

IZCL Communication Driver

Driver for Serial or TCP/IP Communication
with Devices using IZAC/Clipper Network Protocol

Contents

INTRODUCTION2

GENERAL INFORMATION.....3

 DEVICE CHARACTERISTICS3

 LINK CHARACTERISTICS3

 DRIVER CHARACTERISTICS3

 CONFORMANCE TESTING4

INSTALLING THE DRIVER5

CONFIGURING THE DRIVER6

 SETTING THE COMMUNICATION PARAMETERS6

 CONFIGURING THE DRIVER WORKSHEETS8

EXECUTING THE DRIVER 12

TROUBLESHOOTING 13

SAMPLE APPLICATION 15

REVISION HISTORY..... 16

Introduction

The IZCL driver enables communication between the Studio system and devices using the IZAC/Clipper network protocol communicating over Serial or TCP/IP, according to the specifications discussed in this document.

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

- **Introduction:** Provides an overview of the IZCL driver documentation.
- **General Information:** Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the IZCL driver.
- **Installing the Driver:** Explains how to install the IZCL driver.
- **Configuring the Driver:** Explains how to configure the IZCL 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 IZCL 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 environment. If you are unfamiliar with Windows NT/2000/XP, 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 IZCL driver and the IZAC/Clipper PLC.

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:** Solidyne
- **Compatible Equipment:** Any device that is fully compatible with the IZAC/Clipper Network protocol
- **Programming Software:** None
- **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:** Serial or Ethernet Port
- **Physical Protocol:** RS232 or Ethernet
- **Logic Protocol:** IZAC/Clipper Network protocol
- **Specific PC Board:** Any TCP/IP Adapter (Ethernet board) for Ethernet communications

Driver Characteristics

The IZCL driver is composed of the following files:

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

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

- Windows NT/2K/XP

For a list of the operating systems used for conformance testing, see “Conformance Testing” on page 4.

The IZCL driver supports the following commands:

Command Type	Write	Read	Integer
04 (Request load states and override states)	–	•	•
06 (Request one second averaged analog sensor values)	–	•	•
0A (Request analog averages)	–	•	•
57 (Request virtual sensor values)	–	•	•
2C (Set loads)	•	–	•
2D (Reset loads)	•	–	•
40 (Unsolicited Messages)	–	–	•
44 (Unsolicited Alarm Dial-Out Messages)	–	–	•
65 (Write Virtual Sensor Values)	•	–	•
MO (Connect Modem)	•	–	•

Conformance Testing

The following hardware/software was used for conformance testing:

- **Driver Configuration:**
 - **Baud Rate:** 19200
 - **Data Bits:** 8
 - **Stop Bits:** 1
 - **Parity:** None
 - **COM Port:** COM1
- **Cable:** Ethernet cable

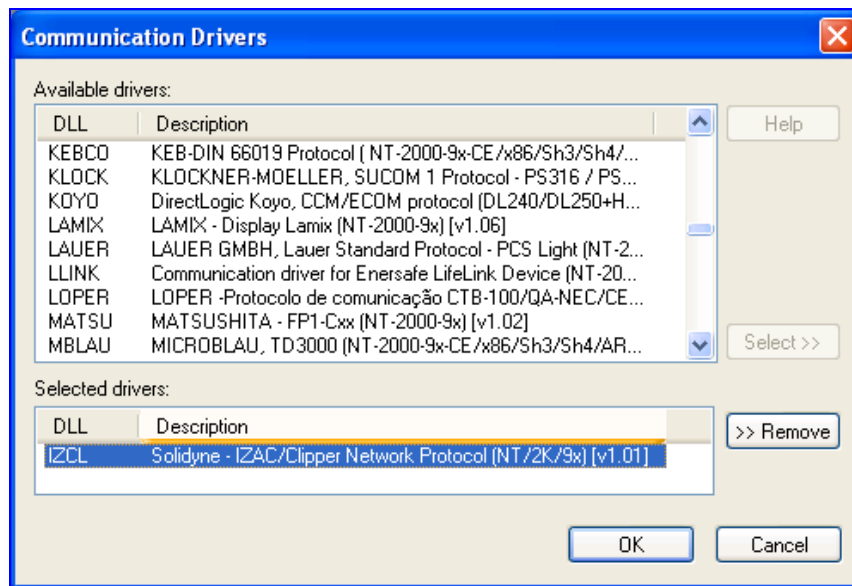
Driver Version	Studio Version	Operating System	Equipment
1.01	6.0	WinXP	IZAC-4

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 **IZCL** driver from the *Available Drivers* list, and then click the **Select** button.



Communication Drivers Dialog

5. When the **IZCL** 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 IZCL driver, you must configure the driver. Configuring the IZCL driver is done in two parts:

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

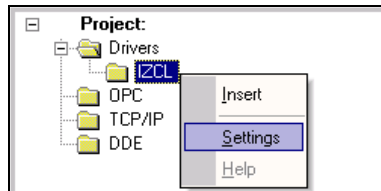
Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header**, and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header**,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

Setting the Communication Parameters

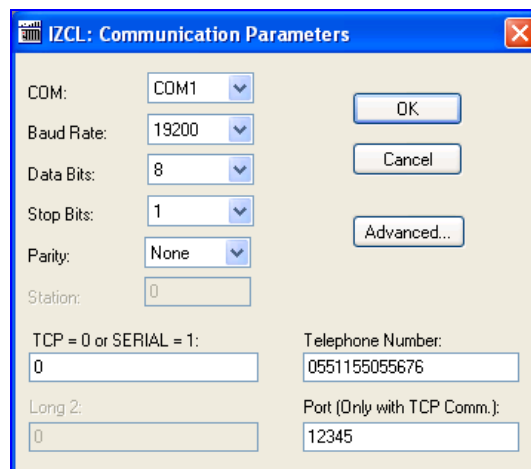
Use the following steps to configure the communication parameters, which are valid for all *Driver* worksheets configured in the system:

1. From the Studio development environment, select the **Comm** tab located below the *Workspace*.
2. Click on the *Drivers* folder in the *Workspace* to expand the folder.
3. Right-click on the IZCL subfolder and when the pop-up menu displays, select the **Settings** option:



Select Settings from the Pop-Up Menu


The *IZCL: Communication Parameters* dialog displays:




Communication Parameters Dialog

4. Specify the custom parameters as noted in the following table:

Parameters	Default Values	Valid Values	Description
Station	0	0	Not used for this driver
0: TCP 1: Serial	0	0 or 1	0: TCP/IP Communication 1: Serial Communication
Telephone Number	00000000000000	Any Number (up to 16 chars)	Telephone Number used in response to \$44 Command (Alarm Dial-Out). Set with the telephone number of the host (Computer)
Port (Only with TCP Comm.)	0	Any Number	Port used by PLC to send Alarm Dial-Out Messages and Unsolicited Messages to the host (Computer). Used only with TCP/IP Communications.

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

5. Click the **Advanced** button on the *Communication Parameters* dialog to open the *Advanced Settings* dialog and configure the settings that are necessary.

 **Notes:**

- Do not change any of the other *Advanced* parameters at this time. You can consult the *Studio Technical Reference Manual* for information about configuring these parameters for future reference.
- Generally, you must change the *Advanced* parameter settings if you are using a DCE (Data Communication Equipment) converter (232/485 for example), modem, and so forth between the PC, driver and the host. You must be familiar with the DCE specifications before adjusting these configuration parameters.

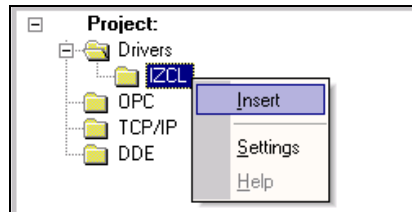
Configuring the Driver Worksheets

This section explains how to configure the *STANDARD DRIVER SHEETS* (or Communication tables) to associate application tags with the device addresses. You can configure multiple *Driver* worksheets—each of which is divided into a *Header* section and *Body* section.

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 <IZCL> subfolder.
3. When the pop-up menu displays, select the **Insert** option:



Inserting a New Worksheet

Note:

To optimize communication and ensure better system performance, you must tie the tags in different *Driver* worksheets to the events that trigger communication between each tag group and the period in which each tag group must be read or written. Also, we recommend configuring the communication addresses in sequential blocks to improve performance.

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

	Tag Name	Address	Div	Add
1	Tag[1]	0:1		
2	Tag[2]	0:2		
3	Tag[3]	0:3		
4	Tag[4]	1:14		
5	Tag[5]	1:25		

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*.

4. Use the following information to complete the **Station**, **Header**, and **Address** fields on this worksheet.

- **Station** field: Specify the device using the following syntax:
 <IP Address>:<Port Number>:<PLC ID> or <PLC ID>

Where:

- **IP Address** is the PLC IP address in the TCP/IP network (only to TCP/IP Communication).
- **Port Number** is the PLC Port number in the TCP/IP network (only to TCP/IP Communication). Port used by Host (Computer) to connect with PLC.
- **PLC ID** is the PLC identification number.

- **Header** field: Use the information in the following table to define the type of commands that will be read from or written to the device and a reference to the initial address. (Default value is 04.)

These variables must comply with the following syntax:

<Type> or <Type> :<NumSensor>

Where

<Type>:

- 04 Request load states and override states
- 06 Request one second averaged analog sensor value
- 0A Request analog averages (past 48 hours, 15 minute averages)
- 57 Request virtual sensor values
- 2C Set loads
- 2D Reset loads
- 40 Unsolicited Messages (sensor values and load/override state)
- 44 Unsolicited Alarm Dial-Out Messages
- 65 Write Virtual Sensor Values
- MO Connect Modem

<NumSensor>: 1 – 8 Sensor number (only with 0A Type)

After you edit the **Header** field, Studio checks the syntax to determine if it is valid. If the syntax is incorrect, Studio automatically inserts the default value in the **Header** field.

Also, you can type a tag string in brackets {**Tag**} into the **Header** field, but you must be certain that the tag's value is correct and that you are using the correct syntax, or you will get an invalid Header error.

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

- **Address** field: Use this field to associate each tag to its respective device address.
Type the tag from your application database into the **Tag Name** column. This tag will receive values from or send values to an address on the device. The address must comply with the following syntax:

<NN> or <ZZ> : <YY>

Type 04: <ZZ> : <YY> where

<ZZ> = 0 or 1 (0 – load, 1 – override)

<YY> = 1 – 32 load/override states number

Type 06: <NN> where

<NN> = 1 - 8 analog sensor number

Type 0A: <NN> where

<NN> = 1 – 192 analog averages (1 – first 15 minutes averages, 192 – last 15 minutes averages)

Type 57: <NN> where

<NN> = 1 - 8 virtual sensor number

Type 2C: <NN> where

<NN> = 1 - 32 loads state number (1's values indicate loads to be set)

Type 2D: <NN> where

<NN> = 1 - 32 loads state number (1's values indicate loads to be reset)

Type 40: <ZZ> : <YY> where

<ZZ> = 0 - 4 (0-analog sensor, 1-loads, 2-override, 3-global override start, 4-global override cancel)

<YY> = 1 - 8 analog sensor number

<YY> = 1 - 32 loads states number

<YY> = 1 - 16 override states number

<YY> = 1 - 8 global override starts number

<YY> = 1 - 8 global override cancels number

Type 44: <NN> where

<NN> = 1 - 12 1 - Alarm hour (integer),

2 - Alarm minute (integer),

3 - Alarm period (AM=0, PM=1) (integer),

4 - Alarm month (integer),

5 - Alarm day (integer),

6 - NODE number (integer),

7 - LAN ID (string),

8 - Sensor number (integer),

9 - Sensor Value (integer),

10 - Load number (integer),

11 - Load hours (integer) and

12 - Load minutes (integer).

Type 65: <NN> where
<NN> = 1 - 8 virtual sensor number

Type MO: <NN> where
<NN> = Telephone number

 **Attention:**

- Types 04, 06, 0A and 57 can be used only with Read commands.
- Types 2C, 2D and 65 can be used only with Write commands.
- Types 40 and 44 are used only with Unsolicited Messages.
- Type MO is to connect to a modem. You must use the WriteGroup command.
 - If the connection of MO (Modem Connect) is OK, the Status of this DriverSheet will be 0.
 - If the line is BUSY, the status will be 10.
 - If there is No Dial Tone, the status will be 11.

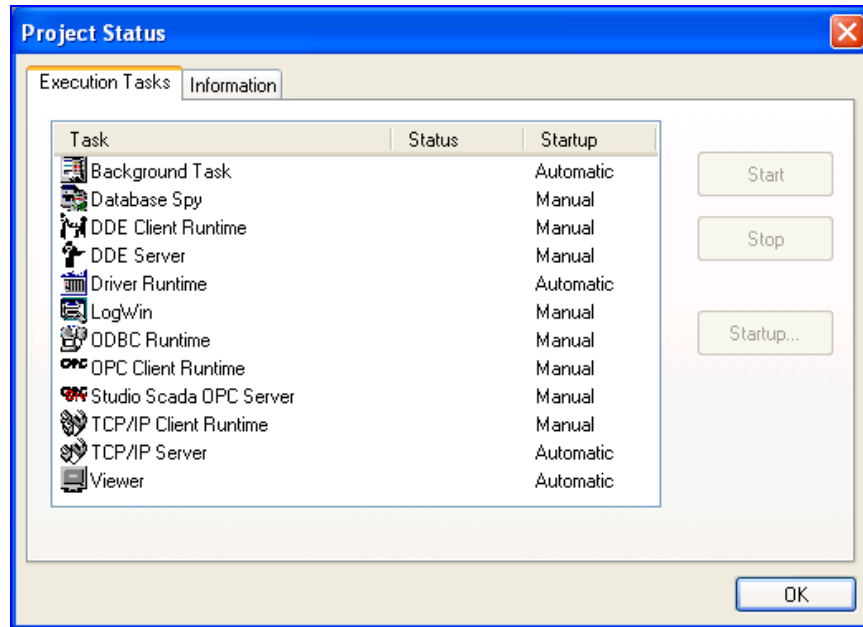
Executing the Driver

After adding the IZCL 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 IZCL 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	Check Sum Error	Protocol Error	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check PLC.
2	Check Sum Error in Unsolicited Message	Protocol Error	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check PLC.
3	Cannot Read or Write with this operand	Read or Write Operands don't support this command	
4	Too much Unsolicited Message	Too many unsolicited Messages coming	
5	Error Connect	Problems with Ethernet communications	Check Ethernet cable and IP number.
6	Wrong ID destination Number	Wrong ID destination Number	
7	Address out of range	Check address configured in the driver sheets	
10	MODEM – BUSY	Phone line is busy	
11	MODEM – NO DIAL TONE	Problem with phone line	
-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 the <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 configuration. ▪ Check the configuration. See the <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**/*<IZCL>* 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 each 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	José Lourenço Teodoro and Fábio H. Y. Komura	Sep/19/2003	Initial version
B	1.01	Fábio H. Y. Komura	Jun/25/2004	<ul style="list-style-type: none">Implemented \$44 command (Alarm Dial-Out)Implemented \$65 command (Write Virtual Sensor Values)