SOFTP - Driver Version 10.9
Doc. Revision C - Oct 19, 2016

SOFTP Communication Driver

Driver for Ethernet Communication with SoftPLCs

Table of Contents

Introduction

The SOFTP driver enables ethernet communication between the Web Studio system and SoftPLC's using the "DF1 on ethernet" protocol, according to the specifications discussed in this document.

This document will help you to select, configure, and execute the SOFTP 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 Web Studio system and the target device.
- **Selecting the Driver**: Explains how to select the SOFTP driver in the Web Studio system.
- **Configuring the Device**: Describes how the target device must be configured to receive communication from the SOFTP driver.
- **Configuring the Driver**: Explains how to configure the SOFTP driver in the Web Studio system, including how to associate database tags with device registers.
- **Executing the Driver**: Explains how to execute the SOFTP 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 SOFTP driver configuration
- **Revision History:** Provides a log of all changes made to the driver and this documentation.

Note

ıis document also assumes that you are familiar with Microsoft Windows environments. 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 Ethernet communication between the SOFTP driver in Web Studio and a SoftPLC.

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:** SoftPLC Corporation
- **Compatible Equipment:** All SoftPLC CPU's with version 3.*x* firmware or later
- **Programming Software:** TOPDOC NexGen

For a description of the device(s) used to test driver conformance, see “Conformance Testing” on page 4.

Network Specifications

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

- **Device Communication Port:** Ethernet Port
- **Physical Protocol:** Ethernet TCP/IP
- **Logic Protocol:** DF1
- **Device Runtime Software:** None
- **Specific PC Board:** Any TCP/IP adapter (Ethernet card)

Driver Characteristics

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

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

Note

You must use Adobe Acrobat® Reader™ to view the **`SOFTP.PDF`** document. You can download it from Adobe's Web site http://www.adobe.com

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

Windows 2000/XP/Vista/7/8/10

Windows Embedded 7/8/10

Windows CE, Embedded Compact

For a description of the operating systems used to test driver conformance, see "Conformance Testing" below.
The SOFTP driver supports the following register types:

Register Type	Length (in Bytes)	Default Format	Write	Read	Bit	Integer	Float	String	BCD
O (Output)	2	Word	●	●	●	●			●
I (Input)	2	Word		●	●	●			●
S (Status)	2	Word	●	●	●	●			
B (Binary)	2	Word	●	●	●	●	●		●
T (Timer)	6	Word	●	●		●			
C (Counter)	6	Word	●	●		●			
R (Control)	6	Word	●	●		●			
F (Float)	4	Float	●	●			●		
N (Integer File)	2	Word	●	●	●	●	●		●
ST (String File)	n	String	●	●				●	

Attention

- BCD format, only the first 12 bits of the register are transcribed to the associated tag. The last 4 bits are transcribed to the tag's **Quality** property. For more information about tag properties, please refer to the *Technical Reference Manual*.
- Float format uses 4 bytes (2 Words). When using the Float format with a register type that has the default size in 2 bytes (1 Word), then you use two consecutive addresses. This happens with Binary and Integer.

Conformance Testing

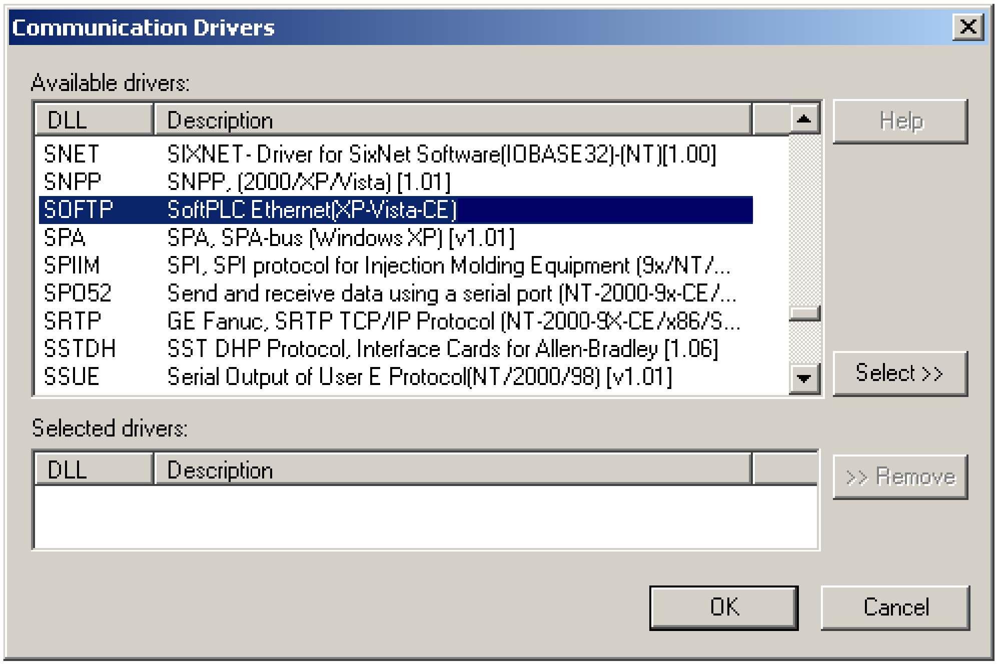
The following hardware/software was used for conformance testing:

Equipment:	HB2-HPLA-1K
Driver Configuration:	SOFTP
IP Address:	192.100.100.102
Protocol:	DF1 over TCP/IP
Cable:	Ethernet Cable
Operating System (development)	Windows 2000 SP4
Operating System (target):	Windows 2000 SP4
Web Studio Version:	6.1+SP5
Driver Version:	10.3

Selecting the Driver

When you install Web 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 SOFTP driver for your Web Studio application:

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



Communication Drivers

Available drivers:

DLL	Description
SNET	SIXNET- Driver for SixNet Software(IOBASE32)-(NT)[1.00]
SNPP	SNPP, (2000/XP/Vista) [1.01]
SOFTP	SoftPLC Ethernet(XP-Vista-CE)
SPA	SPA, SPA-bus (Windows XP) [v1.01]
SPIIM	SPI, SPI protocol for Injection Molding Equipment (9x/NT/...
SP052	Send and receive data using a serial port (NT-2000-9x-CE/...
SRTP	GE Fanuc, SRTP TCP/IP Protocol (NT-2000-9X-CE/x86/S...
SSTDH	SST DHP Protocol, Interface Cards for Allen-Bradley [1.06]
SSUE	Serial Output of User E Protocol(NT/2000/98) [v1.01]

Selected drivers:

DLL	Description

Communication Drivers Dialog

3. When the **SOFTP** 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.

Configuring the SoftPLC

Use TOPDOC NexGen to configure the SoftPLC's Ethernet TCP/IP settings.

Configuring the Driver

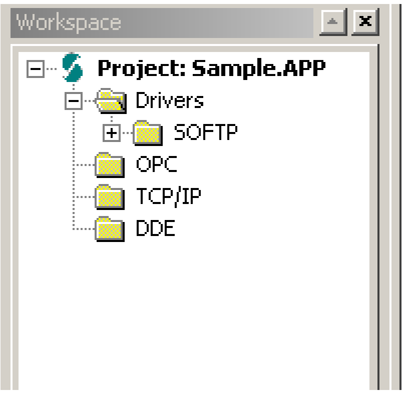
Once you have selected the SOFTP driver in Web Studio, you must properly configure it to communicate with your target SoftPLC. First, you must set the driver's communication settings to match the parameters set on the SoftPLC. Then, you must build driver worksheets to associate database tags in your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

Configuring the Communication Settings

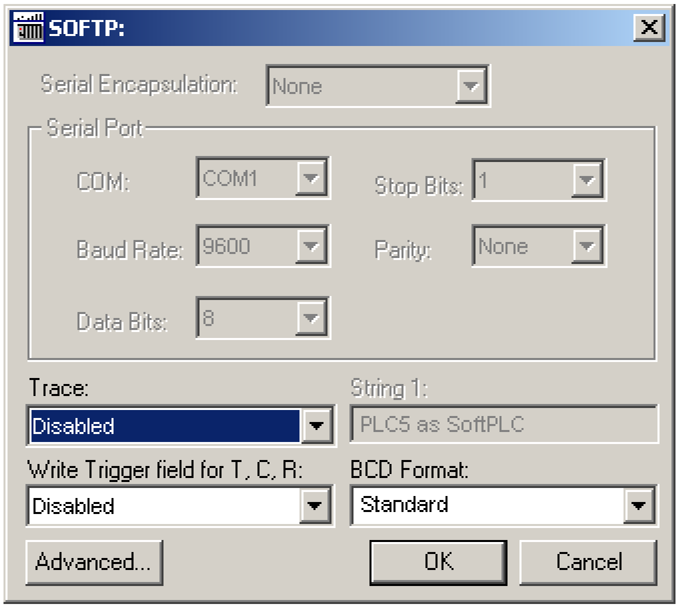
The communication settings are described in detail in the "Communication" chapter of the Web Studio Technical Reference Manual, and the same general procedures are used for all drivers. Please review those procedures before continuing.

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

1. In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The SOFTP driver is listed here as a subfolder.
2. Right-click on the *SOFTP* subfolder and then select the **Settings** option from the pop-up menu. The *SOFTP: Communication Parameters* dialog is displayed:



Select Settings from the Pop-Up Menu


SOFTP:

Serial Encapsulation: None

Serial Port

COM: COM1

Stop Bits: 1

Baud Rate: 9600

Parity: None

Data Bits: 8

Trace: Disabled

String 1: PLC5 as SoftPLC

Write Trigger field for T, C, R: Disabled

BCD Format: Standard

SOFTP: Communication Parameters Dialog

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

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

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:

Parameter	Default Value	Valid Values	Description

Trace	`Disabled`	`Disabled`
		or
		`Enabled`
			When the trace is enabled, the *LogWin* module will display more detailed information about the communication. If you are generating a log file for technical support, then we recommend enabling this option.

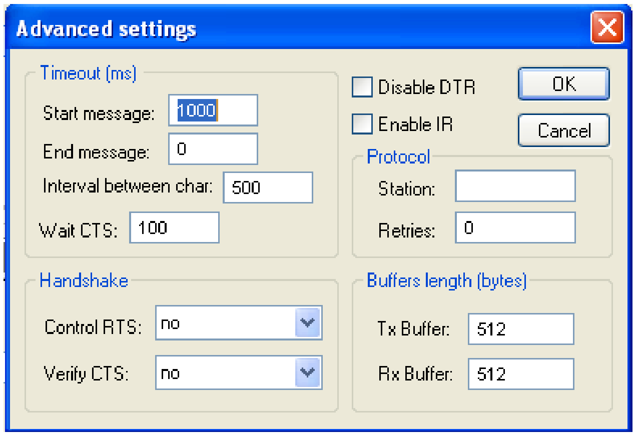
Write Trigger field for T, C, R	`Disabled`	`Disabled`
		or
		`Enabled`
			When this option is enabled, the driver is able to write using the Write Trigger field in the Standard Driver Worksheet. This option has effect only for Timers, Counter and Controls. For all remaining data types the Write Trigger field is always enabled.

BCD Format	`Standard`	`Standard`	**Standard:** Select Standard. Store all the BCD digits in the tag value.

Attention

➲ hen using **Write Trigger field** for the Timers (**T**), Counters (**C**) and Controls(**R**) registers, the driver reads the whole worksheet before writing**,** and then if at the same time the addresses that are not configured in the associated worksheet change the value, they may be overwritten.

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



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

You do not need to change any advanced settings at this time. You can consult the Web 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.

Configuring the Driver Worksheets

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.

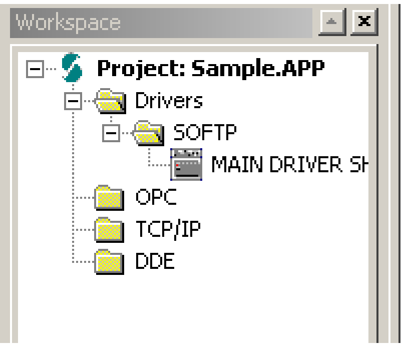
The configuration of these worksheets is described in detail in the “Communication” chapter of the *Web Studio Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

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

MAIN DRIVER SHEET

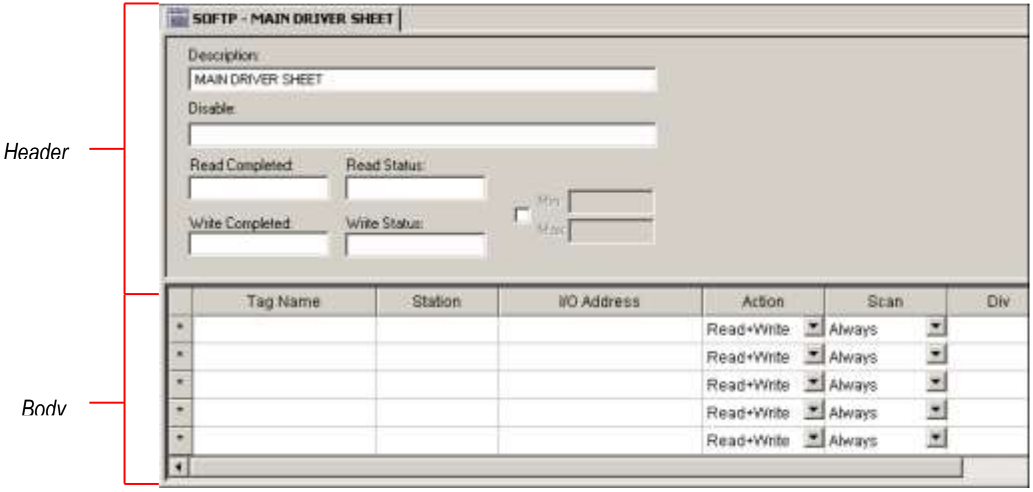
When you select the SOFTP driver and add it to your application, Web Studio automatically inserts the *Main Driver Sheet* in the *SOFTP* driver subfolder. To configure the Main Driver Sheet:

1. Select the *Comm* tab in the *Workspace* pane.Open the *Drivers* folder, and then open the *SOFTP* subfolder:



Main Driver Sheet in the SOFTP Subfolder

2. Double-click on the **MAIN DRIVER SHEET** icon to open the following worksheet:



Description: MAIN DRIVER SHEET

Tag Name	Station	I/O Address	Action	Scan	Div
			Read+Write	Always	
			Read+Write	Always	
			Read+Write	Always	
			Read+Write	Always	
			Read+Write	Always	

Opening the Main Driver Sheet

Most of the fields on this sheet are standard for all drivers; see the “Communication” section of Web Studio Help for more information on configuring these fields. However, the **Station** and **I/O Address** fields use syntax that is specific to the SOFTP driver.

3. For each table row (i.e., each tag/register association), configure the **Station** and **I/O Address** fields as follows…
   - **Station** field: Specify the IP Address of the device, using the following syntax:

     `<IP Address>:[optional Port Number]`

     Examples — **`192.168.2.9`**
     `192.168.125.31:2222`

     Where:

     – ***`<IP Address>`*** is the device’s IP address on the TCP/IP network; and

     – ***`[Port Number]`*** is the port number for the SoftPLC ethernet protocol (usually **2222**). This parameter is optional.

You can also specify a tag (e.g. **`{station}`**), but the tag value that is referenced must follow the same syntax and contain a valid value.

> **Attention**
>
> You must use a non-zero value in the **Station** field, and you cannot leave the field blank.

- **I/O Address:** Specify the address of the associated device register.

  For Inputs and Outputs, use the following syntax:

  `<Type>:<Slot Number>.[Data Format]<Address>/[Bit]`
  or
  `<Type>:<Slot Number>.[Data Format]<Address>.[Bit]`
  Example — **`O:0.2/4`** or **`O:0.2.4`**

> **Note:**
> If you are communicating with a SoftPLC, you do not specify the Slot number. Simply type **0** on it. Example: for the SoftPLC Address **O001/3**, type **O:0.W1/3**. Addresses must be in decimal.

For Status, use the following syntax::

`<Type>:[Data Format]<Address>/[Bit]`
or
`<Type>:[Data Format]<Address>.[Bit]`
Example — **`S:1/15 or S:1.15`**

For Binary, and Integer, use the following syntax:

`<Type><File Number>:[Data Format]<Address>/[Bit]`
or
`<Type><File Number>:[Data Format]<Address>.[Bit]`
Example — **`N7:150/2`** or **`N7:150.2`**

For Timers, Counters and Controls, use the following syntax:

<Type><File Number>*: *[Data Format]<Address>*.*<Element>
Example — **T4:0.DN**

For Float, use the following syntax:

<Type><File Number>*:*[F]*<Address>.*<Number of Bytes>
Example — **F8:3 or F8:F3**

For String, use the following syntax:

<Type><Type Group>*:*[S]*<Address>.*<Number of Bytes>
Example — **ST15:0.50 or ST15:S0.50**

Where:

– ***<Type>*** : Device register type. Valid values are **O** (Output), **I** (Input), **S** (Status), **B** (Binary), **N** (Integer), **T** (Timer), **C** (Counter), **R** (Control), **F** (Float), and **ST** (String).

– ***<File Number>*** : Data Table File number of the specified register type.

– ***<Slot Number>*** : Always 0.

– ***[Data Format]*** : Format of the data being read or written, which determines how Web Studio will handle the data. Valid values are **W** (Word), **B** (BCD), **F** (Floating Point), and **S** (String). This parameter is *optional*; it can be left out of the address if the default format for the register type (as described on page 4) is acceptable.

Attention:

When using the BCD format, only the first 12 bits of the register are transcribed to the associated tag. The last 4 bits are transcribed to the tag's **Quality** property. For more information about tag properties, please refer to the *Technical Reference Manual*.

Float format uses 4 bytes (2 words). When using the Float format with a register type that has the default size in 2 bytes. Then, it will use two consecutive addresses. This happens with Output, Input, Status, Binary, and Integer.

– ***<Address>*** : Decimal address of the desired register.

– ***[Bit]*** (optional): The bit number (from 0 to 15) of the address. For I/O bits, enter in decimal. The driver will convert to octal when sending to the SoftPLC.

– ***<Number of Bytes>*** : Maximum size (in bytes) of the String.

– ***<Element>*** : Element type for Timer, Counter or Control, according to the following table:

Register	Elements																	
	PRE	ACC	CON	DN	EN	TT	UN	OV	CD	CU	LEN	POS	FD	IN	UL	ER	EM	EU
Timer	•	•	•	/	/	/												
Counter	•	•	•	/			/	/	/	/								
Control				/	/						•	•	/	/	/	/	/	/

STANDARD DRIVER WORKSHEET

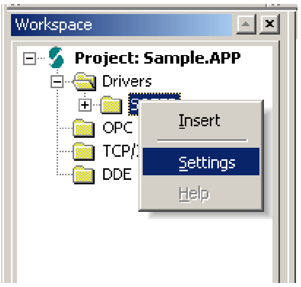
When you select the SOFTP driver and add it to your application, it has only a Main Driver Sheet by default (see previous section). However, you may insert additional Standard Driver Worksheets to define tag/register associations that are triggered by specific application behaviors. Doing this will optimize 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.

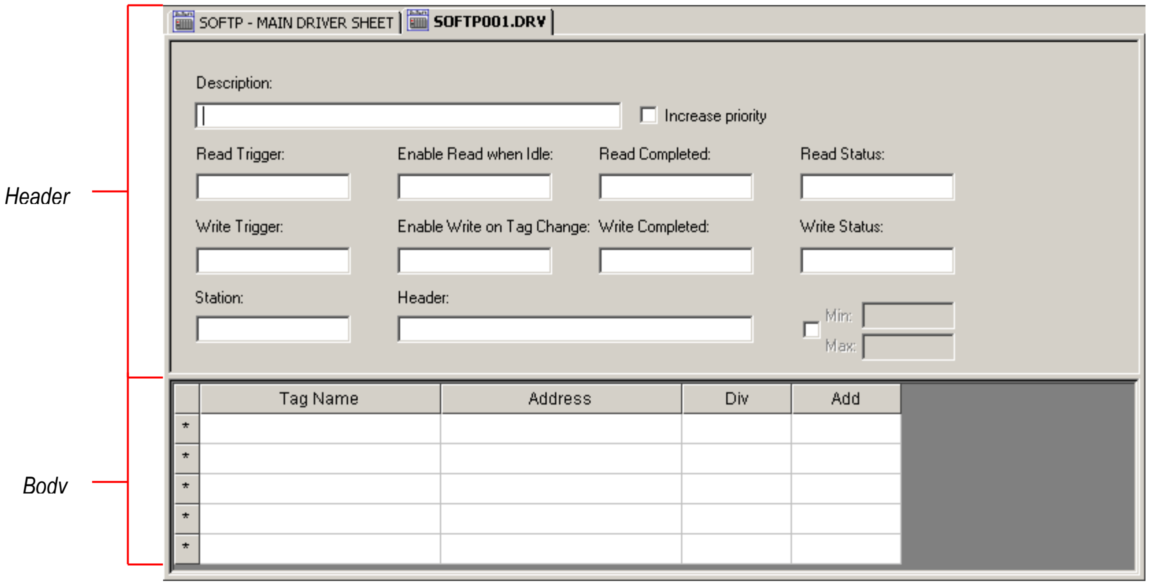
To insert a new Standard Driver Worksheet:

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



Inserting a New Worksheet

A new SOFTP driver worksheet is inserted into the *SOFTP* subfolder, and the worksheet is opened:



SOFTP Driver Worksheet

> **Note**
>
> Worksheets are numbered in order of creation, so the first worksheet is **`SOFTP001.drv`**.

*Most of the fields on this worksheet are standard for all drivers; see the "Communication" section of the Web Studio Help for more information on configuring these fields. However, the **Station**, **Header**, and **Address** fields use syntax that is specific to the SOFTP driver.*

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

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

  `<IP Address>:[optional Port Number]`

  Examples — **`192.168.2.9`**
  `192.168.125.31:2222`

  Where:

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

  – ***`[Port Number]`*** is the port number for the DF1 protocol (usually 2222). This parameter is optional; you need to specify it only if the device cannot autodetect incoming DF1 communication.

You can also specify an indirect tag (e.g. **`{station}`**), but the tag that is referenced must follow the same syntax and contain a valid value.

Attention

You must use a non-zero value in the **Station** field, and you cannot leave the field blank.

- **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.

  For Inputs and Outputs, use the following syntax:

  `<Type>:<Slot Number>.<Address Reference>`

  Example — **`O:0.0`**

Note

If you are communicating with a SoftPLC set the Slot number to **0** (Zero). Example: **O:0.0**

For Status, use the following syntax:

`<Type>:<Address Reference>`

Example — **`S:0`**

For Binary, Integer, Timer, Counter, Control, String, use the following syntax:

`<Type><File Number>:<Address Reference>`

Example — **`N7:0`** or **`ST15:0`**

Where:

- ***`<Type>`*** : Device register type. Valid values are **`O`** (Output), **`I`** (Input), **`S`** (Status), **`B`** (Binary), **`N`** (Integer), **`T`** (Timer), **`C`** (Counter), **`R`** (Control), **`F`** (Float), and **`ST`** (String).
- ***`<File Number>`*** : Data Table File number of the specified register type.
- ***`<Slot Number>`*** : Is always 0 (Zero).
- ***`<Address Reference>`*** : The initial address (reference) of the block of registers configured on this worksheet.

After you edit the **Header** field, Web Studio checks the syntax to determine if it is valid. If the syntax is invalid, then Web Studio automatically inserts a default value of **`N7:0`**.

You can also specify an indirect tag (e.g. **{header}**), but the tag that is referenced must follow the same syntax and contain a valid value.

Information about the Header Parameter			
Register Type	**Example of Syntax**	**Valid Range of Initial Address**	**Comments**
Output	**O:0.0**	0-Varies according to the equipment (in octal)	Physical outputs: Where "O" means output. The first digit is always 0 and the first digit following the dot is the output word address, in decimal.
Input	**I:0.0**	0-Varies according to the equipment (in octal)	Physical inputs: Where "I" means input. The first digit after the colon is always 0 and the digit following the dot is the output input word address, in decimal.
Status	**S:0**	0 ~ 31	Reads the status words.
Binary	**B3:0**	0 ~ 9999	Reads the Binary Operator.
Integer	**N7:0**	0 ~ 9999	Reads and Writes the Integer addresses.
Timer	**T4:0**	0 ~ 9999	Reads and Writes the Timer addresses.
Counter	**C5:0**	0 ~ 9999	Reads and Writes the Counter addresses.
Control	**R6:0**	0 ~ 9999	Reads and Writes the Control addresses.
Float	**F8:0**	0 ~ 9999	Reads and Writes the Float addresses.
String	**ST15:0**	0 ~ 9999	Reads and Writes the String addresses.

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

Register Type	Syntax	Accepted DataType	Examples
Output Input	***[Data Format]<Address>/[Bit]*** or ***[Data Format]<Address>.[Bit]***	W	W0/3 W0.3
Status Binary Integer	***[DataFormat]<Address>/[Bit]*** or ***[Data Format]<Address>.[Bit]***	B, W and F	W10/12 W10.12
Timer Counters Controls	***[Data Format]<Address>.<Element>*** or ***[Data Format]<Address>/<Element>***	W	W2.PRE W2/PRE
Float	***[F]<Address>***	F	F1.2
String	***S<Address>.<Number of WORDS>***	S	S1.2

Where:

- ***[Data Format]*** : Format of the data being read or written, which determines how Web Studio will handle the data. Valid values are **W** (Word), **B** (BCD), **F** (Floating Point), and **S** (String). This parameter is *optional*; it can be left out of the address if the default format for the register type (as described on page 4) is acceptable.

Attention:

When using the BCD format, only the first 12 bits of the register are transcribed to the associated tag. The last 4 bits are transcribed to the tag's **Quality** property. For more information about tag properties, please refer to the *Technical Reference Manual*.

- ***<Address>*** : Decimal value added to the ***<Address Reference>*** parameter (configured in the **Header** field above) to produce complete register address.

- ***[Bit]*** (optional): The bit number (from 0 to 15) of the address.

- ***<Number of Bytes>*** : Maximum size (in bytes) of the String.

- ***<Element>*** : Element type for Timer, Counter or Control, according to the following table:

Register	Elements																	
	PRE	ACC	CON	DN	EN	TT	UN	OV	CD	CU	LEN	POS	FD	IN	UL	ER	EM	EU
Timer	●	●	●	/	/	/												
Counter	●	●	●	/			/	/	/	/								
Control				/	/						●	●	/	/	/	/	/	/

Attention

- You can use the Bit Writing function only when the **Write on tag change** field is enabled; you cannot use the **Write trigger** field for the Bit Writing function.
- When addressing Input and Output registers and bits on SoftPLC devices, the driver will convert the address field from a decimal to an octal value.

For more information about device registers and addressing, please consult TOPDOC NexGen documentation.

Attention

You must not configure a range of addresses greater than the maximum block size of 244 bytes (data buffer length) supported by SoftPLC within the same worksheet.

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

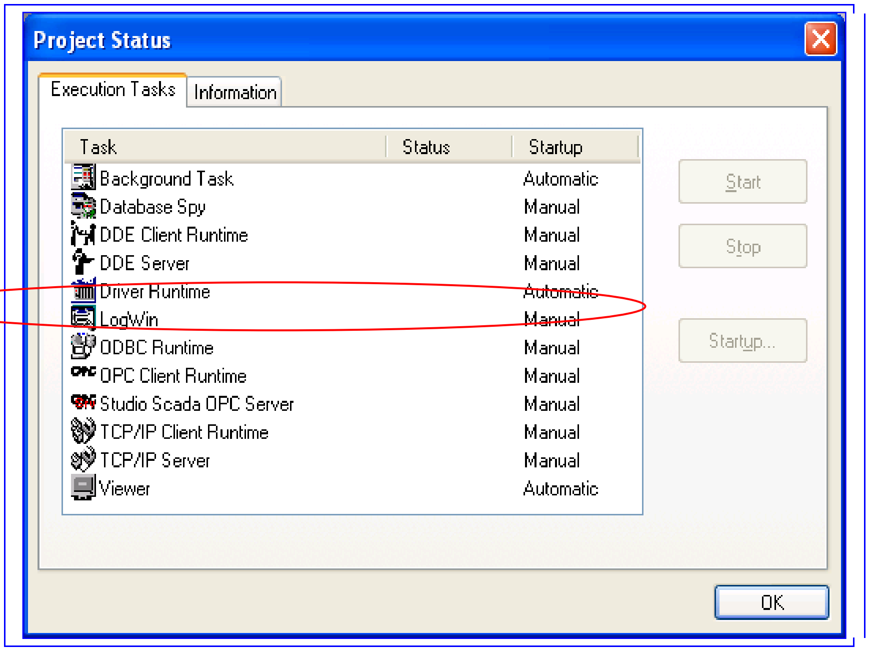
Address on the Device	Header Field	Address Field
I0/7	`I:0.0`	`W0/7` or `0/7`
I0/10	`I:0.0`	`W0/8` or `0/8`
I0/17	`I:0.0`	`W0/15` or `0/15`
I3/4	`I:0.3`	`W0/4` or `0/4`
I10/4	`I:0.8`	`W0/4` or `0/4`
O0/7	`O:0.0`	`W0/7` or `0/7`
O0/10	`O:0.0`	`W0/8` or `0/8`
O0/17	`O:0.0`	`W0/15` or `0/15`
O3/4	`O:0.3`	`W0/4` or `0/4`
O12/4	`O:0.0`	`W10/4` or `10/4`
S0/5	`S:0`	`W0/5` or `0/5`
S10/7	`S:0`	`W10/7` or `10/7`
S10/7	`S:10`	`W0/7` or `0/7`
B3:0/5	`B3:0`	`W0/5` or `0/5`
B3:10/7 or B3/167	`B3:0`	`W10/7` or `10/7`
B3:10/7 or B3/167	`B3:10`	`W0/7` or `0/7`
N7:0	`N7:0`	`W0` or `0`
N7:0/10	`N7:0`	`W0/10` or `0/10`
N7:50	`N7:20`	`W30` or `30`
T4:0.ACC	`T4:0`	`W0.ACC` or `0.ACC`
T4: 0.PRE	`T4:0`	`W0.PRE` or `0.PRE`
T15: 0.EN	`T15:0`	`W0.EN` or `0.EN`
T15: 1.ACC	`T15:0`	`W1.ACC` or `1.ACC`
T15: 0.ACC	`T15:0`	`W0.ACC` or `0.ACC`
C5: 1.ACC	`C5:0`	`W1.ACC` or `1.ACC`
C5: 0.PRE	`C5:0`	`W0.PRE` or `0.PRE`
R6: 0.LEN	`R6:0`	`W0.LEN` or `0.LEN`
R6: 0.POS	`R6:0`	`W0.POS` or `0.POS`
R6: 1.POS	`R6:0`	`W1.POS` or `1.POS`
F8:0	`F8:0`	`F0` or `0`
F8:5	`F8:5`	`F0` or `0`
F8:5	`F8:0`	`F5` or `5`
ST15:0 (String: maximum 20 bytes)	`ST15:0`	`S0.20` or `0.20`
ST15:1 (String: maximum 50 bytes)	`ST15:0`	`S1.50` or `1.50`
ST15:2 (no maximum)	`ST15:1`	`S1` or `1`

Executing the Driver

By default, Web 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 SOFTP 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.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communication without problems	None required.
1	Protocol Error	Wrong Family type specified for the Family field on the *Communication Parameters* dialog Wrong cable or CPU in fault mode	Check PLC family specified in Family field. Check CPU mode. Check cable.
2	Block Size Error	Offset specified for the Driver Configuration worksheet is too big and the message cannot be framed	Change offsets or create a new worksheet.

Error Code	Description	Possible Causes	Procedure to Solve
3	Invalid Family	Wrong family specified for **Family** field	Type the correct PLC Family.
10	Sequence error of message	Error is another protocol error	Check CPU status (high probability of a CPU error).
16	Illegal Command or Format	Any of the addresses configured in Driver Worksheets may not exist in PLC, or the family configured in the Driver Settings does not correspond to the actual PLC family	Check PLC address and compare with the Driver Worksheets. Check the family type
20	Invalid Address	Wrong Element specified for a Timer, Counter, or Control parameter in Address field. Wrong address syntax specified for other files	Check Header field. If you use a TAG, check whether the TAG value is valid for the specified addresses. If you did not use a TAG, you might have changed the Header after configuring the Addresses and the addresses are invalid for the new Header.
30	Invalid Header	Wrong Header typed on the Driver Configuration worksheet	Refer to “Configuring the Station and Header Fields” section for samples of valid Headers.
32	Host has a problem and will not communicate	Verification errors within the host PLC processor	Check entire driver and PLC configuration.
48	Remote node host is missing, disconnected, or shutdown	PLC is not connected, the node configuration is invalid, or in a shutdown process	Check the PLC state to ensure the processor is not at fault. Check to see if the cables are connected correctly.
50	Writing bit is not allowed via Write Trigger	Tag specified for the Write trigger field on a Driver Configuration worksheet was used for Writing Bit	Check Write Target tag. Bit writing is allowed only if you use **Write on tag change**.
64	Host could not complete function due to hardware fault	Local PLC backplane error (either memory parity or timeout/disconnected) and PLC aborted the message execution.	Check parity configuration for both PLC and *Communication Parameters* dialog. Verify that cables are connected properly.
80	Address problem or memory protect rungs	Attempt to access an illegal address in the PLC processor has aborted message execution	Check that you configured the Family on *Communication Parameters* dialog. Check PLC address.
90	Error Commit	Web Studio cannot connect to remote device using TCP/IP	Verify IP address. Use PING command with IP address to verify network communication.
96	Function not allowed due to command protection selection	Execution of a command at the PLC processor disabled by a switch option	Check PLC to see if a command protection is specified. If so, disable it.
112	Processor is in Program mode	PLC is in Program or Remote program mode, or the 1771-KA is in download mode	Switch PLC mode to RUN.
128	Compatibility mode file missing or communication zone problem	Execution of protected commands at the PLC processor inhibited because PROG light is ON	Check to see if PLC is on PROG mode. If so, change to RUN mode. Compare existing files to configured files.
144	Remote node cannot	Driver error: invalid command.	Contact Technical Support Staff.

Error Code	Description	Possible Causes	Procedure to Solve
	buffer command		
150	Send File number error	Web Studio system is sending a message and the PLC does not respond The PLC responds that there is no file specified Incorrectly specified station	Verify that required file exists. Check Station field and PLC address.
176	Remote node problem due to download	PLC is under downloading action or last download generated a problem.	Try downloading PLC program again.
177	Write or Read bit on BCD or Float Type is not allowed	Invalid BCD operation	Configure Bit and BDC Data Format separately.
330	Error connecting	Web Studio cannot connect to remote device using TCP/IP	Verify IP address. Use PING command with IP address to verify network communication.
65535	Error sending DF1 message.	Disconnected cables PLC turned off, in Stop or Error mode Wrong Station number Wrong RTS/CTS control setting.	Check cable wiring. Check PLC state. It must be RUN. Check station number. See ”Link Characteristics” section for valid RTS/CTS configurations. Check required configuration.
-15	Timeout waiting start a message.	▪ Disconnected cables ▪ PLC turned off, in Stop or Error mode ▪ Wrong Station number ▪ Wrong RTS/CTS control settings	Check cable wiring. Check PLC state. It must be RUN. Check station number. See ”Link Characteristics section for valid RTS/CTS configurations. Check required configuration.
-17	Timeout between rx char.	▪ PLC in stop or error mode ▪ Wrong station number ▪ Wrong parity ▪ Wrong RTS/CTS configuration settings	Check cable wiring. Check PLC state (must be RUN) Check station number. See “Network Specifications” section for valid RTS/CTS configurations. Check required configuration.

⇨ **Tip:**

You can monitor communication status by establishing an event log in Web 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 (**Tools** → **LogWin**) to establish an event log on a remote unit that runs Windows CE. 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 Web Studio and the target device, then try instead to establish communication using the device’s own programming software (e.g., TOPDOC NexGen). 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 Web 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 Web Studio *Output* window when the driver is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Sample Application

A sample application that employs the SOFTP driver is provided on the Web Studio installation CD. We strongly recommend that you use this sample application to test the driver *before* you develop your own applications, for the following reasons:

- To better understand the information and instructions provided in this document;
- To verify that your driver configuration is working satisfactorily with the target device; and
- To ensure that the all of hardware used in the test (i.e. the device, adapter, cable, and PC) is functioning safely and correctly.

> **Note:**
>
> The following instructions assume that you are familiar with developing project applications in Web Studio. If you are not, then please review the relevant chapters of the Web Studio *Technical Reference Manual* before proceeding.

To use the sample application:

1. Configure the device's communication settings according to the TOPDOC NexGen documentation.
2. Run Web Studio.
3. From the main menu bar, select **File** **Open Project**.
4. Insert the Web Studio installation CD and browse it to find the sample application. It should be located in the directory **`\EXAMPLES\SOFTP`**.
5. Select and open the sample application.
6. Configure and test the driver, as described in the rest of this document.

When you have thoroughly tested the driver with your target SoftPLC, you may proceed with developing your own Web Studio application projects.

⇨ **Tip:**
You can use the sample application screen as the maintenance screen for your own applications.

Octal - Decimal Conversion

Converting Decimal to Octal:	Converting Octal to Decimal:
Converting decimal to octal is done by successive division by 8 as shown below: Convert 177_{10} to an octal number. 177 / 8 = 22 remainder **1** 22 / 8 = 2 remainder **6** 2 / 8 = 0 remainder **2** *Answer is read from bottom to top (e.g.: 261)* Therefore 177_{10} (decimal) = 261_8 (octal)	In octal numerals each place is a power with base 8. $632_8 = (6 \times 8^2) + (3 \times 8^1) + (2 \times 8^0)$ [$8^0=1$] $= (6 \times 64) + (3 \times 8) + (2 \times 1)$ $= 384 + 24 + 2$ $= 410_{10}$ Therefore 632_8 (octal) = 410_{10} (decimal)

Below is a Decimal to Octal conversion table:

Decimal	Octal	Decimal	Octal	Decimal	Octal	Decimal	Octal	Decimal	Octal	Decimal	Octal
0	000	43	053	86	126	129	201	172	254	215	327
1	001	44	054	87	127	130	202	173	255	216	330
2	002	45	055	88	130	131	203	174	256	217	331
3	003	46	056	89	131	132	204	175	257	218	332
4	004	47	057	90	132	133	205	176	260	219	333
5	005	48	060	91	133	134	206	177	261	220	334
6	006	49	061	92	134	135	207	178	262	221	335
7	007	50	062	93	135	136	210	179	263	222	336
8	010	51	063	94	136	137	211	180	264	223	337
9	011	52	064	95	137	138	212	181	265	224	340
10	012	53	065	96	140	139	213	182	266	225	341
11	013	54	066	97	141	140	214	183	267	226	342
12	014	55	067	98	142	141	215	184	270	227	343
13	015	56	070	99	143	142	216	185	271	228	344
14	016	57	071	100	144	143	217	186	272	229	345
15	017	58	072	101	145	144	220	187	273	230	346
16	020	59	073	102	146	145	221	188	274	231	347
17	021	60	074	103	147	146	222	189	275	232	350
18	022	61	075	104	150	147	223	190	276	233	351
19	023	62	076	105	151	148	224	191	277	234	352
20	024	63	077	106	152	149	225	192	300	235	353
21	025	64	100	107	153	150	226	193	301	236	354
22	026	65	101	108	154	151	227	194	302	237	355
23	027	66	102	109	155	152	230	195	303	238	356
24	030	67	103	110	156	153	231	196	304	239	357
25	031	68	104	111	157	154	232	197	305	240	360
26	032	69	105	112	160	155	233	198	306	241	361
27	033	70	106	113	161	156	234	199	307	242	362
28	034	71	107	114	162	157	235	200	310	243	363
29	035	72	110	115	163	158	236	201	311	244	364
30	036	73	111	116	164	159	237	202	312	245	365
31	037	74	112	117	165	160	240	203	313	246	366
32	040	75	113	118	166	161	241	204	314	247	367
33	041	76	114	119	167	162	242	205	315	248	370
34	042	77	115	120	170	163	243	206	316	249	371
35	043	78	116	121	171	164	244	207	317	250	372
36	044	79	117	122	172	165	245	208	320	251	373
37	045	80	120	123	173	166	246	209	321	252	374
38	046	81	121	124	174	167	247	210	322	253	375
39	047	82	122	125	175	168	250	211	323	254	376
40	050	83	123	126	176	169	251	212	324	255	377
41	051	84	124	127	177	170	252	213	325		
42	052	85	125	128	200	171	253	214	326		

Revision History

Doc. Revision	Driver Version	Date	Description of Changes
A	10.3	12 May 2009	First driver version
B	10.3	29 June 2009	Format changes
C	10.9	19 Oct 2016	Corrected pg 16 table entry for I10/4, Updated O/S info