

Elipse SendMail Driver

| | |
|----------------------------|----------------------------|
| File Name | SendMail.dll |
| Manufacturer | Elipse Software |
| Devices | N/A |
| Protocol | SMTP |
| Version | 2.0.17 |
| Last Update | 12/03/2025 |
| Platform | Win32 |
| Dependencies | IOKit version 2.0 or later |
| Superblock Readings | No |
| Level | 0 |

Introduction

The Elipse SendMail Driver allows communication with e-mail servers, creating and sending e-mail messages.

Legal Notice

This Driver includes software developed by the OpenSSL Project for use by OpenSSL Toolkit, in accordance with the license described next.

LICENSE ISSUES

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License

Copyright (c) 1998-2008 The OpenSSL Project.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)".
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)".

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (ey@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (ey@cryptsoft.com)

All rights reserved.

This package is an SSL implementation written by Eric Young (ey@cryptsoft.com). The implementation was written so as to conform with Netscape's SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes cryptographic software written by Eric Young (ey@cryptsoft.com)" The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-)
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgment: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)".

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED

AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license [including the GNU Public License].

Driver Configuration

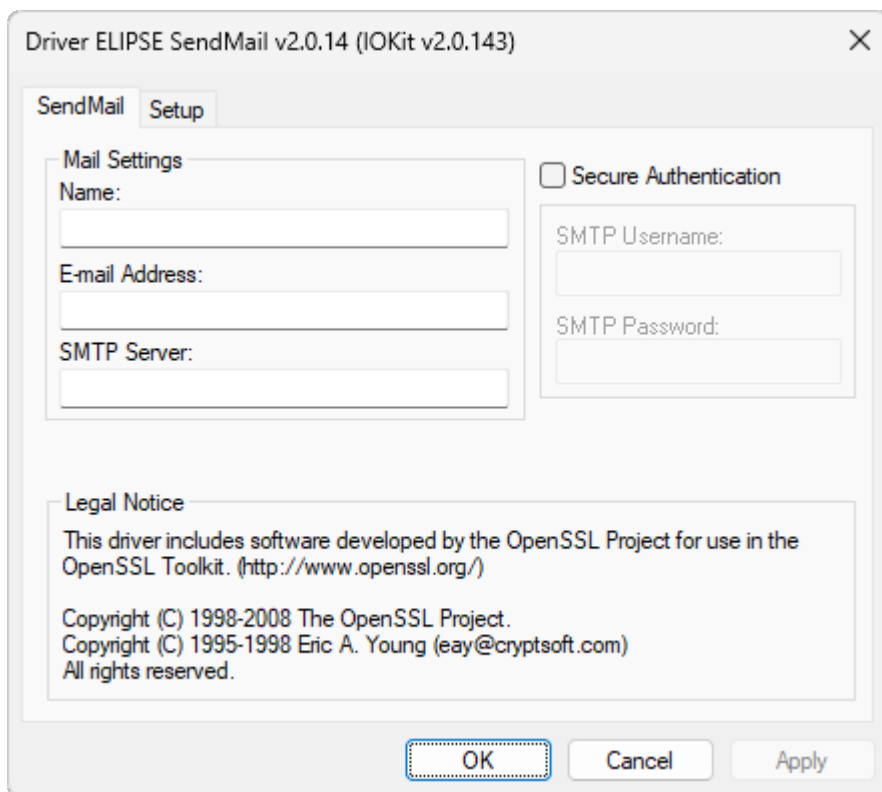
This section contains information about the configuration of this Driver's [P] parameters and their properties.

[P] Parameters for Driver Configuration

| | |
|-----------|---|
| P1 | TCP/IP port of the e-mail server, usually port 25 |
| P2 | Not used |
| P3 | Not used |
| P4 | Not used |

Configuration of Driver Properties

The **SendMail** tab contains specific configurations for this Driver.



SendMail tab

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

Available options on SendMail tab

| OPTION | DESCRIPTION |
|-----------------------|----------------------------|
| Name | Name of an e-mail's sender |
| E-mail Address | E-mail address of a sender |

| OPTION | DESCRIPTION |
|------------------------------|--|
| SMTP Server | Address of the SMTP (<i>Simple Mail Transfer Protocol</i>) server of sender's e-mail |
| Secure Authentication | Select this option to enable a secure configuration of SMTP Username and SMTP Password options, without the possibility of changing these options via Tag and without displaying the password in the log |
| SMTP Username | Name of the user used for authentication in an SMTP server |
| SMTP Password | Password of a user or application used to authenticate in an SMTP server |

NOTE

These options can also be defined by their respective Tags in an application, except for **SMTP Username** and **SMTP Password** options, that is, the *N1* parameter equal to 5 (five) or 6 (six), respectively, when the **Secure Authentication** option is selected.

Tag Reference

This section contains information about the configuration of this Driver's **[N]** parameters.

[N] Parameters for PLC Tag Addressing

Reading or Writing

| | |
|-----------|--|
| N1 | Function for this Driver. For more information, please check the next table |
| N2 | Sub-function for this Driver. For more information, please check the next table |
| N3 | Additional parameter, if needed |
| N4 | Not used |

The Tags containing all supported operations are available to users via Tag Browser. If users want to directly configure these Tags, use the functions described on the next table.

Available functions for this Driver

| N1 | N2 | N3 | MODE | DESCRIPTION |
|----------|----------|----------|---------------------|---|
| 0 | 0 (zero) | Not used | Reading and Writing | Text for the To (<i>recipient</i>) field. For more than one recipient, separate e-mail addresses with a comma or a semicolon |

| N1 | N2 | N3 | MODE | DESCRIPTION |
|----|----------|----------|---------------------|--|
| | 1 (one) | Not used | Reading and Writing | Text for the CC (<i>carbon copy</i>) field. For more than one recipient, separate e-mail addresses with a comma or a semicolon |
| | 2 | Not used | Reading and Writing | Text for the BCC (<i>blind carbon copy</i>) field. For more than one recipient, separate e-mail addresses with a comma or a semicolon |
| 1 | Not used | Not used | Reading and Writing | Text for the Subject field |
| 2 | Not used | Not used | Reading and Writing | Text for an e-mail's body |
| 3 | 0 (zero) | Not used | Write-Only | Defines a priority level and sends a message. Writing values to define an e-mail priority are 0 : Normal, 1 : High, or 2 : Low. The <i>N2</i> parameter changes the format for sending a message. Possible values are 0 : Text without formatting (<i>Plain Text</i>) or 1 : Text in HTML format. For more information, please check the next table |
| | 1 (one) | Not used | Write-Only | |
| 4 | Not used | Not used | Reading and Writing | When reading, generates a log with configurations of the e-mail server. When writing, specifies a path, full or relative, to a file or to several files separated by a vertical bar to send as attachments of an e-mail. For example, to send an attached file, use AttachTag equal to "file1.txt". To send two attached files, use AttachTag equal to "file1.txt file2.txt" |

| N1 | N2 | N3 | MODE | DESCRIPTION |
|----|----------|----------|---------------------|--|
| 5 | Not used | Not used | Write-Only | Name of the user used for SMTP (<i>Simple Mail Transfer Protocol</i>) authentication. Writing is available when the Secure Authentication option is not selected on this Driver's properties window |
| 6 | Not used | Not used | Write-Only | Password of a user or application used for SMTP authentication. For more information, please check this <i>article</i> about two-factor authentication on Elipse Knowledgebase . Writing is available when the Secure Authentication option is not selected on this Driver's properties window |
| 7 | Not used | Not used | Reading and Writing | Text for the sender's name |
| 8 | Not used | Not used | Reading and Writing | Text for the sender's e-mail address |
| 9 | Not used | Not used | Reading and Writing | Text for the SMTP server's address |
| 10 | Not used | Not used | Reading and Writing | Number of the TCP/IP port of an SMTP server |
| 11 | 0 (zero) | 0 (zero) | Reading and Writing | Name of an external text file, with content saved in ANSI standard, to fill an e-mail's body |
| | 0 (zero) | 1 (one) | Reading and Writing | The first line of the external text file fills e-mail's Subject field |
| | 1 (one) | 0 (zero) | Reading and Writing | Name of an external text file, with content saved in Unicode standard, to fill e-mail's body |
| | 1 (one) | 1 (one) | Reading and Writing | The first line of the external text file fills the Subject field of an e-mail |

| N1 | N2 | N3 | MODE | DESCRIPTION |
|---|----------|----------|-----------|---|
| 90 (only for Elipse E3, Elipse Power, or Elipse Water) | Not used | Not used | Read-Only | Tag exclusive for Elipse E3, Elipse Power, or Elipse Water used to monitor the status of asynchronous writing operations using the Tag's WriteEx method with the <i>WriteSyncMode</i> parameter equal to 2 (two). Returns the values 0 : Initial status, received when triggering an asynchronous writing using the Tag's WriteEx method with the <i>WriteSyncMode</i> parameter equal to 2 (two), 1 : Result of a successful asynchronous writing, or Error Code : Result of a failed asynchronous writing. For more information, please check topic Error Codes for Writing Tags |

NOTES

- For the *N1* parameter equal to **11**, the name of the external text file must be informed in **ANSI** standard, regardless of whether its content is saved in **ANSI** or **Unicode** standard. For content saved in **Unicode** standard, its byte order must be in the **UTF-16 Little Endian** format.
- Sending a message occupies an application for a few seconds. To prevent this, asynchronous writings must be performed using the *WriteSyncMode* parameter of the **WriteEx** method with the value 2 (two) in Tags with the *N1* parameter equal to 3 (three) or configuring the **WriteSyncMode** property of this Driver with the value 2 (two, **wsmAsyncUnconfirmed**).
- When configuring the **To**, **CC**, or **BCC** fields in a Tag with the *N1* parameter equal to 0 (zero), users are allowed to enter e-mail addresses in a simple format, such as "name@domain.com". Alternatively, users can use a full format, including the recipient's name, such as "Full Name <name@domain.com>".

Examples of HTML Tags

| HTML TAG | DESCRIPTION | USAGE | RESULT |
|------------------|-----------------|------------------------|------------------------|
| | Bold text | Text in bold | Text in bold |
| <i> | Italic text | Text in <i>italic</i> | Text in <i>italic</i> |
| <u> | Underlined text | <u>Underlined</u> text | <u>Underlined</u> text |

| HTML TAG | DESCRIPTION | USAGE | RESULT |
|---------------------------------|--|---|--|
| <h1> to <h6> | Defines different levels of headings, in which the level 1 (one) is the most important | <code><h1>Main title</h1></code> | Main title |
| <p> | Define a paragraph of text | <code><p>This is an example paragraph</p></code> | This is an example paragraph |
| | Inserts a line break within a paragraph | <code><p>This is an example text. And it has a line break.</p></code> | This is an example text. It has a line break. |
| <a> | Creates a link to another page or resource | <code>Link</code> | Link |
| | Inserts an image | <code></code> | (The image to be displayed) |
| <hr> | Inserts a horizontal line that separates content | <code><p>Paragraph 1</p></code> <code><hr></code> <code><p>Paragraph 2</p></code> | Paragraph 1 <hr/> Paragraph 2 |

NOTE

To change the color of any text, insert a *style* attribute in the HTML Tag with a color value in **Hexadecimal** or **RGB** (*Red, Green, and Blue*) format, according to the next examples:

- `<p style='color:#9BBB59;'>This is an example paragraph</p>`: **This is an example paragraph**
- `<p style='color:rgb(247, 150, 70);'>This is an example paragraph</p>`: **This is an example paragraph**

Error Codes for Writing Tags

By using scripts in an **Elipse E3**, **Elipse Power**, or **Elipse Water** application, users can use the **WriteEx** method of a Tag to write data and retrieve an error code in case of a failure to write this Tag. Use the reading of a PLC Tag with the *N1* parameter equal to 90 if that writing is asynchronous. In case of a failure on writing, a numeric error code is returned, according to the next table.

Error codes returned by the WriteEx method

| PREFIX | XXX | YYY | DESCRIPTION |
|---|--|------------------------------|--|
| 000xxxx: Error related to an internal Driver failure | 9999: Unexpected internal failure | Not used | An internal failure occurred, not expected by this Driver |
| 1000xxx: Error related to application Tags | 100: Invalid parameter | Not used | <i>N/B</i> parameters of Tags have an invalid value |
| | 200: Data type mismatch | Not used | The value to write is in a format different from the one expected by a Tag |
| | 300: Invalid writing | Not used | Writing to a Tag changes parameters that cannot be modified due to selecting the Secure Authentication option |
| 200xxxx: Error related to file access | 10yy: Opening error | yy: System error code | A list with system error codes can be found on this |

| PREFIX | XXX | YYY | DESCRIPTION |
|---|---|---|--|
| | 20yy : Reading error | | <i>article</i> from Microsoft Learn website |
| | 30yy : Seeking error | | |
| 3xxxxxx : Error related to Ethernet socket | 1yyyyy : Connection error | yyyyy : Ethernet socket error code | A list with Ethernet error codes can be checked on this <i>article</i> from Microsoft Learn website |
| | 2yyyyy : Addressing error | | |
| | 3yyyyy : Sending error | | |
| | 4yyyyy : Receiving error | | |
| | 5yyyyy : No data error | Not used | No data was received during Ethernet communication |
| 4xxxxxx : Error related to OpenSSL API | 1yyyyy : Connection error | yyyyy : OpenSSL API error code | Specific error code from OpenSSL API |
| | 2yyyyy : Sending Error | | |
| | 3yyyyy : Receiving error | | |
| 500xxxx : Error related to SMTP protocol | 1000 : EHLO/HELO error | Not used | The SMTP (<i>Simple Mail Transfer Protocol</i>) server could not be queried |
| | 2000 : Authentication error | Not used | Failure authenticating user and password with the SMTP server |
| | 3000 : Base64 encoding error | Not used | Failure in e-mail's Base64 encoding |
| | 4000 : E-mail transmission error | Not used | Failed to send an e-mail |
| | 5yyy : SMTP protocol return code | yyy : SMTP protocol return code | A list with return codes for the SMTP protocol can be found on this <i>article</i> from turboSMTP website |

Documentation of I/O Interfaces

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

Configuration of a Driver

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

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

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

1. Open the Organizer.
2. Select a Driver on Organizer's tree.

3. Click **Extras** on the **Driver** tab.

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

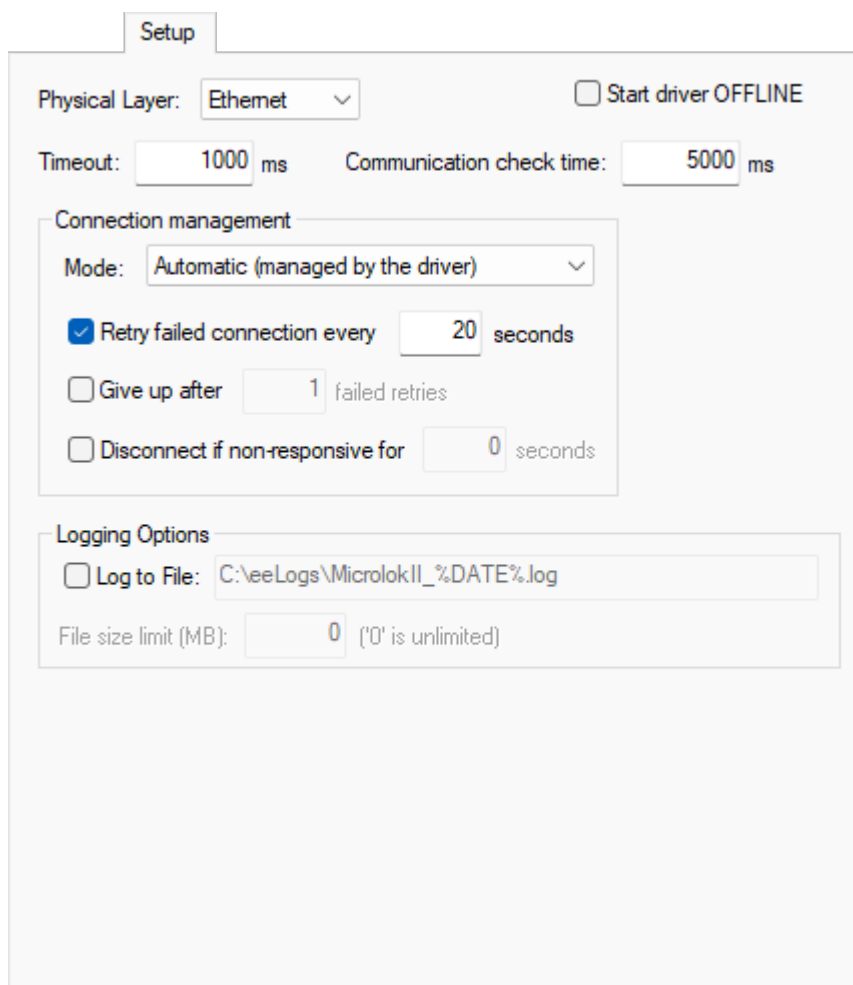
Configuration Dialog Box

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

Setup Tab

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

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



Setup

Physical Layer: Start driver OFFLINE

Timeout: ms Communication check time: ms

Connection management

Mode:

Retry failed connection every seconds

Give up after failed retries

Disconnect if non-responsive for seconds

Logging Options

Log to File:

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

Setup tab

General options on the Setup tab

| OPTION | DESCRIPTION |
|---------------------------------|---|
| Physical Layer | Select the physical layer on a list. Available options are Serial , Ethernet , Modem , and RAS . The selected interface must be configured on its specific tab |
| Timeout | Configure a time-out, in milliseconds, for the physical layer. This is the amount of time an I/O interface waits to receive any byte from the reception's buffer |
| Communication check time | Set the time, in milliseconds, to define the interval at which communication is considered to be in an inactive state. As long as an I/O Driver receives valid data, its communication state is considered active. However, if during operation an I/O Driver does not receive valid data inside this period of time, the state is considered inactive. The communication state is shown in the IO.CommunicationStatus Tag |
| Start driver OFFLINE | Select this option so that a Driver starts in Offline mode or stopped. This means that the I/O interface is not created until this Driver is configured to Online mode by using a Tag in an application. This mode enables a dynamic configuration of an I/O interface at run time |

Options on the Connection management group

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

Options on the Logging Options group

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

General Configurations

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

I/O Tags

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

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

IO.CommunicationStatus

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

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

IO.IOKitEvent

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

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

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

NOTE

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

IO.PhysicalLayerStatus

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

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

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

IO.SetConfigurationParameters

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

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

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

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

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

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

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

A variation of the previous example uses a bidimensional array.

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

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

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

IO.WorkOnline

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

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

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

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

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

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

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

IMPORTANT

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

Properties

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

IO.ConnectionMode

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

IO.GiveUpEnable

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

IO.GiveUpTries

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

IO.InactivityEnable

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

IO.InactivityPeriodSec

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

IO.RecoverEnable

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

IO.RecoverPeriodSec

9 Delay time between two connection attempts, in seconds.

NOTE

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

IO.StartOffline

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

NOTE

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

IO.TimeoutMs

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

IO.Type

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

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

Statistical Configuration

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

I/O Tags

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

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

IO.Stats.Partial.BytesRecv

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

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

IO.Stats.Partial.BytesSent

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

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

IO.Stats.Partial.TimeConnectedSeconds

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

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

IO.Stats.Partial.TimeDisconnectedSeconds

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

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

IO.Stats.Total.BytesRecv

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

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

IO.Stats.Total.BytesSent

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

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

IO.Stats.Total.ConnectionCount

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

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

IO.Stats.Total.TimeConnectedSeconds

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

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

IO.Stats.Total.TimeDisconnectedSeconds

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

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

Properties

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

Driver Revision History

| VERSION | DATE | AUTHOR | COMMENTS |
|---------|------------|-------------|--|
| 2.0.17 | 12/03/2025 | C. Mello | <ul style="list-style-type: none"> Implemented an error by time-out for a lack of response from an SMTP server (<i>Case 38998</i>). Adjustments for compatibility with version 3.0 of OpenSSL library (<i>Case 38988</i>). |
| 2.0.16 | 09/02/2025 | M. Ludwig | <ul style="list-style-type: none"> Driver updated to IOKit library version 3.0 and Visual Studio 2022 (<i>Case 37965</i>). |
| 2.0.15 | 04/28/2025 | A. Fetzner | <ul style="list-style-type: none"> Implemented the possibility of configuring the To, CC, and BCC fields using the format "Full Name <name@domain.com>" (<i>Case 37076</i>). |
| 2.0.14 | 10/16/2024 | A. Fetzner | <ul style="list-style-type: none"> Added the possibility of configuring a username and password directly on this Driver's properties window, without displaying sensitive data in Tags and in the log (<i>Case 36758</i>). Updates in the documentation (<i>Case 36486</i>). |
| 2.0.13 | 07/18/2023 | A. Fetzner | <ul style="list-style-type: none"> Added support for sending e-mails in HTML format (<i>Case 28927</i>). |
| | | P.H. Santos | <ul style="list-style-type: none"> Implemented support for Tag Browser for Tags of the available operations (<i>Case 34525</i>). |
| 2.0.12 | 09/11/2019 | C. Mello | <ul style="list-style-type: none"> Platform update in the source code of this Driver (<i>Case 27519</i>). |
| 2.0.11 | 05/22/2018 | C. Mello | <ul style="list-style-type: none"> Added support for sending e-mails with multiple attached files (<i>Case 24238</i>). |
| 2.0.10 | 02/05/2018 | C. Mello | <ul style="list-style-type: none"> Added an option to configure a time-out limit for connections (<i>Case 23811</i>). |
| 2.0.9 | 08/02/2017 | M. Ludwig | <ul style="list-style-type: none"> Fixed a block caused by a pre-existing connection |

| VERSION | DATE | AUTHOR | COMMENTS |
|---------|------------|-----------|--|
| | | | when sending e-mails (<i>Case 23097</i>). |
| 2.0.8 | 06/30/2017 | C. Mello | <ul style="list-style-type: none"> Added support for asynchronous writings using the WriteEx method with the <i>WriteSyncMode</i> parameter equal to 2 (two) and for returning error codes via callbacks when reading a Tag with the <i>N1</i> parameter equal to 90 (<i>Case 22752</i>). |
| 2.0.7 | 05/09/2017 | C. Mello | <ul style="list-style-type: none"> Added the return of error codes in writing Tags using the WriteEx method of scripts in Eclipse E3, Eclipse Power, or Eclipse Water applications (<i>Case 22360</i>). |
| 2.0.6 | 10/23/2015 | C. Mello | <ul style="list-style-type: none"> Improvements when receiving data for a better compatibility with SMTP servers (<i>Case 19901</i>). |
| 2.0.5 | 02/11/2015 | C. Mello | <ul style="list-style-type: none"> Added support for filling the body of an e-mail with the content of an external text file, saved in ANSI or Unicode standard (<i>Case 17255</i>). Adjustments to define as default the None option for IOKit library's <i>Physical Layer</i> parameter (<i>Case 18157</i>). |
| 2.0.4 | 08/06/2014 | M. Ludwig | <ul style="list-style-type: none"> Adjustments for AUTH LOGIN-type authentication (<i>Case 16654</i>). |
| 2.0.2 | 10/31/2013 | C. Mello | <ul style="list-style-type: none"> Improvements to identify the type of authentication required by an SMTP server (<i>Case 14214</i>). Improvements to manage socket connections and disconnections (<i>Case 14082</i>). Adjustments to migrate to IOKit library version 2.0 (<i>Case 14200</i>). |
| 1.7.1 | 04/04/2011 | C. Mello | <ul style="list-style-type: none"> Added Tags to configure e-mails at run time (<i>Case 12139</i>). Added support for SSL/TLS-type secure authentication (<i>Case 11963</i>). |

| VERSION | DATE | AUTHOR | COMMENTS |
|--------------|------------|------------|---|
| | | | <ul style="list-style-type: none">• Added an option to define the priority of e-mails and recipients with CC and CCO fields (<i>Case 12040</i>). |
| 1.6.1 | 05/04/2007 | M. Ludwig | <ul style="list-style-type: none">• Fixed a problem when authenticating with a server (<i>Case 8157</i>).• Added support for multiple recipients (<i>Case 7952</i>). |
| 1.4.1 | 07/17/2006 | M. Ludwig | <ul style="list-style-type: none">• Updates and corrections in source code (<i>Case 7143</i>). |
| 1.3.1 | 01/16/2006 | A. Corrêa | <ul style="list-style-type: none">• This Driver now works with servers that require authentication (<i>Case 5100</i>). |
| 1.2.1 | 07/28/2004 | F. Damasio | <ul style="list-style-type: none">• Small review adjustments in source code. |
| 1.1.1 | 04/08/2004 | F. Damasio | <ul style="list-style-type: none">• Initial version of this Driver. |

Headquarters

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

90510-002 — Porto Alegre — RS

Phone: (+55 51) 3346-4699

Fax: (+55 51) 3222-6226

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

Branch in Taiwan

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

807 — Kaohsiung City — Taiwan

Phone: (+886 7) 323-8468

Fax: (+886 7) 323-9656

E-mail: evan@elipse.com.br

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

www.elipse.com.br

kb.elipse.com.br

forum.elipse.com.br

www.youtube.com/elipsesoftware

elipse@elipse.com.br



Gartner, Cool Vendors in Brazil 2014, April 2014.

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

Microsoft Partner
Gold Independent Software Vendor (ISV)