

IEC 61850 Driver

| | |
|---------------------|---|
| Filename | IEC61850.dll |
| Manufacturer | |
| Devices | Servers (IEDs) compatible with IEC 61850 Standard (ED1 and ED2) |
| Protocol | IEC 61850 MMS over Ethernet TCP/IP |
| Version | 3.0.44 |
| Last Update | 21/03/2024 |
| Platform | Win32 |
| Dependencies | IOKit v2.00 |
| Superblock Readings | No |
| Level | 31298 (31201 with Power license) |

Introduction

The IEC 61850 Driver communicates with protection relays and other devices using the IEC61850 protocol over Ethernet TCP/IP (MMS). This Driver allows:

- Communication with several devices on the same Driver
- Importing Tags from devices or via SCL configuration files
- Support for messages without Report (Buffered or Unbuffered) confirmation
- Polling of variables not belonging to Reports
- Support for quality and timestamp information with a one-millisecond precision
- Collecting oscillography files in **COMTRADE** format

Please check the next documents, provided with this manual, for more information about this Driver's functionality:

- Elipse IEC 61850 Client Driver PIXIT Ed2.pdf
- Elipse IEC 61850 Client Driver TICS Ed2.pdf
- Elipse IEC 61850 Client Driver PICS Ed2.pdf
- Elipse IEC 61850 Client Driver MICS Ed2.pdf

Configuration

The [P] configuration parameters of this Driver are not used. All settings are performed on the dialog box of this Driver's configurations. The configuration tabs are described on the next topics.

IEC61850 Device Config Tab

This tab allows defining devices (Servers) to establish communication.

Driver IEC 61850 v3.0.3 (IOKit v2.0.108)

Reports

Commands

Files

Setup

Ethernet

IEC61850 Device Config

IEC61850 General

XMPP

Server Name

IP Address

Browse SCL Files...

Add

Update

Delete

Server:

IED00001

IP:

127.0.0.1

Backup IP:

Disable:

☐

Use Backup IP:

☐

MMS

Rem AE Q

PSel

SSel

TSel

12

1

1

1

Rem AP ID:

1,3,999,33

XMPP

JID:

Alias:

OK

Cancel

Apply

IEC61850 Device Config tab

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

Available options on the IEC61850 Device Config tab

| OPTION | DESCRIPTION |
|------------------------------|--|
| Browse SCL Files | Tags from relays can be created online, that is, when users can communicate with relays, or offline by importing SCL files. This option opens a window to select ICD or SCD files to import. After selecting a file, for each description of relay found, an identification is created on the list of servers, and a file with the description of Logical Devices (LD) and Logical Nodes (LN) of a relay is also created, which can be imported to an application by using the Tag Browser window. In case there are no SCL files, users can configure each relay directly by using the Add , Update , and Delete options |
| Server | Informs the name of a device. Used only to identify a device for this Driver, mapping it to an IP address |
| IP | Informs an IP address of a device. Optionally, users can inform a TCP/IP port, if it is not the default port of MMS protocol, using the format X.X.X:X , such as 192.168.0.10:102 |
| Backup IP | Informs a backup IP address of a device, if available. Use the same syntax of the IP option |
| PSel (Presentation Selector) | Selection value for the device used by the OSI presentation layer (ISO/IEC 8823). It must be checked, if fixed, or informed in the device's configuration. Its value is usually 1 (one) |

| OPTION | DESCRIPTION |
|--|---|
| SSel (Session Selector) | Selection value for the device used by the OSI session layer (ISO/IEC 8327). It must be checked, if fixed, or informed in the device's configuration. Its values is usually 1 (one) |
| TSel (Transport Selector) | Selection value for the device used by the OSI transport layer (ISO/IEC 8073). It must be checked, if fixed, or informed in the device's configuration. Its value is usually 1 (one) |
| Rem AP ID (Remote Application Process Identification) | Identifier used by the OSI link layer OSI (ISO/OSI 8650), in ASN.1 (<i>Abstract Syntax Notation 1</i>) format. Indicates the format of data adopted by the AARQ (<i>Association Request</i>) function. Its value is usually 1,1,999,1,1 (iso.1.999.1.1) |
| Rem AE Qual (Remote Application Entity Qualifier) | Identifier used by the communication layer as a formatter. It must be checked, if fixed, or informed in the device's configuration. Its value is usually 12 |
| Disable | Disables this device. Therefore, when starting this Driver, there is no communication with this device |
| Use Backup IP | Informs whether the backup IP address is used |
| XMPP | Function not available on this version |

NOTE

The topic **Limitations of this Driver** contains information about the maximum number of IEDs supported by this Driver.

IEC61850 General Tab

Allows defining all other behaviors of this Driver.

Driver IEC 61850 v3.0.3 (IOKit v2.0.108)

Reports

Commands

Files

Setup

Ethernet

IEC61850 Device Config

IEC61850 General

XMPP

Transport Layer: MMS

App Category: Bay

Msg Timeout (ms): 5000

Status Check (ms): 40000

Local P Selector: 1

Local S Selector: 1

Local T Selector: 1

Local App ID: 1,1,999,1,1

LD File Path: C:

☐ Check Nameplate Mismatches
 ☒ Invert stVal BitString
 ☐ Full Log Details
 ☐ Apply Local Time Offset
 ☐ No LD Database Scan
 ☐ Block LD Cache Delete

Local AE Qualifier: 12

RFC1006 Source TSAP: 1

Proposed MMS PDU Size: 65000

OK

Cancel

Apply

IEC61850 General tab

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

Available options on the IEC61850 General tab

| OPTION | DESCRIPTION |
|--------------------------|---|
| Transport Layer | In this version, the transport layer is fixed with the value MMS (IEC 61850-8-1) . In later versions, the XMPP (IEC 61850-8-2) options is also available |
| App Category | To send commands, this Driver uses information from this option to fill the OrCat (<i>Origin Category</i>) property, which specifies the type of application sending a command, for security reasons or for solving conflicts of commands. The available options are Bay , Station , Remote , or Maintenance |
| Msg Timeout (ms) | Time to wait for a response of a command or a full message, which can be formed by several intermediate messages. The time to wait for each byte or intermediate message is defined on the Setup tab. For more information, please check topic Documentation of I/O Interfaces |
| Status Check (ms) | Time interval to send status messages, which must be answered by an IED. If there is no response, connection health can be checked, forcing an IED's disconnection and reconnection. The value of this option must be greater than the value of the Msg Timeout (ms) option and less than the disconnection interval configured on the Setup tab. For more information, please check topic Documentation of I/O Interfaces |

| OPTION | DESCRIPTION |
|--|---|
| Local P Selector | Selection value for this Driver, used by the OSI presentation layer (ISO/IEC 8823) |
| Local S Selector | Selection value for this Driver, used by the OSI session layer (ISO/IEC 8327) |
| Local T Selector | Selection value for this Driver, used by the OSI transportation layer (ISO/IEC 8073) |
| Local App ID Local Application Process Identification | Identifier used by the OSI link layer (ISO/OSI 8650), in ASN.1 (<i>Abstract Syntax Notation 1</i>) format. Indicates a data format adopted by the AARQ (<i>Association Request</i>) function. Its value is usually 1,1,999,1,1 (iso.1.999.1.1) |
| LD File Path | Default directory where this Driver stores descriptive files for each Logical Device found, to improve the initialization process. In the next initialization, if a file is found on the specified directory, whose name is given by the format SERVER_LDNAME.LD , where SERVER is the name of the device and LDNAME is the name of the Logical Device, then this LD is described based on this file. This Driver provides several ways to detect changes on IED database, to update these files |
| Check Nameplate Mismatches | Enables checking identification parameters of an IED (nameplate) when starting communication, to compare whether there were any changes. If true, a cache update (rebuilding LD files) is performed |
| Invert stVal BitString | Interpretation of bitstrings with two or more bits can be changed by this option, by changing the meaning of Open or Closed statuses in an application. Default behavior is reached by deselecting this option |
| Full Log Details | Enables inserting in this Driver's log, enabled on the Setup tab, detailed information about the notification of events for any Tag. For more information, please check topic Documentation of I/O Interfaces |
| Apply Local Offset to Timestamps | Timestamps adopted by the IEC 61850 standard always refer to UTC (<i>Universal Time Coordinate</i>) standard. By using this options, users indicate that this Driver must apply a local offset (timezone and daylight saving time) to the timestamp sent by a device |
| No LD Database Scan | Informs that no Logical Device or Logical Node request must be processed for IEDs. This option can be used when this Driver is used only to transfer files |
| Block LD Cache Delete | In cases where the initialization process is incomplete, or when any inconsistency is detected between cache files (LD) and IED's or Server's current database. This Driver deletes and updates cache files. Enable this option to block file removal |
| Local AE Qual - Local Application Entity Qualifier | Identifier used by the link layer as a formatter. Its value is usually 12 |
| RFC 1006 Source TSAP | This Driver uses RFC 1006 specification as a way to transport ISO packets over TCP. To do so, users must inform the TSAP (<i>Transport Service Access Point</i>) used to |

| OPTION | DESCRIPTION |
|------------------------------|---|
| | establish a connection using this protocol. Default value is 1 (one) |
| Proposed MMS PDU Size | When the Transport Layer option is configured as MMS , this option defines the size of the PDU negotiated with the server. Default value is 65000 bytes |

Reports Tab

Driver IEC 61850 v3.0.7 [BETA Nov 27 2020 19:28:07] (IOKit v2.0.116) X

| IEC61850 Device Config | | IEC61850 General | | XMPP | |
|--|----------|------------------|-----|-------|----------|
| Reports | Commands | Files | PRP | Setup | Ethernet |
| <input checked="" type="checkbox"/> Prefer Buffered Report Control Blocks (uses Unbuffered if not available) <input type="checkbox"/> Poll Tags not found in any Report <input type="checkbox"/> Force Tag Polling Individually using N1 Parameter <input type="checkbox"/> Check Report Revision Mismatch <input type="checkbox"/> User-Defined Report List <input type="checkbox"/> RW Report List File (.RPT) <input type="checkbox"/> User-Defined Datasets <input type="checkbox"/> RW Dataset List File (.CDS) <input type="checkbox"/> Check BRCB Entry ID <input type="checkbox"/> RW EntryID File (.EID) <input type="checkbox"/> Check BRCB TimeOfEntry <input type="checkbox"/> Wait for Gtw Ref on Startup <input checked="" type="checkbox"/> Copy All Values before Reporting Auto Intg Rpt (s) 0=disable: <input type="text" value="20"/> <input type="checkbox"/> Use Exclusive URCB's Polled Intg Rpt (s) 0=disable: <input type="text" value="0"/> <input type="checkbox"/> Use Quality Change Trigger | | | | | |
| OK | | Cancel | | Apply | |

Reports tab

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

Available options on the Reports tab

| OPTION | DESCRIPTION |
|---|--|
| Prefer Buffered Report Control Blocks (uses Unbuffered if not available) | IEC 61850 protocol uses Report objects to notify client applications about changes on data. Reported data are defined by users in a DataSet . Each Report can only be linked to a single DataSet and a device can have several Reports and DataSets . Please check section Prefer Buffered Report Control Blocks for more information about this option |
| Poll Tags not Found in any Report | When a Tag is not found in any Report , this option allows performing a cyclical reading (polling) of this Tag according to its scan rate. Remember that this method is not the most indicated nor the most efficient, because it may be subject to slow data updates and loss of fast events |
| Force Tag Polling Individually using N1 Parameter | A polling request for specific variables can be defined with a value different from 0 (zero) in the <i>N1</i> parameter if this |

| OPTION | DESCRIPTION |
|---------------------------------------|---|
| | option is configured |
| Check Report Revision Mismatch | Instructs this Driver to check the version of Reports when starting communication. In case these versions are incompatible, an update of cache (LD) files is then performed |
| User Defined Report List | If users do not want this Driver to perform an automatic search for Reports , they can define a list of Reports that must be enabled. Please check section User-Defined Report List for more information about this option |
| RW Report List File (.RPT) | Indicates whether the list of Reports defined in the previous item must be saved to a file, so that the example script of section User-Defined Report List is not needed. Writing to this file can be performed automatically by writing to the Tag or by directly editing that file, which must be on the default directory of cache (LD) files and must have the name of the IED and an .rpt extension. Only one IED is allowed per file. Please check section Format of RPT Files for more information |
| User Defined Datasets | When using a pre-defined list of Reports , users can inform whether DataSets are declared dynamically by this Driver. Please check section User-Defined Datasets for more information about this option. It is advisable to create a configuration script on this Driver's AfterStart event |
| RW DataSet List File (.CDS) | Indicates whether the list of DataSets defined in the previous item must be saved to a file, so that the execution of the example script on section User Defined Datasets is not needed. Writing to this file can be performed automatically based on the previous examples or by directly editing that file, which must be on the default directory of cache (LD) files and must have the name of the IED and a .cds extension. Only one IED is allowed per file. For more information, please check section Format of CDS Files |
| Check BRCB Entry ID | By using this option, users inform that when enabling a Buffered Report (BRCB), they must configure the <i>EntryID</i> parameter that contains a unique identifier for each Report message processed. So, this option allows restarting an application, or when there is a switch-over of a redundant application, that the Report sends only messages not yet processed. When selecting this option, the application must keep synced the <i>EntryID</i> parameters of each Buffered Report in use and also perform writings during this Driver's initialization, informing the last received value. For more information, please check topic Using EntryID |
| RW EntryID List File (.EID) | Indicates whether EntryID values exposed in the previous item must be saved to a file, so that users do not need to execute scripts to retrieve or inform a field's value during initialization. These files are saved to the default directory of cache (LD) files and must have the name of the IED and an .eid extension. Only one IED is allowed per file. For more information, please check topic Using EID Files |

| OPTION | DESCRIPTION |
|---|--|
| Check BRCB TimeOfEntry | This option was deprecated and must be replaced by EntryID |
| Wait for Gtw Ref on Startup | Enables the automatic Report EntryID for Gateway Applications. Please check the topic Gateway Configuration . |
| Copy All Values Before Reporting | If the DataSet linked to a Report contains objects with quality and timestamp after the corresponding properties, this option can be selected to avoid a notification of value to an application happens without the correct update of quality and timestamp |
| Auto Integrity Rpt (s) | If the informed value is different from 0 (zero), this Driver informs this value when enabling a Report as the interval that an unsolicited general interrogation (GI) message is generated by the device |
| Use Exclusive URCBs | Defines if, when enabling an Unbuffered Report (URCB), this Driver is set to exclusive mode (the Reserved property configured to one) or not (the Reserved property configured to zero). A URCB enabled in exclusive mode cannot be used by other clients |
| Use Quality Change Trigger | Indicates an option of trigger by change of quality that must be used by default, when users are not using a fixed list of Reports or if the TrgOps option was not informed |

Prefer Buffered Report Control Blocks

Reports can have two types, **Buffered** and **Unbuffered**. **Buffered** means that all changes on the **DataSet** elements during a disconnection are stored on buffers, so that a client application receives a notification about all those changes, as long as there is sufficient space in memory on the device and the settings are correct. This type of **Report** is used mainly for SOE (Sequencing of Events) data types. **Unbuffered Reports**, on the other hand, only store the last value. Both **Reports** can be configured to send events spontaneously, at cyclical intervals or wait for a client application to explicitly request, by polling or general interrogation (GI), data from that **Report**. However, users must notice the following situations:

- If two client application, such as two **E3** or **Elipse Power** applications, are connected to a device, only one of those applications can connect to each **Buffered Report**. This is due to the fact that, when sending data, it is deleted from the **Report's** internal buffer.
- Two or more client applications can connect to the same **Unbuffered Report**, in case they are not using the exclusive usage option.

This way, the **Prefer Buffered Report Control Blocks** option instructs this Driver, whenever a Tag enters a communication process (enter in advise or in scan), to search among all **Buffered Reports** of a device if the specified Tag belongs to the respective **DataSet** of each **Report**.

If not found, then the **Report** is enabled by this Driver, if not yet, and starts receiving change notifications. Otherwise, this Driver starts to repeat the same search process, now on **Unbuffered Reports** for the same Tag. If found, the **Report** is enabled, if not yet. If the answer is negative again, the Tag can communicate in polling mode if the **Poll Tags not found in any Report** is enabled.

If the **Prefer Buffered Report Control Blocks** option is disabled, this Driver repeats the previous procedure, but directly searching for **Unbuffered Reports** and discarding the search for **Buffered Reports**.

User-Defined Report List

- **TagName:** UserReportList
- **Device:** ServerName
- **Item:** UserDefinedReportList

```
Sub DRV_61850_AfterStart()
'The next script creates an array
'containing the configuration of a Logical Device and its corresponding Report,
'which are activated when writing to the UserReportList Tag.
'Notice that this is a two-position array.
'If users need to activate more Reports,
'configure the size of that array

Dim arr(1)
arr(0) = Array("LogicalDeviceName", "LLN0$BR$brcbEV101")
arr(1) = Array("LogicalDeviceName", "LLN0$BR$brcbEV102")
Set Cmd = Application.GetObject("DriverName.IEDName.UserReportList")
Cmd.WriteEx(arr)
End Sub
```

NOTE

When selecting the **User Defined Report List** option, this Driver only finishes the initialization process after receiving a writing to the **UserReportList** Tag.

Format of RPT Files

```
Number_Of_LogicalDevices
LogicalDeviceN; Number_Of_Reports
ReportName; ReportOption1:ReportOption1Value; ReportOptionN:ReportOptionNValue
```

Available options for the RPT file format

| PARAMETER | DESCRIPTION |
|---|--|
| Number_Of_LogicalDevices | Total number of Logical Devices in this file |
| LogicalDeviceN;Number_Of_Reports | For each Logical Device, insert a name and how many Reports are used |
| ReportName | Right after the name of each Logical Device, follows a list with the Reports used and optional fields |
| ReportOption;ReportValue | The table Optional fields contains optional fields allowed in an RPT file |

Optional fields

| FIELD | DESCRIPTION |
|---------------------------|--|
| DatSet:DataSetName | Name of a DataSet that must be linked to the Report |
| TrgOps:Options | Trigger options for the Report . The Options value corresponds to a mask of six bits with the values 0 : Not used, 1 : DataChange, 2 : Quality Change, 3 : Data Update, 4 : Integrity, and 5 : GI (<i>General Interrogation</i>) |

| FIELD | DESCRIPTION |
|-------------------------|---|
| IntgPd:Period | Period, in milliseconds, for integrity |
| BufTm:Value | Timing period to wait for new events before sending, after the notification of the first event to send, in milliseconds |
| Resv:Value | Use of URCB in exclusive mode. Possible values are 0 (zero) or 1 (one) |
| ResvTms:Value | Time to wait for a disconnection so that BRCB is reserved for the same client connected previously |
| PurgeBuf:Options | Deletes the buffer of messages. Possible values are 0 : Not Set (do nothing), 1 : Set Always, or 2 : Set if EntryID Write Fails |
| RptID:RptName | Identifier of a Report |
| OptFlds:Options | Optional fields for a Report message. The Options value corresponds to a mask of 10 bits with the values 0 : Reserved, 1 : SequenceNumber*, 2 : ReportTimeStamp*, 3 : ReasonForInclusion*, 4 : DataSetName**, 5 : DataReference, 6 : BufferOverflow*, 7 : EntryID*, 8 : ConfRevision*, and 9 : Segmentation. NOTE : Fields marked with * are used by default when the OptFields field is not specified. The DataSetName fields is mandatory and all other parameters are fixed and cannot be changed |

Example of an RPT file:

```
1
Device;1
LLN0$BR$BRCB1;ResvTms:1000
```

User-Defined Datasets

- **TagName**: DeclareDSList
- **Device**: ServerName
- **Item**: DeclareClientDSList

Users must inform an array of descriptors, each descriptor containing an array with three elements, the name of the Logical Device, the name of the **DataSet** (add an @ character at the beginning of this name if volatile) and a unique index for each **DataSet**.

```
Dim arr
arr = Array("LogicalDeviceName", "DataSetName", 1)
Set Cmd = Application.GetObject("DriverName.IEDName.DeclareDSList")
Cmd.WriteEx(arr)
```

- **TagName**: PopulateDSList
- **Device**: ServerName
- **Item**: PopulateClientDSList

Users must inform an array, and each item must contain an array with two elements, **DataSet**'s index and the name of the LN/DO/DA in the format "LogicalDevice\LN\$DO\$DA".

```
Dim arr(4)
arr(0) = Array(1, "LDName\GGIO1$ST$stval")
arr(1) = Array(1, "LDName\GGIO2$ST$stval")
arr(2) = Array(1, "LDName\GGIO3$ST$stval")
arr(3) = Array(1, "LDName\GGIO4$ST$stval")
arr(4) = Array(1, "LDName\GGIO5$ST$stval")
Set Cmd = Application.GetObject("DriverName.IEDName.PopulatedDSList")
Cmd.WriteEx(arr)
```

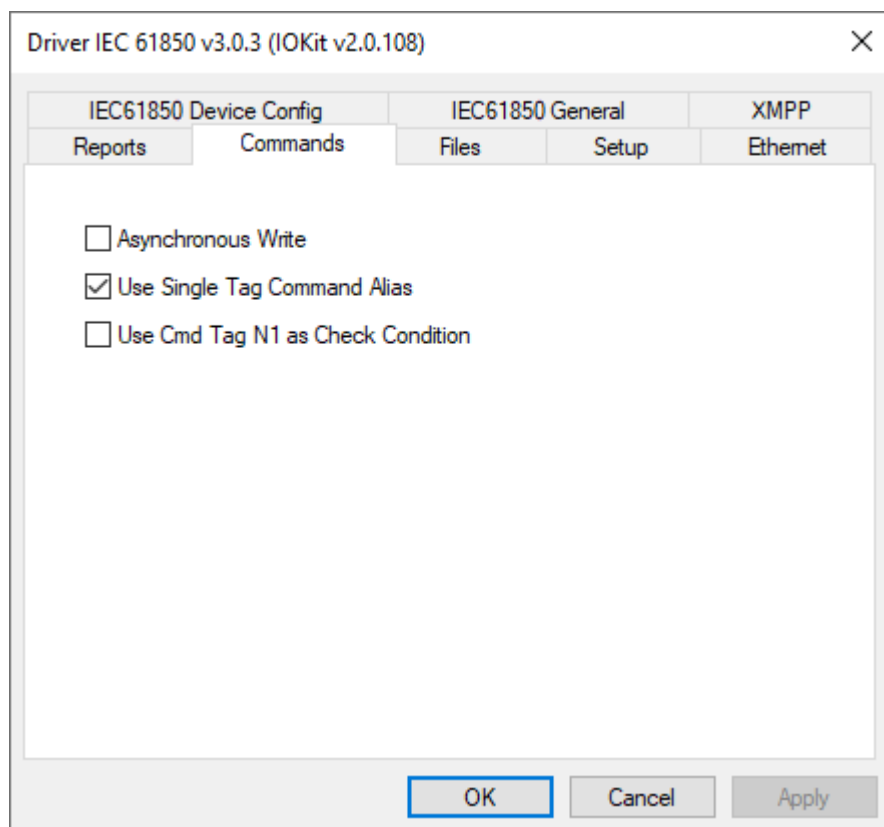
Format of CDS Files

```
Number_Of_LogicalDevices
LogicalDeviceN; Number_Of_DataSets
DataSetName; NumberofMembers
MemberNames1...N
```

Example of a CDS file:

```
1
Device;1
MyDataset1;3
Device/LLN0$DC$NamPlt
Device/LLN0$ST$Mod
Device/LPHD1$DC$PhyNam
```

Commands Tab



Commands tab

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

Available options on the Commands tab

| OPTION | DESCRIPTION |
|--|---|
| Asynchronous Write | Allows request commands to return success immediately, without waiting for an answer, as long as its sending was successful, such as when there is an active connection. This behavior aims to facilitate operations involving sending a large number of commands, to increase general speed. However, it does not check for failures during the processing of that command on the device |
| Use Single Tag Command Alias | Instructs this Driver to create during the Tag Browsing process (please check the next note) a single Tag to send commands. If this option is not selected, then uses a Block Tag with five Elements. For more information, please check topic I/O Tags |
| Use Cmd Tag N1 as Check Condition | In case of using a simple Tag to send commands, this option informs whether the <i>N1</i> parameter of this Tag is used to indicate a security check parameter of a command. The <i>N1</i> parameter, in this case, must be a number between 0 (zero) and 3 (three), as the result of a logic OR between bits 0 : INTERLOCKING and 1 : SYNCHROCHECK |

NOTE

If users are using **E3** or **Eclipse Power** as an OPC server, the command Tag can have its reading enabled, that is, the **AllowRead** property configured to True. This allows this Tag to receive the initial value of the **CtlVal** property of the referred command object, so that an OPC client previously knows the data type. After sending a command, this Driver returns an automatic reading for the same Tag, with the value written in case the operation is successful. This way, the *WriteFeedBackMode* parameter of this Driver, which indicates how a Tag receives a confirmation of the operation by keeping the written value, can be configured in **0**: WaitNextRead, preferably.

Comtrade / Events Tab

Driver IEC 61850 v3.0.38 Beta [BETA Jan 20 2023 16:33:29] (IOPKit v2.0.125) X

| | | | |
|------------------------|-------------------|------|----------|
| IEC61850 Device Config | IEC61850 General | XMPP | Reports |
| Commands | Comtrade / Events | PRP | Setup |
| | | | Ethernet |

☒ Save Comtrade Files
 ☐ Define Parameters per IED
 ☐ Delete Files after Upload
 ☐ Browse Root Folder Only

Start Transfer
 ☐ Wait for Write of LastComtradeFile Time
 ☐ Save/Get LastComtradeTime from file
 Max time waiting after startup (s):

Directory Check Interval (s):

Comtrade Upload Interval (s):

Comtrade Path:

☐ Name+Index File Transfer: (File root name)

☐ Fixed Comtrade Path on Device:

Comtrade / Events tab

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

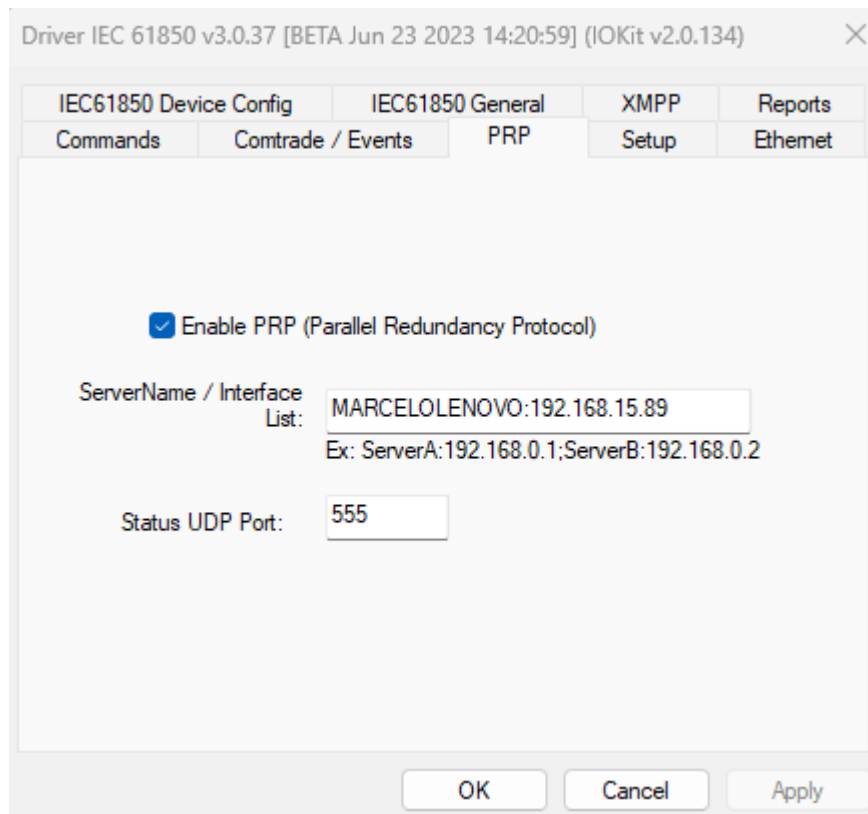
Available options on the Files tab

| OPTION | DESCRIPTION |
|----------------------------------|--|
| Save Comtrade Files | Indicates whether this Driver must perform a search for COMTRADE files, saving them on the directory indicated in the Comtrade Path option |
| Define Parameters per IED | <p>Indicates whether the parameters:</p> <ul style="list-style-type: none"> - Save Comtrade Files; - Delete Files after Upload; - Browse Root Folder Only; - Comtrade Path; - Use Fixed Comtrade Path on Device - Name+Index File Transfer e File Root Name <p>will be defined per IED and not globally for the whole driver.</p> <p>The individual definition can be done through writing the driver internal parameters via IOKIT, as explained at the item "Dynamic Configuration".</p> |
| Delete Files After Upload | Instructs this Driver to delete a device's file after successfully finishing the transfer |
| Browse Root Folder Only | Some relays insert COMTRADE files on a COMTRADE folder from the root folder, and other relays insert these files on the root folder, indicating as |

| OPTION | DESCRIPTION |
|---|--|
| | part of the file name that folder, such as "\\COMTRADE\\File.cfg". Select this option for devices with that last behavior |
| Save/Get Last Comtrade time from File | Gets/Sets the last Comtrade file time from a file COMTRADE.LST. |
| Max Time Waiting after Startup | Tells how many seconds the driver shall wait for the tag LastComtradeFileTime to be written, or by the correct read of the file COMTRADE.LST. If this time expires the driver will start transferring the first file available. If this value is 0, it waits indefinitely. |
| Wait for Write of LastComtradeFileTime | With this option selected, this Driver waits for a writing operation with the time of the last transferred COMTRADE file per IED, so that it can compare this time with the list of existing files on the IED, thus defining which file to transfer |
| Comtrade Directory Check Interval | Interval, in seconds, in which this Driver requests a list of files from a device. This Driver compares the date of the last transferred file with the files on the list, and the most recent file or files are transferred |
| Comtrade Upload Interval (s) | In case of transferring more than one file, this Driver waits the time defined in this option to perform each transfer, to separate a specific time window for this activity. However, receiving events and sending commands are not interrupted while transferring a file |
| Comtrade Path | Indicates a directory where this Driver saves COMTRADE files |
| Name+Index File Transfer | Informs the file transfer will happen based on a root name + index, instead of date/time comparison, which is the default procedure. The root name shall be informed at the property "File Root Name". Example: If an IED generates files following the rule Osc1.dat/cfg, Osc2.dat/cfg e and so on, you should check this option and inform the text "Osc" as File Root Name. |
| Fixed Comtrade Path on Device | Indicates that a Driver must ignore a search for the COMTRADE folder on the device, assuming a fixed folder informed in the corresponding field |

PRP Tab

Through this tab it is possible to configure PRP (Parallel Redundancy Protocol).




PRP Tab

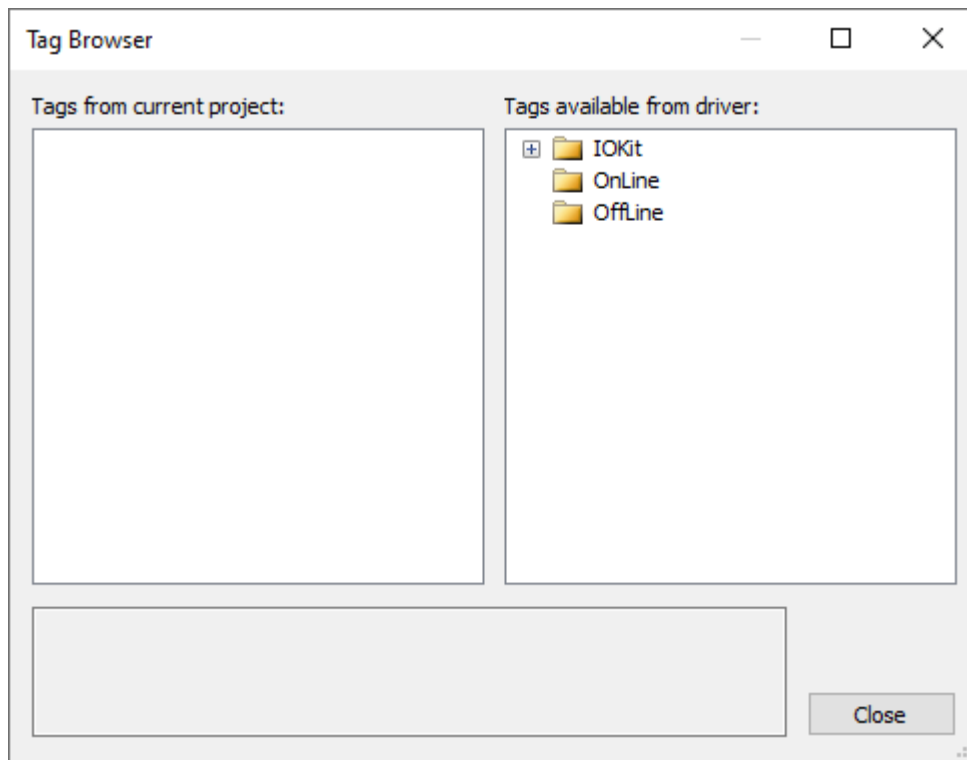
The available options at this tab are explained below. For more information, please consult **PRP Redundancy** section.

PRP Tab available options

| OPTION | DESCRIPTION |
|----------------------------------|--|
| Enable PRP | Indicates if this driver shall use the PRP Service. |
| ServerName/Interface List | Inform a list of server names (DNS) and IP Addresses with the format <i>ServerA:IPA;ServerB:IPB</i> , so the same PRJ file containing this driver can be used without changes in at least 2 different computers. When the driver starts, the local computer name is obtained and compared to the server names of the list. If a match is found, the corresponding IP Address will be set as the Virtual interface. |
| Status UDP Port | Inform an available local TCP/IP port, which will be used by IEC 61850 driver to receive statistics and logs from the ElipsePRPSvc.exe service. |

Tag Browser

Click **Tag Browser**  on this Driver's toolbar in **E3** or **Elipse Power** to open the Tag Browser window, which allows listing and dragging to this Driver the Tags identified in a device, according to the next figure.



Tag Browser window

The list **Tags available from driver** contains the following folders:

- **IOKit**: Contains standard Tags from **Elipse Software's IOKit** library, allowing to read or write general connection parameters and status, among others
- **Online**: Displays Servers declared on the **IEC61850 Device Config** tab. When clicking a Server, this Driver tries to communicate with the device, creating a new child folder for each Logical Device (LD) found in the device. Click an LD to retrieve all respective Tags
- **Offline**: Displays all Servers and LDs, but clicking an LD displays Tags from LD files created in the import directory, whether from importing SCL files or by executing an application

To use these Tags in an application, drag them from the list **Tags available from driver** to the list **Tags from current project**.

NOTE

This Driver inserts in each Logical Device a **DataSets** folder, where users can find all **DataSets** of that Logical Device referenced by a **Report**.

About IEC 61850 Standard

IEC 61850 standard was developed primarily for communication with digital protection relays. Each relay or device internally contains the following structure:

- **Logical Devices (LD)**: These are mapped logical devices that correspond to a real device, such as a Bay, mapped inside a relay. A Logical Device is always mapped to a single IED (*Intelligent Electronic Device*), because Logical Devices are not distributed
- **Logical Nodes (LN)**: These are functions from a real device mapped to a Logical Device. For example, a virtual representation of a switch (*Circuit Breaker*) in a Bay is a Logical Node, which contains a standard class name defined as XCBR. Usually a Logical Device is formed by several Logical Nodes

- **Data Objects (DO) and Data Attributes (DA):** Information inside a Logical Node is organized in sets of specific data (Data Objects) and the properties inside each set are Data Attributes. The set of all Data Attributes from a Data Object is called CDC (Common Data Class)
- **Functional Constraint (FC):** These are specific services that can be used in each Data Attribute. FCs may also be part of a variable's path, depending on the configuration of a device

The format used by this Driver for all Tags representing Data Objects or Data Attributes is the following:

- **Device:** Server:LD, such as "ArevaP139:UPC12AL1Control"
- **Item:** LN\$FC\$Data\$DataAttribute, such as "XCBR1\$ST\$Mod\$stVal"

Tag Reference

This section contains information about the configuration of **I/O Tags** and **Internal Tags**.

I/O Tags

The *N* parameters of I/O Tags are not used. Tags are addressed only using the *Device* and *Item* parameters.

- **Device:** Server:LD, configured on the **IEC61850 Device Config** tab
- **Item:** LN\$Data\$DataAttribute or LN\$FC\$Data\$DataAttribute, according to the next table

Available options for the Item parameter

| ITEM | OPERATION | MEANING |
|--|---------------|--|
| LN\$Data\$DataAttribute or LN\$FC\$Data\$DataAttribute | Read or write | Any Tag to read according to the configuration of Report parameters or polling. If this Tag is for writing, it is accepted by this Driver |
| LN\$RP\$urcbXXXX or LN\$RP\$bcrbXXXX | -- | This Driver does not display the tree corresponding to Reports , for simplicity and Tag saving. The whole handling is performed internally. Starting with version 1.0.14, this Driver displays the TimeofEntry Tag (reading and writing) only on Buffered Reports . Starting with version 1.1.18, also displays the EntryID Tag on Buffered Reports |
| LN\$CO\$xxxx or LN\$SP\$xxxxx | Write | Command or control block. To simplify the use of commands, this Driver provides, instead of a tree with the control object, a Block Tag with five Elements, described on the Elements of a command or control block table. The immediate result of accepting or not a command can be retrieved via script, using a Block Tag's WriteEx method on the <i>wWriteStatus</i> parameter, or via status block. Please check the next item for more information |

| ITEM | OPERATION | MEANING |
|--|-----------|---|
| LN\$CO\$xxxx\$Oper, LN\$CO\$xxxx\$TimeActOper, LN\$SP\$xxxxx\$SBO, LN\$CO\$xxxx\$SBOw, or LN\$CO\$xxxx\$Cancel | -- | If the Use Single Tag Command Alias option is selected, this Driver creates in Tag Browser a single Tag instead of a Block Tag with five Elements. In this case, sending a command is performed according to section Sending Commands with a Single Tag |
| LN\$CO\$xxxx or LN\$SP\$xxxxx | Read | Command status block. For each block or command Tag a second block is created, with two Elements, that displays the status of operations. This status can be changed as the result of denying to send a command, such as an unsupported command, as the result of accepting, such as a command accepted or not, or as the result of an effective action, such as opening or closing a switch. The first Element of this Block Tag (Status) contains a number code and the second Element (StatusText) contains a textual description of that number code. Please check table Number codes of a command status block for more information |

Elements of a command or control block

| ELEMENT | DESCRIPTION |
|------------------|---|
| Operation | Must receive the operation to perform in Text format. Possible values are OPERATE , SELECT , SELECTWITHVALUE , or CANCEL |
| Value | Value to send, depending on the type of object, in the CtlVal property |
| Time | Time instant, in Visual Basic standard in days since 1900, for use in a SELECT command by time and filling the T property |
| Test | Indicates whether a command is a test operation (value equal to one) or not (value equal to zero) |
| Check | Type of check performed before performing a command. It must be a number between 0 (zero) and 3 (three) as the result of a logic OR between bits 0 : INTERLOCKING and 1 : SYNCHROCHECK |

Number codes of a command status block

| STATUS | STATUSTEXT |
|----------|---------------|
| 0 | Terminated OK |

| STATUS | STATUSTEXT |
|--------|--|
| 2 | Select Accepted |
| 3 | Select With Value Accepted |
| 4 | Cancel Accepted |
| 5 | Operate Accepted |
| 50 | Terminate Fail |
| 100 | AppError:Unknown |
| 101 | AppError:NotSupported |
| 102 | AppError:BlockedBySwitchingHierarchy |
| 103 | AppError:SelectFailed |
| 104 | AppError:InvalidPosition |
| 105 | AppError:PositionReached |
| 106 | AppError:ParameterChangeInExecution |
| 107 | AppError:StepLimit |
| 108 | AppError:BlockedByMode |
| 109 | AppError:BlockedByProcess |
| 110 | AppError:BlockedByInterlocking |
| 111 | AppError:BlockedBySynchrocheck |
| 112 | AppError:CommandAlreadyInExecution |
| 113 | AppError:BlockedByHealth |
| 114 | AppError:1_Of_N_Control |
| 115 | AppError:AbortionByCancel |
| 116 | AppError:TimeLimitOver |
| 117 | AppError:AbortionByTrip |
| 118 | AppError:ObjectNotSelected |
| 200 | WriteError:Object_invalidated |
| 201 | WriteError:Hardware_fault |
| 202 | WriteError:Temporarily_unavailable |
| 203 | WriteError:Object_access_denied |
| 204 | WriteError:Object_undefined |
| 205 | WriteError:Invalid_address |
| 206 | WriteError:Type_unsupported |
| 207 | WriteError:Type_inconsistent |
| 208 | WriteError:Object_attribute_inconsistent |
| 209 | WriteError:Object_access_unsupported |
| 210 | WriteError:Object_non_existent |
| 211 | WriteError:Object_value_invalid |
| 212 | WriteError:Error_unknown |

| STATUS | STATUSTEXT |
|--------|----------------------------|
| 213 | WriteError:Timeout |
| 214 | WriteError:OutOfMemory |
| 215 | WriteError:DecodeError |
| 216 | WriteError:WrongParameters |
| 217 | WriteError:CmdnotSupported |

Sending Commands with a Single Tag

- Tag's value is the value used in the **CtlVal** property
- Tag's timestamp is used in the **T** property
- The **Test** property has a fixed value of 0 (zero)
- The **Check** property has a fixed value of 3 (three, INTERLOCKING and SYNCHROCHECK), except when the *N1* parameter is enabled in the option **Use Cmd Tag N1 as Check Condition**

This operation is defined in the **Item** property with the **Oper** (*Operate*), **SBO** (*Select Before Operate*), **SBOw** (*Select Before Operate With Value*), or **Cancel** suffixes.

The Tag for the **TimeActivatedOperate** operation, with a **TimeActOper** suffix, is available when the LN contains the **OpertTm** property. This operation occurs in a way similar to the **Operate** command, except that the Tag's timestamp is used as the operation's time and a scheduled operation can be canceled using the **Cancel** Tag.

Internal Tags

| D E V I C E | ITEM | OPERATION | MEANING |
|--|--------------|-----------|---|
| S e r v e r N a m e | ServerStatus | Read | Returns the internal status of this Driver. If the <i>Device</i> parameter is only equal to <i>ServerName</i> , then this status is relative to the device as a whole. Possible values are 0 : Starting, 1 : Reading LD files, 2 : Retrieving directory (GetServerDirectory), 3 : Processing a single LD (Tag Browsing), 4 : Retrieving a directory of files, 5 : Processing LDs, 6 : Finishing, 7 : Finished OK, 8 : Finishing with error, 9 : Finished with error, 10 : Reconnecting, or 11 : Finished (waiting) |
| S e r v e r N a | ServerStatus | Read | If the <i>Device</i> parameter is equal to <i>ServerName:LDName</i> , then status is relative to the <i>Logical Device</i> . Possible values are 0 : Waiting, 1 : Reading LD file, 2 : Retrieving LD directory, 3 : Retrieving LN directory, 4 : Retrieving data definition, 5 : Creating a cache of Tags, 6 : Discovering Reports , 7 : Retrieving DataSets , 8 : Retrieving the content of DataSets , 9 : Retrieving Nameplates, 10 : Retrieving versions of Reports , 11 : Checking versions, 12 : Programming Reports , 13 : Operation (runtime), 14 : Finishing, or 15 : Finishing with error |

| D E V I C E | ITEM | OPERATION | MEANING |
|--|------------------------|-----------|--|
| | | | |
| | | | |
| | | | |
| | | | |
| n e : L D N a m e | | | |
| S e r v e r N a m e | ComtradeTransferStatus | Read | Returns the activity status of file transfers via COMTRADE. Possible values are 0 : Not connected, 1 : Connected, 4 : Waiting to list files, 5 : Retrieving a list of files, 6 : Files transferred (synced), 7 : Waiting to transfer files, 8 : Transferring file, 101 : Empty list of files in the relay, 102 : Format errors in the reading command, 103 : Error when saving COMTRADE file, or 104 : Error when transferring file |
| S e r v e r N a m e | LastComtradeFileName | Read | Informs the name of the last transferred COMTRADE file |
| S e r v e r N a m e | ServerRebuild | Read | Removes cache files (.LD) and restarts communication with a device. The <i>Device</i> parameter can have the same name of the IED, with or without a Logical Device |
| S e r v e r N a m e | UserDefinedReportList | Write | Allows defining a list of user-created Reports |

| DEVICE | ITEM | OPERATION | MEANING |
|-------------|----------------------|----------------|--|
| Server Name | LastComtradeFileTime | Read and write | <p>Indicates the date of the last collected COMTRADE file. When starting an application and avoid files already collected to be retrieved again, this Tag must receive a writing with the date of the last existing file. So, It is recommended that an application stores the value of this Tag when quitting and write back this value when starting, or use the options Save/Get Last Comtrade Time from File.</p> <p>When the option Name+Index File Transfer is set, this tag indicates the last file transfer index, and it can also be R/W. Important to observe that if the IED is totally reset, the IED file index can return to an initial value (ex: 1), so if you want the new files to be transferred starting at this index, it is necessary to write at this tag the new value. (Note: Files are overwritten if names are duplicated).</p> |
| Server Name | ComtradeInfo | Read | Returns a Block Tag with three Elements with COMTRADE properties. Values for Elements are 0 : ComtradeTransferStatus, 1 : LastComtradeFileTime, and 2 : LastComtradeFileName |
| Server Name | BRCBActualList | Read | Returns a Block Tag with four Elements with a list of Buffered Reports in use. Values for Elements are 1 : LDName, 2 : ReportName, 3 : TimeOfEntry, and 4 : EntryID |
| Server Name | URCBActualList | Read | Returns a Block Tag with two Elements with a list of Unbuffered Reports in use. Values for Elements are 1 : LDName and 2 : ReportName |
| Elips | InternalClock | Read | Returns a Block Tag with two Elements with the current time and quality from internal clock. Values for Elements are 0 : Current time (TIME) and 1 : TimeQuality. The meaning of the bits of Element 1 (one) is 7 : Leap Seconds Known, 6 : Not used, 5 : Sync Error, and 4-0 : Precision (in power of 2-y) |

| D E V I C E | ITEM | OPERATION | MEANING |
|---|--------------------|-----------|--|
| | | | |
| | | | |
| | | | |
| | | | |
| e C l i e n t I n f o | | | |
| - | ServerInitialBuild | Write | Allows changing IEDs parameters before starting communication. Please check section ServerInitialBuild Item for more information |
| - | GatewayRef | Write | Block tag with 4 elements which allows to inform an external gateway reference to a 61850 object, through a write operation. The elements are: 0: IEDName (Name of 61850 Server configured at the driver) 1: LDName (Logical Device Name) 2: Object (61850 object/item ex: LLN0\$ST\$Data\$DataAttr) 3: Reference (String that represents this item address in an external driver, like DNP. Ex: "202;34" indicates an object 2 variation 2 and index 34 in a DNP Slave driver. For more information please check the topic Gateway Configuration . |
| - | FinishedGtwRef | Write | IOTag which allows to inform through a write operation, the end of gateway reference addition. For more information please check the topic Gateway Configuration . |
| - | GatewayEventOK | Write | Block tag with 4 elements, which informs that an event was confirmed by a 3rd party master in an external protocol in a gateway application, allowing the next EntryID to be set at the EID File, in order to avoid event loss or duplication on startup. The elements are: 0: Reference: (String that represents this item address in an external driver, like DNP. Ex: "202;34" indicates an object 2 variation 2 and index 34 in a DNP Slave driver). 1: Value (Value of the received event) 2: TSValid (Tells if the timestamp is valid and is being used at the slave protocol) 3: Timestamp (at the format of seconds since 1970) |

| D E V I C E | ITEM | OPERATION | MEANING |
|--|--------------------------|-----------|--|
| | | | For more information please check the topic Gateway Configuration . |
| S e r v e r N a m e | IPSelect | Read | Informs which IP address is in operation, in cases of using main and backup IP addresses |
| S e r v e r N a m e | IOKitEvent | Read | Returns IOKit events for the specified IED especificado, according to the Read Driver Events Tag available on topic Documentation of I/O Interfaces |
| S e r v e r N a m e | GetFileDirectory | Read | Informs a list of existing files in the IED after a GetFileDirectory (write) request. If there is no previous request or if there is no data available, returns an empty list |
| S e r v e r N a m e | FinishedWriteTimeOfEntry | Write | Please check topic Reports Tab for more information |
| S e r v e r | FinishedWriteEntryID | Write | Please check topic Using EntryID for more information |

| D E V I C E | ITEM | OPERATION | MEANING |
|--|----------------------|-----------|---|
| N a m e | | | |
| S e r v e r N a m e | IPSwitch | Write | Requests a switch from the main IP address to the backup address or vice versa |
| S e r v e r N a m e | DeclareClientDSList | Write | Declares a list of DataSets defined in the client. Please check topic Reports Tab for more information |
| S e r v e r N a m e | PopulateClientDSList | Write | Populates the members of each DataSet defined in the previous item. Please check topic Reports Tab for more information |
| S e r v e r N a m e | GetFileDirectory | Write | Requests a list of files in the IED, based on a directory informed in Tag's value or on the root directory if no directory is informed |
| S e r v | GetFile | | Requests a transfer of a file with the name informed in the Tag. This file is saved with the same name on the default directory of COMTRADE files |

| D E V I C E | ITEM | OPERATION | MEANING |
|---|-------------------------------|-----------|--|
| | | | |
| | | | |
| | | | |
| | | | |
| e r N a m e | | | |
| S e r v e r N a m e | DeleteFile | | Requests a removal of a file with the name informed in the Tag directly in the IED |
| S e r v e r N a m e | CompareLogicalDeviceDirectory | | Compares the content of a Logical Device (LD) retrieved from a cache file (.LD) with the current content in the IED. If they do not match, a new cache file is generated |
| S e r v e r N a m e : L D N a m e | CompareDataDefinition | | Compares the content of a Logical Node (LN) retrieved from a cache file (.LD) with the current content in the IED. If they do not match, a new cache file is generated |
| S e r v e r | GetDataValues | Write | Requests a reading of an item, which can be LN , DO , or DA . Values are returned in their respective Tags |

| D E V I C E | ITEM | OPERATION | MEANING |
|---|------------------|-----------|--|
| | | | |
| | | | |
| | | | |
| | | | |
| N a m e : L D N a m e | | | |
| S e r v e r N a m e : L D N a m e | SetDataValues | Write | Requests a writing of current values of the informed item, which can be LN , DO , or DA |
| S e r v e r N a m e : L D N a m e | GetDataSetValues | Write | Requests a reading of all items belonging to a DataSet . Values are returned in their respective Tags |
| S e r v | GetAllDataValues | Write | Requests a reading of all items belonging to a Functional Constraint . Tag's value must be in the format LN\$FC . Values are returned in their respective Tags |

| D E V I C E | ITEM | OPERATION | MEANING |
|---|---------------|-----------|---|
| e r N a m e : L D N a m e | | | |
| S e r v e r N a m e : L D N a m e | DeleteDataSet | Write | Request the removal of a DataSet informed in the Tag |

ServerInitialBuild Item

In the next example, supposing there is an **IniBuild** Tag with its *Item* parameter equal to **ServerInitialBuild**, an IED with index 0 (zero) is enabled and the *TSel* parameter is changed.

```
Dim CommArr(1)
CommArr(0) = Array("IEC61850.Device[0].Disable", 0)
CommArr(1) = Array("IEC61850.Device[0].TSel", 1)
Write -1, 0, 0, 3, CommArr
Item("IniBuild").WriteEx(1)
```

Redundancy

The ways this Driver avoids receiving duplicated events during start up are the following:

- By defining the last **EntryID** of each **Report** by script
- By defining the last **EntryID** with automatic generation of a file with an .eid extension

The next topics describe each one of these ways.

Using EntryID

When selecting the **Check BRCB EntryID** option on the **Reports** tab without the automatic files (.EID) option, an application must perform the next procedures to work properly:

1. Create in the application, or import via **Tag Browser**, the **EntryID** Tags available in each **Buffered Report** and in use by this Driver. This Tag's value is a **Text**-type and represents a sequence of bytes.
2. If users do not know whether a **Buffered Report** is in use, please check the list of BRCBs, as described on topic **Reports Tab**.
3. Create in the application an Internal Tag for each **EntryID** to sync. This Internal Tag must have its **Retentive** property configured to True. **NOTE:** Starting with version 3.2 of **E3**, users can also perform this procedure using XObject Properties, which can have their **Retentive** property enabled directly.
4. Each Internal Tag, or XObject Property, must receive value changes from the corresponding **EntryID** via script or Link.
5. When starting this application, there must be a script that writes the value of each **EntryID** according to the value of the corresponding Internal Tag, which must be updated according to the value sent by the redundant station. At the end of this process, a special Tag must be written indicating that the process of writing **EntryID** properties ended and this Driver can resume initialization. This Tag must be configured with the next properties.
 - **Name:** FinishedWriteEntryID
 - **Device:** ServerName
 - **Item:** FinishedWriteEntryID

Any writing operation is enough for this process to be accepted.

NOTE

If the **EntryID** check option is configured, this Driver only resumes initialization after writing that Tag.

Using EID Files

When selecting the **R/W EntryID File (.EID)** option on the **Reports** tab, this Driver generates a file for each ID with an .eid extension, containing the last received **EntryIDs** for each **Report** in use. This file is generated in at least two seconds after receiving a Report message and also before shutting down an application. This file is generated on the same directory of cache files (.LD).

During this Driver's initialization, the application waits for the availability of this file to proceed, to retrieve the values of the processed **EntryIDs** until this operation is interrupted. In a redundant system, a user-created application must be responsible for syncing this file between main and Standby machines.

The next example, which must be executed cyclically, copies **EntryID** files among redundant servers. As this Driver is loaded by the **IOServer** module, which executes on the **System** account, the example code uses the **psExec (SysInternals)** application to elevate privileges of a Windows user, by executing a batch (.bat) file that specifies a user and a password.

NOTE

For more information about the **psExec** application, please check article *PsExec v2.2* on **Microsoft Developer Network**.

```

Sub MyTimer_OnPreset()
  Set objWScript = CreateObject("WScript.Network")
  strServer = objWScript.ComputerName
  Set sw = CreateObject("WScript.shell")
  Set fso = CreateObject("Scripting.FileSystemObject")
  Set folder = fso.GetFolder("C:\MyAppDir\MyEntryIdDir")
  'Loads .eid files
  For Each file In folder.Files
    GetAnExtension = fso.GetExtensionName(file.path)
    If GetAnExtension = "EID" then
      'Copies files
      fso.CopyFile file.path, "C:\MyAppDir\MyEntryIdDir\EID\"
    End If
  Next
  If strServer = "Server2" Then
    sw.run "C:\MyAppDir\Bat\psExec.exe -i 0 -u UserAccount -p Domain@password cmd /c C:\MyAppDir\Bat\toServer1.bat", 0
  End If
  If strServer = "Server1" Then
    sw.run "C:\MyAppDir\Bat\psExec.exe -i 0 -u UserAccount -p Domain@password cmd /c C:\MyAppDir\Bat\toServer2.bat", 0
  End If
End Sub

```

Content of **toServer1.bat** file:

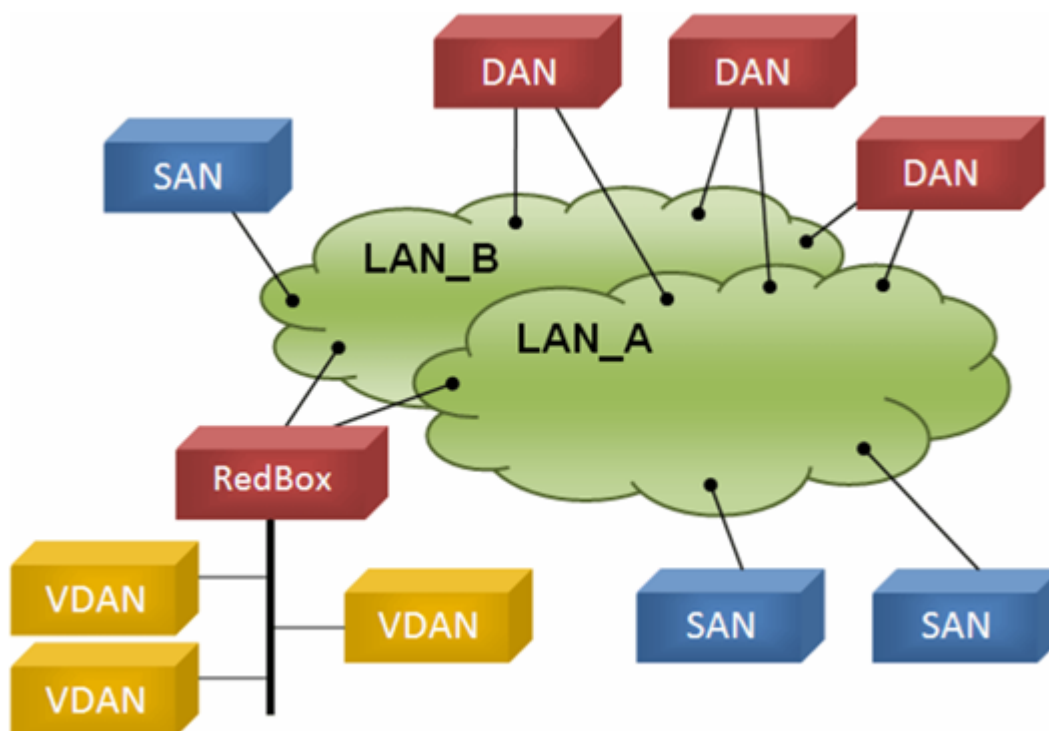
```
copy C:\MyAppDir\EntryId\EID, \\Server1\EntryID
```

Content of **toServer2.bat** file:

```
copy C:\MyAppDir\EntryId\EID, \\Server2\EntryID
```

PRP Redundancy

The PRP (Parallel Redundancy Protocol) is an Ethernet network protocol standard which allows seamless network redundancy against any simple network failure, with transparent redundancy to the application.



In a PRP architecture there must be two parallel networks (LAN_A and LAN_B), with the preferable and totally redundant adherent devices called DANs (Dual Attached Nodes), i.e. they have 2 interfaces, one on each LAN. Devices with a single

interface (called SAN - Single Attached Nodes) can also take part of the architecture, but don't benefit from redundancy, as well as other devices that do not directly support PRP (called VDANs - Virtual Dual Attached Nodes) which can be connected through a RedBox (Redundancy Box).

Eclipse IEC 61850 driver behaves as a DAN, considering that the computer shall have 2 physical network adapters and one virtual network adapter, installed and configured correctly.

The chapters below show the steps for this configuration:

- 1 - Network traffic sniffer Install and Config
- 2 - Virtual Adapter install and Config
- 3 - Physical Network Adapters Config
- 4 - EclipsePRPSvc.exe: Install, Config and Execution
- 5 - Network Monitoring and Statistics

Traffic Sniffer Library

The basic PRP Service behavior offered with this driver is based on packet capture leaving from the virtual adapter and being transferred to the physical adapters A and B. At the same time, packets received at real adapters A or B are transferred back to the virtual adapter.

In order to perform packet capture and injection to/from the adapters it is necessary to install the NPCAP traffic sniffer library, which is installed with Wireshark or obtained directly at the site <https://nmap.org/npcap/>.

In order to work correctly, it is necessary to install NPCAP with WinPcap API Compatibility mode, by choosing this option at the install dialog.

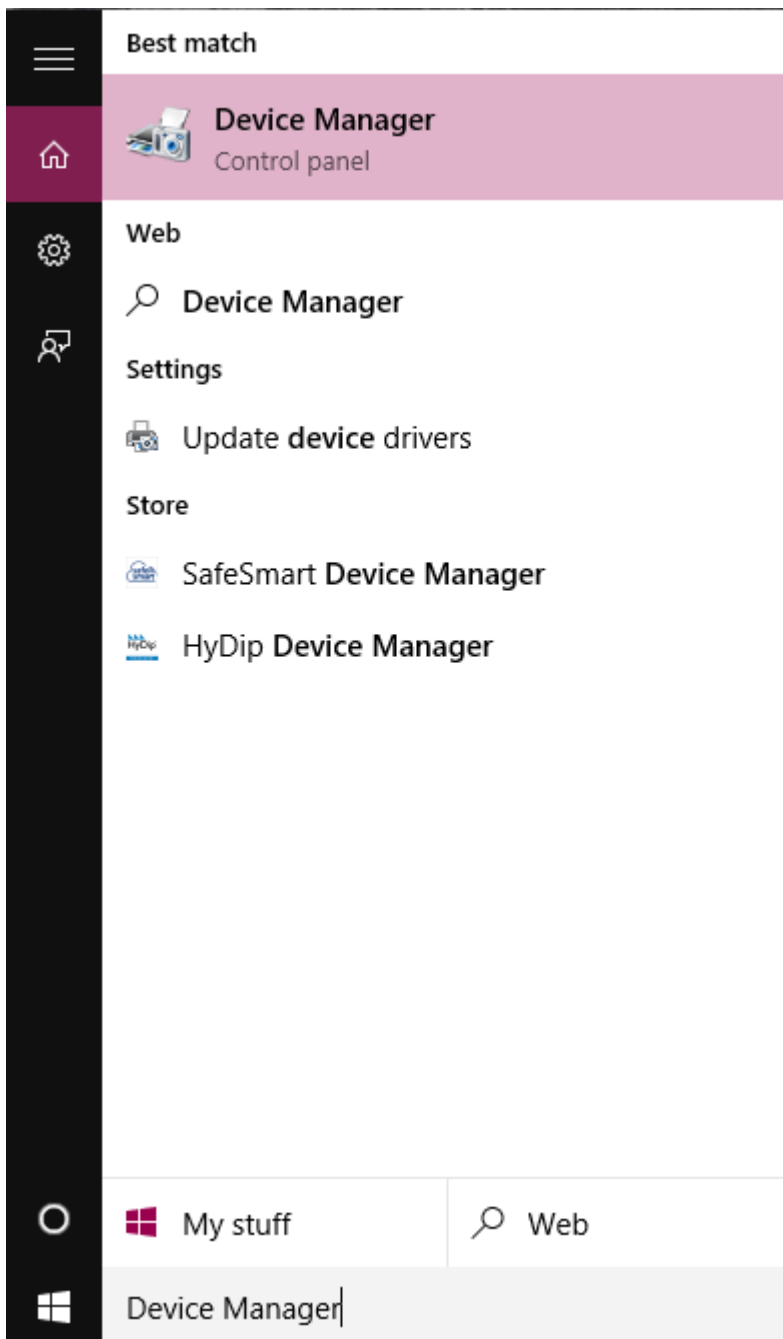
Virtual Adapter

The virtual adapter is a non-physical network adapter installed on Windows, with the objective to provide a single interface for the IEC 61850 driver (or even other protocols) independent of which physical network (A, B or both) is/are active.

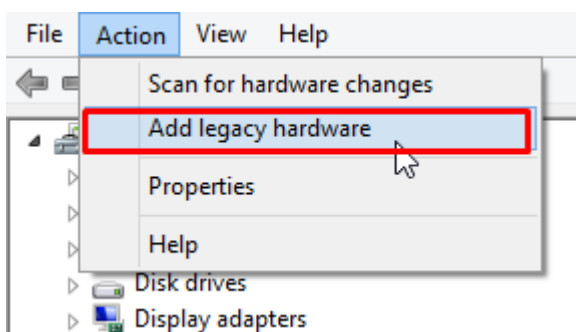
Install

The simplest way to install a virtual adapter is explained below (example for Windows 10):

- 1 - On search bar, besides the windows icon, type "Device Manager", selecting this program.



2: On Device Manager, click on Action - Add legacy hardware.



3: At the "Add Hardware" window, click at "Install the hardware that I manually select from a list (Advanced)" and after that click on Next.

The wizard can help you install other hardware

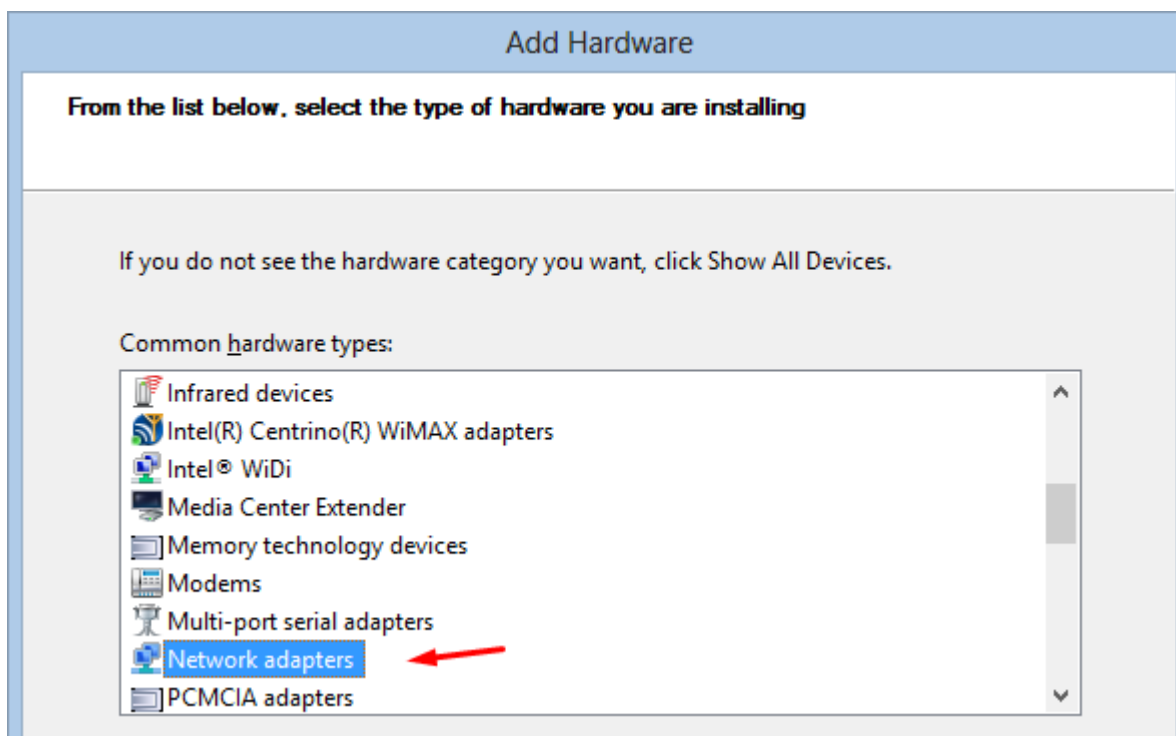
The wizard can search for other hardware and automatically install it for you. Or, if you know exactly which hardware model you want to install, you can select it from a list.

What do you want the wizard to do?

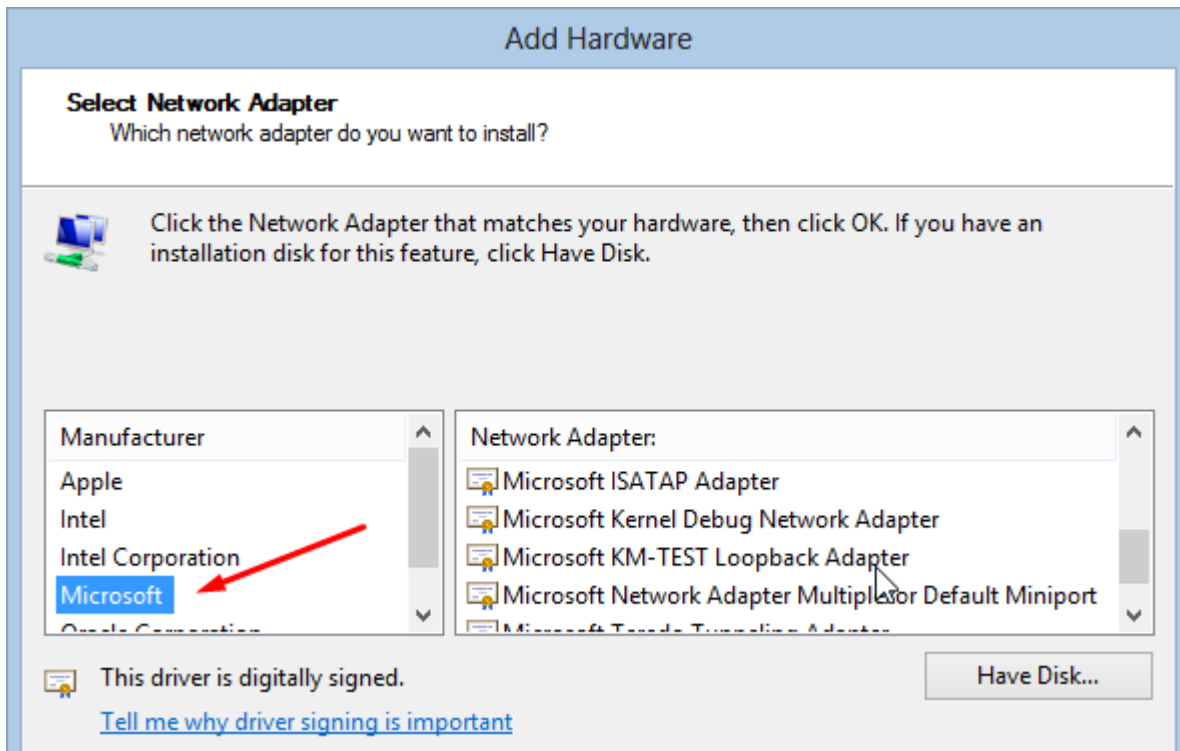
- ☐ Search for and install the hardware automatically (Recommended)
- ☒ Install the hardware that I manually select from a list (Advanced)



4: Now choose "Network Adapters" and select Next.



5: Select "Microsoft" at left panel. Next, choose Microsoft KM-TEST Loopback and hit Next until finish.



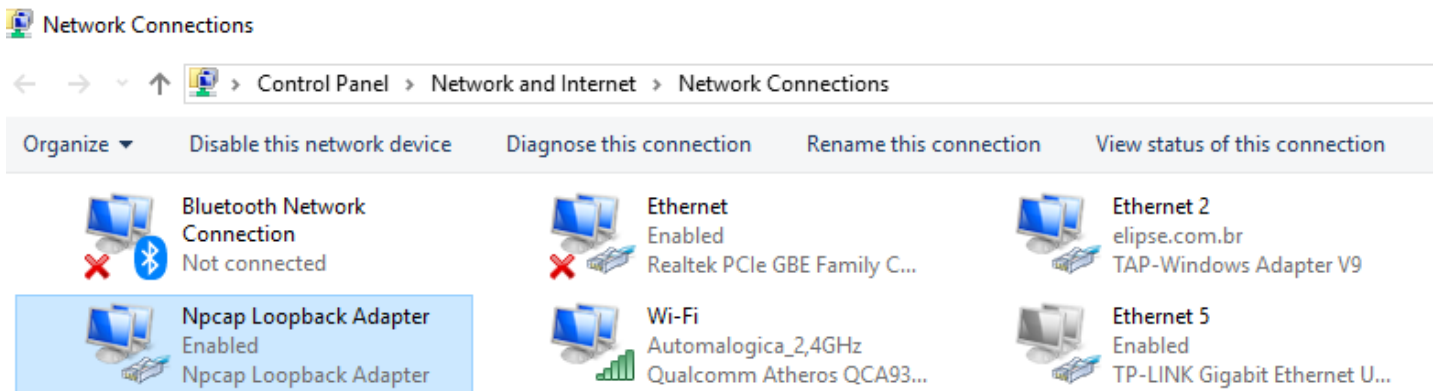
It is also possible to use the Virtual Adapter offered at NPCAP install, but it is a shortcut to the adapter above - Microsoft (KM-Test Loopback Adapter) . You can choose this option at the installed obtained at: <https://nmap.org/npcap/>

Configuration

Once the virtual adapter is installed, it is necessary to:

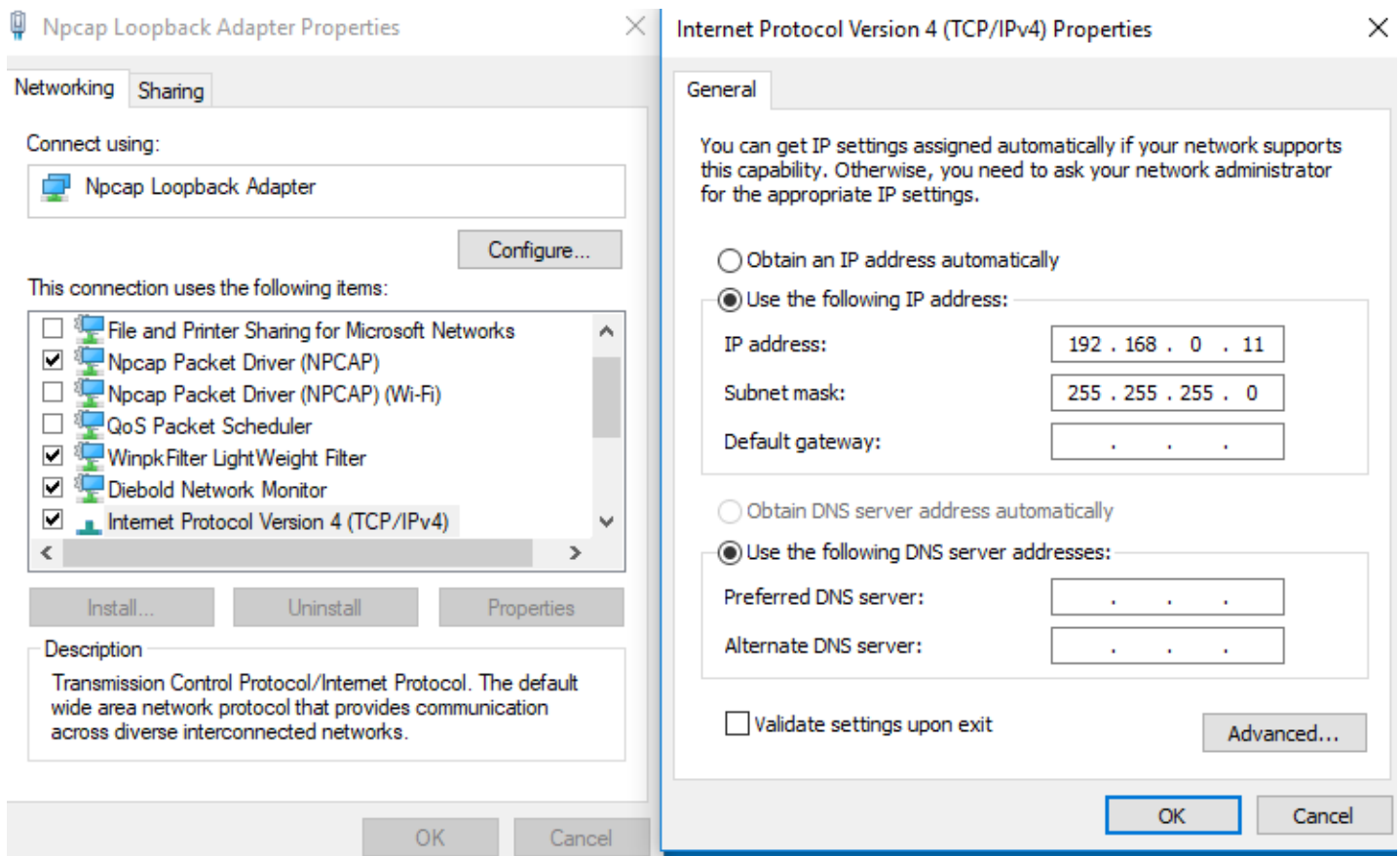
- Enable at the adapter the usage of the sniffer library npcap;
- Define a fixed IP Address.

To do it, go to Control Panel - Network and Internet - Network Connections, where all connections are exhibited. Select the Virtual Adapter properties.



At the adapter properties window, enable the minimum number of options, in order to decrease the network traffic to the IED network. The main options to enable shall be Npcap Packet Driver and TCP/IPv4.

Select TCP/IP properties and inform a valid and fixed IP Address to access the IED network.



Click at OK and apply the changes.

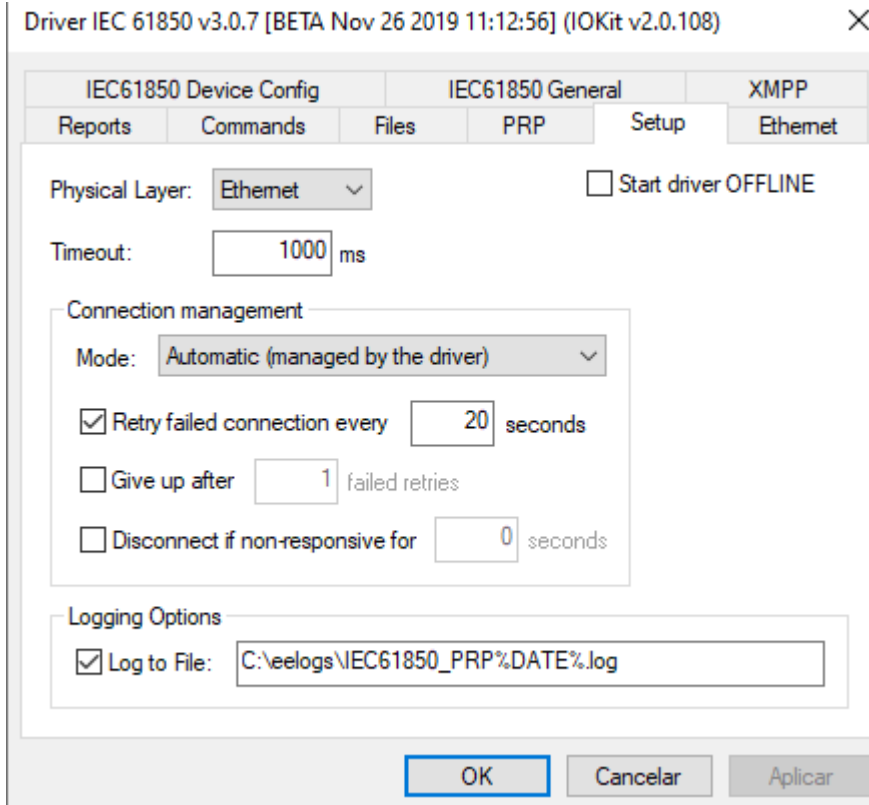
Associate the Virtual Adapter to the IEC 61850 driver

In order to assure that 61850 MMS messages will be directed **only** to the virtual adapter and not for other adapters, it is necessary to select this interface for the driver.

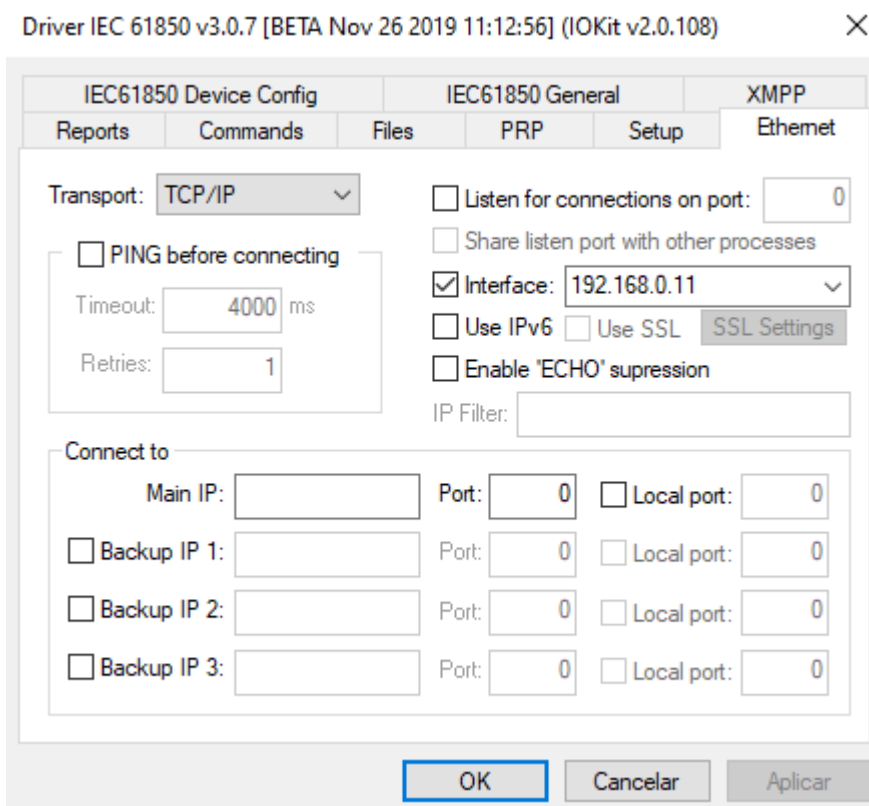
There are 2 ways to do it. First one is defining manually which adapter shall be used by the driver. The second one is using the property "ServerName/Interface List" at PRP tab, with a list of ServerNames and respective Interface IP Addresses.

Fixed Virtual Network Adapter Configuration

To define a fixed network adapter, open driver configuration, and at the Setup tab, set Physical Layer as "Ethernet".



Then, on Ethernet tab, select the option "Interface" and choose at the list, the IP Address that was defined for the Virtual Adapter. If this step is not performed, the service will not start!



Finally, return to the Setup tab and put back the Physical Layer to "None". Click on OK and Save the driver.

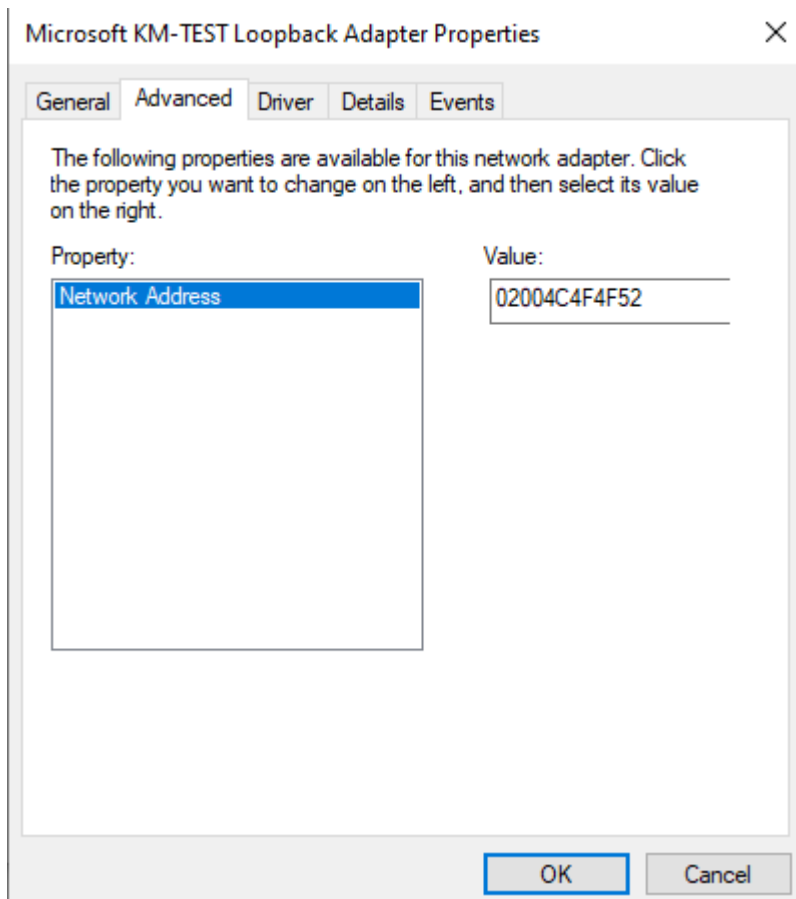
Dynamic Virtual Adapter Definition

At PRP tab, on "ServerName/Interface List", inform a list of Server Names (DNS) and it's corresponding IP Addresses at the format *ServerA:IPA;ServerB:IPB*, so the same PRJ file with this driver can be used without changes in at least 2

computers. When the driver starts, it checks the local computer name against the list, and if there is a match, it sets the corresponding interface as the adapter.

Important Observations

NOTE1: If other computers running this PRP Service are on the same network(s), it is necessary to define a different MAC Address for each virtual adapter on each computer. This is because the Virtual Adapter is installed with a standard MAC Address, which will be identical in all computers. To change the MAC Address, we shall open the adapter properties and click at the "Configure" button. Select the tab "Advanced" and at the option "Network Address" define the new MAC Address, using a 12-digit hex string.



NOTE2 : Due to a virtual adapter behavior, if you disable the adapter on windows, during the execution of the PRP Service, and enable it again, it is necessary to restart the service to make it work again.

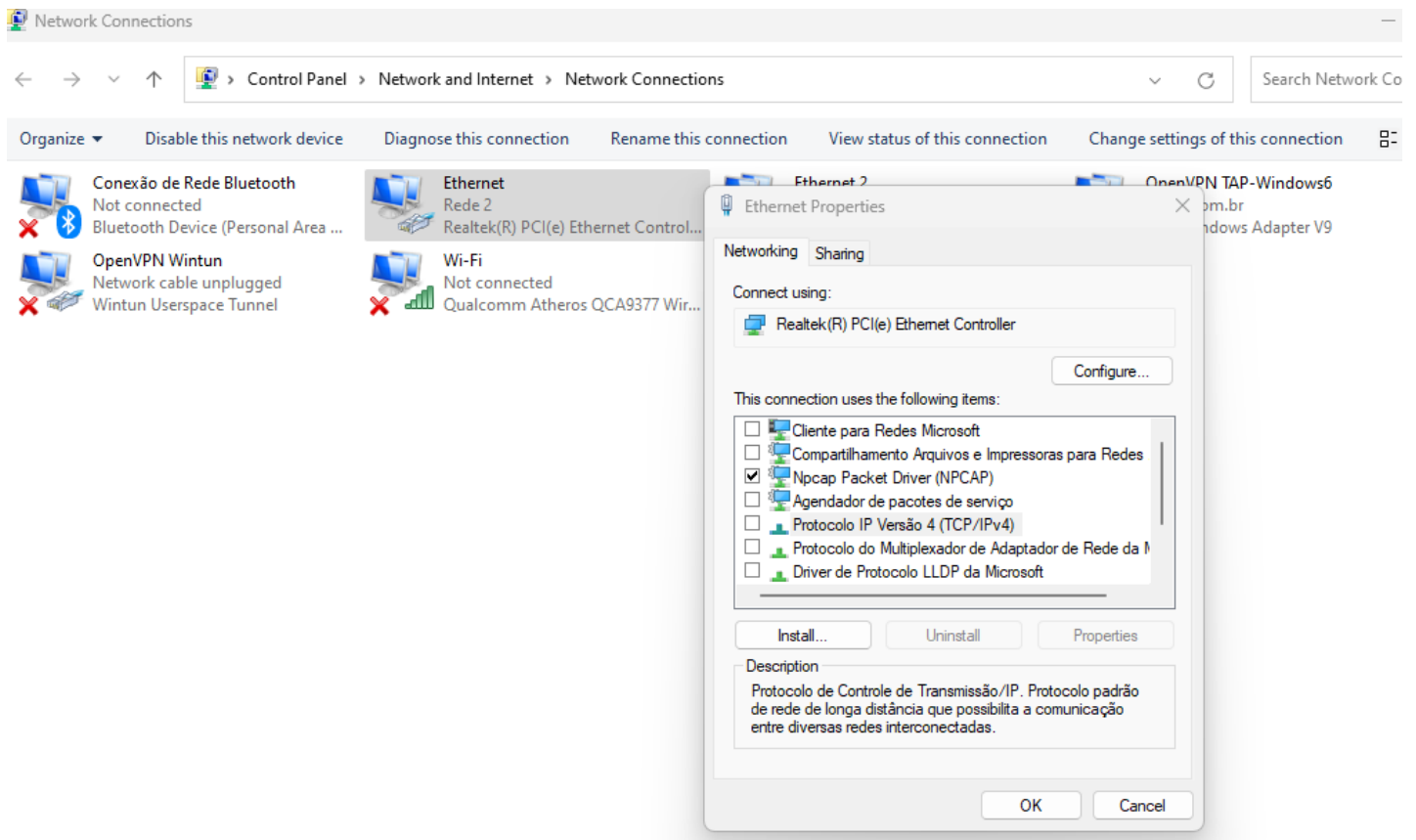
Physical Adapters

The expected behavior of the PRP option is based on two physical network adapters, one connected to the network A and the other to the network B. Anyway it is possible for the service to start and work temporarily with a single network, since it has been correctly configured at least once.

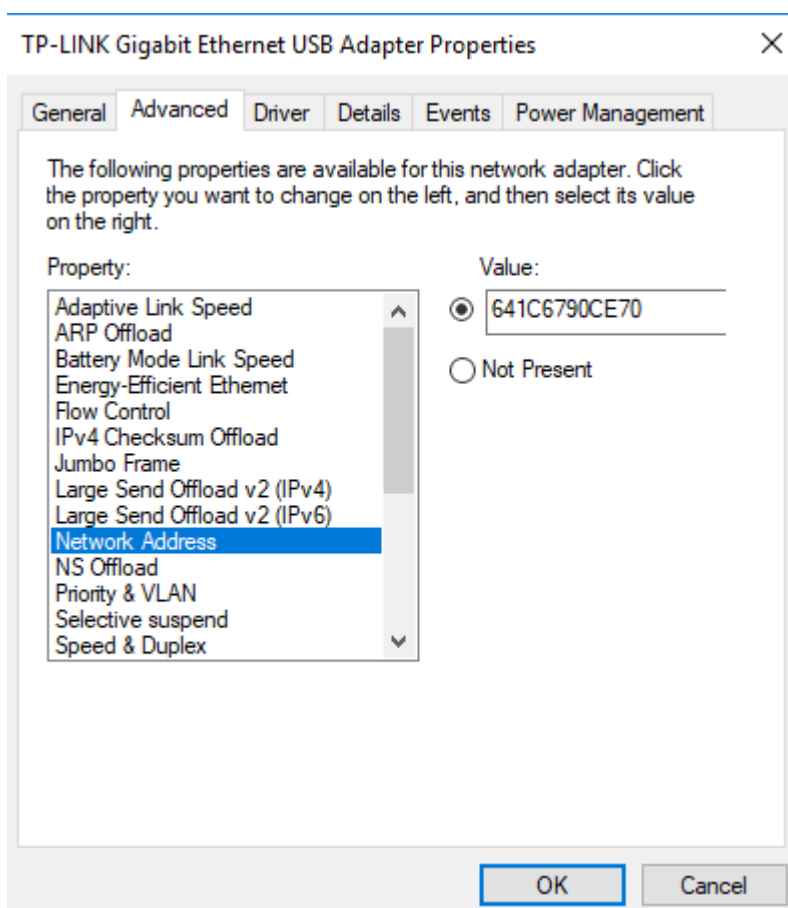
The configuration of these two real adapters shall respect 2 rules:

- 1 - They can't have an IP Address enabled, as well as other network protocols.
- 2 - Shall have the same MAC Address.

To perform this configuration, go to Control Panel - Network and Internet - Network Connections, where all adapters are shown. Select the properties of the Ethernet Adapter that will receive network A and **disable all network protocols, including TCP/IP**, leaving enabled only the sniffer library NPCAP. Click at OK.



After that, open again the adapter properties and click at the Configure button. Select the tab Advanced, and at the option "Network Address" define the desired MAC Address, at the 12 hex digits format.



Repeat the procedure for the network B adapter.

NOTE: Since both adapters shall have the same MAC, this can be obtained defining the MAC Address of the first adapter at the second or vice-versa, or even defining a new MAC for both.

In order to know the actual MAC Address of your network adapter, open a new command session (cmd.exe) and type the command "ipconfig /all".

Install, Configure and Execute PRP Service

Together with the IEC61850.DLL driver, starting from version 3.0.42, two other programs are provided:

- **ElipsePRPSvc.exe:** It is a windows service which implements PRP Protocol;
- **ElipsePRPTray.exe:** It is a console program (Windows Tray) which allows to configure and monitor ElipsePRPSvc.exe.

Both programs shall be copied in a folder of your choice, in order to allow PRP communication. The folder can be the same used for the IEC61850.dll or a different one. **It is also necessary to copy to the same folder, the file et32b.dll which is found at the E3 installation folder Bin or Bin32.**

In order for the service to work correctly, it has to be installed at the machine, and, at the same folder, there must be a initialization file **ElipsePRPSvc.ini**. Both configurations can be done through ElipsePRPTray.

ElipsePRPTray

ElipsePRPTray shall be run as administrator in order to allow service installation and configuration. For that, right-click at the program and choose "Run as administrator".

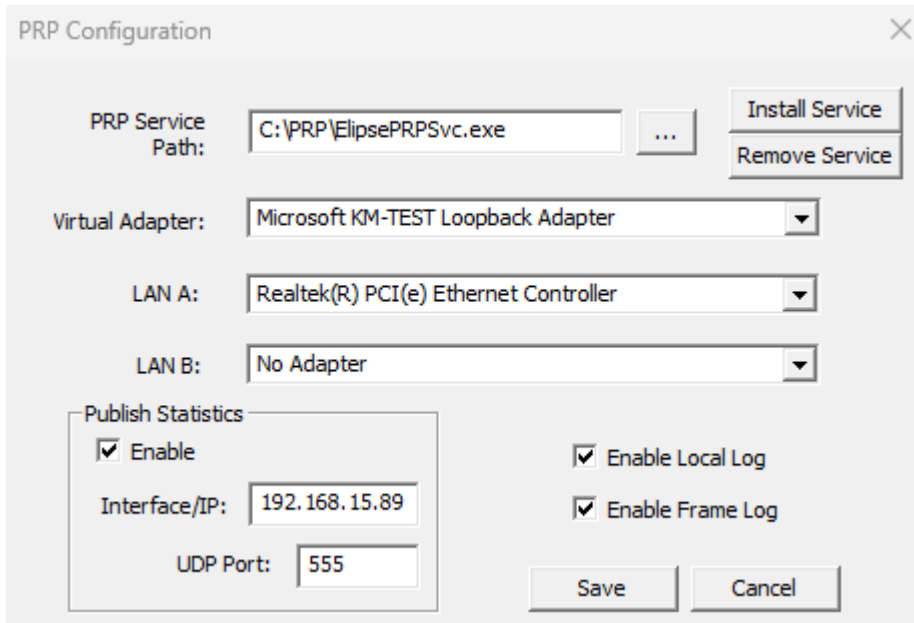
After opening, it shows a box with the following colors:

- Grey: Service not installed;
- Red: Service installed, but stopped;
- Green: Service running.



By right-clicking over the icon, the following options appear:

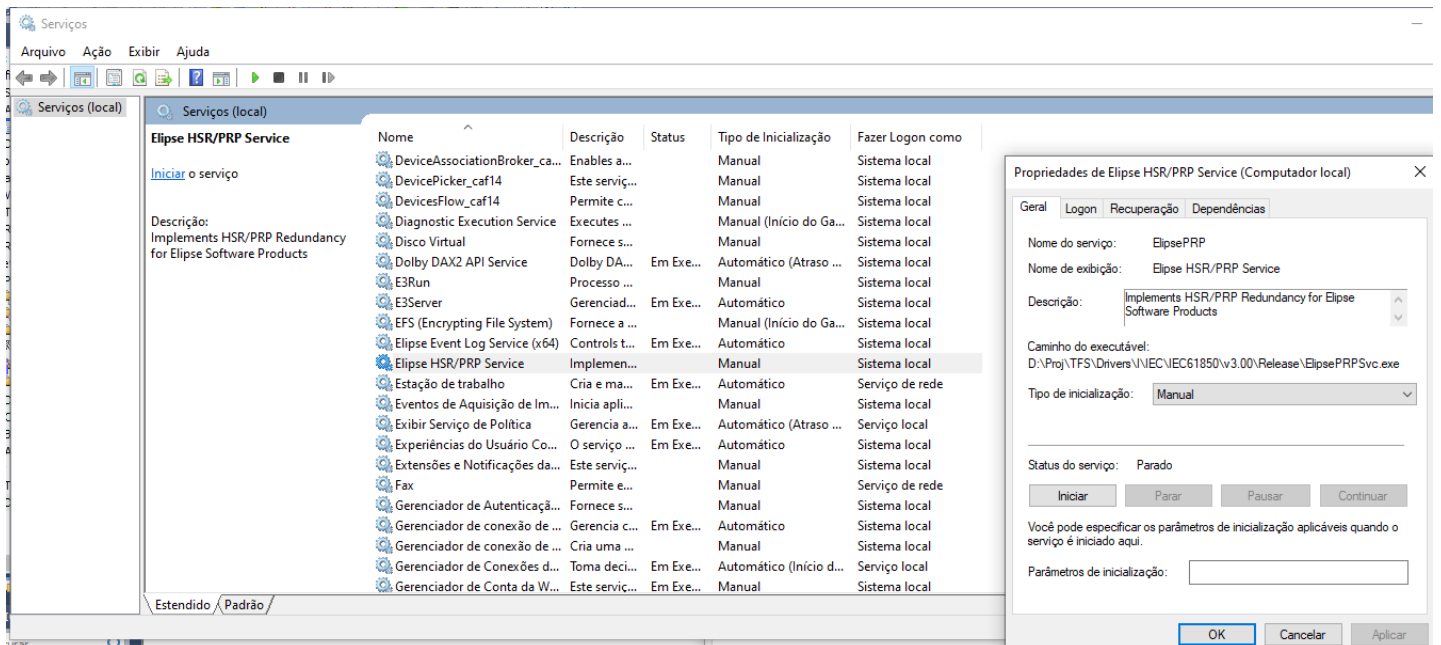
- About:** Opens a dialog box with product information;
- Exit:** Closes ElipsePRPTray program, without affecting service status;
- Configuration:** Allows to install and configure the service;
- Run/Stop:** Once installed and configured, it is possible to start or stop the service.



Configuration - Install

To install the service, it is necessary to inform the folder where the file ElipsePRPSvc.exe was copied. Once informed, just click at "Install Service". To remove the service (for example, if you need to change service folder), click at "Remove Service".

The service is installed with the Startup Option as "Automatic", when windows starts. If you want to change this configuration, type at windows search bar "Services", select the service ElipsePRPSvc - Properties and choose the Startup type.



Service Configuration

After installed, it is necessary to define the service configuration, with the following options:

- Choose at the list, the 3 network adapters (Virtual, LAN A and LAN B);
- Define if the service will publish packet statistics to IEC61850 driver, through UDP/IP messages. You shall choose the local interface IP Address and a destination port;
- Enable the basic log generation and/or packet logs.

When clicking at **Save** button, the file **ElipsePRPSvc.ini** will be created at the same folder.

Run or Stop the Service

Once the ini file is created, you can start the service, choosing the option **Run/Stop**. If the configuration is correct, the service shall start, showing the Tray icon as **green**.

Information:

- Service operation is independent from the IEC61850 driver and Elipse E3/Power application being executed or not.
- If you want to use the ElipsePRPTray every time windows starts, you should add a shortcut to windows startup group. You can do this by right-clicking at the program and choosing the option "Pin to Start".
- If you want to change the service folder after installed, it is necessary to remove the service. This can be done by ElipseTray, or manually, in 2 steps: a) Stop the service; b) Open a command window (cmd.exe with admin profile) and execute the command **sc delete ElipsePRP**.

Monitoring and Statistics

The service startup information, as configuration errors for example, are logged by the service in a file **ElipsePRPSvc.log**, generated at the same service folder, if this option is enabled.

After startup, the service starts to send other logs and statistics to an UDP port, informed at the configuration window, which is received by the IEC 61850 driver.

The log messages are redirected to the driver log, with the prefix "PRPService: XXXXXXXX".

Status information are available through 6 tags, that can be created at the driver.

| | | | | | | | | | |
|------------------------------|------------------|----------------------------|---|---|---|---|------|-------------------------------------|--|
| • PRPStat_LanA_Active | ElipseClientInfo | PRPStat_LanA_Active | 0 | 0 | 0 | 0 | 5000 | <input checked="" type="checkbox"/> | |
| • PRPStat_LanB_Active | ElipseClientInfo | PRPStat_LanB_Active | 0 | 0 | 0 | 0 | 5000 | <input checked="" type="checkbox"/> | |
| • PRPStat_LanA_TotalSent | ElipseClientInfo | PRPStat_LanA_TotalSent | 0 | 0 | 0 | 0 | 5000 | <input checked="" type="checkbox"/> | |
| • PRPStat_LanB_TotalSent | ElipseClientInfo | PRPStat_LanB_TotalSent | 0 | 0 | 0 | 0 | 5000 | <input checked="" type="checkbox"/> | |
| • PRPStat_LanA_TotalReceived | ElipseClientInfo | PRPStat_LanA_TotalReceived | 0 | 0 | 0 | 0 | 5000 | <input checked="" type="checkbox"/> | |
| • PRPStat_LanB_TotalReceived | ElipseClientInfo | PRPStat_LanB_TotalReceived | 0 | 0 | 0 | 0 | 5000 | <input checked="" type="checkbox"/> | |

They are:

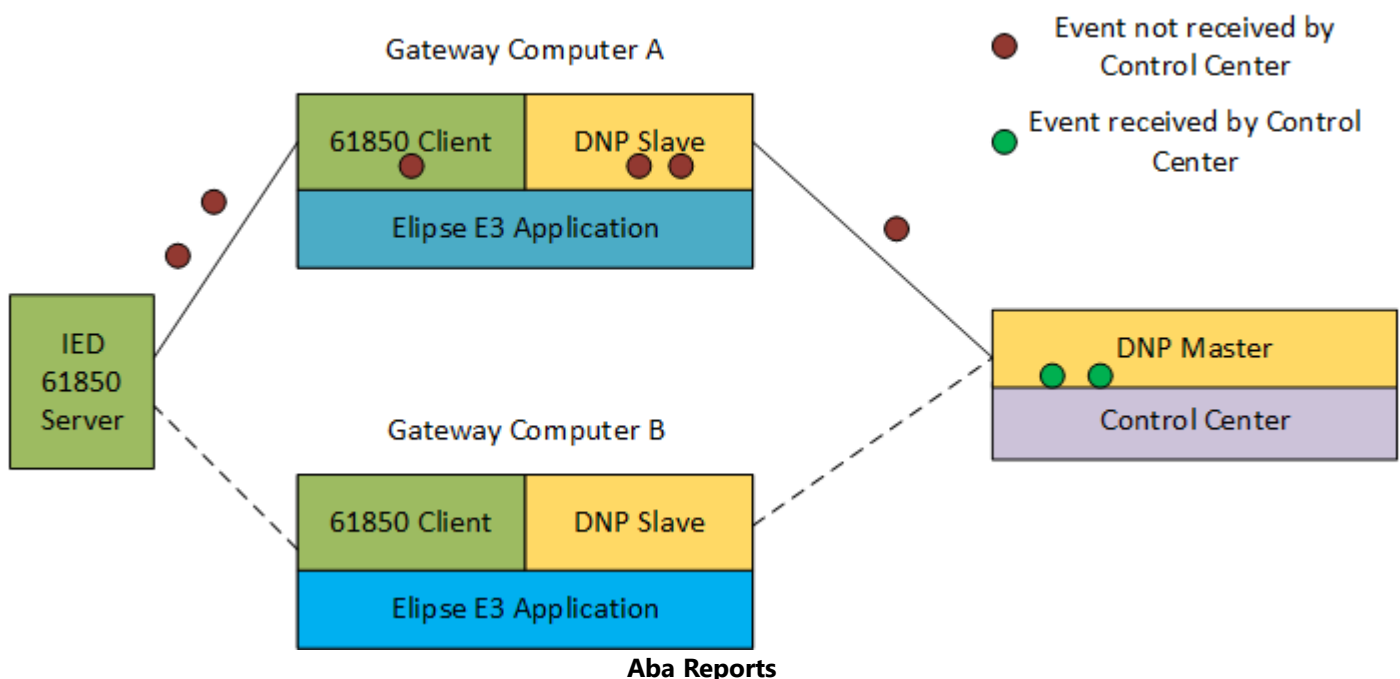
| NAME | DESCRIPTION | ITEM | DEVICE |
|----------------------------|--|-------------------------|----------------------------|
| LANA_Active | Indicates if PRP is logically active on LAN A. Inverted networks (for example) do not allow any of the LAN's to become active. | ElipseClientInfo | PRPStat_LANA_Active |
| LANB_Active | Indicates if PRP is logically active on LAN B. Inverted networks (for example) do not allow any of the LAN's to become active. | ElipseClientInfo | PRPStat_LANB_Active |
| LANA_Total Sent | Indicates the total number of packets sent on LAN A. | ElipseClientInfo | PRPStat_LANA_TotalSent |
| LANB_TotalSent | Indicates the total number of packets sent on LAN B. | ElipseClientInfo | PRPStat_LANB_TotalSent |
| LANA_Total Received | Indicates the total number of packets received on LAN A. | ElipseClientInfo | PRPStat_LANA_TotalReceived |
| LANB_Total Received | Indicates the total number of packets received on LAN B. | ElipseClientInfo | PRPStat_LANB_TotalReceived |

Define a reasonable scan interval for these tags, in order to avoid driver log pollution. As suggestion, the scan can be between 5000 and 10000 ms (5 and 10 seconds).

Gateway Configuration

In gateway applications, where it's necessary to send data from this 61850 driver to other protocols (like DNP 3.0 or IEC 104), it is possible to find situations of event loss or duplication, especially at application startup or gateway switch-over (hot standby).

This may happen when an event is received by 61850 driver and forwarded, but it not received yet by a remote master in another protocol. Example:

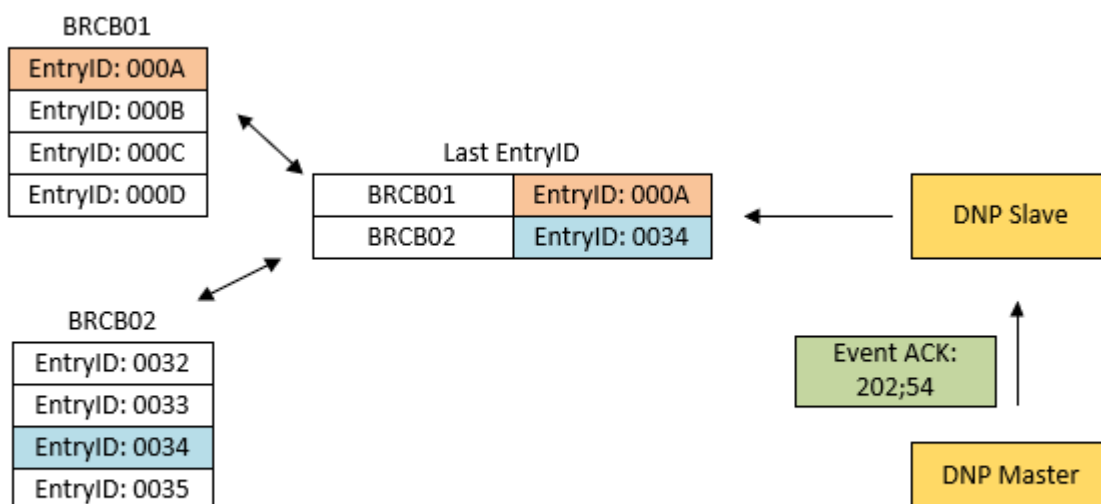


The image above show events being transferred and pending of transmission in several gateway components (61850 Server, 61850 Client, DNP Slave and DNP Master). This event queue size can change depending on the protocol driver configuration and network architecture, between other factors.

When a switch-over from Gateway A to Gateway B happens, for example, the pending events can be lost or duplicated, depending on the application configuration. This behavior can be controlled by programming the EntryID field of the 61850 buffered reports. The EntryID is a unique identifier for each report message sent.

This driver already has a feature to generate files with the last BRCB EntryID, which is used to enable the reports at driver startup. However, these EntryID's are updated based on the moment the report messages are received by 61850 Client driver, and not by the real reception of the event in an external system, i.e. in the example above the Control Center.

When enabling the Gateway feature by the option "Wait for Gtw Ref on Startup", the driver will wait for E3 application to inform a string text which represents each 61850 tag inside another driver (ex: DNP or IEC 104 Slave). When this other driver receives the confirmation the event was delivered to Control Center, it can inform back 61850 driver about this event, so it's possible to position EntryID values forward and save the EntryID File. If a new swtichover happens, the driver will enable the reports starting from this point - avoiding the event loss or repetition.



EntryID Management

The steps to configure this resource are the following:

1 - Enable the options "**Check BRCB Entry ID**", "**R/W Entry ID File (.EID)**", "**Wait for Gtw Ref on Startup**". Further, if the IED uses the same reports for both gateways, you should evaluate using a fixed report list - "**User Defined Report List**", avoiding the redundant application to enable unexpected reports.

2 - At 61850 driver, create the tag "FinishedGtwRef", and the blocks "GatewayRef" and "GatewayEventOK" as below. These 3 tags are described at the topic **Referência de Tags - Tags Internos**.

| | | | | | | | | | | | | |
|----------------|----------|----------------|---|---|---|---|---|------|--------------------------|--------------------------|--|--|
| IEC61850 | | | 0 | 0 | 0 | 0 | | | | | | |
| FinishedGtwRef | | FinishedGtwRef | 0 | 0 | 0 | 0 | | 1000 | <input type="checkbox"/> | <input type="checkbox"/> | | |
| GatewayRef | | GatewayRef | 0 | 0 | 0 | 0 | 4 | 1000 | <input type="checkbox"/> | <input type="checkbox"/> | | |
| IEDName | | | | | | | 0 | | | | | |
| LDName | | | | | | | 1 | | | | | |
| Object | | | | | | | 2 | | | | | |
| Ref | | | | | | | 3 | | | | | |
| GatewayEventOK | IED00001 | GatewayEventOK | 0 | 0 | 0 | 0 | 4 | 1000 | <input type="checkbox"/> | <input type="checkbox"/> | | |
| Ref | | | | | | | 0 | | | | | |
| Value | | | | | | | 1 | | | | | |
| TSValid | | | | | | | 2 | | | | | |
| Time | | | | | | | 3 | | | | | |

3 - Taking DNP Slave driver as example, a block with N2=-22 shall be created. It receives at the OnRead script a list with master confirmed events. Please consult DNP Slave driver documentation for more details.

| | | | | | | | | | | | | |
|-----------|--|--|----|-----|------|---|---|------|-------------------------------------|-------------------------------------|--|--|
| DNP Slave | | | 0 | 0 | 0 | 0 | | | | | | |
| DI_0 | | | 21 | 1 | 402 | 0 | | 1000 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| DI_1 | | | 21 | 1 | 402 | 1 | | 1000 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| DI_2 | | | 21 | 1 | 402 | 2 | | 1000 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| AI_0 | | | 12 | 1 | 3205 | 0 | | 1000 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| AI_1 | | | 12 | 1 | 3205 | 1 | | 1000 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| AI_2 | | | 12 | 1 | 3205 | 2 | | 1000 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | | |
| EventOK | | | 1 | -22 | 0 | 0 | 6 | 1000 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | | |
| ObjVar | | | | | | | 0 | | | | | |
| Index | | | | | | | 1 | | | | | |
| Class | | | | | | | 2 | | | | | |
| Value | | | | | | | 3 | | | | | |
| TSValid | | | | | | | 4 | | | | | |
| Time | | | | | | | 5 | | | | | |

4 - DNP Slave tags shall receive links from 61850 tags. At the image below we have an example where tag AI_0 (DNP object 32 variation 05 index 0) is receiving a link from a phase A current from 61850 IED.

The screenshot displays the configuration interface for the DNP Slave driver. A table lists various tags, including DI_0, DI_1, DI_2, AI_0, AI_1, AI_2, and EventOK, with their respective parameters. A detailed view of the 'AI_0' tag properties is shown, including its connection and source.

| Item | Área de Alarmes | Associações |
|--------------|-----------------|--|
| Propriedades | Conexão | Fonte |
| Value | ← | IEC61850.I3pMMXU1.MX.A.phsA.cVal.mag.f |
| AdviseType | | |
| AllowRead | | |
| AllowWrite | | |
| Bit00 | | |
| Bit01 | | |
| Bit02 | | |
| Bit03 | | |
| Bit04 | | |
| Bit05 | | |
| Bit06 | | |
| Bit07 | | |

5 - Upon startup, the application shall initialize the external references, passing a string text which represents the DNP Address for each 61850 tag which needs the EntryID management. This write operation, however, can only be performed from the moment the driver has the Logical Devices created, otherwise the write will fail. To solve this problem, the example below shows an internal tag script which tries to send the configuration every 5 seconds. If the operation is successful, the event is not fired anymore. At the process end, the tag "**FinishedGtwRef**" is written.

Also in this example, the reference string is obtained through the union of DNP object/variation and the index. At the DNP Slave driver these values are available at N3 and N4 parameters.

Initial Value of the tag InitGateway: TRUE (Boolean)

Event WhileNotOK: Expression: Value - Repeat event every 5000 ms

```

Sub InitGateway_WhileNotOK()
    Set DNPSlave = Application.GetObject("DNPSlave")
    Set Ref = Application.GetObject("IEC61850").Item("GatewayRef")
    bFirst = FALSE
    for each tag in DNPSlave
        if TypeName(tag) = "IOTag" Then
            if tag.N2 = 1 Then
                Source = tag.Links.Item("Value").Source
                Set s = Application.GetObject(Source)
                arr = Split(s.ParamDevice,":")
                Ref.Item("IEDName").Value = arr(0)
                Ref.Item("LDName").Value = arr(1)
                Ref.Item("Object").Value = s.ParamItem
                Ref.Item("Ref").Value = CStr(tag.N3) + ";" + CStr(tag.N4)
                if NOT bFirst Then
                    bFirst = Ref.Write(1) 'sync
                    if NOT bFirst Then Exit Sub
                Else
                    Ref.Write 2 'async unconfirmed
                End If
            End If
        End If
    Next
    res = Application.GetObject("IEC61850").Item("FinishedGtwRef").WriteEx(2) 'async unconfirmed
    if res Then Value = FALSE
End Sub

```

6 - When **"EventOK"** Block is read at DNP Slave driver, it means a new event was received and confirmed by DNP Master. The script below gets the event information and forwards to 61850 driver. At the DNP Slave specific case, this block returns object and variation, index, class, value, valid timestamp and timestamp. The script sends all these parameters to 61850 (except DNP class, which is not relevant) using the block **GatewayEventOK**.

```

Sub EventOK_OnRead()
    Set IECBlock = Application.GetObject("IEC61850.GatewayEventOK")
    IECBlock.Item("Ref").Value = CStr(Item("ObjVar").Value) + ";" + CStr(Item("Index").Value)
    IECBlock.Item("Value").Value = Item("Value").Value
    IECBlock.Item("TSValid").Value = Item("TSValid").Value
    IECBlock.Item("Time").Value = Item("Time").Value
    IECBlock.Write 2 'async
End Sub

```

7 - At this moment the driver will position the EntryIDs at the next pending event. To synchronize the EntryIDs with the redundant computer, it is necessary to follow the steps indicated at the section **Redundancy - Using EID Files**.

8 - To position the EntryID's, the driver uses the following algorithm:

- When a new 61850 report is received, it checks if any dataset object with a gateway reference has changed quality or value. If positive, a new event is added to an internal individual list per 61850 object.

- When an event is confirmed by external driver and informed at the block "GatewayEventOK", the internal list is compared with the informed event:

- If timestamp is not valid (DNP/IEC104 Slave tag is not using timestamp), all events are removed from the list;

- If timestamp is valid, all events are removed if they are equal or happened before the informed event.

In order to avoid unnecessary processing, it is recommended to only add gateway references to tags that can't have event loss or duplication, typically digital and simple/double state tags. It is also recommended in this case, to separate report Datasets by tag types. By configuring digital and analog tags in different datasets, you can avoid receiving useless old analog values if the digital reports have to roll back in time.

Dynamic Configuration

This Driver allows some configuration parameters to be informed at run time by script, by using a command for writing parameters from the **IOKit** library. These parameters are described on the next table.

Configuration parameters

| PROPERTY | PARAMETER | DATA TYPE |
|---|--|-----------|
| Total Device Number | IEC61850.DeviceCount | DWORD |
| Server | IEC61850.Device[%u].Name | STRING |
| IP | IEC61850.Device[%u].IP | STRING |
| Backup IP | IEC61850.Device[%u].IPBackup | STRING |
| PSEL | IEC61850.Device[%u].PSel | DWORD |
| SSEL | IEC61850.Device[%u].SSel | DWORD |
| TSEL | IEC61850.Device[%u].TSel | DWORD |
| Rem AE Qual | IEC61850.Device[%u].AEQualifier | DWORD |
| Rem AP ID | IEC61850.Device[%u].AppID | STRING |
| Disable | IEC61850.Device[%u].Disable | BYTE |
| Use Backup IP | IEC61850.Device[%u].UseIPBackup | BYTE |
| Save Comtrade (Per Device) | IEC61850.Device[%u].SaveComtrade | BYTE |
| Delete Comtrade Files (Per Device) | IEC61850.Device[%u].DeleteComtrade | BYTE |
| Browse Root Folder Only (Per Device) | IEC61850.Device[%u].BrowseRootFolderOnly | BYTE |
| Comtrade Path (Per Device) | IEC61850.Device[%u].ComtradePath | STRING |
| Use Fixed Comtrade Path On Device (Per Device) | IEC61850.Device[%u].UseFixedPathOnDevice | BYTE |
| Name+Index File Transfer | IEC61850.Device[%u].NameAndIndexFileTransfer | BYTE |
| File Root Name | IEC61850.Device[%u].FileTransferRootName | STRING |
| Prefer Buffered Report Control Blocks | IEC61850.UseReports | BYTE |
| User-Defined Report List | IEC61850.UserReportList | BYTE |
| Poll Tags not found in any report | IEC61850.PollTags | BYTE |
| Local P Selector | IEC61850.LocalPSel | DWORD |
| Local S Selector | IEC61850.LocalSSel | DWORD |
| Local T Selector | IEC61850.LocalTSel | DWORD |
| Local AE Qualifier | IEC61850.LocalAEQualifier | DWORD |
| Local App ID | IEC61850.LocalAppID | STRING |
| RFC 1006 Source TSAP | IEC61850.SourceTSAP | DWORD |
| App Category | IEC61850.AppCategory | STRING |
| Polled Intg Rpt | IEC61850.RptGI | DWORD |

| PROPERTY | PARAMETER | DATA TYPE |
|---|--|-----------|
| Auto Intg Prt | IEC61850.IntgPd | DWORD |
| Use Quality change trigger | IEC61850.UseQChgTrgOps | |
| Conformance Blocks (internal use) | IEC61850.CBB | STRING |
| Services (internal use) | IEC61850.Services | STRING |
| LD File Path | IEC61850.LDPath | STRING |
| Comtrade Path | IEC61850.ComtradePath | STRING |
| Comtrade Directory on Device (internal use) | IEC61850.ComtradeDir | STRING |
| Msg Timeout | IEC61850.Timeout | DWORD |
| Full Log Details | IEC61850.DetailedLog | BYTE |
| Save Comtrade Files | IEC61850.SaveComtrade | BYTE |
| Save Comtrade Per IED | IEC61850.SaveComtradePerIED | BYTE |
| Delete Files after upload | IEC61850.DeleteComtrade | BYTE |
| Comtrade Directory Check Interval | IEC61850.CheckComtrade | BYTE |
| Comtrade Upload Interval | IEC61850.UploadComtrade | DWORD |
| Check Report Time of Entry | IEC61850.RedundantBRCB | BYTE |
| Status Check | IEC61850.StatusCheck | DWORD |
| Asynchronous Write | IEC61850.AsyncWrite | BYTE |
| Use Single Tag Command Alias | IEC61850.SingleTagCmdAlias | BYTE |
| Apply Local Time Offset to Timestamps | IEC61850.ApplyLocalTime | BYTE |
| Use Cmd Tag N1 as check condition | IEC61850.SingleTagCmdN1 | BYTE |
| No LD Database Scan | IEC61850.NoLDScan | BYTE |
| Browse Root Folder Only | IEC61850.ComtradeBrowseRootFolder Only | BYTE |
| Use Exclusive URCB | IEC61850.ReserveURCB | BYTE |

Below there is an example of COMTRADE dynamic configuration:

1 – The driver shall have all IED's already declared, and shall be set with the option **Disable** checked (value = 1). This is to avoid the communication to start with incorrect parameters.

2 – The driver shall also have the option "Define Parameters Per IED" checked.

3 – At AfterStart event, it is necessary to have a script that changes the device parameters, using an IOKIT special tag. The parameter names are informed below, where %u shall be changed by the IED index at the driver, starting with 0. These parameters and index can also be seen at a driver log, if they are not 0.

```
"IEC61850.Device[%u].SaveComtrade"
"IEC61850.Device[%u].DeleteComtrade"
"IEC61850.Device[%u].BrowseRootFolderOnly"
"IEC61850.Device[%u].ComtradePath"
"IEC61850.Device[%u].UseFixedPathOnDevice"
"IEC61850.Device[%u].NameAndIndexFileTransfer"
```

"IEC61850.Device[%u].FileTransferRootName"

4 – Consider that the parameters ComtradePath e FileTransferRootName are text strings, and if their individual value (per IED) is not set, the default value set at the Dialog will be considered. The default value for other properties is 0 (FALSE).

5 – After writing the desired parameters, you should enable the IED by writing 0 at the parameter "IEC61850.Device[%u].Disable".

6 – Final step, ask the driver to start communication by writing 1 at the special tag "ServerInitialBuild". This tag starts the device communication that are enabled and which haven't been created yet.

Script Example:

' Example considering an 8-element Array. The empty indexes are ignored

Dim Arr(8)

Arr(1) = Array("IEC61850.Device[0].SaveComtrade",1) ' This IED will transfer Comtrades

Arr(2) = Array("IEC61850.Device[1].SaveComtrade",0) ' This IED will NOT transfer Comtrades

Arr(3) = Array("IEC61850.Device[2].SaveComtrade",1) ' This IED will transfer Comtrades

Arr(4) = Array("IEC61850.Device[2].NameAndIndexFileTransfer",1) ' This IED will transfer Comtrades using the RootName + Index method

Arr(5) = Array("IEC61850.Device[0].Disable",0) ' Enables IED

Arr(6) = Array("IEC61850.Device[1].Disable",0) ' Enables IED

Arr(7) = Array("IEC61850.Device[2].Disable",0) ' Enables IED

Driver.Write -1,0,0,3 Arr ' Write parametros

Driver.Item("ServerInitialBuild").WriteEx 1 'Considering a tag called ServerInitialBuild with the Item parameter = ServerInitialBuild, upon writing 1 it will start communication with all enabled IED's

7 – When stopping a driver, these parameters are not saved, so the AfterStart script shall be kept, so at every driver startup, the IED's will be correctly configured.

Limitations of this Driver

The IEC 61850 Driver is provided with the following types of licenses by **Elipse Software**:

- **IEC61850.dll**: Allows communication with up to 25 relays (IEDs), to ensure communication performance. For more relays, use a new instance of this Driver
- **Power_IEC61850.dll**: Allows communication with a single relay (IED), using a standard license that allows replacement by another power system Driver. For a full list of Drivers sharing this license, please check with **Elipse Software**


Documentation of I/O Interfaces

This section contains the documentation of I/O Interfaces referring to the **IEC61850** Driver.

Driver Configuration

I/O Interface configuration is performed on a Driver's configuration dialog box. To access the configuration of this dialog box in **E3** (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 **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 Driver for each serial port.

Configuration Dialog Box

The I/O Interfaces dialog box 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 each Driver, on the configuration dialog box.

Setup Tab

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

- **General configurations:** Configurations of 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 |

| OPTION | DESCRIPTION |
|-----------------------------|--|
| 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 reception's buffer |
| Start driver OFFLINE | Select this option so that the 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. Please check topic Driver Statuses for more details |
| 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, the 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, the Driver goes to the Offline mode, assuming that a hardware problem was detected. If a Driver establishes a successful connection, the number of unsuccessful retries is cleared. If this new connection is lost, then the retry counter starts at zero |
| Disconnect if non-responsive for ... seconds | Enable this option to force a Driver to disconnect if no byte was received by the I/O interface during the specified time-out, in seconds. This time-out must be greater than the time-out configured in the Timeout option |

Options on the Logging Options group

| OPTION | DESCRIPTION |
|-------------|--|
| Log to File | <p>Enable this option and configure the name of a file to write a log. Log files can be large, so use this option for short periods of time, only for testing and debugging purposes.</p> <p>If the %PROCESS% macro is used in the log file name, it is replaced by the ID of the current process. This option is particularly useful when using several instances of the same Driver in E3, thus allowing each instance to generate a separate log file. For example, when configuring this option as c:\e3logs\drivers\sim_%PROCESS%.log, a file named c:\e3logs\drivers\sim_00000FDA.log is generated for process 0FDAh.</p> <p>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 as c:\e3logs\drivers\sim_%DATE%.log, a file named c:\e3logs\drivers\sim_2005_12_31.log is generated in 12/31/2005 and a file named c:\e3logs\drivers\sim_2006_01_01.log is generated in 01/01/2006</p> |

Ethernet Tab

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

Ethernet

Transport: TCP/IP

☐ PING before connecting

Timeout: 4000 ms

Retries: 1

☐ Listen for connections on port: 0

☐ Share listen port with other processes

☐ Interface: (All Interfaces)

☐ Use IPv6
☐ Use SSL

SSL Settings

☐ Enable 'ECHO' supression

IP Filter:

Connect to

Main IP:

Port: 502

☐ Local port: 0

☐ Backup IP 1:

Port: 0

☐ Local port: 0

☐ Backup IP 2:

Port: 0

☐ Local port: 0

☐ Backup IP 3:

Port: 0

☐ Local port: 0

Ethernet tab

Available options on Ethernet tab

| OPTION | DESCRIPTION |
|---|---|
| Transport | Select TCP/IP for a TCP socket (stream). Select 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, the 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 listen port with other Drivers and processes |
| Interface | Select the local network interface, identified by its IP address, that is used by the Driver to establish and receive connections, or select the (All Interfaces) item to use any local network interface |
| Use IPv6 | Check this option to force the Driver to use IPv6 addresses on all Ethernet connections. If this option is unchecked the Driver will work with IPv4 addresses |
| 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 details |
| PING before connecting | <p>Enable this option to execute a ping command, that is, check whether a device can be reached on a network, for a device before trying a socket connection. This is a quick way of determining a successful connection before trying to open a socket with a device. The time-out of a connection with a socket can be very high. The available options are:</p> <ul style="list-style-type: none"> • Timeout: Specify the number of milliseconds to wait for a reply from the ping command. Users must use the 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 one and 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 the remote device. Users can use an IP address separated by dots, as well as a URL. In case of a URL, the 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 the remote device, between 0 (zero) and 65535 |

| OPTION | DESCRIPTION |
|------------------------------|---|
| 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 here the IP address, the IP port, and the fixed local IP port of up to 3 (three) backup addresses of a remote device |

General Configurations

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

I/O Tags

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

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

IO.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 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:** Event message, 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. Its possible values are the following:

- **0:** Physical layer stopped, that is, the 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, the Driver is in **Online** mode but the physical layer is not connected. If the **Connection management** option is configured as **Automatic**, the physical layer can be connecting, disconnecting, or waiting for a reconnection attempt. If the **Connection management** option is configured as **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 the device is connected, only 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 the 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 three parameters, then the size of the Block must be 6 (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 writing 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 **E3**, the ability to create arrays at run time allows using an I/O Tag as well as a Block Tag. Users can use Driver's **Write** method to send all parameters to the 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 Driver's log or use the *writeStatus* parameter of the **WriteEx** method to find out the exact cause of the 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 on the **IO.PhysicalLayerStatus** Tag

In the next example, using **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 can fail when configuring a Driver to **Online** mode, that is, writing the value 1 (one). In this case, the 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
- 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 failure depends on the type of physical layer. It can be an invalid serial port number, a failure when starting Windows Sockets, or a failure when starting TAPI (modem), among others. This cause is recorded on the log file

IMPORTANT

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

Properties

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

IO.ConnectionMode

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

IO.GiveUpEnable

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

IO.GiveUpTries

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

IO.InactivityEnable

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

IO.InactivityPeriodSec

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

IO.RecoverEnable

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

IO.RecoverPeriodSec

9 Delay time between two connection attempts, in seconds.

NOTE

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

IO.StartOffline

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

NOTE

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

IO.TimeoutMs

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

IO.Type

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

- **N or None:** Does not use a physical interface, that is, a Driver must provide a customized interface
- **S or Serial:** Uses a local serial port (COM n)
- **M or Modem:** Uses a local modem, internal or external, accessed via TAPI (*Telephony Application Programming Interface*)
- **E or Ethernet:** Uses a TCP/IP or UDP/IP socket

- **R or RAS:** Uses a **RAS** (*Remote Access Server*) Interface. A Driver connects to a RAS device using the **Ethernet** Interface and then sends an **AT** (*dial*) command

Statistical Configuration

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

I/O Tags

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

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

IO.Stats.Partial.BytesRecv

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

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

IO.Stats.Partial.BytesSent

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

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

IO.Stats.Partial.TimeConnectedSeconds

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

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

IO.Stats.Partial.TimeDisconnectedSeconds

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

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

IO.Stats.Total.BytesRecv

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

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

IO.Stats.Total.BytesSent

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

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

IO.Stats.Total.ConnectionCount

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

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

IO.Stats.Total.TimeConnectedSeconds

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

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

IO.Stats.Total.TimeDisconnectedSeconds

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

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

Properties

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

Ethernet Interface Configuration

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

I/O Tags

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

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

IMPORTANT

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

IO.Ethernet.IPSelect

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

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

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

During the connection process, if the active IP address is not available, the I/O Interface tries to connect using the next alternative 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 IP address or **1, 2, 3**: Alternative IP address) if the Driver is currently connected. If the Driver is disconnected, this Tag configures the active IP address for the next connection attempt.

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 Driver tries to connect to each one of the alternative or backup IP addresses and back to the main IP address until a connection is established.

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

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 the Driver must not accept external connections, that is the Driver behaves as a master, or configure to True to enable the reception of connections, that is, the 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, the Driver tries to use an alternative IP address. Configure to False to disable its usage.

IO.Ethernet.BackupLocalPort[2,3]

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

IO.Ethernet.BackupLocalPortEnable[2,3]

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

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.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. 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 from 192.168.0.41 to 192.168.0.50
- **192.168.0.***: Accepts connections from IPv4 addresses from 192.168.0.0 to 192.168.0.255
- **fe80:3bf:877::** (expands to fe80:03bf:0877:0000:0000:0000:0000:0000 to fe80:03bf:0877:0000:0000:0000:ffff:ffff)**: Accepts connections from IPv6 addresses from fe80:03bf:0877:0000:0000:0000:0000:0000 to 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 from 192.168.0.0 to 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 listen mode, where a Driver can receive packets from different IP addresses, blocks or permissions are performed at each packet received. If a packet is received from a blocked IP address, it logs a message "Blocked incoming packet from {IP} (discarding {N} bytes)!".


IO.Ethernet.ListenIP

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


IO.Ethernet.ListenPort

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


IO.Ethernet.MainIP

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


IO.Ethernet.MainLocalPort

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


IO.Ethernet.MainLocalPortEnable

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


IO.Ethernet.MainPort

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

IO.Ethernet.PingEnable

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


IO.Ethernet.PingTimeoutMs

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


IO.Ethernet.PingTries

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

IO.Ethernet.ShareListenPort

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


IO.Ethernet.SuppressEcho

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

IO.Ethernet.Transport

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

IO.Ethernet.UseIPv6

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

Revision History

| VERSÃO | DATA | AUTOR | COMMENTS |
|--------|------------|-------------|---|
| 3.0.44 | 21/03/2024 | M. Salvador | <ul style="list-style-type: none"> Adicionado a suporte a gateway Arquivos de horario do último comtrade |
| 3.0.9 | 11/12/2019 | M. Salvador | <ul style="list-style-type: none"> PRP Support |
| 3.0.6 | 01/09/2019 | M. Ludwig | <ul style="list-style-type: none"> Driver portado para o Visual Studio 2017 (<i>Case 27095</i>). |
| 3.0.1 | 04/06/2019 | M. Salvador | <ul style="list-style-type: none"> Implementada a Certificação IEC 61850 ED2. |
| 2.0.23 | 08/02/2019 | M. Salvador | <ul style="list-style-type: none"> Implementada uma opção de espera na escrita do Tag LastComtradeFileTime, configurável na aba Files (<i>Case 25806</i>). |
| 2.0.20 | 19/11/2018 | M. Salvador | <ul style="list-style-type: none"> Corrigido um vazamento de memória quando o servidor não implementa ou não responde um pedido de GetVariableAccessAttributes (<i>Case 25546</i>). |
| 2.0.19 | 17/09/2018 | M. Salvador | <ul style="list-style-type: none"> Agora um IED que não suporta a utilização de DataSets dinâmicos pode ser usado junto com outros IEDs que suportam essa operação no mesmo Driver (<i>Case 25072</i>). |
| 2.0.16 | 16/11/2017 | M. Salvador | <ul style="list-style-type: none"> Criar um novo Driver define como padrão o uso de um Tag para cada comando na opção Single Tag Cmd Alias (<i>Case 23693</i>). Realizadas correções ao utilizar DataSets dinâmicos, incluindo o tratamento de reconexões e reinicialização dos IEDs (<i>Case 23644</i>). |
| 2.0.3 | 11/02/2014 | M. Salvador | <ul style="list-style-type: none"> Implementada a Certificação KEMA. |

| VERSÃO | DATA | AUTOR | COMMENTS |
|--------|------------|-------------|---|
| 1.1.27 | 03/12/2012 | M. Salvador | <ul style="list-style-type: none"> Correções no processo de conexão e desconexão. Corrigido o tratamento de Reports com BitInclusion incoerente com o tamanho do Dataset. Correções no processo de conexão e desconexão, criando uma nova <i>thread</i> de verificação do estado das conexões. Modificado o tratamento de exceções de leituras. Mensagem de Reject estava sendo tratada como ConfirmedError em alguns casos. Correção de erros de tratamento de listas. Tratamento de números reais inválidos (QNaN). Correções na verificação de status da conexão. Corrigida a informação de status de execução de um comando. Opções adicionais para busca de arquivos COMTRADE. Correção de um possível <i>deadlock</i> entre semáforos na reconexão. Bloco de status de COMTRADE. Suporte a EntryID na inicialização dos Reports. Criação dinâmica de <i>devices</i>. Propagação de eventos de IOKit. Correção na coleta de oscilografia para relés Areva e Alstom. Suporte a tipos de dados Double em mms_float. DNSNames com especificação de porta no endereço IP do IED. Correção na importação ICD SEL, proteção adicional para evitar escritas e polling |

| VERSÃO | DATA | AUTOR | COMMENTS |
|--------|------|-------|--|
| | | | <p>quando estiver em reconexão.</p> <ul style="list-style-type: none"> • Suporte a <i>user session requirements</i> no Presentation CPA PPDU. • Correção de vazamento de memória (VLD). • Correção de vazamento de memória quando desconectado (COTP) e teste de semáforo SEL. • Log do fuso horário ao iniciar o Driver. • Considera <i>TimeZone Unknown</i> como válido para o cálculo do <i>offset</i> local. • Mudança no modo de reabilitação dos Reports depois da reconexão. • Não é mais permitido criar LDs, a não ser na leitura de arquivos LD ou após a execução de um comando GetServerDirectory (Case 11378). • Eliminados os zeros à esquerda dos endereços IP (Case 11379). • Eliminar zeros à esquerda não considerava o uso de porta depois do endereço IP. • Reprogramados os Reports completamente após uma reconexão. • Criados os Tags IPSelect e IPSwitch e removida a propriedade TimeofEntry do <i>polling</i>, se habilitada. • Implementada uma <i>thread</i> de verificação de status dos <i>hosts</i>. • Limite de 20 Tags para leitura por <i>scan</i> e de atualização de RptEna quando há Tags no limbo. • Problema de recepção errada aleatória de pacotes nas camadas <i>Session</i> e <i>Presentation</i>. |

| VERSÃO | DATA | AUTOR | COMMENTS |
|--------|------------|-------------------------|--|
| 1.0.1 | 31/03/2010 | M. Salvador M. Bihre | <ul style="list-style-type: none"> • Modificado o parâmetro <i>Nesting Level Requested</i> de cinco para seis (INGETEAM). • O comando GetVariableAccessAttributes é realizado um diretório abaixo caso falhe (INGETEAM). EntryID e OptFlds agora não são solicitados nos OptFlds quando o Report é do tipo Unbuffered. • Corrigida uma potencial falha dentro de MMS::SendRequest, que pode ocorrer quando há queda de conexão. • Melhoria de performance dos Tags de status. • Pedido de desconexão TCP/IP não estava sendo realizado quando o pedido de MMS::Status falhava ou não cobria todas as situações. • Modificada a forma de realizar uma operação de ServerRebuild, que agora é assíncrona. • Melhorada a performance de inicialização do sistema, principalmente na leitura dos arquivos LD. • Corrigido o envio de números inteiros para comandos na propriedade ctlNum (<i>Case 11097</i>) • O parâmetro <i>WriteStatus</i> das operações de escrita pelo método WriteEx do E3 ou Elipse Power não era informado em alguns casos (<i>Case 11104</i>). • Adicionado suporte a <i>download</i> de oscilografias. • Adicionado suporte a comandos assíncronos. • Adicionado suporte a Tags individuais para comandos. • Corrigido suporte a arquivos ICD e SCD. • Versão original deste Driver. |

Headquarters

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

90510-002 — Porto Alegre — RS

Phone: (+55 51) 3346-4699

Fax: (+55 51) 3222-6226

E-mail: elipse-rs@elipse.com.br

Branch in Taiwan

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

807 — Kaohsiung City — Taiwan

Phone: (+886 7) 323-8468

Fax: (+886 7) 323-9656

E-mail: evan@elipse.com.br

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

www.elipse.com.br

kb.elipse.com.br

forum.elipse.com.br

www.youtube.com/elipsesoftware

elipse@elipse.com.br



Gartner, Cool Vendors in Brazil 2014, April 2014.

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

Microsoft Partner

Gold Independent Software Vendor (ISV)