TSAA Communication Driver

Driver for UDP/IP Communication
with Triconex Devices Using TSAA Protocol

Contents

Introduction

The TSAA driver enables communication between the Studio system and Triconex devices using the TSAA protocol over UDP/IP, according to the specifications discussed in this document.

This document will help you to select, configure and execute the TSAA driver, and it is organized as follows:

- **Introduction**: This section, which provides an overview of the document.
- **General Information**: Identifies all of the hardware and software components required to implement communication between the Studio system and the target device.
- **Selecting the Driver**: Explains how to select the TSAA driver in the Studio system.
- **Configuring the Device**: Describes how the target device must be configured to receive communication from the TSAA driver.
- **Configuring the Driver**: Explains how to configure the TSAA driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver**: Explains how to execute the TSAA driver during application runtime.
- **Troubleshooting**: Lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.
- **Sample Application**: Explains how to use a sample application to test the TSAA driver configuration
- **Revision History**: Provides a log of all changes made to the driver and this documentation.

Notes:
- This document assumes that you have read the “Development Environment” chapter in Studio’s *Technical Reference Manual*.
- This document also assumes that you are familiar with the Microsoft Windows environment. If you are not familiar with Windows, then we suggest using the **Help** feature (available from the Windows desktop **Start** menu) as you work through this guide.

General Information

This chapter identifies all of the hardware and software components required to implement communication between the TSAA driver in Studio and a target device using the TSAA protocol over UDP/IP.

The information is organized into the following sections:

- Device Specifications
- Network Specifications
- Driver Characteristics
- Conformance Testing

Device Specifications

To establish communication, your target device must meet the following specifications:

- **Manufacturer**: Triconex
- **Compatible Equipment**:
  - Tricon Series PLC
  - Trident Series PLC
- **Triconex Programmer Software**: TriStation 1131
- **Device Runtime Software**: None
- **PLC Addressing Method**: PLC Tags Modbus Aliases (no Tagnames)

For a description of the device(s) used to test driver conformance, see "Conformance Testing" on the next page.

Network Specifications

To establish communication, your device network must meet the following specifications:

- **Device Communication Port**: NCM (Network Communication Module) TSAA Port
- **Physical Protocol**: Ethernet/UDP-IP
- **Logic Protocol**: TSAA
- **Specific PC Board**: Any Ethernet Adapter (Ethernet board)

Driver Characteristics

The TSAA driver package consists of the following files, which are automatically installed in the **`/DRV`** subdirectory of Studio:

- **`TSAA.INI:`** Internal driver file. *You must not modify this file*.
- **`TSAA.MSG:`** Internal driver file containing error messages for each error code. *You must not modify this file*.
- **`TSAA.PDF:`** This document, which provides detailed information about the TSAA driver.
- **`TSAA.DLL:`** Compiled driver.

Note:
You must use Adobe Acrobat® Reader™ to view the **`TSAA.PDF`** document. You can install Acrobat Reader from the Studio installation CD, or you can download it from Adobe's Web site.

You can use the TSAA driver on the following operating systems:

- Windows XP/7/8 and Servers

For a description of the operating systems used to test driver conformance, see “Conformance Testing” below.

The TSAA driver supports the following registers:

Register Type	Length	Write	Read	Bit	Integer	Float
0x (Coil Status)	1 Bit	●	●	●	–	–
1x (Input Status)	1 Bit	–	●	●	–	–
3x (Input Register)	1 Word	–	●	–	●	●
4x (Holding Register)	1 Word	●	●	–	●	●

Conformance Testing

The following hardware/software was used for conformance testing:

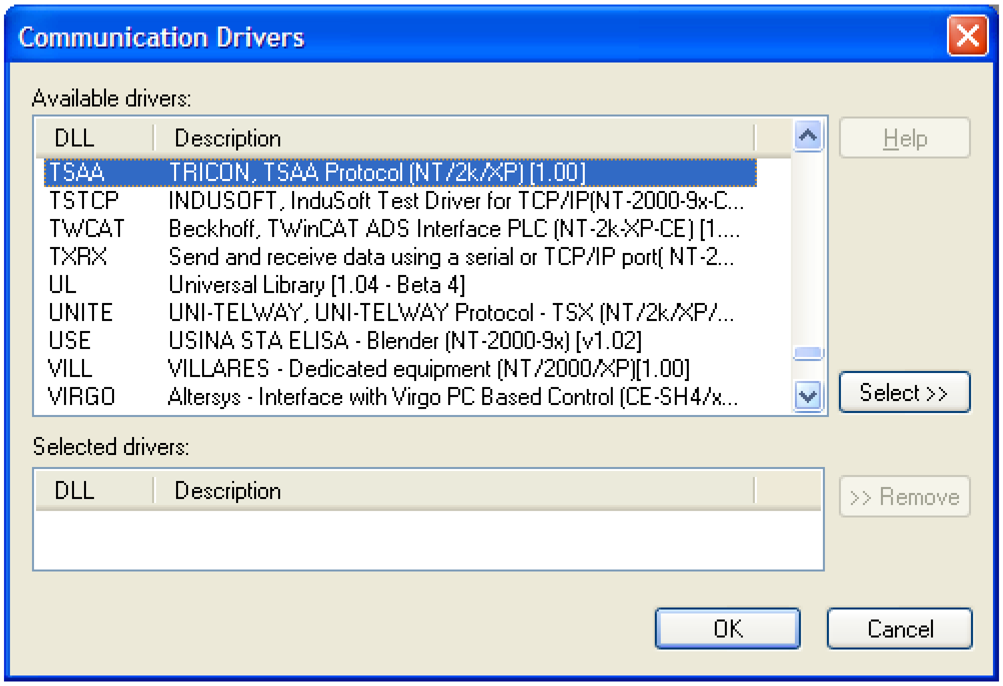
- **Driver Configuration**:
  - **Operation System (development)**: Windows XP
  - **Operation System (target)**: Windows XP
  - **Driver Version**: 1.01
  - **Cable**: Ethernet Cable (10BaseT RJ45 on PC) + Switch + Ethernet Cable (10Base2 on PLC NCM Card)

Driver Version	Studio Version	Operating System (development)	Operating System (runtime)	Equipment
1.01	6.1 + SP5	WinXP + SP2	▪ WinXP + SP2	Triconex 3008/N Tricon Enhanved Main Processor + 4329/N/G NCM (Network Communications Module)
1.02	6.1 + SP5	WinXP + SP2	▪ WinXP + SP2	Triconex Trident MP3101 Main Processor Module + CM3201 (Communication Module)
2.2	8.0	Windows 7	▪ WinXP + SP3	Triconex Trident MP3101 Main Processor Module + CM3201 (Communication Module)

Selecting the Driver

When you install Studio, all of the communication drivers are automatically installed in the **\DRV** subdirectory but they remain dormant until manually selected for specific applications. To select the TSAA driver for your Studio application:

1. From the main menu bar, select **Insert** → **Driver** to open the *Communication Drivers* dialog.
2. Select the **TSAA** driver from the *Available Drivers* list, and then click the **Select** button.



DLL	Description
TSAA	TRICON, TSAA Protocol [NT/2k/XP] [1.00]
TSTCP	INDUSOFT, InduSoft Test Driver for TCP/IP[NT-2000-9x-C...
TWCAT	Beckhoff, TWinCAT ADS Interface PLC (NT-2k-XP-CE) [1....
TXRX	Send and receive data using a serial or TCP/IP port(NT-2...
UL	Universal Library [1.04 - Beta 4]
UNITE	UNI-TELWAY, UNI-TELWAY Protocol - TSX (NT/2k/XP/...
USE	USINA STA ELISA - Blender (NT-2000-9x) [v1.02]
VILL	VILLARES - Dedicated equipment (NT/2000/XP)[1.00]
VIRGO	Altersys - Interface with Virgo PC Based Control (CE-SH4/x...

Communication Drivers Dialog

3. When the **TSAA** driver is displayed in the **Selected Drivers** list, click the **OK** button to close the dialog. The driver is added to the *Drivers* folder, in the *Comm* tab of the Workspace.

Note:
It is not necessary to install any other software on your computer to enable communication between Studio and your target device. However, this communication can only be used by the Studio application; it cannot be used to download control logic to the device. To download control logic to a TSAA device, you must also install the TSAA programming software (e.g., TriStation 1131). For more information, please consult the documentation provided by the device manufacturer.

Attention:
For safety reasons, you must take special precautions when installing any physical hardware. Please consult the manufacturer's documentation for specific instructions.

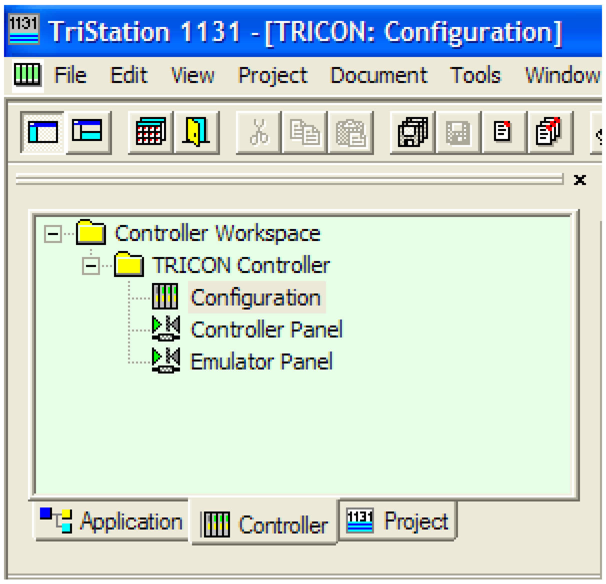
Configuring the Device

You must configure the following parameters on the NCM card using your TriStation Package:

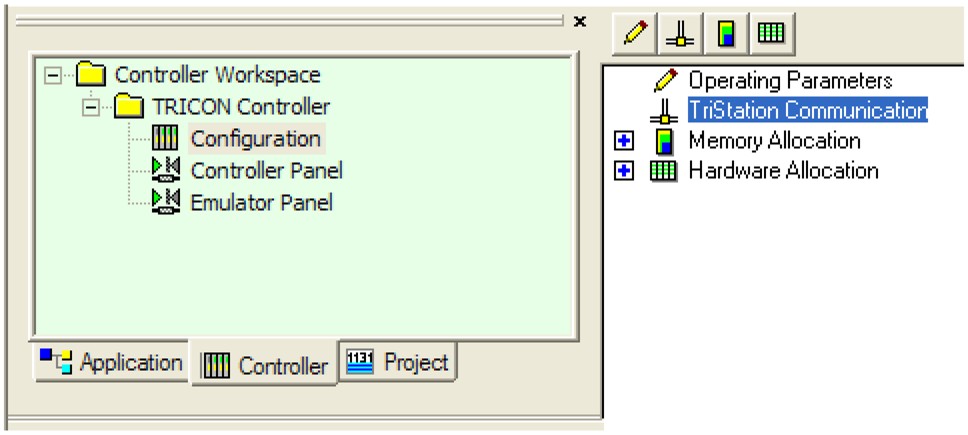
- Node Number
- IP Address
- Node Name

In order to do that, using TriStation 1131 v4.1, you can follow these steps below:

On the **Controller** Tab, go to the TreeView window, expand **Controller Interface,** then **Tricon Controller**, finally selecting **Configuration**

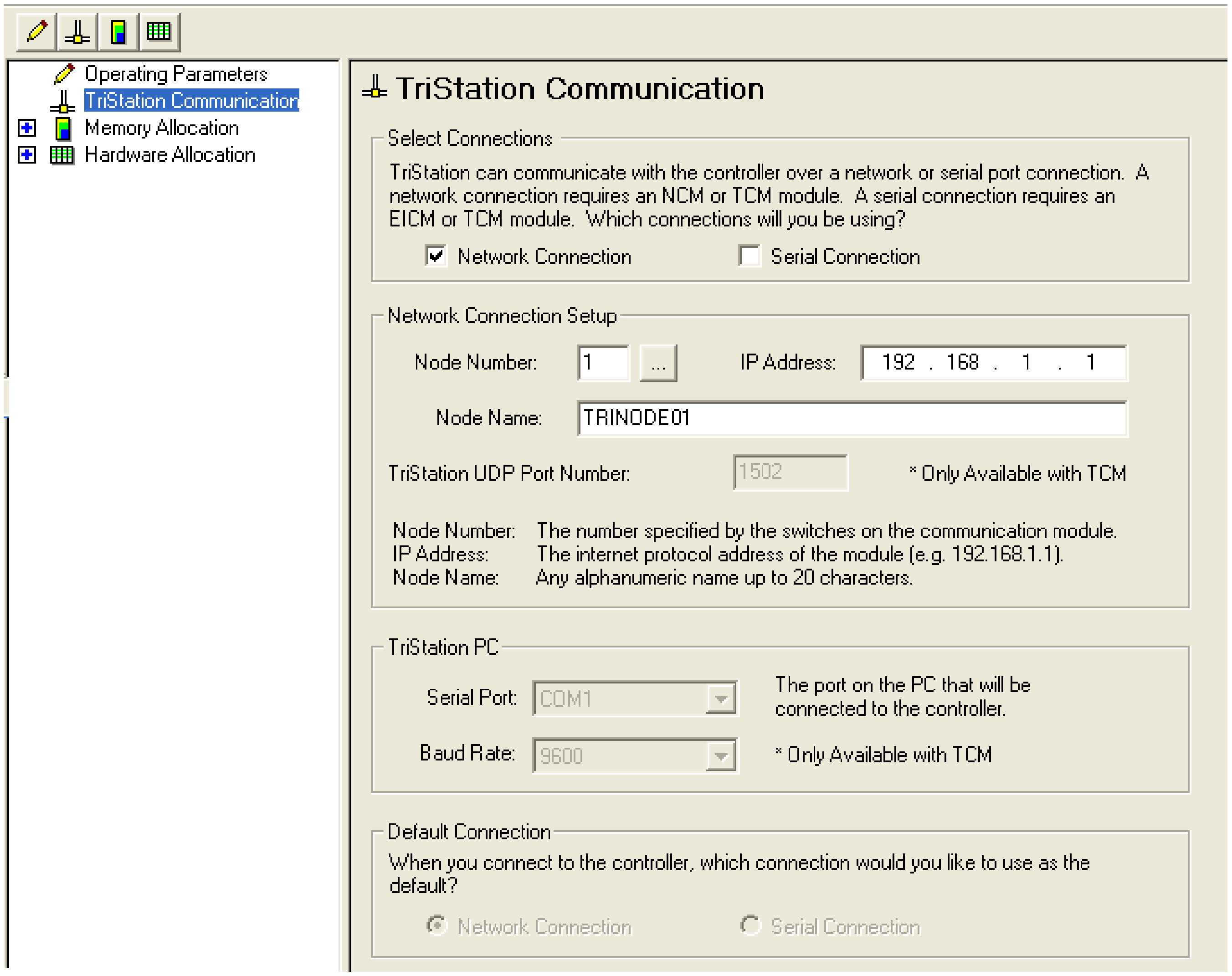


Now you can look on the right panel, and select **TriStation Communication**



Now you will be able to select **Network** connections, and, in the *Network Connection Setup* setting, configure the **Node Number, IP Address** and **Node Name.**

UDP/IP Port Number: the TSAA Driver uses by default the UDP/IP port number **1500**. If you change this port, you will need to configure it properly on the driver configurations.



TriStation Communication

Select Connections

TriStation can communicate with the controller over a network or serial port connection. A network connection requires an NCM or TCM module. A serial connection requires an EICM or TCM module. Which connections will you be using?

[✓] Network Connection   [ ] Serial Connection

Network Connection Setup

Node Number: 1   IP Address: 192 . 168 . 1 . 1

Node Name: TRINODE01

TriStation UDP Port Number: 1502   * Only Available with TCM

Node Number:	The number specified by the switches on the communication module.
IP Address:	The internet protocol address of the module (e.g. 192.168.1.1).
Node Name:	Any alphanumeric name up to 20 characters.

TriStation PC

Serial Port: COM1   The port on the PC that will be connected to the controller.

Baud Rate: 9600   * Only Available with TCM

Default Connection

When you connect to the controller, which connection would you like to use as the default?

[✓ Network Connection]   Serial Connection

Once the selected driver and the target device are both properly configured, it is not necessary to install any other software on your computer to enable communication between the host and the device. All runtime communication is handled within your Studio application project. However, programming the device itself — that is, developing control logic and downloading it to the device — still requires using the device's own programming tool (TriStation 1131).

Configuring the Driver

Once you have selected the TSAA driver in Studio, you must properly configure it to communicate with your target device. First, you must set the driver's communication settings to match the parameters set on the device. Then, you must build driver worksheets to associate database tags in your Studio application with the appropriate addresses (registers) on the device.

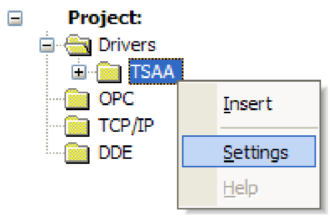
Configuring the Communication Settings

The communication settings are described in detail in the "Communication" chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only TSAA driver-specific settings and procedures will be discussed here. To configure the communication settings for the TSAA driver:

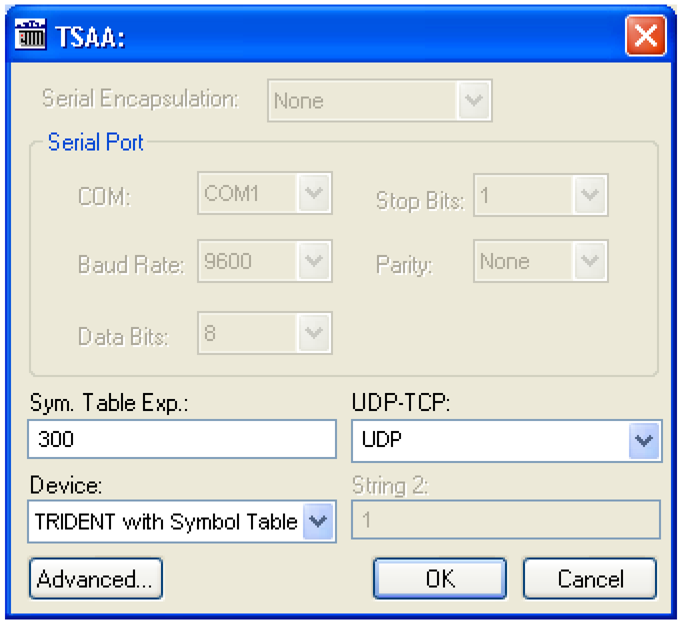
1. In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The TSAA driver is listed here as a subfolder.

2. Right-click on the *TSAA* subfolder and then select the **Settings** option from the pop-up menu:



Select Settings from the Pop-Up Menu

The *TSAA: Communication Settings* dialog is displayed:



Serial Encapsulation: None
COM: COM1
Stop Bits: 1
Baud Rate: 9600
Parity: None
Data Bits: 8
Sym. Table Exp.: 300
UDP-TCP: UDP
Device: TRIDENT with Symbol Table
String 2: 1

TSAA: Communication Settings Dialog

3. In the *Communication Settings* dialog, configure the driver settings to enable communication with your target device. To ensure error-free communication, the driver settings must *exactly match* the corresponding settings on the device. Please consult the manufacturer's documentation for instructions how to configure the device and for complete descriptions of the settings.

   Depending on your circumstances, you may need to configure the driver *before* you have configured your target device. If this is the case, then take note of the driver settings and have them ready when you later configure the device.

Attention:
For safety reasons, you **must** take special precautions when connecting and configuring new equipment. Please consult the manufacturer's documentation for specific instructions.

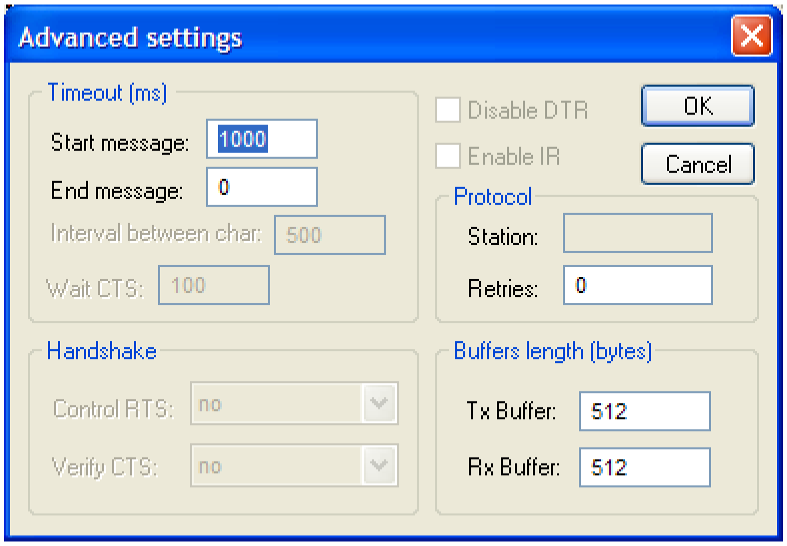
The communication settings and their possible values are described in the following table:

Parameters	Default Values	Valid Values	Description
Sym. Table Exp.	`300`	**Any Integer Number**	This parameter is applicable only when using the "TRIDENT with Symbol Table" mode (see parameter TRICON-TRIDENT). When the TRICON-TRIDENT parameter is set to "TRIDENT with Symbol Table", the communication is performed using the PLC Symbol table, which is downloaded when the communication starts. Because the symbol table can be changed overtime, the driver needs check for changes and re-download the symbol table if necessary. The following options are valid: < 0: The symbol table is downloaded in the first request and no checks for update are made after that. 0: The symbol table is checked in every request. > 0: Version check interval, in seconds. Notice that each check for symbol table changes will require from 560 to 1560 bytes from the PLC (depending on the PLC firmware version). You can also manually control the symbol table update by using the RST header.
UDP – TCP	**`UDP`**	**`UDP or TCP`**	Choose the Ethernet Transport Layer, where the options are **UDP** or **TCP** over IP Network. **IMPORTANT: TCP** mode was not tested by the time of the release of this version
TRICON - TRIDENT	**`TRICON`**	**`TRICON, TRIDENT or TRIDENT with Symbol Table`**	▪ `TRICON`: use it to communicate with Tricon PLC family directly with the Modbus Aliases addresses ▪ `TRIDENT with Symbol Table`: used to communicate with Trident PLC family directly with the Modbus Aliases addresses (**RECOMMENDED** for Trident PLCs) ▪ `TRIDENT`: obsolete - do not use it.

mbol

Note:
The device must be configured with *exactly the same* parameters that you configured in the *TSAA Communication Parameters* dialog.

4. In the *Communication Settings* dialog, click the **Advanced** button to open the *Advanced Settings* dialog:


Advanced settings

Timeout (ms)
Start message: 1000
End message: 0
Interval between char: 500
Wait CTS: 100

Disable DTR
Enable IR

Protocol
Station:
Retries: 0

Handshake
Control RTS: no
Verify CTS: no

Buffers length (bytes)
Tx Buffer: 512
Rx Buffer: 512

Advanced Settings Dialog

When the dialog is displayed, configure the Buffer Length to match with the **Block Size** parameter. You can also configure the timeout time in milliseconds as well as a number of retries.

You can consult the Studio *Technical Reference Manual* later for more information about configuring these settings.

5. Click **OK** to close the *Advanced Settings* dialog, and then click **OK** to close the *Communication Settings* dialog.

<u>Configuring the Driver Worksheets</u>

Each selected driver includes a Main Driver Sheet and one or more Standard Driver Worksheets. The Main Driver Sheet is used to define tag/register associations and driver parameters that are in effect at all times, regardless of application behavior. In contrast, Standard Driver Worksheets can be inserted to define additional tag/register associations that are triggered by specific application behaviors.

The configuration of these worksheets is described in detail in the "Communication" chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only TSAA driver-specific parameters and procedures are discussed here.

MAIN DRIVER SHEET

This driver version does not support Main Driver Sheet.

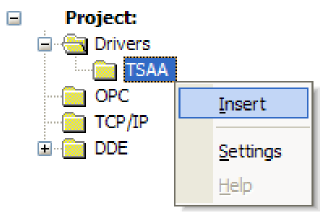
STANDARD DRIVER WORKSHEET

When you select the TSAA driver and add it to your application, it has only a Main Driver Sheet by default (see previous section). However, you may insert additional Standard Driver Worksheets to define tag/register associations that are triggered by specific application behaviors. Doing this will optimize communication and improve system performance by ensuring that tags/registers are scanned only when necessary – that is, only when the application is performing an action that requires reading or writing to those specific tags/registers.

Note:
We recommend configuring device registers in sequential blocks in order to maximize performance.

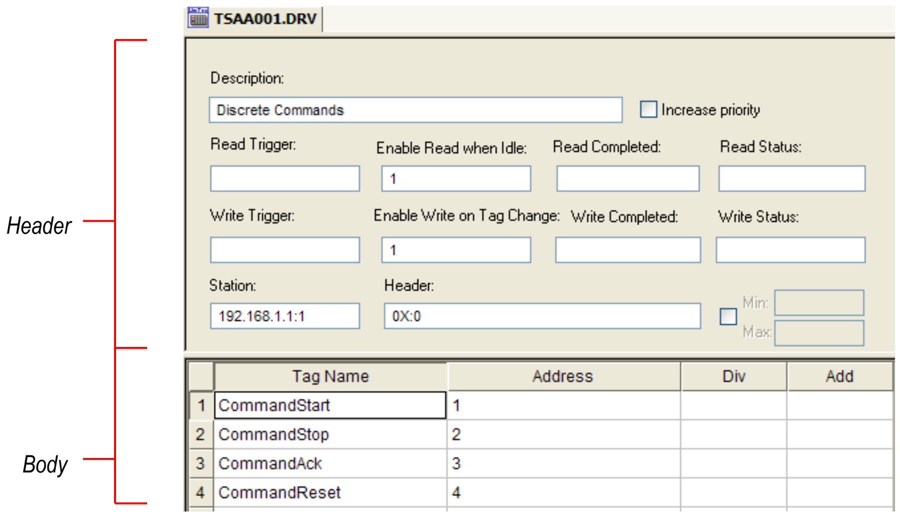
To insert a new Standard Driver Worksheet:

1. In the *Comm* tab, open the *Drivers* folder and locate the *TSAA* subfolder.
2. Right-click on the *TSAA* subfolder, and then select **Insert** from the pop-up menu:



Inserting a New Worksheet

A new TSAA driver worksheet is inserted into the *TSAA* subfolder, and the worksheet is opened for configuration:



TSAA001.DRV

Description: Discrete Commands
Increase priority: ☐
Read Trigger:
Enable Read when Idle: 1
Read Completed:
Read Status:
Write Trigger:
Enable Write on Tag Change: 1
Write Completed:
Write Status:
Station: 192.168.1.1:1
Header: 0X:0
Min:
Max:

	Tag Name	Address	Div	Add
1	CommandStart	1		
2	CommandStop	2		
3	CommandAck	3		
4	CommandReset	4		

TSAA Driver Worksheet

> **Note:**
> Worksheets are numbered in order of creation, so the first worksheet is **`TSAA001.drv`**.

Most of the fields on this worksheet are standard for all drivers; see the "Communication" chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station**, **Header**, and **Address** fields use syntax that is specific to the TSAA driver.

3. Configure the **Station** and **Header** fields as follows:

- **Station** field: Specify the IP Address of the device and the slot number, using the following syntax:

  `<IP Address>:<PLC ID>:<optional Port Number>`

  Example — **`192.168.1.1:1`** or **`192.168.1.1:1:1502`**

  Where:

  – ***`<IP Address>`*** is the device's IP address on the Ethernet network;

  – ***`<PLC ID>`*** is the PLC identification number (from 1 to 31); and

  – ***`<Port Number>`*** is the UDP port number for the TSAA protocol (usually 1500).

  In this field, you can also specify an string tag (e.g. **`{station}`**). Keep in mind that the value of the tag that is referenced must follow the same syntax and contain a valid value.

  Example: String Tag **Station** = "192.168.1.1:1"

> **Attention:**
> - You must use a non-zero value in the **Station** field, and you cannot leave the field blank.
> - TSAA operands must start in an address that is greater than zero.

- **Header** field: Specify the address of the first register of a block of registers on the target device. The addresses declared in the *Body* of the worksheet are simply offsets of this **Header** address. When Read/Write operations are executed for the entire worksheet (see **Read Trigger** and **Write Trigger** above), it scans the entire block of registers from the first address to the last.

  The **Header** field uses the following syntax:

  ***`<Type>:0 or (`*`ReadSymbolTable or RST)`**

  Example — **`4X:0`**

  Where:

  – ***`<Type>`*** is the register type (**`0X`**, **`1X`**, **`3X`**, **`4X`**).

  – ***`0 –`*** simply used to complement the header. Although you can type other numbers there, please make it always **0**

  After you edit the **Header** field, Studio checks the syntax to determine if it is valid. If the syntax is invalid, then Studio automatically inserts a default value of **`0X:0`**.

  In this field, you can also specify an string tag (e.g. **`{header}`**), Keep in mind that the value of the tag that is referenced must follow the same syntax and contain a valid value.

  Example: String tag **DriverHeader** = "4X:0"

The following table lists all of the data types and address ranges that are valid for the TSAA driver:

Type	Syntax	Valid Address Range	Comments (Note: the aliases range below applies only to the Tricon family, since Trident PLCs allow custom ranges)
0X	**`0X:0`**	1 – 4000	Coil Status: Read and write the following TSAA BIN types • BIN_DISCRETE_OUTPUT – Aliases 1 to 2000 • BIN_MEMORY_DISCRETE_RW – Aliases 2001 to 4000
1X	**`1X:0`**	1-9999	Input Status: Read the following TSAA BIN Types • BIN_DISCRETE_INPUT – Aliases 10001 to 12000 • BIN_MEMORY_DISCRETE_RO – Aliases 12001 to 14000 • BIN_SYSTEM_STATUS_DISCRETE – Aliases 14001 – 19999
3X	**`3X:0`**	1 – 9999	Input Register: Read the following TSAA BIN Types • BIN_ANALOG_INPUT – Aliases 30001 to 31000 • BIN_MEMORY_INTEGER_RO – Aliases 31001 to 31382 • BIN_REAL_INPUT – Aliases 32001 to 32120 • BIN_MEMORY_REAL_RO – Aliases 33001 to 34000 • BIN_SYSTEM_STATUS_INTEGER – Aliases 39631 – 39999
4X	**`4X:0`**	1 – 2000	Holding Register: Read and write the following TSAA BIN Types: • BIN_ANALOG_OUTPUT – Aliases 40001 to 40250 • BIN_MEMORY_INTEGER_RW – Aliases 40251 to 40632 • BIN_MEMORY_REAL_RW – Aliases 41001 to 42000
RST or Read Symbol Table	**`ReadSymbol Table or RST`**	Not Applicable	This command can be used to force Reading the entire Symbol Table from a Trident PLC. It is particularly useful when the PLC program is being modified constantly, generating new Modbus Aliases and BIN/OFFSET allocations for them.
Dump Symbol Table to Text File	**`DumpSymbol Table`**	Not Applicable	This command stores the contents of the symbol table to a file. The file name is specified by a string tag which should be configured in the first line of the driver worksheet. The tag should have a full path (e.g.: C:\Symbol.txt)

4. For each table row (i.e., each tag/register association), configure the **Address** field using the following syntax…

   For all register types use the following syntax:

   <Address Offset>

   Examples — `1, 2, 10, 1001`

Note:

For hear type **ReadSymbolTable** or **RST,** simply configure **0** on the first ***Address*** , and there is no need to configure anything on the Tag Name column. Use a tag in the **Read Trigger** field to force this command execution

Where:

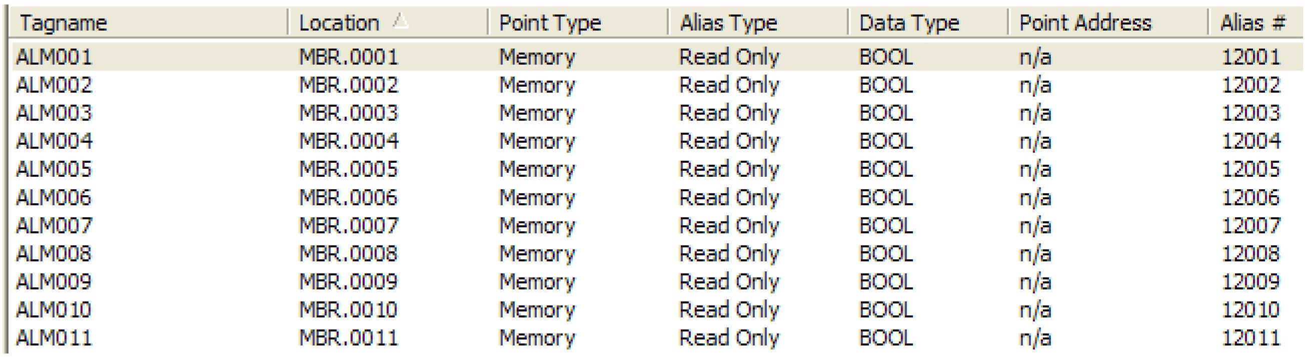
– ***<Address Offset>***: Parameter that is added to the **Header**, to compose the specific address of the register in the block. TSAA operands must start in an address that is greater than zero. For example, for the PLC address **41001**, the Header would be **4x:0** and the Address Offset **1001.**

Attention:

1. Keep in mind that when using the **Write Trigger** feature, the driver writes to the entire block of registers from the first address through the last. If there is a register that has not been declared in the worksheet, and its address is within the block, then the register will receive a zero (0) value. Check the worksheet for holes in the address range.

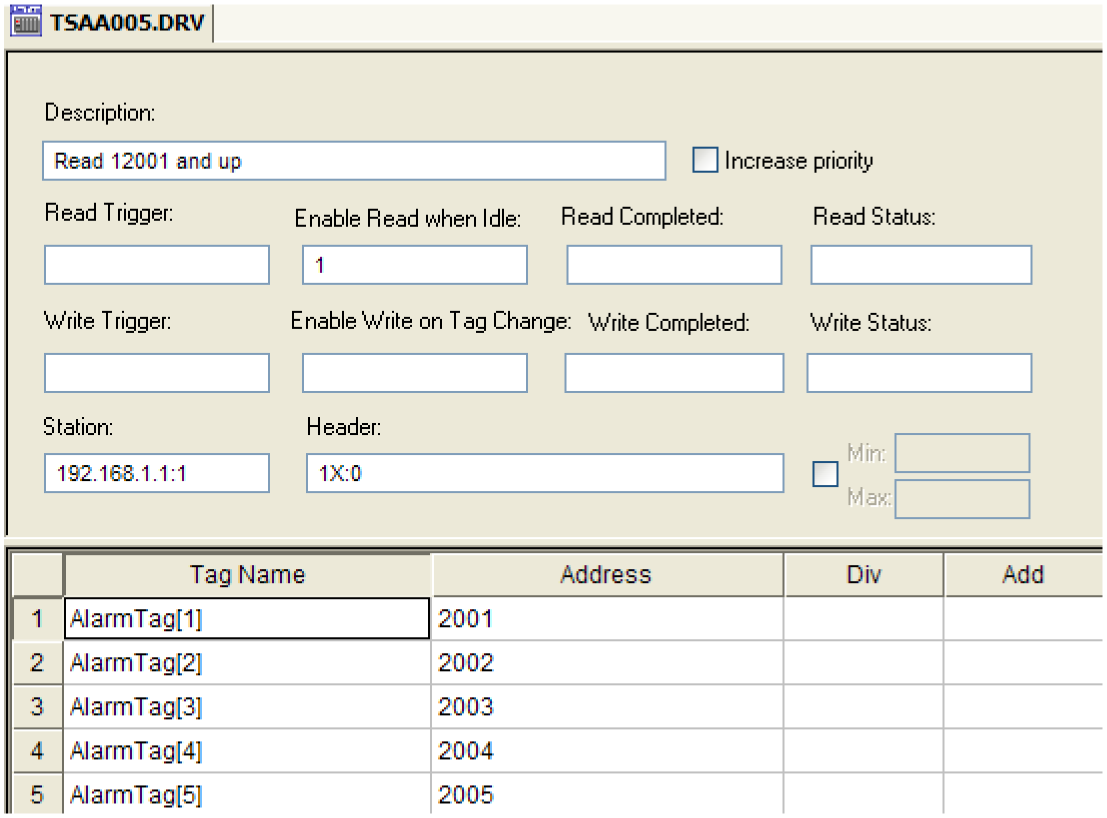
Example:

1. Let's say that on the PLC you have a configuration like this:



Tagname	Location	Point Type	Alias Type	Data Type	Point Address	Alias #
ALM001	MBR.0001	Memory	Read Only	BOOL	n/a	12001
ALM002	MBR.0002	Memory	Read Only	BOOL	n/a	12002
ALM003	MBR.0003	Memory	Read Only	BOOL	n/a	12003
ALM004	MBR.0004	Memory	Read Only	BOOL	n/a	12004
ALM005	MBR.0005	Memory	Read Only	BOOL	n/a	12005
ALM006	MBR.0006	Memory	Read Only	BOOL	n/a	12006
ALM007	MBR.0007	Memory	Read Only	BOOL	n/a	12007
ALM008	MBR.0008	Memory	Read Only	BOOL	n/a	12008
ALM009	MBR.0009	Memory	Read Only	BOOL	n/a	12009
ALM010	MBR.0010	Memory	Read Only	BOOL	n/a	12010
ALM011	MBR.0011	Memory	Read Only	BOOL	n/a	12011

You should configure the driver sheet like this:



TSAA005.DRV

Description: Read 12001 and up

Increase priority: ☐

Read Trigger:
Enable Read when Idle: 1
Read Completed:
Read Status:

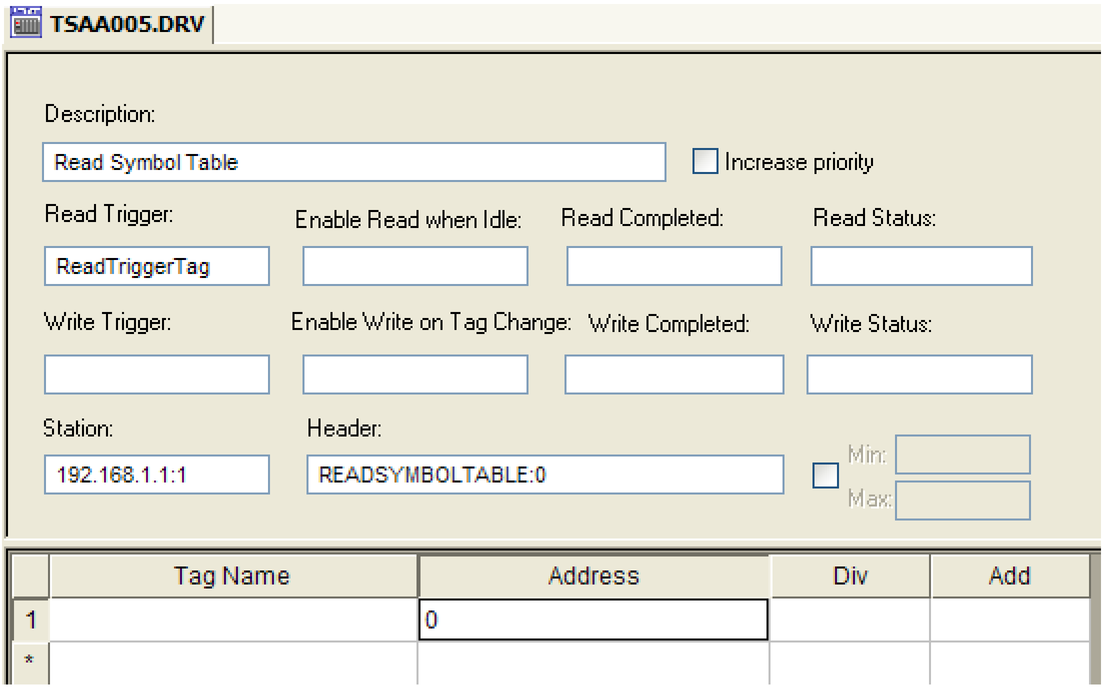
Write Trigger:
Enable Write on Tag Change:
Write Completed:
Write Status:

Station: 192.168.1.1:1
Header: 1X:0

☐ Min:
Max:

	Tag Name	Address	Div	Add
1	AlarmTag[1]	2001		
2	AlarmTag[2]	2002		
3	AlarmTag[3]	2003		
4	AlarmTag[4]	2004		
5	AlarmTag[5]	2005		

2. Here is an example about how to force reading the Symbol Table when using Trident PLCs



TSAA005.DRV

Description: Read Symbol Table
Increase priority: ☐

Read Trigger:	Enable Read when Idle:	Read Completed:	Read Status:
ReadTriggerTag			
Write Trigger:	Enable Write on Tag Change:	Write Completed:	Write Status:

Station: 192.168.1.1:1
Header: READSYMBOLTABLE:0
☐ Min: Max:

	Tag Name	Address	Div	Add
1		0		
*				

For more examples of how device registers are specified using **Header** and **Address**, see the following table:

PLC Tag Alias Number	Header	Address	Remarks
1	`0x:0`	`1`	`Discrete Output`
10	`0x:0`	`10`	`Discrete Output`
2020	`0x:0`	`2020`	`Discrete Memory Read/Write`
10001	`1x:0`	`1`	`Discrete Input (Read Only)`
10010	`1x:0`	`10`	`Discrete Input (Read Only)`
11101	`1x:0`	`1101`	`Discrete Memory Read Only`
30001	`3x:0`	`1`	`Analog Input (Read Only)`
30010	`3x:0`	`10`	`Analog Input (Read Only)`
31020	`3x:0`	`1020`	`Memory Integer Read Only`
33301	`3x:0`	`3301`	`Memory Real Read Only`
40001	`4x:0`	`1`	`Analog Output`
40010	`4x:0`	`10`	`Analog Output`
40251	`4x:0`	`251`	`Memory Integer Read/Write`
41010	`4x:0`	`1010`	`Memory Real Read/Write`

For more information about the device registers and addressing, please consult the manufacturer's documentation.

➲ **Attention:**

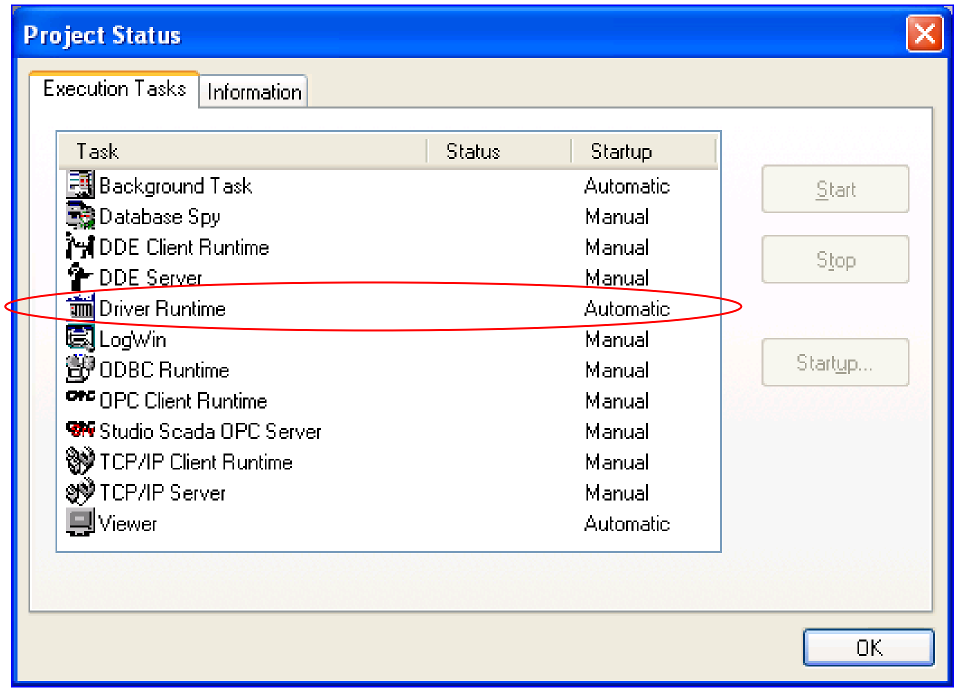
You must not configure a range of addresses greater than the maximum block size (data buffer length) supported by the target device. The default block size is 512 bytes, but this can be changed in the communication settings for the driver.

Executing the Driver

By default, Studio will automatically execute your selected communication driver(s) during application runtime. However, you may verify your application's runtime execution settings by checking the *Project Status* dialog.

To verify that the communication driver(s) will execute correctly:

1. From the main menu bar, select **Project** → **Status**. The *Project Status* dialog displays:



Task	Status	Startup
Background Task		Automatic
Database Spy		Manual
DDE Client Runtime		Manual
DDE Server		Manual
Driver Runtime		Automatic
LogWin		Manual
ODBC Runtime		Manual
OPC Client Runtime		Manual
Studio Scada OPC Server		Manual
TCP/IP Client Runtime		Manual
TCP/IP Server		Manual
Viewer		Automatic

Project Status Dialog

2. Verify that the *Driver Runtime* task is set to **Automatic**.
   - If the setting is correct, then proceed to step 3 below.
   - If the **Driver Runtime** task is set to **Manual**, then select the task and click the **Startup** button to toggle the task's *Startup* mode to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.
4. Start the application to run the driver.

Troubleshooting

If the TSAA driver fails to communicate with the target device, then the database tag(s) that you configured for the **Read Status** or **Write Status** fields of the Main Driver Sheet will receive an error code. Use this error code and the following table to identify what kind of failure occurred.

Driver error codes

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communication without problems	None required
-57	Message could not be sent	The computer is cannot reach the PLC network	- Check if your IP address is configured correctly - If the PLC is not in the same network as your studio computer, check if you have a gateway properly configured to provide access to the PLC network.
-15	Timeout	Time out while waiting for a PLC response	- Check if the PLC is on-line - Check if the cable is connected correctly - Try to ping the PLC - Confirm that the Slave ID configured on the PLC is the as the one configured on the *Station* field
1	All symbols are invalid	None of the variables specified in the driver worksheet are valid	Check if the PLC program and make sure that you have the correct symbol. If the worksheet has at least one valid symbol, the driver will read it and the other ones will be shown as an error in the Protocol Analyzer.
2	Invalid Response	The PLC responded with an invalid value for one or more of the protocol parameters.	Check if you are connected to a valid device. If the problem persists, you might be connected to a PLC with a firmware version that is not compatible with this driver, please contact your technical support to verify that.
3	PLC Returned an error code	The PLC responded the request with an error code	Enable the protocol analyzer to get more details, the protocol analyzer will show the “Frame Error” as well as the”Frame Error – Sub Reason”. Check the list of “Frame Errors” for more information. The protocol documentation does not have description about the sub reasons, you might have to contact the PLC manufacturer technical support if you need more information.
4	Invalid response to read command	The PLC returned status OK, but the frame has no data bytes	Contact your Studio technical support with the communication logs.
5	Communication channel could not be established	- The firewall is blocking the communication port - The cable is not connected between the computer and the PLC	- Try to ping the PLC IP address to make sure that the PLC is on line. - Check if the port 1500 is open on your firewall.
6	Error to allocate device list	The computer does not have enough memory to store the device list and the driver internal buffers	Open the task manager and check the amount of RAM memory available, you might have to increase the memory available on this computer.

Error Code	Description	Possible Causes	Procedure to Solve
7	The frame received is too big	An updated in the PLC firmware might have changed the PLC behavior and the PLC could be sending more data than the maximum supported when the driver was designed	Split your worksheets into smaller groups to reduce the amount of data being requested. Also notify the Studio technical support, providing the PLC firmware version.
8	Error to allocate device	The computer does not have enough memory to store the device buffers	Open the task manager and check the amount of RAM memory available, you might have to increase the memory available on this computer.
9	Invalid frame size	The number of bytes received exceeded the message length field specified in the protocol length field. This would happen if the PLC firmware was changed and a new protocol structure was implemented	Contact your Studio technical support with the communication logs.
9	Invalid frame size	The number of bytes received exceeded the message length field specified in the protocol length field. This would happen if the PLC firmware was changed and a new protocol structure was implemented	Contact your Studio technical support with the communication logs.
10	None of the responses received by the driver matches the request	The PLC has sent several responses after the request, but none of them match the driver expected response. if you have a value lower than 100ms in the time out and a lot of traffic on the network this problem could happen quite often	Increase your time out configuration.
11	Symbol Table Frame received has no entries	The PLC responded to a symbol table request, but the response does not have any entries	Contact thyoure Studio technical support with the communication logs.
12	Symbol table version changed while downloading	While retrieving the symbol table from the PLC a new PLC program was sent using the programming software. This changed the symbol table version.	This error is temporary, in the next request the driver will download the symbol table again and the problem will be resolved.
13	All symbols are invalid	None of the Aliases specified in your driver worksheet exist in the PLC symbol table.	Check your aliases and make sure that you have tags configured in the PLC with them.
14	Internal Error	There was a programming error inside the driver source code	Contact your Studio technical support with the communication logs.
15	Invalid CRC	An invalid CRC was sent by the PLC	If the problem persists, contact your Studio technical support with the communication logs.
16	Invalid Command	It is not possible to use write commands with the READSYMBOLTABLE header and read commands with the DUMPSYMBOLTABLE headers	Remove the triggers from the worksheet
17	Invalid Data	When using the DUMPSYMBOLTABLE the user specified a Non-String tag	Modify the tag in the worksheet to a valid string tag
18	Error to open dump file	The file name specified is invalid, the or the application does not have the access rights to it	
19	Symbol table cannot be dumped	The symbol table is empty and therefore cannot be dumped	Make sure that the PLC has a valid program with a symbol table.
20	PLC Not supported	The PLC configuration specified in the driver settings is not supported	Contact the technical support for more information on the supported devices.

Driver frame error codes

If you receive the error code #3 from the previous table, enable the Protocol Analyzer option, you will see a frame error code, which should match one of the following errors. This table is provided by the protocol documentation.

Error Code	Description
0	Request was successful
1	No buffer available to process the request. Retry the request
2	Bin number specified in the request was not in the range from 0 to12
3	The Tricon NCM or Trident CM module is busy processing previous requests and cannot accept another request. This can happen if more than four WRITE_TRICON_DATA requests are outstanding at one time.
4	No MP is running
5	TSX has rejected the request. The subReason field contains the specific Reason
6	Request to TSX timed out
7	Invalid response from TX.
8	Message was too big.
9	Offset or numberOfValues in the request was invalid
10	No control program (TriStation application)
11	Read-only port
236	Bad SOE number
237	Invalid SOE type
238	Invalid SOE state.

➪ **Tip:**

You can monitor communication status by establishing an event log in Studio's *Output* window (*LogWin* module). To establish a log for **Field Read Commands**, **Field Write Commands** and **Protocol Analyzer,** right-click in the *Output* window and select the desired options from the pop-up menu.

You can also use the *Remote LogWin* module (**Tools** → **Remote LogWin**) to establish an event log on a remote unit

If you are unable to establish communication between Studio and the target device, then try instead to establish communication using the device's own programming software (e.g., ModSoft). Quite often, communication is interrupted by a hardware or cable problem or by a device configuration error. If you can successfully communicate using the programming software, then recheck the driver's communication settings in Studio.

To test communication between Studio and the device, we recommend using the sample application provided rather than your new application.

If you must contact us for technical support, please have the following information available:

- **Operating System** (type and version): To find this information, select **Tools** → **System Information**.
- **Project Information**: To find this information, select **Project** → **Status.**
- **Driver Version** and **Communication Log**: Displays in the Studio *Output* window when the driver is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of changes
A	1.00	Eric Vigiani	Aug/2007	First Driver version
B	1.01	Graziane C. Forti	Feb/2008	Modified to work with TRIDENT device
C	1.02	Eric Vigiani	May/2008	Modified to support Symbol Table for TRIDENT PLCs
D	1.03	Eric Vigiani	Jun/2008	Implemented protection to avoid wrong messages
E	2.0	Lourenço Teodoro	July/2010	Implemented Symbol Table expiration Reviewed the driver error codes Implemented DumpSymbolTable command
F	2.1	Paulo Balbino	Jun/2012	Released Driver
G	2.2	Anushree Phanse	Dec/2015	Fixed communication problems with the emulator.Increased the UDP packet size. Added verification for missing frames in read responses.