

CPI CIF Driver

File Name	CIF.dll
Manufacturer	Communications & Power Industries (CPI)
Devices	Remote Control Panel
Protocol	Computer Interface (CIF)
Version	1.0.2
Last Update	08/26/2025
Platform	Win32
Dependencies	IOKit version 2.0 or later
Superblock Readings	No
Level	0

Introduction

This Driver implements the Computer Interface (CIF) protocol, allowing an application developed by **Elipse Software** to communicate with Remote Control Panel devices by Communications & Power Industries (CPI).

Preparing a Device

HPA Amplifiers must be connected to a Remote Control Panel device, which must be then connected to a server via Ethernet.

Driver Configuration

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

Configuring Properties

Configuration parameters to communicate with devices are available on this Driver's properties window. In addition to this properties window, these settings can also be defined at run time in **Elipse E3**, **Elipse Power**, or **Elipse Water** applications. To do so, initialize this Driver in **Offline** mode, that is, execute an application with the **Start driver OFFLINE** option enabled, which is configured on the **Setup** tab of the properties window. The configuration options of this Driver are described on the next table.

Configuration options for CPI CIF Driver

TAB	PARAMETER	OFFLINE STRING	DATA TYPE	DESCRIPTION
CIF	Header/Ending Option	CIF.Header/Ending	Number	Defines a format for bytes of message headers and endings. Valid values for this option are 0 : STX/ETX - STX/ACK/NAK on header and ETX on ending or 1 : ASCII - '{'

TAB	PARAMETER	OFFLINE STRING	DATA TYPE	DESCRIPTION
				on header and '}' on ending
	Check Option	CIF.Check	Number	Defines a method to calculate a byte check. Valid values for this option are 0 : Parity or 1 : Checksum. The Checksum option can only be used if the Header/Ending option is configured as 1 (one, ASCII)
	Append Options	CIF.Append.CR CIF.Append.LF	Number	Enables or disables appending Carriage Return (CR) or Line Feed (LF) bytes at the end of each message sent. Valid values for this option are 0 : Disabled or 1 : Enabled
Setup	Physical Layer	IO.Type	Text	Physical layer. By default, use the Ethernet option
	Timeout	IO.TimeoutMs	Number	A time limit, in milliseconds, to receive data from a device's response. For example, the value 1000 defines a limit of 1 (one) second
Ethernet	Transport	IO.Ethernet.Transport	Text	Transport layer. By default, use the TCP option
	Main IP	IO.Ethernet.MainIP	Text	IP address of a device, in the format [0-255].[0-255].[0-255].[0-255]
	Port	IO.Ethernet.MainPort	Number	TCP/IP addressing port

All properties can be configured via Tags in **String** format, by using the syntactic parameter *Item* with "IO.SetConfigurationParameters" or the *N1* parameter equal to -1 (minus one), *N2* equal to 0 (zero), *N3* equal to 0 (zero), and *N4* equal to 3 (three). For more details and examples, please check topic **Documentation of I/O Interfaces**.

Tag Reference

This section contains information about the configuration of I/O Tags and Block Tags via **Syntactic Parameters** or **Numeric Parameters (N or B)**. If both formats are used at the same time, syntactic parameters overwrite numeric parameters.

In **Elipse E3**, **Elipse Power**, or **Elipse Water**, users can retrieve pre-configured I/O Tags and Block Tags by using the Tag Browser window. All commands are available with the syntactic parameter *Item* filled in and the corresponding names of I/O Tags, Block Tags, and Block Elements.

Configuration by Syntactic Parameters

Use the next syntax to configure I/O Tags and Block Tags using syntactic parameters.

- **Device:** Address of a device to which a Tag communicates. It must be a value between 48 and 111
- **Item:** This text must follow the next syntax, in which items are separated by a single space

```
<Type> <Command>[ <Parameter>]
```

In which:

- **Type:** Type of a command. The available options are **Set:** Writing or action commands or **Query:** Reading commands
- **Command:** A single character identifying an operation to perform. For more information, please check topic **Commands**
- **Parameter:** Additional numeric parameters. For a **c** command, identifies the log entry number to read and, for all other commands, it must be ignored

NOTE

Writing commands must use the **Set** type and reading commands must use the **Query** type. If a command does not match its type, this Driver indicates a syntax error.

Examples of using syntactic parameters

DEVICE	ITEM
48	Set A
48	Set ~
48	Query 0
48	Query c 1234

Configuration by Numeric Parameters

Use the syntax described on the next table to configure I/O Tags and Block Tags using numeric parameters.

Configuration by numeric parameters (N or B)

PARAMETER	DESCRIPTION	RANGE
N1 or B1	Address of a device to which a Tag communicates	From 48 to 111
N2 or B2	Command type. Possible values are 0: Set, writing commands or actions, or 1: Query, reading commands	0 (zero) or 1 (one)
N3 or B3	Command. Decimal value of the character that identifies a command, according to ASCII encoding. For more	From 32 to 127

PARAMETER	DESCRIPTION	RANGE
	information, please check topic Commands	
N4 or B4	Additional parameter. For the c command, identifies the log entry number to read. For all other commands, it must receive 0 (zero)	From 0 (zero) to 9999

NOTE

Writing commands must have their *N2* or *B2* parameter equal to 0 (zero) and reading commands must have their *N2* or *B2* parameter equal to 1 (one). If a command does not match its type, this Driver indicates an error in the parameters.

Examples of using numeric parameters (N or B)

NUMERIC PARAMETER (N/B)				EQUIVALENT SYNTACTIC PARAMETER	
N1 OR B1	N2 OR B2	N3 OR B3	N4 OR B4	DEVICE	ITEM
48	0	65	0	48	Set A
48	0	126	0	48	Set ~
48	1	48	0	48	Query 0
48	1	99	1234	48	Query c 1234

Commands

This topic presents all commands supported by this Driver, separated into **Writing** and **Reading** commands.

Writing Commads

The next table shows all writing commands and actions supported by this Driver. Please check topic **Tag Reference** for instructions on how to configure an I/O Tag or a Block Tag with a given command. For detailed information on commands, please refer to a device's manufacturer manual.

NOTES

- An empty size indicates that a command must be used with an I/O Tag. A value for the size indicates that a command must be used with a Block Tag with the given size.
- An empty data type indicates that a command does not send values. In this case, writing any value to that I/O Tag triggers a communication.

Writing commands and actions

COMMAND CHARACTER	ASCII ENCODING	SIZE	COMMAND	ELEMENTS	DATA TYPE	DESCRIPTION
+ (plus)	43		Set CIF Word Length		Number	0: 7-bit or 1: 8-bit
@ (at)	64		Transmit Request			

COMMAND CHARACTER	ASCII ENCODING	SIZE	COMMAND	ELEMENTS	DATA TYPE	DESCRIPTION
A	65		Standby			
B	66		Reset Faults			
C	67		Set Attenuator (dB)		Number	From 0 (zero) to 99.9
D	68		Set Manual RF Output Power (dBm)		Number	From 0 (zero) to 99.9
E	69		Set Manual RF Output Power (W)		Number	From 0 (zero) to 9999
F	70		Set ALC RF Output Power (dBm)		Number	From 0 (zero) to 99.9
G	71		Set ALC RF Output Power (W)		Number	From 0 (zero) to 9999
H	72		Disable ALC			
I	73		Enable Computer Interface RF Inhibit			
J	74		Disable Computer Interface RF Inhibit			
L	76		Set Low RF Alarm Trip Point (W)		Number	From 0 (zero) to 9999
M	77		Set High RF Alarm Trip Point (W)		Number	From 0 (zero) to 9999
N	78		Set Low RF Fault Trip Point (W)		Number	From 0 (zero) to 9999
O	79		Set High RF Fault Trip Point (W)		Number	From 0 (zero) to 9999
P	80		Set Auto Log Time (M)		Number	From 0 (zero) to 9999.9
T	84		Set High Reflected RF Fault Trip Point (W)		Number	From 0 (zero) to 9999
U	85		Set High Helix Over Current		Number	From 0 (zero) to 99.9

COMMAND CHARACTER	ASCII ENCODING	SIZE	COMMAND	ELEMENTS	DATA TYPE	DESCRIPTION
			Fault Trip Point (mA)			
V	86	3	Set Linearizer. For more information, please check the next note	Linearizer Gain Offset	Number	From 0 (zero) to 255
				Linearizer Phase Offset	Number	From 0 (zero) to 255
				Linearizer Magnitude	Number	From 0 (zero) to 255
Y	89		Set Helix Over Voltage Fault Trip Point (kV)		Number	From 0 (zero) to 99.99
Z	90		Set Helix Under Voltage Fault Trip Point (kV)		Number	From 0 (zero) to 99.99
a	97		Reset To Defaults			
b	98		Set W/G Switch Preset Configuration		Number	From 0 (zero) to 3 (três)
d	100		Reset Meter Log			
e	101		Set CIF CR Enable		Number	0 : Disabled or 1 : Enabled
f	102		Set CIF LF Enable		Number	0 : Disabled or 1 : Enabled
g	103		Set CIF Header/Ending		Number	0 : ASCII or 1 : STX/ETX
h	104		Set CIF Baud Rate		Number	0 : 600, 1 : 1200, 2 : 2400, 3 : 4800, 4 : 9600, or 5 : 19200
i	105		Set CIF Parity		Number	0 : No parity, 1 : Even, or 2 : Odd
j	106		Set CIF Unit Address		Text / Number	From '0' to 'Z'
k	107		Lamp Test			
q	113		Set Helix Voltage Detected during Standby or HTD State Fault Trip Point (kV)		Number	From 0 (zero) to 99.99
s	115		Set Time		Date/Time	

COMMAND CHARACTER	ASCII ENCODING	SIZE	COMMAND	ELEMENTS	DATA TYPE	DESCRIPTION
t	116		Set Switch Controller Mode		Number	0 : Manual or 1 : Auto
u	117	2	Set W/G Switch Position	W/G Switch	Number	0 : No switch expander board installed and from 1 to 4 : Switch on the expander board
				Position	Number	1 or 2
v	118		Heater Control		Number	0 : Disabled or 1 : Enabled
y	121		Set Heater Under Current Fault Trip Point (A)		Number	From 0 (zero) to 9.99
z	122		Set Heater Over Current Fault Trip Point (A)		Number	From 0 (zero) to 9.99
~ (tilde)	126	15	Configuration Command	Relay 1 Flexible	Number	
				Relay 1 HV On	Number	
				Relay 1 Latched Fault	Number	
				Relay 1 RF Enabled	Number	
				Relay 1 Rigid	Number	
				Relay 1 Sum Fault	Number	
				Relay 2 Flexible	Number	
				Relay 2 HV On	Number	
				Relay 2 Latched Fault	Number	
				Relay 2 RF Enabled	Number	
				Relay 2 Rigid	Number	
				Relay 2 Sum Fault	Number	
				Amplifier Ident	Number	0 : Amplifier 1, 1 : Amplifier 2, or 2 : Amplifier 3
				Priority Amplifier	Number	0 : None, 1 : Amplifier 1, or

COMMAND CHARACTER	ASCII ENCODING	SIZE	COMMAND	ELEMENTS	DATA TYPE	DESCRIPTION
						2: Amplifier 2
				BUC Unlock Type	Number	0: Alarm or 1: Fault

NOTE

Linearizer offsets are added to factory presets. A value of 0 (zero) represents an actual offset of -128. A value of 128 represents an actual offset of 0 (zero). A value of 255 represents an actual offset of 127.

Reading Commands

The next table shows all reading commands supported by this Driver. Please check topic **Tag Reference** for instructions on how to configure an I/O Tag or a Block Tag with a given command. For detailed information on commands, please refer to the device's manufacturer manual.

NOTE

An empty size indicates that a command must be used with an I/O Tag. A value for the size indicates that a command must be used with a Block Tag with the given size.

Reading commands

COMMAND CHARACTER	ASCII ENCODING	SIZE	COMMAND	ELEMENTS	DATA TYPE	DESCRIPTION
0	48	2	ID Version Query	Boot Kernel Firmware Version	Text	
				Main Firmware Version	Text	
1	49	8	Summary Status Query	ALC Enabled	Number	
				W/G Switch Controller in Auto Mode	Number	
				Fault/Misc. Status Data Check	Number	
				Secondary Status Data Check	Number	
				System State. For more information, please check table Summary Status Query - System State	Number	

COMMAND CHARACTER	ASCII ENCODING	SIZE	COMMAND	ELEMENTS	DATA TYPE	DESCRIPTION
				Latched Fault	Number	
				Transmit Request	Number	
				RF Inhibit	Number	
2	50	15	Secondary Status Query	Low RF Alarm	Number	
				High RF Alarm	Number	
				BUC Alarm	Number	
				Heater Timeout Alarm	Number	
				Tube Over Temperature Alarm	Number	
				CIF Control Point	Number	
				W/G Switch 4 Position	Number	
				Heater On	Number	
				Power Sensor Failure	Number	
				W/G Switch 1 Position	Number	
				W/G Switch 2 Position	Number	
				W/G Switch 3 Position	Number	
				Other Amplifier in Fault	Number	
				W/G Switch Stall Alarm	Number	
W/G Switch Preset	Number					
3	51	25	Fault/Misc. Query	Helix Voltage Detected during HTD or Standby State	Number	
				Power Supply Arc Fault	Number	
				Helix Over Current Fault	Number	
				Helix Under Voltage Fault	Number	
				Helix Over Voltage Fault	Number	

COMMAND CHARACTER	ASCII ENCODING	SIZE	COMMAND	ELEMENTS	DATA TYPE	DESCRIPTION
				External Interlock Fault	Number	
				Low RF Fault	Number	
				High RF Fault	Number	
				Tube Switch Over Temp Fault	Number	
				Cover Interlock Fault	Number	
				High Reflected RF Fault	Number	
				SSIPPA Over Current Fault	Number	
				Tube Over Temperature Fault	Number	
				Heater Timeout Fault	Number	
				BBRAM Data Invalid Fault	Number	
				Internal Reflected RF Fault	Number	
				Fan Stalled Fault	Number	
				Heater Over Current Fault	Number	
				Heater Under Current Fault	Number	
				Cabinet Over Temperature Fault	Number	
				CIF Inhibit	Number	
				External Interlock Inhibit	Number	
				W/G Switch Inhibit	Number	
				Heater Interlock Fault	Number	
				BUC Fault	Number	
6	54		RF Output Power Query (dBm)		Number	

COMMAND CHARACTER	ASCII ENCODING	SIZE	COMMAND	ELEMENTS	DATA TYPE	DESCRIPTION
7	55		RF Output Power Query (W)		Number	
8	56		Attenuator Setting Query (dB)		Number	
: (colon)	58		Low RF Alarm Trip Point Query (W)		Number	
; (semicolon)	59		High RF Alarm Trip Point Query (W)		Number	
< (less)	60		Low RF Fault Trip Point Query (W)		Number	
= (equal)	61		High RF Fault Trip Point Query (W)		Number	
> (greater)	62	10	Meter Readings Query	Attenuator (dB)	Number	
				Cabinet Temperature (°C)	Number	
				Fan Control Voltage (V)	Number	
				Heater Current (A)	Number	
				Helix Current (mA)	Number	
				Helix Voltage (kV)	Number	
				Reflected RF (W)	Number	
				RF Output (dBm)	Number	
				RF Output (W)	Number	
				Tube Temperature (°C)	Number	
? (question mark)	63	6	Time Query	Date/Time	Date and time	
				Heater Time Delay (s)	Number	
				Standby Elapsed Time (s)	Number	

COMMAND CHARACTER	ASCII ENCODING	SIZE	COMMAND	ELEMENTS	DATA TYPE	DESCRIPTION
				Transmit Elapsed Time (s)	Number	
				Unit On Elapsed Time (s)	Number	
				Heater Off Time (s)	Number	
Q	81		Auto Log Time Query (M)		Number	
W	87	6	Linearizer Query. For more information, please check the next notes	Linearizer Gain Offset	Number	
				Linearizer Phase Offset	Number	
				Linearizer Magnitude Offset	Number	
				Linearizer Gain	Number	
				Linearizer Phase	Number	
				Linearizer Magnitude	Number	
c	99	12	Meter Log Entry Query. For more information, please check the next notes	Entry Date/Time	Date and time	
				Activity Type. For more information, please check table Meter Log Entry Query - Activity Type and Condition	Number	
				Activity Condition. For more information, please check table Meter Log Entry Query - Activity Type and Condition	Number	
				Attenuation (dB)	Number	
				Cabinet Temperature (°C)	Number	

COMMAND CHARACTER	ASCII ENCODING	SIZE	COMMAND	ELEMENTS	DATA TYPE	DESCRIPTION
				Fan Control Voltage (V)	Number	
				Heater Current (A)	Number	
				Helix Current (mA)	Number	
				Helix Voltage (kV)	Number	
				Reflected RF Power (W)	Number	
				RF Output Power (W)	Number	
				Tube Temperature (°C)	Number	
m	109		Helix Under Voltage Fault Trip Point Query (kV)		Number	
n	110		Helix Over Voltage Fault Trip Point Query (kV)		Number	
o	111		Helix Over Current Fault Trip Point Query (mA)		Number	
p	112		Helix Voltage Detected during Standby or HTD State Fault Trip Point Query (kV)		Number	
r	114		High Reflected RF Fault Trip Point Query (W)		Number	
w	119		RF Output Set Point Query (W)		Number	
x	120	10	Settings Query	High RF Alarm Trip Point (W)	Number	
				Low RF Alarm Trip Point (W)	Number	
				High Tube Temperature Alarm Trip Point (°C)	Number	

COMMAND CHARACTER	ASCII ENCODING	SIZE	COMMAND	ELEMENTS	DATA TYPE	DESCRIPTION
				High Cabinet Temperature Fault Trip Point (°C)	Number	
				Heater Over Current Fault Trip Point (A)	Number	
				Heater Under Current Fault Trip Point (A)	Number	
				Helix Over Current Fault Trip Point (mA)	Number	
				Helix Over Voltage Fault Trip Point (kV)	Number	
				Helix Under Voltage Fault Trip Point (kV)	Number	
				Helix Voltage Detected During Standby or HTD Fault Trip Point (kV)	Number	
(vertical bar)	124		Configuration Query	Relay 1 Flexible	Number	
				Relay 1 HV On	Number	
				Relay 1 Latched Fault	Number	
				Relay 1 RF Enabled	Number	
				Relay 1 Rigid	Number	
				Relay 1 Sum Fault	Number	
				Relay 2 Flexible	Number	
				Relay 2 HV On	Number	
				Relay 2 Latched Fault	Number	
				Relay 2 RF Enabled	Number	
				Relay 2 Rigid	Number	
				Relay 2 Sum Fault	Number	
Attenuator Installed	Number					

COMMAND CHARACTER	ASCII ENCODING	SIZE	COMMAND	ELEMENTS	DATA TYPE	DESCRIPTION
				BUC Installed	Number	
				IPA Installed	Number	
				Linearizer Installed	Number	
				PS Test Mode	Number	
				Switch Sytem Installed	Number	
				RF Monitoring Installed	Number	
				Switch Expander Board Installed	Number	
				Amplifier Ident	Number	0: Amplifier 1, 1: Amplifier 2, or 2: Amplifier 3
				Priority Amplifier	Number	0: None, 1: Amplifier 1, or 2: Amplifier 2
				Switch System Configuration	Number	0: 1:1 Power Combiner, 1: 1:2A, 2: 1:2B, 3: 1:1D, or 4: 1:1E
				BUC Unlock Type	Number	0: Alarm or 1: Fault

NOTES

- Linearizer offsets are added to factory presets. A value of 0 (zero) represents an actual offset of -128. A value of 128 represents an actual offset of 0 (zero). A value of 255 represents an actual offset of 127. *Gain Offset*, *Phase Offset*, and *Magnitude Offset* represent user settings. *Gain*, *Phase*, and *Magnitude* represent actual device settings.
- The **Meter Log Entry Query** command requires a log entry number to read. If using syntactic parameters, this value must be at the end of the *Item* parameter. If using numeric parameters (*N* or *B*), this value must be in the *N4* or *B4* parameter. Valid values are in the range from 0 (zero) to 9999.

Values and Meanings

The following tables show the meaning of numeric values that appear in reading commands.

Summary Status Query - System State

VALUE	SYSTEM STATE
0	Power On State
1	Heater Time Delay State
2	Standby State

VALUE	SYSTEM STATE
3	Beam On Sequence State
4	Transmit State
5	Transmit State, RF Inhibited
6	Beam Off Sequence State
7	Fault State

Meter Log Entry Query - Activity Type and Condition

ACTIVITY TYPE VALUE	ACTIVITY TYPE	ACTIVITY CONDITION VALUE	ACTIVITY CONDITION
0	No Activity		
1	Alarm	0	BUC Alarm
		1	Heater Timeout Alarm
		2	High RF Alarm
		3	Low RF Alarm
		4	Tube Over Temperature Alarm
		5	Power Sensor Failure Alarm
		6	W/G Switch Stall Alarm
2	Fault	0	BBRAM Data Invalid Fault
		1	Helix Over Voltage Fault
		2	Helix Under Voltage Fault
		3	Helix Voltage Detected during HTD or Standby State
		4	Helix Over Current Fault
		5	Cover Interlock Fault
		6	DC Buss Fault
		7	External Interlock Fault
		8	Heater Timeout Fault
		9	High RF Fault
		10	Low RF Fault
		11	High Reflected RF Fault
		12	Power Supply Arc Fault
		13	SSIPPA Over Current Fault
		14	Tube Switch Over Temp Fault
		15	Tube Over Temperature Fault
16	Internal Reflected RF Fault		

ACTIVITY TYPE VALUE	ACTIVITY TYPE	ACTIVITY CONDITION VALUE	ACTIVITY CONDITION
		17	Fan Stalled Fault
		18	Heater Over Current Fault
		19	Heater Under Current Fault
		20	Cabinet Over Temperature Fault
		21	Heater Interlock Fault
		22	BUC Fault
3	Inhibit	0	External Interlock Inhibit
		1	CIF Inhibit
		2	W/G Switch Inhibit
4	Heater Time Display		
5	Power Off		
6	Power On		
7	Standby		
8	Transmit		
9	Pre Event		
10	Beam On Sequence		
11	Warning		
12	Beam Off Sequence		
13	Reset		
14	Auto Log		

Documentation of I/O Interfaces

This section contains the documentation of I/O Interfaces referring to the **CIF** 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

Setup

Physical Layer: Start driver OFFLINE

Timeout: ms Communication check time: ms

Connection management

Mode:

Retry failed connection every seconds

Give up after failed retries

Disconnect if non-responsive for seconds

Logging Options

Log to File:

File size limit (MB): ('0' is unlimited)

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	<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 %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 OFDAh. 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.</p>
File size limit (MB)	<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 ▼

PING before connecting
 Timeout: 4000 ms
 Retries: 1

Listen for connections on port: 0
 Share listen port with other processes
 Interface: (All Interfaces) ▼
 Use IPv6 Use SSL SSL Settings
 Enable 'ECHO' supression
 IP Filter:

Connect to

<input type="checkbox"/> Main IP:	 	Port:	502	<input type="checkbox"/> Local port:	0
<input type="checkbox"/> Backup IP 1:	 	Port:	0	<input type="checkbox"/> Local port:	0
<input type="checkbox"/> Backup IP 2:	 	Port:	0	<input type="checkbox"/> Local port:	0
<input type="checkbox"/> Backup IP 3:	 	Port:	0	<input type="checkbox"/> Local port:	0

Ethernet tab

Available options on the Ethernet tab

OPTION	DESCRIPTION
Transport	Select the value TCP/IP for a TCP socket (<i>stream</i>) or select the value UDP/IP to use a UDP socket (<i>connectionless datagram</i>)
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' supression	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 (<i>Firewall</i>). 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
	<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"> • 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 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, 3×2). 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 **Eclipse 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

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]

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: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
1.0.2	08/26/2025	M. Ludwig	Driver updated to IOKit library version 3.0 and Visual Studio 2022 (Case 37951).
1.0.1	12/19/2019	G. Beal	Initial version of this Driver.

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, 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)