

Siemens ISO-on-TCP Driver

File Name	IsoTcp.dll
Manufacturer	Siemens AG
Devices	S7-300, S7-400, S7-1200, S7-1500 devices and other devices compatible with the protocol of this Driver
Protocol	S7Comm, ISO-on-TCP (RFC1006 on Ethernet interface)
Version	1.0.1
Last Update	09/24/2026
Platform	Win32
Dependencies	IOKit version 3.0
Superblock Readings	No
Level	0

Introduction

This Driver implements the S7Comm protocol over ISO-on-TCP (IsoTCP), a Siemens AG proprietary protocol carried over TCP/IP on port 102. It allows applications developed by **Eclipse Software** to communicate with devices from SIMATIC S7 family (S7-300, S7-400, S7-1200, and S7-1500 devices) via a standard Ethernet connection.

Preparing a Device

This section describes the required configuration on a Siemens AG device to allow communication via IsoTCP. These procedures vary depending on the model of a device.

S7-1200 and S7-1500 (TIA Portal)

S7-1200 and S7-1500 devices have access restrictions enabled by default. To allow this Driver to read and write data, users must set, in TIA Portal, the configurations indicated on the next table.

Configuration of S7-1200 and S7-1500 devices

CONFIGURATION	PROCEDURE
Enable PUT/GET access	In the properties of a device, on the Protection & Security tab, select the Permit access with PUT/GET communication from remote partner (PLC, HMI, OPC, ...) option. Without this option enabled, a device refuses reading or writing connections via S7Comm protocol
Disable optimized block access on Data Blocks	For each Data Block (DB) that this Driver needs to access, deselect the Optimized block access option in the properties of a block. Data Blocks with optimized access use a different internal layout, which is not accessible via S7Comm protocol with absolute addressing, such as "DB1.DBW0"
Set a fixed IP address	Configure a fixed IP address for the Ethernet interface of a device in the properties of a CPU module, on the

CONFIGURATION	PROCEDURE
	PROFINET interface tab. Use the same IP address from the Main IP option on the Ethernet tab of this Driver

S7-300 and S7-400 (STEP 7 Classic or TIA Portal)

For S7-300 and S7-400 devices, there are no **PUT/GET** protocol restrictions nor optimized block access settings. The necessary configurations are indicated on the next table.

Configuration of S7-300 and S7-400 devices

CONFIGURATION	PROCEDURE
Set the IP address	Configure an IP address on the CPU module or CP Ethernet module via HW Config (STEP 7) or TIA Portal. For S7-300 devices, the default CPU slot is slot 2 (two), rack 0 (zero). For S7-400 devices, the slot may vary. Configure the Rack and Slot options according to the installed hardware
Verify network reachability	Ensure that the computer executing an Eclipse Software application can connect to a device via network using the ping command for the configured IP address. The TCP/IP port 102 must be accessible, without a firewall blocking

NOTE

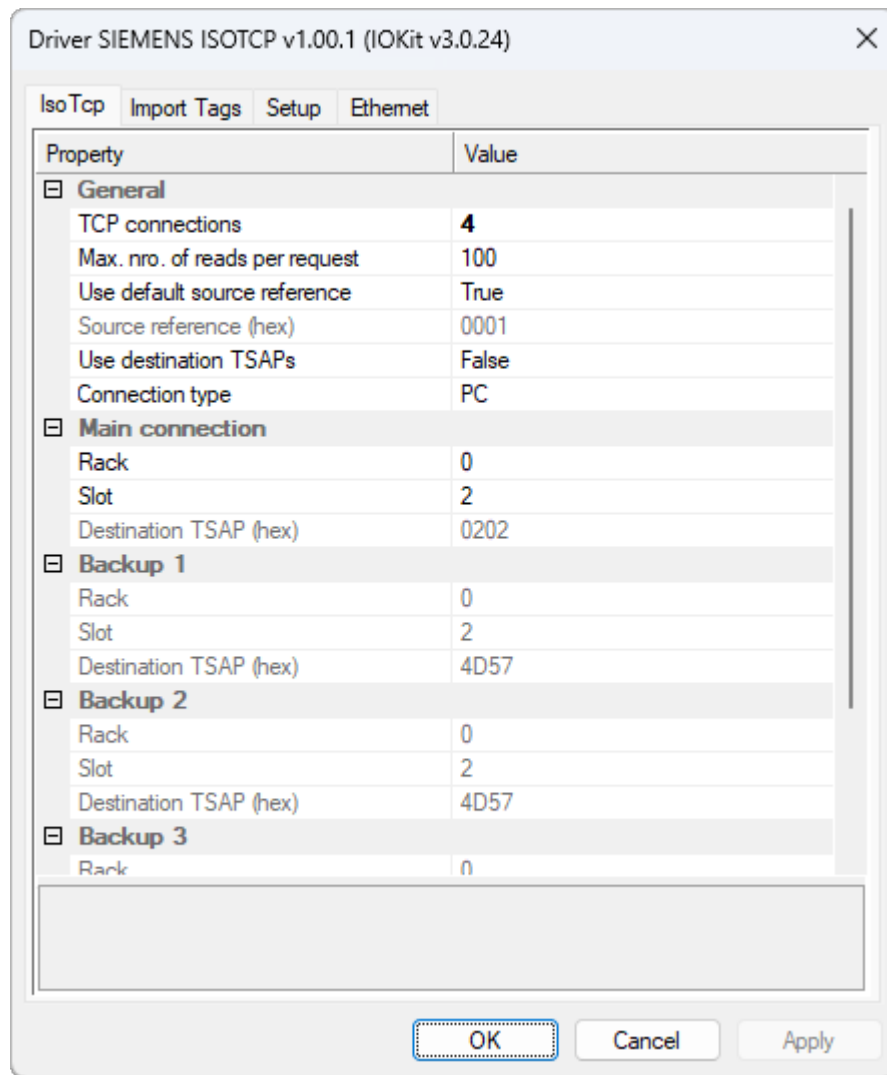
The default TCP port for the ISO-on-TCP protocol is **102**. This port must be open in the firewall of a device and in the supervisory server to establish a connection.

Driver Configuration

The **[P]** parameters of this Driver are not used. All settings must be performed on the properties window of this Driver. For more information about the configuration tabs, please check topic **Documentation of I/O Interfaces**.

Configuring Properties

The configuration parameters for communicating with a device are available on the properties window of this Driver, shown on the next figure. In addition to this properties window, these settings can also be defined at run time in **Eclipse E3**, in **Eclipse Power**, or in **Eclipse Water** applications by starting this Driver in **Offline** mode, that is, by selecting the **Start driver OFFLINE** option on the **Setup** tab.



IsoTcp tab

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

Configuration options of Siemens ISO-on-TCP Driver

PARAMETER	OFFLINE STRING	DATA TYPE	DESCRIPTION
Connection type	ISOTCP.ConnectionType	Number	Type of S7 connection. Valid values are 0 : PG (Programming Device), 1 : OP (Operator Panel), or 2 : PC (Standard)
TCP connections	ISOTCP.TCPConnections	Number	Number of simultaneous TCP connections to the main device. Valid values are in the range between 1 (one, default) and 8 (eight). Multiple connections increase reading throughput by executing requests in parallel
Max. nro. of reads per request	ISOTCP.MaxReadPerRequest	Number	Maximum number of variables grouped into a single S7 reading request. Valid values are in the range

PARAMETER	OFFLINE STRING	DATA TYPE	DESCRIPTION
			between 1 (one) and 100 (default). Reduce this value if a device reports errors of exceeded PDU
Use default source reference	ISOTCP.UseDefaultSourceReference	Number	Uses the default protocol source reference. Valid values are 0 : Disabled (uses the value configured in the Source reference (hex) parameter) or 1 : Enabled (default)
Source reference (hex)	ISOTCP.SourceReference	Text	Source reference of ISO connection, in hexadecimal, with up to 4 (four) digits. This parameter is used only when the Use default source reference parameter is disabled. The default value of this parameter is 0001
Source TSAP (hex)	ISOTCP.SourceTSAP	Text	Source TSAP (<i>Transport Service Access Point</i>) of an ISO connection, in hexadecimal, with up to 4 (four) digits. The default value of this parameter is 0100. On most devices, this default value does not need to be changed
Rack	ISOTCP.CPURack	Number	Rack number of the main CPU. Valid values are in the range between 0 (zero, default) and 7 (seven). This parameter is used to calculate the destination TSAP when the Use destination TSAPs parameter is disabled
Slot	ISOTCP.CPUSlot	Number	Slot number of the main CPU. Valid values are in the range between 1 (one) and 31. The default value of this parameter is 2 (two, default slot for S7-300 or 400 devices). For S7-1200 and S7-1500 devices, use slot 1 (one)
Use destination TSAPs	ISOTCP.UseDestTSAP	Number	When enabled, the destination TSAPs are informed manually instead of calculated based on Rack and Slot parameters. Valid values are 0 : Disabled (default, TSAP calculated

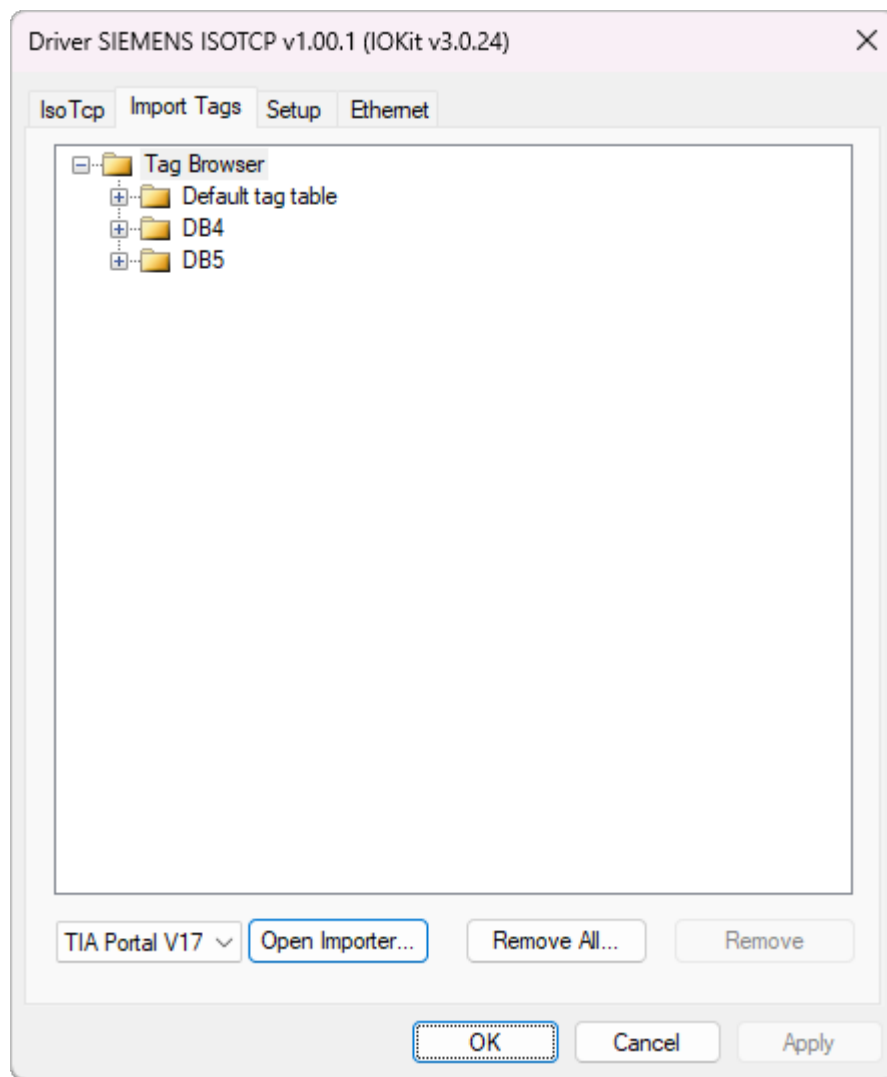
PARAMETER	OFFLINE STRING	DATA TYPE	DESCRIPTION
			based on Rack , Slot , and Connection type parameters) or 1 : Enabled (uses values from the Destination TSAP (hex) parameter)
Destination TSAP (hex)	ISOTCP.DestTSAP	Text	Destination TSAP of the main CPU, in hexadecimal, with up to 4 (four) digits. This parameter is Used only when the Use destination TSAPs parameter is enabled. The default value of this parameter is 0202
Enable CPU Backup	ISOTCP.EnableCPUBackup	Number	Enables the configuration of a backup or redundancy device. Valid values are 0 : Disabled (default) or 1 : Enabled (the IP address of a backup device is configured in the Backup IP option on the Ethernet tab)
Rack (Backup 1)	ISOTCP.CPUBackupRack	Number	Rack number of backup CPU 1 (one). Valid values are in the range between 0 (zero, default) and 7 (seven)
Slot (Backup 1)	ISOTCP.CPUBackupSlot	Number	Slot number of backup CPU 1 (one). Valid values are in the range between 1 (one) and 31. The default value of this parameter is 2 (two)
Destination TSAP Backup (hex)	ISOTCP.DestTSAPBackup	Text	Destination TSAP of backup device 1 (one), in hexadecimal. This parameter is used only when the Use destination TSAPs parameter is enabled. The default value of this parameter is 4D57
Enable CPU Backup 2	ISOTCP.EnableCPUBackup2	Number	Enables the configuration of a second backup device. Valid values are 0 : Disabled or 1 : Enabled
Rack (Backup 2)	ISOTCP.CPUBackup2Rack	Number	Rack number of backup CPU 2 (two). Valid values are in the range between 0 (zero, default) and 7 (seven)
Slot (Backup 2)	ISOTCP.CPUBackup2Slot	Number	Slot number of backup CPU 2 (two). Valid values are in the range between 1 (one) and 31. The default value of this parameter is 2 (two)

PARAMETER	OFFLINE STRING	DATA TYPE	DESCRIPTION
Destination TSAP Backup 2 (hex)	ISOTCP.DestTSAPBackup2	Text	Destination TSAP of backup device 2 (two), in hexadecimal. This parameter is used only when the Use destination TSAPs parameter is enabled. The default value of this parameter is 4D57
Enable CPU Backup 3	ISOTCP.EnableCPUBackup3	Number	Enables the configuration of a third backup device. Valid values are 0 : Disabled or 1 : Enabled
Rack (Backup 3)	ISOTCP.CPUBackup3Rack	Number	Rack number of backup CPU 3 (three). Valid values are in the range between 0 (zero, default) and 7 (seven)
Slot (Backup 3)	ISOTCP.CPUBackup3Slot	Number	Slot number of backup CPU 3 (three). Valid values are in the range between 1 (one) and 31. The default value of this parameter is 2 (two)
Destination TSAP Backup 3 (hex)	ISOTCP.DestTSAPBackup3	Number	Destination TSAP of backup device 3 (three), in hexadecimal. This parameter is used only when the Use destination TSAPs parameter is enabled. The default value of this parameter is 4D57

All offline properties must be configured via PLC Tags in **String** format using the *N1* parameter equal to -1 (minus one), the *N2* parameter equal to 0 (zero), the *N3* parameter equal to 0 (zero), and the *N4* parameter equal to 3 (three). For more information and examples, please check topic **Documentation of I/O Interfaces**.

Importing Tags

The **Import Tags** tab brings some features for importing Tags from a **TIA Portal** (*Totally Integrated Automation Portal*) automation project.



Import Tags tab

This tab shows a tree view structure of Tags already imported and saved to the project of this Driver. When including a Driver in a project, the initial state is an empty structure, only with a **Tag Browser** folder. Users can include imported Tags by selecting the TIA Portal version and clicking the **Open Importer** option, described on topic **IsoTcp TIA Portal Importer**.

If there are Tags already imported to this tree view structure, users can use the **Remove** option to remove the selected Tag and the **Remove All** option to remove all imported Tags. Click **OK** to save all Tags and display them on the **Tag Browser** tool.

NOTES

- Only the **LReal** data type is not supported when importing Tags. If there are variables with this data type in a project, they are ignored by the importer, which proceeds with the import process of the variables with other data types.
- Blocks with **ArrayDB** and **InstanceDB** data types are not supported when importing Tags.

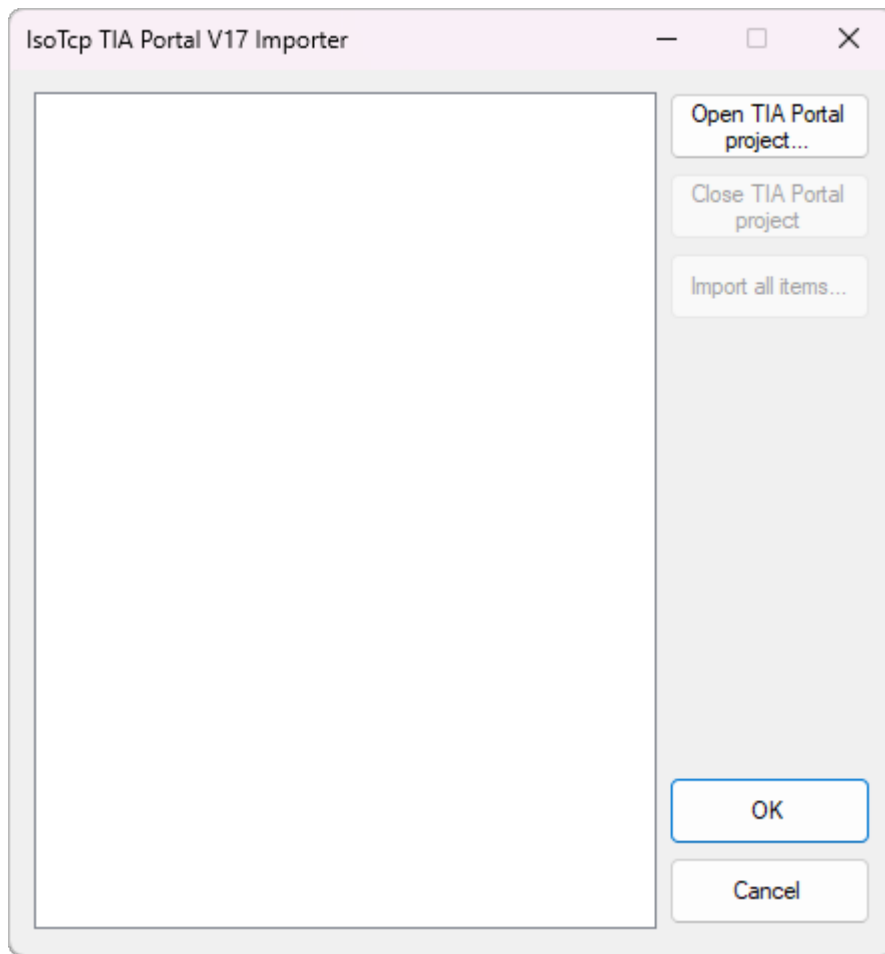
IsoTcp TIA Portal Importer

IsoTcp TIA Portal Importer is an executable file, distributed along with Siemens ISO-on-TCP Driver, intended to import Tags from an automation project created using TIA Portal by Siemens AG. To ensure its proper functioning, this executable must match the following requirements:

- Siemens TIA Portal versions 16, 17, 18, 19, or 20 installed
- Microsoft .NET Framework version **4.8** or later installed

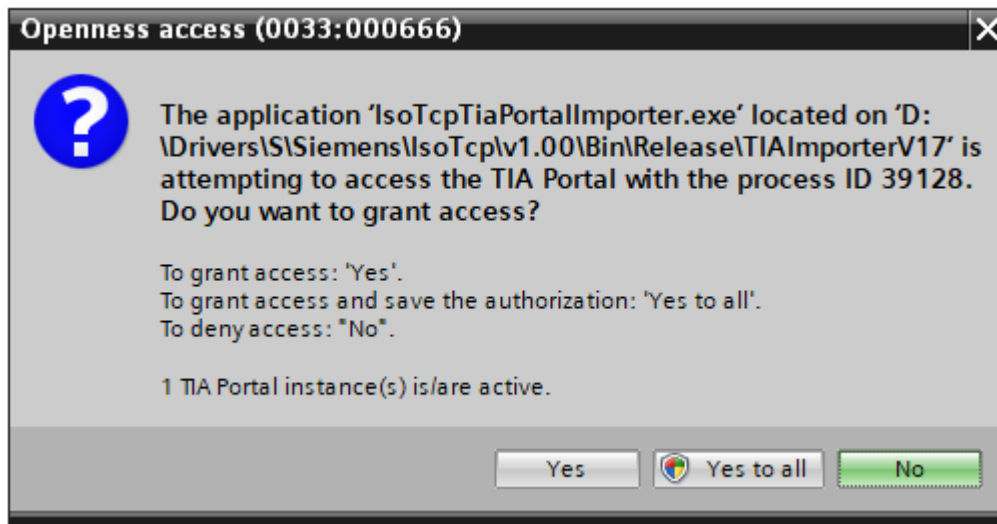
- Folders **TIAImporterV16**, **TIAImporterV17**, **TIAImporterV18**, **TIAImporterV19**, and **TIAImporterV20**, provided with this Driver, must be located in the same folder as this Driver
- Files **IsoTcpTiaPortalImporter.exe** and **Siemens.Engineering.dll**, provided with this Driver, must be located within the folders mentioned earlier
- The user account using the Importer must be added to the **Siemens TIA Openness** local group of users, located on Windows Administrative Tools

To open IsoTcp TIA Portal Importer, click **Open Importer** on the **Import Tags** tab of the properties window of this Driver to open the window on the next figure.



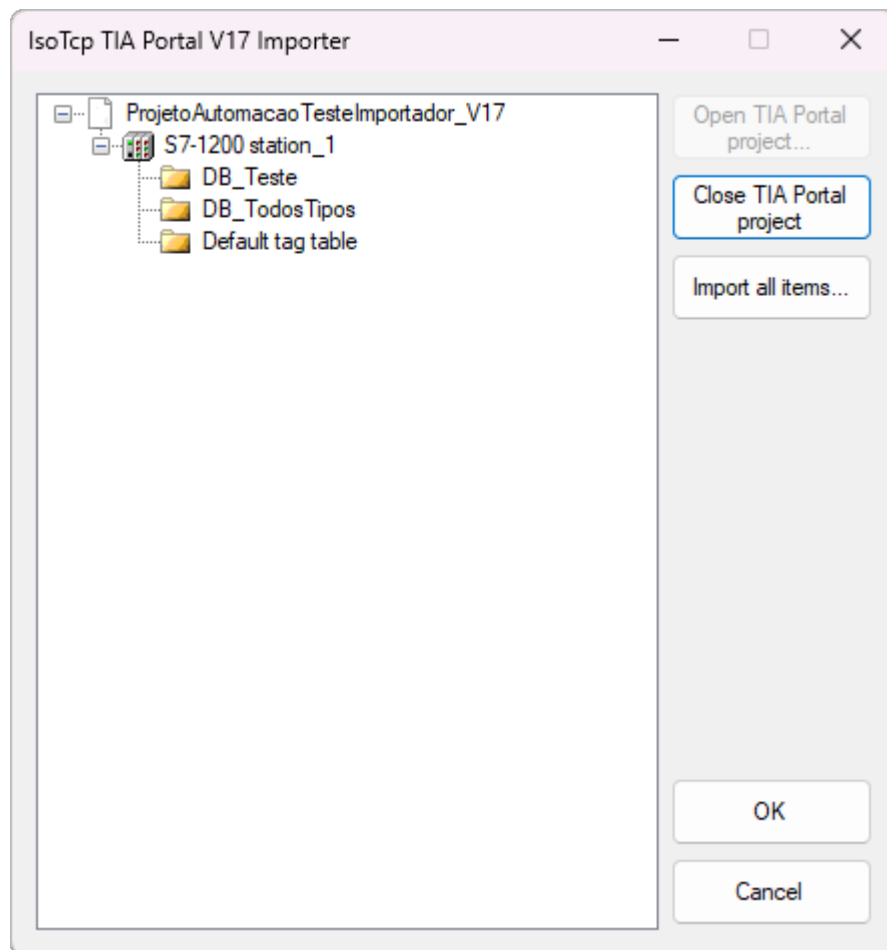
IsoTcp TIA Portal Importer

Click **Open TIA Portal project** to select a TIA Portal project file and click **OK** to open an instance of TIA Portal in background. If TIA Portal firewall does not find **IsoTcp TIA Portal Importer** on the authorization list, a dialog box is opened with options **Yes** to allow access, **Yes to all** to allow access and save the access authorization on the authorization list, and **No** to refuse access, according to the next figure.



Access configuration

After authorizing **IsoTcp TIA Portal Importer**, it is then opened displaying the project's tree structure with the items to import.



Items to import

To import all Tags from a project, click **Import all items**. After finishing the import process, click **OK** to confirm it and close the Importer. When closing the Importer, the Tags are automatically imported into this Driver's properties window, where they can be saved. The **Close TIA Portal project** option allows closing a project and the **Cancel** option closes the Importer without performing an import process.

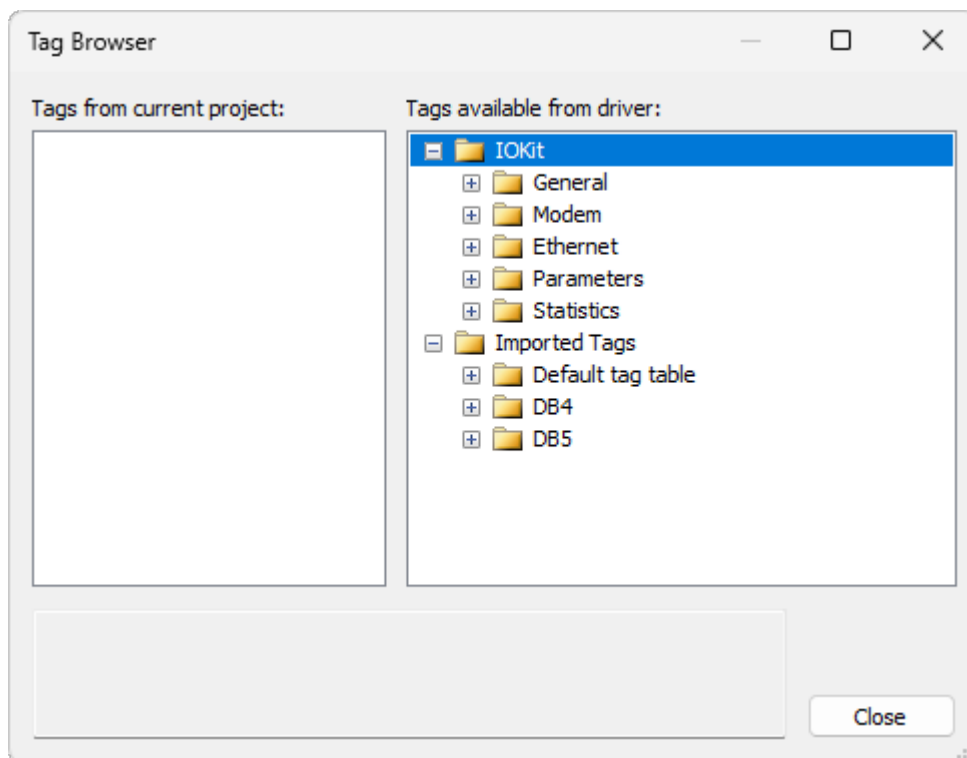
Tag Reference

This section describes how to configure Tags for Siemens ISO-on-TCP Driver to access memory areas of a device by Siemens AG. The address of each Tag is defined by the **Address** (Item) field using the S7 addressing syntax described in the next topics. Please check topic **Data Tags** to check a full description of that addressing syntax and configuration examples.

Using Tag Browser

Elipse E3 starting with version **2.0**, **Elipse Power**, and **Elipse Water** have a tool called **Tag Browser**, which allows a Driver to help users on creating and configuring Tags.

Tag Browser aims to automate Tag configuration by automatically filling the **Item** field and the proper size of Elements, if it is a Block Tag.



Tag Browser window

The **Tags from current project** list displays Tags and folders already in the current project. The **Tags available from driver** list displays a tree with Tags provided by a Driver. To create a Tag in an application, drag one of the Tags defined by this Driver to a folder on the current directory. Siemens ISO-on-TCP Driver provides the following nodes on this tree:

- The **IOKit** node displays Tags available on **IOKit** library, grouped into the following categories:
 - **General**: Tags for general use
 - **Ethernet**: Tags for manipulating communication via Ethernet network
 - **Parameters**: Tags for configuring **IOKit** library parameters
 - **Statistics**: Tags with communication layer statistics
- The **Imported Tags** node shows all Tags imported and saved on the **Import Tags** tab. This node only appears if there are actually imported Tags. In this case, this node contains a folder for each separate group of imported Tags. If there is a Data Block (DB), the Tags referring to variables are grouped into one folder in reference to this DB. The other Tags are grouped into folders referring to the table to which they belong. Each Tag is created with the **Device** field with a default value of 0 (zero)

Data Tags

Reading or Writing

Use a **PLC** or **Block** Tag to read and write values in memory areas of a device by Siemens AG. The address is informed in the Tag **Address** (Item) field using the S7 syntax described in this topic. The field is case-insensitive.

Supported memory areas

AREA	ADDRESS SYNTAX	EXAMPLES
Data Block	DB<n>.DB<type> <offset> DB<n>.DBX<offset>.<bit> DB<n>.DBS(<len>)	DB1.DBW0, DB1.DBD10, DB1.DBF4 DB1.DBX0.3 (bit three of byte zero) DB1.DBS(254) (String with up to 254 characters)
Merker (M)	M<type> <offset> M<offset>.<bit>	MB0, MW10, MD20, MF4 M5.7 (bit seven of byte five)
Inputs (I)	I<type> <offset> I<offset>.<bit>	IB0, IW2, ID4 I0.0 (bit zero of input byte zero)
Outputs (Q)	Q<type> <offset> Q<offset>.<bit>	QB0, QW2, QD4 Q0.5 (bit five of output byte zero)
Analog Inputs (AI)	AIW<offset>	AIW0, AIW2
Analog Outputs (AQ)	AQW<offset>	AQW0, AQW2
Timers (T)	T<n>	T0, T10
Counters (C)	C<n>	C0, C5
High-Speed Counters (HC)	HC<n>	HC0, HC1

Values of <type> for Data Blocks (DB<n>.DB<type><offset>)

TYPE	DATA TYPE	SIZE	EXAMPLE
DBX	Bool (requires a .bit suffix, from zero to seven)	1 (one) bit	DB1.DBX0.3
DBB	Byte (U8)	1 (one) byte	DB1.DBB0
DBW	Word (U16)	2 (two) bytes	DB1.DBW0
DBD	DWord (U32)	4 (four) bytes	DB1.DBD0
DBF	Real (R32, floating point)	4 (four) bytes	DB1.DBF4
DBI	Int (I16)	2 (two) bytes	DB1.DBI0
DBDI	DInt (I32)	4 (four) bytes	DB1.DBDI0
DBS[n]	S7 String (maximum of <i>n</i> characters)	<i>n</i> + 2 (two) bytes	DB1.DBS[254]
DBWS[n]	S7 WString (maximum of <i>n</i> characters in UTF-16 format)	<i>n</i> × 2 (two) + 4 (four) bytes	DB1.DBWS[50]

Values of <type> for other areas (M, I, Q, SM, S, AI, and AQ)

TYPE	DATA TYPE	SIZE	EXAMPLE (M AREA)
X	Bool (requires a .bit suffix, from zero to seven) or omit X and use only M0.3	1 (one) bit	MX0.3 or M0.3
B	Byte (U8)	1 (one) byte	MB0
W	Word (U16)	2 (two) bytes	MW0
D	DWord (U32)	4 (four) bytes	MD0
F	Real (R32, single-precision float)	4 (four) bytes	MF0
I	Int (I16, signed integer)	2 (two) bytes	MI0
DI	DInt (I32, signed long integer)	4 (four) bytes	MDI0
S[n]	S7 String (maximum of <i>n</i> characters)	<i>n</i> + 2 (two) bytes	MS[100]
WS[n]	S7 WString (maximum of <i>n</i> characters in UTF-16 format)	<i>n</i> × 2 (two) + 4 (four) bytes	MWS[50]

An address also accepts the % prefix (TIA Portal/IEC 61131-3 syntax), which is ignored, such as "%MW0" or "%DB1.DBX0.0".

NOTE

For S7-1200 and S7-1500 devices, Data Blocks must have the **Optimized block access** option disabled in TIA Portal for absolute addressing, such as "DB1.DBW0", to work correctly. Please check topic **Preparing a Device** for more information.

Documentation of I/O Interfaces

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

Configuration of a Driver

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

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

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

1. Open the Organizer.

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

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

Configuration Dialog Box

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

Setup Tab

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

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

The screenshot shows the 'Setup' tab of a configuration dialog. It features several input fields and checkboxes organized into sections. The 'Physical Layer' section has a dropdown menu currently showing 'Ethernet' and an unchecked checkbox labeled 'Start driver OFFLINE'. The 'Timeout' section includes a text box with '1000' followed by 'ms'. The 'Communication check time' section has a text box with '5000' followed by 'ms'. The 'Connection management' section contains a dropdown menu set to 'Automatic (managed by the driver)', a checked checkbox for 'Retry failed connection every' with a text box of '20' and the unit 'seconds', an unchecked checkbox for 'Give up after' with a text box of '1' and the unit 'failed retries', and another unchecked checkbox for 'Disconnect if non-responsive for' with a text box of '0' and the unit 'seconds'. The 'Logging Options' section at the bottom has an unchecked checkbox for 'Log to File' with a text box containing the path 'C:\eeLogs\MicrolokII_%DATE%.log', and a text box for 'File size limit (MB)' with the value '0' and a note '('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	Enable this option and configure the name of a file to write a log. Log files can be large, so use this option for short periods of time, only for testing and debugging purposes. If the %PROCESS% macro is used in the log file name, it is replaced by the identifier of the current process. This option is particularly useful when using several instances of the same Driver in Elipse E3, thus allowing each instance to generate a separate log file. For example, when configuring this option with value "c:\e3logs\drivers\sim_%PROCESS%.log", it generates a file named c:\e3logs\drivers\sim_00000FDA.log for process 0FDAh. Users can also use the %DATE% macro in the file name. In this case a log file is generated every day, in the format aaaa_mm_dd. For example, when configuring this option with value "c:\e3logs\drivers\sim_%DATE%.log", it generates a file named c:\e3logs\drivers\sim_2005_12_31.log in 12/31/2005 and a file named c:\e3logs\drivers\sim_2006_01_01.log in 01/01/2006. Similarly, the %DATE_HOUR% macro generates one log file per hour, in the format aaaa_mm_dd_hh
File size limit (MB)	Configure the log file size limit, in megabytes. A value equal to 0 (zero) means that there is no size limit for the log file

Ethernet Tab

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

Ethernet

Transport: TCP/IP ☐ Listen for connections on port: 0

☐ PING before connecting ☐ Share listen port with other processes

Timeout: 4000 ms ☐ Interface: (All Interfaces)

Retries: 1 ☐ Use IPv6 ☐ Use SSL SSL Settings

☐ Enable 'ECHO' suppression

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 the Ethernet tab**

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

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

Available options on the Connect to group

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

General Configurations

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

I/O Tags

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

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

IO.CommunicationStatus

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

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

IO.IOKitEvent

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

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

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

NOTE

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

IO.PhysicalLayerStatus

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

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

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

IO.SetConfigurationParameters

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

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

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

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

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

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

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

A variation of the previous example uses a bidimensional array.

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

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

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

IO.WorkOnline

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

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

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

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

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

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

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

IMPORTANT

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

Properties

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

IO.ConnectionMode

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

IO.GiveUpEnable

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

IO.GiveUpTries

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

IO.InactivityEnable

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

IO.InactivityPeriodSec

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

IO.RecoverEnable

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

IO.RecoverPeriodSec

9 Delay time between two connection attempts, in seconds.

NOTE

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

IO.StartOffline

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

NOTE

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

IO.TimeoutMs

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

IO.Type

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

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

Statistical Configuration

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

I/O Tags

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

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

IO.Stats.Partial.BytesRecv

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

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

IO.Stats.Partial.BytesSent

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

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

IO.Stats.Partial.TimeConnectedSeconds

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

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

IO.Stats.Partial.TimeDisconnectedSeconds

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

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

IO.Stats.Total.BytesRecv

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

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

IO.Stats.Total.BytesSent

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

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

IO.Stats.Total.ConnectionCount

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

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

IO.Stats.Total.TimeConnectedSeconds

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

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

IO.Stats.Total.TimeDisconnectedSeconds

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

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

Properties

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

Ethernet Interface Configuration

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

I/O Tags

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

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

IMPORTANT

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

IO.Ethernet.IPSelect

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

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

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

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

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

IO.Ethernet.IPSwitch

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

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

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

IO.Ethernet.SocketState

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

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

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

Properties

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

NOTE

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

IO.Ethernet.AcceptConnection

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

IO.Ethernet.BackupEnable[2,3]

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

IO.Ethernet.BackupIP[2,3]

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

IO.Ethernet.BackupLocalPort[2,3]

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

IO.Ethernet.BackupLocalPortEnable[2,3]

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

IO.Ethernet.BackupPort[2,3]

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

IO.Ethernet.IPFilter

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

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

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

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

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

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

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

IO.Ethernet.ListenIP

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

IO.Ethernet.ListenPort

 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 sent to a device.

IO.Ethernet.Transport

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

IO.Ethernet.UseIPv6

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

Driver's Revision History

VERSION	DATE	AUTHOR	COMMENTS
1.0.1	09/24/2026	M. Ludwig	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)