

VILL Communication Driver

Driver for Ethernet Communication
with the SDA proprietary system

Contents

INTRODUCTION	2
GENERAL INFORMATION.....	3
DEVICE CHARACTERISTICS	3
LINK CHARACTERISTICS.....	3
DRIVER CHARACTERISTICS	3
INSTALLING THE DRIVER	4
CONFIGURING THE DRIVER	5
EXECUTING THE DRIVER	7
TROUBLESHOOTING	8
SAMPLE APPLICATION	10
REVISION HISTORY.....	11

Introduction

The VILL driver enables communication between the Studio system and devices using the proprietary protocol communicating over Ethernet, according to the specifications discussed in this document.

This document was designed to help you install, configure, and execute the VILL driver to enable communication with these devices. The information in this document is organized as follows:

- **Introduction:** Provides an overview of the VILL driver documentation.
- **General Information:** Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the VILL driver.
- **Installing the Driver:** Explains how to install the VILL driver.
- **Configuring the Driver:** Explains how to configure the VILL driver.
- **Executing the Driver:** Explains how to execute the driver to verify that you installed and configured the driver correctly.
- **Troubleshooting:** Lists the most common error codes for this protocol and explains how to fix these errors.
- **Sample Application:** Explains how to use a sample application to test the VILL driver configuration.
- **Revision History:** Provides a log of all modifications made to the driver and the documentation.

Notes:

- This document assumes that you have read the “Development Environment” chapter in the *Studio Technical Reference Manual*.
- This document also assumes that you are familiar with the Windows NT/2000/XP/CE environment. If you are unfamiliar with Windows NT/2000/XP/CE, we suggest using the **Help** feature (available from the Windows desktop **Start** menu) as you work through this guide.

General Information

This chapter explains how to identify all the hardware and software components used to implement communication between the Studio VILL driver and the SDA proprietary system.

The information is organized into the following sections:

- Device Characteristics
- Link Characteristics
- Driver Characteristics

Device Characteristics

To establish communication, you must use devices with the following specifications:

- **Manufacturer:** SDA proprietary system (SDA is an Informix-based system, proprietary from Villares Company).
- **Device Runtime Software:** None

For a list of the devices used for conformance testing, see “Conformance Testing.”

Link Characteristics

To establish communication, you must use links with the following specifications:

- **Device Communication Port:** Ethernet
- **Physical Protocol:** Ethernet
- **Logic Protocol:** Proprietary
- **Specific PC Board:** None

Driver Characteristics

The VILL driver is composed of the following files:

- **VILL.INI:** Internal driver file. *You must not modify this file.*
- **VILL.MSG:** Internal driver file containing error messages for each error code. *You must not modify this file.*
- **VILL.PDF:** Document providing detailed information about the VILL driver.
- **VILL.DLL:** Compiled driver.

Notes:

- All of the preceding files are installed in the /DRV subdirectory of the Studio installation directory.
- You must use Adobe Acrobat® Reader™ (provided on the Studio installation CD-ROM) to view the VILL.PDF document.

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

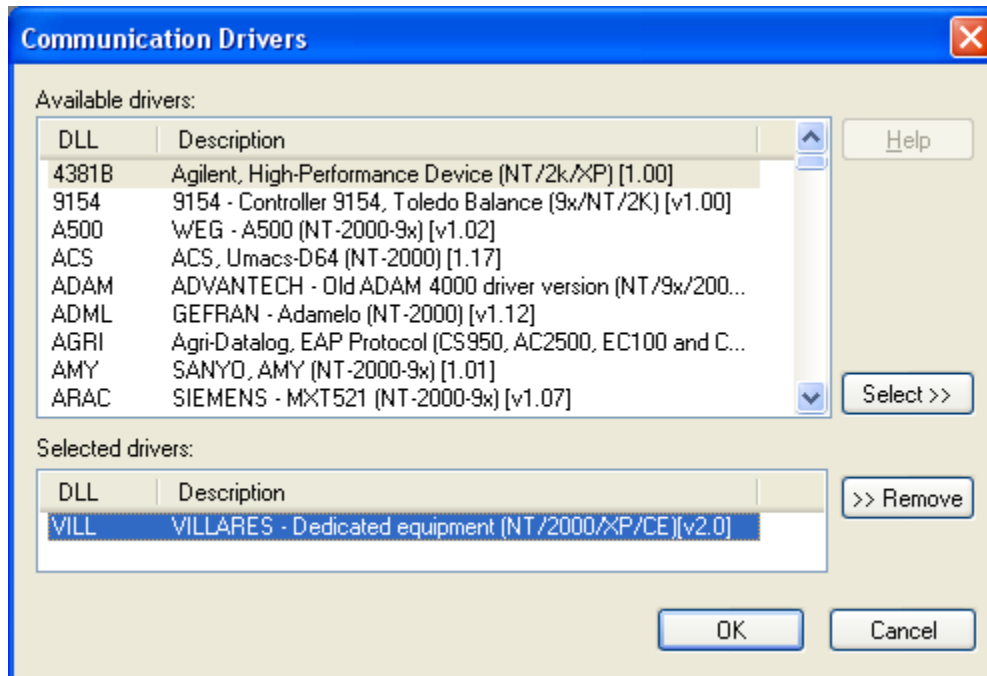
- Windows NT/2K/XP/CE

Installing the Driver

When you install Studio version 5.1 or higher, all of the communication drivers are installed automatically. You must select the driver that is appropriate for the application you are using.

Perform the following steps to select the driver from within the application:

1. Open Studio from the **Start** menu.
2. From the Studio main menu bar, select **File** → **Open Project** to open your application.
3. Select **Insert** → **Driver** from the main menu bar to open the *Communication Drivers* dialog.
4. Select the **VILL** driver from the *Available Drivers* list, and then click the **Select** button:



Communication Drivers Dialog

5. When the **VILL** driver displays in the *Selected Drivers* list, click the **OK** button to close the dialog.

Note:

It is not necessary to install any other software on your computer to enable communication between the host and the device.

Attention:

For safety reasons, you must use special precautions when installing the physical hardware. Consult the hardware manufacturer's documentation for specific instructions in this area.

Configuring the Driver

After opening Studio and selecting the VILL driver, you must configure the driver. Configuring the VILL driver is done in two parts:

- Specifying communication parameters
- Defining tags and controls in the *STANDARD DRIVER SHEETS* (or Communication tables)

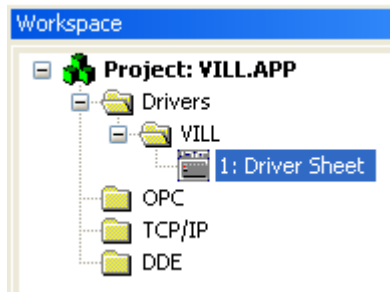
Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header**, and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header**, and **Address** fields only.

Note:
For a detailed description of the Studio *STANDARD DRIVER SHEETS*, and information about configuring the standard fields, review the product's *Technical Reference Manual*.

Configuring the *STANDARD DRIVER SHEET*

Use the following steps to create a new *STANDARD DRIVER SHEET*:

1. From the Studio development environment, select the **Comm** tab, located below the *Workspace* pane.
2. In the *Workspace* pane, expand the *Drivers* folder and right-click the *VILL* subfolder.
3. When the pop-up menu displays, select the **Insert** option:



Inserting a New Worksheet

The **STANDARD DRIVER SHEET** displays (similar to the following figure):

The screenshot shows a configuration window titled "VILLO01.DRV". It includes the following fields and a table:

- Description:** "Driver Sheet" (text input), Increase priority
- Read Trigger:** (text input)
- Enable Read when Idle:** (text input)
- Read Completed:** (text input)
- Read Status:** (text input)
- Write Trigger:** (text input)
- Enable Write on Tag Change:** (text input)
- Write Completed:** (text input)
- Write Status:** (text input)
- Station:** "127.0.0.1" (text input)
- Header:** "DATA" (text input)
- Min: (text input)
- Max: (text input)

	Tag Name	Address	Div	Add
1	VAR[1]	INT:VAR1		
2	VAR[2]	REAL:VAR2		
3	VAR[3]	VAR3		
4	VAR[4]	VAR4		
5	VAR[5]	STRING:VAR5		
6	VAR[6]	INT:VAR6		
7	VAR[7]	INT:VAR7		
8	VAR[8]	VAR8		
9	VAR[9]	VAR9		
10	VAR[10]	VAR10		

STANDARD DRIVER SHEET

In general, all parameters on the *Driver* worksheet (except the **Station**, **Header**, and **Address** fields) are standard for all communication drivers, but they will not be discussed in this document. For detailed information about configuring the standard parameters, consult the *Studio Technical Reference Manual*.

- Use the following information to complete the **Station**, **Header**, and **Address** fields on this worksheet.
 - Station** field: Use this field to specify the IP Address of the device (*unit name*) and the slot number.
 The following syntax:
<IP Address>:<Slot>
 - Header** field: Set this field always with the expression "DATA".
 - Address** field: Use this field to associate each tag to its respective variable from the SDA system. Type the tag from your application database into the **Tag Name** column. This tag will receive values from or send values to a variable from SDA system. The address must comply with the following syntax:
[<Type>:]<Variable Name>

The **<Type>** parameter is optional, and might be **INT**, **REAL** or **STRING**. If not used, the default is **STRING**. The type is used to convert a tag value to the expected type on the SDA system. It is also used when values are received, for example if an address is of type **INT** and the received value is a float it is properly converted. Also, if the tag contains a numeric value and the address is of type **STRING** the value is converted to a floating-point number.

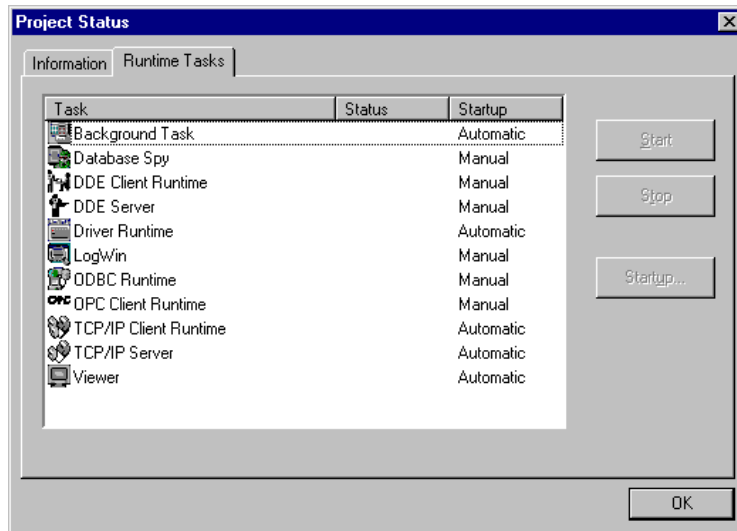
Executing the Driver

After adding the VILL driver to a project, Studio sets the project to execute the driver automatically when you start the run-time environment.

To verify that the driver run-time task is enabled and will start correctly, perform the following steps:

1. Select **Project** → **Status** from the main menu bar.

The *Project Status* dialog displays:



Project Status Dialog

2. Verify that the *Driver Runtime* task is set to **Automatic**.
 - If the setting is correct, click **OK** to close the dialog.
 - If the **Driver Runtime** task is set to **Manual**, select the **Driver Runtime** line. When the **Startup** button becomes active, click the button to toggle the *Startup* mode to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.
4. Start the application to run the driver.

Troubleshooting

If the VILL driver fails to communicate with the device, the tag you configured for the **Read Status** or **Write Status** fields 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	Invalid Command	Command not supported.	Check the command.
2	Invalid Address	Specified an invalid address.	Type a valid address.
6	Invalid Message Length	Invalid Message.	<ul style="list-style-type: none"> ▪ Check the message length (Max. 4096 bytes). ▪ Check the tag length (Max. 128 bytes). ▪ Check the tag number in the Communication Sheet (Max. 1024 tags).
-15	Timeout Start Message	<ul style="list-style-type: none"> ▪ Disconnected cables. ▪ PLC is turned off, in stop mode, or in error mode. ▪ Wrong station number. ▪ Wrong RTS/CTS control settings. 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the configuration. See <i>Studio Technical Reference Manual</i> for information about valid RTS/CTS configurations.
-17	Timeout between rx char	<ul style="list-style-type: none"> ▪ PLC in stop mode or in error mode. ▪ Wrong station number. ▪ Wrong parity. ▪ Wrong RTS/CTS configuration settings. 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the configuration. See <i>Studio Technical Reference Manual</i> for information about valid RTS/CTS configurations.

⇒ **Tip:**

You can verify communication status using the Studio development environment *Output* window (*LogWin* module). To establish an event log for **Field Read Commands**, **Field Write Commands**, and **Serial Communication** right-click in the *Output* window. When the pop-up menu displays, select the option to set the log events. If you are testing a Windows CE target, you can use the Remote LogWin of Studio (**Tools** → **Remote LogWin**) to get the log events from the target unit remotely.

If you are unable to establish communication with the PLC, try to establish communication between the PLC Programming Tool and the PLC. Quite frequently, communication is not possible because you have a hardware or cable problem, or a PLC configuration error. After successfully establishing communication between the device's Programming Tool and the PLC, you can retest the supervisory driver.

To test communication with Studio, 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**.
- **Studio Version**: To find this information, select **Help** → **About**.
- **Driver Version**: To find this information, read the full description of the driver on the *Communication Drivers* dialog.
- **Communication Log**: Displays in the Studio *Output* window (or *LogWin* window) when the driver is running. Be sure to enable the **Field Read Commands**, **Field Write Commands**, and **Serial Communication** for the *LogWin* window.
- **Device Model and Boards**: Consult the hardware manufacturer's documentation for this information.

Sample Application

You will find a sample application for drivers in the /COMMUNICATION EXAMPLES/<Driver Name> directory. We strongly recommend that you check if there is a sample application for this driver and use it to test the driver before configuring your own customized application, for the following reasons:

- To better understand the information provided in the section of this document.
- To verify that your configuration is working satisfactorily.
- To certify that the hardware used in the test (device, adapter, cable, and PC) is working satisfactorily before you start configuring your own, customized applications.



Note:

This application sample is not available for all drivers.

Use the following procedure to perform the test:

1. Configure the device's communication parameters using the manufacturer's documentation.
2. Open and execute the sample application.



Tip:

You can use the sample application screen as the maintenance screen for your custom applications.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of changes
A	1.00	Eric Vigiani	Mar/04/2004	Initial version.
B	2.00	André Körbes	Jul/05/2010	Added support for type conversion on addresses.
C	2.1	Anushree Phanse	May/09/2017	Driver version incremented. No change in the driver.