TAGW

Contents

TAGW Driver

TAGW Driver for SoftPLC TagWell (version 1.3., last revised 23 March 2018).

The TAGW driver enables communication between the Studio system and remote devices using the protocol, according to the specifications discussed in this document.

This document assumes that you have read the "Development Environment" section in the main Studio documentation.

This document also assumes that you are familiar with the Microsoft Windows XP/Vista/7 environment. If you are not familiar with Windows, then we suggest using the **Help and Support** feature (available from the Windows **Start** menu) as you work through this document.

Driver specifications

This section identifies all of the software and hardware components required to implement communication between the TAGW driver in Studio and remote devices using the protocol.

Driver files

The TAGW driver package comprises the following files, which are automatically installed in the `Drv` folder of the Studio application directory:

- `TAGW.DLL`: Compiled driver.
- `TAGW.INI`: Internal driver file. *You must not modify this file.*
- `TAGW.MSG`: Internal driver file defining error messages for the possible error codes. (These error codes are described in detail in the Troubleshooting section.) *You must not modify this file.*
- `TAGW.PDF`: This document, which provides complete information about using the driver.

> **Note:** You must use a compatible PDF reader to view the `TAGW.PDF` file. You can install Acrobat Reader from the Studio installation CD, or you can download it from Adobe's website.

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

- Windows 7/8/2008/2012

Device specifications

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

- Manufacturer: SoftPLC
- Compatible Equipment: TagWell Cloud
- Programmer Software:

Network specifications

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

- Device Communication Port:
- Physical Protocol: HTTP
- Logic Protocol:
- Device Runtime Software: None
- Specific PC Board: None
- Cable Wiring Scheme: Regular Ethernet cable

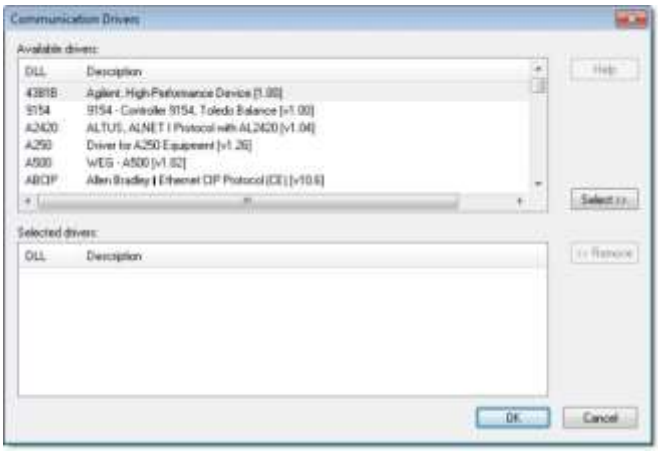
Additional specifications

This driver also requires that the library file libcurl.dll be properly installed in the Bin sub-folder of the Studio program folder, at Bin\libcurl.dll. If you downloaded and installed this driver separately, some time after you installed Studio, the library file should be automatically installed in the correct location. If it is not, or if it is not correctly named, this driver will not initialize. Note that versions before TAGW v1.2 required this file to be installed in the Drv subfolder API of the Studio Program folder in Drv\API\libcurl.dll

Adding a communication driver to your project

This section explains how to add a communication driver to your project.

1. On the **Insert** tab of the ribbon, in the **Communication** group, click **Add/Remove Driver**.
   The *Communication Drivers* dialog is displayed.



Communication Drivers dialog

2. In the *Available drivers* list, click the communication driver that you want to add.
3. Click **Select**.
   The driver is added to the *Selected drivers* list.
4. Click **OK**.
   The *Communication Drivers* dialog is closed and the selected driver is inserted in the **Drivers** folder in the Project Explorer.

Configuring the driver's communication settings

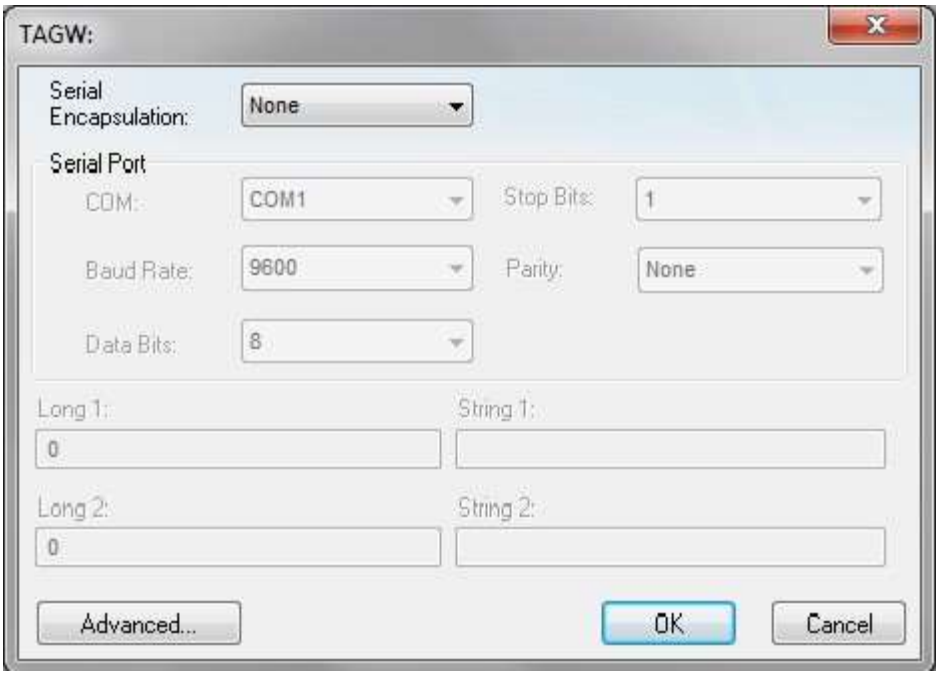
This section explains how to configure the communication settings for the driver.

You must add the communication driver to your project before you can configure its settings. For more information, see Adding a communication driver to your project on page 6.

The general procedure for configuring a driver's communication settings is the same for all drivers. However, the specific settings are different for each driver, depending on the options and protocols used by the target device.

To configure the communication settings:

1. In the **Comm** tab of the Project Explorer, expand the **Drivers** folder.
   The folder contains the drivers that are currently enabled. If you do not see the driver that you want to configure, then you need to add it.
2. Right-click the driver that you want to configure, and then click **Settings** on the shortcut menu.
   The *Communication Settings* dialog is displayed.



Communication Settings: TAGW dialog

3. Configure the remaining, driver-specific settings as needed.

Driver-specific communication settings

Setting	Default Value	Valid Values	Description

4. Click **OK**.
The settings are saved and the *Communication Settings* dialog is closed.

About driver worksheets

Like the other parts of your project, communication with remote devices is controlled by worksheets. This section explains how to add worksheets to your project and then configure them to associate project tags with device registers.

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

The configuration of these worksheets is described in detail in the "Communication" chapter of the *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 TAGW driver-specific parameters and procedures are discussed here.

Adding and configuring a Standard Driver Sheet

By default, a communication driver does not include any Standard Driver Sheets. This section explains how to add a Standard Driver Sheet to your project and then configure it.

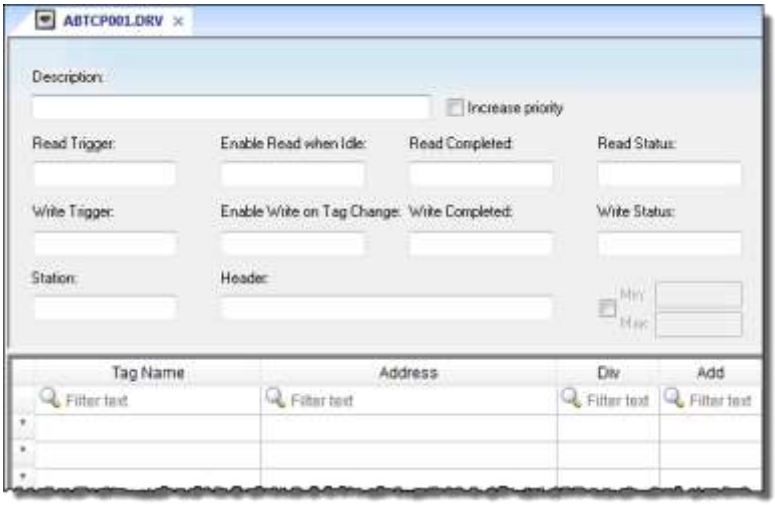
The TAGW driver must be added to the project before you can configure any of its worksheets. For more information, see Adding a communication driver to your project on page 6.

Standard Driver Sheets can be inserted to define additional tag/register associations that are triggered by specific project behaviors.

> **Note:** Most of the settings on this worksheet are standard for all drivers; for more information about configuring these settings, see the "Communication" chapter of the *Technical Reference Manual*. The **Station** and **I/O Address** fields, however, use syntax that is specific to the TAGW driver.

1. Do one of the following.
   - On the **Insert** tab of the ribbon, in the **Communication** group, click **Driver Sheet** and then select **TAGW** from the list.
   - In the **Comm** tab of the Project Explorer, right-click the **TAGW** folder and click **Insert** on the shortcut menu.

   A new TAGW driver worksheet is inserted into the **TAGW** folder, and then it is automatically opened for configuring.



Standard Driver Sheet

> **Note:** Worksheets are numbered in order of creation, so the first worksheet is `TAGW001.drv`.

2. Configure the Station and Header fields as described below.

Station

Specify the cloud URL and the custom directories along with the user name and password in the station in the driver sheet using the following syntax. Note that this field cannot be left empty. Examples are given in the table below.

[*URL*] / **[*CUSTOMDIR*] / [*REMOTEDIR*] :[*UserName*] :[*Password*]** Where:

[*URL*] This is the URL of the cloud to be accessed, i.e. the Base Directory. For eg. http:// tagwell.net.

[*CUSTOMDIR*] This is the name of the custom directory to be accessed on the TagWell cloud. This will be appended to the Base Directory in the URL.

Eg: SoftPLC

[*REMOTEDIR*] This is the remote directory to be accessed on the TagWell cloud inside the custom directory. This will be appended to the URL.

Eg: "Demo"

[*UserName*] This is the user name of the remote login. The request will be sent only when the user is authenticated on the cloud.

[*Password*] This is the password of the user logging on the cloud.

You can also specify a tag in curly brackets to change the station during the runtime (e.g. `{Station}`), but the tag that is referenced must follow the same syntax and contain a valid value.

Station Formats

Station	Examples
URL/**[*CUSTOMDIR*]/[*REMOTEDIR*]:[*UserName*]:[*Password*]**	http://tagwell.net/SoftPLC/Demo:username:password

Header

The header field is not in use by this driver. Note that NAME is an internal header used.

3. For each tag/register association that you want to create, insert a row in the worksheet body and then configure the row's fields as described below.

Tag Name

Type the name of the project tag.

Address

Specify the tagname you want to communicate with.

tagname

Where:

tagname is the name of the tag to communicate on the TagWell cloud.

> **Note:** NOTE THAT THE TAGNAMES ARE CASE-SENSITIVE. Tags with incorrect spelling or case will result in errors.

> **Note:** Each Standard Driver Sheet can have up to 4096 rows. However, the **Read Trigger**, **Enable Read When Idle**, and **Write Trigger** commands attempt to communicate the entire block of addresses that is configured in the sheet, so if the block of addresses is larger than the maximum block size that is supported by the driver protocol, then you will receive a communication error (e.g., "invalid block size") during run time. Therefore, the maximum block size imposes a practical limit on the number of rows in the sheet.

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

Examples of Header and Address fields in Standard Driver Sheet

Address on Studio
Well_Temp
Well_Press
Alarm_Clear
IWS_Float9
IWS_String0
MLights_MO

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

4. Save and close the worksheet.

Configuring the Main Driver Sheet

When you add the TAGW driver to your project, the Main Driver Sheet is automatically included in the **TAGW** folder in the Project Explorer. This section describes how to configure the worksheet.

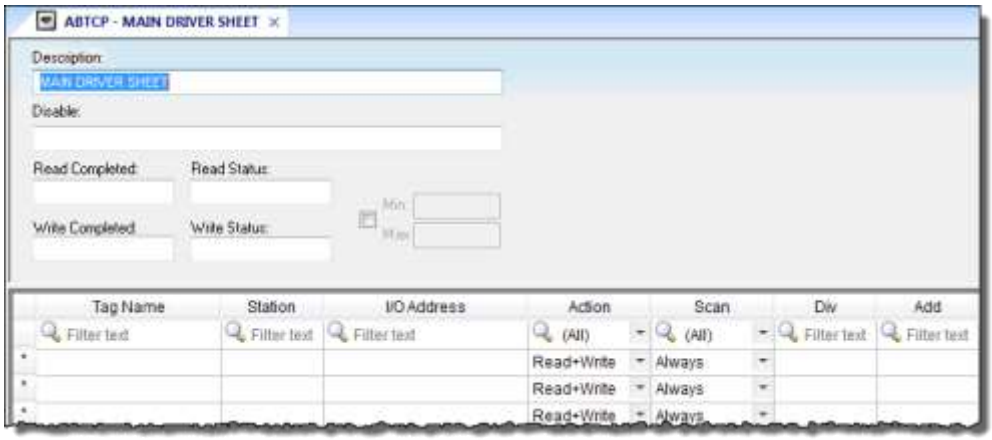
The TAGW driver must be added to the project before you can configure any of its worksheets. For more information, see Adding a communication driver to your project on page 6.

The Main Driver Sheet is used to define tag/register associations and driver parameters that are in effect at all times, regardless of project behavior. The worksheet is continuously processed during project runtime.

> **Note:** Most of the settings on this worksheet are standard for all drivers; for more information about configuring these settings, see the “Communication” chapter of the *Technical Reference Manual*. The **Station** and **I/O Address** fields, however, use syntax that is specific to the TAGW driver.

1. Do one of the following.
   - On the **Insert** tab of the ribbon, in the **Communication** group, click **Main Driver Sheet** and then select **TAGW** from the list.
   - In the **Comm** tab of the Project Explorer, expand the **TAGW** folder and then double-click **MAIN DRIVER SHEET**.

   The Main Driver Sheet is displayed.



Main Driver Sheet

2. For each tag/register association that you want to create, insert a row in the worksheet body and then configure the row's fields as described below.

Tag Name

Type the name of the project tag.

Station

Please see station for standard driver sheet.

I/O Address

Please see address field for the standard driver sheet.

> **Note:** The Main Driver Sheet can have up to 32767 rows. If you need to configure more than 32767 communication addresses, then either configure additional Standard Driver Sheets or create additional instances of the driver.

3. Save and close the worksheet.

Additional notes

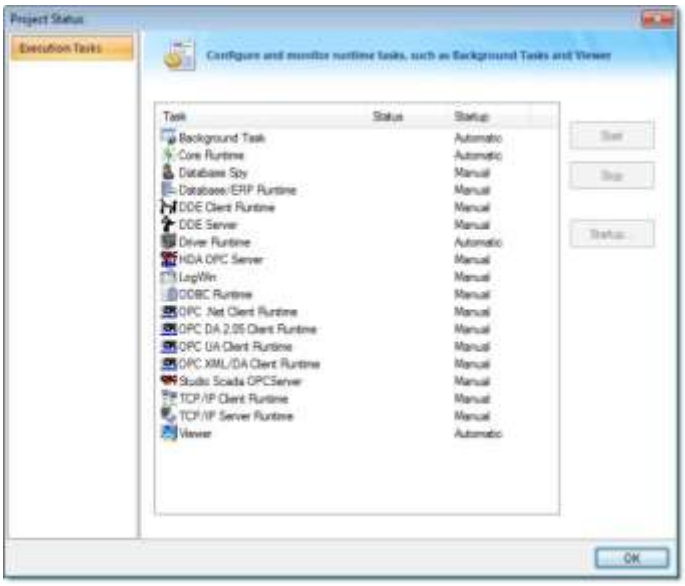
Additional notes about the TAGW driver.

Checking the Driver Runtime task

This section describes how to check the status of the Driver Runtime task in the list of execution tasks.

The Driver Runtime task handles communication with remote devices and the processing of the driver worksheets. By default, the task is configured to start up automatically when the project is run, but you can check it for yourself.

1. On the **Home** tab of the ribbon, in either the **Local Management** or the **Remote Management** group (depending on where you project server will be running), click **Tasks**.
   The *Project Status* dialog is displayed.



Project Status dialog

2. Verify that the **Driver Runtime** task is set to **Automatic**.
   - If the setting is correct, then proceed to the next step.
   - If the **Driver Runtime** task is set to **Manual**, select the task and then click **Startup** to change the task to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.

Troubleshooting

This section lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.

Checking status codes

If the TAGW driver fails to communicate with the target device, then the database tag(s) that you configured for the **Read Status** and **Write Status** fields of the driver sheets will receive a status code. Use this status code and the following tables to identify what kind of failure occurred and how it might be resolved. **Status codes for the driver**

Error	Description	Possible Causes	Procedure To Solve
1	User is not Logged in	The user is not logged in or is not authorized to access the cloud.	Please check the user name and password given in the station format
2	The Tag is not found in the Datatables.	The tag trying to be read or written is not present in the cloud.	Check the name of the tag given on IWS.
3	Unable to resolved Host. Please check the station format.	The URL of the cloud or the tags directory is incorrect.	Check the path and directory name given in the Station field.
4	No Response received from Host. Please check the inputs and try again.	The cloud server did not respond or time out.	Check the url, user name and password of the station format and also the tag names given on IWS.

5	Bad Function Argument. Please check the station and tagnames.	There is something incorrect in the station or the tags given.	Check the station and tagnames given.
6	Error writing the received data.	This can be due to an internal error or due to bad network.	Try sending the request again.
7	Unable to read the given tags.	The HTTP POST request failed to return data probably due to bad network.	Check the station, tagnames and network connectivity and try again.
8	Timeout	The request timed out.	Check the station, tagnames and network connectivity and try again.
9	Invalid Tag	The given tag does not exist in the cloud.	Check if the given tag exists on the cloud datatables and has the correct spelling and case sensitivity.
10	Unknown Error	The error is not known exactly.	Check all settings, network and configurations and try again.
11	CURL Init Failed	The CURL library failed to initialize.	Check network configurations and try again.

Common status codes

Status Code	Description	Possible Causes	Procedure To Solve
0	OK	Communicating without error.	None required.
-15	Timeout waiting for message to start	• Disconnected cables. • PLC is turned off, in stop mode, or in error mode. • Wrong station number. • Wrong parity (for serial communication). • Wrong RTS/CTS configuration (for serial communication).	• Check cable wiring. • Check the PLC mode — it must be RUN. • Check the station number. • Increase the timeout in the driver's advanced settings. • Check the RTS/CTS configuration (for serial communication).
-33	Invalid driver configuration file	The driver configuration file (*drivername*.INI) is missing or corrupt.	Reinstall the driver.
-34	Invalid address	The specified address is invalid or out of range.	Check the supported range of addresses described in this document, and then correct the address.
-35	Driver API not initialized	The driver library was not initialized by the driver.	Contact technical support.
-36	Invalid data type	The specified data type is invalid or out of range.	Check the supported data types described in this document, and then correct the data type.
-37	Invalid header	The specified header in the driver worksheet is invalid or out of range.	Check the supported range of headers described in this document, and then correct the header.
-38	Invalid station	The specified station in the driver worksheet is invalid or out of range.	Check the supported station formats and parameters described in this document, and then correct the station.

-39	Invalid block size	Worksheet is configured with a range of addresses greater than the maximum block size.	Check the maximum block size number of registers described in this document, and then configure your driver worksheet to stay within that limit. Keep in mind that you can create additional worksheets. **Note:** If you receive this error from a Main Driver Sheet or Tag Integration configuration, please contact Technical Support.
-40	Invalid bit write	Writing to a bit using the attempted action is not supported.	• Writing to a bit using Write Trigger is not supported in some drivers. Modify the driver worksheet to use Write On Tag Change. • The bit is read-only.
-42	Invalid bit number	The bit number specified in the address is invalid. The limit for the bit number depends on the registry type.	Check the addresses to see if there are bit numbers configured outside the valid range for the registry.
-43	Invalid byte number	The byte number specified in the address is invalid. The limit for the byte number depends on the registry type.	Check the addresses to see if there are byte numbers configured outside the valid range for the registry.
-44	Invalid byte write	Writing to a byte using the attempted action is not supported.	The byte is read-only or inacessible.
-45	Invalid string size	The string is more than 1024 characters.	Modify the addresses that have string data type to be less than 1024 characters.

Monitoring device communications

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 Serial Communication, 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 unit that runs Windows Embedded. The log is saved on the unit in the `celog.txt` file, which can be downloaded later.

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.

Contacting Technical Support

If you must contact Technical Support, please have the following information ready:

- **Operating System** and **Project Information**: To find this information, click **Support** in the **Help** tab of the ribbon.
- **Driver Version** and **Communication Log**: Displays in the *Output* window (LogWin module) when the driver is enabled and the project is running is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Revision history

This section provides a log of all changes made to the driver.

Revision history

Driver Version	Revision Date	Description of Changes	Author
1.0	Dec 31, 2014	First driver revision	Priya Yennam
1.1	Mar 21, 2017	Changed driver websetup to include API file (libcurl.dll)	Anushree Phanse
1.2	Mar 23, 2018	Support for the new libcurl	Thiago Henrique dos Santos Anushree Phanse