

# SNMP Agent Driver

|                     |                                                         |
|---------------------|---------------------------------------------------------|
| File Name           | SNMPAgent.dll                                           |
| Manufacturer        | SNMP (Simple Network Management Protocol) cf. RFCs IETF |
| Devices             |                                                         |
| Protocol            | SNMP V1 and V2C                                         |
| Version             | 1.0.10                                                  |
| Last Update         | 06/19/2026                                              |
| Platform            | Win32, Eclipse E3 version 3.2 or later                  |
| Dependencies        | IOKit version 2.0 or later                              |
| Superblock Readings | No                                                      |
| Level               | 0                                                       |

## Introduction

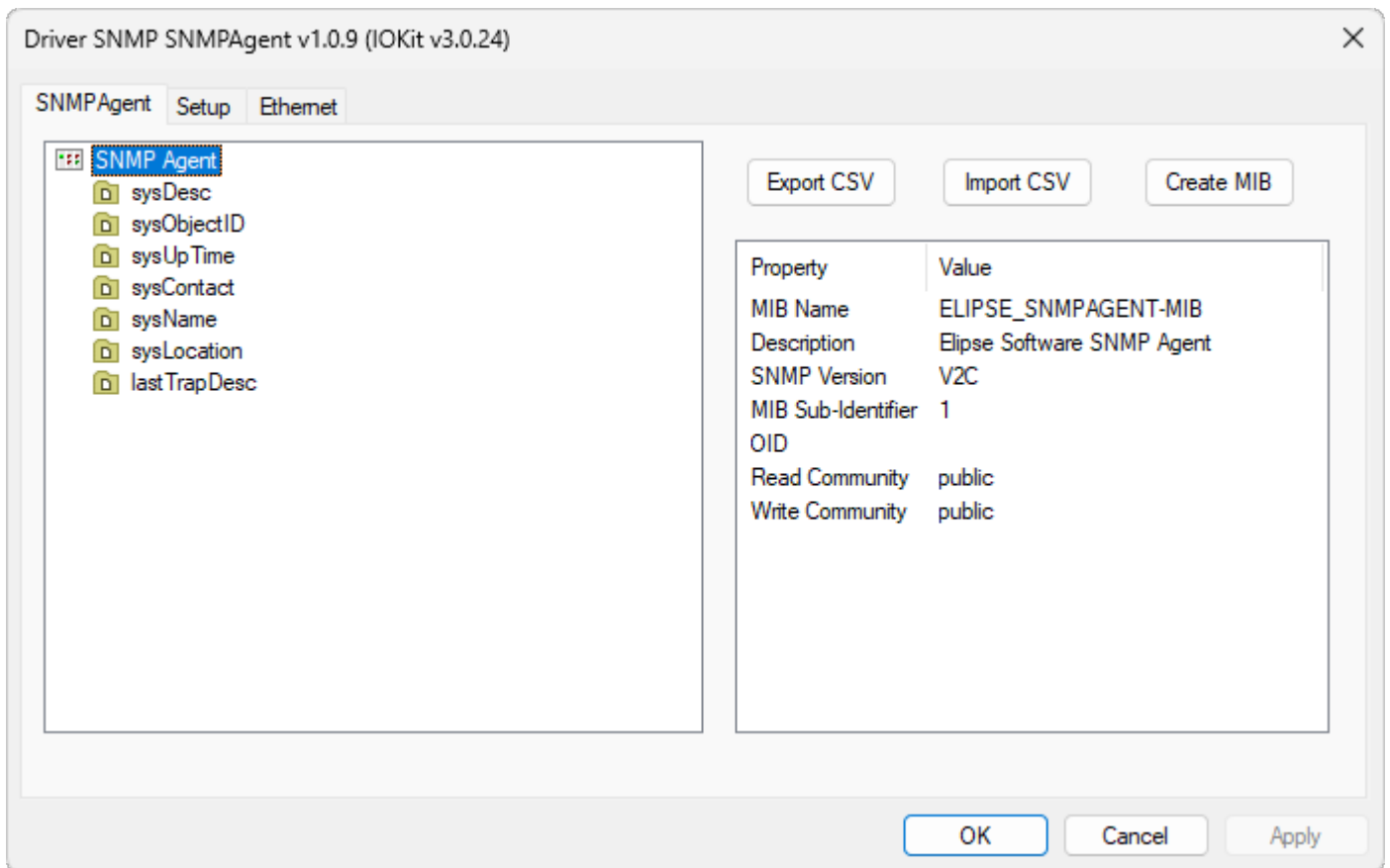
The SNMP Agent Driver communicates using the SNMP (*Simple Network Management Protocol*) protocol. According to SNMP nomenclature, **Agents** are network devices (switches, routers, multiplexers, and no-breaks, among others) that originate information that must be managed, which is then sent to a management system, such as **Eclipse Software's** SNMP Manager Driver.

This Driver currently supports versions V1 and V2C of SNMP protocol, which allows:

- Defining an Agent's data structure (Address Space)
- Generating a MIB (*Management Information Base*) file for importing by SNMP Managers
- Reporting information collected from other protocols for SNMP Managers, using the **GetRequest**, **SetRequest**, and **GetNextRequest** commands
- Sending unsolicited messages, that is, sending SNMP Traps

## Driver Settings

Select the **SNMPAgent** tab on Driver's configuration window, according to the next figure.



**SNMPAgent tab**

On this tab, users can define a configuration for SNMP Agent Driver and its variables or Address Spaces. The tree on the left allows defining or selecting Address Space items and the list on the left allows configuring the selected item.

## SNMP Agent

### SNMP Agent properties

| PROPERTY                  | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MIB Name</b>           | Defines a name for a MIB ( <i>Management Information Base</i> ) used on the file to generate, for importing by an SNMP Manager                                                                                                                                                                                                                                                                        |
| <b>Description</b>        | Description of a MIB                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>SNMP Version</b>       | Defines the version of the protocol used when sending SNMP Traps. This Driver responds to <b>Get</b> , <b>Set</b> , and <b>GetNext</b> requests in versions V1 and V2C                                                                                                                                                                                                                                |
| <b>MIB Sub Identifier</b> | Composes this Agent's initial address, after an initial fixed OID ( <i>Object Identifier</i> ) "1.3.6.1.4.1.4380.12". The meaning of these values is <b>1</b> : ISO Assigned OIDs, <b>3</b> : ISO Identified Organization, <b>6</b> : US Department of Defense, <b>1</b> : OID Assignments for Internet, <b>4</b> : Internet Private, <b>4380</b> : Elipse Software, and <b>12</b> : Starting Address |
| <b>Read Community</b>     | A text identifying the <i>community</i> parameter in SNMP <b>Get</b> requests. The default value of this property is "public"                                                                                                                                                                                                                                                                         |
| <b>Write Community</b>    | A text identifying the <i>community</i> parameter in SNMP <b>Set</b> requests. The default value of this property is "public"                                                                                                                                                                                                                                                                         |

## Default SNMPAgent Items

Items on the next table cannot be removed or edited.

### Non-editable items

| ITEM                       | DESCRIPTION                                                                           |
|----------------------------|---------------------------------------------------------------------------------------|
| <b>sysDesc</b>             | Textual description of an Agent                                                       |
| <b>sysObjectID</b>         | Full identifier of an Agent (OID, <i>Object Identifier</i> )                          |
| <b>sysUpTime</b>           | Time, in hundredths of a second, since this Driver was started for the last time      |
| <b>sysContact</b>          | Identification and contact information of the user responsible for system maintenance |
| <b>sysName</b>             | A name for this Agent                                                                 |
| <b>sysLocation</b>         | Physical location of this Agent                                                       |
| <b>lastTrapDescription</b> | Description of the last Trap sent by this Agent                                       |

## SNMP Groups

Every information to transfer or to receive from an SNMP Manager must be defined inside SNMP groups. To add a group, right-click the **SNMPAgent** item and select the **Add Group** option. Each group has the properties described on the next table.

### Properties of an SNMP group

| PROPERTY                  | DESCRIPTION                                                                                      |
|---------------------------|--------------------------------------------------------------------------------------------------|
| <b>Name</b>               | Name of a group                                                                                  |
| <b>Description</b>        | Description of a group                                                                           |
| <b>OID</b>                | Object Identifier (OID) considering the OID of an upper level and a sub-identifier of this group |
| <b>MIB Sub Identifier</b> | Composes the OID of this item and its children                                                   |

## SNMP Group Items

Every information to transfer or to receive from an SNMP Manager must be declared as items inside an SNMP group. To add an item, right-click a group and select the **Add Item** option. Each item has the properties described on the next table.

### Properties of an SNMP group item

| PROPERTY                  | DESCRIPTION                                                                                     |
|---------------------------|-------------------------------------------------------------------------------------------------|
| <b>Name</b>               | Name of an item                                                                                 |
| <b>Description</b>        | Description of an item                                                                          |
| <b>OID</b>                | Object Identifier (OID) considering the OID of an upper level and a sub-identifier of this item |
| <b>MIB Sub Identifier</b> | Composes the OID of this item                                                                   |

| PROPERTY             | DESCRIPTION                                                                                                                                                                                                             |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SNMP DataType</b> | Possible values are <b>Integer32</b> , <b>Unsigned32</b> , <b>Gauge32</b> , <b>Counter32</b> , <b>Counter64</b> , <b>TimeTicks</b> , <b>OctetString</b> , <b>Opaque</b> , <b>IPAddress</b> , or <b>ObjectIdentifier</b> |
| <b>Access Type</b>   | <b>Read-Only</b> or <b>Read-Write</b>                                                                                                                                                                                   |

## SNMP TrapGroups

Every information to transfer as SNMP Traps must be defined inside a TrapGroup. To add a TrapGroup, right-click the **SNMPAgent** item and select the **Add TrapGroup** option. Each TrapGroup has the properties described on the next table.

### Properties of a TrapGroup

| PROPERTY                         | DESCRIPTION                                                                                                                                                                                                                                                                                                                               |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>                      | Name of a TrapGroup                                                                                                                                                                                                                                                                                                                       |
| <b>Description</b>               | Description of a TrapGroup                                                                                                                                                                                                                                                                                                                |
| <b>OID</b>                       | Object Identifier (OID) considering the OID of an upper level and a sub-identifier of this group                                                                                                                                                                                                                                          |
| <b>MIB Sub Identifier</b>        | Composes the OID of this item and its children                                                                                                                                                                                                                                                                                            |
| <b>Trap Destination</b>          | Destination's IP address or DNS name                                                                                                                                                                                                                                                                                                      |
| <b>Trap Port</b>                 | Destination's UDP port. The default value of this property is 162                                                                                                                                                                                                                                                                         |
| <b>Send All Members together</b> | If this property is configured as True, only when a child item with its <b>TrapTrigger</b> property equal to True changes, the SNMP Trap is sent with the current value of all children of that group, including the Trigger's Tag. If this property is configured as False, each child item triggers an independent Trap when it changes |
| <b>Force RFC 3418 Trap</b>       | Indicates that the Trap must include the <b>SysUpTime</b> members and the SNMP <b>TrapOID</b> in the message, thus becoming compatible with RFC 3418                                                                                                                                                                                      |

## SNMP TrapGroupItems

Every information to transfer as an SNMP Trap must be declared as an item inside a TrapGroup. To add a TrapGroupItem, right-click a group and select the **Add Item** option. Each item has the properties described on the next table.

### Properties of a TrapGroupItem

| PROPERTY                  | DESCRIPTION                                                                                       |
|---------------------------|---------------------------------------------------------------------------------------------------|
| <b>Name</b>               | Name of an item                                                                                   |
| <b>Description</b>        | Description of an item                                                                            |
| <b>OID</b>                | Object Identifier (OID) considering the OID of an upper level and the sub-identifier of this item |
| <b>MIB Sub Identifier</b> | Composes the OID of this item                                                                     |

| PROPERTY                  | DESCRIPTION                                                                                                                                                                                                             |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SNMP DataType</b>      | Possible values are <b>Integer32</b> , <b>Unsigned32</b> , <b>Gauge32</b> , <b>Counter32</b> , <b>Counter64</b> , <b>TimeTicks</b> , <b>OctetString</b> , <b>Opaque</b> , <b>IPAddress</b> , or <b>ObjectIdentifier</b> |
| <b>Trap Trigger</b>       | Informs if a change in this item triggers the TrapGroup, sending the value of all members in a single Trap message                                                                                                      |
| <b>Force Specific OID</b> | A Boolean indicating that the automatic OID, based on trap group OID, must be replaced by the OID informed in the <b>User Specific OID</b> property                                                                     |
| <b>User Specific OID</b>  | OID to replace the default OID                                                                                                                                                                                          |

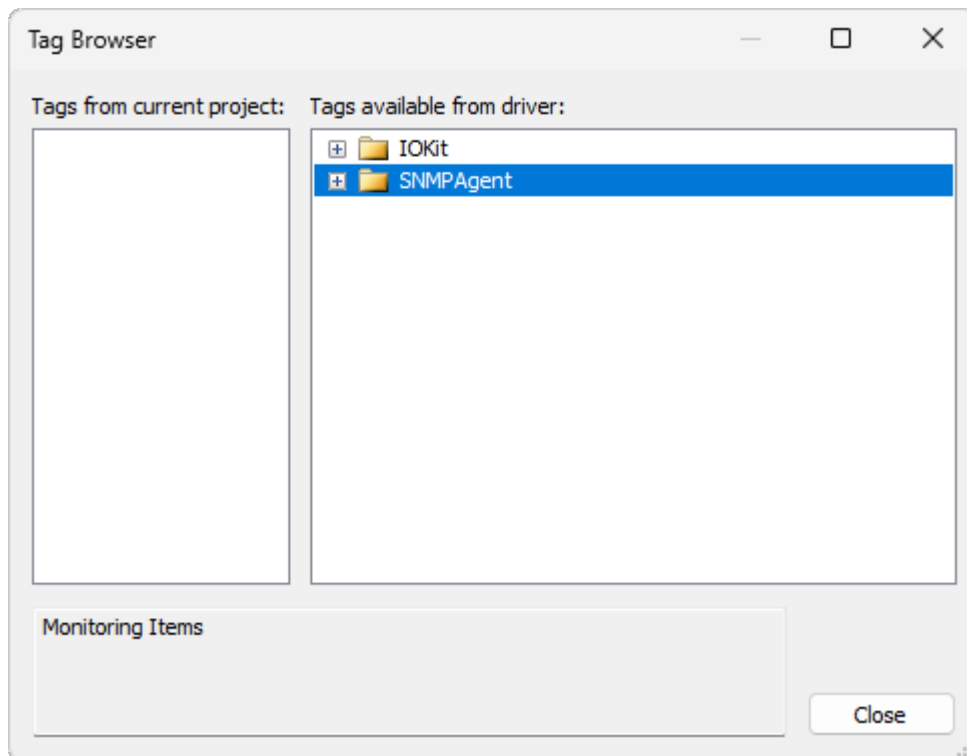
## Create MIB

Click **Create MIB** to save the current Address Space, including groups and TrapGroups, to the selected directory. This MIB file can be imported by an SNMP Manager, such as **Elipse Software's** SNMP Manager Driver.

For more information about the **Setup**, **Serial**, **Ethernet**, **Modem**, and **RAS** tabs, please check topic **Documentation of I/O Interfaces**.

## Tag Browsing

Select the **Tag Browser**  option to import the Address Space defined as Tags in this Driver, according to the next figure. Drag items from the **Tags available from driver** list to the **Tags from current project** list.



Tag Browser window

## Tag Reference

Parameters used for Tags are the following:

- **Device:** Not used, leave it empty

- **Item:** OID (*Object identification*)

Tags defined to receive information from other sources or protocols must be defined with their **AllowWrite** property in True and their **AllowRead** property in False. These Tags receive writings from other sources and this Driver responds with values to SNMP requests.

Tags defined to receive information from SNMP Managers, by using SNMP **Set** requests, must be defined with their **AllowRead** property in True.

## Additional Information

**SNMP** is a typical protocol for management of TCP/IP networks in their application layer, which makes it easy to exchange information among network devices, such as routers, switches, and servers. With this protocol, network administrators can manage connectivity and performance, find and solve problems, and provide information for planning and expansion, among others.

Management software does not follow a conventional Client/Server model. If for some operations a management station behaves as a client and devices as servers, in Trap operations is the opposite. For this, the term *Manager* is used for an application executing in the management station and the term *Agent* is used for an application executing in the device or acting as a gateway for other systems, that is, a protocol translator.

## Documentation of I/O Interfaces

This section contains the documentation of I/O Interfaces referring to the **SNMPAgent** Driver.

### Configuration of a Driver

I/O Interface configuration is performed on a Driver's configuration dialog box. To access the configuration of this dialog box in **Eclipse E3** in version 1.0, follow these steps:

1. Right-click a Driver object (IODriver).
2. Select the **Properties** item on the contextual menu.
3. Select the **Driver** tab.
4. Click **Other parameters**.

In **Eclipse E3** version 2.0 or later, click **Configure driver**  on a Driver's toolbar. In **Eclipse SCADA**, follow these steps:

1. Open the Organizer.
2. Select a Driver on Organizer's tree.
3. Click **Extras** on the **Driver** tab.

Currently, an I/O Interface allows opening only one connection for each Driver. This means that, if users want to access two serial ports, they must add two Drivers to an application and then configure each one of these Drivers for each serial port.

### Configuration Dialog Box

The dialog box of I/O Interfaces allows configuring the I/O connection used by a Driver. This dialog box contains the **Setup**, **Serial**, **Ethernet**, **Modem**, and **RAS** tabs, described on the next topics. If a Driver does not implement a specific I/O

connection, its corresponding tab is not available for configuration. Some Drivers may contain additional tabs, specific for that Driver, on the configuration dialog box.

## Setup Tab

The **Setup** tab contains general configurations of a Driver. This tab is divided into the following groups:

- **General configurations:** Configurations of a Driver's physical layer, time-out, and initialization mode
- **Connection management:** Configurations on how the I/O Interface keeps a connection and which recovery policy is used on failure
- **Logging options:** Controls the generation of log files

The screenshot shows the 'Setup' tab of a configuration dialog. It is divided into three main sections: 'Physical Layer', 'Connection management', and 'Logging Options'. In the 'Physical Layer' section, a dropdown menu is set to 'Ethernet', and there is an unchecked checkbox for 'Start driver OFFLINE'. Below this, 'Timeout' is set to 1000 ms and 'Communication check time' is set to 5000 ms. The 'Connection management' section has a dropdown for 'Mode' set to 'Automatic (managed by the driver)'. It includes three options: 'Retry failed connection every 20 seconds' (checked), 'Give up after 1 failed retries' (unchecked), and 'Disconnect if non-responsive for 0 seconds' (unchecked). The 'Logging Options' section has an unchecked checkbox for 'Log to File' with the path 'C:\eeLogs\MicrolokII\_%DATE%.log'. Below this, 'File size limit (MB)' is set to 0, with a note that '0' is unlimited.

Setup tab

### General options on the Setup tab

| OPTION                          | DESCRIPTION                                                                                                                                                                                                                                        |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Physical Layer</b>           | Select the physical layer on a list. Available options are <b>Serial</b> , <b>Ethernet</b> , <b>Modem</b> , and <b>RAS</b> . The selected interface must be configured on its specific tab                                                         |
| <b>Timeout</b>                  | Configure a time-out, in milliseconds, for the physical layer. This is the amount of time an I/O interface waits to receive any byte from the reception's buffer                                                                                   |
| <b>Communication check time</b> | Set the time, in milliseconds, to define the interval at which communication is considered to be in an inactive state. As long as an I/O Driver receives valid data, its communication state is considered active. However, if during operation an |

| OPTION                      | DESCRIPTION                                                                                                                                                                                                                                                                                      |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                             | I/O Driver does not receive valid data inside this period of time, the state is considered inactive. The communication state is shown in the <b>IO.CommunicationStatus</b> Tag                                                                                                                   |
| <b>Start driver OFFLINE</b> | Select this option so that a Driver starts in <b>Offline</b> mode or stopped. This means that the I/O interface is not created until this Driver is configured to <b>Online</b> mode by using a Tag in an application. This mode enables a dynamic configuration of an I/O interface at run time |

#### Options on the Connection management group

| OPTION                                              | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Mode</b>                                         | Selects a management mode of a connection. Selecting the <b>Automatic</b> option allows a Driver to manage the connection automatically, as specified in the next options. Selecting the <b>Manual</b> option allows an application to fully manage a connection                                                                                                                                            |
| <b>Retry failed connection every ... seconds</b>    | Select this option to enable a Driver's connection retry in a certain interval, in seconds. If the <b>Give up after failed retries</b> option is not selected, this Driver keeps retrying until a connection is performed, or until the application is stopped                                                                                                                                              |
| <b>Give up after ... failed retries</b>             | Enable this option to define a maximum number of connection retries. When the specified number of consecutive connection retries is reached, a Driver goes to the <b>Offline</b> mode, assuming that a hardware problem was detected. If a Driver establishes a successful connection, the number of unsuccessful retries is cleared. If this new connection is lost, then the retry counter starts at zero |
| <b>Disconnect if non-responsive for ... seconds</b> | Enable this option to force a Driver to disconnect if no byte was received by the I/O interface during the specified time-out, in seconds. This time-out must be greater than the time-out configured in the <b>Timeout</b> option                                                                                                                                                                          |



## Options on the Logging Options group

| OPTION                      | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Log to File</b>          | <p>Enable this option and configure the name of a file to write a log. Log files can be large, so use this option for short periods of time, only for testing and debugging purposes. If the <b>%PROCESS%</b> macro is used in the log file name, it is replaced by the identifier of the current process. This option is particularly useful when using several instances of the same Driver in <b>Eclipse E3</b>, thus allowing each instance to generate a separate log file. For example, when configuring this option with value "c:\e3logs\drivers\sim_%PROCESS%.log", it generates a file named c:\e3logs\drivers\sim_00000FDA.log for process <b>0FDAh</b>. Users can also use the <b>%DATE%</b> macro in the file name. In this case a log file is generated every day, in the format <b>aaaa_mm_dd</b>. For example, when configuring this option with value "c:\e3logs\drivers\sim_%DATE%.log", it generates a file named c:\e3logs\drivers\sim_2005_12_31.log in 12/31/2005 and a file named c:\e3logs\drivers\sim_2006_01_01.log in 01/01/2006. Similarly, the <b>%DATE_HOUR%</b> macro generates one log file per hour, in the format <b>aaaa_mm_dd_hh</b></p> |
| <b>File size limit (MB)</b> | <p>Configure the log file size limit, in megabytes. A value equal to 0 (zero) means that there is no size limit for the log file</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

## Ethernet Tab

Use this tab to configure parameters of an **Ethernet** Interface. These parameters, except port configurations, must also be configured for use in the **RAS** Interface.

**Ethernet**

Transport: TCP/IP ☐ Listen for connections on port: 0

☐ PING before connecting ☐ Share listen port with other processes

Timeout: 4000 ms ☐ Interface: (All Interfaces)

Retries: 1 ☐ Use IPv6 ☐ Use SSL SSL Settings

☐ Enable 'ECHO' suppression

IP Filter:

Connect to

Main IP:  Port: 502 ☐ Local port: 0

☐ Backup IP 1:  Port: 0 ☐ Local port: 0

☐ Backup IP 2:  Port: 0 ☐ Local port: 0

☐ Backup IP 3:  Port: 0 ☐ Local port: 0

**Ethernet tab****Available options on the Ethernet tab**

| OPTION                                        | DESCRIPTION                                                                                                                                                                                                         |
|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Transport</b>                              | Select the value <b>TCP/IP</b> for a TCP socket ( <i>stream</i> ) or select the value <b>UDP/IP</b> to use a UDP socket ( <i>connectionless datagram</i> )                                                          |
| <b>Listen for connections on port</b>         | Use this option to wait for new connections in a specific IP port, common in Slave Drivers. If this option remains unselected, a Driver connects to the address and port specified in the <b>Connect to</b> option  |
| <b>Share listen port with other processes</b> | Select this option to share the listening port with other Drivers and processes                                                                                                                                     |
| <b>Interface</b>                              | Select the local network interface, identified by its IP address, that a Driver uses to establish and receive connections, or select the value <b>(All Interfaces)</b> to allow connection in any network interface |
| <b>Use IPv6</b>                               | Select this option to force a Driver to use addresses in <b>IPv6</b> format on all Ethernet connections. Leave this option deselected to use the <b>IPv4</b> format                                                 |
| <b>Enable 'ECHO' suppression</b>              | Enable this option to remove the echo from received data. An echo is a copy of sent data, which can be returned before a reply message                                                                              |
| <b>IP Filter</b>                              | List of restricted or allowed IP addresses from where a Driver accepts connections ( <i>Firewall</i> ). Please check the <b>IO.Ethernet.IPFilter</b> property for more information                                  |
| <b>PING before connecting</b>                 | Enable this option to execute a <b>ping</b> command, that is, to check whether a device can be reached on a network, for a device before trying a socket connection. This is a quick way                            |

| OPTION | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | <p>of determining a successful connection before trying to open a socket with a device. The time-out of a connection with a socket can be very high. The available options are:</p> <ul style="list-style-type: none"> <li>• <b>Timeout:</b> Specify the number of milliseconds to wait for a reply from a <b>ping</b> command. Users must use a <b>ping</b> command to check the normal reply time, configuring this option for a value above that average. Usually this value can be configured between 1000 and 4000 milliseconds, that is, between 1 (one) and 4 (four) seconds</li> <li>• <b>Retries:</b> Number of retries of a <b>ping</b> command, not counting the first attempt. If all attempts fail, then the socket connection is aborted</li> </ul> |

#### Available options on the Connect to group

| OPTION                       | DESCRIPTION                                                                                                                                                                                                                              |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Main IP</b>               | Type the IP address of a remote device. Users can use an IP address separated by dots, as well as a URL. In case of a URL, a Driver uses the available DNS service to map that URL to an IP address, such as "192.168.0.13" or "Server1" |
| <b>Port</b>                  | Type the IP port of a remote device, between 0 (zero) and 65535                                                                                                                                                                          |
| <b>Local port</b>            | Select this option to use a fixed local IP port when connecting to a remote device                                                                                                                                                       |
| <b>Backup IP 1, 2, and 3</b> | Indicate the IP address, the IP port, and the fixed local IP port of up to 3 (three) backup addresses of a remote device                                                                                                                 |

## General Configurations

This section contains information about the configuration of general **I/O Tags** and **Properties** of I/O Interfaces.

### I/O Tags

#### General I/O Interfaces Tags (N2/B2 = 0)

The Tags described next are provided for all supported I/O Interfaces.

#### IO.CommunicationStatus

|                             |                        |
|-----------------------------|------------------------|
| <b>Type of Tag</b>          | I/O Tag                |
| <b>Type of Access</b>       | Reading                |
| <b>N1 Parameter</b>         | -1 (minus one)         |
| <b>N2 Parameter</b>         | 0 (zero)               |
| <b>N3 Parameter</b>         | 0 (zero)               |
| <b>N4 Parameter</b>         | 6 (six)                |
| <b>String Configuration</b> | IO.CommunicationStatus |

This Tag informs the communication status of a Driver. It indicates how communication works relative to receiving valid data within a time period arbitrated in the configuration. For more information, please check topic **Setup Tab**. Possible values are **0 - Inactive communication**: The Driver did not receive valid data or stopped receiving data after *n* milliseconds, as configured in the properties window, or **1 - Active communication**: The Driver is receiving valid data.

## IO.IOKitEvent

|                    |                |
|--------------------|----------------|
| Type of Tag        | Block Tag      |
| Type of Access     | Read-Only      |
| B1 Parameter       | -1 (minus one) |
| B2 Parameter       | 0 (zero)       |
| B3 Parameter       | 0 (zero)       |
| B4 Parameter       | 1 (one)        |
| Size Property      | 4 (four)       |
| ParamItem Property | IO.IOKitEvent  |

This Block returns Driver events generated by several sources in I/O Interfaces. The **TimeStamp** property of this Block represents the moment this event occurred. The Block Elements are the following:

- **Element 0**: Type of event. Possible values are **0**: Information, **1**: Warning, or **2**: Error
- **Element 1**: Source of an event. Possible values are **0**: Driver (specific of a Driver), **-1**: IOKit (generic events of I/O Interfaces), **-2**: **Serial** Interface, **-3**: **Modem** Interface, **-4**: **Ethernet** Interface, or **-5**: **RAS** Interface
- **Element 2**: Error number, specific for each source of event
- **Element 3**: Message of an event, a **String** specific for each event

### NOTE

A Driver keeps a maximum number of 100 events internally. If additional events are reported, older events are discarded.

## IO.PhysicalLayerStatus

|                      |                        |
|----------------------|------------------------|
| Type of Tag          | I/O Tag                |
| Type of Access       | Read-Only              |
| N1 Parameter         | -1 (minus one)         |
| N2 Parameter         | 0 (zero)               |
| N3 Parameter         | 0 (zero)               |
| N4 Parameter         | 2 (two)                |
| String Configuration | IO.PhysicalLayerStatus |

This Tag indicates the status of a physical layer. Possible values are the following:

- **0:** Physical layer stopped, that is, a Driver is in **Offline** mode, the physical layer failed when initializing, or exceeded the maximum number of reconnection attempts
- **1:** Physical layer started but not connected, that is, a Driver is in **Online** mode but the physical layer is not connected. If the **Connection management** option is configured with the value **Automatic**, the physical layer can be connecting, disconnecting, or waiting for a reconnection attempt. If the **Connection management** option is configured with the value **Manual**, then the physical layer remains in this status until forced to connect
- **2:** Physical layer connected, that is, the physical layer is ready for use. This **DOES NOT** mean a device is connected, only that the access layer is working

## IO.SetConfigurationParameters

|                    |                               |
|--------------------|-------------------------------|
| Type of Tag        | Block Tag                     |
| Type of Access     | Read-Only                     |
| B1 Parameter       | -1 (minus one)                |
| B2 Parameter       | 0 (zero)                      |
| B3 Parameter       | 0 (zero)                      |
| B4 Parameter       | 3 (three)                     |
| Size Property      | 2 (two)                       |
| ParamItem Property | IO.SetConfigurationParameters |

Use this Tag to change any property of a Driver's configuration dialog box at run time.

This Tag works only while a Driver is in **Offline** mode. To start a Driver in **Offline** mode, select the **Start driver OFFLINE** option on that Driver's configuration dialog box. Users can write to a PLC Tag or to a Block Tag containing the parameters to change. Writing individual Block Elements is not supported, the whole Block must be written at once.

In **Eclipse SCADA**, users must use a Block Tag. Every parameter to configure uses two Block Elements. For example, if users want to configure 3 (three) parameters, then the size of the Block must be 6 (six,  $3 \times 2$ ). The first Element is the property's name, as a **String**, and the second Element is the property's value, according to the next example.

```
// 'Block' must be a Block Tag with automatic reading,
// scan reading, and automatic writings disabled.
// Configure all parameters
Block.element001 = "IO.Type" // Parameter 1
Block.element002 = "Serial"
Block.element003 = "IO.Serial.Port" // Parameter 2
Block.element004 = 1
Block.element005 = "IO.Serial.BaudRate" // Parameter 3
Block.element006 = 19200
// Writes the whole Block
Block.Write()
```

When using **Eclipse E3**, the ability to create arrays at run time allows using an I/O Tag as well as a Block Tag. Users can use the **Write** method of a Driver to send the parameters directly to that Driver, without creating a Tag, according to the next example.

```
Dim arr(6)
' Configure all array elements
arr(1) = "IO.Type"
arr(2) = "Serial"
arr(3) = "IO.Serial.Port"
arr(4) = 1
arr(5) = "IO.Serial.BaudRate"
arr(6) = 19200
' There are two methods to send parameters
' Method 1: Using an I/O Tag
tag.WriteEx arr
' Method 2: Without using a Tag
Driver.Write -1, 0, 0, 3, arr
```

A variation of the previous example uses a bidimensional array.

```
Dim arr(10)
' Configure all array elements. Notice the array was resized
' to 10 elements. Empty array elements are ignored by a Driver
arr(1) = Array("IO.Type", "Serial")
arr(2) = Array("IO.Serial.Port", 1)
arr(3) = Array("IO.Serial.BaudRate", 19200)
Driver.Write -1, 0, 0, 3, arr
```

A Driver does not validate parameter names or passed values, therefore be careful when writing parameters and values. The **Write** method fails if the configuration array is incorrectly created. Users can check the log of a Driver or use the **writeStatus** parameter of the **WriteEx** method to find out the exact cause of an error.

```
Dim arr(10), strError
arr(1) = Array("IO.Type", "Serial")
arr(2) = Array("IO.Serial.Port", 1)
arr(3) = Array("IO.Serial.BaudRate", 19200)
If Not Driver.WriteEx -1, 0, 0, 3, arr, , , strError Then
    MsgBox "Failed configuring Driver parameters: " + strError
End If
```

## IO.WorkOnline

| Type of Tag          | I/O Tag            |
|----------------------|--------------------|
| Type of Access       | Reading or Writing |
| N1 Parameter         | -1 (minus one)     |
| N2 Parameter         | 0 (zero)           |
| N3 Parameter         | 0 (zero)           |
| N4 Parameter         | 4 (four)           |
| String Configuration | IO.WorkOnline      |

This Tag informs the current status of a Driver and allows starting or stopping the physical layer. Possible values are the following:

- **0 - Driver Offline:** Physical layer is closed or stopped. This mode allows a dynamic configuration of a Driver's parameters using the **IO.SetConfigurationParameters** Tag
- **1 - Driver Online:** Physical layer is open or executing. While in **Online** mode, the physical layer can be connected or disconnected and its current status can be checked using the **IO.PhysicalLayerStatus** Tag

In the next example, using **Elipse E3**, a Driver is configured to **Offline** mode, its COM port is changed, and then configured to **Online** mode again.

```
'Configure to Offline mode
Driver.Write -1, 0, 0, 4, 0
'Change port to COM2
Driver.Write -1, 0, 0, 3, Array("IO.Serial.Port", 2)
'Configure to Online mode
Driver.Write -1, 0, 0, 4, 1
```

The **Write** method may fail when configuring a Driver to **Online** mode, that is, writing the value 1 (one). In this case, this Driver remains in **Offline** mode. The cause of failure can be:

- Type of physical layer incorrectly configured, probably an invalid value was configured in the **IO.Type** property
- This Driver may have run out of memory
- Physical layer probably did not create its working thread. Search the log file for a message "Failed to create physical layer thread!"
- Physical layer could not start. The cause of this failure depends on the type of physical layer. It can be an invalid serial port number, a failure when starting Windows Sockets, or a failure when starting TAPI (modem), among others. This cause is recorded on the log file

#### IMPORTANT

Even if the configuration of a Driver to **Online** mode is successful, this does not necessarily mean the physical layer is ready to use, that is, ready to execute input and output operations with an external device. The **IO.PhysicalLayerStatus** Tag must be checked to ensure the physical layer is connected and ready for communication.

## Properties

These are general properties of all supported I/O Interfaces.

### IO.ConnectionMode

**9** Controls the management mode of a Connection. Possible values are **0**: Automatic mode, in which a Driver manages the connection or **1**: Manual mode, in which an application manages the connection.

### IO.GiveUpEnable

☒ When configured to True, defines a maximum number of reconnection attempts. If all reconnection attempts fail, a Driver enters the **Offline** mode. When configured to False, a Driver tries until a reconnection is successful.

### IO.GiveUpTries

**9** Number of reconnection attempts before this one is aborted. For example, if the value of this property is equal to 1 (one), a Driver tries only one reconnection when the connection is lost. If this one fails, this Driver enters the **Offline** mode.

### IO.InactivityEnable

☒ Configure to True to enable and to False to disable inactivity detection. The physical layer is disconnected if inactive for a certain period of time. The physical layer is considered inactive only if it is capable of sending data but not capable of receiving it back.

### IO.InactivityPeriodSec

**9** Number of seconds to check for inactivity. If the physical layer is inactive for this period of time, it is then disconnected.

## IO.RecoverEnable

■ Configure to True to enable a Driver to recover lost connections and to False to leave a Driver in **Offline** mode when a connection is lost.

## IO.RecoverPeriodSec

9 Delay time between two connection attempts, in seconds.

### NOTE

The first reconnection is executed immediately after a connection is lost.

## IO.StartOffline

■ Configure to True to start a Driver in **Offline** mode and to False to start a Driver in **Online** mode.

### NOTE

It is pointless to change this property at run time, as it can only be changed when a Driver is already in **Offline** mode. To configure a Driver in **Online** mode at run time, write the value 1 (one) to the **IO.WorkOnline** Tag.

## IO.TimeoutMs

9 Defines a time-out for the physical layer, in milliseconds. One second is equal to 1000 milliseconds.

## IO.Type

A Defines the type of physical interface used by a Driver. Possible values are the following:

- **N or None**: Does not use a physical interface, that is, a Driver must provide a customized interface
- **S or Serial**: Uses a local serial port (COM $n$ )
- **M or Modem**: Uses a local modem, internal or external, accessed via TAPI (*Telephony Application Programming Interface*)
- **E or Ethernet**: Uses a TCP/IP or UDP/IP socket
- **R or RAS**: Uses a **RAS** (*Remote Access Server*) Interface. A Driver connects to a RAS device using the **Ethernet** Interface and then sends an **AT** (*dial*) command

## Statistical Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of I/O Interfaces statistics.

## I/O Tags

### Tags of I/O Interface Statistics (N2/B2 = 0)

The Tags described next display statistics for all I/O Interfaces.



## IO.Stats.Partial.BytesRecv

|                                |                            |
|--------------------------------|----------------------------|
| <b>Type of Tag</b>             | I/O Tag                    |
| <b>Type of Access</b>          | Read-Only                  |
| <b>N1 Parameter</b>            | -1 (minus one)             |
| <b>N2 Parameter</b>            | 0 (zero)                   |
| <b>N3 Parameter</b>            | 0 (zero)                   |
| <b>N4 Parameter</b>            | 1101                       |
| <b>Configuration by String</b> | IO.Stats.Partial.BytesRecv |

This Tag returns the number of bytes received in the current connection.

## IO.Stats.Partial.BytesSent

|                                |                            |
|--------------------------------|----------------------------|
| <b>Type of Tag</b>             | I/O Tag                    |
| <b>Type of Access</b>          | Read-Only                  |
| <b>N1 Parameter</b>            | -1 (minus one)             |
| <b>N2 Parameter</b>            | 0 (zero)                   |
| <b>N3 Parameter</b>            | 0 (zero)                   |
| <b>N4 Parameter</b>            | 1100                       |
| <b>Configuration by String</b> | IO.Stats.Partial.BytesSent |

This Tag returns the number of bytes sent through the current connection.

## IO.Stats.Partial.TimeConnectedSeconds

|                                |                                       |
|--------------------------------|---------------------------------------|
| <b>Type of Tag</b>             | I/O Tag                               |
| <b>Type of Access</b>          | Read-Only                             |
| <b>N1 Parameter</b>            | -1 (minus one)                        |
| <b>N2 Parameter</b>            | 0 (zero)                              |
| <b>N3 Parameter</b>            | 0 (zero)                              |
| <b>N4 Parameter</b>            | 1102                                  |
| <b>Configuration by String</b> | IO.Stats.Partial.TimeConnectedSeconds |

This Tag returns the number of seconds a Driver is connected in the current connection or 0 (zero) if a Driver is disconnected.

## IO.Stats.Partial.TimeDisconnectedSeconds

|                                |                                          |
|--------------------------------|------------------------------------------|
| <b>Type of Tag</b>             | I/O Tag                                  |
| <b>Type of Access</b>          | Read-Only                                |
| <b>N1 Parameter</b>            | -1 (minus one)                           |
| <b>N2 Parameter</b>            | 0 (zero)                                 |
| <b>N3 Parameter</b>            | 0 (zero)                                 |
| <b>N4 Parameter</b>            | 1103                                     |
| <b>Configuration by String</b> | IO.Stats.Partial.TimeDisconnectedSeconds |

This Tag returns the number of seconds a Driver is disconnected since the last connection ended or 0 (zero) if a Driver is connected.

## IO.Stats.Total.BytesRecv

|                                |                          |
|--------------------------------|--------------------------|
| <b>Type of Tag</b>             | I/O Tag                  |
| <b>Type of Access</b>          | Read-Only                |
| <b>N1 Parameter</b>            | -1 (minus one)           |
| <b>N2 Parameter</b>            | 0 (zero)                 |
| <b>N3 Parameter</b>            | 0 (zero)                 |
| <b>N4 Parameter</b>            | 1001                     |
| <b>Configuration by String</b> | IO.Stats.Total.BytesRecv |

This Tag returns the number of bytes received since a Driver was loaded.

## IO.Stats.Total.BytesSent

|                                |                          |
|--------------------------------|--------------------------|
| <b>Type of Tag</b>             | I/O Tag                  |
| <b>Type of Access</b>          | Read-Only                |
| <b>N1 Parameter</b>            | -1 (minus one)           |
| <b>N2 Parameter</b>            | 0 (zero)                 |
| <b>N3 Parameter</b>            | 0 (zero)                 |
| <b>N4 Parameter</b>            | 1000                     |
| <b>Configuration by String</b> | IO.Stats.Total.BytesSent |

This Tag returns the number of bytes sent since a Driver was loaded.

## IO.Stats.Total.ConnectionCount

|                         |                                |
|-------------------------|--------------------------------|
| Type of Tag             | I/O Tag                        |
| Type of Access          | Read-Only                      |
| N1 Parameter            | -1 (minus one)                 |
| N2 Parameter            | 0 (zero)                       |
| N3 Parameter            | 0 (zero)                       |
| N4 Parameter            | 1004                           |
| Configuration by String | IO.Stats.Total.ConnectionCount |

This Tag returns the number of connections a Driver already established, successfully, since it was loaded.

## IO.Stats.Total.TimeConnectedSeconds

|                         |                                     |
|-------------------------|-------------------------------------|
| Type of Tag             | I/O Tag                             |
| Type of Access          | Read-Only                           |
| N1 Parameter            | -1 (minus one)                      |
| N2 Parameter            | 0 (zero)                            |
| N3 Parameter            | 0 (zero)                            |
| N4 Parameter            | 1002                                |
| Configuration by String | IO.Stats.Total.TimeConnectedSeconds |

This Tag returns the number of seconds a Driver remained connected since it was loaded.

## IO.Stats.Total.TimeDisconnectedSeconds

|                         |                                        |
|-------------------------|----------------------------------------|
| Type of Tag             | I/O Tag                                |
| Type of Access          | Read-Only                              |
| N1 Parameter            | -1 (minus one)                         |
| N2 Parameter            | 0 (zero)                               |
| N3 Parameter            | 0 (zero)                               |
| N4 Parameter            | 1003                                   |
| Configuration by String | IO.Stats.Total.TimeDisconnectedSeconds |

This Tag returns the number of seconds a Driver remained disconnected since it was loaded.

## Properties

Currently, there are no properties defined specifically to display I/O Interface statistics at run time.

# Ethernet Interface Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of an **Ethernet** Interface.

## I/O Tags

### Tags of an Ethernet Interface (N2/B2 = 4)

The Tags described next allow controlling and identifying an **Ethernet** Interface at run time and they are also valid when the **RAS** Interface is selected.

#### IMPORTANT

These Tags are available **ONLY** while a Driver is in **Online** mode.

### IO.Ethernet.IPSelect

| Type of Tag          | I/O Tag              |
|----------------------|----------------------|
| Type of Access       | Reading or Writing   |
| N1 Parameter         | -1 (minus one)       |
| N2 Parameter         | 0 (zero)             |
| N3 Parameter         | 4 (four)             |
| N4 Parameter         | 0 (zero)             |
| String Configuration | IO.Ethernet.IPSelect |

Indicates the active IP address. Possible values are **0**: The main IP address is selected, **1**: The first alternative or backup IP address is selected, **2**: The second alternative or backup IP address is selected, or **3**: The third alternative or backup IP address is selected.

If the **Ethernet** or **RAS** Interface is connected, this Tag indicates which one of the four configured IP addresses is in use. If the Interface is disconnected, this Tag indicates which IP address is used first on the next attempt to connect.

During the connection process, if the active IP address is not available, the I/O Interface tries to connect using the other IP address. If the connection with the alternative IP address works, it is configured as the active IP address (automatic switchover).

To force a manual switchover, write values from 0 (zero) to 3 (three) to this Tag. This forces a reconnection with the specified IP address (**0**: Main address or **1, 2, 3**: Alternative address) if a Driver is currently connected. If a Driver is disconnected, this Tag configures the active IP address for the next attempt to connect.

## IO.Ethernet.IPSwitch

|                      |                      |
|----------------------|----------------------|
| Type of Tag          | I/O Tag              |
| Type of Access       | Write-Only           |
| N1 Parameter         | -1 (minus one)       |
| N2 Parameter         | 0 (zero)             |
| N3 Parameter         | 4 (four)             |
| N4 Parameter         | 1 (one)              |
| String Configuration | IO.Ethernet.IPSwitch |

Any value written to this Tag forces a manual switchover. If the main IP address is active, then the first alternative or backup IP address is activated, and so on for all alternative IP addresses and returning to the main address until a connection is established.

If a Driver is disconnected, this Tag configures the active IP address for the next attempt to connect.

## IO.Ethernet.SocketState

|                      |                         |
|----------------------|-------------------------|
| Type of Tag          | I/O Tag                 |
| Type of Access       | Read-Only               |
| N1 Parameter         | -1 (minus one)          |
| N2 Parameter         | 0 (zero)                |
| N3 Parameter         | 4 (four)                |
| N4 Parameter         | 2 (two)                 |
| String Configuration | IO.Ethernet.SocketState |

The Value property of this Tag corresponds to socket states as a map of bits:

- **Bit 0:** 0 (zero, not listening) or 1 (one, listening)
- **Bit 1:** 0 (zero, disconnected) or 1 (one, connected)

## Properties

These properties control the configuration of an **Ethernet** Interface.

### NOTE

The **Ethernet** Interface is also used by the **RAS** Interface.

## IO.Ethernet.AcceptConnection

☐ Configure to False if a Driver must not accept external connections, that is, if a Driver behaves as a master, or configure to True to enable the reception of connections, that is, if a Driver behaves as a slave.

## IO.Ethernet.BackupEnable[2,3]

■ Configure to True to enable an alternative or backup IP address. If the reconnection attempt with the main IP address fails, a Driver tries to use an alternative IP address. Configure to False to disable its usage.

## IO.Ethernet.BackupIP[2,3]

▲ Alternative or backup IP address of a remote device. Users can use a numerical address, as well as a device's host name, such as "192.168.0.7" or "SERVER2".

## IO.Ethernet.BackupLocalPort[2,3]

9 Local port number to be used when connecting to an alternative IP address of a remote device. Used only if **IO.Ethernet.BackupLocalPortEnable** is equal to True.

## IO.Ethernet.BackupLocalPortEnable[2,3]

■ Configure to True to force the use of a specific local port when connecting to an alternative or backup IP address or configure to False to use any available local port.

## IO.Ethernet.BackupPort[2,3]

9 Port number of an alternative or backup IP address of a remote device, used with the **IO.Ethernet.BackupIP** property.

## IO.Ethernet.IPFilter

▲ List with a comma-separated IPv4 or IPv6 addresses, which defines from which addresses a Driver accepts or blocks connections. Users can use asterisks, such as "192.168.\*.\*", or intervals, such as "192.168.0.41-50", in any part of IP addresses. To block an IP address or a range of IP addresses, use the tilde ("~") character at the beginning of the address, according to the next examples:

- **192.168.0.24**: Accepts only connections from IPv4 address 192.168.0.24
- **192.168.0.41-50**: Accepts connections from IPv4 addresses in the interval between 192.168.0.41 and 192.168.0.50
- **192.168.0.\***: Accepts connections from IPv4 addresses in the interval between 192.168.0.0 and 192.168.0.255
- **fe80:3bf:877::\*\* (expands to fe80:03bf:0877:0000:0000:0000:0000:0000:\*\*\*\*)**: Accepts connections from IPv6 addresses in the interval between fe80:03bf:0877:0000:0000:0000:0000:0000 and fe80:03bf:0877:0000:0000:0000:0000:ffff:ffff
- **192.168.0.10, 192.168.0.15, 192.168.0.20**: Accepts connections from IPv4 addresses 192.168.0.10, 192.168.0.15, and 192.168.0.20
- **~192.168.0.95, 192.168.0.\***: Accepts connections from IPv4 addresses in the interval between 192.168.0.0 and 192.168.0.255, except the IPv4 address 192.168.0.95

When a Driver receives a connection attempt, the list of filters is scanned sequentially from left to right, searching for a specific authorization or block for the IP address where the connection comes from. If no element on the list corresponds to the IP address, the authorization or block are dictated by the last element of that list:

- If the last element on the list is an authorization, such as "192.168.0.24", then all IP addresses not found on the list are blocked
- If the last element on the list is a block, such as "~192.168.0.24", then all IP addresses not found on the list are authorized

If an IP address appears on more than one filter on the list, the leftmost filter has precedence. For example, in case of "~192.168.0.95, 192.168.0.\*", the IP address 192.168.0.95 fits both rules, but the rule that wins is the leftmost one, "~192.168.0.95", and therefore this IP address is blocked.

When **IOKit** blocks a connection, it logs a message "Blocked incoming socket connection from {IP}!".

In case of UDP connections in broadcast listening mode, in which a Driver can receive packets from different IP addresses, blocks or permissions are performed at each packet received. If a packet is received from a blocked IP address, it logs a message "Blocked incoming packet from {IP} (discarding {N} bytes)!".

## IO.Ethernet.ListenIP

 IP address of the local network interface that a Driver uses to establish and accept connections. Leave this property empty to establish and accepts connections using any local network interface.

## IO.Ethernet.ListenPort

 Number of the IP port used by a Driver to listen to connections.

## IO.Ethernet.MainIP

 IP address of a remote device. Users can use a numerical address, as well as a device's host name, such as "192.168.0.7" or "SERVER2".

## IO.Ethernet.MainLocalPort

 Local port number to use when connecting to the main IP address of a remote device. This value is only used if the **IO.Ethernet.MainLocalPortEnable** property is equal to True.

## IO.Ethernet.MainLocalPortEnable

 Configure to True to force the use of a specific local port when connecting to the main IP address of a remote device or configure to False to use any available local port.

## IO.Ethernet.MainPort

 Number of the IP port of a remote device, used with the **IO.Ethernet.MainIP** property.

## IO.Ethernet.PingEnable

 Configure to True to enable sending a **ping** command to the IP address of a remote device, before trying to connect to the socket. This socket's connection time-out cannot be controlled, therefore sending a **ping** command before connecting is a fast way to detect if the connection is going to fail. Configure to False to disable a **ping** command.

## IO.Ethernet.PingTimeoutMs

 Delay time to wait for a response from a **ping** command, in milliseconds.

## IO.Ethernet.PingTries

 Maximum number of attempts of a **ping** command. Minimum value is 1 (one), including the first **ping** command.

## IO.Ethernet.ShareListenPort

■ Configure to True to share a listening port with other Drivers and processes or False to open a listening port in exclusive mode. To successfully share a listening port, all Drivers and processes that use that port must open it in shared mode. When a listening port is shared, each incoming connection is distributed to one of the processes listening. This way, if a Slave Driver only supports one connection at a time, users can use several instances of this Driver listening on the same port, therefore simulating a Driver with support for multiple connections.

## IO.Ethernet.SupressEcho

■ Configure to True to eliminate echoes in communication. An echo is the unwanted reception of an exact copy of all data packets a Driver sent to a device.

## IO.Ethernet.Transport

⚠ Defines a transport protocol. Possible values are **T or TCP**: Uses the TCP/IP protocol or **U or UDP**: Uses the UDP/IP protocol.

## IO.Ethernet.UseIPv6

■ Configure to True to use IPv6 addresses on all Ethernet connections or configure to False to use IPv4 addresses (default).

## Driver Revision History

| VERSION | DATE       | AUTHOR      | COMMENTS                                                                                                                                                                                                                                                                                                 |
|---------|------------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.0.10  | 06/19/2026 | M. Ludwig   | <ul style="list-style-type: none"> <li>Driver updated to <b>IOKit</b> library version <b>3.0</b> and Visual Studio 2022 (Case 38019).</li> </ul>                                                                                                                                                         |
| 1.0.8   | 11/25/2020 | M. Salvador | <ul style="list-style-type: none"> <li>Messages received with a size greater than 127 bytes are now supported (Case 30098).</li> </ul>                                                                                                                                                                   |
| 1.0.7   | 12/18/2019 | M. Salvador | <ul style="list-style-type: none"> <li>Support for RFC 3418 added in Traps (Case 25318).</li> <li>SNMP Walk procedure corrected including new items from the same RFC (Case 25318).</li> </ul>                                                                                                           |
| 1.0.4   | 09/03/2019 | M. Ludwig   | <ul style="list-style-type: none"> <li>Driver ported to Visual Studio 2017 (Case 27192).</li> </ul>                                                                                                                                                                                                      |
| 1.0.2   | 06/08/2017 | M. Salvador | <ul style="list-style-type: none"> <li>Fixed a problem in case of an OID containing numbers greater than 9 (nine) starting with group's OID (Case 21513).</li> <li>The configuration window does not allow creating variables with their first letter in uppercase anymore. That is mandatory</li> </ul> |



| VERSION      | DATE       | AUTHOR      | COMMENTS                                                                           |
|--------------|------------|-------------|------------------------------------------------------------------------------------|
|              |            |             | by SNMP protocol specification ( <i>Case 19452</i> ).                              |
| <b>1.0.1</b> | 02/14/2017 | M. Salvador | <ul style="list-style-type: none"><li>• Original version of this Driver.</li></ul> |

**Headquarters**

**Rua Mostardeiro, 322/Cj. 902, 1001 e  
1002**

**90510-002 — Porto Alegre — RS**

**Phone: (+55 51) 3346-4699**

**Fax: (+55 51) 3222-6226**

**E-mail: [elipse-rs@elipse.com.br](mailto:elipse-rs@elipse.com.br)**

**Branch in Taiwan**

**9F., No.12, Beiping 2nd St., Sanmin Dist.**

**807 — Kaohsiung City — Taiwan**

**Phone: (+886 7) 323-8468**

**Fax: (+886 7) 323-9656**

**E-mail: [evan@elipse.com.br](mailto:evan@elipse.com.br)**

**Check our website for information about a representative in your country.**

**[www.elipse.com.br](http://www.elipse.com.br)**

**[kb.elipse.com.br](http://kb.elipse.com.br)**

**[forum.elipse.com.br](http://forum.elipse.com.br)**

**[www.youtube.com/elipsesoftware](http://www.youtube.com/elipsesoftware)**

**[elipse@elipse.com.br](mailto:elipse@elipse.com.br)**



Gartner, Cool Vendors in Brazil 2014, April 2014.

Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability of fitness for a particular purpose.

**Microsoft Partner**

Gold Independent Software Vendor (ISV)