CSTAS Communication Driver

Cotsco AS400 Warehouse Systems

Contents

Introduction

The CSTAS driver enables communication between Studio and an AS400 client, using a simple protocol. The CSTAS driver is a TCP/IP server which listens to the configured port, waiting for a client to connect and, after this happens, it sends an initial message by default “Costco” and a big message with all port status at each specified period of time.

> **Important (About TCP/IP Server capabilities):**
> This driver (server) supports only one connection at a time. If a connection between a client and the server exists, and another client connects to the same server, the previous client will be disconnected

This document will help you to select, configure and execute the CSTAS 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 CSTAS driver in the Studio system.
- **Configuring the Driver**: Explains how to configure the CSTAS driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver**: Explains how to execute the CSTAS 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 CSTAS 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 XP/7 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 CSTAS driver in Studio and remote devices.

The information is organized into the following sections:

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

Device Specifications

You can use this driver to communicate with any device using a simple ASCII protocol. (The devices used for conformance testing are listed on the next page.)

Network Specifications

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

- **Device Communication Port**: Ethernet port
- **Physical Protocol**: TCP/IP
- **Logic Protocol**: ASCII
- **Device Runtime Software**: None
- **Specific PC Board**: None
- **Adapters/Converters**: None
- **Cable Wiring Scheme**: None

Driver Characteristics

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

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

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

- Windows XP/7, Embedded
- Windows CE 5.x, 6.x

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

Conformance Testing

The following hardware/software was used for conformance testing:

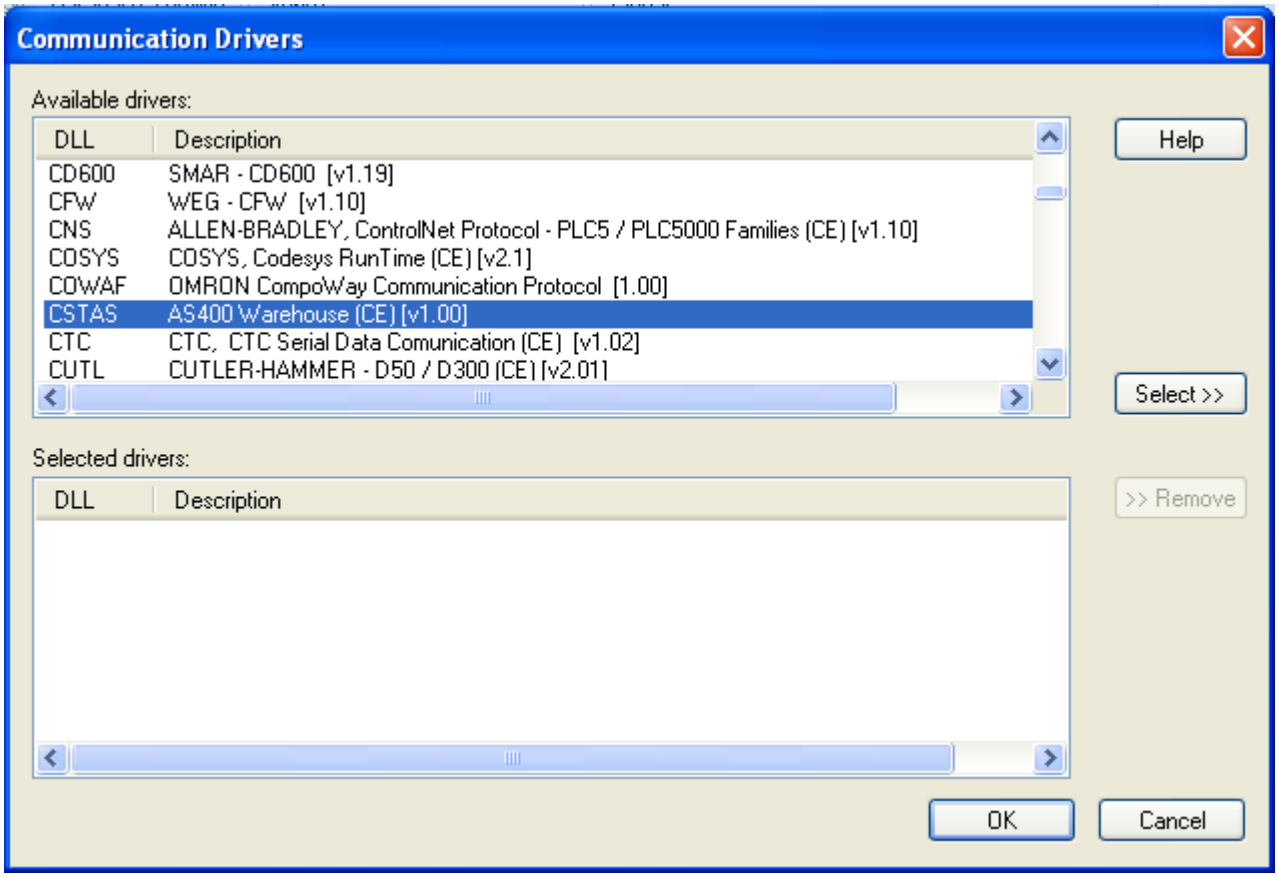
- **Driver Configuration (a)**:
  - **Protocol**: ASCII
  - **TCP/IP Port**: 5050
  - **Cable**: Ethernet Cable

Driver Version	Studio Version	Operating System	Equipment
1.0	6.1+SP5	Windows XP SP3	– Two PCs connected with a TCP/IP connection

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 CSTAS driver for your Studio application:

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



Communication Drivers Dialog

3. When the **CSTAS** 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:**
> CSTAS driver is used as a server, to enable connections it is necessary to install an external software that uses the AS400 protocol and make it connect to Studio.

> **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 Driver

Once you have selected the CSTAS driver in Studio, you must properly configure it to communicate with your target device.

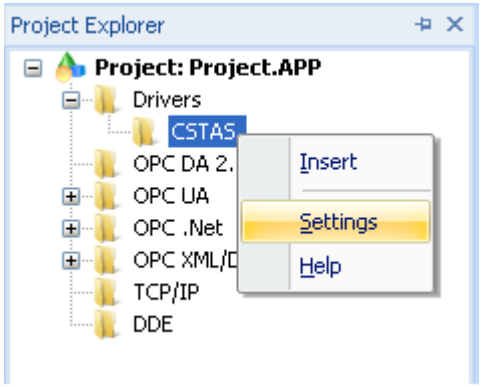
Configuring the Communication Settings

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 CSTAS driver-specific settings and procedures will be discussed here. To configure the communication settings for the CSTAS driver:

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

2. Right-click on the CSTAS subfolder and then select the **Settings** option from the pop-up menu. The CSTAS*: Communication Parameters* dialog is displayed:



Select Settings from the Pop-Up Menu

CSTAS: Communication Parameters Dialog

3. Configure the additional driver-specific settings, as described in the following table:

Setting	Default Value	Valid Values	Description
TCP/IP Port	`5050`	Integer value	TCP/IP port number used to receive messages from other clients. The default is 5050
Initial Message	**"Costco"**	`String`	It is a initial message, it will be sent after a connection is accepted.

4. Click **OK** to close the *Communication Settings* dialog.

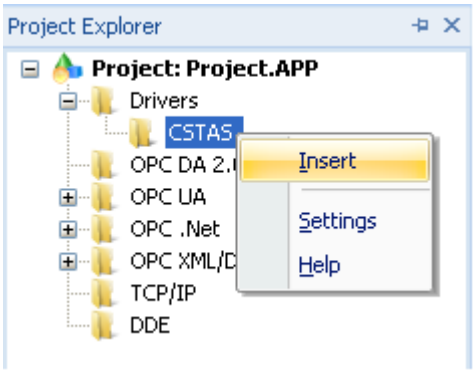
Configuring the Driver Worksheets

A selected driver includes one or more driver worksheets, which are used to associate database tags in Studio with operands on the target device. Each worksheet is triggered by specific application behavior, so that the tags / operands defined on that worksheet are scanned only when necessary – that is, only when the application is doing something that requires reading from or writing to those specific tags / operands. Doing this optimizes communication and improves system performance.

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.

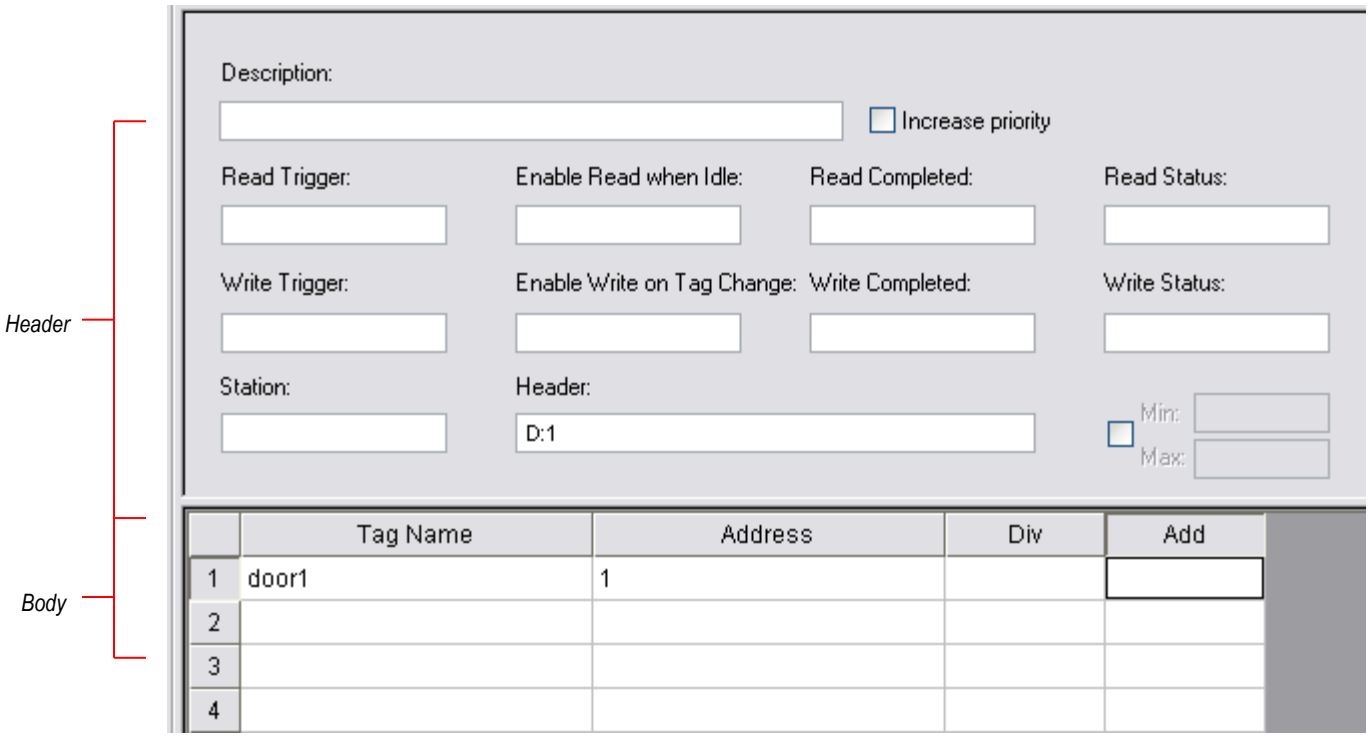
To insert a new driver worksheet:

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



Inserting a New Worksheet

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



CSTAS Driver Worksheet

> **Note:**
> Worksheets are numbered in order of creation, so the first worksheet is **`CSTAS001.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 **Header**, and **Address** fields use syntax that is specific to the CSTAS driver.

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

   - **Station** field (*not used*)
   - **Header** field: Define the specific building for sending the door status

     Type of Header:

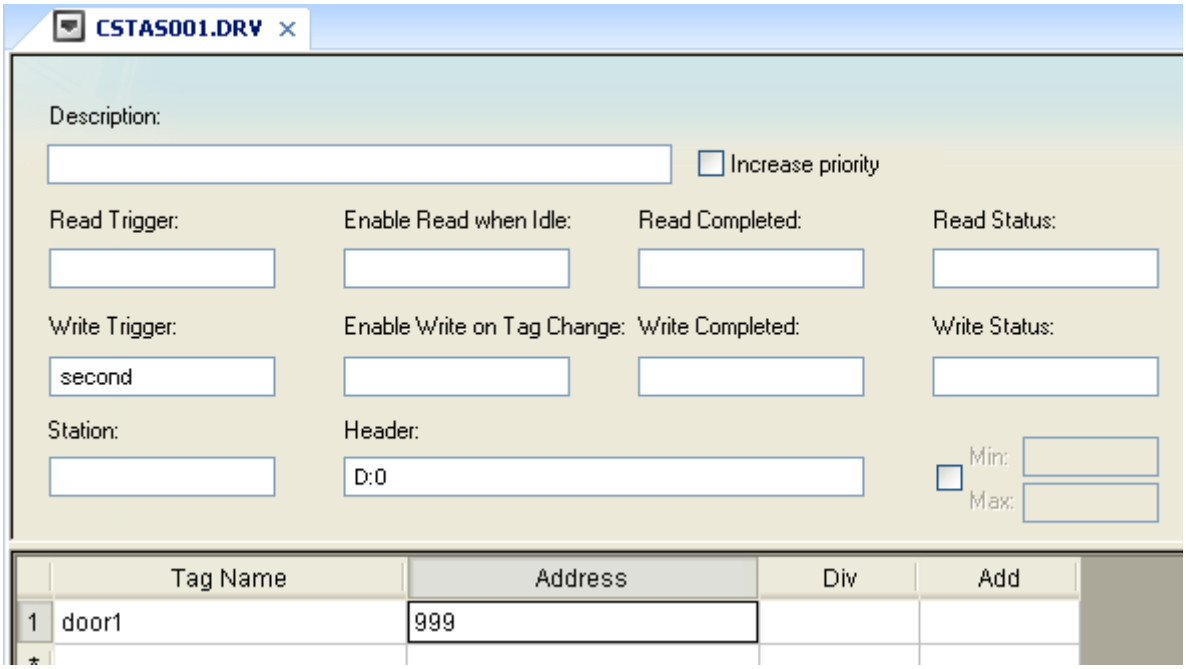
     - **`D:<Building Number>`** : Writes the doors status for a specific building after a period of time (1 minute) as default.

> **Note:**
> Station is **not used** on this driver! The selection of port for listen is made on communication parameters.

> **Note:**
> There is **no Read** operation on this driver!

Sample of configuration

Header Field	Address Field
D:1	1
D:2	10
D:1	142
D:2	999



Sample of configuration

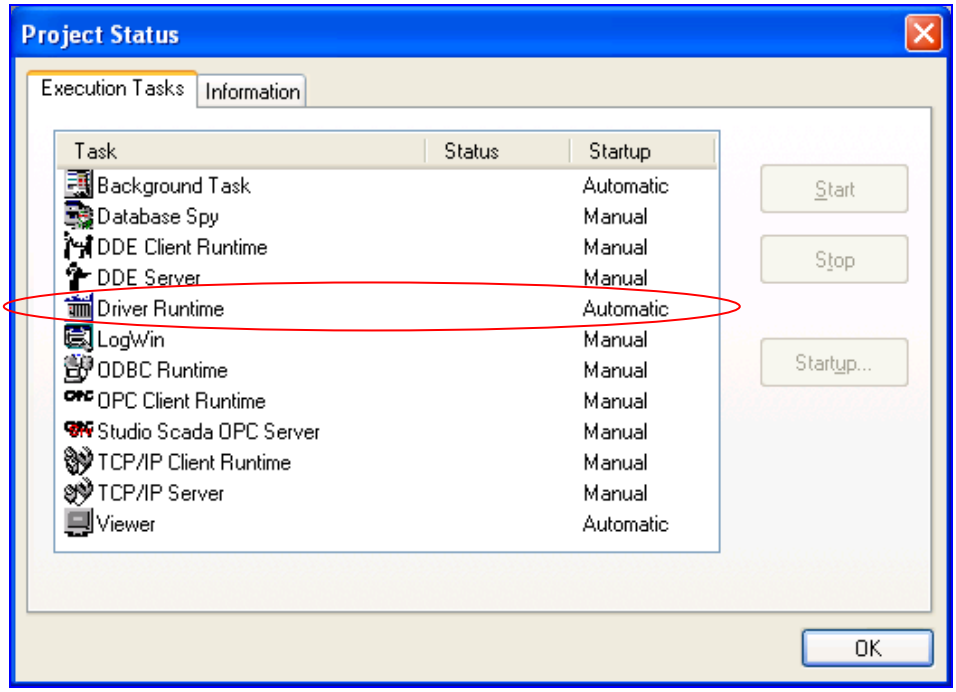
> **Note:**
> This example writes at each second, by default following the protocol it should be 1 minute.

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:



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 CSTAS 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 Standard Driver Sheet will receive an error code. Use this error code and the following table to identify what kind of failure occurred.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communicating without problems	Not required
10	Invalid door status	The value of a Tag is not valid	The value should be (-1 or 0 or 1)
11	No connection	No client connected	Make a connection with the client, check the port configuration on the client and on Indusoft.
12	Error sending message	Error while communicating	Check ports and check network functionality.
13	Error invalid address	The address type is not valid.	The value should be 1 to 999

⇨ **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 *LogWin* module to establish an event log on a remote Runtime, even the ones running Windows CE.

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. 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.

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.

Sample Application

This driver does not have a sample application

Revision History

Doc. Revision	Driver Version	Author	Date	Description of Changes
A	1.0	Paulo R. Balbino	8 Sep 2009	Initial version
B	1.0	Lucas Caccavaro	29 Aug 2011	Documentation changes only