JETTE Communication Driver

Driver for Communication with
JETTER Devices Using the Jet32 API

Contents

Introduction

The JETTE driver enables communication between the Studio system and JETTER devices using the PCOMX protocol over the **Jet32** API, according to the specifications discussed in this document.

This document will help you to select, configure and execute the JETTE driver, and it is organized as follows:

- **Introduction**: This section, which provides an overview of the document.
- **General Information**: Identifies all of the hardware and software components required to implement communication between the Studio system and the target device.
- **Selecting the Driver**: Explains how to select the JETTE driver in the Studio system.
- **Configuring the Device**: Describes how the target device must be configured to receive communication from the JETTE driver.
- **Configuring the Driver**: Explains how to configure the JETTE driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver**: Explains how to execute the JETTE driver during application runtime.
- **Troubleshooting**: Lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.
- **Revision History**: Provides a log of all changes made to the driver and this documentation.

> **Notes:**
> – This document assumes that you have read the “Development Environment” chapter in Studio’s *Technical Reference Manual*.
>
> – This document also assumes that you are familiar with the Microsoft Windows XP environment. If you are not familiar with Windows, then we suggest using the **Help** feature (available from the Windows desktop **Start** menu) as you work through this guide.

General Information

This chapter identifies all of the hardware and software components required to implement serial communication between the JETTE driver in Studio and Jetter devices.

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**: Jetter
- **Compatible Equipment**:

  PASE-E+
  Mikro
  Delta, Delta-CPU2
  Nano-A, Nano-B, Nano-C, Nano-D
  PC-PPLC
  JetControl24x, JetControl64x

- **Jetter PLC programmer software**: Jetter JetSym

This driver has been tested successfully with Jetter NANO-B over RS-232 Serial Connection. (For a list of the devices used for conformance testing, see “Conformance Testing” on page 4.)

Network Specifications

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

- **Device Communication Port**: Selectable
- **Physical Protocol**: RS-232, Ethernet, RS-485 (JETWay)
- **Logic Protocol**:
  - Serial interface - PCOM3 and PCOM5
  - JETWay and PC-PPLC board unit for ESA slot
  - TCP/IP - JetIP IP, PCOM6 onwards.
- **Device Runtime Software**: Jet32.dll API provided by Jetter. This files needs to be installed on your Studio installation folder under the `\DRV\API\` subfolder
- **Specific PC Board**: in case of JETWay or PCPPLC, a compatible Jetter board
- **Cable**: Jetter cables

Driver Characteristics

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

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

- **`Jet32.DLL:`** Jetter API. It must be on the **`/DRV/API`** subdirectory of Studio. This file is provided by Jetter

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

- Windows XP

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

The JETTE driver supports the following registers:

Register Type	Length	Write	Read	Bit	Integer	Float	String
REGISTER	4 Bytes	•	•	—	•	—	—
FLOAT	4 Bytes	•	•	—	—	•	—
FLAG	1 Bit	•	•	•	—	—	—
TEXT	Up to 127 Characters	•	•	—	—	—	•

Conformance Testing

The following hardware/software was used for conformance testing:

Driver Configuration:

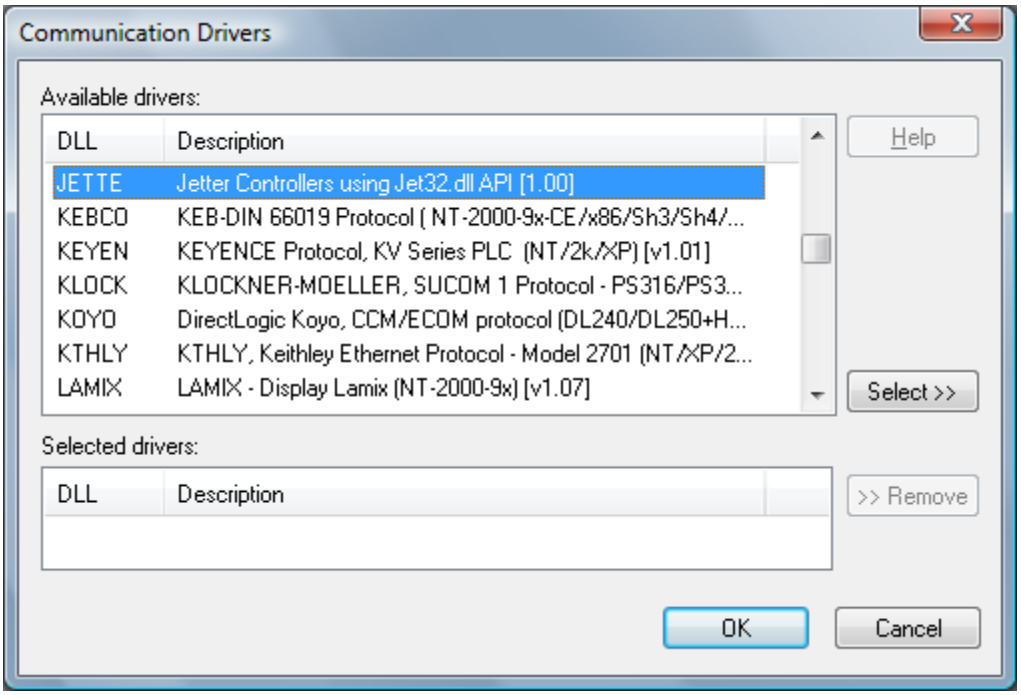
- **Serial**:
  - **Cable**: Jetter EM-PK 9-Pin Serial Cable
  - **Baud Rate**: 9600
  - **Data Bits**: 8
  - **Stop Bits**: 1
  - **Parity**: None

Driver Version	Studio Version	Operating System (development)	Operating System (runtime)	Equipment
1.0	6.1+SP6	Windows XP + SP3	Windows XP + SP3	NANO-B CPU

Selecting the Driver

When you install Studio, all of the communication drivers are automatically installed in the **\DRV** subdirectory but they remain dormant until manually selected for specific applications. To select the JETTE driver for your Studio application:

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



Communication Drivers Dialog

3. When the **JETTE** driver is displayed in the **Selected Drivers** list, click the **OK** button to close the dialog. The driver is added to the *Drivers* folder, in the *Comm* tab of the Workspace.

> **Note:**
>
> It is not necessary to install any other software on your computer to enable communication between Studio and your target device.
>
> Also, this communication can only be used by the Studio application; it cannot be used to download control logic to the device. To download control logic to an Jetter device, you must also install the Jetter programming software (e.g., JetSym). For more information, please consult the documentation provided by the device manufacturer.

> **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 Device

Consult your Jetter documentation for information about configuring your device.

Configuring the Driver

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

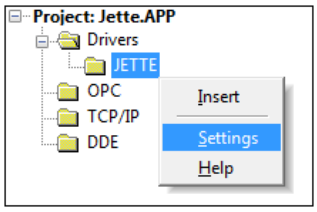
Configuring the Communication Settings

The communication settings are described in detail in the "Communication" chapter of the 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 JETTE driver-specific settings and procedures will be discussed here. To configure the communication settings for the JETTE driver:

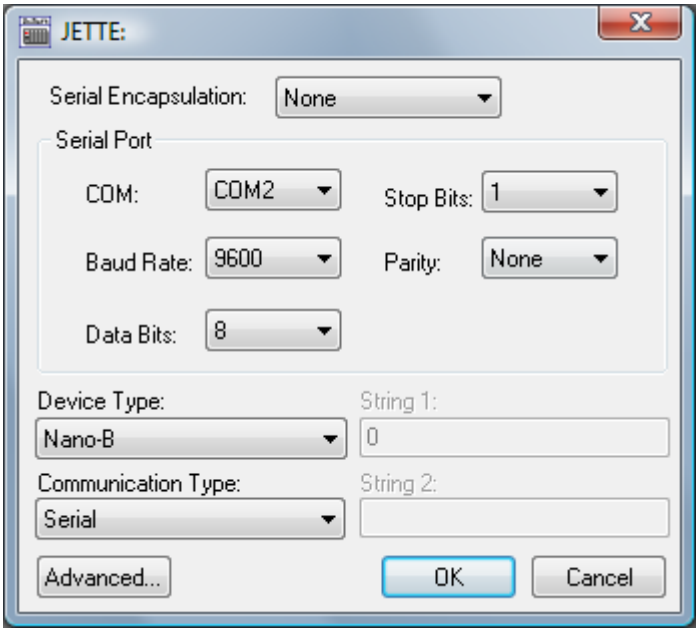
1. In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The JETTE driver is listed here as a subfolder.

2. Right-click on the *JETTE* subfolder and then select the **Settings** option from the pop-up menu:



Select Settings from the Pop-Up Menu

The *JETTE: Communication Settings* dialog is displayed:



JETTE: Communication Settings Dialog

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

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

> ➲ **Attention:**
>
> For safety reasons, you **must** take special precautions when connecting and configuring new equipment. Please consult the manufacturer's documentation for specific instructions.

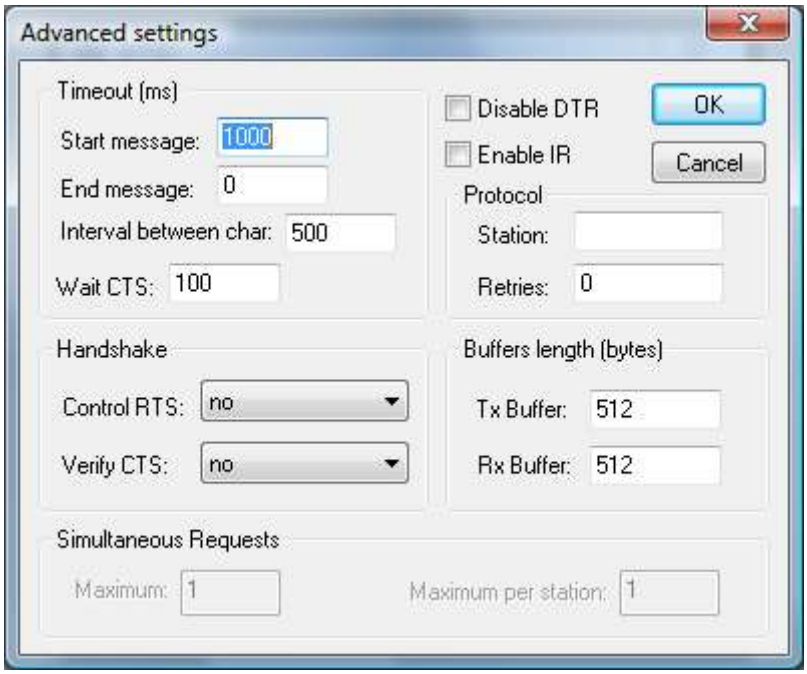
The communication settings and their possible values are described in the following table:

Parameters	Default Values	Valid Values	Description
Serial Port – COM	`COM2`	`COM1 – COM8`	Although you are able to select any COM port number on this interface, this driver works only with a PORT within the COM1-COM8 range

Device Type	`Nano-B`	`PASE-E+`
		`Mikro`
		`Delta`
		`Nano-A`
		`Nano-B`
		`Nano-C`
		`PCPPLC`
		`Nano-D`
			CPU type that you are connecting with.
			In case you want to communicate with more than 1 CPU type, you can configure that on the ***Station*** field on the driver Sheets, and the CPU configured here will be the default CPU

		`Delta-CPU2` **`JetControl24x`** **`JetControl64x`**	
Communication Type	**`Serial`**	**`Ethernet`** **`Serial`** **`JETWay`** **`PCPPLC`**	Communication mode. **Ethernet**: TCP/IP - JetIP **Serial**: RS-232 point-to-point Serial connection (PCOM3 and PCOM5) **JetWay**: Jetter RS-485 network connection. Requires a compatible Jetter PC card **PCPPLC**: Connection with PC-PPLC CPU using a compatible Jetter PC-PPLC PC Card

4. Click on the **Advanced…** button in the *Communication Parameters* dialog. The *Advanced settings* dialog will display.



Configure the parameters on this dialog as appropriate for your worksite. Consult the *Studio Technical Reference Manual* for instructions about configuring this dialog.

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 may include 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 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 JETTE driver-specific parameters and procedures are discussed here.

Main Driver Sheet

The JETTE driver does not support Main Driver Sheet in this version

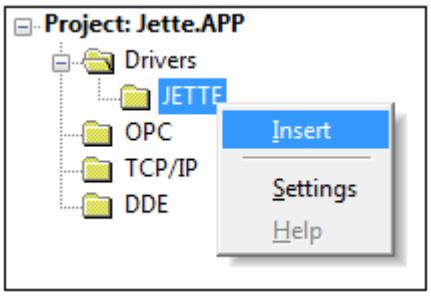
Standard Driver Worksheet

When you select the JETTE 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.

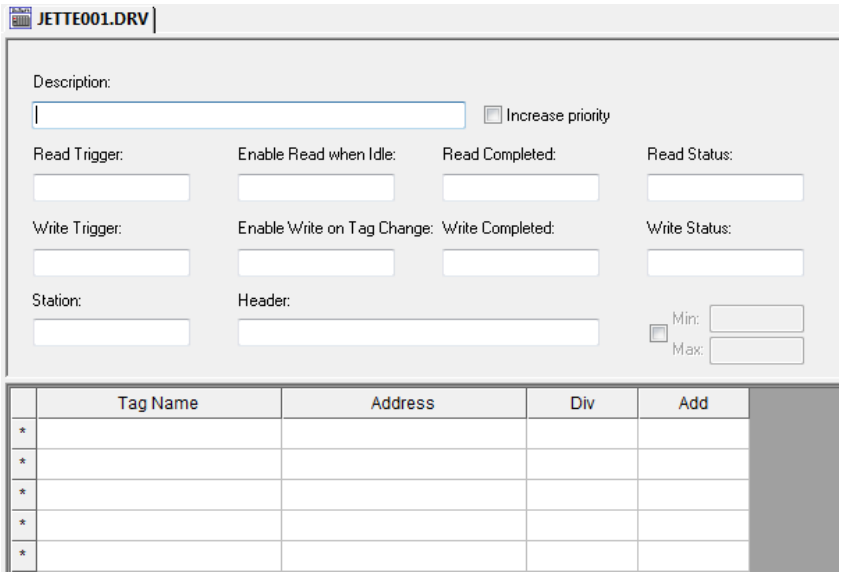
To insert a new Standard Driver Worksheet:

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



Inserting a New Worksheet

A new JETTE driver worksheet is inserted into the *JETTE* subfolder, and the worksheet is open for configuration:



JETTE Driver Worksheet

> **Note:**
> Worksheets are numbered in order of creation, so the first worksheet is **`JETTE001.drv`**.

Most of the fields on this worksheet are standard for all drivers; see the "Communication" chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station**, **Header**, and **Address** fields use syntax that is specific to the JETTE driver.

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

   - **Station** field:

     On the **Station** field you will specify the Device address that you are going to communicate with. Optionally, you can configure here also a **Device Type** if this specific worksheet will communicate with a Device different than the one configured on the *Communication Settings* window. In this case, you must start the **Station** field value with the **Device Type**, separating it from the rest of the Device Address by ***Colon*** **" : "**

     Valid Device Types: **`PASE-E+, Mikro, Delta, Nano-A, Nano-B, Nano-C, PCPPLC, Nano-D, Delta-CPU2, JetControl24x, JetControl64x`**

     Example: `Delta-CPU2:192.168.10.63`

              `JetControl24x:1:340`

   - The Device address uses the following syntaxes according to the communication type:

     - For Ethernet Communication:

       <IPAddress>:<optPortNumberSend>:<optPortNumberReceive>

       For example:

       `192.168.2.4`

       `Delta-CPU2:192.168.10.63`

       Where:

       – ***<IP Address>*** is the IP address of the device on your Ethernet network;

       – ***<optPortNumberSend >*** Optional TCP/IP port number used to send messages to the device. If this parameter is omitted, the default port used by the Jet32.dll is used, which is 50000;

       – ***<optPortNumberReceive>*** Optional TCP/IP port number used to receive messages from the device. If this parameter is omitted, the default port used by the Jet32.dll is used, which is 50000;

     - For Serial Communication:

       Nothing

     - For JetWay and PC-PPLC:

       <Slave Number>:<Port Address>

       For example, **`1:340`**

       Where:

- ***<Slave Number>*** is the Slave Number that you are connecting to in the JetWay network; The valid values are from 0 to 127.For PC-PPLC, you can use **220** for controller on the board itself
- ***<Port Address>*** is a String with a Number in Hexadecimal Format with the port number of the JetWay PC-Card that you configure using the DIP Switches in the Card itself. The valid values are 300 to 360

- **Header** field: Specify the type of Register that you are going to be accessing in this worksheet

  The Header field uses the following syntax:

  <Register Type>

  For example, `Float`

  Where:
  - ***<Area>*** is the Register Type.

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

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

Register Types	Comments
REGISTER	Register in Signed Integer format
FLOAT	Register in Floating Point format
TEXT	Registers in String format (up to 127 characters)
FLAG	Register in Boolean format (0 or 1)

4. **Address**: For each table row (i.e., each tag/register association), configure the **Address** field with the Register number.

   You can have access to individual bits inside of a Register number. The syntax for that is:

   <Register>.<Bit>

   The bits go from 0 to 23

➲ **Attention:**
The bit write is implemented using the following steps:
1. Read the **`REGISTER`** from the PLC
2. Modify the bit specified
3. Write the value back

When using this functionality, you can get multiple bits changed by the write command in case the device modifies the value right after the driver reads the current value on step 1.
Write Trigger is not enabled for bit writing.

Attention:

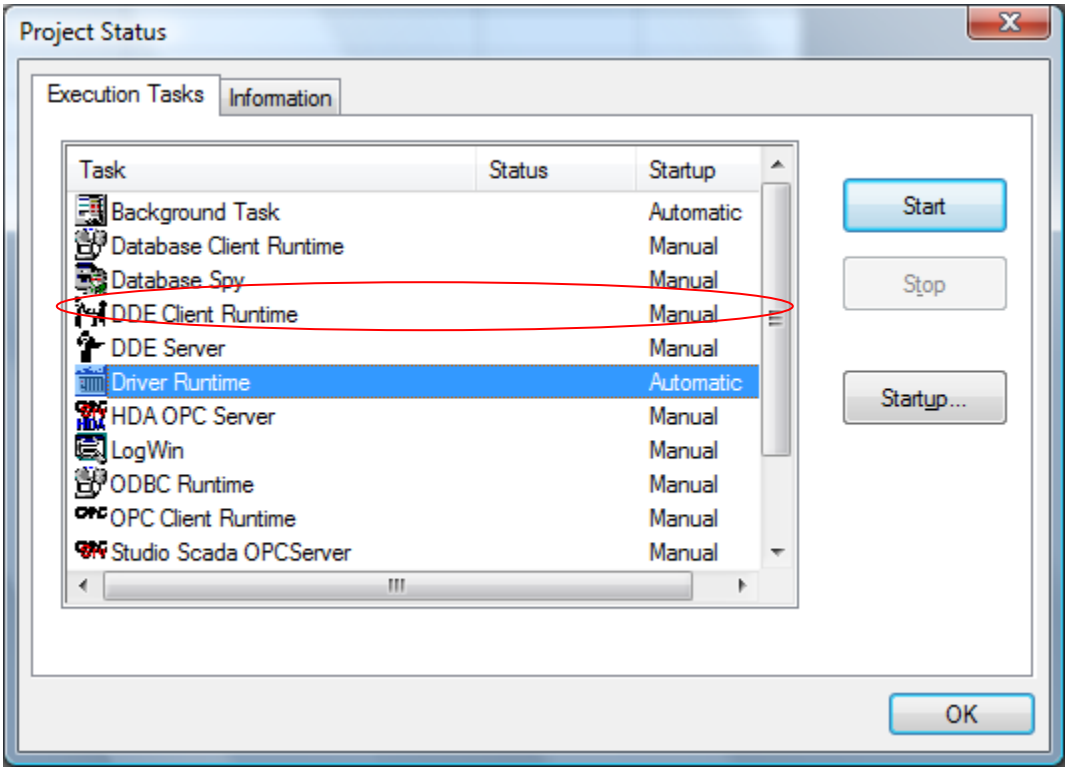
- You must not configure a range of addresses greater than the maximum block size (data buffer length) supported within the same worksheet which is 300 registers per worksheet.

Executing the Driver

By default, Studio will automatically execute your selected communication driver(s) during application runtime. However, you may verify your application's runtime execution settings by checking the *Project Status* dialog.

To verify that the the communication driver(s) will execute correctly:

1. From the main menu bar, select **Project** → **Status**. The *Project Status* dialog displays:



Project Status Dialog

2. Verify that the *Driver Runtime* task is set to **Automatic**.
   - If the setting is correct, then proceed to step 3 below.
   - If the **Driver Runtime** task is set to **Manual**, then select the task and click the **Startup** button to toggle the task's *Startup* mode to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.
4. Start the application to run the driver.

Troubleshooting

If the JETTE 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	Not applicable
1	Could not load library	API file (Jet32.dll) was not found or this file is corrupted	Reinstall the driver
2	API error	Generic Error related to the Jet32.dll API	Check Protocol Analyzer to get Error Code and follow procedure described on the API error table
3	General Error	Internal Driver error	Unexpected Error – please contact Technical Support
-37	Invalid Header	This usually happens when you configure a string Tag in the **Header** field and the value of this tag does not comply with the header syntax	Change the value of the tag in the **Station** field to one that complies with the **Header** syntax
-38	Invalid Station	Invalid station specified in Station field	Specify a valid station in the Driver Worksheet.
-39	Block Size Error	Offset between the highest and the lowest register configured in the driver worksheet is greater than **300**	Reconfigure the driver worksheet so the offset between the highest and the lowest address is equals to or less than 300
-15	Timeout Start Message	▪ Disconnected cables ▪ Wrong station number	▪ Check the cable wiring. ▪ Check the station number.

⇨ **Tip:**

You can monitor communication status by establishing an event log in Studio's *Output* window (*LogWin* module). To establish a log for **Field Read Commands**, **Field Write Commands** and **Serial Communication,** right-click in the *Output* window and select the desired options from the pop-up menu.

You can also use the *Remote LogWin* module (**Tools** → **Remote LogWin**) to establish an event log on a remote unit that runs Windows CE.

Error Code(Hex)	Description	Possible Causes	Procedure to Solve
0	OK	Communication without problems	Not applicable
20001002	Illegal controller type	Invalid controller type was configured in Settings	Reinstall the driver. Make sure that a valid controller was specified
20001003	Illegal baud rate	A illegal value of baud rate was configured	Configure a valid baud rate in driver Settings
20001004 20001010	Illegal timeout	▪ Disconnected cables ▪ Wrong station number	▪ Check the cable wiring. ▪ Check the station number.

200014E			
20001005	Cannot detect controller	The controller type has not been be detected during a connection attempt. This error code should occur only if there are old Mikro versions in use.	Check cables and connections
20001006	Different Controller in use	The controller type detected differs from the one entered in Settings or in Station field	Check the device type and configure correctly settings or station field
20001007	Out of memory	RAM memory has exceeded when attempting to allocate some inside the DLL.	Close all open programs and try again. If it persists, reboot the computer and try again
2000100B	Serial port not available	The serial port number is not available.	Close any program that may be using the serial port or configure another serial port
2000100C	Illegal port number	An illegal serial port number has been configured. Port numbers available are going from COM1 to COM8.	Configure in Settings a port number from COM1 to COM8
20001015 20001033	Illegal IO port	An illegal IO port number has been configured for Jetway or PCPPLC connections	Configure a valid IO port number in Settings or in Station field. I/O-port numbers available are going from 0x300 up to 0x360 for Jetway and or PCPPLC connections
20001016 20001034	board not found	Unable to find a board unit at port address specified	Configure a valid board unit in Settings or in Station field.
2000100D 20001018 20001036 2000104C	controller offline	The controller connected did not respond to the request.	Check cables, connections and if the PLC is not in error mode
20001019 20001037	Jetway no response	The JETWay or PCPPLC board unit did not respond to the request.	▪ Check if board unit is properly configured
2000101A 20001038	Cannot open the device	Kernel driver is unable to open JETWay / PCPPLC board device. This error code only happens if the Jet32.dll is used with the unsupported operating systems Windows NT 4.0 and Windows 2000.	Upgrade the Operating System to Windows XP
2000101B 20001039	Device driver not found	Kernel driver of JETWay or PCPPLC board device cannot be found in registry. This error code only occurs when Jet32.dll is used with the unsupported operating systems Windows NT 4.0 and Windows 2000.	Upgrade the Operating System to Windows XP
20001021	Illegal block size	An illegal block size value been entered in SDS. Because driver treat this error, it should never occur	Internal Driver Error – Please contact Technical Support
20001029	Value not written	Failure when attempting to write a value to the Controller	▪ Check cables and connections. ▪ Restart the driver and Jetter PLC
20001046	IP load lib failed	Attempt to load dynamic link library	Reinstall the driver

		"JetIP32.dll" failed. In most cases this file needed for Ethernet communication (JetIP)	
20001047	IP wrong lib failed	The dynamic link library "JetIP32.dll" has an invalid format or is corrupted.	▪ Contact support to get the latest version of JetIP32.dll ▪ Reinstall the driver
20001048	Ethernet initialization failed	Failure during initialization of Ethernet communication	Restart the driver and Jetter PLC
20001051	Illegal IP port number	An illegal IP port has been configured.	Configure a valid IP port in Station field.

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 (e.g., JetSym). Quite often, communication is interrupted by a hardware or cable problem or by a device configuration error. If you can successfully communicate using the programming software, then recheck the driver's communication settings in Studio.

If you must contact us for technical support, please have the **Support Information** available - **Help → Support Information**.

- **Driver Version** and **Communication Log**: Displays in the Studio *Output* window or *Remote LogWin* when the driver is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of Changes
A	1.00	Fellipe Peternella	26-Jan-2010	Initial version
B	1.1	Lourenco Teodoro	17-Feb-2010	Added support to writing single bits from a REGISTER Added access to REGISTERs from 10000 to 65279