

DNP Mirror Master & Slave Driver

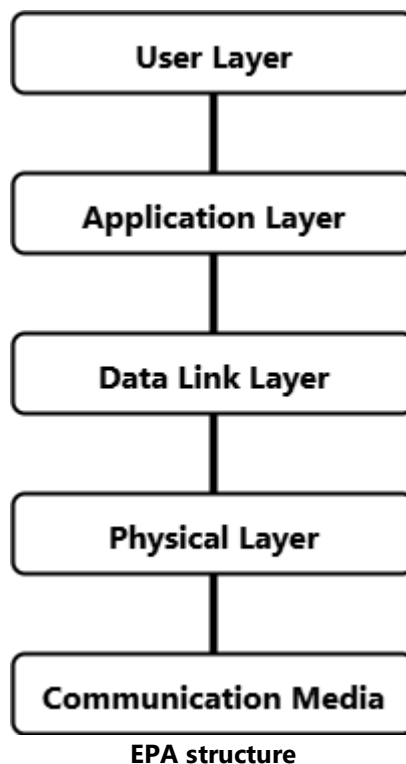
File Name	DNPMirror.dll
Manufacturer	DNP (Distributed Network Protocol)
Devices	Devices compatible with protocol DNP 3.0
Protocol	DNP 3.0
Version	1.0.7
Last Update	09-02-2025
Platform	Win32
Dependencies	IOKit version 2.0 or later
Superblock Readings	No
Level	31201

Introduction to DNP 3.0 Protocol

DNP Mirror Master & Slave Driver implements DNP protocol version **3.0** in **Master** and **Slave** modes, according to levels 2 (two) and 3 (three).

DNP (*Distributed Network Protocol*) is an open, non-proprietary communication protocol, based on specifications by IEC (*International Electrotechnical Commission*), adapted for use with highly secure applications, with a moderated speed and amount of data. It is extremely flexible and can be used on any hardware platform.

The model specified by ISO - OSI (*International Standards Organization - Open System Interconnection*) establishes 7 (seven) layers for a network protocol. IEC, on the other hand, specifies a simplified model, which contains only the **Physical**, **Data Link**, and **Application** layers. This model is called EPA (*Enhanced Performance Architecture*). The next figure shows the structure of that architecture and the communication system.



The **User** layer can be defined as the location where users manipulate data after all communications. In **Elipse Software** applications, this layer is represented by a user application. The **User** layer uses the **Application** layer of this Driver to send or receive full messages from and to a station.

The **Application** layer is responsible for specifying in details the requests from the **User** layer, and back to that layer when a message comes from the **Data Link** layer. In other words, this layer joins messages from the **User** layer, called fragments, into a message with multiple fragments with full information for processing and sending to a station by the **Data Link** layer.

The **Data Link** layer is used to send messages between primary, or origin, and secondary, or destination, stations. This layer also packs data, checks transmission errors, and sends this data to the TCP/IP network.

Introduction to DNP Mirror Master & Slave Driver

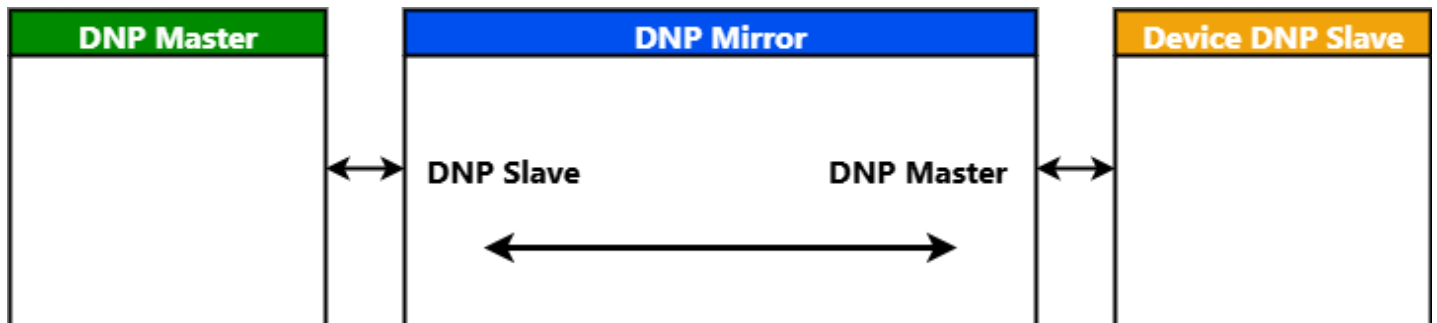
DNP Mirror Master & Slave Driver implements the **Master** and **Slave** behavior of DNP protocol. The goal of this product is to work as a DNP-to-DNP gateway, thus matching incompatible implementations of this protocol.

This Driver is composed of well-defined parts, a **Master** part and a **Slave** part. The **Master** part must be configured using almost the same configuration options available on **Elipse Software's DNP Master** Driver. Likewise, the **Slave** part follows the same configuration options available on **IOKit's DNP Slave** Driver.

However, in this Driver there is no need to create any Tag so that information received on the **Master** side be sent to the **Slave** side, because that operation is performed automatically.

Thus, any value or event received on the **Master** side is automatically sent to the **Slave** side. Likewise, any command received on the **Slave** side is automatically sent to the **Master** side.

DNP Mirror Master & Slave Driver does not have a dedicated database of points, because it retrieves that database from the **Master** and sends that same database to another **Master** from a **Slave**.

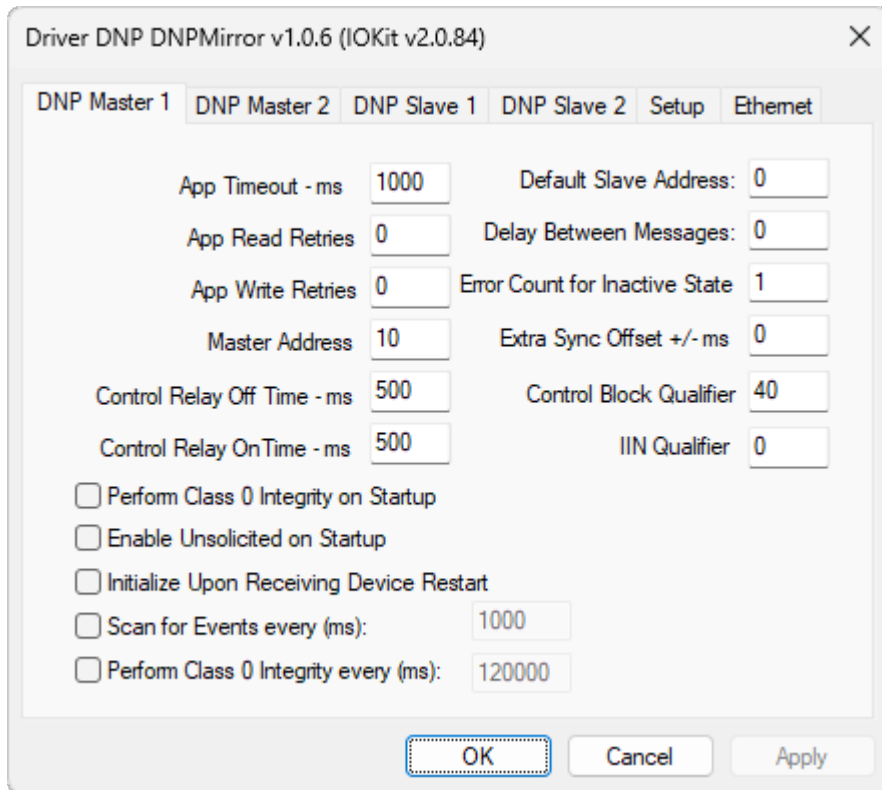


Architecture of DNP Mirror Master & Slave Driver

Configuration

The configuration of this Driver is performed on the tabs described on the next topics.

DNP Master 1 Tab



DNP Master 1 tab

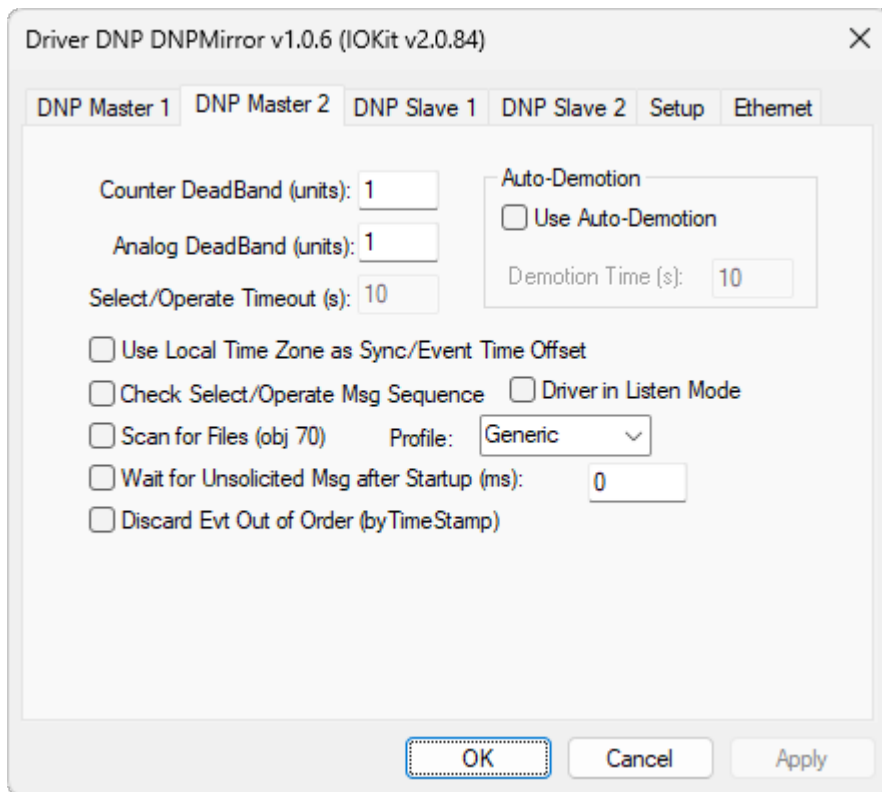
The available options on this tab are described on the next table.

Available options on the DNP Master 1 tab

OPTION	DESCRIPTION
App Timeout - ms	Maximum time that the Application layer waits for a full response from the Data Link layer. If receiving a request is ongoing in the Data Link layer, this time is extended automatically up until the end of the reception by the Data Link channel. The default value of this option depends on the communication rate used, but it is recommended that this value be equal to or greater than the time-out value of IOKit library, defined on the Setup tab. This time-out value of IOKit library represents the time-out byte by byte of a message, while the value of this option represents one or more full messages from the Data Link layer
App Read Retries	Number of communication retries performed by the Application layer in case of a reading error. The default value of this option is 0 (zero)
App Write Retries	Number of communication retries performed by the Application layer in case of a writing error. The default value of this option is 0 (zero)
Master Address	Indicates the address of the Master station
Control Relay Off Time - ms	When sending Control Relay Commands with individual Tags, this option indicates the normal time for Turned Off or off-time for Pulse On/Off or Latch On/Off commands

OPTION	DESCRIPTION
Control Relay On Time - ms	When sending Control Relay Commands with individual Tags, this option indicates the normal time for Turned On or off-time for Pulse On/Off or Latch On/Off commands
Default Slave Address	Default DNP address of a device, used when the <i>N1</i> parameter of each Tag is configured in 0 (zero)
Delay Between Messages	Delay time applied between each message sent by this Driver, in milliseconds
Error Count for Inactive State	Indicates how many consecutive errors this Driver must consider to set a device into the Inactive status. This Driver tries to communicate again with that device during the next scan of any Tag from the same address or only during the time informed in the Demotion Time option, if that feature is used
Extra Sync Offset +/- ms	Time, positive or negative, added to syncing commands
Control Block Qualifier	When performing a control-block writing command or a digital output command, such as Direct Operate , Select , or Operate , among others, users must send a qualifier that may vary from device to device. Possible values in this option are 39 , or 27h, for the range field with 1 (one) byte and prefixed object with 2 (two) bytes, or 40 , or 28h, for the range field with 2 (two) bytes and prefixed object with 2 (two) bytes. Please check the device profile of a device for the correct qualifier
IIN Qualifier	When a Slave device informs a restart, this Driver performs a writing to the <i>Internal Indications Object</i> , informing the acknowledgment of an event. Use this option to inform the qualifier, available on the Slave Device Profile document of a Slave device. The most common values are 0 : 1 (one) starting or ending index byte or 1 : 2 (two) starting or ending index byte
Perform Class 0 Integrity on Startup	Performs a Class 0 request when starting this Driver, which retrieves the whole database of a device to update all Tags. Event Tags, in this case, have the TimeStamp property as the time of data reception and the Quality property is equal to 216
Enable Unsolicited on Startup	Indicates whether this Driver must send a command to enable unsolicited messages when starting a communication
Initialize Upon Receiving Device Restart	Indicates whether this Driver must send initialization commands of a device (Reset Link , Class 0 , and Enable Unsolicited , if enabled) when receiving a message that a device was restarted
Scan for Events every (ms)	Sends a command to read Classes 1 (one), 2 (two), or 3 (three) to check whether there are events in a device. Indicate the time interval that a command repeats
Perform Class 0 Integrity every (ms)	Performs a Class 0 request cyclically, aiming only to check whether this Driver's database is equal to a device's database. The default time is 10 or 15 minutes

DNP Master 2 Tab



DNP Master 2 tab

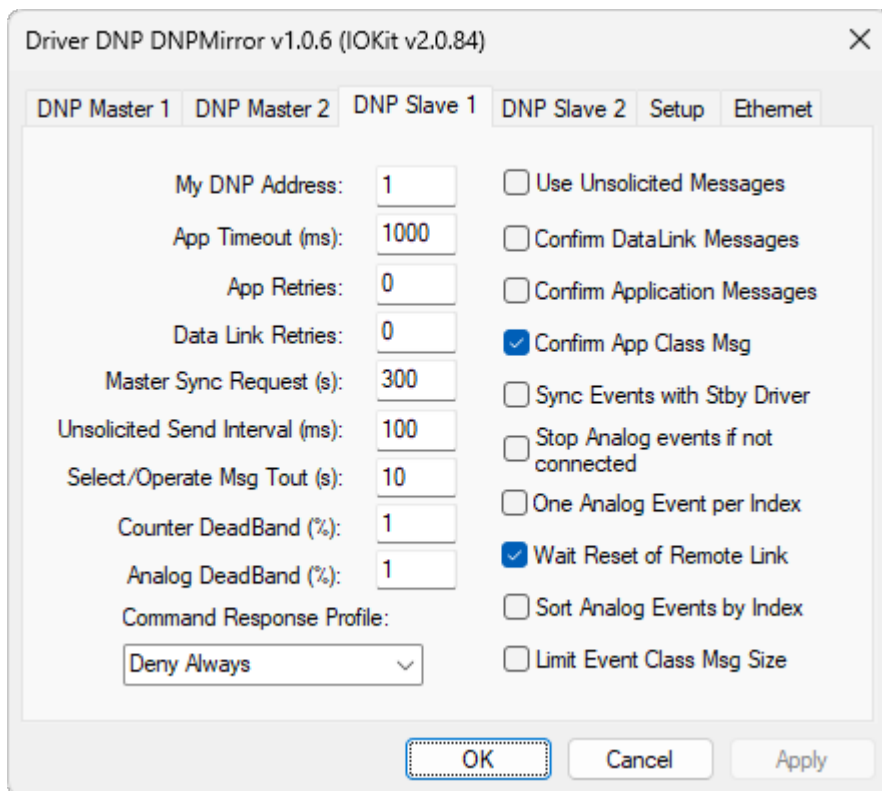
The available options on this tab are described on the next table.

Available options on the DNP Master 2 tab

OPTION	DESCRIPTION
Counter DeadBand (units)	Dead band, in units, during the verification for new events for counters
Analog DeadBand (units)	Dead band, in units, during the verification for new events for analog points
Select/Operate Timeout (s)	Maximum time for blocking other messages, except the Operate message, right after a Select command
Use Auto-Demotion	Enable the auto-demotion system, which automatically remove from and insert into communication the Slaves that are in the Inactive status, that is, with communication error. This procedure is used when two or more Slaves use the same channel, thus avoiding that a Slave in that status monopolize the usage of a channel
Demotion Time (s)	Time, in seconds, that this Driver tries to communicate with each Slave in the Inactive status, to check whether communication is performing normally, setting this Slave back to the Active status
Use Local Time Zone as Sync/Event Time Offset	Instructs this Driver to consider the difference between Windows official time and UTC (<i>Coordinated Universal Time</i>) or GMT (<i>Greenwich Mean Time</i>) time for all events and requests for clock syncing. This function is useful when the DNP's Server -side has its time in UTC but users must inform the local relative time to an application

OPTION	DESCRIPTION
Check Select/Operate Msg Sequence	Allows using Operate commands controlling the sequence number of a message so that it is immediately after the Select command. When selecting this option, this Driver does not send any command, except the Operate command, until a maximum time for an operation is exceeded
Driver in Listen Mode	In this mode, this Driver does not transmit any data, but it can interpret responses received by a communication sniffer between other Masters and a Slave
Scan for Files (obj 70)	Indicates that this Driver must try to perform file transfers from a device. Users must select a collecting profile. Currently, only the Pextron profile is supported
Profile	Collecting profile. The available options are Generic or Pextron
Wait for Unsolicited Msg after Startup (ms)	Indicates how much time this Driver must wait for unsolicited messages after connection, aiming to retrieve recent events before performing an integrity request, or Class 0 (zero). Thus, points with events receive a quality 192 and the correct timestamp, because in the integrity response the object do not have a timestamp
Discard Evt Out of Order (by TimeStamp)	For objects with information about timestamp, this option allows events whose timestamp is older than the last processed event are discarded

DNP Slave 1 Tab



DNP Slave 1 tab

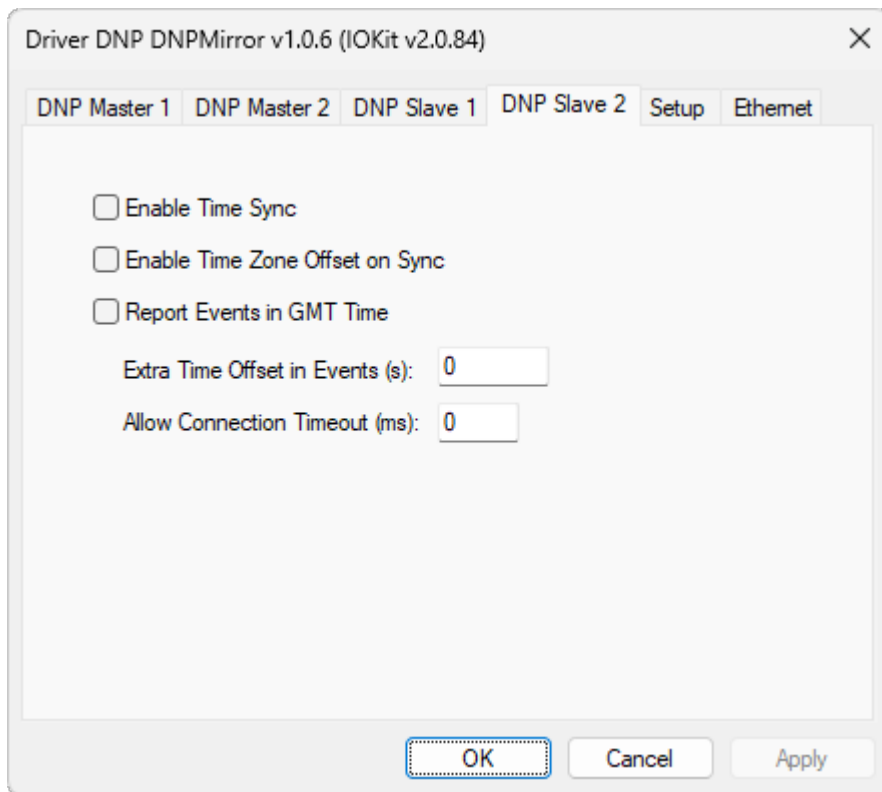
The available options on this tab are described on the next table.

Available options on the DNP Slave 1 tab

OPTION	DESCRIPTION
My DNP Address	Address of this Driver (Slave). If there are other Slaves on the same link, this value must be unique
App Timeout (ms)	Maximum time the Application layer waits for a full response from the Data Link layer. If receiving a request is ongoing by the Data Link layer, this time is automatically extended up until the end of reception by the Data Link layer. The default value of this option depends on the communication rate used. The time-out of the Data Link layer, or byte by byte, is defined by the time-out of IOKit library, defined on the Setup tab. The value of this option must be equal or greater than the time-out of the Data Link layer
App Retries	Number of communication retries performed by the Application layer in case of transaction error. The default value of this option is 0 (zero)
Data Link Retries	Number of communication retries performed by the Data Link layer in case of transaction error. The default value of this option is 0 (zero)
Master Sync Request (s)	Time interval that this Driver requests to a Master a clock syncing, which may or may not be accepted, according to the Enable Time Sync option. To disable sending, configure this option with the value 0 (zero)
Unsolicited Send Interval (ms)	Defines a time interval in which this Driver checks for the existence of events available from Classes 1 (one), 2 (two), or 3 (three) to be sent using unsolicited messages, if the Use Unsolicited Messages option is enabled. The Master side can also enable or disable sending unsolicited messages from a Slave when sending Enable and Disable Unsolicited Message requests, that is, the 20 and 21 functions
Select/Operate Msg Tout (s)	Maximum time, in seconds, between a Select command and an Operate command. After this time, the Operate command is no longer accepted by this Driver
Counter DeadBand (%)	Informs a dead band, as a percentage, for the purpose of notifying events for counters
Analog DeadBand (%)	Informs a dead band, as a percentage, for the purpose of notifying events for analog points
Command Responsive Profile	Informs how to handle commands requested to this Driver, that is, the objects 12 variation 1 (one) and 41 variations 1 (one), 2 (two), 3 (three), and 4 (four). The available options are Deny Always : All commands are responded negatively and instantly, with the Status field of a command reporting code 4 (four, unsupported command), Accept Always : All commands are responded positively, with the Status field of a command reporting code 0 (zero, command OK) after the respective command Tag is read by an application. Its purpose is to indicate that the requested command was acknowledged and is going to be processed by an application, or Wait for Application

OPTION	DESCRIPTION
	<p>Response: Commands are only responded after the respective Tag is read and written by an application. The purpose of this reading is that an application acknowledges that there is a request from the Master, and it is sent to another Driver or system output. After that processing, an application must write back to the Tag the same value read to indicate that this processing was successful, or a different value if it is not successful</p>
Use Unsolicited Messages	Informs whether this Driver sends unsolicited messages
Confirm DataLink Messages	Informs whether this Driver requests confirmation messages on the Data Link layer
Confirm Application Messages	Informs whether this Driver requests confirmation messages on the Application layer
Confirm App Class Msg	Informs whether this Driver requests confirmation messages for class or event messages. NOTE: This option, for security purposes, is always enabled internally
Sync Events with Stby Driver	If there is a second Slave DNP Driver in the same application and it is operating redundantly, that is, the Master side selects in which channel the communication is performed, this option instructs this Driver that, at each confirmation message of class events received, generates a corresponding message to be sent to the redundant Driver. NOTE: This very option must be configured in the second Driver, the Confirm App Class Msg option must be configured in both Driver, and there must be an additional code, or script, to send a message from one Driver to another
Stop Analog events if not connected	Stops accumulating analog events when the connection to a Master is interrupted. The current value of each analog Tag keeps being stored. Applies to objects 32 and 33
Wait Reset of Remote Link	Instructs this Driver to only respond communications after receiving a Reset of Remote Link command from a Master . The default value of this option is selected
Sort Analog Events by Index	Instructs this Driver to sort analog events, that is, objects 3X with any variation, by the index and not by the timestamp, as the standard for all other events. This allows Sag/Swell -type events from a point to be reported together, enabling the Master side to identify that this is an event of that type
Limit Event Class Msg Size	Limits event messages to a fragment of the Data Link layer at a time

DNP Slave 2 Tab



DNP Slave 2 tab

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

Available options on the DNP Slave 2 tab

OPTION	DESCRIPTION
Enable Time Sync	Informs whether this Driver accepts requests for clock syncing, which by their turn sync Windows clock
Enable Time Zone Offset on Sync	Informs whether, at the time of syncing, this Driver considers the time offset relative to the time in GMT
Report Events in GMT Time	Changes the timestamp of Tags, transforming them to GMT time based on Windows regional settings and daylight saving time

Supported Objects

The next table describes all object supported by DNP Mirror Master & Slave Driver.

Objects supported by DNP Mirror Master & Slave Driver

OBJECT CODE	TYPE	VARIATION	OBJECT NAME	FUNCTION
1	Static	1 (one)	Binary Input without Status	1 (one)
1	Static	2 (two)	Binary Input with Status	1 (one)
2	Event	1 (one)	Binary Input Change without Time	1 (one)

OBJECT CODE	TYPE	VARIATION	OBJECT NAME	FUNCTION
2	Event	2 (two)	Binary Input Change with Time	1 (one)
2	Event	3 (three)	Binary Input Change with Relative Time	1 (one)
3	Static	1 (one)	Double bit binary Input without Status	1 (one)
3	Static	2 (two)	Double bit binary Input with Status	1 (one)
4	Event	1 (one)	Double bit binary Input Change without Time	1 (one)
4	Event	2 (two)	Double bit binary Input Change with Time	1 (one)
10	Static	1 (one)	Binary Output	1 (one)
10	Static	2 (two)	Binary Output Status	1 (one)
11	Event	1 (one)	Binary Output Event without Time	1 (one)
11	Event	2 (two)	Binary Output Event with Time	1 (one)
12	--	1 (one)	Control Relay Output Block	3 (three), 4 (four), 5 (five), and 6 (six)
20	Static	1 (one)	32-bit Counter	1 (one)
20	Static	2 (two)	16-bit Binary Counter	1 (one)
20	Static	3 (three)	32-bit Delta Counter	1 (one)
20	Static	4 (four)	16-bit Delta Counter	1 (one)
20	Static	5 (five)	32-bit Counter without Flag	1 (one)
20	Static	6 (six)	16-bit Counter without Flag	1 (one)
20	Static	7 (seven)	32-bit Delta Counter without Flag	1 (one)
20	Static	8 (eight)	16-bit Delta Counter without Flag	1 (one)
21	Static	1 (one)	32-bit Frozen Counter	1 (one)
21	Static	2 (two)	16-bit Frozen Counter	1 (one)
21	Static	3 (three)	32-bit Frozen Delta Counter	1 (one)
21	Static	4 (four)	16-bit Frozen Delta Counter	1 (one)
21	Static	5 (five)	32-bit Frozen Counter with Time Of Freeze	1 (one)

OBJECT CODE	TYPE	VARIATION	OBJECT NAME	FUNCTION
21	Static	6 (six)	16-bit Frozen Counter with Time Of Freeze	1 (one)
21	Static	7 (seven)	32-bit Frozen Delta Counter with Time Of Freeze	1 (one)
21	Static	8 (eight)	16-bit Frozen Delta Counter with Time Of Freeze	1 (one)
21	Static	9 (nine)	32-bit Frozen Counter without Flag	1 (one)
21	Static	10	16-bit Frozen Counter without Flag	1 (one)
21	Static	11	32-bit Frozen Delta Counter without Flag	1 (one)
21	Static	12	16-bit Frozen Delta Counter without Flag	1 (one)
22	Event	1 (one)	32-bit Counter Change Event without Time	1 (one)
22	Event	2 (two)	16-bit Counter Change Event without Time	1 (one)
22	Event	3 (three)	32-bit Delta Counter Change Event without Time	1 (one)
22	Event	4 (four)	16-bit Delta Counter Change Event without Time	1 (one)
22	Event	5 (five)	32-bit Counter Change Event with Time	1 (one)
22	Event	6 (six)	16-bit Counter Change Event with Time	1 (one)
22	Event	7 (seven)	32-bit Delta Counter Change Event with Time	1 (one)
22	Event	8 (eight)	16-bit Delta Counter Change Event with Time	1 (one)
23	Event	1 (one)	32-bit Counter Change Event without Time	1 (one)
23	Event	2 (two)	16-bit Frozen Counter Event without Time	1 (one)
23	Event	3 (three)	32-bit Frozen Delta Counter Event without Time	1 (one)
23	Event	4 (four)	16-bit Frozen Delta Counter without Time	1 (one)

OBJECT CODE	TYPE	VARIATION	OBJECT NAME	FUNCTION
23	Event	5 (five)	32-bit Frozen Counter Event with Time	1 (one)
23	Event	6 (six)	16-bit Frozen Counter Event with Time	1 (one)
23	Event	7 (seven)	32-bit Frozen Delta Counter Event with Time	1 (one)
23	Event	8 (eight)	16-bit Frozen Delta Counter Event with Time	1 (one)
30	Static	1 (one)	32-bit Analog Input	1 (one)
30	Static	2 (two)	16-bit Analog Input	1 (one)
30	Static	3 (three)	32-bit Analog Input without Flag	1 (one)
30	Static	4 (four)	16-bit Analog Input without Flag	1 (one)
30	Static	5 (five)	32-bit Analog Input Floating Point	1 (one)
31	Static	1 (one)	32-bit Frozen Analog Input	1 (one)
31	Static	2 (two)	16-bit Frozen Analog Input	1 (one)
31	Static	3 (three)	32-bit Frozen Analog Input with Time Of Freeze	1 (one)
31	Static	4 (four)	16-bit Frozen Analog Input with Time Of Freeze	1 (one)
31	Static	5 (five)	32-bit Frozen Analog Input without Flag	1 (one)
31	Static	6 (six)	16-bit Frozen Analog Input without Flag	1 (one)
31	Static	7 (seven)	32-bit Frozen Analog Input Floating Point	1 (one)
32	Event	1 (one)	32-bit Change Event without Time	1 (one)
32	Event	2 (two)	16-bit Change Event without Time	1 (one)
32	Event	3 (three)	32-bit Analog Change with Time	1 (one)
32	Event	4 (four)	16-bit Analog Change Event with Time	1 (one)
32	Event	5 (five)	32-bit Analog Change Floating Point without Time	1 (one)

OBJECT CODE	TYPE	VARIATION	OBJECT NAME	FUNCTION
32	Event	7 (seven)	32-bit Analog Change Floating Point with Time	1 (one)
33	Event	1 (one)	32-bit Frozen Analog Event without Time	1 (one)
33	Event	2 (two)	16-bit Frozen Analog Event without Time	1 (one)
33	Event	3 (three)	32-bit Frozen Analog Event with Time	1 (one)
33	Event	4 (four)	16-bit Frozen Analog Event with Time	1 (one)
33	Event	5 (five)	32-bit Frozen Analog Floating Point without Time	1 (one)
33	Event	7 (seven)	32-bit Frozen Analog Floating Point with Time	1 (one)
34	Static	1 (one)	16-bit Analog Input Dead band	1 (one)
34	Static	2 (two)	32-bit Analog Input Dead band	1 (one)
34	Static	3 (three)	32-bit Analog Input Floating Point Dead band	1 (one)
40	Static	1 (one)	32-bit Analog Output Status	1 (one)
40	Static	2 (two)	16-bit Analog Output Status	1 (one)
40	Static	3 (three)	32-bit Analog Output Status Floating Point	1 (one)
41	--	1 (one)	32-bit Analog Output Block	2 (two), 3 (three), 4 (four), 5 (five), and 6 (six)
41	--	2 (two)	16-bit Analog Output Block	2 (two), 3 (three), 4 (four), 5 (five), and 6 (six)
41	--	3 (three)	32-bit Floating Point Analog Output Block	2 (two), 3 (three), 4 (four), 5 (five), and 6 (six)
42	Event	1 (one)	32-bit Analog Output Event without Time	1 (one)
42	Event	2 (two)	16-bit Analog Output Event without Time	1 (one)
42	Event	3 (three)	32-bit Analog Output Event with Time	1 (one)

OBJECT CODE	TYPE	VARIATION	OBJECT NAME	FUNCTION
42	Event	4 (four)	16-bit Analog Output Event with Time	1 (one)
42	Event	5 (five)	32-bit Floating Point Analog Output Event without Time	1 (one)
42	Event	7 (seven)	32-bit Floating Point Analog Output Event with Time	1 (one)
50	Static	1 (one)	Time and Date	1 (one) and 2 (two)
51	Static	1 (one)	Time and Date CTO (Common Time of Occurrence)	1 (one)
51	Static	2 (two)	Unsynchronized Time and CTO (Common Time of Occurrence)	1 (one)
52	Static	2 (two)	Time Delay Fine	1 (one)
60	Static	1 (one)	Class 0 Data	1 (one)
60	Event	2 (two)	Class 1 Data	1 (one)
60	Event	3 (three)	Class 2 Data	1 (one)
60	Event	4 (four)	Class 3 Data	1 (one)
80	--	1 (one)	Internal Indications	1 (one) and 2 (two)
83	Event	1 (one)	Pro	1 (one)
110	Static	X	Octet String	1 (one)
111	Event	X	Octet String Event	1 (one)

Documentation of I/O Interfaces

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

The screenshot shows the 'Setup' tab of a configuration dialog. It is organized into several sections:

- Physical Layer:** A dropdown menu is set to 'Ethernet'. To the right is a checkbox labeled 'Start driver OFFLINE' which is currently unchecked.
- Timeout:** A text box contains '1000' followed by 'ms'. To its right, 'Communication check time:' is followed by a text box containing '5000' followed by 'ms'.
- Connection management:** A dropdown menu is set to 'Automatic (managed by the driver)'. Below it are three options:
 - 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).

Revision History

VERSION	DATE	AUTHOR	COMMENTS
1.0.7	09-02-2025	M. Ludwig	<ul style="list-style-type: none"> Driver updated to IOKit library version 3.0 and Visual Studio 2022 (Case 38038).
1.0.6	05/08/2017	F. Englert	<ul style="list-style-type: none"> Performed adjustments in this Driver to use IOKit library version 2.0.84 (Case 22649).
1.0.5	02/15/2017	M. Salvador	<ul style="list-style-type: none"> Created an option for manual offset, positive or negative, to add to the timestamp of events on the Slave tab (Case 21903).
1.0.4	12/02/2016	M. Salvador	<ul style="list-style-type: none"> Implemented a delay of 5 (five) seconds before requesting a Class 0 (zero) after a reset notification (Case 21651).
1.0.3	07/01/2015	M. Salvador	<ul style="list-style-type: none"> Implemented improvements when handling disconnection for better notifications about these situations for a Master (Case 18194).
1.0.2	10/24/2014	M. Salvador	<ul style="list-style-type: none"> 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)