IEC870-104 Driver

Filename	IEC870-104.dll
Protocols	**IEC 870-5-104 Master** and **IEC 870-5-104 Slave**
Version	4.0.23
Last Update	08/27/2025
Platform	Win32
Level	31201
Superblock Reading	No

Introduction

The IEC870-104 Driver implements the **Master** and **Slave** modes of **IEC 870-5-104** protocol.

Driver Configuration

This section contains information about the configuration of this Driver.

[P] Parameters

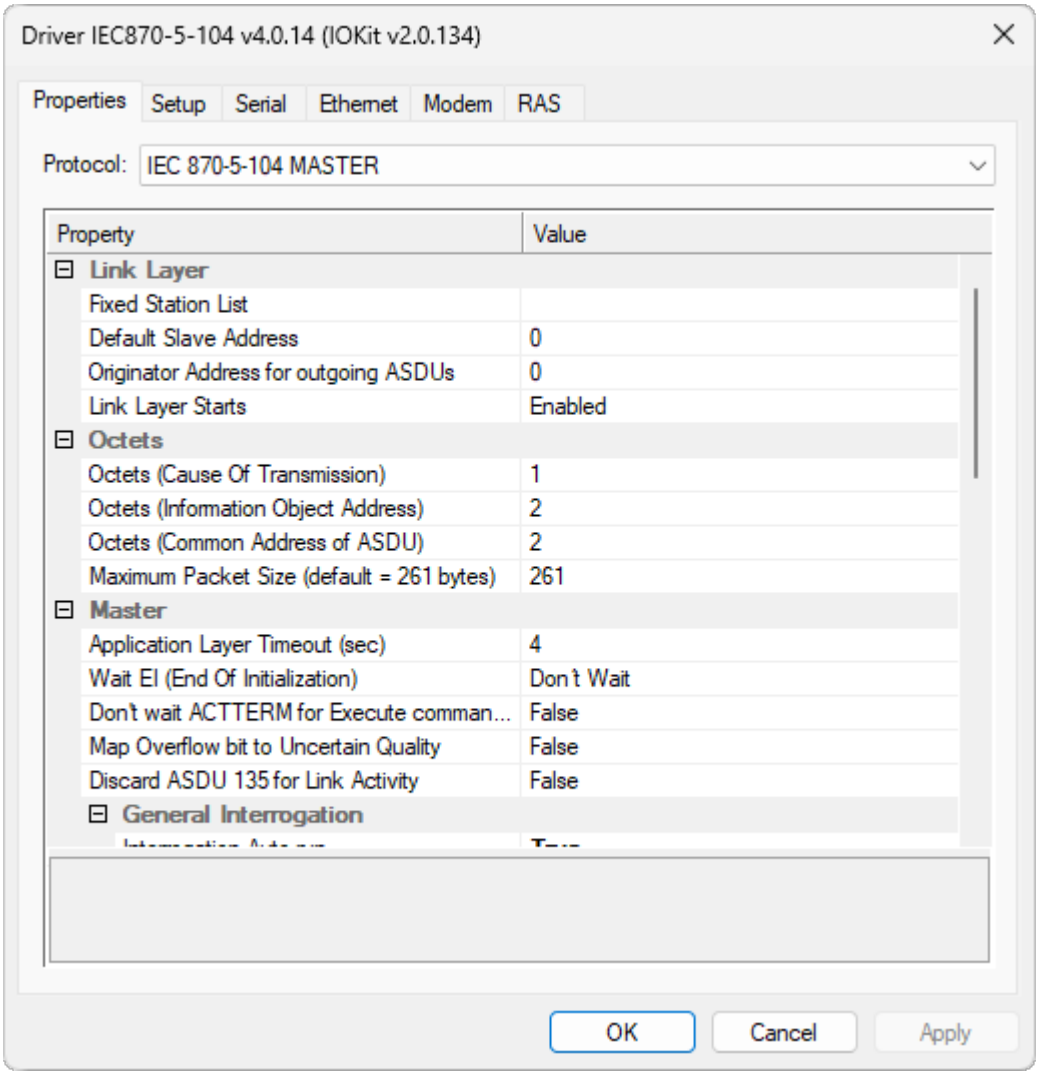
The IEC870-104 Driver does not use the **P1**, **P2**, **P3**, and **P4** parameters. It is advisable, for future compatibility issues, that the value of these parameters be configured as 0 (zero).

Properties Window

The Properties Window of this Driver is composed of several tabs. The tab that configures the protocol itself is the **Properties** tab. The **Setup**, **Serial**, **Ethernet**, **Modem**, and **RAS** tabs configure logs and the I/O layer of this Driver. For more information about those tabs, please check topic **Documentation of I/O Interfaces**.

Properties Tab

The **Properties** tab allow viewing and configuring all configuration **properties** of this Driver, shown on the next figure.



Properties tab of IEC870-104 Driver

This Driver's properties are grouped by categories and described next.

Protocol Option

Option that allows selecting the protocol for use by this Driver. The available options are **IEC 870-5-104 MASTER** or **IEC 870-5-104 SLAVE**. Next there is a list with all properties available for the selected protocol. Properties in bold indicate properties with a value different from the default. To return a property's value to its default, use the DEL key.

Link Layer Group

General options for the data layer.

Available options for the Link Layer group

OPTION	DESCRIPTION
Fixed Station List	In protocols where this Driver is in **Master** mode, a Station represents a Slave to which this Driver communicates. In protocols where this Driver is in **Slave** mode, a Station represents a logical device served by this Driver, to which an external Master communicates. By using this option, users can declare a list of Stations used by this Driver. If this option is not filled, this Driver operates in automatic

OPTION	DESCRIPTION
	mode, creating Stations as they are needed. The automatic mode is discouraged for protocols in **Slave** mode, because wrong Stations can be created according to requests performed by a Master. If this Driver is in **Slave** mode, only one Station must be created, informing only the address of this Slave. If this Driver is in **Master** mode, more than one Station can be created, by informing a sequence of addresses (Common Address of ASDU) separated by commas
Default Slave Address	The Station address of each Tag can be informed in the **Device** parameter or in the **N1** parameter. This option configures the default Station address used by this Driver when the **N1** or **B1** parameter of a Tag is equal to 0 (zero)
Originator Address for Outgoing ASDUs	An Originator Address is a second address field in the protocol **IEC 870-5-104**. This address is transmitted in the second octet of the **Octets (Cause of Transmission)** option. In case users need to send the Originator Address, the **Octets (Cause of Transmission)** option must be configured with the value 2 (two)
Link Layer Starts	Informs whether this Driver already starts communicating or it waits for an initialization through a writing to a special Tag with the **N2** parameter equal to 996

Octets Group

Configures the size, in octets (bytes), of several fields of the data packet used by this Driver. These configurations must comply with the configuration of the devices to which this Driver is communicating.

Available options for the Octets group

OPTION	DESCRIPTION
Octets (Cause of Transmission)	Configures the number of octets of an ASDU's Cause of Transmission. Allowed values are 1 (one) or 2 (two). The second octet, when enabled, contains the value of the **Originator Address** option
Octets (Information Object Address)	Configures the number of octets of an object's address, used in ASDUs. Valid values are in the range between 1 (one) and 3 (three)
Octets (Common Address of ASDU)	Configures the number of octets of an ASDU's common address, or address of a logical device inside a physical device. By default this property follows the number of octets of the **Default Slave Address** option. Allowed values are 1 (one) or 2 (two)
Maximum Packet Size	Informs the maximum size of a data packet. Default value of this option is 261

Master Group

General options for when this Driver is in **Master** mode.

Available options for the Master group

OPTION	DESCRIPTION
Application Layer Timeout (sec)	Defines the maximum time, in seconds, that this Driver waits for the end of sending any Command. It is important to define a value large enough to allow potentially lengthy Commands, such as **General Interrogation** or **Select or Execute**, to be executed successfully
Wait EI (End of Initialization)	Select this option to force this Driver to wait for receiving an **M_EI_NA** ASDU and then consider a Slave as connected
Don't Wait ACTTERM for Execute Commands	Deselect this option to force this Driver to wait for receiving a signal that an Execute Command ended. Some Slaves do not send an end signal, therefore in these cases this option must remain selected
Map Overflow bit to Uncertain Quality	When processing flags from analog ASDUs, informs whether the overflow bit (OV) must be mapped to an uncertain quality (84, OPC's EU_EXCEEDED)
Discard ASDU 135 for Link Activity	Discards ASDU 135 (private) to consider a link as active. Therefore, if only this message is received in a channel, the link is considered inactive
General Interrogation	Configures the way this Driver in **Master** mode sends and processes a **General Interrogation Command**. The available options are **Interrogation Auto-run**: Enables sending a General Interrogation Command whenever this Driver establishes a connection with a Slave, **Interrogation Repeat**: Enables the repetition of sending a General Interrogation Command, **Interrogation Repeat Period (sec)**: Informs a repetition interval for a General Interrogation Command in seconds, **Interrogation Wait ACTTERM**: Forces this Driver to wait for receiving an end signal of a General Interrogation Command. Some Slaves do not send an end signal, therefore in these cases this option must remain deselected, **Interrogation Process At End**: Select this option to **process data only at the end of a General Interrogation Command**, **Disconnect if General Interrogation returns empty results**: Forces a disconnection from this Driver if the result of a General Interrogation Command returns an empty list, and **Use Broadcast**: Informs whether a General Interrogation Command must use a broadcast address
Test Command	Configures the automatic sending of a **Test Command**. The available options are **Test Command Enabled**: Enables sending periodic Test Commands whenever this Driver remains a certain time, in seconds, without performing a communication in application level and **Test Command Idle Time (sec)**: Informs the period of a Test Command
Clock Sync	Configures the automatic sending of a **Clock Sync Command**. The available options are **Clock Sync Auto-run**: Enables sending a Clock Sync Command whenever this Driver established a connection with a Slave or **Clock Sync Repeat**: Enables sending a Clock Sync Command at fixed

OPTION	DESCRIPTION
	intervals, and **Clock Sync Repeat Period (sec)**: Informs a repetition period for syncing, in seconds

Slave Group

General options for when this Driver is in **Slave** mode.

Available options for the Slave group

OPTION	DESCRIPTION
Send EI (End of Initialization) When Connected	Configures the sending of an EI (*End of Initialization*) when connected, informing that a Slave is ready for communication. The available options are **Send EI** or **Don't Send EI**
Analog Values Send Interval (ms)	Interval for sending changes of analog values, in milliseconds
Analog Deadband (% of previous value)	Default dead band for analog Tags. Users can also inform an individual dead band per Tag, by configuring the **Item** parameter
Treat First Write As	Informs whether the first writing of a Tag is considered a new event (the **New Event** option) or it only stores as the current value (the **Current Value** option). In this last case, only starting with the second writing an event can be generated, if a quality change or dead band violation occurs
Use Cache File	Indicates that a cache file will be used to store and retrieve the driver's last values during startup. In this case, the previous "**Treat First Write As**" option will be forced into "**New Event**" mode, as the first value will be obtained from the cache file. The cache file is binary and contains two sections: one with current data and another with events. Every 1 hour of execution (fixed value), the file is rewritten with only the current data, consolidating (removing) all existing events. As events occur within this interval, they are appended to the end of the file to achieve the best possible write performance.
Cache File Path	Folder where the file will be written or read. It can be an absolute or relative path. Example: "\folder\file". The extension is free. Suggestion: ".cache" or ".104cache".
Cache Startup/Shutdown Mode Time (s)	Time in seconds that the driver will wait to report null values. During startup, this time is counted automatically. During shutdown, the application must "notify" the driver that the data source (e.g., master driver) is being stopped by writing to the **Shutdown tag**. Example: Sub DriverMaster_BeforeStop()

OPTION	DESCRIPTION
	Application.GetObject(":IEC104_LAN_Slave.ShutDown").WriteEx 1 End Sub
General Interrogation	Configures a set of data returned to a Master as a response to a General Interrogation Command. For more information, please check topic **General Interrogation**
Command Handling	The available options are **Handle Mode for Clock Sync commands**: Configures the action this Slave executes when receiving a **Clock Sync Command**, **Handle Mode for Select/Execute commands**: Configures the action this Slave executes when receiving a Select or Execute Command, and **Handle Mode for Test commands**: Configures the action this Slave executes when receiving a Test Command
Cyclic	The available options are **Cyclic Data Transmission (1): Send Period (ms)**: Period for sending Class 1 (one) data. Default value is 5000, **Cyclic Data Transmission (1): Object Range**: Configures a range of objects to send in Class 1 (one). A range of objects must be informed the same way as the list of objects for General Interrogation Commands, **Cyclic Data Transmission (2): Send Period (ms)**: Period for sending Class 1 (one) data. Default value is 10000, and **Cyclic Data Transmission (2): Object Range**: Configures a range of objects to send in Class 1 (one). A range of objects must be informed the same way as the list of objects for General Interrogation Commands
Offline Buffers	The available options are **Offline Class 1 buffering**: Enables storing Class 1 (one, high priority, typically digital values) events in memory, **Offline Class 1 buffer size**: Informs the maximum number of events of each Tag that accumulate in memory for Class 1 (one), **Offline Class 2 buffering**: Enables storing Class 2 (two, low priority, typically analog values) events in memory, and **Offline Class 2 buffer size**: Informs the maximum number of events of each Tag that accumulate in memory for Class 2 (two)

104 Group

Options for timing and handling messages in protocol **IEC 870-5-104**, both in **Master** mode and in **Slave** mode.

Available options for the 104 group

OPTION	DESCRIPTION
T0 (sec)	Timeout to establish a connection
T1 (sec)	Timeout for sending or testing APDUs
T2 (sec)	Timeout for acknowledging in case there is no data messages t2 < t1

OPTION	DESCRIPTION
T3 (sec)	Timeout for sending test frames in case of a long period of inactivity
K	Maximum difference of messages to send new changes. Default value is 12 APDUs
W	Last acknowledgment after receiving information APDUs (I). Default value is 8 APDUs

Timestamp Group

General options for timestamps.

Available options for the Timestamp group

OPTION	DESCRIPTION
Timestamp SU Flag (Summertime adjustment)	Informs the behavior relative to bit **SU** (summer time) of timestamp flags. The available options are **Ignore** or **Automatic**
Enable Station TimeZone	Enables a timezone offset for Stations
Station TimeZone (in minutes)	Informs the timezone of Stations, in minutes

Timestamp (Slave) Group

Additional options for timestamps when this Driver is in **Slave** mode.

Available options for the Timestamp for Slaves group

OPTION	DESCRIPTION
SP Timestamp Format	Defines the timestamp format for **SP** data types. The available options are **No timestamp**, **Short timestamp (3 octets)**, or **Full timestamp (7 octets)**
DP Timestamp Format	Defines the timestamp format for **DP** data types. The available options are **No timestamp**, **Short timestamp (3 octets)**, or **Full timestamp (7 octets)**
BO Timestamp Format	Defines the timestamp format for **BO** data types. The available options are **No timestamp**, **Short timestamp (3 octets)**, or **Full timestamp (7 octets)**
ST Timestamp Format	Defines the timestamp format for **ST** data types. The available options are **No timestamp**, **Short timestamp (3 octets)**, or **Full timestamp (7 octets)**
MEn Timestamp Format	Defines the timestamp format for **MEn** data types. The available options are **No timestamp**, **Short timestamp (3 octets)**, or **Full timestamp (7 octets)**
MEs Timestamp Format	Defines the timestamp format for **MEs** data types. The available options are **No timestamp**, **Short timestamp (3 octets)**, or **Full timestamp (7 octets)**

OPTION	DESCRIPTION
MEf Timestamp Format	Defines the timestamp format for **MEf** data types. The available options are **No timestamp**, **Short timestamp (3 octets)**, or **Full timestamp (7 octets)**

Send Priority (Slave) Group

Sending priority for each type of object when this Driver is in **Slave** mode.

Available options for the Send Priority for Slaves group

OPTION	DESCRIPTION
SP Send Priority	Informs how data is sent for **SP** data types. The available options are **Send with HIGH Priority**, **Send with LOW Priority**, or **Send on General Interrogation only**
DP Send Priority	Informs how data is sent for **DP** data types. The available options are **Send with HIGH Priority**, **Send with LOW Priority**, or **Send on General Interrogation only**
BO Send Priority	Informs how data is sent for **BO** data types. The available options are **Send with HIGH Priority**, **Send with LOW Priority**, or **Send on General Interrogation only**
ST Send Priority	Informs how data is sent for **ST** data types. The available options are **Send with HIGH Priority**, **Send with LOW Priority**, or **Send on General Interrogation only**
MEn Send Priority	Informs how data is sent for **MEn** data types. The available options are **Send with HIGH Priority**, **Send with LOW Priority**, or **Send on General Interrogation only**
MEs Send Priority	Informs how data is sent for **MEs** data types. The available options are **Send with HIGH Priority**, **Send with LOW Priority**, or **Send on General Interrogation only**
MEf Send Priority	Informs how data is sent for **MEf** data types. The available options are **Send with HIGH Priority**, **Send with LOW Priority**, or **Send on General Interrogation only**

Boolean Values Group

Options for converting native values to **Boolean** values.

Available options for the Boolean Values group

OPTION	DESCRIPTION
Convert DCO to Boolean	This option allows using **Double Digital Point Commands (DC)** with a **Boolean** value
Convert DIQ to Boolean	This option allows using **Double Digital Point Commands (DP)** with a **Boolean** value
Convert RCO to Boolean	This option allows using **Step Regulation Commands (RC)** with a **Boolean** value

OPTION	DESCRIPTION
Convert SEP to Boolean	This option allows using Device Protection Events with a **Boolean** value

Stations

In protocols where this Driver is in **Master** mode, a Station represents a Slave with which this Driver communicates. In protocols where this Driver is in **Slave** mode, a Station represents a logical device server by this Driver, with which an external Master communicates.

Stations must be declared on this Driver's properties window, in the **Fixed Station List** property on the **Properties** tab.

If this option is not defined, this Driver operates in automatic mode, creating Stations with default parameters as they are needed. This automatic mode is discouraged for protocols in **Slave** mode, because incorrect Stations can be created according to requests performed by a Master.

Tags with Statistics and Diagnostic of a Station

The Tags in this section allow access to diagnostics and statistics kept by this Driver for each Station.

Status of Communication Channel Tag

Protocols	Possible values are **IEC 870-5-104 MASTER** or **IEC 870-5-104 SLAVE**
Numeric Parameters	**N1**: Station number, **N2**: 993, **N3**: 0 (zero), and **N4**: 0 (zero)
Parameters by String	Not available
Writing	Not available
Reading	Returns the status of the I/O channel. Possible values are **0**: The channel is not active or **1**: The channel is active
ASDUs	Not available

Status of Station Connection Tag

Protocols	Possible values are **IEC 870-5-104 MASTER** or **IEC 870-5-104 SLAVE**
Numeric Parameters	**N1**: Station number, **N2**: 992, **N3**: 0 (zero), and **N4**: 0 (zero)
Parameters by String	Not available
Writing	Not available
Reading	Returns the connection status of a Station. Possible values are **0**: The Station is not connected or is not responding to the link layer or **1**: The Station is connected
ASDUs	Not available

Event Log Reading Block Tag

Protocols	Possible values are **IEC 870-5-104 MASTER** or **IEC 870-5-104 SLAVE**
Numeric Parameters	**B1**: Station number, **B2**: 99, **B3**: 0 (zero), and **B4**: 0 (zero)
Parameters by String	Not available
Block Size	8 (eight) Elements
Writing	Not available
Reading	Returns the value, the quality, and the timestamp of received data
ASDUs	Not available

This Block Tag returns a log with all ASDUs received by a Station. Use this Tag's **OnRead** event to analyze all ASDUs received by a Station. The Elements of this Block Tag are the following:

- **Element 0 (Word)**: ASDU's data type
- **Element 1 (Word)**: Object's address
- **Element 2 (DWord)**: Flags 1 (one)
- **Element 3 (DWord)**: Flags 2 (two)
- **Element 4 (Double)**: Object's value
- **Element 5 (Double)**: Timestamp
- **Element 6 (String)**: ASDU's name
- **Element 7 (Word)**: Cause of Transmission (COT)

Statistics Tag: Counter of Packets Sent

Protocols	Possible values are **IEC 870-5-104 MASTER** or **IEC 870-5-104 SLAVE**
Numeric Parameters	**N1**: Station number, **N2**: 998, **N3**: 0 (zero), and **N4**: 1 (one)
Parameters by String	Not available
Writing	Not available
Reading	Returns the number of I/O packets already sent by this Driver to a Station
ASDUs	Not available

Statistics Tag: Counter of No-Replies

Protocols	Possible values are **IEC 870-5-104 MASTER** or **IEC 870-5-104 SLAVE**
Numeric Parameters	**N1**: Station number, **N2**: 998, **N3**: 0 (zero), and **N4**: 2 (two)
Parameters by String	Not available
Writing	Not available
Reading	Returns the number of I/O packets sent by this Driver without a response from the Station
ASDUs	Not available

Statistics Tag: Counter of Errors When Receiving Frames

Protocols	Possible values are **IEC 870-5-104 MASTER** or **IEC 870-5-104 SLAVE**
Numeric Parameters	**N1**: Station number, **N2**: 998, **N3**: 0 (zero), and **N4**: 3 (three)
Parameters by String	Not available
Writing	Not available
Reading	Counter for the number of packets received from a Station with error
ASDUs	Not available

Statistics Tag: Counter of Errors When Sending Frames

Protocols	Possible values are **IEC 870-5-104 MASTER** or **IEC 870-5-104 SLAVE**
Numeric Parameters	**N1**: Station number, **N2**: 998, **N3**: 0 (zero), and **N4**: 4 (four)
Parameters by String	Not available
Writing	Not available
Reading	Returns the number of failures when sending packets to a Station
ASDUs	Not available

Statistics Tag: Counter of Retries When Sending Packets

Protocols	Possible values are **IEC 870-5-104 MASTER** or **IEC 870-5-104 SLAVE**
Numeric Parameters	**N1**: Station number, **N2**: 998, **N3**: 0 (zero), and **N4**: 5 (five)
Parameters by String	Not available
Writing	Not available
Reading	Returns the number of retries of sending packets to a Station
ASDUs	Not available

Advanced Statistics Block Tag

Protocols	Possible values are **IEC 870-5-104 MASTER** or **IEC 870-5-104 SLAVE**
Numeric Parameters	**B1**: Station number, **B2**: 999, **B3**: 0 (zero), and **B4**: 0 (zero)
Parameters by String	Not available
Block Size	17 Elements
Writing	Not available
Reading	Returns a Block Tag with the current value of several statistics
ASDUs	Not available

The Elements of this Block Tag are the following:

- **Element 0 (Long)**: Number of created ASDUs
- **Element 1 (Long)**: Number of removed ASDUs
- **Element 2 (Long)**: Number of invalid ASDUs
- **Element 3 (Long)**: Number of ASDUs in the internal cache memory of events of a Slave
- **Element 4 (Long)**: Number of ASDUs in the internal cache memory of data of a Slave
- **Element 5 (Long)**: Number of ASDUs in the cache memory of logs of a Slave
- **Element 6 (Long)**: Number of allocated **HVALUEs**
- **Element 7 (Long)**: Number of references to **HVALUEs**
- **Element 8 (Long)**: Number of allocated **HTIMERs**
- **Element 9 (Long)**: Number of timer categories
- **Element 10 (Long)**: Number of created **HVALUEs**
- **Element 11 (Long)**: Number of bytes used in the process heap
- **Element 12 (Long)**: Number of heap blocks in the process

- **Element 13 (Long)**: Number of available bytes in the process heap
- **Element 14 (Long)**: Number of heap blocks available in the process
- **Element 15 (Long)**: Number of system processes
- **Element 16 (Long)**: Number of system threads

Tags and Data Types

This section contains the data types that can be read and written by this Driver.

Monitoring Data Types

This Driver supports monitoring, or reading, data types described on the next table.

Supported monitoring data types

DATA TYPE	DESCRIPTION	MNEMONIC	ASDU WITHOUT TIMESTAMP	ASDU WITHOUT 24-BIT TIMESTAMP	ASDU WITH 56-BIT TIMESTAMP
SP	**Simple Digital Point**	SP	M_SP_NA_1 (1)	M_SP_TA_1 (2)	M_SP_TB_1 (30)
DP	**Double Digital Point**	DP	M_DP_NA_1 (3)	M_DP_TA_1 (4)	M_DP_TB_1 (31)
ST	**Step Position**	ST	M_ST_NA_1 (5)	M_ST_TA_1 (6)	M_ST_TB_1 (32)
BO	**Bitstring**	BO	M_BO_NA_1 (7)	M_BO_TA_1 (8)	M_BO_TB_1 (33)
ME	**Analog Measurement Value**	**MEn**, **MEs**, or **MEf**	M_ME_NA_1 (9), M_ME_NB_1 (11), and M_ME_NC_1 (13)	M_ME_TA_1 (10), M_ME_TB_1 (12), and M_ME_TC_1 (14)	M_ME_TD_1 (34), M_ME_TE_1 (35), and M_ME_TF_1 (36)
IT	**Integrated Total**	IT	M_IT_NA_1 (15)	M_IT_TA_1 (16)	M_IT_TB_1 (37)

Monitoring data is sent from a Slave to a Master. When this Driver is in **Slave** mode, an application needs to feed this Driver with values to send to a Master, and this is performed by writings. When this Driver is in **Master** mode, the received values are provided to an application via readings.

Simple Digital Point (SP)

A **Simple Digital Point** is an object that stores an **ON** (1, one) or **OFF** (0, zero) value. Corresponds to ASDUs **1**: **M_SP_NA_1** without **timestamp**, **2**: **M_SP_TA_1** with **24-bit timestamp**, and **30**: **M_SP_TB_1** with **56-bit timestamp**. Allowed values for this data type are **0**: OFF or **1**: ON (when writing, any value different from zero is sent as ON). This object supports **quality** bits **IV**, **NT**, **SB**, and **BL**.

Double Digital Point (DP)

A **Double Digital Point** is an object that stores an **ON** or **OFF** value, and each state corresponds to an individual bit. Corresponds to ASDUs **3**: **M_DP_NA_1** without **timestamp**, **4**: **M_DP_TA_1** with **24-bit timestamp**, and **31**: **M_DP_TB_1** with **56-bit timestamp**.

In protocol **IEC 870-5-104**, the value transmitted in these ASDUs is a DIQ (*Double Point Information with Quality Descriptor*), which contains four **quality** bits (**IV**, **NT**, **SB**, and **BL**) and two value bits (called DPI, *Double Point Information*):

- **0**: Intermediate State (invalid)
- **1**: OFF
- **2**: ON
- **3**: Intermediate State (invalid)

The value of this Tag depends on the configuration performed on this Driver's properties window, in the **Convert DIQ to Boolean** option of the **Boolean Values** group on the **Properties** tab. If this option is not selected, the value of this Tag when reading is the DPI's value itself, both when reading and when writing. Writings on this Driver fail if the value is not one of the values described previously. If the **Convert DIQ to Boolean** option is selected, the value of Double Digital Point Tags is the following:

- **DPI equal to 0**: Returns **Null**, **Uncertain** quality
- **DPI equal to 1**: Returns 0 (zero)
- **DPI equal to 2**: Returns 1 (one)
- **DPI equal to 3**: Returns **Null**, **Bad** quality

When writing, any value different from 0 (zero) sends a DPI equal to 2 (two). The value 0 (zero) in this Tag sends a DPI equal to 1 (one).

Step Position (ST)

A **Step Position** is an object that stores an integer value in the range between -64 and 63. Corresponds to ASDUs **5**: **M_ST_NA_1** without **timestamp**, **6**: **M_ST_TA_1** with **24-bit timestamp**, and **32**: **M_ST_TB_1** with **56-bit timestamp**. This object supports **quality** bits **IV**, **NT**, **SB**, **BL**, and **OV**.

NOTE

The **T** (transient) flag is not supported by this Driver, that is, it is ignored when reading and configured to 0 (zero) when writing.

Bitstring (BO)

A **Bitstring** is an object that stores a sequence of 32 bits. Corresponds to ASDUs **7**: **M_BO_NA_1** without **timestamp**, **8**: **M_BO_TA_1** with **24-bit timestamp**, and **33**: **M_BO_TB_1** with **56-bit timestamp**. This Driver handles **Bitstring** values as 32-bit integers. This object supports **quality** bits **IV**, **NT**, **SB**, **BL**, and **OV**.

Analog Measurement Value (ME)

Analog Measurements can be represented in the formats described on the next topics.

Normalized Analog Measurement (MEn)

A **Normalized Analog Measurement** is an object that stores a floating-point value in the range between -1.0 and 1.0. Corresponds to ASDUs **9**: **M_ME_NA_1** without **timestamp**, **10**: **M_ME_TA_1** with **24-bit timestamp**, and **34**: **M_ME_TD_1** with **56-bit timestamp**. The range of valid values in this Tag goes from -1.0 to 1.0 and from 2 (two) to 15. This value's precision is 16 bits, therefore not all value inside that interval can be represented with precision, that is, there may be an approximation in sent and received values. This object supports **quality** bits **IV**, **NT**, **SB**, **BL**, and **OV**.

Scaled Analog Measurement (MEs)

A **Scaled Analog Measurement** is an object that stores a signed 16-bit value with a fixed decimal point. Corresponds to ASDUs **11**: **M_ME_NB_1** without **timestamp**, **12**: **M_ME_TB_1** with **24-bit timestamp**, and **35**: **M_ME_TE_1** with **56-bit timestamp**. In this Driver, the decimal point of all Scaled Analog Measurements is fixed in one decimal place. This way, the range of valid values for Tags of this type ranges from -3276.8 to 3276.7. If the decimal point of a device is different from one decimal place, then an appropriate scale must be configured in this Tag. This object supports **quality** bits **IV**, **NT**, **SB**, **BL**, and **OV**.

Floating-Point Analog Measurement (MEf)

A **Floating-Point Analog Measurement** is an object that stores 32-bit floating-point value in **IEEE STD 754** (*Short Floating Point Number*) format. Corresponds to ASDUs **13**: **M_ME_NC_1** without **timestamp**, **14**: **M_ME_TC_1** with **24-bit timestamp**, and **36**: **M_ME_TF_1** with **56-bit timestamp**. This object supports **quality** bits **IV**, **NT**, **SB**, **BL**, and **OV**.

Integrated Total (IT)

An **Integrated Total** is an object that stores the value of a signed 32-bit counter. Corresponds to ASDUs **15**: **M_IT_NA_1** without **timestamp**, **16**: **M_IT_TA_1** with **24-bit timestamp**, and **37**: **M_IT_TB_1** with **56-bit timestamp**. This Driver handles Integrated Total values as signed 32-bit integers. This object supports the **quality** bit **IV**. Fields **SQ** (sequence number), **CY** (counter overflow), and **CA** (adjusted counter) are ignored by this Driver when receiving and sent with value 0 (zero) during transmission.

Tag Configuration by Strings

To use a configuration by **String**, the **N** parameters of a Tag must be zeroed. Users must provide the address of a device in the **ParamDevice** field and the address of the monitored object in the **ParamItem** field, according to the next syntax.

```
<type>:<objectAddress>[<priorityFlag>][<DB:DeadBand>]
```

In which:

- **<type>** is a mnemonic for a data type, that is, **SP**, **DP**, **ST**, **BO**, **MEn**, **MEs**, or **MEf**. The colon that separates the mnemonic from the object's address is optional. The next table contains the monitoring data types and their respective mnemonics.

Data types and mnemonics

DATA TYPE	ASDU	MNEMONIC	EXAMPLE
Simple Digital Point	**M_SP_NA_1** (1), **M_SP_TA_1** (2), and **M_SP_TB_1** (30)	SP	SP:1
Double Digital Point	**M_DP_NA_1** (3), **M_DP_TA_1** (4), and **M_DP_TB_1** (31)	DP	DP:328
Step Position	**M_ST_NA_1** (5), **M_ST_TA_1** (6), and **M_ST_TB_1** (32)	ST	ST:10.434
Bitstring	**M_BO_NA_1** (7), **M_BO_TA_1** (8), and **M_BO_TB_1** (33)	BO	BO:99
Normalized Analog Measurement	**M_ME_NA_1** (9), **M_ME_TA_1** (10), **M_ME_ND_1** (21), and **M_ME_TD_1** (34)	MEn	MEn:10938
Scaled Analog Measurement	**M_ME_NB_1** (11), **M_ME_TB_1** (12), and **M_ME_TE_1** (35)	MEs	MEs:9833

DATA TYPE	ASDU	MNEMONIC	EXAMPLE
Floating-Point Analog Measurement	**M_ME_NC_1** (13), **M_ME_TC_1** (14), and **M_ME_TF_1** (36)	MEf	MEf:10.35.22

- **<objectAddress>** is the address of the monitoring object, which can be in the formats described on the next table.

Formats for object addresses

OBJECT ADDRESS	SUPPORTED FORMATS	EXAMPLE	GENERATED OCTETS
One octet	<0-255>	SP:255	FF
Two octets	<0-65535>	SP:65535	FF FF
	<0-255>.<0-255>	SP:1.2	01 02
Three octets	<0-16777215>	SP:16777215	FF FF FF
	<0-255>.<0-65535>	SP:1.65535	01 FF FF
	<0-255>.<0-255>.<0-255>	SP:26.5.3	1A 05 03

- **<priorityFlag>** is an optional character that allows defining a priority for sending data, only for protocols in **Slave** mode, according to the next table.

Available options for priority flags

FLAG	DESCRIPTION	EXAMPLE
omitted	Data is sent with a default priority, that is, digital data with high priority and analog data with low priority. The default priority can be redefined in the **Send Priority** group on the **Properties** tab of this Driver's settings	SP:255
!	When the character is an exclamation point, data is sent with **High** priority (Class 1)	SP:255!
<	When the character is a less than signal, data is sent only on the next General Interrogation Command	SP:255<
=	When the character is an equal sign, data is sent with **Low** priority (Class 2)	SP:255=

- **<DB:DeadBand>** is an optional information by Tag of a dead band in percentage, relative to the current value, used to define whether there were a significant change in value to send an information, when this Driver is configured in Slave mode. If it is not informed, the general definition of the **Analog Deadband (% of previous value)** property is used as dead band. For example, a value "MEs:9833DB:0.7" indicates the use of a 0.7% of dead band.

Command Data Types

This Driver supports the data types in Select or Execute Commands described on the next table.

Data types in Select or Execute Commands

DATA TYPE	DESCRIPTION	MNEMONIC	ASDU WITHOUT TIMESTAMP	ASDU WITH 56-BIT TIMESTAMP
SC	**Simple Digital Point Command**	SC	**C_SC_NA_1** (45)	**C_SC_TA_1** (58)
DC	**Double Digital Point Command**	DC	**C_DC_NA_1** (46)	**C_DC_TA_1** (59)
RC	**Step Regulation Command**	RC	**C_RC_NA_1** (47)	**C_RC_TA_1** (60)
SE	**Setpoint Command**	**SEn**, **SEs**, or **SEf**	**C_SE_NA_1** (48), **C_SE_NB_1** (49), and **C_SE_NC_1** (50)	**C_SE_TA_1** (61), **C_SE_TB_1** (62), and **C_SE_TC_1** (63)
BO	**Bitstring Command**	BO	**C_BO_NA_1** (51)	**C_BO_TA_1** (64)

All Commands have the next fields in common, which are sent together with the Command's value:

- **SE**: Determines whether it is a **Select** (1, one) or **Execute** (0, zero) Command
- **QU**: Determines a Command qualifier:
  - **0**: Without definition
  - **1**: Short Circuit Pulse (circuit breaker)
  - **2**: Long Circuit Pulse
  - **3**: Persistent Output
  - **4 to 8**: Reserved for new definition of IEC standard
  - **9 to 15**: Reserved for other pre-defined functions
  - **16 to 31**: Reserved for special use (private interval)

This Driver in **Master** mode only supports sending Commands with the **QU** field equal to 0 (zero), 1 (one), 2 (two), or 3 (three).

Simple Digital Point Command (SC)

A **Simple Digital Point Command** uses ASDUs **45**: **C_SC_NA_1** without **timestamp** and **58**: **C_SC_TA_1** with **56-bit timestamp**. Allowed values for this data type are **0**: OFF or **1**: ON.

Double Digital Point Command (DC)

A **Double Digital Point Command** corresponds to ASDUs **46**: **C_DC_NA_1** without **timestamp** and **59**: **C_DC_TA_1** with **56-bit timestamp**. In protocol **IEC 870-5-104**, the value transmitted in these ASDUs is a DCO (*Double Command*) that contains the fields **SE** and **QU**, in addition to two value bits, called DCS (*Double Command State*):

- **0**: Intermediate State (invalid)
- **1**: OFF
- **2**: ON

- **3**: Intermediate State (invalid)

The value of this Command depends on the configuration performed on this Driver's properties window, in the **Convert DCO to Boolean** option of the **Boolean Values** group on the **Properties** tab. If this option is not selected, the value written is the DCS value itself, described previously. If the option to handle as a **Boolean** is selected, the value of this Command is **0**: OFF (DPI equal to one) or **1**: ON (DPI equal to two).

Step Regulation Command (RC)

A **Step Regulation Command** allows incrementing or decrementing a Step Position. Corresponds to ASDUs **47**: **C_RC_NA_1** without **timestamp** and **60**: **C_RC_TA_1** with **56-bit timestamp**. In protocol **IEC 870-5-104**, the value transmitted in these ASDUs is an RCO (*Regulating Step Command*), which contains an **SE** bit, indicating whether it is a Select or Execute Command, five **QU** bits, indicating a Command qualifier, such as Short Circuit Pulse, Long Circuit Pulse, or Persistent Output, and two value bits, called RCS (*Regulating Step Command State*):

- **0**: Not allowed
- **1**: Next Step down (decrements)
- **2**: Next Step up (increments)
- **3**: Not allowed

The value of this Command depends on the configuration performed on this Driver's properties window, in the **Convert RCO to Boolean** option of the **Boolean Values** group on the **Properties** tab. If this option is not selected, the value written is the RCS value itself, described previously. If the option to handle as a **Boolean** is selected, the Command value is **0**: Next Step down (decrements, RCS equal to one) or **1**: Next Step up (increments, RCS equal to two).

Setpoint or Analog Command (SE)

Setpoint Commands, or writings to Analog Measurements, can be performed using the values described on the next topics.

Setpoint Command with Normalized Value (SEn)

A **Setpoint Command** allows sending a normalized value in the range between -1.0 and 1.0 for this Driver in **Slave** mode. Corresponds to ASDUs **48**: **C_SE_NA_1** without **timestamp** and **61**: **C_SE_TA_1** with **56-bit timestamp**. Values informed above or below the allowed limits are adjusted to the nearest limit. This value's precision is 16 bits, therefore not all values inside the range can be represented with precision, that is, there may be approximations in the sent and received values.

Setpoint Command with Scaled Value (SEs)

A **Setpoint Command** allows sending a scaled value, in the range between -3276.8 and 3276.7 for this Driver in **Slave** mode. Corresponds to ASDUs **49**: **C_SE_NB_1** without **timestamp** and **62**: **C_SE_TB_1** with **56-bit timestamp**. The value actually transmitted in the protocol corresponds to the value multiplied by 10, because a fixed decimal place is assumed, resulting in sending an integer value in the range between -32768 and 32767.

Setpoint Command with Floating-Point Value (SEf)

A **Setpoint Command** allows sending a 32-bit floating-point value, in **IEEE STD 754** (*Short Floating Point Number*) format. Corresponds to ASDUs **50**: **C_SE_NC_1** without **timestamp** and **63**: **C_SE_TC_1** with **56-bit timestamp**.

Bitstring Command (BO)

A **Bitstring Command** allows sending a 32-bit integer value. It corresponds to ASDUs **51**: **C_BO_NA_1** without **timestamp** and **64**: **C_BO_TA_1** with **56-bit timestamp**.

Command Configuration by Strings

To use a Command Configuration by **Strings** only in **Master** mode, the **N** parameters of a Tag must be zeroed. Users must set the address of a device in the **ParamDevice** field and the description of the Command in the **ParamItem** field, according to the next syntax.

```
<command>:<objectAddress>.<action>[(<qualifier>)]
```

In which:

- **<command>** is the mnemonic of a Command type, that is, **CBO**, **CBOt**, **DC**, **DCt**, **RC**, **RCt**, **SC**, **SCt**, **SEn**, **SEnt**, **SEs**, **SEst**, **SEf**, or **SEft**. The colon that separates the mnemonic from the object's address is optional. The next table contains Command data types and their respective mnemonics.

Data types and mnemonics

COMMAND TYPE	ASDU	MNEMONIC	EXAMPLE
Simple Digital Point	**C_SC_NA_1** (45) and **C_SC_TA_1** (58)	SC or SCt	SC:1.Select or SCt:1.Select
Double Digital Point	**C_DC_NA_1** (46) and **C_DC_TA_1** (59)	DC or DCt	DC:328.Execute or DCt:328.Execute
Step Regulation	**C_RC_NA_1** (47) and **C_RC_TA_1** (60)	RC or RCt	RC:10.434.Execute or RCt:10.434.Execute
Setpoint with Normalized Value	**C_SE_NA_1** (48) and **C_SE_TA_1** (61)	SEn or SEnt	SEn:10938.Execute(ShortPulse) or SEnt:10938.Execute(ShortPulse)
Setpoint with Scaled Value	**C_SE_NB_1** (49) and **C_SE_TB_1** (62)	SEs or SEst	SEs:9833.Execute or SEst:9833.Execute
Setpoint with Floating-Point Value	**C_SE_NC_1** (50) and **C_SE_TC_1** (63)	SEf or SEft	SEf:10.35.22.Execute or SEft:10.35.22.Execute
Bitstring	**C_BO_NA_1** (51) and **C_BO_TA_1** (64)	CBO or CBOt	CBO:99.Select or CBOt:99.Select

NOTE

Mnemonics ending in "t" correspond to Commands with a timestamp. All other mnemonics do not have a timestamp.

- **<objectAddress>** is the address of a Command object, which can be in the formats described on the next table

Formats for object addresses

INFORMATION OBJECT ADDRESS	SUPPORTED FORMATS	EXAMPLE	GENERATED OCTETS
One octet	<0-255>	SC:255.Select	FF

INFORMATION OBJECT ADDRESS	SUPPORTED FORMATS	EXAMPLE	GENERATED OCTETS
Two octets	<0-65535>	SC:65535.Select	FF FF
	<0-255>.<0-255>	SC:1.2.Select	01 02
Three octets	<0-16777215>	SC:16777215.Select	FF FF FF
	<0-255>.<0-65535>	SC:1.65535.Select	01 FF FF
	<0-255>.<0-255>.<0-255>	SC:26.5.3.Select	1A 05 03

- **<action>** defines the type of Command, which can be **Select**: Sends a **Select Command**, **Execute**: Sends an **Execute Command**, or **Deactivate**: Sends a **Cancel Select Command**
- **<qualifier>** is an optional parameter and must be informed between parenthesis:
  - **(LongPulse)**: Sends a Long Circuit Pulse Command
  - **(ShortPulse)**: Sends a Short Circuit Pulse Command
  - **(PersistentOutput)**: Sends a Persistent Output Command
  - **omitted**: Sends a normal Command

Master 104

In **Master** mode, this Driver supports the functions described on the next topics.

NOTE

The default TCP/IP port for **IEC 870-5-104** protocol is port 2404.

Reading Monitoring Data

Protocol **IEC 870-5-104** in **Master** mode allows receiving monitoring data sent by a Slave spontaneously or as an answer to some Command sent by this Master, such as a General Interrogation or Counter Interrogation Command. Data sent by a device is stored in an internal memory of this Driver, called **Cache**, and it is available until read by a Tag for reading cache memory.

NOTE

Cache memory stores up to 100 unread values for each Tag. If the number of pending reading events for a certain Tag exceeds this limit, the oldest events are discarded to give room to newer events.

The way to read data from the cache memory is described on the next topic.

Tag for Reading Events from Cache Memory

Protocols	**IEC 870-5-104 MASTER**
Numeric Parameters	**N1**: Station number, **N2**: 4, 5, 6 (four, five, six, normal reading), or 7 (seven, reading with automatic reset), **N3**: ASDU, and **N4**: Object's address
Parameters by String	Please check topic **Tag Configuration by Strings**
Writing	Not available
Reading	Returns the value, the quality, and the timestamp of received data
ASDUs	• **Defined in protocol IEC 870-5-101**: **M_SP_NA_1** (1), **M_SP_TA_1** (2), **M_DP_NA_1** (3), **M_DP_TA_1** (4), **M_ST_NA_1** (5), **M_ST_TA_1** (6), **M_BO_NA_1** (7), **M_BO_TA_1** (8), **M_ME_NA_1** (9), **M_ME_TA_1** (10), **M_ME_NB_1** (11), **M_ME_TB_1** (12), **M_ME_NC_1** (13), **M_ME_TC_1** (14), **M_IT_NA_1** (15), **M_IT_TA_1** (16), **M_PS_NA_1** (20), **M_ME_ND_1** (21), and **M_EI_NA_1** (70) • **Definid in protocol IEC 870-5-104**: **M_SP_TB_1** (30), **M_DP_TB_1** (31), **M_ST_TB_1** (32), **M_BO_TB_1** (33), **M_ME_TD_1** (34), **M_ME_TE_1** (35), **M_ME_TF_1** (36), and **M_IT_TB_1** (37)

This Tag returns events stored on this Driver's internal cache memory. After returning an event, it is removed from the cache memory. If the memory is empty, no value is returned to this Tag, that is, readings do not affect the value stored in this Tag. This Tag's **OnRead** event is activated whenever a new value is received, and can be used to store events on a historical file, for example. The **N2** parameter has the following meanings:

- **N2 equal to 4, 5, or 6**: Reading events from cache memory, only returns values this Driver received from a Slave
- **N2 equal to 7**: Reading events from cache memory with automatic reset. For Simple Digital Point and Double Digital Point data types, for each **ON** value received, an **OFF** event is generated one millisecond after the **ON** event

Reading this Tag do not cause any type of communication, it is always an immediate reading. This Tag does not support Block readings.

NOTE

It is recommended that a General Interrogation Command be enabled to force a periodic update of all values sent by a Slave.

Clock Syncing

A **Clock Syncing of a Slave** (ASDU 103, **C_CS_NA_1**) can be performed automatically or manually.

Automatic Syncing

To configure an Automatic Clock Syncing, configure the **Clock Sync Auto-run** property on the **Master - Clock Sync** group of the **Properties** tab of this Driver's configuration window. This way, this Driver syncs a Slave's clock whenever a communication is established. Plus, users can enable the repetition of that syncing at fixed intervals using the **Clock Sync Repeat** and **Clock Sync Repeat Period (sec)** options on the same group.

Manual Syncing

To perform a Manual Clock Syncing, write any value to the **Tag for Sending a Clock Syncing**.

Tag for Sending a Clock Syncing

Protocols	**IEC 870-5-104 MASTER**
Numeric Parameters	**N1**: Station number, **N2**: 2 (two), **N3**: 0 (zero), and **N4**: 0 (zero)
Parameters by String	Not available
Writing	Write any value to this Tag to perform a Clock Syncing of a Slave
Reading	Not supported
ASDUs	**C_CS_NA_1** (103)

Any value written to this Tag sends a Clock Syncing Command to a Slave. A Slave's clock is synced with Master's current time. If this Command is successful, this Tag returns the value 1 (one). If this Command fails for any reason, this Tag returns the value 0 (zero).

NOTE

If both Clock Syncing Command and General Interrogation Command are enabled, the Clock Syncing Command is always executed first.

General Interrogation

A **General Interrogation** is a procedure used to ask a Slave for sending the current status of all objects, which usually are sent without a timestamp.

To enable sending General Interrogation Commands to a Slave, configure the next properties on the **Master - General Interrogation** group of the **Properties** tab of this Driver's configuration window:

- **Interrogation Auto-run**: Select this option to enable sending a General Interrogation Command whenever a communication is established with a Slave
- **Interrogation Repeat e Interrogation Repeat Period (sec)**: Select this option and configure an interval, in seconds, to repeat a General Interrogation Command
- **Interrogation Wait ACTTERM**: Select this option to force this Driver to wait for a notification of a General Interrogation Command ending (**ACTTERM**). It is recommended that this option be deselected only if users detect that a Slave does not send that notification, because it is optional
- **Interrogation Process At end**: Select this option to **process data received only at the end of a General Interrogation Command**

Tag for Sending a General Interrogation

Protocols	**IEC 870-5-104 MASTER**
Numeric Parameters	**N1**: Station number, **N2**: 100, **N3**: 0 (zero), and **N4**: 0 (zero)
Parameters by String	Not available
Writing	Write any value to this Tag to send an Interrogation Command of a group of counters
Reading	Not supported
ASDUs	**C_IC_NA_1** (100)

This Driver in **Master** mode already performs a General Interrogation Command (GI) every time a Slave Station activates the communication, or when the interval of cyclic GI (*Interrogation Repeat*) expires. To request a GI Command at any moment, write any value to this Tag. Any value written to this Tag sends a General Interrogation Command to a Slave. If this Command is successful, this writing returns the value 1 (one). If this Command fails for any reason, this writing returns the value 0 (zero). Reading this Tag is not supported.

Tag for Sending a Group Interrogation

Protocols	**IEC 870-5-104 MASTER**
Numeric Parameters	**N1**: Station number, **N2**: 101 a 116, **N3**: 0 (zero), and **N4**: 0 (zero)
Parameters by String	Not available
Writing	Not supported
Reading	Possible values are **1**: Indicates that the Interrogation Command was successful or **0**: There was a failure
ASDUs	**C_IC_NA_1** (100)

Enable reading this Tag to request a Group Interrogation Command, indicated by the **N2** parameter equal to 100 plus the group index, from 1 (one) to 16. If this Command is successful, this Tag returns the value 1 (one). If this Command fails for any reason, this Tag returns the value 0 (zero).

Processing Data at the End of a General Interrogation

A Slave is free to send spontaneous data during a General Interrogation Command. However, this Slave must ensure that the last value sent is the correct one, the latest. For example, when a Master starts a General Interrogation Command, the Slave creates a snapshot of all sent data and starts the sending process, which may last a few seconds. During this sending process, if any data changes their status, that change can be sent spontaneously:

- If the Slave already sent data in the General Interrogation Command, it can send a change spontaneously without problems
- If it has not sent data yet, it can:
  - Wait for sending the change spontaneously at the end of the General Interrogation Command
  - Send the change spontaneously and then send the same information, already with an updated value, as part of the General Interrogation Command

Some Slaves implement that sequence incorrectly, sending spontaneous changes during a General Interrogation Command, but without ensuring that the sent interrogation data after the spontaneous value is updated. For example, at the beginning of a General Interrogation Command an Analog Measurement had a value of 20 but, during the general interrogation , before that Measurement could be sent, the value of the Measurement changed to 30 and this value was sent spontaneously. Later in the General Interrogation Command, the Slave sends the value 20, which was the value of the Measurement at the beginning of the General Interrogation Command, so that this Driver keep this last value, 20, as it was the current value of the Measurement.

To circumvent this problem in the Slave, users can enable the **Interrogation Process At End** option. When this option is selected, interrogation data, that is, cause of transmission equal to 20, received during a General Interrogation Command are stacked by this Driver until the end of the interrogation process. Then, this Driver excludes from stacked data any point that received spontaneous values during the interrogation. If this option is not selected, interrogation data received during the process are processed immediately. As this option does not cause collateral problems, except a delay to process interrogation data, users can keep it always selected.

NOTE

Spontaneous data received during a General Interrogation Command is always processed immediately by this Driver.

Counter Interrogation Command

A **Counter**, called *Integrated Total* in IEC documentation, is a value accumulator. The way a Counter works is **Incremental**: The Slave memorizes, or freezes, the Counters from time to time and transmits these Counters to the Master. The Counters keep accumulating values without being affected by this freezing operation. The Master, in this case, is responsible for calculating the difference between the values returned by the Slave to find out how much was incremented in each interval or **With Restart**: The Slave memorizes, or freezes, the Counters from time to time and restarts the Counter's values to 0 (zero). The frozen value is then sent to the Master.

Usually, the moment of freezing is controlled by the Slave. Optionally, the Master can assume this role, by sending **Counter Interrogation Commands** or **ASDU 101**, which in a Driver in Master mode is a Tag with the **N2** parameter equal to 3 (three).

Each device can have several groups of Integration Totals, which allows different integration periods for each group of Counters. This configuration, that is, which Counters belong to which groups, is managed only by the device.

Tag for Sending a Counter Interrogation

Protocols	**IEC 870-5-104 MASTER**
Numeric Parameters	**N1**: Station number, **N2**: 3 (three), **N3**: Group of Counters, and **N4**: Operation
Parameters by String	Not available
Writing	Write any value to this Tag to send an Interrogation of Counter Groups Command to a Slave
Reading	Not supported
ASDUs	**C_CI_NA_1** (101)

Any value written to this Tag sends a Counter Interrogation Command to a Slave. If this Command is successful, this writing returns the value 1 (one). If this Command fails for any reason, this writing returns the value 0 (zero). The **N3** parameter defines the number of the affected Group of Counters. Configure this parameter with the value 0 (zero) to specify no group, which may have a special meaning, depending on the device. The **N4** parameter defines the operation to perform:

- **0**: Interrogation only, without freezing or restarting
- **1**: Freezing Counters without restarting
- **2**: Freezing Counters with restarting
- **3**: Restarting Counters

As a response to an Interrogation Command, the Slave sends memorized data from Counters of the group specified in the Command. The ASDUs for Counters' data are the following:

- **M_IT_NA_1** (ASDU 15, integrated totals)
- **M_IT_TA_1** (ASDU 16, integrated totals with short timestamp)
- **M_IT_TB_1** (ASDU 37, integrated totals with full timestamp)

Data returned by an Interrogation Command can be read by the Master application using Tags for reading cache memory, that is, the **N2** parameter equal to 5 (five), the **N3** parameter with the number of any ASDU, that is, 15, 16, or 37, and in the **N4** parameter the Counter's address. When using a configuration by **Strings**, configure in the **ParamItem** field the value "ITxxx", in which *xxx* is the Counter's address.

Select or Execute Command

The **Select or Execute Commands** allow a Master to send information to a Slave.

Usually, a Slave requires a **Select Command** to be sent first, or several Select Commands to different objects, the moment in which several interlocks are checked by this Slave.

If the Select Commands are successful, then the Master can send the **Execute Commands** when the Command is performed.

The Master can also send **Cancel Select Commands** to cancel Select Commands previously sent.

There are Slaves that also allow directly sending **Execute Commands** without a previous selection.

Tag for Sending a Select Command

Protocols	**IEC 870-5-104 MASTER**
Numeric Parameters	**N1**: Station number, **N2**: 10 (Normal), 20 (Short Circuit Pulse), 30 (Long Circuit Pulse), or 40 (Persistent Output), **N3**: Object's type (45-51, 58-64), and **N4**: Command's address
Parameters by String	Please check topic **Command Configuration by Strings**
Writing	Write any value to this Tag to send a Select Command to a Slave
Reading	Not supported
ASDUs	• **Defined in protocol IEC 870-5-101**: **C_SC_NA_1** (45), **C_DC_NA_1** (46), **C_RC_NA_1** (47), **C_SE_NA_1** (48), **C_SE_NB_1** (49), **C_SE_NC_1** (50), and **C_BO_NA_1** (51) • **Defined in protocol IEC 870-5-104**: **C_SC_TA_1** (58), **C_DC_TA_1** (59), **C_RC_TA_1** (60), **C_SE_TA_1** (61), **C_SE_TB_1** (62), **C_SE_TC_1** (63), and **C_BO_TA_1** (64)

Write a value in this Tag to send a Select Command to a Slave. If this Command is successful, later users can send an Execute Command or a Cancel Select Command. The **N2** parameter defines the type of Command to send:

- **10**: Normal Command
- **20**: Short Circuit Pulse
- **30**: Long Circuit Pulse
- **40**: Persistent Output

The **N3** parameters defines the type of object to command. The next types of objects are sent without a timestamp, and defined in protocol **IEC 870-5-101**:

- **45**: **Simple Digital Point** Command
- **46**: **Double Digital Point** Command
- **47**: **Step Position** Command
- **48**: Setpoint Command (**Normalized Analog Measurement**)
- **49**: Setpoint Command (**Scaled Analog Measurement**)
- **50**: Setpoint Command (**Floating-Point Analog Measurement**)
- **51**: **Bitstring** Command

The next types of objects are sent with a full timestamp, and defined in protocol **IEC 870-5-104**:

- **58**: **Simple Digital Point** Command with Timestamp
- **59**: **Double Digital Point** Command with Timestamp
- **60**: **Step Position** Command with Timestamp
- **61**: Setpoint Command (**Normalized Analog Measurement with Timestamp**)
- **62**: Setpoint Command (**Scaled Analog Measurement with Timestamp**)
- **63**: Setpoint Command (**Floating-Point Analog Measurement with Timestamp**)
- **64**: **Bitstring** Command with Timestamp

The value sent in a Command must comply with the type of object indicated in the **N3** parameter. If this Command was successful, this writing returns 1 (one). If any failure happened when sending this Command, this writing returns 0 (zero).

NOTES

- **Bitstring** Commands, 51 and 64, do not support sending Short Circuit, Long Circuit, and Persistent Output qualifiers. These Commands, in these cases, are sent to a Slave without specifying a qualifier. If the Slave is from another IEC870 Driver, this one receives the Command with a Normal Command qualifier.
- **Bitstring** Commands, 51 and 64, do not have Select or Execute information. Both Commands are accepted by this Driver, but a Command sent to a Slave does not indicate whether it is a Select or Execute Command. If the Slave is from another IEC870 Driver, this one receives the Command always as an Execute Command.

Tag for Sending an Execute Command

Protocols	**IEC 870-5-104 MASTER**
Numeric Parameters	**N1**: Station number, **N2**: 11 (Normal), 21 (Short Circuit Pulse), 31 (Long Circuit Pulse), or 41 (Persistent Output), **N3**: Object's type (45-51, 58-64), and **N4**: Command's address
Parameters by String	Please check topic **Command Configuration by Strings**
Writing	Write any value to this Tag to send an Execute Command to a Slave
Reading	Not supported
ASDUs	• **Defined in protocol IEC 870-5-101**: **C_SC_NA_1** (45), **C_DC_NA_1** (46), **C_RC_NA_1** (47), **C_SE_NA_1** (48), **C_SE_NB_1** (49), **C_SE_NC_1** (50), and **C_BO_NA_1** (51) • **Defined in protocol IEC 870-5-104**: **C_SC_TA_1** (58), **C_DC_TA_1** (59), **C_RC_TA_1** (60), **C_SE_TA_1** (61), **C_SE_TB_1** (62), **C_SE_TC_1** (63), and **C_BO_TA_1** (64)

Write a value to this Tag to send an Execute Command to a Slave. If that Command is successful, this writing returns 1 (one) and in case of failure returns 0 (zero). Many devices require an Execute Command to be preceded by a Select Command containing the same object and value. The **N2** parameter defines the type of Command to send:

- **11**: Normal Command
- **21**: Short Circuit Pulse
- **31**: Long Circuit Pulse
- **41**: Persistent Output

The **N3** parameter defines the type of object to command. The next types of objects are sent without a timestamp, and defined in protocol **IEC 870-5-101**:

- **45**: **Simple Digital Point** Command
- **46**: **Double Digital Point** Command
- **47**: **Step Position** Command
- **48**: Setpoint Command (**Normalized Analog Measurement**)
- **49**: Setpoint Command (**Scaled Analog Measurement**)
- **50**: Setpoint Command (**Floating-Point Analog Measurement**)
- **51**: **Bitstring** Command

The next types of objects are sent with a full timestamp, and defined in protocol **IEC 870-5-104**:

- **58**: **Simple Digital Point with Timestamp** Command
- **59**: **Double Digital Point with Timestamp** Command
- **60**: **Step Position with Timestamp** Command

- **61**: Setpoint Command (**Normalized Analog Measurement with Timestamp**)
- **62**: Setpoint Command (**Scaled Analog Measurement with Timestamp**)
- **63**: Setpoint Command (**Floating-Point Analog Measurement with Timestamp**)
- **64**: **Bitstring with Timestamp** Command

The value sent in this Command must comply with the type of object indicated in the **N3** parameter. Users are allowed in protocol **IEC 870-5-101** to send Commands defined in protocol **IEC 870-5-104** and viceversa, as long as the Slave support those Commands.

Tag for Sending a Cancel Select Command

Protocols	**IEC 870-5-104 MASTER**
Numeric Parameters	**N1**: Station number, **N2**: 12 (Normal), 22 (Short Circuit Pulse), 32 (Long Circuit Pulse), or 42 (Persistent Output), **N3**: Object's type (45-51, 58-64), and **N4**: Command's address
Parameters by String	Please check topic **Tag Configuration by Strings**
Writing	Write any value to this Tag to cancel a Select Command sent previously to a Slave
Reading	Not supported
ASDUs	• **Defined in protocol IEC 870-5-101**: **C_SC_NA_1** (45), **C_DC_NA_1** (46), **C_RC_NA_1** (47), **C_SE_NA_1** (48), **C_SE_NB_1** (49), **C_SE_NC_1** (50), and **C_BO_NA_1** (51) • **Defined in protocol IEC 870-5-104**: **C_SC_TA_1** (58), **C_DC_TA_1** (59), **C_RC_TA_1** (60), **C_SE_TA_1** (61), **C_SE_TB_1** (62), **C_SE_TC_1** (63), and **C_BO_TA_1** (64)

Write a value to this Tag to send a Cancel Command for a previously Select Command sent to a Slave. If this Command is successful, this writing returns 1 (one) and if it fails for any reason, this writing returns 0 (zero). The **N2** parameter defines the type of Command to cancel:

- **12**: Normal Command
- **22**: Short Circuit Pulse
- **32**: Long Circuit Pulse
- **42**: Persistent Output

The **N3** parameter defines the type of object whose Select Command is being canceled. The next types of objects are sent without a timestamp, defined in protocol **IEC 870-5-101**:

- **45**: **Simple Digital Point** Command
- **46**: **Double Digital Point** Command
- **47**: **Step Position** Command
- **48**: Setpoint Command (**Normalized Analog Measurement**)

- **49**: Setpoint Command (**Scaled Analog Measurement**)
- **50**: Setpoint Command (**Floating-Point Analog Measurement**)
- **51**: **Bitstring** Command

The next types of objects are sent with a full timestamp, defined in protocol **IEC 870-5-104**:

- **58**: **Simple Digital Point with Timestamp** Command
- **59**: **Double Digital Point with Timestamp** Command
- **60**: **Step Position with Timestamp** Command
- **61**: Setpoint Command (**Normalized Analog Measurement with Timestamp**)
- **62**: Setpoint Command (**Scaled Analog Measurement with Timestamp**)
- **63**: Setpoint Command (**Floating-Point Analog Measurement with Timestamp**)
- **64**: **Bitstring with Timestamp** Command

The value sent in this Command must comply with the type of object indicated in the **N3** parameter. Users are allowed in protocol **IEC 870-5-101** to send Commands defined in protocol **IEC 870-5-104** and viceversa, as long as the Slave supports these Commands.

Process Restart Command

A Master can request the restart of a Slave's process by sending a **Process Restart Command**.

Tag for Sending a Process Restart Command

Protocols	**IEC 870-5-104 MASTER**
Numeric Parameters	**N1**: Station number, **N2**: 1105, **N3**: 0 (zero), and **N4**: 0 (zero)
Parameters by String	Not available
Writing	Write any value to this Tag to request the restart of a Slave's process
Reading	Not supported
ASDUs	**C_RP_NA_1** (105)

The value written to this Tag is a **Qualifier of a Process Restart Command**, an integer between 0 (zero) and 255:

- **0**: Not used
- **1**: General restart of a process
- **2**: Removes timestamp information pending on the event buffer
- **3 to 127**: Reserved for future definitions of the IEC standard
- **128 to 255**: Reserved for specific uses of each manufacturer

If this Command is successful, this writing returns 1 (one) and if it fails for any reason, returns 0 (zero).

Test Command

A **Test Command** allows validating how the application layer of the selected protocol works.

This Driver in **Master** mode can be configured to send a Test Command automatically, whenever the application layer remains a certain time without being used.

To enable sending a Test Command, enable the options on the **Test Command** group in the **Master** option of the **Properties** tab on this Driver's properties window.

In protocol **IEC 870-5-104**, the ASDU sent is **C_TS_TA_1** (107). If this Command fails, the connection is closed and reopened.

Slave 104

In **Slave** mode, this Driver supports the functions described on the next topics.

NOTE
The default TCP/IP port for **IEC 870-5-104** protocol is port 2404.

Shutdown Tag

Protocol	Only **IEC 870-5-104 SLAVE**
Numeric Parameters	**N1**: 0, **N2**: 990, **N3**: 0 (zero) and **N4**: 0 (zero)
Parameters by String	Not available
Writing	When writing the value 1, it tells to this driver that the data source is being shut down, initiating the shutdown mode of the slave channel 104. In this mode, tag write operations with a NULL value will be temporarily blocked during the interval specified in the Cache Startup/Shutdown Mode Time (s) property. If, after this interval, the 104 slave driver is still running, the tags that continue with the NULL value will be reported with poor quality to the IEC 104 Master.
Reading	Not available
ASDUs	Not available

Sending Monitoring Data

Sending Monitoring Data, digital and analog, among others, to a Master can occur as a General Interrogation Command, as a Spontaneous Sending with high priority or as a Spontaneous Sending with low priority.

The **General Interrogation Command** is a Command started by a Master from time to time. As a response to this General Interrogation Command, a Slave sends the current value of a set of pre-configured points.

The spontaneous Sending allows a Slave to inform a Master about value changes in objects, without the Master sending a Command for this.

The application is responsible for feeding Slave values by writing to Tags. A aplicação é responsável por alimentar os valores do Escravo através de escritas em Tags. It is a common situation that, during the application initialization, several

writings are performed to a Slave to feed the initial status of all Tags. During this period, it may be necessary **to turn off the link layer**, to prevent incorrect or incomplete responses to a Master.

Tag for Writing Events on the Cache Memory

Protocols	**IEC 870-5-104 SLAVE**
Numeric Parameters	**N1**: Station number, **N2**: 4 (four), 5 (five), 6 (six), or 7 (seven), **N3**: ASDU, and **N4**: Object's address
Parameters by String	Please check topic **Tag Configuration by Strings**
Writing	Loads in this Driver's internal cache memory, that is, in the sending queues, the value of a point
Reading	Returns the current value of this Driver's internal cache memory
ASDUs	• **Defined in protocol IEC 870-5-101**: **M_SP_NA_1** (1), **M_SP_TA_1** (2), **M_DP_NA_1** (3), **M_DP_TA_1** (4), **M_ST_NA_1** (5), **M_ST_TA_1** (6), **M_BO_NA_1** (7), **M_BO_TA_1** (8), **M_ME_NA_1** (9), **M_ME_TA_1** (10), **M_ME_NB_1** (11), **M_ME_TB_1** (12), **M_ME_NC_1** (13), **M_ME_TC_1** (14), **M_IT_NA_1** (15), **M_IT_TA_1** (16), **M_PS_NA_1** (20), **M_ME_ND_1** (21), and **M_EI_NA_1** (70) • **Defined in protocol IEC 870-5-104**: **M_SP_TB_1** (30), **M_DP_TB_1** (31), **M_ST_TB_1** (32), **M_BO_TB_1** (33), **M_ME_TD_1** (34), **M_ME_TE_1** (35), **M_ME_TF_1** (36), and **M_IT_TB_1** (37)

Values written to this Tag are stored on this Driver's internal cache memory and reported to a Master the following way:

- **N2 equal to 4 (four) or N2 equal to 7 (seven)**: The written values are reported only in the General Interrogation Command
- **N2 equal to 5 (five)**: The written values are sent immediately to a Master by a Class 2 (two, low priority) Spontaneous Sending and can also be reported in the General Interrogation Command
- **N2 equal to 6 (six)**: The written values are sent immediately to a Master by a Class 1 (one, high priority) Spontaneous Sending and can also be reported in the General Interrogation Command

This Tag returns events stored on this Driver's internal cache memory. After an event is returned, it is removed from the cache memory. If the cache memory is empty, no value is returned to this Tag, that is, a reading does not affect the value stored in this Tag. This Tag's **OnRead** event is activated whenever a new value is received and can be used to store events in a historical file, for example. Reading and writing this Tag does not cause any type of communication. This Tag does not support block readings or writings.

On a data Spontaneous Sending, Class 1 (one) or Class 2 (two), the ASDU used for sending data is exactly the one specified in the **N3** parameter. For example, when using the **N3** parameter equal to 4 (four), an ASDU **M_DP_TA_1** (Double Point with a 24-bit Timestamp) is sent to the Slave.

The ASDU used for sending data in a General Interrogation Command can be different from the one configured in the **N3** parameter if the **Remove timestamp from GI data** option is enabled. In this case, data loaded in this Driver with the **N3** parameter equal to 4 (four) can be sent in the General Interrogation Command with ASDU **M_DP_NA_1** (Double Point without Timestamp).

Starting this Driver with Link Layer Disabled

Initial values served by a Slave must be written by an application during initialization. To prevent a Master from connecting to this Slave before loading the initial value of all Tags, start this Driver with the link layer disabled:

1. On this Driver's properties window, select the **Properties** tab and, on the **Link Layer** group, configure the **Link Layer Starts** option with the value **Disabled**.

2. When starting the application, after the value of all Tags is loaded using writings, write the value 1 (on) to the **Tag for Enabling the Link Layer**.

Tag for Enabling Link Layer

Protocols	**IEC 870-5-104 SLAVE**
Numeric Parameters	**N1**: 0 (zero), **N2**: 996, **N3**: 0 (zero), and **N4**: 0 (zero)
Parameters by String	Not available
Writing	The value 0 (zero) disables the Link Layer, preventing this Driver from establishing connections with a Master. The value 1 (one) disables the Link Layer
Reading	Returns 1 (one) if the Link Layer is enabled and 0 (zero) if it is disabled
ASDUs	None

This Tag for reading and writing allows turning on and off this Driver's Link Layer. With the Link Layer disabled, this Driver accepts connections from a Master but closes that connection as soon as it receives any message.

Cyclic Sending Analog Data

This Driver, in **Slave** mode, can be programmed to send two sets of data at cyclic intervals, such as at every 5 (five) seconds.

To configure sending cyclic data, open this Driver's properties window, select the **Properties** tab and, on the **Cyclic** group, configure the following options:

- **Cyclic Data Transmission (1): Send Period (ms)**: Defines the interval between sending cyclic data set 1 (one), in milliseconds

- **Cyclic Data Transmission (1): Object Range**: Defines data to send cyclically in set 1 (one). It must follow the same data syntax of a General Interrogation Command. Leave this option blank to send no set 1 (one). For example, the interval "M_ME_TC_1:1-10; M_ME_NA_1:15-20, 24" sends points from 1 (one) to 10 with type **M_ME_TC_1** (Floating-Point Analog Measurement with a 24-bit Timestamp) and points from 15 to 20 plus point 24 with type **M_ME_NA_1** (Normalized Analog Measurement without a Timestamp)

- **Cyclic Data Transmission (2): Send Period (ms)**: Defines the interval between sending cyclic data set 2 (two), in milliseconds

- **Cyclic Data Transmission (2): Object Range**: Defines data to send cyclically in set 2 (two). It must follow the same syntax of a General Interrogation Command

Cyclic data is sent with a Cause of Transmission 1 (one) and low priority. If a Tag is configured for sending cyclically, automatically this Driver in **Slave** mode captures all writings to this Tag, even if writings asking the Tag to be sent immediately with high priority. Cyclic data is always sent in the programmed interval, even when the value changes.

Local Forcing

An application can, at any time, force a specific value for each Tag. Regardless of other values written later to the same Tag, this Driver in **Slave** mode reports to a Master always a forced value, until an application clears that forced situation.

Forced Activation

To activate a local forced activation, write an array with two elements in the **Tag for Writing Events on the Cache Memory**. The first element of that array must contain the text "FORCE" and the second element must contain the value to force, according to the next example.

```
'Forces value 1 on tag001
tag001.WriteEx Array("FORCE", 1)
```

The forced value is sent to a Master using the same method for Tag's normal values, that is, only in a General Connection Command, with high or low priority, with the **SB** flag turned on, and with computer's current timestamp.

While a Tag contains a forced value, it keeps receiving writings from other non-forced values. In this case, this Driver stores the last value set to a Tag and, when deactivating forced values, this last value is sent to a Master. Users can overwrite a forced value with another forced value.

Forced Deactivation

To clear a forced value, write an array with two elements containing a "CLEAR" text in the first element. The second element of this array is ignored and can have any value, according to the next example.

```
'Clears the value forced to tag001
tag001.WriteEx Array("CLEAR", 0)
```

When canceling forced values, the last non-forced value written to this Driver is sent again to a Master. If there is no value available, then an invalid value with the computer's current timestamp is sent.

Clock Syncing

This Driver supports changing the computer's clock according to Clock Syncing Commands sent by a Master (ASDU 103, **C_CS_NA_1**).

To configure the way this Driver handles clock syncing, configure the **Handle mode for Clock sync command** option on the **Slave** group of the **Properties** tab of this Driver's configuration window the following way:

- **disabled**: This Driver rejects Clock Syncing Commands received, responding them with a Cause of Transmission 71 (negative activation confirmation)
- **pass to the application**: This Driver responds to a Master that a Command was successful, that is, sends a Cause of Transmission 7 (seven, activation confirmation) and let the application handle Clock Syncing
- **handle automatically**: This Driver sets the computer's local time according to a time received from a Master. If that setting fails, returns an error to the Master (Cause of Transmission 71, negative activation confirmation). Otherwise, responds with a Cause of Transmission 7 (seven, activation confirmation) indicating a successful operation

This Driver handles internally Delay Acquisition Commands sent by a Master using the ASDU 106 (**C_CD_NA_1**). The value returned by this Driver in this Command is the same one sent by a Master, that is, internal delays are disregarded when processing this Driver.

Tag for Handling a Clock Syncing Command

Protocols	**IEC 870-5-104 SLAVE**
Numeric Parameters	**N1**: Station number, **N2**: 4 (four, reading data from the cache memory), **N3**: 103, and **N4**: 0 (zero)
Parameters by String	Not supported
Writing	Not supported
Reading	The value of this Tag contains the new time sent by a Master
ASDUs	**C_CS_NA_1** (103)

When this Driver accepts a Clock Syncing Command sent by a Master, the new time is indicated when reading this Tag. If this Driver is configured to pass to the application, configured in the **pass to the application** item of the **Clock sync command** option, the Clock Syncing Commands, then this Tag's **OnRead** event must update the computer's clock.

General Interrogation

This Driver in **Slave** mode responds automatically to General Interrogation Commands sent by a Master. The data set returned in this Driver's response can be configured on this Driver's properties window, on the **General Interrogation** option of the **Slave** group of the **Properties** tab:

- **Interrogation Enable Range**: Enables sending a pre-defined list of data
- **List of Data Objects returned by General Interrogation**: Select this option to force this Driver to always return a pre-defined list of data as a response to a General Interrogation Command. If this option is not selected, this Driver returns all values stored on the internal cache memory

The list of points is composed by one or more objects with the following syntax:

```
<type>:<addressList>;<type>:<addressList>;<type>:<addressList>
```

In which:

- **<type>**: Corresponds to the name of an ASDU
  - **SP (M_SP_NA_1, M_SP_TA_1, M_SP_TB_1)**: **Simple Digital Point**
  - **DP (M_DP_NA_1, M_DP_TA_1, M_DP_TB_1)**: **Double Digital Point**
  - **ST (M_ST_NA_1, M_ST_TA_1, M_ST_TB_1)**: **Step Position**
  - **BO (M_BO_NA_1, M_BO_TA_1, M_BO_TB_1)**: **Bitstring**
  - **MEn (M_ME_NA_1, M_ME_TA_1, M_ME_TD_1)**: **Analog Measurement**
  - **MEs (M_ME_NB_1, M_ME_TB_1, M_ME_TE_1)**: **Analog Measurement**
  - **MEf (M_ME_NC_1, M_ME_TC_1, M_ME_TF_1)**: **Analog Measurement**
  - **IT (M_IT_NA_1, M_IT_TA_1, M_IT_TB_1)**: **Integrated Total**
  - **EPa (M_EP_TA_1, M_EP_TD_1)**: Event of Protection Equipment
  - **EPb (M_EP_TB_1, M_EP_TE_1)**: Packed Start Events of Protection Equipment

- **EPc (M_EP_TC_1, M_EP_TF_1)**: Packed Output Circuit Information of Protection Equipment

- **<addressList>**: A list with object addresses, separated by commas. It also allows contiguous addresses separated by dashes, such as "1,2,3,10-15", that is, addresses 1 (one), 2 (two), and 3 (three) and from 10 to 15

For example, to send simple points with address 100 to 119 and double points with address 200, 205, and 210 to 219, use the next syntax.

```
SP:100-119;DP:200,205,210-219
```

- **Remove timestamp from data sent on General Interrogation**: If this option is selected, data returned by a General Interrogation Command is converted to an ASDU without timestamp, such as data from ASDUs 2 (two) and 30 are sent with ASDU 1 (one). If this option is not selected, data is returned using the same ASDU with which it was loaded in this Driver. For more information, please check topic **Tag for Writing Events on the Cache Memory**

Select or Execute Command

When receiving a **Select or Execute Command**, this Driver may have the following behavior:

1. **Deny the Command (disabled)**: This Driver responds to a Master with a Cause of Transmission 135 (Negative activation confirmation).

2. **Notify the application and wait for a Command response from it (pass to application)**: This Driver notifies the application by reading the **Block Tag for Handling a Select or Execute Command**, delegating to the application the responsibility for generating answers to a Master.

3. **Handle the Command automatically and notify the application (handle automatically)**: This Driver responds to a Master as if the Command was successfully executed and notifies the application using the **Block Tag for Handling a Select or Execute Command**, but in this case the application does not need to generate answers to a Master.

This configuration is performed on this Driver's properties window, in the **Command Handling** option of the **Slave** group on the **Properties** tab.

Block Tag for Handling a Select or Execute Command

Protocols	**IEC 870-5-104 SLAVE**
Numeric Parameters	**B1**: Station number, **B2**: 4 (four, reading data from the cache memory), **B3**: ASDU (45 to 51, 58 to 64), and **B4**: Object's address
Parameters by String	Not available
Block Size	Four Elements
Writing	Sends to a Master the answer of a Select or Execute Command
Reading	Receives the Select or Execute Commands sent by a Master
ASDUs	**C_SC_NA_1** (45), **C_DC_NA_1** (46), **C_RC_NA_1** (47), **C_SE_NA_1** (48), **C_SE_NB_1** (49), **C_SE_NC_1** (50), **C_BO_NA_1** (15), **C_SC_TA_1** (58), **C_DC_TA_1** (59), **C_RC_TA_1** (60), **C_SE_TA_1** (61), **C_SE_TB_1** (62), **C_SE_TC_1** (63), and **C_BO_TA_1** (64)

The **B3** parameter can have one of the following values:

- **45 (C_SC_NA_1)**: Single Command
- **46 (C_DC_NA_1)**: Double Command
- **47 (C_RC_NA_1)**: Regulating Step Command
- **48 (C_SE_NA_1)**: Setpoint Command, Normalized Value
- **49 (C_SE_NB_1)**: Setpoint Command, Scaled Value
- **50 (C_SE_NC_1)**: Setpoint Command, Short Floating Point Number
- **51 (C_BO_NA_1)**: 32-bit Bitstream
- **58 (C_SC_TA_1)**: Single Command (with 7-octet timestamp)
- **59 (C_DC_TA_1)**: Double Command (with 7-octet timestamp)
- **60 (C_RC_TA_1)**: Regulating Step Command (with 7-octet timestamp)
- **61 (C_SE_TA_1)**: Setpoint Command, Normalized Value (with 7-octet timestamp)
- **62 (C_SE_TB_1)**: Setpoint Command, Scaled Value (with 7-octet timestamp)
- **63 (C_SE_TC_1)**: Setpoint Command, Short Floating Point Number (with 7-octet timestamp)
- **64 (C_BO_TA_1)**: 32-bit Bitstream (with 7-octet timestamp)

When this Driver receives one of the previously described Commands, this Block Tag receives a Block with information about this Command. On this Block Tag's **OnRead** event, an application must handle that information received, changing the Block Elements needed and writing the Block back to this Driver, so that this Driver can send a response to a Master. This Block Tag's Elements are the following:

- **0**: Command value, depending on the type of Command
- **1**: Cause of Transmission (**6**: Activate or **8**: Deactivate)
- **2**: Select or Execute (**0**: Execute or **1**: Select)
- **3**: Command quality (**0**: Undefined, **1**: Short Circuit Pulse, **2**: Long Circuit Pulse, or **3**: Persistent Output)

When receiving an indication of an Activation Command, that is, the Element 1 (one) equal to 6 (six), the application must respond immediately with one of the following values:

- A positive Cause of Transmission **ACTCON** if this Command can proceed, that is, the Element 1 (one) equal to 7 (seven)
- A negative Cause of Transmission **ACTCON** if this Command cannot proceed, that is, the Element 1 (one) equal to 135 (128 + 7)

If a positive Cause of Transmission **ACTCON** is sent, the application must check Element 2 (two, Select or Execute) and execute the appropriate action.

If an Execute Command is requested, the application must execute that Command and then send a Cause of Transmission **ACTTERM**, that is, the Element 1 (one) equal to 10 (activation termination), to indicate that this Command was executed. The next example shows an **OnRead** event for this Block Tag in **E3** or **Elipse Power**.

```
' CMDIndication is a Block Tag with size 4. This Tag's parameters are:
' AdviseType     = 0 (AlwaysInAdvise)
' AllowRead      = True
' AllowWrite     = False
' B1             = <station address>
' B2             = 4
' B3             = 45 to 51, 58 to 64
' B4             = <command point address>
' EnableDeadband = False
' Size           = 4

Sub CMDIndication_OnRead()
  Dim CMD, SE, COT, QU
  CMD = Item(1).Value ' Command value (depends on the command)
  COT = Item(2).Value ' COT (cause of transmission)
  SE  = Item(3).Value ' 1=Select, 0=Execute
  QU  = Item(4).Value ' Quality of command

  Select Case COT
    Case 6
      ' ACT(6) must be answered with an ACTCON(7)
      WriteEx Array(CMD, 7, SE, QU), TimeStamp

      If SE = 0 Then
        ' TODO: Execute here the command, probably
        ' a writing to another Driver

        ' Sends an ACTTERM(10)
        WriteEx Array(CMD, 10, SE, QU), TimeStamp 'ACTTERM
      End If
    Case 8
      ' DEACT(8) must be answered with a DEACTCON(9)
      WriteEx Array(CMD, 9, SE, QU), TimeStamp
  End Select
End Sub
```

NOTES

- Writings on this Block Tag fail if the handling mode of Select or Execute Commands is not configured as **Pass to the Application**.
- **Bitstring** Commands (51 and 64) do not support sending Short Circuit Pulse, Long Circuit Pulse, or Persistent Output qualifiers. In these cases the fourth Element of this Block (QU) is always 0 (zero), that is, a qualifier for a Normal Command.
- **Bitstring** Commands (51 and 64) do not have Select or Execute information. In these cases, the third Element of this Block (SE) always receives 0 (zero, Execute).

Process Restart Command

This Driver in **Slave** mode accepts and responds automatically with an **ACK** Cause of Transmission to Process Restart Commands sent by a Master (ASDU **C_RP_NA_1**), and right after sends and indication so that an application can, optionally, **handle that Command**.

Tag for Handling a Process Restart Command

Protocols	**IEC 870-5-104 SLAVE**
Numeric Parameters	**N1**: Station number, **N2**: 4 (four, reading data from the cache memory), **N3**: 105, and **N4**: 0 (zero)
Parameters by String	Not supported
Writing	Not supported
Reading	The value of this Tag contains a process restart qualifier sent by a Master, that is, an integer between 0 (zero) and 255
ASDUs	**C_RP_NA_1** (105)

When this Driver accepts a Process Restart Command sent by a Master, the qualifier sent by this Master in this Command is flagged in this Tag's reading. An application can, when receiving a reading value from this Tag, force via script a Driver restart. To check a list of qualifiers for process restart defined in protocol **IEC 870-5-104**, please check **Tag for Sending a Process Restart Command**.

Test Command

Test Commands sent by a Master are handled automatically by this Driver in **Slave** mode. This Driver supports ASDUs **C_TS_NA_1** (104) and **C_TS_TA_1** (107).

Technical Articles

This section contains miscellaneous technical features of IEC870-104 Driver.

Internal Cache Memory

An **Internal Cache Memory** is a memory of this Driver that stores both data received from a device and data sent to a device.

Cache Memory of Received Data

The ASDUs of data received from a device are stored on the internal cache memory, grouped by type and address. To read data stored on the cache memory, declare a **Tag for Reading Events from Cache Memory**, informing the type of ASDU and the object's address.

Cache Memory of Sent Data

To handle data to send, this Driver in **Slave** mode keeps the following memory areas:

- **Cache**: Keeps the current value of an object
- **Queue of events with high priority**: Contains a list of events to send to a Master with high priority
- **Queue of events with low priority**: Contains a list of events to send to a Master with low priority
- **Queue of events with responses from Commands**: Contains a list of results of Commands to send to a Master

When a connection to a Master is lost, a Slave clears the event queues and keeps only the cache memory with the current value of each object.

This Driver in **Slave** mode can be configured to keep in memory the event queues with high and low priority even in case of disconnection. To enable and configure the size of these queues, configure the **Offline Buffers** option of the **Slave** group on the **Properties** tab of this Driver's configuration window according to the next values:

- **Offline Class 1 buffering and Offline Class 1 Buffer size (events)**: Enables keeping *n* events with high priority (Class 1) while the Slave is disconnected from a Master
- **Offline Class 2 buffering and Offline Class 2 Buffer size (events)**: Enables keeping *n* events with low priority (Class 2) while the Slave is disconnected from a Master

Timestamps

This Driver, whatever the selected protocol, supports sending and receiving data with or without timestamp in the following formats:

- **Without timestamp**: In this case, the timestamp is set by this Driver according to system's current time
- **24-bit timestamp**: This format is defined in protocol **IEC 870-5-101**, but can be used in both protocol **IEC 870-5-101** and protocol **IEC 870-5-104**. Includes only information about hour, minute, second, and millisecond. Date is filled by this Driver using system's current date
- **56-bit timestamp**: This format is defined in protocol **IEC 870-5-104**, but can be used in both protocol **IEC 870-5-104** and protocol **IEC 870-5-101**. Includes a full timestamp, with year, month, day, hour, minute, second, and millisecond

In data received without a timestamp or with an invalid timestamp, that is, with the **IV** flag equal to 1 (one), the timestamp is replaced by the current computer's time and the **value's quality** returned to the application is adjusted to indicate that the timestamp is local. Timestamps received and sent by this Driver can be adjusted to **summer time**, depending on this Driver's configuration and on the **SU** flag of the timestamp.

Summer Time

This Driver can be configured to handle or ignore the **SU** flag of timestamps. This flag, when configured, that is, equal to 1 (one), indicates that a timestamp is in summer time.

To configure the handling of summer time by this Driver, configure the **Timestamp SU Flag (Summertime adjustment)** option of the **Timestamp** group on the **Properties** tab of this Driver's configuration window with the following values:

- **Ignore (default value)**: Ignores the **SU** flag of timestamps received and sent and ignores the computer's summer time configuration. Warning messages are generated in the log whenever this Driver detects timestamps with the **SU** flag non-consistent with the computer's summer time:
  - **When receiving an incoherent timestamp**: The message is "Warning: driver is ignoring mismatching summertime configurations on the PLC (<PlcBias> minutes) and on the server (<LocalBias> minutes). The resulting timestamp may be incorrect!"
  - **When sending an incoherent timestamp**: The message is "Warning: the driver is configured to ignore the summertime flag, but the system currently is operating on summertime (<LocalBias> minutes). The outgoing timestamp may be incorrect!"
- **Automatic**: Handles the **SU** flag of timestamps received and sent and considers the computer's summer time in the following way:
  - When receiving a timestamp with the **SU** flag equal to 1 (one) and if the computer is not in summer time, delays the timestamp in one hour

- When receiving a timestamp with the **SU** flag equal to 1 (one) and if the computer is in summer time, does not change the timestamp
- When receiving a timestamp with the **SU** flag equal to 0 (zero) and if the computer is not in summer time, does not change the timestamp
- When receiving a timestamp with the **SU** flag equal to 0 (zero) and if the computer is in summer time, brings the timestamp forward one hour
- When sending a timestamp, if the computer is in summer time, the **SU** flag is equal to 1 (one)
- When sending a timestamp, if the computer is not in summer time, the **SU** flag is equal to 0 (zero)
- In cases when the computer is in summer time but the summer time configuration is not standard, that is, the bias is different from 60 minutes:
  - When receiving a timestamp with the **SU** flag equal to 1 (one), delays the timestamp in one hour, assuming that in the device the summer time bias is always 60 minutes, and adds the computer's summer time bias
  - When sending a timestamp, delays the timestamp according to the computer's summer time bias and send the timestamp with the **SU** flag equal to 0 (zero)
- In cases when the timestamp is adjusted, a message is generated in the log informing:
  - **When a received timestamp is adjusted**: The message is "Timestamp adjusted: From PLC (<PlcTimestamp> [<PlcBias> minutes]) to Driver (<LocalTimestamp> [<LocalBias> minutes])"
  - **When a timestamp is adjusted for sending**: The message is "Timestamp adjusted: From Driver (<LocalTimestamp> [<LocalBias> minutes]) marked as *summertime*" or "Timestamp adjusted: From Driver (<LocalTimestamp> [<LocalBias> minutes]) to PLC (<PlcTimestamp> [<PlcBias> minutes])"

Time Zone

This Driver assumes that the time zone of the Stations to which it communicates is the same as the computer where it is executing. Timestamps sent by this Driver are always expressed in the local time of this Driver, and not in **UTC** format. This Driver assumes that received timestamps are all expressed in the local time of the Stations, and not in **UTC** format.

Quality

An **OPC Quality** is composed by 8 (eight) bits, divided into 3 (three) fields, described on the next table.

Fields of an OPC quality

STATUS		SUB-STATUS				LIMIT	
B7	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
0: Bad		**0**: Non-specific, **1**: Configuration error, **2**: Not connected, **3**: Device failure, **4**: Sensor failure, **5**: Last known value, **6**: Communication failure, or **7**: Out of service				**0**: Free, **1**: Low, **2**: High, or **3**: Constant	
1: Uncertain		**0**: Non-specific, **1**: Last usable value, **4**: Inaccurate sensor, **5**: Engineering units exceeded, or **6**: Sub-normal					
3: Good		**0**: Non-specific, **1**: Local overwriting, **6**: Local timestamp (defined by this Driver), or **7**: Local overwriting and timestamp (defined by this Driver)					

The quality bits defined in protocol **IEC 870-5-104** are mapped in the **Quality** property of a Tag. The higher bits on the list have greater precedence. For example, if bits **IV** and **NT** are turned on, the bit **IV** is returned. The mapping of bits is the following:

- **IV (invalid)**: If this bit is turned on, then quality is equal to 4 (four, **Status** equal to **Bad**, **Sub-Status** equal to **Configuration Error**, and **Limit** equal to **Free**)
- **NT (non-topic)**: If this bit is turned on and the bit **IV** is turned off, then quality is equal to 8 (eight, **Status** equal to **Bad**, **Sub-Status** equal to **Not Connected**, and **Limit** equal to **Free**)
- **SB (replaced)**: If this bit is turned on and quality is good (**Status** equal to **Good**), then changes **Sub-Status** to **Local Overwriting**, that is, sums 4 (four) to quality
- **BL (blocked)**: If this bit is turned on, adds 3 (three, **Limit** equal to **Constant**) to quality, without changing **Status** and **Sub-Status**
- **OV (overflow)**: If this bit is turned on and the bit **BL** is turned off, adds 2 (two, **Limit** equal to **High**) to quality. If the **Map Overflow bit to Uncertain Quality** option is configured, changes **Status** to Uncertain and **Sub-Status** to Engineering Units exceeded
- **Timestamp IV (ASDU came without timestamp, or if the timestamp had the flag IV (invalid) turned on)**: If quality is good (**Status** equal to **Good**), then changes **Sub-Status** from 0 (zero, **Non-specific**) to 6 (six, **Local timestamp**) or from 1 (one, **Local overwriting**) to 7 (seven, **Local overwriting and timestamp**)

The next table contains the combinations of flags received by this Driver and the resulting quality of data read.

Combinations of flags received by this Driver

IV	NT	SB	BL	OV	TIMESTAMP IV	QUALITY
1	x	x	0	0	x	**4**: Bad / Configuration Error / Free (00 0001 00)
			0	1		**6**: Bad / Configuration Error / High (00 0001 10)
			1	x		**7**: Bad / Configuration Error / Constant (00 0001 11)
0	1	x	0	0	x	**8**: Bad / Not connected / Free (00 0010 00)
			0	1		**10**: Bad / Not connected / High (00 0010 10)
			1	x		**11**: Bad / Not connected /

IV	NT	SB	BL	OV	TIMESTAMP IV	QUALITY
						Constant (00 0010 11)
0	0	0	0	0	0	**192**: Good / Non-Specific / Free (11 0000 00)
			0	1		**194**: Good / Non-Specific / High (11 0000 10)
			1	x		**195**: Good / Non-Specific / Constant (11 0000 11)
0	0	1	0	0	0	**196**: Good / Local Overwriting / Free (11 0001 00)
			0	1		**198**: Good / Local Overwriting / High (11 0001 10)
			1	x		**199**: Good / Local Overwriting / Constant (11 0001 11)
0	0	0	0	0	1	**216**: Good / Local Timestamp / Free (11 0110 00)
			0	1		**218**: Good / Local Timestamp / High (11 0110 10)
			1	x		**219**: Good / Local Timestamp / Constant (11 0110 11)
0	0	1	0	0	1	**220**: Good / Local Overwriting and Timestamp / Free (11 0111 00)

IV	NT	SB	BL	OV	TIMESTAMP IV	QUALITY
			0	1		**222**: Good / Local Overwriting and Timestamp / High (11 0111 10)
			1	x		**223**: Good / Local Overwriting and Timestamp / Constant (11 0111 11)

Inversely, when an application performs writings to this Driver, the quality from a Tag is transformed into the corresponding bits of the protocol according to the previous table, only replacing the *x* from the table by the value 0 (zero). For example, a quality 4 (four) results in bit **IV** equal to 1 (one), in bit **NT** equal to 0 (zero), in bit **SB** equal to 0 (zero), in bit **BL** equal to 0 (zero), in bit **OV** equal to 0 (zero), and in bit **Timestamp IV** equal to 0 (zero). The next exceptions must be observed in the quality of writings:

- Any **Good** quality not listed on the previous table is mapped with all flags in 0 (zero), except for flags **BL** and **OV**, which can always be extracted from quality
- Any **Bad** or **Uncertain** quality not listed on the previous table is mapped with the flag **IV** equal to 1 (one) and all other flags in 0 (zero), also except for flags **BL** and **OV**
- Sending invalid timestamps by this Driver is not supported, the timestamp sent is always valid, that is, the flag **IV** is equal to 0 (zero). Then, Sub-Status **Local Timestamp** (qualities 216, 218, and 219) and **Local Overwriting and Timestamp** (qualities 220, 222, and 223) are remapped to **Non-Specific Sub-Status** (qualities 192, 194, and 195)

Offline Configuration

In situations in which users must configure this Driver at run time, this Driver can be set to **Offline** mode and change the following parameters:

- **Individually**: Declare a Tag with the **ParamItem** property containing the name of the property to change, without quotation marks, and write the new value of the property to this Tag
- **Together**: Use the **Set Configuration Parameters** Tag of **IOKit**, that is, the **B1** parameter equal to -1 (minus one), the **B2** parameter equal to 0 (zero), the **B3** parameter equal to 0 (zero), and the **B4** parameter equal to 3 (three), and write to this Block Tag an array in which each two Elements contains the name of a property and its respective value

Causes of Transmission

A **Cause of Transmission** is an information that is part of an ASDU sent or received in protocol **IEC 870-5-104**, and indicates the reason why that ASDU is being sent or requested. The most common values for Cause of Transmission are listed on the next table.

Values for Cause of Transmission in protocol IEC 870-5-104

MEANING	CAUSE OF TRANSMISSION	NEGATIVE TRANSMISSION (SUM 64)	TEST TRANSMISSION (SUM 128)
Cyclic sending (CYC)	1 (one)	65	129

MEANING	CAUSE OF TRANSMISSION	NEGATIVE TRANSMISSION (SUM 64)	TEST TRANSMISSION (SUM 128)
Background sending (BACK)	2 (two)	66	130
Spontaneous sending (SPONT)	3 (three)	67	131
Initialization (INIT)	4 (four)	68	132
Requested (REQ)	5 (five)	69	133
Activation (ACT)	6 (six)	70	134
Activation confirmation (ACTCON)	7 (seven)	71	135
Deactivation (DEACT)	8 (eight)	72	136
Deactivation confirmation (DEACTCON)	9 (nine)	73	137
Activation finished (ACTTERM)	10	74	138
Return information caused by a remote command (RETREM)	11	75	139
Return information caused by a local command (RETLOC)	12	76	140
File Transfer (FILE)	13	77	141
Interrogated by a general interrogation (GENINT)	20	84	148
Interrogated by an interrogation from group 1 to 6 (INRO1 to INRO16)	21 to 36	85 to 100	149 to 164
Requested by a general counter interrogation (REQCOGEN)	37	101	165
Requested by an interrogation from group 1 to 4 of counters (REQCO1 to REQCO4)	38 to 41	102 to 105	166 to 169
Identification type (ASDU) unknown (UNKTYPE)	44	108	172
Cause of transmission unknown (UNKCOT)	45	109	173
Common ASDU address unknown (UNKCOMMONADDR)	46	110	174
Information object address unknown (UNKOBJADDR)	47	111	175

Reference of Driver Properties

The next table lists all properties of IEC870-104 Driver. These properties are available for configuration on the **Properties** tab. Users can also configure these properties at run time. For more information, please check topic **Offline Configuration**.

Properties of IEC870-104 Driver

PROPERTY	DEFAULT VALUE	COMMENTS
Protocol	104	Configures the protocol to used by this Driver. This property is stored as "IEC.Protocol". Possible values are **104**: Protocol **IEC 870-5-104 Master** or **1005**: Protocol **IEC 870-5-104 Slave**
Link Layer - Fixed Station List	Empty **String**	Fixed list of Stations. Stored as "IEC.FixedStationList"
Link Layer - Default Slave Address	0 (zero)	Determines the identifier of a Station to use when the **N1** or **B1** parameter is equal to 0 (zero). Stored as "IEC.DefaultSlaveAddress"
Link Layer - Originator Address for outgoing ASDUs	0 (zero)	Value of the **Originator Address** field in all ASDUs sent by this Driver. This field is only sent if the number of octets of the Cause of Transmission is equal to 2 (two). Stored as "IEC.OriginatorAddress"
Link Layer - Link Layer Starts	0 (zero)	Determines whether the **link layer** of this Driver starts enabled or disabled. Stored as "IEC.LinkLayerStartsDisabled". Possible values are **0**: Link layer starts enabled or **1 or any other value different from 0**: Link layer starts disabled
Octets - Octets (Cause Of Transmission)	1 (one)	Number of octets (bytes) of the **Cause of Transmission** field of packet FT 1.2. Stored as "IEC.Octets.COT". Possible values are **1**: One octet or **2**: Two octets, one for each **Cause of Transmission** and one for the **Originator Address**
Octets - Octets (Information Object Address)	2 (two)	Number of octets (bytes) of the **Information Object Address** field of packet FT 1.2. Stored as "IEC.Octets.INFOOBJ". Possible values are **1**: One octet (object addresses between 0 and 255), **2**: Two octets (object addresses between 0 and 65535), or **3**: Three octets (object addresses between 0 and 16777215)
Octets - Octets (Common Address of ASDU)	-1 (minus one)	Number of octets (bytes) of the **Common Address of ASDU** field of packet FT 1.2. Stored as "IEC.Octets.CommonAddress". Possible

PROPERTY	DEFAULT VALUE	COMMENTS
		values are **-1**: Uses the same configuration of "IEC.Octets.SlaveAddress", **1**: One octet (addresses between 0 and 255), or **2**: Two octets (addresses between 0 and 65535)
Octets - Maximum Packet Size (default = 261 bytes)	261	Configures the maximum size of each packet, in bytes. The default value of this property is 261. Stored as "IEC.MaxPacketSize"
Master - Application Layer Timeout (sec)	4 (four)	Defines the time-out of application Commands, in seconds. This is the time this Driver in **Master** mode waits by a confirmation of finishing several Commands, such as General Interrogation, Select or Execute, or Clock Syncing. Stored as "IEC.AppTimeout"
Master - Wait EI (End Of Initialization)	0 (zero)	Indicates whether this Driver in **Master** mode must wait for receiving an ASDU 70 (**M_EI_NA_1**, End of initialization) to consider a Slave as connected. Stored as "IEC.WaitEI". Possible values are **0**: Does not wait or **1 or any other value different from 0**: Waits (if the ASDU is not received within the time configured in application's time-out, the Slave is disconnected)
Master - Don't wait ACTTERM for Execute commands	0 (zero)	Defines whether this Driver in **Master** mode waits for a signal of terminating **Execute Commands** (*ACTivation TERMination*). Stored as "IEC.CommandExecute.DontWaitACTTERM". Possible values are **0**: Waits or **1 or any other value different from 0**: Does not wait
*** Master - Map Overflow bit to Uncertain Quality**	0 (zero)	Defines whether receiving bit **OV** (*Overflow*) of this Driver in **Master** mode is mapped to uncertain quality. Stored as "IEC.MapOvAsUncertain". Possible values are **0**: Does not map bit **OV** to uncertain or **1**: Maps bit **OV** to Uncertain, Sub-Status Eu_Exceeded
Master - Discard ASDU 135 for Link Activity	0 (zero)	Discards ASDU 135 (private) to consider a link as active. Stored as "IEC.DiscardAsdu135ForLinkActivity". Possible values are **0**: Does not discard or **1**: Discards
Master - General Interrogation - Interrogation Auto-run	0 (zero)	Enables the Automatic **General Interrogation** Command in this Driver in **Master** mode. Stored as "IEC.Interrogate.AutoRun". Possible

PROPERTY	DEFAULT VALUE	COMMENTS
		values are **0**: Disables or **1 or any other value different from 0**: Enables
Master - General Interrogation - Interrogation Repeat	0 (zero)	Enables repeating the **General Interrogation** Command in this Driver in **Master** mode. Stored as "IEC.Interrogate.Repeat". Possible values are **0**: Disables and performs the General Interrogation Command only once when connecting to a Slave or **1 or any other value different from 0**: Enables repeating the General Interrogation Command
Master - General Interrogation - Interrogation Repeat Period (sec)	60	Defines a time interval, in seconds, between each repetition of a **General Interrogation** Command in this Driver in **Master** mode. Stored as "IEC.Interrogate.PollTime"
Master - General Interrogation - Interrogation Wait ACTTERM	1 (one)	Determines whether this Driver in **Master** mode waits or not for receiving a notification of finishing a **General Interrogation** Command. Stored as "IEC.Interrogate.WaitACTTERM". Possible values are **0**: Does not wait for a notification of finishing (*ACTivation TERMination*) or **1 or any other value different from 0**: Waits for receiving a notification of finishing a General Interrogation Command
Master - General Interrogation - Interrogation Process At End	0 (zero)	Defines the **moment in which data from a General Interrogation Command is processed** in this Driver in **Master** mode. Stored as "IEC.Interrogate.ProcessAtEnd". Possible values are **0**: Processes during the General Interrogation Command or **1 or any other value different from 0**: Processes only at the end of a General Interrogation Command
Master - General Interrogation - Disconnect if General Interrogation returns empty results	0 (zero)	Determines whether this Driver in **Master** mode accepts or not an empty response from a General Interrogation Command. Stored as "IEC.Interrogate.DisconnectIfEmpty". Possible values are **0**: Accepts an empty response from a General Interrogation Command or **1 or any other value different from 0**: If the response from a General Interrogation Command is empty forces a disconnection, allowing this Driver to

PROPERTY	DEFAULT VALUE	COMMENTS
		connect to another device on a backup connection
Master - General Interrogation - Use Broadcast	0 (zero)	Defines whether this Driver in **Master** mode, when sending a General Interrogation Command, uses the broadcast address. Stored as "IEC.UseBroadcast". Possible values are **0**: Does not use the broadcast address or **1 or any other value different from 0**: Uses the broadcast address
Master - Test Command - Test Command Enabled	0 (zero)	Enables using the automatic sending of a **Test Command** in this Driver in **Master** mode. Stored as "IEC.TestCommand.Enabled". Possible values are **0**: Disables or **1 or any other value different from 0**: Enables
Master - Test Command - Test Command Idle Time (sec)	15	Defines a period of time, in seconds, that this Driver in **Master** mode must remain without sending application level Commands to activate the automatic sending of a **Test Command**. Stored as "IEC.TestCommand.IdleTimeSec"
Master - Clock Sync - Clock Sync Auto-run	0 (zero)	Enables sending a **Clock Syncing Command** in this Driver in **Master** mode. Stored as "IEC.ClockSync.AutoRun". Possible values are **0**: Disables or **1 or any other value different from 0**: Enables
Master - Clock Sync - Clock Sync Repeat	0 (zero)	Enables the repetition of sending a **Clock Syncing Command** in this Driver in **Master** mode. Stored as "IEC.ClockSync.Repeat". Possible values are **0**: Disables or **1 or any other value different from 0**: Enables
Master - Clock Sync - Clock Sync Repeat Period (sec)	60	Defines a interval, in seconds, to repeat sending a **Clock Syncing Command** in this Driver in **Master** mode. Stored as "IEC.ClockSync.PollTime"
Slave - Send EI (End Of Initialization) When Connected	0 (zero)	Determines whether this Driver in **Slave** mode sends or not an ASDU **M_EI_NA_1** (70) indicating that a connection was established with a Master. Stored as "IEC.DontSendEI". Possible values are **0**: Sends the ASDU **M_EI_NA_1** whenever a connection is established or **1 or any other value different from 0**: Does not send the ASDU **M_EI_NA_1**

PROPERTY	DEFAULT VALUE	COMMENTS
Slave - Analog Values Send Interval (ms)	0 (zero)	Interval between **cyclic sending analog data**, in milliseconds, in this Driver in **Slave** mode. Stored as "IEC.AnalogSendIntervalMs"
Slave - Analog Deadband (% of previous value)	0 (zero)	Dead band for **sending cyclic analog data** in this Driver in **Slave** mode, as a percentage of the variation of the previous value. Stored as "IEC.AnalogDeadband"
Slave - Treat First Write As	0 (zero)	Determines whether the first writing performed on this Driver's Tags in **Slave** mode is considered a new event or the Tag's initial value. This property works until the first General Interrogation Command is performed. After the first General Interrogation Command, all writings to Tags in **Slave** mode are returned to the Master as new events. Stored as "IEC.FirstWriteIsCurrentValue". Possible values are **0 (New Event)**: Handles the first writing of each Tag as an event or **1 or any other value different from 0 (Current Value)**: Handles the first writing of each Tag as the initial value, that is, sends the value by General Interrogation Command but does not send a spontaneous notification
Slave - General Interrogation - Interrogation Enable Range	0 (zero)	Enables returning a fixed list of objects in the **General Interrogation** Command in this Driver in **Slave** mode. Stored as "IEC.EnableInterrogationRange". Possible values are **0**: Returns all objects from the cache memory of this Driver or **1 or any other value different from 0**: Returns only the objects declared on the General Interrogation list
Slave - General Interrogation - List of data objects returned by General Interrogation	Empty **String**	Defines a **list of objects returned in a General Interrogation Command** in this Driver in **Slave** mode. Stored as "IEC.InterrogationRange"
Slave - General Interrogation - Remove timestamp from data sent on General Interrogation	0 (zero)	Determines whether data from a **General Interrogation** Command of this Driver in **Slave** mode is sent without a timestamp. Stored as "IEC.InterrogationRemoveTimestamp". Possible values are **0**: Sends data using the original ASDU, that is, may or may not have a timestamp, or **1 or any other value different from 0**: Removes the timestamp of data sent in

PROPERTY	DEFAULT VALUE	COMMENTS
		a General Interrogation Command, converting it to an ASDU without timestamp
Slave - Command Handling - Handle Mode for Clock Sync commands	0 (zero)	Defines how to handle **Clock Syncing Commands** received by this Driver in **Slave** mode. Stored as "IEC.CommandHandleMode.ClockSync". Possible values are **0**: Default (disabled), **1**: Disabled, **2**: **Passes to the application**, or **3**: Handles automatically
Slave - Command Handling - Handle Mode for Select/Execute commands	0 (zero)	Defines how to handle **Select or Execute Commands** received by this Driver in **Slave** mode. Stored as "IEC.CommandHandleMode.SelectExecute". Possible values are **0**: Default (passes to the application), **1**: Disabled, **2**: **Passes to the application**, or **3**: Handles automatically
Slave - Command Handling - Handle Mode for Test commands	0 (zero)	Defines how to handle **Test Commands** received by this Driver in **Slave** mode. Stored as "IEC.CommandHandleMode.Test". Possible values are **0**: Default (handle automatically), **1**: Disabled, **2**: Passes to the application, or **3**: Handles automatically
Slave - Cyclic - Cyclic Data Transmission (1): Send Period (ms)	10000	Interval, in milliseconds, for **sending cyclic data** declared in the "IEC.CyclicRange" property of this Driver in **Slave** mode. Stored as "IEC.CyclidPeriodMs". **NOTE**: Notice that the stored name of this variable contains a typo (*Cyclid* instead of *Cyclic*), kept for compatibility reasons with older versions of this Driver
Slave - Cyclic - Cyclic Data Transmission (1): Object Range	Empty **String**	Defines data to be **sent cyclically** by this Driver in **Slave** mode. Stored as "IEC.CyclicRange"
Slave - Cyclic - Cyclic Data Transmission (2): Send Period (ms)	5000	Interval, in milliseconds, for **sending cyclic data** declared in the "IEC.CyclicRange2" property of this Driver in **Slave** mode. Stored as "IEC.CyclicPeriodMs2"
Slave - Cyclic - Cyclic Data Transmission (2): Object Range	Empty **String**	Defines a second set of data to be **send cyclically** by this Driver in **Slave** mode. Stored as "IEC.CyclicRange2"
Slave - Offline Buffers - Offline Class1 buffering	0 (zero)	Enables keeping Class 1 (one, high priority) events **in memory** in this Driver in **Slave** mode when the Master is disconnected. Stored as

PROPERTY	DEFAULT VALUE	COMMENTS
		"IEC.OfflineClass1.EnableBuffering". Possible values are **0**: Disables or **1 or any other value different from 0**: Enables
Slave - Offline Buffers - Offline Class1 buffer size	100	Number of Class 1 (one, high priority) events to **keep in memory** by this Driver in **Slave** mode when the Master is disconnected. Stored as "IEC.OfflineClass1.BufferSize"
Slave - Offline Buffers - Offline Class2 buffering	0 (zero)	Enables keeping Class 2 (two, low priority) events **in memory** by this Driver in **Slave** mode when the Master is disconnected. Stored as "IEC.OfflineClass2.EnableBuffering". Possible values are **0**: Disables or **1 or any other value different from 0**: Enables
Slave - Offline Buffers - Offline Class2 buffer size	100	Number of Class 2 (two, low priority) events to **keep in memory** by this Driver in **Slave** mode when the Master is disconnected. Stored as "IEC.OfflineClass2.BufferSize"
104 - T0 (sec)	30	Configures the **T0** parameter, in seconds, of protocol **IEC 870-5-104**. Possible values range between 1 (one) and 255 seconds. Stored as "IEC.IEC104.T0"
104 - T1 (sec)	15	Configures the **T1** parameter, in seconds, of protocol **IEC 870-5-104**. Possible values range between 1 (one) and 255 seconds. Stored as "IEC.IEC104.T1"
104 - T2 (sec)	10	Configures the **T2** parameter, in seconds, of protocol **IEC 870-5-104**. Possible values range between 1 (one) and 255 seconds. Stored as "IEC.IEC104.T2"
104 - T3 (sec)	20	Configures the **T3** parameter, in seconds, of protocol **IEC 870-5-104**. Possible values range between 1 (one) and 255 seconds. Stored as "IEC.IEC104.T3"
104 - K	12	Configures the **K** parameter, or number of packets, of protocol **IEC 870-5-104**. Possible values range between 1 (one) and 32767 packets. Stored as "IEC.IEC104.K"
104 - W	8 (eight)	Configures the **W** parameter, or number of packets, of protocol **IEC 870-5-104**. Possible values range

PROPERTY	DEFAULT VALUE	COMMENTS
		between 1 (one) and 32767 packets. Stored as "IEC.IEC104.W"
Timestamp - Timestamp SU Flag (Summertime adjustment)	0 (zero)	Determines how to handle **summer time** on timestamps. Stored as "IEC.TimestampSuFlag". Possible values are **0**: Ignores the **SU** flag on timestamps or **1 or any other value different from 0**: Uses the **SU** flag on timestamps
Timestamp - Enable Station TimeZone	0 (zero)	Enables using the "Station TimeZone" parameter, which allows indicating a time zone for all Stations different from the computer's time zone. Stored as "IEC.TimeZoneEnable"
Timestamp - Station TimeZone (in minutes)	0 (zero)	Defines a difference, in minutes, between the time zone of Stations and Greenwich time zone. For example, use the value -180 for Brasilia time zone (GMT-3). Stored as "IEC.TimeZoneMinutes"
Timestamp - SP Timestamp Format	2 (two)	Defines the timestamp format for **Simple Point** data sent by this Driver in **Slave** mode to a Master outside the General Interrogation Command, only for Tags configured by **String**. Stored as "IEC.TimestampFormat.SP". Possible values are **0**: Without timestamp (uses ASDU **M_SP_NA_1**, 1), **1**: With 24-bit timestamp (uses ASDU **M_SP_TA_1**, 2), or **2**: With 56-bit timestamp (uses ASDU **M_SP_TB_1**, 30)
Timestamp - DP Timestamp Format	2 (two)	Defines the timestamp format for **Double Point** data sent by this Driver in **Slave** mode to a Master outside the General Interrogation Command, only for Tags configured by **String**. Stored as "IEC.TimestampFormat.DP". Possible values are **0**: Without timestamp (uses ASDU **M_DP_NA_1**, 3), **1**: With 24-bit timestamp (uses ASDU **M_DP_TA_1**, 4), or **2**: With 56-bit timestamp (uses ASDU **M_DP_TB_1**, 31)
Timestamp - BO Timestamp Format	2 (two)	Defines the timestamp format for **Bitstring** data sent by this Driver in **Slave** mode to a Master outside the General Interrogation Command, only for Tags configured by **String**. Stored as "IEC.TimestampFormat.BO". Possible values are **0**: Without timestamp (uses ASDU **M_BO_NA_1**, 7), **1**: With 24-bit timestamp (uses ASDU **M_BO_TA_1**, 8), or **2**: With 56-bit timestamp (uses ASDU **M_BO_TB_1**, 33)

PROPERTY	DEFAULT VALUE	COMMENTS
Timestamp - ST Timestamp Format	2 (two)	Defines the timestamp format for **Step Position** data sent by this Driver in **Slave** mode to a Master outside the General Interrogation Command, only for Tags configured by **String**. Stored as "IEC.TimestampFormat.ST". Possible values are **0**: Without timestamp (uses ASDU **M_ST_NA_1**, 5), **1**: With 24-bit timestamp (uses ASDU **M_ST_TA_1**, 6), or **2**: With 56-bit timestamp (uses ASDU **M_ST_TB_1**, 32)
Timestamp - MEn Timestamp Format	2 (two)	Defines the timestamp format for **Normalized Analog Measurements** data sent by this Driver in **Slave** mode to a Master outside the General Interrogation Command, only for Tags configured by **String**. Stored as "IEC.TimestampFormat.MEn". Possible values are **0**: Without timestamp (uses ASDU **M_ME_NA_1**, 9), **1**: With 24-bit timestamp (uses ASDU **M_ME_TA_1**, 10), or **2**: With 56-bit timestamp (uses ASDU **M_ME_TD_1**, 34)
Timestamp - MEs Timestamp Format	2 (two)	Defines the timestamp format for **Scaled Analog Measurements** data sent by this Driver in **Slave** mode to a Master outside the General Interrogation Command, only for Tags configured by **String**. Stored as "IEC.TimestampFormat.MEs". Possible values are **0**: Without timestamp (uses ASDU **M_ME_NB_1**, 11), **1**: With 24-bit timestamp (uses ASDU **M_ME_TB_1**, 12), or **2**: With 56-bit timestamp (uses ASDU **M_ME_TE_1**, 35)
Timestamp - MEf Timestamp Format	2 (two)	Defines the timestamp format for **Floating-Point Analog Measurements** data sent by this Driver in **Slave** mode to a Master outside the General Interrogation Command, only for Tags configured by **String**. Stored as "IEC.TimestampFormat.MEf". Possible values are **0**: Without timestamp (uses ASDU **M_ME_NC_1**, 13), **1**: With 24-bit timestamp (uses ASDU **M_ME_TC_1**, 14), or **2**: With 56-bit timestamp (uses ASDU **M_ME_TF_1**, 36)
Send Priority - SP Send Priority	1 (one)	Defines the way and the priority of sending **Simple Digital Point** data from this Driver in **Slave** mode to a Master, only for Tags configured by **String**. Stored as "IEC.SendFlags.SP". Possible values are **1**: Sends with high

PROPERTY	DEFAULT VALUE	COMMENTS
		priority, **2**: Sends with low priority, or **3**: Sends only in a General Interrogation Command
Send Priority - DP Send Priority	1 (one)	Defines the way and the priority of sending **Double Digital Point** data from this Driver in **Slave** mode to a Master, only for Tags configured by **String**. Stored as "IEC.SendFlags.DP". Possible values are **1**: Sends with high priority, **2**: Sends with low priority, or **3**: Sends only in a General Interrogation Command
Send Priority - BO Send Priority	2 (two)	Defines the way and the priority of sending **Bitstring** data from this Driver in **Slave** mode to a Master, only for Tags configured by **String**. Stored as "IEC.SendFlags.BO". Possible values are **1**: Sends with high priority, **2**: Sends with low priority, or **3**: Sends only in a General Interrogation Command
Send Priority - ST Send Priority	2 (two)	Defines the way and the priority of sending **Step Position** data from this Driver in **Slave** mode to a Master, only for Tags configured by **String**. Stored as "IEC.SendFlags.ST". Possible values are **1**: Sends with high priority, **2**: Sends with low priority, or **3**: Sends only in a General Interrogation Command
Send Priority - MEn Send Priority	2 (two)	Defines the way and the priority of sending **Normalized Analog Measurements** from this Driver in **Slave** mode to a Master, only for Tags configured by **String**. Stored as "IEC.SendFlags.MEn". Possible values are **1**: Sends with high priority, **2**: Sends with low priority, or **3**: Sends only in a General Interrogation Command
Send Priority - MEs Send Priority	2 (two)	Defines the way and the priority of sending **Scaled Analog Measurements** from this Driver in **Slave** mode to a Master, only for Tags configured by **String**. Stored as "IEC.SendFlags.MEs". Possible values are **1**: Sends with high priority, **2**: Sends with low priority, or **3**: Sends only in a General Interrogation Command
Send Priority - MEf Send Priority	2 (two)	Defines the way and the priority of sending **Floating-Point Analog Measurements** from this Driver in **Slave** mode to a Master, only for Tags

PROPERTY	DEFAULT VALUE	COMMENTS
		configured by **String**. Stored as "IEC.SendFlags.MEf". Possible values are **1**: Sends with high priority, **2**: Sends with low priority, or **3**: Sends only in a General Interrogation Command
Boolean Values - Convert DCO to Boolean	0 (zero)	Defines the format of values from a **Double Digital Point Command**. Stored as "IEC.UseBoolean.DCO". Possible values are **0**: Uses as a DCO (**0**: Undetermined, **1**: OFF, **2**: ON, or **3**: Undetermined) or **1 or any other value different from 0**: Uses as a **Boolean** value (**0**: OFF or **1**: ON)
Boolean Values - Convert DIQ to Boolean	0 (zero)	Enables using **Double Digital Points** as **Boolean** values. Stored as "IEC.UseBoolean.DIQ". Possible values are **0**: Uses as a DIQ (**0**: Undetermined, **1**: OFF, **2**: ON, or **3**: Undetermined) or **1 or any other value different from 0**: Uses as a **Boolean** value (**0**: OFF or **1**: ON)
Boolean Values - Convert RCO to Boolean	0 (zero)	Defines the format of values from a **Step Regulation Command**. Stored as "IEC.UseBoolean.RCO". Possible values are **0**: Uses as an RCS (**0**: Not allowed, **1**: Decrements, **2**: Increments, or **3**: Not allowed) or **1 or any other value different from 0**: Uses as a **Boolean** value (**0**: Decrements or **1**: Increments)
Boolean Values - Convert SEP to Boolean	0 (zero)	Defines the format of values from a Simple Event of a Protection Device. Stored as "IEC.UseBoolean.SEP". Possible values are **0**: Uses as a SEP (**0**: Undetermined, **1**: OFF, **2**: ON, or **3**: Undetermined) or **1 or any other value different from 0**: Uses as a **Boolean** value (**0**: OFF or **1**: ON)

Reference of Errors in Log

The next table lists some possible errors that may appear in this Driver's log, with the respective tips for solving them.

Errors in this Driver's log

ERROR	DESCRIPTION OR SOLUTION
LinkLayer104: Wrong frame sequence (received #XXX, expected #YYY), disconnecting...	Modes **Master** and **Slave**. This message reports receiving a message out of order in protocol **IEC 870-5-104**. When this Driver detects that situation, forces a reconnection. Its cause is probably a message lost in the network. To solve it, investigate the physical connection or the network between this Driver and the device

ERROR	DESCRIPTION OR SOLUTION
ERROR: Failed to send command 'XXX', application layer is locked	Only in **Master** mode. This message occurs when this Driver tries to send a Command at application level while another Command is already executing. For example, trying to send a Select or Execute Command while a General Interrogation Command is executing. This Driver only generates that message and quits sending the Command after waiting for the time specified by the application's time-out. The cause is that the Application Layer of this Driver is locked for a long time. To solve it, increase the application's time-out
GeneralInterrogation: received actterm with no GI active, retrying last GI	Only in **Master** mode. This message occurs when this Driver receives a confirmation of finishing a General Interrogation Command and there is no active general interrogation process. Causes can be a General Interrogation Command started previously by this Driver which was interrupted due to a problem with device's connection or this Driver is configured to not wait for a notification of finishing a General Interrogation Command and the Slave is sending that finishing notification. Possible solution are increase application's time-out of this Driver, investigate what caused the problem with device's connection, or configure this Driver to wait for the finishing of the General Interrogation Command, and also possibly consider increasing application's time-out of this Driver

Documentation of I/O Interfaces

This section contains the documentation of I/O Interfaces referring to the **IEC870-104 Driver**.

Configuration of a Driver

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

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

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

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

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

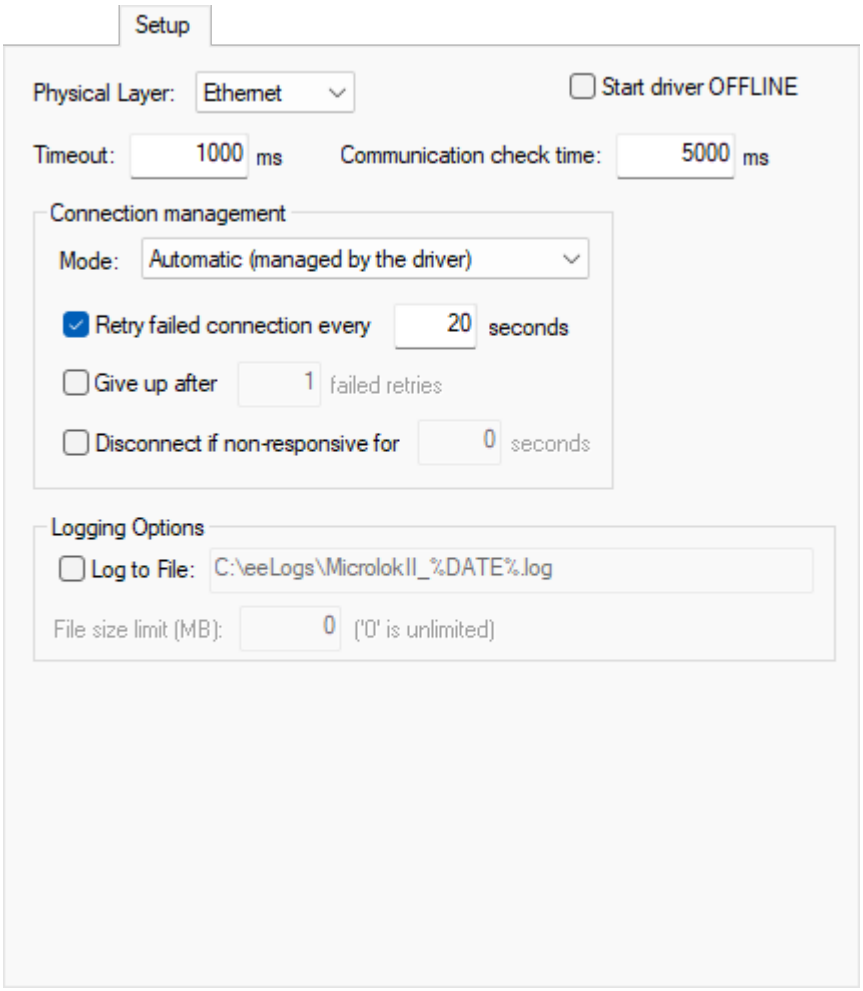
Configuration Dialog Box

The dialog box of I/O Interfaces allows configuring the I/O connection used by a Driver. This dialog box contains the **Setup**, **Serial**, **Ethernet**, **Modem**, and **RAS** tabs, described on the next topics. If a Driver does not implement a specific I/O connection, its corresponding tab is not available for configuration. Some Drivers may contain additional tabs, specific for that Driver, on the configuration dialog box.

Setup Tab

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

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



Setup tab

General options on the Setup tab

OPTION	DESCRIPTION
Physical Layer	Select the physical layer on a list. Available options are **Serial**, **Ethernet**, **Modem**, and **RAS**. The selected interface must be configured on its specific tab
Timeout	Configure a time-out, in milliseconds, for the physical layer. This is the amount of time an I/O interface waits to receive any byte from the reception's buffer

OPTION	DESCRIPTION
Communication check time	Set the time, in milliseconds, to define the interval at which communication is considered to be in an inactive state. As long as an I/O Driver receives valid data, its communication state is considered active. However, if during operation an I/O Driver does not receive valid data inside this period of time, the state is considered inactive. The communication state is shown in the **IO.CommunicationStatus** Tag
Start driver OFFLINE	Select this option so that a Driver starts in **Offline** mode or stopped. This means that the I/O interface is not created until this Driver is configured to **Online** mode by using a Tag in an application. This mode enables a dynamic configuration of an I/O interface at run time

Options on the Connection management group

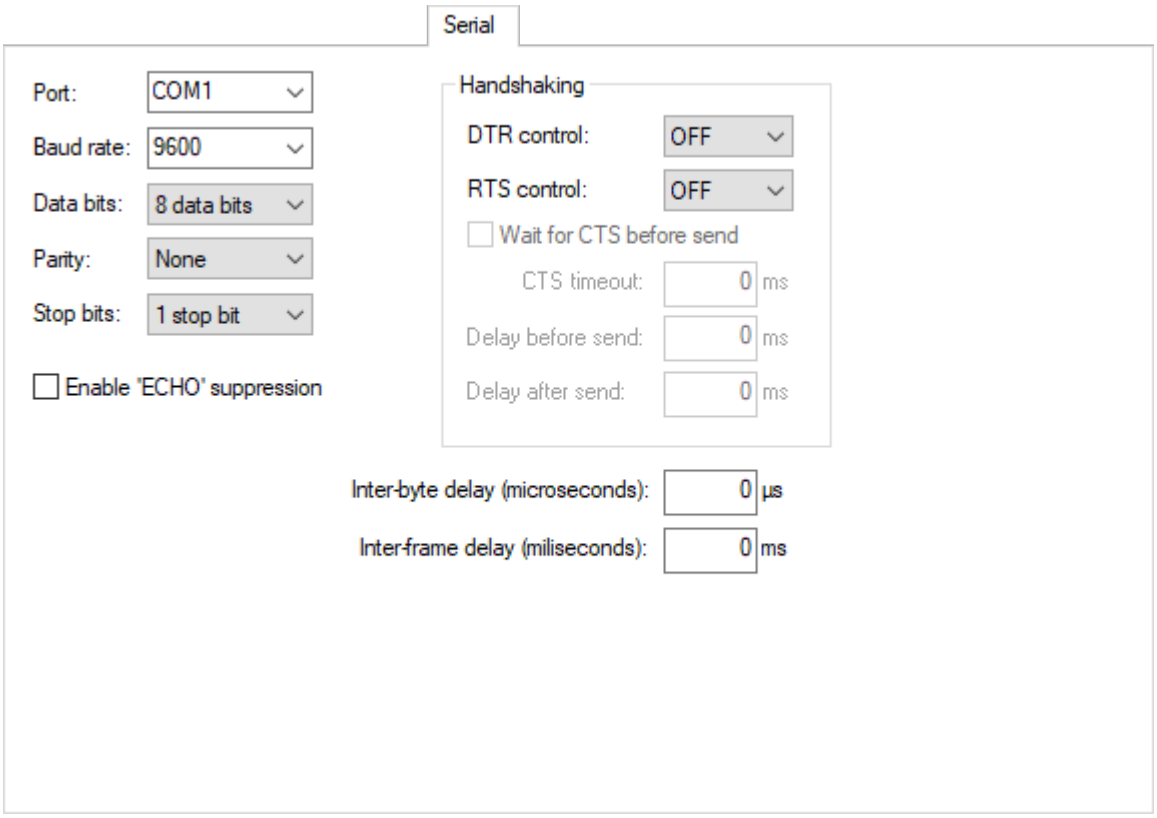
OPTION	DESCRIPTION
Mode	Selects a management mode of a connection. Selecting the **Automatic** option allows a Driver to manage the connection automatically, as specified in the next options. Selecting the **Manual** option allows an application to fully manage a connection
Retry failed connection every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

Options on the Logging Options group

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

Serial Tab

Use this tab to configure parameters for a **Serial** Interface.



Serial tab

General options on the Serial tab

OPTION	DESCRIPTION
Port	Select a serial port on the list, from **COM1** to **COM4**, or type the name of a serial port in the format **COM*n***, such as "COM15". When typing the name of a serial port manually, the dialog box only accepts names of serial ports starting with the expression "COM"
Baud rate	Select a baud rate on the list (**1200**, **2400**, **4800**, **9600**, **19200**, **38400**, **57600**, or **115200**) or type a baud rate, such as 600
Data bits	Select 7 (seven) or 8 (eight) data bits on the list
Parity	Select a parity on the list. The available options are **None**, **Even**, **Odd**, **Mark**, or **List**
Stop bits	Select the number of stop bits on the list. The available options are **1**, **1.5**, or **2** stop bits
Enable 'ECHO' supression	Enable this option to remove the echo received after the I/O Interface sends data via serial port. If this echo is not equal to the bytes just sent, then the I/O Interface aborts communication
Inter-byte delay (microseconds)	Defines a delay between each byte transmitted by the I/O Interface, in millionths of a second, that is, 1000000 is equal to a second. This option must be used with small delays of less than a millisecond
Inter-frame delay (milliseconds)	Defines a delay between packets sent or received by the I/O Interface, in thousandths of a second, that is, 1000 is equal to a second. This delay is applied if the I/O Interface

OPTION	DESCRIPTION
	sends two consecutive packets, or between a received packet and the next sending

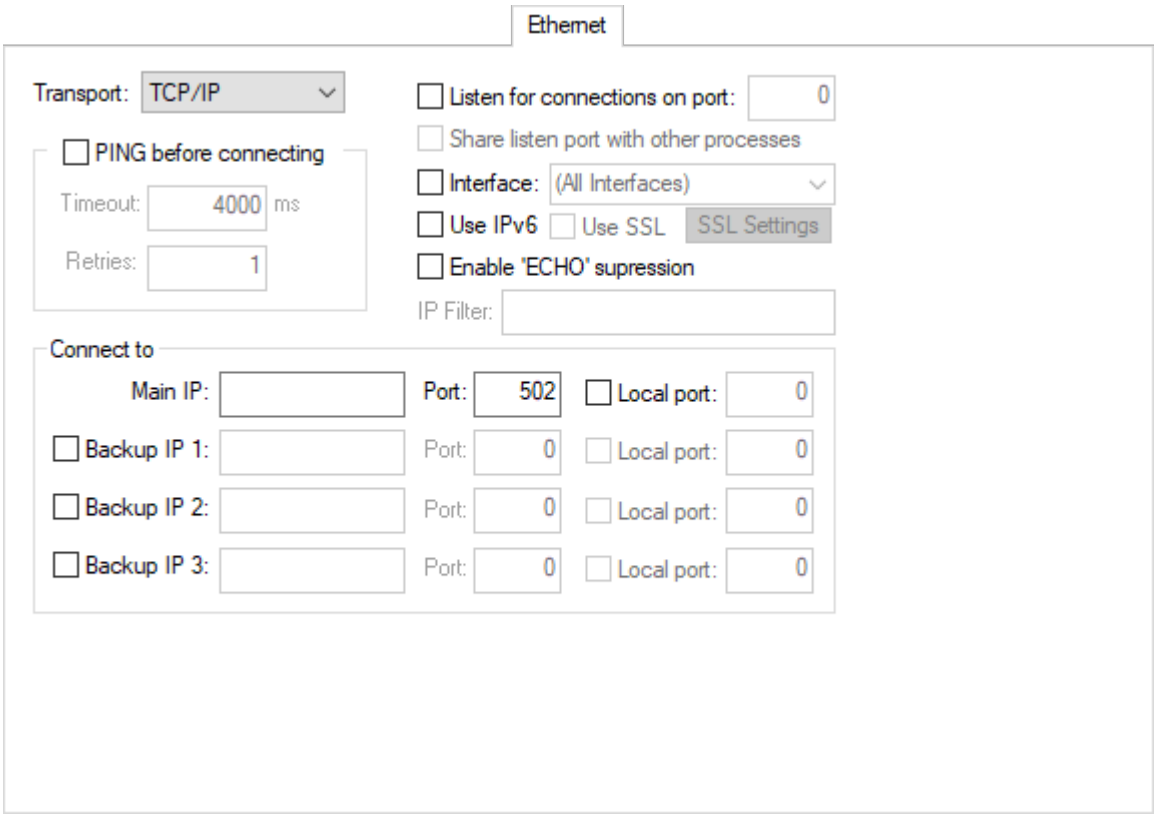
The **Handshaking** group configures the usage of **RTS**, **CTS**, and **DTR** signals in the handshaking process, that is, it controls when data can be sent or received via serial line. Most of the time, configuring the **DTR control** option to **ON** and the **RTS control** option to **Toggle** works with **RS232**-type serial lines as well as with **RS485**-type serial lines.

Available options on the Handshaking group

OPTION	DESCRIPTION
DTR control	Select the value **ON** to keep the **DTR** signal always on while the serial port is open. Select the value **OFF** to turn the **DTR** signal off while the serial port is open. Some devices require the **DTR** signal always on to allow communication
RTS control	Select the value **ON** to keep the **RTS** signal always on while the serial port is open. Select the value **OFF** to turn the **RTS** signal off while the serial port is open. Select the value **Toggle** to turn the **RTS** signal on while sending bytes via serial port and turn it off when not sending bytes, therefore enabling the reception
Wait for CTS before send	Available only when the **RTS control** option is configured with the value **Toggle**. Use this option to force a Driver to check the **CTS** signal before sending bytes via serial port, after turning the **RTS** signal on. In this mode, the **CTS** signal is handled as a permission flag for sending
CTS timeout	Determines a maximum time, in milliseconds, that a Driver waits for the **CTS** signal after turning the **RTS** signal on. If the **CTS** signal is not turned on within this time-out, that Driver then fails the current communication and returns an error
Delay before send	Some serial port devices have a delay when enabling a data sending circuit after the **RTS** signal is turned on. Configure this option to wait a certain number of milliseconds after turning the **RTS** signal on and before sending the first byte. **IMPORTANT**: This delay must be used carefully, because it uses 100% of CPU resources while waiting. System's general performance degrades as this value increases
Delay after send	This is the same effect of the **Delay before send** option, but in this case the delay is performed after sending the last byte, before turning the **RTS** signal off

Ethernet Tab

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



Ethernet tab

Available options on the Ethernet tab

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

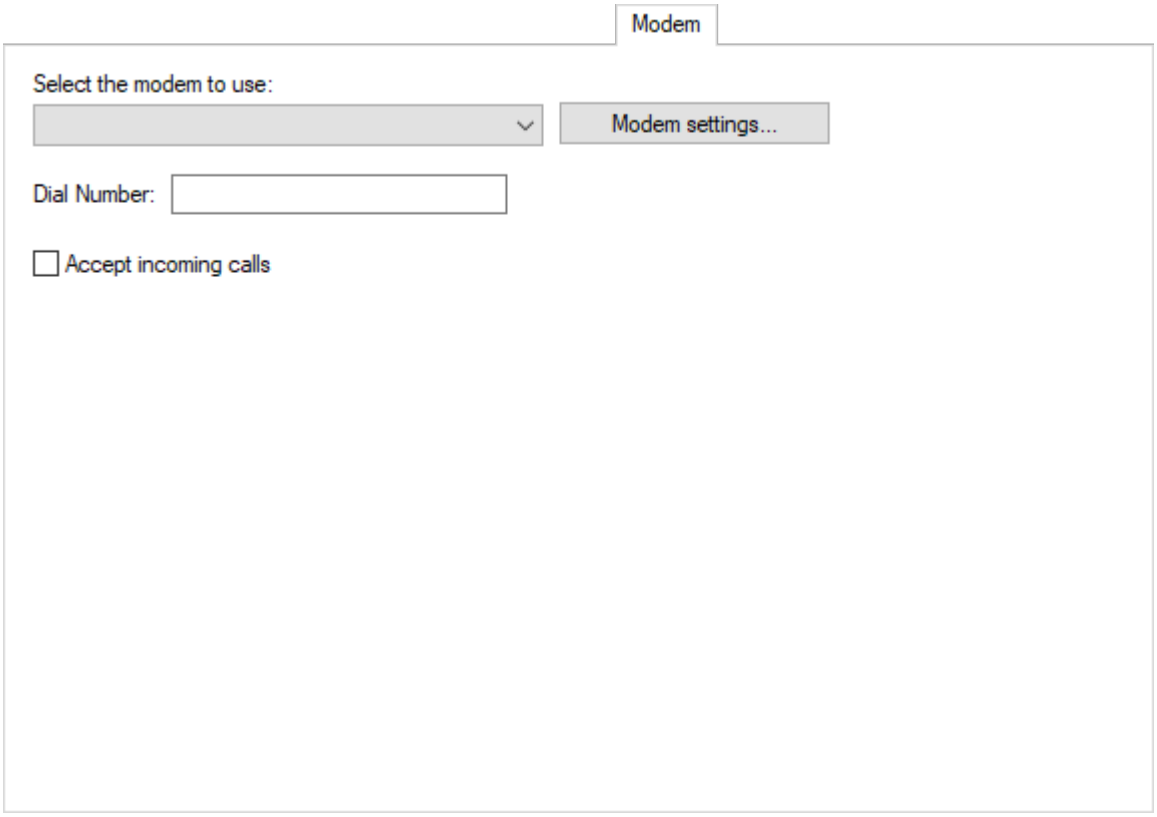
OPTION	DESCRIPTION
	of determining a successful connection before trying to open a socket with a device. The time-out of a connection with a socket can be very high. The available options are: • **Timeout**: Specify the number of milliseconds to wait for a reply from a **ping** command. Users must use a **ping** command to check the normal reply time, configuring this option for a value above that average. Usually this value can be configured between 1000 and 4000 milliseconds, that is, between 1 (one) and 4 (four) seconds • **Retries**: Number of retries of a **ping** command, not counting the first attempt. If all attempts fail, then the socket connection is aborted

Available options on the Connect to group

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

Modem Tab

Use this tab to configure parameters of a **Modem** Interface. Some options on the **Serial** tab affect the configuration of a modem, therefore users must also configure the **Serial** Interface.



Modem tab

The **Modem** Interface uses the TAPI modems installed on the computer.

Available options on the Modem tab

OPTION	DESCRIPTION
Select the modem to use	Select a modem on the list of available modems on the computer. If the value **Default modem** is selected, then the first available modem is used. Selecting this option is recommended specially when an application is used on another computer
Modem settings	Click to open the configuration window of the selected modem
Dial Number	Type a default number for dialing. This value can be changed at run time. Users can use the **w** character to represent a pause or a waiting time for a dial tone. For example, "0w33313456" dials the number 0 (zero), waits, and then dials the number "33313456"
Accept incoming calls	Enable this option so that a Driver answers the phone when receiving an external call. To use this option, users must configure the **Connection management** option on the **Setup** tab to the value **Manual**

RAS Tab

Use this tab configure parameters of a **RAS** Interface. Users must also configure the **Ethernet** tab.

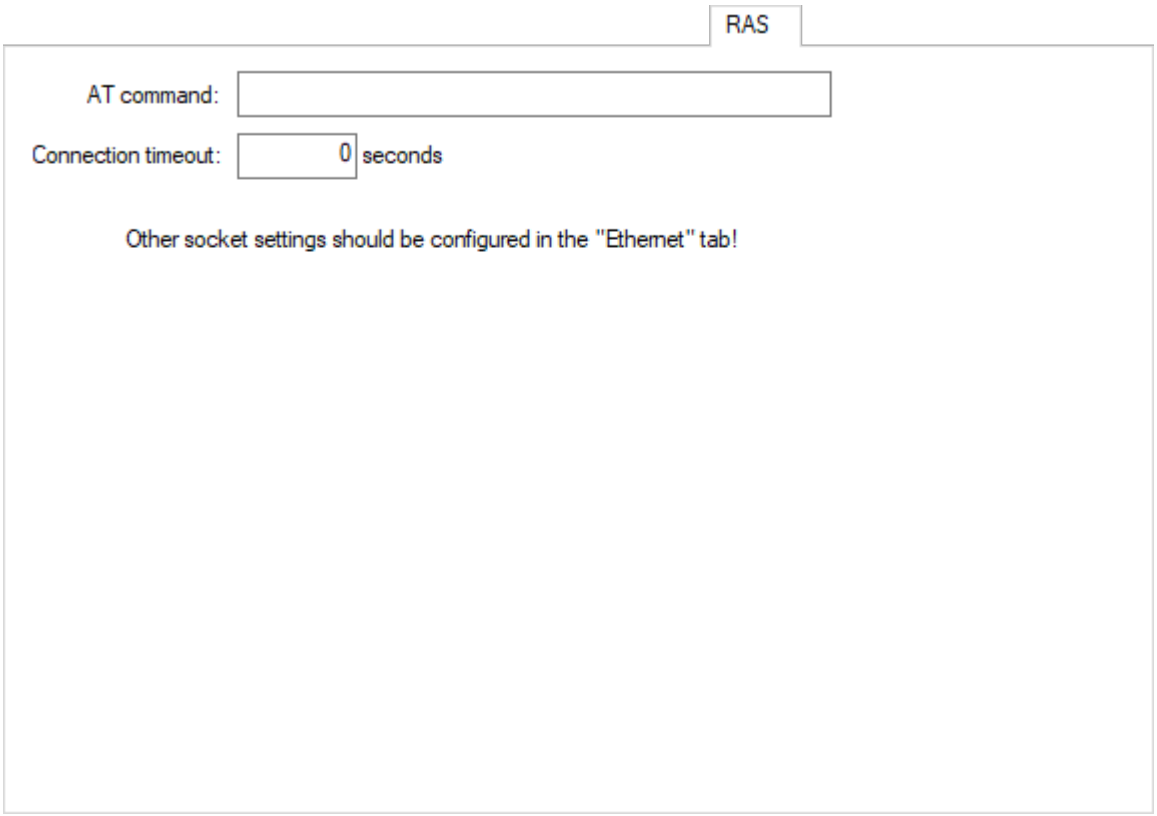
A **RAS** Interface opens a socket connection with a RAS device. A RAS device is a server of modems available through TCP/IP, waiting for socket connections on an IP port. For each connection accepted on this port, users have access to one modem.

When connecting to a RAS device, first the I/O Interface**IOKit** connects to the socket on the IP address and port configured on the **Ethernet** tab. After opening the socket, the following initialization or connection steps are performed:

1. Clears the socket, that is, removes any **TELNET** greeting message received from a RAS device.
2. Sends an **AT** dial message, in **ASCII** format, in the socket.
3. Waits for a **CONNECT** reply.
4. If the time-out expires, the connection is aborted.
5. If the **CONNECT** reply is received within the time-out, the socket is available for communication with a device, that is, the connection was established.

If step 5 (five) is successful, then the socket behaves as a normal socket, with the RAS device working as a router between a Driver and the device. Bytes sent by a Driver are received by the RAS device and sent to the destination device using a modem. Bytes received by the modem's RAS device are sent back to a Driver using the same socket.

After establishing a connection, the **RAS** interface monitors data received by a Driver. If a "NO CARRIER" **String** is found, the socket is closed. If the RAS device does not send a **NO CARRIER** signal, the **RAS** Interface cannot detect when the modem connection between the RAS device and the final I/O device fails. To recover from this failure, users are strongly advised to enable the **Disconnect if non-responsive** option on the **Setup** tab.



RAS tab

Available options on RAS tab

OPTION	DESCRIPTION
AT command	A **String** with the full **AT** command used to dial to a destination device. For example, "ATDT33313456" dials by tone to number "33313456"
Connection timeout	Number of seconds to wait for a modem's **CONNECT** reply, after sending an **AT** command

General Configurations

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

I/O Tags

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

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

IO.CommunicationStatus

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

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

IO.IOKitEvent

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

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

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

- **Element 3**: Message of an event, a **String** specific for each event

NOTE

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

IO.PhysicalLayerStatus

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

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

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

IO.SetConfigurationParameters

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

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

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

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

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

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

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

A variation of the previous example uses a bidimensional array.

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

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

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

IO.WorkOnline

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

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

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

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

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

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

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

IMPORTANT

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

Properties

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

IO.ConnectionMode

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

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

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

IO.RecoverEnable

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

IO.RecoverPeriodSec

Delay time between two connection attempts, in seconds.

NOTE

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

IO.StartOffline

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

NOTE

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

IO.TimeoutMs

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

IO.Type

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

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

Statistical Configuration

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

I/O Tags

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

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

IO.Stats.Partial.BytesRecv

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

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

IO.Stats.Partial.BytesSent

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

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

IO.Stats.Partial.TimeConnectedSeconds

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

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

IO.Stats.Partial.TimeDisconnectedSeconds

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

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

IO.Stats.Total.BytesRecv

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

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

IO.Stats.Total.BytesSent

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

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

IO.Stats.Total.ConnectionCount

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

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

IO.Stats.Total.TimeConnectedSeconds

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

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

IO.Stats.Total.TimeDisconnectedSeconds

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

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

Properties

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

Ethernet Interface Configuration

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

I/O Tags

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

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

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

IO.Ethernet.IPSelect

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

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

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

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

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

IO.Ethernet.IPSwitch

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

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

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

IO.Ethernet.SocketState

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

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

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

Properties

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

NOTE

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

IO.Ethernet.AcceptConnection

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

IO.Ethernet.BackupEnable[2,3]

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

IO.Ethernet.BackupIP[2,3]

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

IO.Ethernet.BackupLocalPort[2,3]

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

IO.Ethernet.BackupLocalPortEnable[2,3]

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

IO.Ethernet.BackupPort[2,3]

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

IO.Ethernet.IPFilter

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

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

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

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

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

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

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

IO.Ethernet.ListenIP

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

IO.Ethernet.ListenPort

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

IO.Ethernet.MainIP

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

IO.Ethernet.MainLocalPort

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

IO.Ethernet.MainLocalPortEnable

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

IO.Ethernet.MainPort

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

IO.Ethernet.PingEnable

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

IO.Ethernet.PingTimeoutMs

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

IO.Ethernet.PingTries

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

IO.Ethernet.ShareListenPort

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

IO.Ethernet.SupressEcho

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

IO.Ethernet.Transport

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

IO.Ethernet.UseIPv6

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

Modem Interface Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of a **Modem** (TAPI) Interface.

I/O Tags

Tags of a Modem Interface (N2/B2 = 3)

The Tags described next allow controlling and diagnosing a **Modem** (TAPI) Interface at run time.

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

IO.TAPI.ConnectionBaudRate

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	5 (five)
String Configuration	IO.TAPI.ConnectionBaudRate

Indicates a baud rate value for the current connection. If a modem is not connected, returns the value 0 (zero).

IO.TAPI.Dial

Type of Tag	I/O Tag
Type of Access	Write-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	1 (one)
String Configuration	IO.TAPI.Dial

Write any value to this Tag to force a **Modem** Interface to start a call. This is an asynchronous command, only starting the call process. Users can monitor the **IO.TAPI.IsModemConnected** Tag to detect when a call is established.

IO.TAPI.HangUp

Type of Tag	I/O Tag
Type of Access	Write-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	4 (four)
String Configuration	IO.TAPI.HangUp

Any value written to this Tag hangs the current call up.

NOTE

Use this command only when managing the physical layer manually or when explicitly trying to force a Driver to restart the communication. If the physical layer is configured for automatic reconnection, a Driver immediately tries to reestablish the connection.

IO.TAPI.IsModemConnected

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	3 (three)
String Configuration	IO.TAPI.IsModemConnected

This Tag indicates the status of a modem connection. Possible values are **0**: The modem is not connected, but it may be performing or receiving an external call or **1**: The modem is connected and a Driver completed or received an external call successfully. While it is in this status, the physical layer can send or receive data.

IO.TAPI.IsModemConnecting

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	6 (six)
String Configuration	IO.TAPI.IsModemConnecting

This Tag indicates the status of a modem connection, with more details than the **IO.TAPI.IsModemConnected** Tag. Possible values are **0**: Modem is not connected, **1**: Modem is connecting, that is, performing or receiving an external call, **2**: Modem is connected. While in this status, the physical layer can send or receive data, or **3**: Modem is disconnecting the current call.

IO.TAPI.ModemStatus

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	2 (two)
String Configuration	IO.TAPI.ModemStatus

Returns a **String** with the current status of a modem. Possible values are the following:

- **"No status!"**: The **Modem** Interface was not open yet or was already closed
- **"Modem initialized OK!"**: The **Modem** Interface was initialized successfully
- **"Modem error at initialization!"**: A Driver could not initialize modem's line. Check that Driver's log file for more details
- **"Modem error at dial!"**: A Driver could not start or accept a call
- **"Connecting..."**: A Driver started a call successfully, and is currently processing that call
- **"Ringing..."**: Indicates that the modem is receiving an external call, but it did not accepted it yet
- **"Connected!"**: A Driver connected successfully, that is, completed or accepted an external call
- **"Disconnecting..."**: A Driver is turning the current call off
- **"Disconnected OK!"**: A Driver turned the current call off
- **"Error: no dial tone!"**: A Driver aborted a call because the available line signal was not detected
- **"Error: busy!"**: A Driver aborted a call because the line was busy
- **"Error: no answer!"**: A Driver aborted a call because no answer was received from the other modem
- **"Error: unknown!"**: Current call was aborted because of an unknown error

IO.TAPI.PhoneNumber

Type of Tag	I/O Tag
Type of Access	Reading or Writing
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	0 (zero)
String Configuration	IO.TAPI.PhoneNumber

This Tag is a **String** that reads or changes the telephone number used by the **IO.TAPI.Dial** Tag. When changing this Tag, the new value is used only on the next **Dial** command.

Properties

These properties control the configuration of a **Modem** (TAPI) Interface.

IO.TAPI.AcceptIncoming

9 Configure to False if a modem cannot accept external calls, that is, if a Driver behaves as a master, and configure to True to enable receiving calls, that is, if a Driver behaves as a slave.

IO.TAPI.ModemID

9 This is the identification number of a modem. This ID is created by Windows and used internally to identify a modem on a list of devices installed on a computer. This ID may not remain valid if a modem is reinstalled or an application is executed on another computer.

NOTE

It is advisable to configure this property as 0 (zero), indicating that a Driver must use the first available modem.

IO.TAPI.PhoneNumber

A A telephone number used by **Dial** commands, such as "0w01234566", in which the "w" character forces a modem to wait for a call sign.

RAS Interface Configuration

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

I/O Tags

Tags of a RAS Interface (N2/B2 = 5)

Currently, there are no Tags defined specifically to manage a **RAS** Interface at run time.

Properties

These properties control the configuration of a **RAS** Interface.

NOTE

A **RAS** Interface uses the **Ethernet** Interface, which therefore must be also configured.

IO.RAS.ATCommand

A An **AT** command to send through a socket to force a RAS device to perform a call using the current RAS channel, such as "ATDT6265545".

IO.RAS.CommandTimeoutSec

Time to wait for a **CONNECT** message in response to an **AT** command, in seconds.

Serial Interface Configuration

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

I/O Tags

Tags of a Serial Interface (N2/B2 = 2)

Currently, there are no Tags defined specifically to manage a **Serial** Interface at run time.

Properties

These properties control the configuration of a **Serial** Interface.

IO.Serial.Baudrate

Specifies a baud rate of a serial port, such as 9600.

IO.Serial.CTSTimeoutMs

Time to wait for a **CTS** signal, in milliseconds. After turning the **RTS** signal on, a timer is started to wait for a **CTS** signal. If this timer expires, a Driver aborts sending bytes through the serial port. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle** and the **IO.Serial.WaitCTS** property is configured to True.

IO.Serial.DataBits

Specifies the number of data bits to configure a serial port. Possible values are **5**: Five data bits, **6**: Six data bits, **7**: Seven data bits, or **8**: Eight data bits.

IO.Serial.DelayAfterMs

Number of milliseconds to delay after the last byte is sent through a serial port, but before turning the **RTS** signal off. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle** and the **IO.Serial.WaitCTS** property is configured to False.

IO.Serial.DelayBeforeMs

Number of milliseconds to delay after turning the **RTS** signal on, but before data is sent. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle** and the **IO.Serial.WaitCTS** property is configured to False.

IO.Serial.DTR

Indicates how a Driver deals with the **DTR** signal. Possible values are **OFF**: **DTR** signal is always turned off or **ON**: **DTR** signal is always turned on.

IO.Serial.InterbyteDelayUs

Delay time, in milliseconds (1/1000000 of a second), for each two bytes sent through a **Serial** Interface.

IO.Serial.InterframeDelayMs

Delay time, in milliseconds, before sending a packet after the last packet sent or received.

IO.Serial.Parity

Specifies a parity for the configuration of a serial port. Possible values are **E or Even**: Even parity, **N or None**: No parity, **O or Odd**: Odd parity, **M or Mark**: Mark parity, or **S or Space**: Space parity.

IO.Serial.Port

Number of the local serial port. Possible values are **1**: Uses the COM1 port, **2**: Uses the COM2 port, **3**: Uses the COM3 port, or **n**: Uses the COM*n* port.

IO.Serial.RTS

Indicates how a Driver deals with the **RTS** signal. Possible values are **OFF**: **RTS** signal always off, **ON**: **RTS** signal always on, or **Toggle**: Turns the **RTS** signal on when transmitting data and turns the **RTS** signal off when not transmitting data.

IO.Serial.StopBits

Specifies the number of stop bits for the configuration of a serial port. Possible values are **1**: One stop bit, **2**: One and a half stop bit, or **3**: Two stop bits.

IO.Serial.SupressEcho

Use a value different from 0 (zero) to enable suppressing the echo or 0 (zero) to disable it.

IO.Serial.WaitCTS

Configure to True to force a Driver to wait for the **CTS** signal before sending bytes when the **RTS** signal is turned on. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle**.

Driver Revision History

VERSION	DATE	AUTHOR	COMMENTS
4.0.23	08/27/2025	M. Ludwig	• Driver updated to IOKit library version 3.0 and Visual Studio 2022 (*Case 37983*).
4.0.22	03/28/2025	M. Salvador	• It is now possible to declare the list of Stations using numbers or ranges of numbers, separated by commas. Example: 3,5-9. In this case, the addresses 3, 5, 6, 7, 8, and 9 are created. (*Case 37272*)
4.0.19	04/08/2024	F. Englert	• The driver now validates the MEs type data received from the application, adjusting for the limit and assigning the flag OV=1 if

VERSION	DATE	AUTHOR	COMMENTS
			they are outside the valid limit (*Case 35807*).
4.0.18	12/22/2023	M. Salvador	• Corrected eventual GPF no 104 Master when losing communication or ending the driver (*Case 35352*).
4.0.16	11/16/2023	M. Salvador	• Eventual GPF corrected when ending an application command (*Case 35170*)
4.0.15	05/08/2023	M. Salvador	• Created the N2=100 tag writing that forces a GI for the Station informed in the N1 parameter (*Case 34153*).
4.0.14	03/24/2023	M. Salvador	• Option created to discard ASDU135 to reset the channel idle time count (*Case 33908*).
4.0.13	03/22/2023	F. Englert	• Fixed an issue in the Master that could fail to send the General Inquiry (GI) in case of a quick reconnection (*Case 34029*).
4.0.12	03/21/2023	F. Englert	• Fixed the issue that caused the data sending performance to drop drastically when there were many data to send (*Case 34018*).
4.0.11	01/12/2023	M. Salvador	• Corrected the sending of TESTDT when in Idle mode in Master (*Case 33714*). • Reimplemented the InterrogationGroup tags (N2=100 to 115) that existed in the previous version (3.0), allowing granular control over which group and frequency we want to interrogate (*Case 33702*).
4.0.9	11/23/2022	F. Englert	• The master driver now sends GIs correctly when configured not to wait for the M_EI_NA or if it is not configured with a fixed list of stations (*Case 33414*).
4.0.8	11/23/2022	F. Englert	• Implemented the tag N2=993 that returns the link layer status for the station indicated in N1 (*Case 33508*).

VERSION	DATE	AUTHOR	COMMENTS
4.0.6	10/20/2022	M. Salvador	• Invalid values in objects of types DIQ and SEP no longer invalidate the GI (*Case 33329*).
4.0.5	08/02/2022	M. Salvador	• Corrected reading of the tag "Reset Process" in slave mode (*Case 33036*). • Created option "Map Overflow bit to Uncertain Quality" in Master profile M. Salvador (*Case 32653*).
4.0.4	02/01/2021	F. Englert	• The slave driver can now send Integrated Totals data (ASDUS 15, 16, and 37) (Case 30396).
4.0.2	07/13/202	F. Englert	• Fixed the issue in the Slave driver that sometimes sent copies of received packets to the master (*Case 29139*). • The driver now supports multiple master and slave stations (*Case 28195*).
4.0.1	01/03/2020	F. Englert M. Salvador	• Created an independent version for protocol **IEC 870-5-104** in **Master** and **Slave** modes. The Driver from protocol **IEC 870-5-101** is still supported in version 3.0 (101/104).
3.0.4	08/27/2019	M. Ludwig	• Driver ported to Visual Studio 2017 (*Case 27164*).
3.0.1	06/20/2017	F.Englert	• Removed protocols **102**, **ZIV**, and **Procome**, now handled by specific Drivers (*Case 22924*). • Changed the name of this Driver to IEC870-101-104.dll, which now uses the protection ID of Power Drivers and the license is interchangeable with other Drivers from the same group (*Case 13800*).
2.22.80	05/22/2017	F.Englert	• Implemented the **Disconnect if General Interrogation returns empty results** option (*Case 22256*).
2.22.77	05/11/2017	F. Englert	• Implemented the **Enable Callbacks** option also for Slave protocols **IEC 870-5-**

VERSION	DATE	AUTHOR	COMMENTS
			101 Balanced, **IEC 870-5-101 Unbalanced**, and **IEC 870-5-104** (*Case 22652*). • Analog Commands **SEn**, **SEs**, and **SEf** of **Select/Execute** with qualifiers **ShortPulse**, **LongPulse**, and **PersistentOutput** are now sent correctly (*Case 22662*).
2.22.76	04/12/2017	F. Englert	• This Driver now supports the **IP Filter** option, which allows indicating from which TCP/IP addresses this Driver accepts or rejects connections (*Case 22540*).
2.22.75	10/17/2016	F. Englert	• Added two properties in this Driver to configure timezones, in minutes, from **Master** and **Slave** Stations from which this Driver communicates. This Driver converts the computer's local timestamps to the Station's timezone and viceversa (*Case 21621*).
2.22.74	09/30/2016	F. Englert	• The Block Tag for receiving **SD**, **DC**, and **RC** Commands from this Driver in **Slave IEC 870-5-101** mode and **IEC 870-5-104** now returns an array of Variants instead of an array of integers, allowing the manipulation in scripts of an application (*Case 21588*).
2.22.72	09/01/2016	F. Englert	• Updated documentation with implementations from version 2.22, only in Portuguese (*Case 21167*).
2.22.70	08/18/2016	F. Englert	• The properties window was redesigned. The **Properties** tab now shows all configuration properties. The **Stations** tab was moved into the **IEC870** tab (*Case 21385*).
2.22.63	03/30/2016	F. Englert	• Added a property to configure the Originator Address that comes with all ASDUs sent by this Driver, if the number of octets of the

VERSION	DATE	AUTHOR	COMMENTS
			Cause of Transmission is configured with value two (*Case 21126*).
2.22.61	03/17/2016	F. Englert	• When receiving a packet with a repeated **FCB** bit, this Driver in **Slave** mode now correctly resends **ACK** or **NACK** packets (*Case 20685*, caused on v2.18).
2.22.60	02/11/2016	F. Englert	• This Driver now correctly marks as **LinkActive** the Stations from protocol **IEC 870-5-101 Unbalanced Master** after the correct initialization (*Case 20531*). • This Driver in **Master** mode of protocol **IEC 870-5-101 Balanced** now responds immediately to **RESET_LINK** Commands sent by a Slave, hence the initialization is faster (*Case 20533*).
2.22.59	02/04/2016	F. Englert	• Removed the **Respond using** option of protocol **IEC 870-5-101 Unbalanced** in **Slave** mode (*Case 20515*).
2.22.58	02/03/2016	F. Englert	• Implemented the **Procome** protocol (*Case 19889*).
2.22.55	12/04/2015	F. Englert	• Added an option to control using the **SU** flag (summer time), **Timestamp SU Flag (Summertime adjustment)** (*Case 20163*).
2.22.54	11/21/2014	F. Englert	• This Driver now stores on log, near the number of a Cause of Transmission, its meaning (*Case 17694*). • This Driver in **Slave** mode now supports two data intervals for cyclic sending (*Case 17708*).
2.22.53	11/13/2014	F. Englert	• This Driver in **Master** mode does not occupy 100% of CPU if the **Slave Polling Rate** option is configured as zero (*Case 15377*).
2.22.52	10/29/2014	F. Englert	• This Driver in **Slave** mode of protocols **IEC 870-5-101** and **IEC 870-5-104** now discard data from pending

VERSION	DATE	AUTHOR	COMMENTS
			General Interrogations (*Case 17626*). • Improved answers from this Driver in **Slave** mode and General Interrogation requests with invalid or unsupported **COT** (*Cause of Transmission*) or **IOA** (*Information Object Address*) (*Case 17629*).
2.22.51	10/28/2014	F. Englert	• Fixed a GPF that occurred in this Driver in **Slave** mode of protocols **IEC 870-5-104** or **IEC 870-5-101 Balanced** right after responding a General Interrogation (*Case 17643*, caused on v2.22.50).
2.22.50	10/23/2014	F. Englert	• Created two new configuration parameters to define whether this Driver can send Class 1 (one) data in Class 2 (two) requests, if there is no Class 2 (two) data to send, and viceversa (*Case 17608*).
2.22.46	05/23/2013	F. Englert	• This Driver in **Slave** mode now supports sending cyclic data in the **Cyclic Data Transmission: Object Range** and **Cyclic Data Transmission: Send Period (ms)** options of the **Properties** tab (*Case 13952*). • Implemented support for callbacks in this Driver in **Master** mode in the **Enable Callbacks** option of the **Properties** tab (*Case 9178*).
2.22.45	04/03/2013	F. Englert	• Fixed a problem when handling packets of this Driver in **Slave** mode of protocol **IEC 870-5-101 Balanced**, which could incorrectly discard a response from the Master, if it sends a new request before sending a response (*Case 13942*).
2.22.44	03/22/2013	F. Englert	• Fixed the packing of ASDUs 5 (five), 7 (seven), 13, 15, and 21, which could be sent using a sequence of objects, option not supported by the

VERSION	DATE	AUTHOR	COMMENTS
			protocol's standard (*Case 13898*, caused on v2.22.34).
2.22.43	02/26/2013	F. Englert	• Fixed a problem when handling packets, which could reuse old packets received before the last disconnection, leaving this Driver continually disconnecting and reconnecting. Also improved the process of forced disconnection or reconnection when detecting any sequence error in the packets (*Case 13808*).
2.22.41	11/07/2012	F. Englert	• Fixed a memory leak that occurred whenever this Driver received data and placed it on the internal cache memory (*Case 13454*, caused on v2.22.34).
2.22.37	02/15/2012	F. Englert	• Added an option **Send EI (End Of Initialization) When Connected** to the **Properties** tab of this Driver's properties window in **Slave** mode, which allows configuring whether it sends an **EI** (*End of Initialization*) Command when a Master connects (*Case 12761*).
2.22.36	11/18/2011	F. Englert	• This Driver in **Slave** mode of protocol **IEC 870-5-104** now sends ASDU 70 (**M_EI_NA_1**) when communication starts (*Case 12575*). • Bits **BL** and **OV** are now mapped in the quality's **Limit** field, that is, the two least significant bits (*Case 12569*).
2.22.35	05/13/2011	F. Englert	• Created a new option on the **Properties** tab of Master protocols, **Don't wait ACTTERM for Execute commands** (*Case 12229*).
2.22.34	04/18/2011	F. Englert	• Fixed the packing of some data types that were not allowing send data in blocks (*Case 12161*).

VERSION	DATE	AUTHOR	COMMENTS
			• This Driver in **Slave** mode now allows users to configure a dead band for sending analog values in the **Analog Deadband (% of previous value)** option of the **Properties** tab (*Case 12109*).
2.22.33	02/11/2011	F. Englert	• This Driver in **Master** mode of protocol **IEC 870-5-101 Balanced** responded the **DIR** bit with value 0 (zero) in the **RESPOND_STATUS** message, when it should respond the **DIR** bit with value 1 (one). Added the **DIR (Physical Transmission Direction)** option, which controls the behavior of the **DIR** bit in the packet of protocol **IEC 870-5-101 Balanced**, both in **Master** mode and in **Slave** mode (*Case 10174*). • Fixed the Cause of Transmission used by this Driver in **Slave** mode of protocol **IEC 870-5-101** when returning data from Commands, such as an answer for a General Interrogation, for Class one (*Case 12057*). • Protocols **IEC 870-5-101 Balanced** and **Unbalanced** of this Driver in **Slave** mode now support the **Delay Acquisition Command** (**C_CD_NA_1**, ASDU 106) (*Case 12058*). • This Driver in **Master** mode now allows enabling sending a **Test Command** whenever the application layer remains *n* seconds without receiving a message from a Slave. Also removed Tag **N2** equal to 1104, which sent a Test Command whenever that Tag was read (*Case 12042*).
2.22.32	02/08/2011	F. Englert	• This Driver, both in **Master** and in **Slave** mode, now forces a disconnection if

VERSION	DATE	AUTHOR	COMMENTS
			they do not receive a **TESTDTcon** as a response to a **TESTDT act** (*Case 12043*). • Added a **Treat First Write As** option on the **Properties** tab of this Driver's configuration window in **Slave** mode (*Case 12045*).
2.22.31	01/18/2011	F. Englert	• Added a new option **Link Layer Starts** on the **Properties** tab of this Driver's configuration window, which allows defining the initial status of the link layer, enabled or disabled (*Case 12023*).
2.22.30	12/09/2010	F. Englert	• This Driver, when in **Master** mode, now forces a reconnection if receiving a confirmation, positive or negative, of a General Interrogation failure (*Case 11931*).
2.22.29	12/08/2010	F. Englert	• This Driver now handles graciously timestamps greater than year 2035. When detecting an invalid timestamp, converts it to local date and time and enables the **IV** flag of timestamp's quality (*Case 11855*). • This Driver now groups values with different timestamp format but the same data type, eliminating the need for clients do explicitly declare the timestamp format of a Tag. The **Remove timestamp from received data** option was removed, because this operation is performed automatically. • Implemented the option of local overriding in this Driver's Tags in **Slave** mode (*Case 11662*). • Added the **Analog Values Send Interval (ms)** option to control the frequency of sending analog values by

VERSION	DATE	AUTHOR	COMMENTS
			this Driver in **Slave** mode (*Case 11732*). • Added a new property to this Driver, **IEC.MaxPacketSize**, which allows configuring the maximum size of packets assembled by this Driver, both in **Master** and in **Slave** mode (*Case 11731*).
2.22.27	08/13/2010	F. Englert	• This Driver in **Slave** mode now handles correctly Clock Syncing Commands sent by a Master, according to the handle profile configured in the properties (*Case 11660*). • This Driver now reports the value of the **SB** flag (*substituted*) in the OPC quality of Tags (Case 11661).
2.22.26	08/05/2010	F. Englert	• The **Read Slave Statistics** Tag is not updating values in any of the options available for configuration, with this Driver configured in **Slave Unbalanced** mode (*Case 11623*).
2.22.25	08/04/2010	F. Englert	• Added a new option **Common Address of ASDU** on the **Field sizes (in octets)** group of this Driver's properties window, which allows configuring a number of octets of the **Common Address of ASDU** option different from the **Slave Address** (*Case 11634*).
2.22.23	08/18/2009	F. Englert	• When the number of octets of a Cause of Transmission was 2 (two), this Driver considered the second octet a part of that Cause of Transmission, but in fact that octet is the **Originator Address**. Now, this Driver discards the **Originator Address** in the received packets. When transmitting, this Driver always sends 0 (zero) in the **Originator Address** (*Case 10723*).

VERSION	DATE	AUTHOR	COMMENTS
			• Now the configuration of this Driver is performed only on one tab, **IEC870**. On this tab, according to the selected protocol, two other specific sub-tabs are displayed, **General**, **Master** or **Slave**, **Properties**, and **Stations** (*Case 10643*).
2.22.22	08/07/2009	F. Englert	• This Driver in **Slave** mode now takes advantage of bandwidth, grouping in the same packet several unsolicited events to send them to a Master (*Case 10702*).
2.22.21	08/03/2009	F. Englert	• Now users can inform the address of objects with 3 (three) octets using parameters in **String** format in the **ParamItem** property (*Case 9396*).
2.22.20	06/29/2009	F. Englert	• Fixed the handling of reconnection operations in this Driver in **Master** mode of protocol **IEC 870-5-104** in cases when a really fast disconnection and a reconnection occurred (*Case 8039*). • Added a new **Properties** tab to this Driver. This tab allows editing, as a list, the properties of each protocol, previously available on the **IEC** tab, as well as the properties **T0**, **T1**, **T2**, **T3**, **K**, and **W** of protocol **IEC 870-5-104** (*Case 10608*).
2.22.19	05/19/2009	F. Englert	• This Driver now performs a disconnection from the physical layer if the Slave stops responding (*Case 10126*).
2.22.18	10/14/2008	F. Englert	• This Driver now forces a disconnection from the physical layer when receives a malformed or out-of-order packet (*Case 10040*).
2.22.16	03/12/2008	F. Englert	• This Driver in **Master** mode now returns automatically a General Interrogation if it

VERSION	DATE	AUTHOR	COMMENTS
			receives and indication of General Interrogation ending, that is, **cot** equal to 10, there is no active General Interrogation, and the last General Interrogation failed (*Case 9260*).
2.22.15	03/10/2008	F. Englert	• Created a new function (*N2*) to read data from this Driver's cache memory (*Case 9166*).
2.22.14	02/19/2008	F. Englert	• Added a new option on the **Commands** tab of this Driver, available only for **Master** mode, **Process data only at GI end** (*Case 9165*).
2.22.13	11/08/2007	F. Englert	• Fixed the behavior of **Read Interrogation Group** Command in this Driver in **Master** mode (*Case 8402*). • This Driver in **Slave** mode now allows configuring buffers to store Class 1 (one) and Class 2 (two) events while the Master is disconnected (*Case 7658*). • Implemented a new **Stations** tab on this Driver's configuration window. This tab allows users to configure whether this Driver creates Stations automatically, the old mode, or restricts communication to user-declared Stations, the new mode (*Case 7687*). • Now this Driver in **Master** mode of protocol **IEC 870-5-104** uses the **Wait M_EI_NA on activation** option to control the behavior when waiting for **STARTDT.CON** (*Case 8963*).
2.22.12	06/19/2007	F. Englert	• Fixed the handling of **FCB** bit, which was incremented before knowing whether a packet was successfully sent or not (*Case 8405*). • Fixed the handling of **Read Interrogation Group**

VERSION	DATE	AUTHOR	COMMENTS
			Commands in this Driver in **Slave** mode (*Case 8406*). • Implemented a discard of invalid messages. If a **RESPOND_NACK** is received as a response to a **SEND/CONFIRM**, it is discarded and this Driver keeps waiting for a **CONFIRM_ACK** or **CONFIRM_NACK** (*Case 8151*). • Added a new option on the **Commands** tab to allow waiting or not by an **ACTTEM** from General Interrogation, **Wait ACTTERM** (*Case 8401*).
2.22.11	04/13/2007	F. Englert	• Now writings to Tag **N2** equal to 1102 only return error when communication fails. If a device returns an error code, writing is considered successful and the error code returns on the writing status (*Case 8018*).
2.22.10	04/13/2007	F. Englert	• Fixed the behavior of this Driver in **Slave** mode, which was ignoring the quality of received data from an application and sending to the Master always with a **Good** quality (*Case 8111*). • Added a new Tag **N2** equal to 6 (six) in this Driver in **Slave** mode, which allows spontaneously sending data with Class 1 (one) and also in General Interrogation, if configured (*Case 7653*).
2.22.8	03/23/2007	F. Englert	• Fixed the behavior of timer **T2** in protocol **IEC 870-5-104**, which could lead this Driver to exceed the maximum time for sending an **ACK** packet, that is, 10 seconds (*Case 8038*).
2.22.7	03/06/2007	F. Englert	• Fixed the format of parameter **N2** equal to 99, which was returning a date in **Double** format from **Elipse SCADA** (*Case 7954*).

VERSION	DATE	AUTHOR	COMMENTS
			• Added to the **Commands** tab a list that allows selecting the way this Driver in **Slave** mode handles Select and Execute Commands, **default (pass to application)** (*Case 5963*).
2.22.6	02/16/2007	F. Englert	• Fixed the **Common Slave Address** of packets sent by this Driver in **Slave** mode during a General Interrogation, if users have configured a pre-defined list of points to return (*Case 7940*). • Added a new group of controls on the **Commands** tab to configure automatically sending Clock Syncing Commands, **Auto-run clock synchronization** and **Repeat every (sec)** (*Case 7509*). • Implemented support for ASDU 105 (**C_RP_NA_1**, *Reset process command*) both on **Master** mode and on **Slave** mode of protocols **IEC 870-5-101** and **IEC 870-5-104** (*Case 7836*).
2.22.3	01/10/2007	F. Englert	• Fixed the maximum size of packets of protocol **IEC 870-5-104**, which allowed generating packets with up to four more bytes than expected, provoking communication problems (*Case 7798*). • This Driver is now protected by ID 31290 (*Case 7797*). • Now quality changes to 216 when generating a timestamp. This only occurs if quality before was **Good** (192). If quality's value was any value different from 192, it does not change (*Case 6569*).
2.22.1	09/28/2006	F. Englert	• Fixed the handling of double points received with an invalid value, 00b or 11b, which were discarded if configured to handle

VERSION	DATE	AUTHOR	COMMENTS
			double points as a **Boolean** value (*Case 7423*). • Changed the handling of double points with invalid values (*Case 6571*).
2.21.1	08/18/2006	F. Englert	• Fixed the **FCB** bit sent in this Driver's packets in **Master** mode of protocols **IEC 870-5-101** and **102**, which was sent inverted (*Case 7286*).
2.20.1	08/11/2006	F. Englert	• Fixed the example script for handling Commands received by a Slave, which did not work (*Case 5452*). • Now the log registers events discarded when the internal cache memory is full, that is, more than 100 events for a single object (*Case 6563*). • Fixed an incorrect timestamp (zero) sent when using data types from protocol **IEC 870-5-104** in **Slave** mode (*Case 6564*). • Fixed the **Slave polling rate** parameter, which has no effect when communicating with a Slave (*Case 6570*). • Fixed the receiving of data with a timestamp marked as invalid (**IV** equal to one), and data itself was marked as invalid and only the timestamp should be discarded (*Case 6666*). • **Master** and **Slave** modes were extensively tested and performed small improvements (*Case 6798*). • Fixed the usage of a fixed one-second time-out, instead of using the time configured in the **App Layer Timeout** option, when users must send a Command and the application layer is busy, such as when using a General Interrogation, leading to a Command failure (*Case 6848*).

VERSION	DATE	AUTHOR	COMMENTS
			• Fixed the timestamps sent or received, which could be incorrect with a one-millisecond difference when received or sent by an application. This fix only works together with **E3** v2.5 Build 131 and **IOKit** v1.13 (*Case 7256*). • Added support for Command ASDUs 48 (**C_SE_NA_1**), 49 (**C_SE_NB_1**), 50 (**C_SE_NC_1**), and 51 (**C_BO_NA_1**) (*Case 5361*). • Added support for protocol **IEC 870-5-104** in **Slave** mode. Added Commands for ASDUs 58 (**C_SC_TA_1**), 59 (**C_DC_TA_1**), 60 (**C_RC_TA_1**), 61 (**C_SE_TA_1**), 62 (**C_SE_TB_1**), 63 (**C_SE_TC_1**), and 64 (**C_BO_TA_1**). Also added support for Test Command using ASDU 107 (**C_TS_TA_1**) (*Case 5506*). • Implemented updates in protocol **ZIV** (*Case 5846*). • Changed the reading of **VTCD** adjustments to adapt to the new format defined in protocol **ZIV** (*Case 6561*). • Added the **Modify the timestamp on received values** option on the **Interrogation** tab (*Case 6562*). • Added support for Clock Syncing Commands (ASDU 103 or **C_CS_NA_1**) in protocols **IEC 870-5-101** and **IEC 870-5-104** in **Slave** mode. Also added the **Commands** tab (*Case 6826*). • Added support for protocol **IEC 870-5-104** in **Master** mode (*Case 7259*).
2.19.1	01/17/2005	F. Englert	• Added an option to use a **Common Address** different from **Link Address** (*Case 4364*).

VERSION	DATE	AUTHOR	COMMENTS
			• Fixed a situation where Slave addresses greater than 127 were not supported when configured to use an octet for the Slave address, which should allow addresses between 0 (zero) and 255 (*Case 4766*). • Fixed a situation of not acknowledging **CC1** packets received as **ACK** packets (*Case 5210*). • Fixed a situation in which data types of protocol **IEC 870-5-104** were returning their timestamp as the value of the Tag (*Case 5211*). • Fixed the block of protocol **ZIV**, **Read Configuration Block 2**, with parameter **B2** equal to 1141, which was not working (*Case 5212*). • Now **IOKit** v1.08 is used, with support for turning on and off the runtime log.
2.18.1	07/28/2004	F. Englert	• Fixed a situation in which data from a General Interrogation was not sent in Blocks in **Master** mode of protocol **Unbalanced** (*Case 4238*). • Fixed a situation in which, only in **Slave** mode, the last packet was not resent when the Master sent a packet with the same previous **FCB** (*Case 4276*). • Added an option to select the type of response to send when receiving a request from the Master, **Respond with** (*Case 3741*). • Data types defined in protocol **IEC 870-5-104** are now available on the list of pre-defined points returned in the General Interrogation (*Case 4035*).
2.17.1	06/09/2004	F. Englert	• Added retries in link layer (*Case 3937*).
2.16.1	05/04/2004	F. Englert	• Changed the default value of application's time-out,

VERSION	DATE	AUTHOR	COMMENTS
			from 4000 to 4 (four) seconds. • Added the **Add timestamp to interrogation data** option (*Case 3608*). • Added an option to allow handling **DIQ**, **DCO**, **RCO**, and **SEP** data as **Boolean** values (*Case 3609*).
2.15.1	04/01/2004	F. Englert	• Added the **Remove timestamp from interrogation data** option (*Case 3506*). • Added support for data types of protocol **IEC 870-5-104** encapsulated in protocol **IEC 870-5-101**. Protocol **IEC 870-5-104 IS NOT** supported (*Case 3419*). • Documentation translated to Portuguese (*Case 3568*).
2.14.1	02/20/2004	F. Englert	• Timestamps sent are incorrect, that is, the computer's current time is always used (*Case 3385*).
2.13.1	01/29/2004	F. Englert	• Double Point Commands (DCO) and Simple Events for Protection Devices (SEP) now use the possible statuses, **Intermediate** or 0 (zero), **OFF** or 1 (one), **ON** or 2 (two), and **Undetermined** or 3 (three), instead of a **Boolean** value indicating an **On** or **Off** status (*Case 3214*). • This Driver is not responding to **Test Link** Commands, function 2 (two) of protocol IEC, received in **Slave** or **Balanced** mode (*Case 3231*). • Added an **Interrogation** option, which allows configuring a list of objects returned in a General Interrogation, only in **Slave** mode (*Case 3284*). • Added a **TEST LINK** Tag only in **Slave** or **Balanced** mode. This Tag sends a Test Command from the link

VERSION	DATE	AUTHOR	COMMENTS
			to the Master, returning the value one if the Master responds (*Case 3285*). • Fixed a problem in which this Driver in **Slave** or **Balanced** mode sometimes locked when receiving a message from a Master while trying to send another message to the same Master (*Case 3286*).
2.12.1	01/08/2004	F. Englert	• Double point data (**M_DP_NA_1** and **M_DP_TA_1**) now return possible statuses, **Intermediate** or 0 (zero), **OFF** or 1 (one), **ON** or 2 (two), and **Undetermined** or 3 (three), instead of a **Boolean** value indicating an **On** or **Off** status (*Case 3214*). • Added an option to enable or disable using the **CC1** frame instead of the **ACK** frame when sending Acknowledgment Messages on the link layer in **Balanced** mode (*Case 3215*).
2.11.1	12/19/2003	F. Englert	• Added handling Commands sent by a Master in **Slave** mode (*Case 3211*).
2.10.1	12/18/2003	F. Englert	• Implemented protocol **IEC 870-5-101 Slave** in **Balanced** mode (*Case 195*).
2.9.1	12/02/2003	F. Englert	• This Driver in **Slave** mode is not created anymore when in **Offline** mode (*Case 3039*).
2.8.1	11/27/2003	F. Englert	• Implemented protocol **IEC 870-5-101** in **Slave** mode (*Case 195*).
2.7.1	11/18/2003	F. Englert	• Listening mode is now working correctly, because values received in listening mode were not returned (*Case 2960*).
2.6.1	11/13/2003	F. Englert	• Errors in Tag quality are now reported if the connection is lost while in listening mode (*Case 2935*).

VERSION	DATE	AUTHOR	COMMENTS
2.5.1	11/07/2003	F. Englert	• Added a quality field to the mass memory reading, in parameter **B2** equal to 1189 or 1190 (*Case 2893*).
2.4.1	11/05/2003	F. Englert	• Added Tag **Read MAE ID**, with parameter **N2** equal to 1212 for this Driver in **Slave** mode of protocol **ZIV**.
2.3.1	09/29/2003	F. Englert	• Improved the message log in application level.
2.2.1	08/11/2003	F. Englert	• Added the documentation for extensions of protocol **ZIV**.
2.1.1	07/29/2002	F. Englert	• Driver ported to **IOKit**. • Implemented protocols **102** and **102 ZIV Extensions**.

Headquarters
Rua Mostardeiro, 322/Cj. 902, 1001 e 1002
90510-002 — Porto Alegre — RS
Phone: (+55 51) 3346-4699
Fax: (+55 51) 3222-6226
E-mail: elipse-rs@elipse.com.br

Branch in Taiwan
9F., No.12, Beiping 2nd St., Sanmin Dist.
807 — Kaohsiung City — Taiwan
Phone: (+886 7) 323-8468
Fax: (+886 7) 323-9656
E-mail: evan@elipse.com.br

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

www.elipse.com.br

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

Gartner 2014 Cool Vendor

OPC FOUNDATION MEMBER

Gartner, Cool Vendors in Brazil 2014, April 2014.
Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability of fitness for a particular purpose.

Microsoft Partner
Gold Independent Software Vendor (ISV)

Copyright © 2002-2025 Elipse Software Ltda. All rights reserved.