT`	When performing read and write operations with unsigned integers datatypes.

For SNMP Traps

Specify the address of the Trap with the following syntax

`TRAP:<OID>/<Variable Binding>`

When using traps, the required header is TRAP. The header is blank when performing SET and GET operations.

When using traps, OID can be either the object identifier of a specific trap (including Enterprise traps) or it can be written as one of the following generic traps:

- ***COLDSTART***
- ***WARMSTART***
- ***LINKDOWN***
- ***LINKUP***
- ***AUTHFAIL***
- ***EGPLOSS***

`Variable Binding` is the variable binding for the particular trap OID that is specified in the OID parameter. This parameter is optional, and it is used only for traps.

✍ **Note:**

If *Variable Binding* is specified, the project tag that is associated with the address will receive the value of the trap. If Variable Binding is not specified, the value of the associated project tag will be incremented each time a trap is received.

For SNMP Tables

Specify the address of the table element (used for SET and GET operations) with the following syntax:

<Datatype>:<Column OID>[<Row index>]

Datatype is the data type of the table element. This parameter is optional when performing GET operations, and it is required when performing SET operations.

Column OID is the identifier of the table column where the element sits.

Row index is the table row of the element

Consider the following table as an example:

	Column 1	**Column 2**	**Column 3**	**Column 4**	**Column 5**
	ifIndex	**ifDescr**	**ifType**	**ifMtu**	**ifSpeed**
Row 1	0.0.0.0	lo0	8	32768	0
Row 2	127.0.0.1	lo1	6	1600	1500
Row 3	127.0.1.2	hi0	4	6456	10000000

In the above table, the actual OID's and the corresponding address in the SNMP driver of some table cells are given below:

Row#	**Column#**	**Column OID**	**Element OID**	**Driver address**
1	**1**	.1.2.3.1.2.1	.1.2.3.1.2.1.0.0.0.0	.1.2.3.1.2.1[1]
1	**2**	.1.2.3.1.2.2	.1.2.3.1.2.2.0.0.0.0	.1.2.3.1.2.2[1]
3	**4**	.1.2.3.1.2.1	.1.2.3.1.2.4.127.0.1.2	.1.2.3.1.2.4[3]

In the above case, each cell of the table has the OID of the table along with the column index and the row index value. Here the table OID is '.1.2.3.1.2' and rest of the OID is the corresponding column number and row index value. In the driver worksheet you can use the row number in '[ ]' instead of the row index value as shown above. The SNMP driver internally validates the row number from the item name and translates it with the Index value as required by the SNMP 'Get' and 'Set' command accordingly. The row number always starts with 1. If you enter an invalid row number, the item returns a bad quality.

For examples of how device registers are specified using Header and Address, see the following table.

Examples of Header and Address fields in Standard Driver Sheet

Header	Address
Blank (when performing GET and SET operations)	`INT:.1.3.6.1.2.1.1.7.0`
	`OCTETSTRING:.1.3.6.1.2.1.1.1.0`
	`OID:.1.3.6.1.2.1.2.2.1.22.6`
	`IPADDRESS:.1.3.6.1.2.1.4.20.1.2.127.0.0.1`
	`COUNTER32:.1.3.6.1.2.1.2.2.1.11.14`
	`GAUGE:.1.3.6.1.2.1.2.2.1.5.5`
	`TIMETICKS:.1.3.6.1.2.1.1.3.0`
	`OPAQUE:.1.3.6.1.2.1.1.1.5`
	`COUNTER64:.1.3.6.1.2.1.1.1.6`
	`UINT:.1.3.6.1.2.1.1.1.4`
	`.1.3.6.1.2.1.1.1.4`
	`.1.3.6.1.2.1.2.2.1.1[1]`
	`INT:.1.3.6.1.2.1.2.2.1.1[2]`
	`OCTETSTRING: .1.3.6.1.2.1.2.2.1.2[1]`
`TRAP`	`COLDSTART`
	`WARMSTART`
	`LINKDOWN/.1.3.6.1.2.1.2.2.1.1.16`
	`LINKUP/.1.3.6.1.2.1.2.2.1.1.19`
	`LINKUP`
	`AUTHFAIL`
	`EGPLOSS`
	`LINKDOWN`
	`.1.3.6.1.1.0/.3.3.3.3`
	`.1.3.1.1.0`

Note:

Each Standard Driver Sheet can have up to 4096 rows. However, the **Read Trigger**, **Enable Read When Idle**, and **Write Trigger** commands attempt to communicate the entire block of addresses that is configured in the sheet, so if the block of addresses is larger than the maximum block size that is supported by the driver protocol, then you will receive a communication error (e.g., "invalid block size") during run time. Therefore, the maximum block size imposes a practical limit on the number of rows in the sheet.

5. Save and close the worksheet

Additional Notes

Additional notes about the SNMP driver.

Simultaneous Requests when receiving Traps

When using the SNMP driver to receive Traps the Advanced Settings fields under Simultaneous Requests : Maximum and Maximum per station should both be set to 1 as the driver does not support multiple simultaneous requests when receiving traps.

Traps

When receiving traps of the type Trap Inform (supported by SNMP v2) it is important to note that the driver does not send acknowledged messages back to the agent or device.

If Trap port is changed on the driver settings, the driver runtime has to be re-started for the changes to be applied.

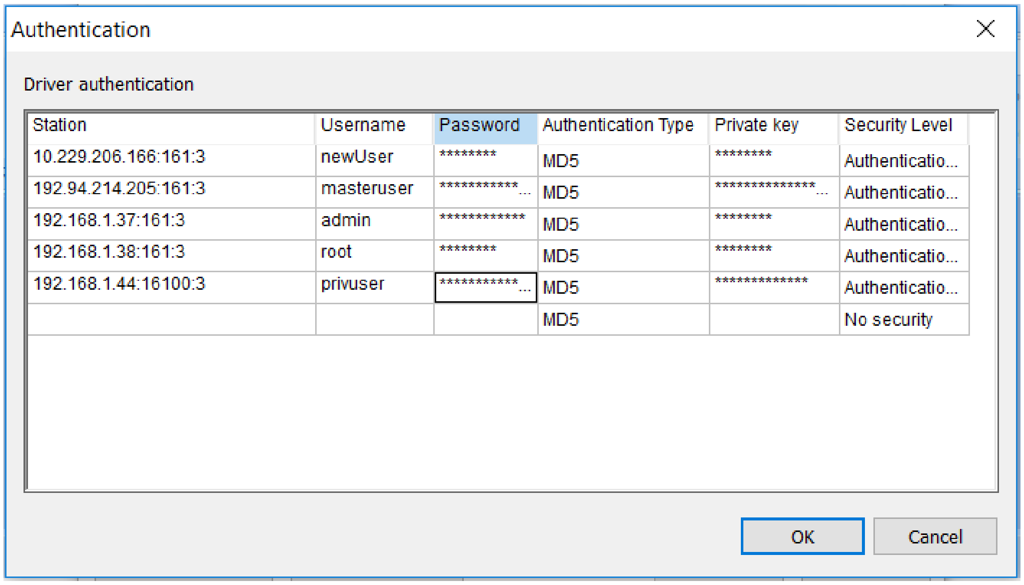
Traps are received from all trap communities present on the device and the driver does not support getting traps only from a specific trap community on the device.

Block size

The maximum block size for the driver is 50. So for successful communication the driver should have no more than 50 tags in one Standard Driver Sheet.

Driver communication using Authentication

Version 3 of SNMP protocol supports security features like authentication and privacy. To configure the authentication information click Authentication button in the driver configuration dialog. The Authentication dialog will be displayed.



Station	Username	Password	Authentication Type	Private key	Security Level
10.229.206.166:161:3	newUser	********	MD5	********	Authenticatio...
192.94.214.205:161:3	masteruser	***********...	MD5	*************...	Authenticatio...
192.168.1.37:161:3	admin	************	MD5	********	Authenticatio...
192.168.1.38:161:3	root	********	MD5	********	Authenticatio...
192.168.1.44:16100:3	privuser	***********...	MD5	*************	Authenticatio...
			MD5		No security

The authentication dialog allows the configuration of security settings for every station specified on main driver sheet or standard driver sheets. The authentication parameters are:

- **Username**: Set the security name used for authenticated SNMPv3 messages
- **Password**: Set the authentication pass phrase used for authenticated SNMPv3 messages.
- **Authentication Type**: Set the authentication protocol (MD5 or SHA) used for authenticated SNMPv3 messages.

- **Private Key**: Set the privacy pass phrase used for encrypted SNMPv3 messages.
- **Security Level**: Set the security level used for SNMPv3 messages (No security; Authentication only; Authentication and Privacy). Appropriate pass phrase(s) must be provided when using any level higher than “No security”.

Authentication field is mandatory only when using security level of Authentication only or Authentication and Privacy.

Privacy Key is mandatory only when using security level of Authentication and Privacy.

✍ **Limitations:**

- Only DES privacy protocol is supported
- When using SNMP v3 with Authentication and Privacy, context name is not used so it is set to empty string.
- Only one user per station is supported.
- Traps are not supported for SNMP v3

✍ **Note:**

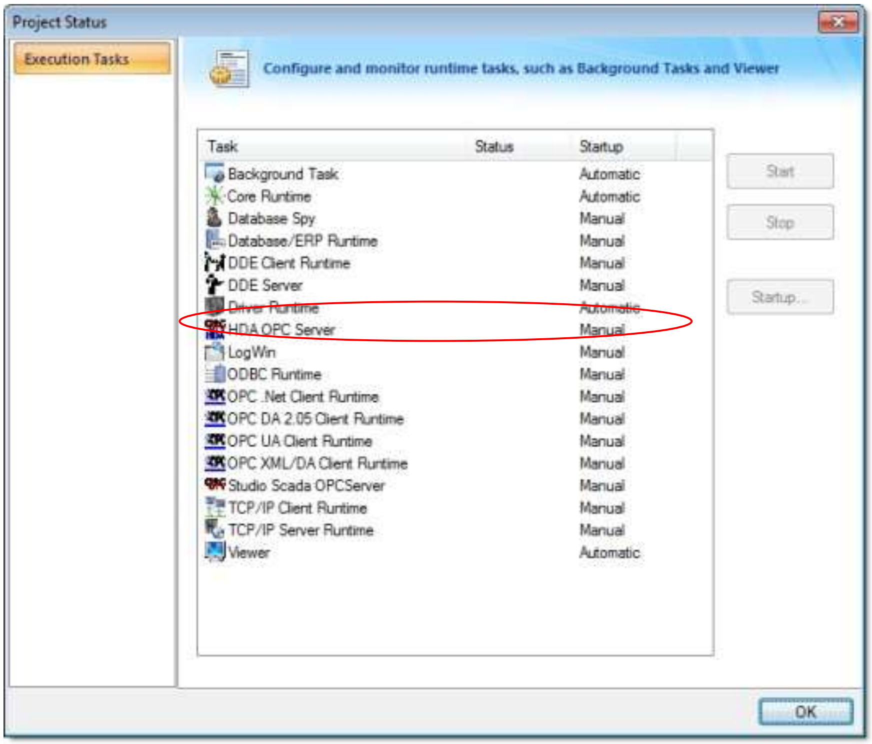
It is not necessary to add information about stations configured with SNMP versions 1 or 2.

Executing the Driver

By default, Studio will automatically execute your selected communication driver(s) during application runtime. However, you may verify your application's runtime execution settings by checking the *Project Status* dialog.

To verify if the communication driver(s) will execute correctly:

1. From the main menu bar, select **Project** → **Status**. The *Project Status* dialog displays:



Configure and monitor runtime tasks, such as Background Tasks and Viewer

Task	Status	Startup
Background Task		Automatic
Core Runtime		Automatic
Database Spy		Manual
Database/ERP Runtime		Manual
DDE Client Runtime		Manual
DDE Server		Manual
Driver Runtime		Automatic
HDA OPC Server		Manual
LogWin		Manual
ODBC Runtime		Manual
OPC .Net Client Runtime		Manual
OPC DA 2.05 Client Runtime		Manual
OPC UA Client Runtime		Manual
OPC XML/DA Client Runtime		Manual
Studio Scada OPCServer		Manual
TCP/IP Client Runtime		Manual
TCP/IP Server Runtime		Manual
Viewer		Automatic

Project Status Dialog

2. Verify that the *Driver Runtime* task is set to **Automatic**.
   - If the setting is correct, then proceed to step 3 below.
   - If the **Driver Runtime** task is set to **Manual**, then select the task and click the **Startup** button to toggle the task's *Startup* mode to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.
4. Start the application to run the driver.

Troubleshooting

This section lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.

Checking status codes

If the SNMP driver fails to communicate with the target device, then the database tag(s) that you configured for the **Read Status** and **Write Status** fields of the driver sheets will receive a status code. Use this status code and the following tables to identify what kind of failure occurred and how it might be resolved.

Status codes for the driver

Error	Description	Possible Causes	Procedure To Solve
19	Invalid Address, please check the format of the address.	An invalid datatype or OID is used	Please refer to the correct address format supported on the Main Driver Sheet or the Standard driver sheet by refering to the sections above.
20	Invalid Station, please check the station format.	The station format is incorrect or the field is blank.	Please refer to the correct station format supported by the driver in the sections above.
21	Failed to initialize the SNMP Library.	SNMP library failed to initialize.	Internal error, restart driver. If issue persists please contact support
22	Connection Failure	Failed to make a connection with the agent	Check if the IP address of the station is valid. Check if the station port and SNMP version number are valid.
23	Invalid Write	Attempted to write to an OID (perform a SET operation) which does not support it.	Check if a SET operation can be performed on the OID. Check if the datatype of the OID is included in the address. Check that the datatype of the OID in the address is correct. Note: SET operation is not supported on traps so write to the OID is not supported when header TRAP is used in the address.
24	Invalid Header	The Header is not valid	The only header supported is TRAP. If a SET or GET operation is to be performed on the OID then the header should be left blank.
25	Invalid Port Number	The port number is not valid or is not available on the device.	Check the port number used. The default port for SNMP is 161 for SET and GET operations and is 162 for TRAPS.
26	Invalid Trap Session	The Trap session is not working as expected.	Check the Trap port number. Check to see if Traps are enabled on the device and if the IP address of the station is correct.Restart the driver if changes are made to the Trap Port. Contact support if problem persists.
27	SNMP version 3 is not supported	The station has the SNMP version number parameter configured as 3.	The driver currently only supports SNMP versions 1 and 2. Check the device and see if communication can be

			successfully done with SNMP version 2.
28	Invalid SNMP version	The station has SNMP version number parameter configured as a value other than 1 or 2.	The driver currently only supports SNMP versions 1 and 2. Check the device and see if communication can be successfully done with SNMP version 1 or 2.
29	Invalid port number	The port for the station or the trap is not available on the device.	Check the Trap port number. Check the port number used for SET and GET operations on the station. The default values for trap port and station port are 162 and 161 respectively. Restart the driver if changes are made.
30	Request failed	The driver is not able to successfully request data for the OIDs.	Check the station format and validity. Check that the device for which driver requests data supports SNMP and that the SNMP agent is running on it. Check the OID format and validity for the request. Restart the driver if changes are made. Contact Support if problem persists.
31	Response is too big	The driver is not able to successfully receive data for the OIDs because the agent is not able to send the data successfully.	Check the block size of the OIDs in the request. Force a smaller block size by adding the OIDs to a Standard driver sheet instead of the Main Driver Sheet (maximum allowed = 50, so make the block size smaller than 50). Restart the driver if changes are made.
32	No such name	The driver is not able to successfully receive data for the OID requested.	Check if the OID exists in the device.
33	Bad value	The driver is not able to successfully receive or write data for the OID requested.	Check if the OID has a datatype supported by the driver. This usually occurs if a SET operation is performed with an incompatible type of data for the OID.
34	Read Only	The driver tried to perform a SET operation on an OID that does not support it.	Check if the OID supports a SET operation. If it does, check that the datatype specified for the OID on the address is correct.
35	Response error	The driver is not able to receive a response for the request.	Check the station format and validity. Check that the device for which driver requests data supports SNMP. Check the OID format and validity for the request. Check that a SET operation is not being performed on an OID that doesnt support it. Restart the driver if changes

			are made. Contact Support if problem persists.
36	Response timeout	The driver is not able to receive a response for the request due to timeout with the agent.	Check the timeout settings on the driver.Check the station format and validity. Check that the device for which driver requests data supports SNMP. Check the OID format and validity for the request. Check that a SET operation is not being performed on an OID that doesnt support it. Restart the driver if changes are made. Contact Support if problem persists.
	Note: The SNMP protocol has a comprehensive list of errors, not all of which are included here. For a complete list, please refer to the SNMP protocol documentation.		

Common status codes

Status Code	Description	Possible Causes	Procedure To Solve
0	OK	Communicating without error.	None required.
-15	Timeout waiting for message to start	• Disconnected cables. • PLC is turned off, in stop mode, or in error mode. • Wrong station number. • Wrong parity (for serial communication). • Wrong RTS/CTS configuration (for serial communication).	• Check cable wiring. • Check the PLC mode — it must be RUN. • Check the station number. • Increase the timeout in the driver's advanced settings. • Check the RTS/CTS configuration (for serial communication).
-33	Invalid driver configuration file	The driver configuration file (`*drivername*.INI`) is missing or corrupt.	Reinstall the driver.
-34	Invalid address	The specified address is invalid or out of range.	Check the supported range of addresses described in this document, and then correct the address.
-35	Driver API not initialized	The driver library was not initialized by the driver.	Contact technical support.
-36	Invalid data type	The specified data type is invalid or out of range.	Check the supported data types described in this document, and then correct the data type.

Status Code	Description	Possible Causes	Procedure To Solve
-37	Invalid header	The specified header in the driver worksheet is invalid or out of range.	Check the supported range of headers described in this document, and then correct the header.
-38	Invalid station	The specified station in the driver worksheet is invalid or out of range.	Check the supported station formats and parameters described in this document, and then correct the station.
-39	Invalid block size	Worksheet is configured with a range of addresses greater than the maximum block size.	Check the maximum block size number of registers described in this document, and then configure your driver worksheet to stay within that limit. Keep in mind that you can create additional worksheets. ☞ **Note:** If you receive this error from a Main Driver Sheet or Tag Integration configuration, please contact Technical Support.
-40	Invalid bit write	Writing to a bit using the attempted action is not supported.	• Writing to a bit using Write Trigger is not supported in some drivers. Modify the driver worksheet to use Write On Tag Change. • The bit is read-only.
-42	Invalid bit number	The bit number specified in the address is invalid. The limit for the bit number depends on the registry type.	Check the addresses to see if there are bit numbers configured outside the valid range for the registry.
-43	Invalid byte number	The byte number specified in the address is invalid. The limit for the byte number depends on the registry type.	Check the addresses to see if there are byte numbers configured outside the valid range for the registry.
-44	Invalid byte write	Writing to a byte using the attempted action is not supported.	The byte is read-only or inacessible.
-45	Invalid string size	The string is more than 1024 characters.	Modify the addresses that have string data type to be less than 1024 characters.
-56	Invalid connection handle	The connection is no longer valid.	Please contact Technical Support.
-57	Message could not be sent	The socket was unable to send the TCP or UDP message.	• Check the station IP address and port number. • Confirm that the device is active and accessible. Try to ping the address.
-58	TCP/IP could not send all bytes	The TCP/IP stack was not able to send all bytes to destination.	• Check the station IP address, port number and/or ID number. • Confirm that the device is active and accessible. • Try to ping the address.
-60	Error to establish TCP/IP connection	Error while establishing a TCP/IP connection with the slave device. Possibly incorrect IP address or port number in the specified station.	• Check the station IP address, port number and/or ID number. • Confirm that the device is active and accessible. • Try to ping the address.
-61	TCP/IP socket error	The TCP/IP connection has been closed by the device.	Confirm that the device is active and accessible. Try to ping the address.

-62	UDP/IP receive call returned error	The UDP socket is in error.	• Check the station IP address, port number and/or ID number. • Confirm that the device is active and accessible. • Try to ping the address.
-63	UDP/IP error initializing	The UDP socket initialization failed.	Confirm that the operating system supports UDP sockets.
-64	UDP/IP receive call returned error	The UDP socket is in error.	• Check the station IP address, port number and/or ID number. • Confirm that the device is active and accessible. • Try to ping the address.
-65	UDP/IP bind error, port number may already be in use	The driver was not able to bind the UDP port.	• Check the port number used by the driver. Check for other programs that might be bound to the UDP port.

Monitoring device communications

You can monitor communication status by establishing an event log in Studio's Output window (LogWin module). To establish a log for Field Read Commands, Field Write Commands and Serial Communication, right-click in the Output window and select the desired options from the pop-up menu.

You can also use the LogWin module to establish an event log on a remote unit that runs Windows Embedded. The log is saved on the unit in the celog.txt file, which can be downloaded later.

If you are unable to establish communication between Studio and the target device, then try instead to establish communication using the device's own programming software. Quite often, communication is interrupted by a hardware or cable problem or by a device configuration error. If you can successfully communicate using the programming software, then recheck the driver's communication settings in Studio.

Contacting Technical Support

If you must contact Technical Support, please have the following information ready:

- **Operating System** and **Project Information**: To find this information, click Support in the Help tab of the ribbon.
- **Driver Version** and **Communication Log**: Displays in the Output window (LogWin module) when the driver is enabled and the project is running is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Revision History

Doc. Revision	Driver Version	Date	Description of Changes
A	1.0	02 Aug 2016	▪ Initial release.
B	1.1	02 Sep 2016	▪ Fixed issue of receiving Enterprise traps on SNMP v1. ▪ Improved log messages to display wrong OID. ▪ mproved support for 64bit integers. ▪ Changed default value of simultaneous requests and simultaneous requests per station to be 1 each. ▪ Improved validation of trap port configuration field.
C	1.2	16 Feb 2017	▪ Fixed issue related to several requests made for SNMP OI server
D	1.3	17 Feb 2017	▪ Minor Internal changes in the driv
E	1.4	20 Dec 2017	▪ Added support for SNMP v3 ▪ Added support for Tables
F	1.5	09 Oct 2018	▪ Fixed issue with write when using OCTETSTRINGHEX header
G	1.6	06 Dec 2018	▪ Fixed the behavior of max items per request parameter