Alstom ALSPA S8000E Driver

File Name	S8K.dll
Manufacturer	Alstom
Devices	80-35 and 80-75 ALSPA Controllers
Protocol	S8000E
Version	1.0.17
Last Update	08/22/2025
Platform	Win32
Dependencies	IOKit version 2.0 or later
SuperBlocks reading	No
Level	31307

Introduction

The Alstom ALSPA S8000E Driver supports communication with 80-35 and 80-75 ALSPA Controllers. The **Elipse E3**, **Elipse Power**, or **Elipse Water** system, combined with this Driver, is designed to replace ALSTOM ALSPA system, but keeping the DCS features. An ALSPA System P320 is formed by the following components:

- **ControCAD**: System where Controllers, database, and control logic are configured. The result of this configuration is stored in an Oracle database, which can be accessed by an **Elipse E3**, **Elipse Power**, or **Elipse Water** application
- **CVS**: Graphical interface server
- **CIS**: Real-time database, performing communication with Controllers. It is usually redundant, with servers 1 (one) and 2 (two)
- **Controllers**: Models 8075, 8035 and MFC3000 are supported

This Driver, together with data models provided by **Elipse E3**, **Elipse Power**, or **Elipse Water** applications, plays the role of CIS stations, while the viewers and screen subsystem plays the role of CVS stations.

Preparing a Device

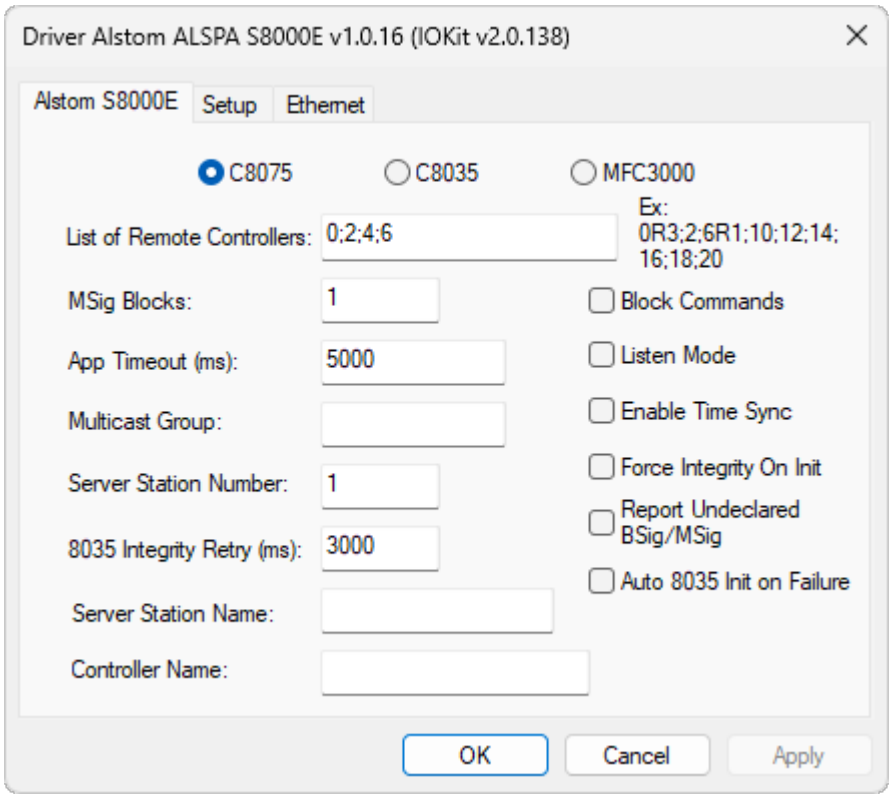
A Driver must be created for each Head of Cell Controller. The I/O racks are configured as remote addresses within the same Driver.

Driver Configuration

This Driver's **[P]** parameters are not used. All configurations are performed on this Driver's configuration dialog box. For more information about **Setup** and **Ethernet** tabs, please check topic **Documentation of I/O Interfaces**.

Configuring Properties

Configuration parameters are defined on the configuration window of this Driver, on the **Alstom S8000E** tab, according to the next figure.



Alstom S8000E tab

The available options on this tab are described on the next table.

Available options on the Alstom S8000E tab

OPTION	DESCRIPTION
C8075, C8035, or MFC3000	Defines the model for the *Head of Cell* Controller
List of Remote Controllers	Declares a list of cell's Controllers, including the *Head of Cell*, with number 0 (zero), and all other remote racks separated by semicolons. Each address can be declared with a special character, as detailed next
MSig Blocks	Number of blocks of **MSig** Tags, 64 **MSig** Tags each, only used for model **8075**
App Timeout (ms)	Default time for a command response, in milliseconds
Multicast Group	This is usually the UDP/IP address 224.0.9.5
Server Station Number	Default CIS address. Leave this option in 0 (zero)
8035 Integrity Retry (ms)	Default time interval, in milliseconds, for a new integrity request for Controller **8035**, in case of not receiving one. Default value of this option is 3000
Server Station Name	Name of a station, or **Elipse E3**, **Elipse Power**, or **Elipse Water** server, in the communication with a Controller model **MFC 3000**
Controller Name	Name of a Controller model **MFC 3000**
Block Commands	Blocks sending commands from this Driver
Listen Mode	Blocks sending any message, only processing messages received from Controllers

OPTION	DESCRIPTION
Enable Time Sync	Enables sending clock syncing messages to the multicast addresses 224.0.9.1 and 224.0.9.5, or to the address configured in the **Multicast Group** option
Force Integrity on Init	Enables a request for a digital integrity, **BSig** and **MSig**, when initializing this Driver. The S8000E protocol uses the concept of integrity and changes for digital and discrete signals, **BSig** and **MSig**
Report Undeclared BSig/MSig	Indicates if, when receiving an integrity or change, this Driver must report the value to the point, even though the Tag is not declared by the application. In this case, the value remains in an **Elipse E3**, **Elipse Power**, or **Elipse Water** memory cache and, as soon as a corresponding Tag is created, the value is then sent to this Tag
Auto 8035 Init on Failure	In case of a full restart of Controller **8035**, this Controller must receive an initialization by this Driver. This procedure can be automatic or performed by writing to a Tag. Select this option for this procedure to be automatic. This initialization consists of sending a configuration file to a device

The addresses declared in the **List of Remote Controllers** option can be declared with an optional character, in the following way:

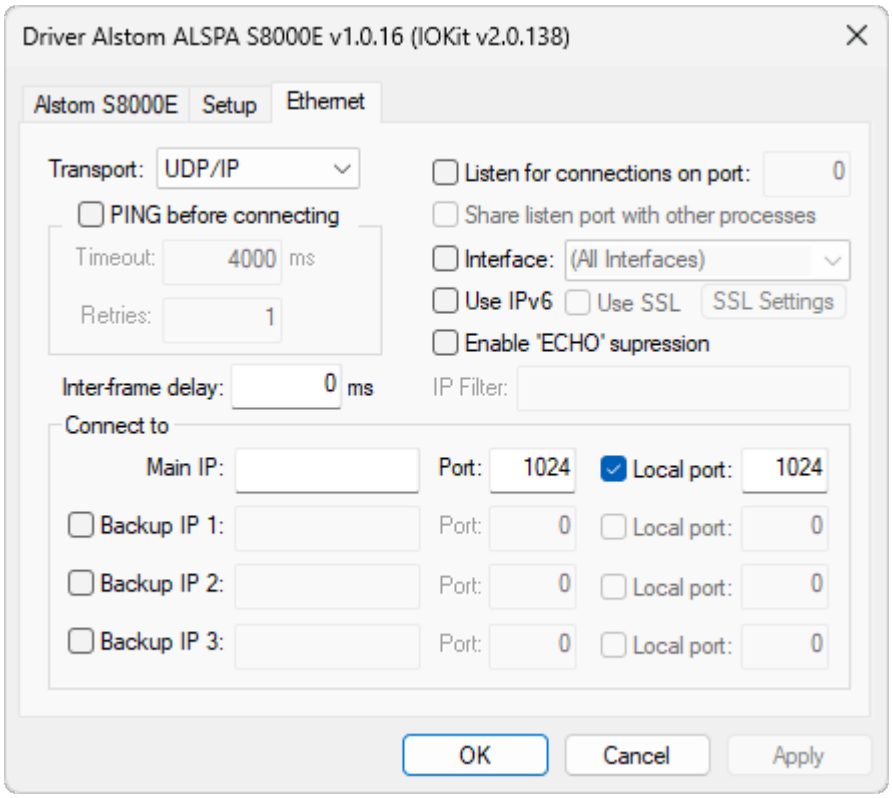
- **R1**: Redundant Configuration 1 (N 00 N 01)
- **R2**: Redundant Configuration 2 (N 00 N 02)
- **R3**: Redundant Configuration 3 (N 00 N 01 N 02 N 03)
- **R4**: Redundant Configuration 4 (N 00 N 01 N 03 N 03 N 04)
- **R5**: Redundant Configuration 5 (N 00 N 01 N 02)
- **R6**: Redundant Configuration 6 (N 00 N 01 N 03 N 03 N 04 N 05)
- **R8**: Redundant Configuration 8 (N 00 N 01 N 03 N 03 N 04 N 05 N 06 N 07 N 08)
- **No additional character**: No redundancy information (N 00)

An example of configuration is displayed on the **Alstom S8000E** tab. This configuration is used to request a data integrity when initializing this Driver, and it must be compatible with a system's real configuration.

These files are generated on ControCAD's FTP server, on folder **C:\Inetpub\ftproot\Echange\res\ProjectName\reference\Res5**, in the format **SRVNNN.CNF**, where *NNN* is the index of the Controller's CPU, such as SRV001.CNF.

These files must be renamed with the IP address of each CPU, using an underscore (_) character to replace the dot character, such as "195_3_0_1.CNF". These files must be placed on the same directory of this Driver's library file. To send this configuration using a Tag, create a Tag with its **Item** parameter equal to "Init8035,U8,0" and perform a writing operation, by sending any value.

Ethernet Tab



Ethernet tab

On the **Ethernet** tab, select in the **Interface** option the network interface in the same range of IP addresses of Controllers. This Driver always starts internally forcing the **Transport**, **Listen for connections on port**, and **Share listen port with other processes** options. The listening port must be correctly configured.

Models C8035 and C8075

On the **Connect to** group, inform the main and backup IP addresses, if available, of the Head of Cell remote station, as well as the origin and destination UDP/IP ports with the value 1024. The listening port must also be equal to 1024.

This Driver processes all multicast messages whose byte with address S8000 available in the message corresponds to the end of the main or backup IP address. According to the configuration of the S8000 network, the IP address of the backup CPU must have an IP address identical to the main IP address, except by the last octet, to which users must add the value 64. For example, if the main IP address is 195.5.0.38, the backup IP address must be 195.5.0.102 (38 + 64). Therefore, all messages are processed for the IP addresses whose last octets are 38 and 102, handled as redundants.

Model MFC3000

On the **Connect to** group, inform the main and backup IP addresses, if available, of the Head of Cell remote station, as well as the destination UDP/IP port with the value 2002 and the origin port with value 2001. The listening port must also be equal to 2001.

This Driver processes all multicast messages whose byte with address S8000 available in the message corresponds to the end of the main or backup IP address.

Other Important Information

- **Time Multicast Address**: 224.0.9.1 (Fixed)
- **Commands for model MFC3000**: Always sent through controller's UDP/IP port 2002
- **Commands for models C8035 and C8075**: Always sent through controller's UDP/IP port 1024

Tag Reference

Tags are configured using the **Device** and **Item** parameters. The **Device** parameter indicates the address of a Controller on the S8000E network. If a Tag represents data coming from a main Controller, *Head Of Cell*, the **Device** parameter must be equal to 0 (zero). If a Tag represents direct data from a remote station, it must have the address of that remote station, as declared in the **List of Remote Controllers** option. The **Item** parameter indicates the address of a Tag and it is composed of the parts indicated next, separated by commas.

Area

Indicates a data area. The available options are described on the next table.

Available options for Area

NAME	DESCRIPTION	SYMBOL
ASig	Analog variables	A
BSig	Digital variables	B
MSig	Discrete variables	M
Setpoint	Setpoint, CReal, or Integer	SP
Command	Commands	COs, CCs, COd, or CCd
Status	Hardware status	ST
Integrity	Requests integrity	Integrity
Init8035	Requests initialization of 8035 Controller	Init8035

Data Type

Indicates the size and type of an information. The available options are described on the next table.

Available options for Data Type

NAME OR SYMBOL	DESCRIPTION
U8	Unsigned byte, used for **BSig**, **MSig**, and **ST** Tags
I8	8-bit signed integer
U16	Word
I16	16-bit signed integer
U32	32-bit unsigned integer
I32	32-bit signed integer
R32	32-bit real number

Address

Address of a variable inside an area. The size of an area depends on the allocation performed in ControCAD. The available options are described on the next table.

Available options for Address

AREA	ADDRESS
ASig	For the *Head of Cell* Controller of models **8075** and **8035**, values are allocated in linear addresses in **Word** format, such as "A,R32,0" (first **Float**), "A,R32,2" (second **Float**), and so on. For remote stations, addressing is always in **Word** format, considering the rule *slot * 100 + position*, such as "A,U16,1000" (input 0 of slot 10), "A,U16,1001" (input 1 of slot 10), or "A,U16,1101" (input 1 of slot 11)
BSig	For models **8075** and **8035**, values are allocated in linear addresses per bit. For the *Head of Cell* Controller, values are allocated in continuous addresses starting from 0 (zero), such as "B,U8,123" (position 123). For remote stations, physical points are allocated continuously starting from address 0 (zero). Calculated points remain in their own area and users must add 2000 to the position, such as in address "B,U8,25" (position 25) the value must be equal to "B,U8,2003" (position 3, area TSX)
MSig	Allocated in linear addresses per byte. There is **MSig** only in the *Head of Cell*, such as M,U8,30 -> MSig 30
SP	Linear address that can have a **U16** or **R32** data type
COs	**Open Loop Single** pulse command. Does not use Tag's value to perform the command. It must have an **U8** data type. Linear address starts at 0 (zero)
CCs	**Closed Loop Single** pulse command. Does not use Tag's value to perform the command. It must have an **U8** data type. Linear address starts at 0 (zero)
COd	**Open Loop** double command. Uses the value of Tag 0 (zero) or 1 (one) to define which command to send. It must have an **U8** data type. Linear address starts at 0 (zero)
CCd	**Open Loop** double command. Uses the value of Tag 0 (zero) or 1 (um) to define which command to send. It must have an **U8** data type. Linear address starts at 0 (zero)

ST	The following areas are declared:
	• Head Of Cell - Main Controller
	  • ST,U8,80000 to 80127 - %S Area (16 bytes, 128 bits)
	  • ST,U8,81000 to 81127 - %SA Area (16 bytes, 128 bits)
	  • ST,U8,82000 to 82127 - %SB Area (16 bytes, 128 bits)
	  • ST,U8,83000 to 83127 - %SC Area (16 bytes, 128 bits)
	  • ST,U8,10000 to 10015 - Main Status (Original Controller data, 2 bytes, 16 bits)
	  • ST,U8,0 to ST,U8,21 - (22 bits) S8000 Overall Status (Calculated)
	• Head Of Cell - Secondary Controller
	  • ST,U8,90000 to 90127 %S Area (16 bytes, 128 bits)
	  • ST,U8,91000 to 91127 %SA Area (16 bytes, 128 bits)
	  • ST,U8,92000 to 92127 %SB Area (16 bytes, 128 bits)
	  • ST,U8,93000 to 93127 %SC Area (16 bytes, 128 bits)

<table>
<tr><th>AREA</th><th>ADDRESS</th></tr>
<tr><td></td><td>

ST,U8,20000 to 20001 - Main Status (Original Controller data, 2 bytes, 16 bits)
Remote Station

ST,U8,0 to ST,U8,17 - (18 bits) Remote Overall Status (Calculated)
ST,U8,1000 to ST,U8,1071 - Main Status (9 bytes)
ST,U8,2000 to ST,U8,2127 - Rack Status (16 card positions - 16 bytes, 127 bits)

Card Status:

<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>Present</td><td></td><td>Failure</td></tr>
</table>
</td></tr>
<tr><td>Integrity</td><td>Integrity,U8,0</td></tr>
<tr><td>Init8035</td><td>Init8035,U8,0</td></tr>
</table>

Quality Information

ASig Tags contain the following quality information:

- **NORMAL**: GOOD 192
- **INVALID**: BAD 20
- **LINE BREAK**: BAD, NOT CONNECTED 28
- **FORCED VALUE**: GOOD, OVERRIDE: 216
- **LOW LIMIT**: Adds the value 1 (one), such as 192 + 1 = 193
- **HIGH LIMIT**: Adds the value 2 (two), such as 192 + 2 = 194

BSig Tags contain the following quality information:

- **NORMAL**: GOOD 192
- **INVALID**: BAD 20
- **FORCED VALUE**: GOOD, OVERRIDE: 216

MSig Tags only inform validity:

- **NORMAL**: GOOD 192
- **INVALID**: BAD 20

Documentation of I/O Interfaces

This section contains the documentation of I/O Interfaces referring to the **S8K** 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 **Elipse 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 **Elipse E3** version 2.0 or later, click **Configure driver** on a Driver's toolbar. In **Elipse 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.

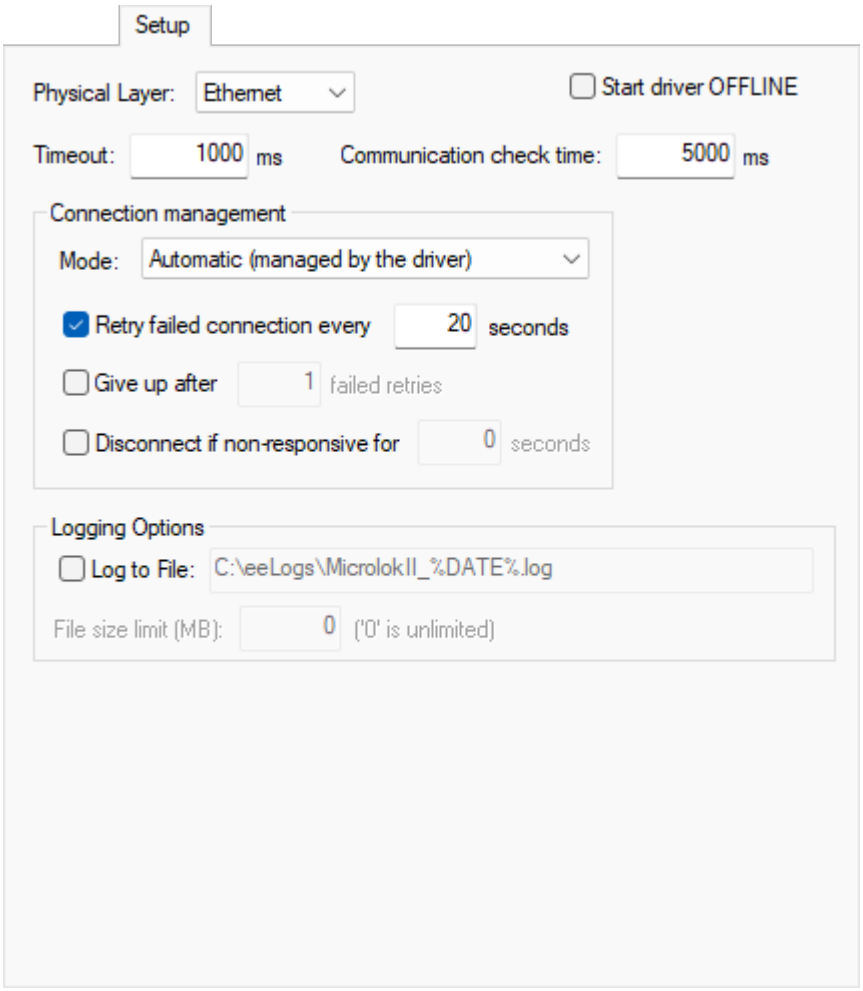
Configuration Dialog Box

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



Setup tab

General options on the Setup tab

OPTION	DESCRIPTION
Physical Layer	Select the physical layer on a list. Available options are **Serial**, **Ethernet**, **Modem**, and **RAS**. The selected interface must be configured on its specific tab
Timeout	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
Communication check time	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 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 **IO.CommunicationStatus** Tag
Start driver OFFLINE	Select this option so that a Driver starts in **Offline** mode or stopped. This means that the I/O interface is not created until this Driver is configured to **Online** 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

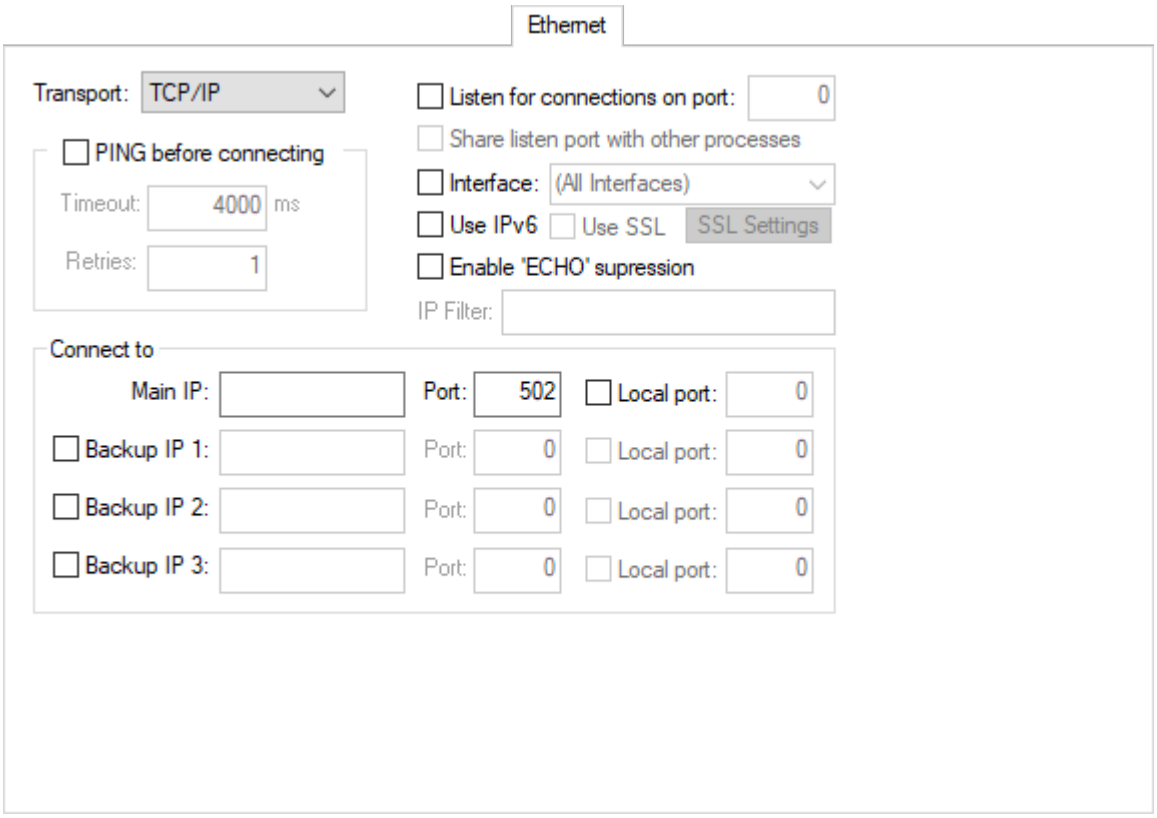
OPTION	DESCRIPTION
Mode	Selects a management mode of a connection. Selecting the **Automatic** option allows a Driver to manage the connection automatically, as specified in the next options. Selecting the **Manual** option allows an application to fully manage a connection
Retry failed connection every ... seconds	Select this option to enable a Driver's connection retry in a certain interval, in seconds. If the **Give up after failed retries** option is not selected, this Driver keeps retrying until a connection is performed, or until the application is stopped
Give up after ... failed retries	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 **Offline** 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
Disconnect if non-responsive for ... seconds	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 **Timeout** option

Options on the Logging Options group

OPTION	DESCRIPTION
Log to File	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 **%PROCESS%** 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 **Elipse E3**, 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 **0FDAh**. Users can also use the **%DATE%** macro in the file name. In this case a log file is generated every day, in the format **aaaa_mm_dd**. 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 **%DATE_HOUR%** macro generates one log file per hour, in the format **aaaa_mm_dd_hh**
File size limit (MB)	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

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 tab

Available options on the Ethernet tab

OPTION	DESCRIPTION
Transport	Select the value **TCP/IP** for a TCP socket (*stream*) or select the value **UDP/IP** to use a UDP socket (*connectionless datagram*)
Listen for connections on port	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 **Connect to** option
Share listen port with other processes	Select this option to share the listening port with other Drivers and processes
Interface	Select the local network interface, identified by its IP address, that a Driver uses to establish and receive connections, or select the value **(All Interfaces)** to allow connection in any network interface
Use IPv6	Select this option to force a Driver to use addresses in **IPv6** format on all Ethernet connections. Leave this option deselected to use the **IPv4** format
Enable 'ECHO' suppression	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
IP Filter	List of restricted or allowed IP addresses from where a Driver accepts connections (*Firewall*). Please check the **IO.Ethernet.IPFilter** property for more information
PING before connecting	Enable this option to execute a **ping** 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
	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: • **Timeout**: Specify the number of milliseconds to wait for a reply from a **ping** command. Users must use a **ping** 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 • **Retries**: Number of retries of a **ping** command, not counting the first attempt. If all attempts fail, then the socket connection is aborted

Available options on the Connect to group

OPTION	DESCRIPTION
Main IP	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"
Port	Type the IP port of a remote device, between 0 (zero) and 65535
Local port	Select this option to use a fixed local IP port when connecting to a remote device
Backup IP 1, 2, and 3	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

Type of Tag	I/O Tag
Type of Access	Reading
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	6 (six)
String Configuration	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 vlaue **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 **Elipse 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 × 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 **Elipse 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

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

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

IO.Type

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

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	1101
Configuration by String	IO.Stats.Partial.BytesRecv

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

IO.Stats.Partial.BytesSent

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	1100
Configuration by String	IO.Stats.Partial.BytesSent

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

IO.Stats.Partial.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	1102
Configuration by String	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

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	1103
Configuration by String	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

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	1001
Configuration by String	IO.Stats.Total.BytesRecv

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

IO.Stats.Total.BytesSent

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	1000
Configuration by String	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]

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]

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:*:*)**: Accepts connections from IPv6 addresses in the interval between fe80:03bf:0877:0000:0000:0000:0000:0000 and fe80:03bf:0877:0000:0000:0000: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

A 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

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

IO.Ethernet.MainIP

A 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

9 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

9 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

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

IO.Ethernet.PingTries

9 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
1.0.17	08/22/2025	M. Ludwig	• Driver updated to **IOKit** library version **3.0** and Visual Studio 2022 (*Case 37933*).
1.0.12	10/06/2020	M. Salvador	• Added support for Controller MFC3000 (*Case 27585*). • The timestamp's quality is now being processed (*Case 29212*). • The **App Timeout** option is now used to indicate bad quality if messages are not received within the informed interval (*Case 29210*). • Added support for IHR rack and event handling (*Case 29169*).
1.0.11	07/27/2020	C. Mello	• Platform update on this Driver's source code (*Case 27425*).
		M. Salvador	• Added an **L2** option to initialize model **8035** and a new field to indicate how many **MSig** blocks are available in model **8075** (*Case 27398*).

VERSION	DATE	AUTHOR	COMMENTS
1.0.7	04/23/2019	M. Salvador	• Adjustments on communication initialization for model **8035** (*Case 26248*).
1.0.4	08/31/2018	M. Salvador	• Adjustments on sending commands and setpoints with a **Type L** data type for model **8035** (*Case 24045*).
1.0.2	05/25/2017	M. Salvador	• Initial version of this Driver (*Case 17925*).

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

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

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

www.elipse.com.br

kb.elipse.com.br
forum.elipse.com.br
www.youtube.com/elipsesoftware
elipse@elipse.com.br

Gartner 2014 Cool Vendor

OPC FOUNDATION MEMBER

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)