VMASC Communication Driver

Driver for Serial and TCP/IP Communication
with Devices Using VisualMotion Protocol

Contents

Introduction

The VMASC driver enables communication between the Studio system and devices using the VisualMotion protocol communicating over Serial and TCP/IP communication, according to the specifications discussed in this document.

This document will help you to select, configure and execute the VMASC 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 VMASC driver in the Studio system.
- **Configuring the Device**: Describes how the device must be configured to receive communication from the VMASC driver.
- **Configuring the Driver**: Explains how to configure the VMASC driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver**: Explains how to execute the VMASC 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 NT/2000/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 and TCP/IP communication between the VMASC driver in Studio and a target device using the VisualMotion protocol.

The information is organized into the following sections:

- Device Specifications
- Network Specifications
- Driver Characteristics
- Conformance TCP/IP Testing

Device Specifications

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

- **Manufacturer**: Bosch RexRoth
- **Compatible Equipment**:

  - Bosch Rexroth Indramat PPC-R and any device that is fully compatible with the VisualMotion protocol

For a description of the device(s) used to test driver conformance, see ***Conformance TCP/IP Testing*** on page 5.

Network Specifications

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

- **Device Communication Port**: COM port (X16) / VisualMotion Ethernet Port
- **Physical Protocol**: RS232 / Ethernet/TCP-IP
- **Logic Protocol**: VisualMotion protocol
- **Specific PC Board**: None / Any TCP/IP Adapter (Ethernet board)

Driver Characteristics

The VMASC driver is composed of the following files:

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

Note:
You must use Adobe Acrobat® Reader™ to view the **`VMASC.PDF`** document. You can install Acrobat Reader from the Studio installation CD, or you can download it from Adobe's Web site.

The VMASC driver requests the **`DrvStd.DLL`** into the **`/BIN`** `folder.`

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

- Windows NT/2000/XP
- Windows CE

The VMASC driver supports the following commands:

Parameters	
Command Class	**Command Subclass**
A - Axis Parameters	A - Attributes
	B - Block List Parameter
	D - Lists/Tables
	H - Upper Limit
	L - Lower Limit
	P - Parameter Data
	T - Name Text
	U - Units Text
C –Control Parameters	(Same Subclasses as Axis Parameters)
D - SERCOS Drive Parameters	(Same Subclasses as Axis Parameters)
T - Task Parameters	(Same Subclasses as Axis Parameters)

Variables	
Command Class	**Command Subclass**
F - Float Variables	P - Data
	T - Text Label
G - Global Integer Variables	(Same Subclasses as Float Variables)
H - Global Float Variables	(Same Subclasses as Float Variables)
I - Integer Variables	(Same Subclasses as Float Variables)

I/O Registers	
Command Class	**Command Subclass**
R - I/O Registers	B - Current State in Binary
	C - Forcing State Change
	D - Current State in Decimal
	E - Erase all Forcing Masks
	F - Forcing Mask

	S - Binary Forcing State
	T -NameText
	X - Current State in Hexadecimal

Conformance TCP/IP Testing

The following hardware/software was used for conformance testing:

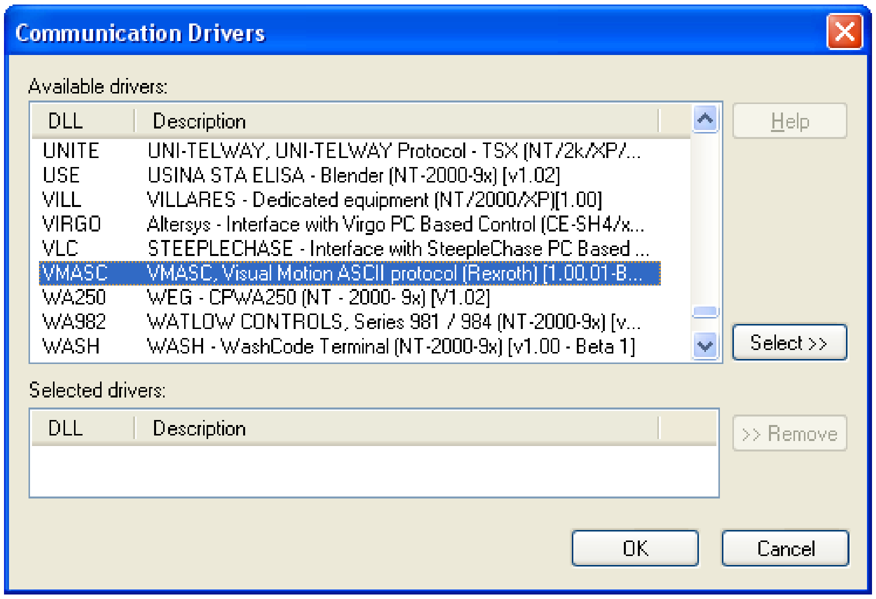
- **Configuration**:
  IP Address: 10.168.23.79
  Port Number: 5001

Driver Version	Studio Version	Operating System (development)	Operating System (target)	Equipment
1.00	6.1 + SP2	WinXP+SP2	WinXP+ SP2 WinCe 5.0 X86	PPC-R22.1

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 VMASC driver for your Studio application:

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



DLL	Description
UNITE	UNI-TELWAY, UNI-TELWAY Protocol - TSX (NT/2k/XP/...
USE	USINA STA ELISA - Blender (NT-2000-9x) [v1.02]
VILL	VILLARES - Dedicated equipment (NT/2000/XP)[1.00]
VIRGO	Altersys - Interface with Virgo PC Based Control (CE-SH4/x...
VLC	STEEPLECHASE - Interface with SteepleChase PC Based ...
VMASC	VMASC, Visual Motion ASCII protocol (Rexroth) [1.00.01-B...
WA250	WEG - CPWA250 (NT - 2000- 9x) [V1.02]
WA982	WATLOW CONTROLS, Series 981 / 984 (NT-2000-9x) [v...
WASH	WASH - WashCode Terminal (NT-2000-9x) [v1.00 - Beta 1]

DLL	Description

Communication Drivers Dialog

3. When the **VMASC** 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 the host and the device. Consult your VisualMotion software documentation for installation instructions.

Attention:
For safety reasons, you must take special precautions when installing any physical hardware. Please consult the manufacturer's documentation for specific instructions.

Configuring the Driver

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

- Specifying communication parameters
- Defining tags and controls in the *MAIN* and *Standard Driver* worksheets (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 *MAIN* and *Standard Driver* worksheets, and information about configuring the standard fields, review the product's *Technical Reference Manual*.

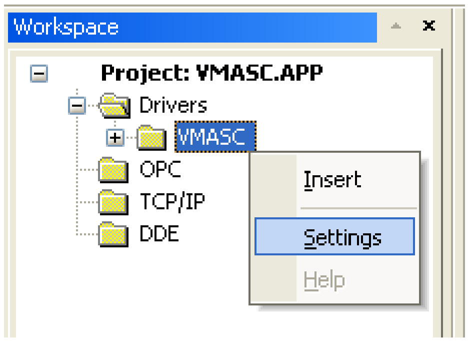
Once you have selected the VMASC driver in Studio, you must properly configure it to communicate with your target device. First, you must set the driver's communication settings to match the parameters set on the device. Then, you must build driver worksheets to associate database tags in your Studio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

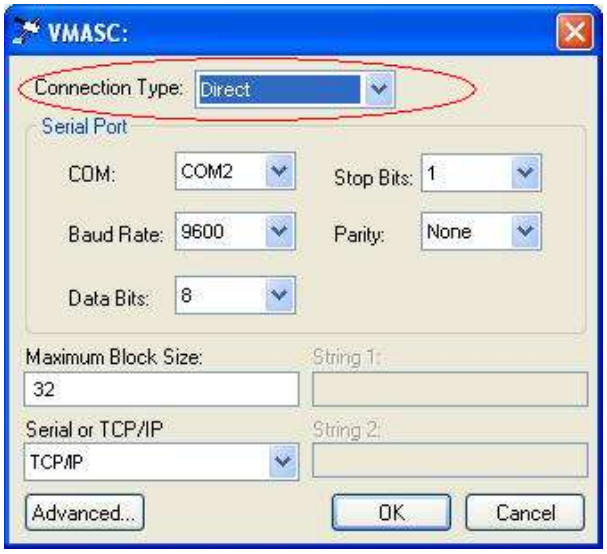
For the purposes of this document, only VMASC driver-specific settings and procedures will be discussed here. To configure the communication settings for the VMASC driver:

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

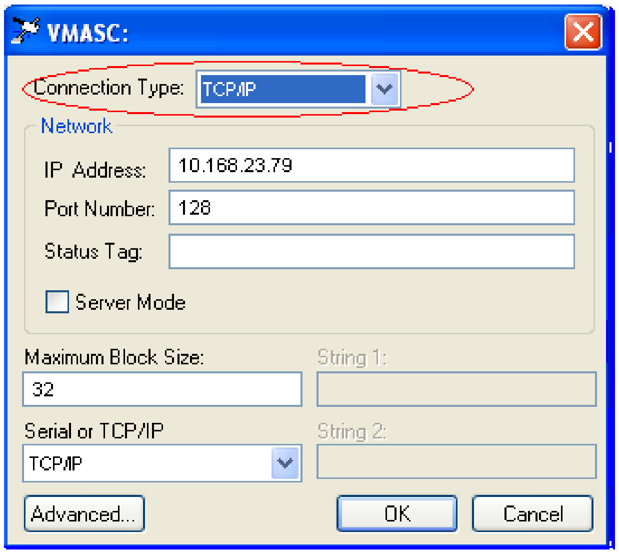


Select Settings from the Pop-Up Menu

The VMASC: *Communication Settings* dialog is displayed:



Settings for Direct (Serial) Connection Type



Settings for TCP/IP Connection Type

Different communication settings are displayed according to the connection type selected: **Direct** (Serial) or **TCP/IP**. To change the connection to match the VisualMotion device, click on the **Connection Type** combo box and select the correct type from the menu.

3. Proceed to configure the remaining 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.

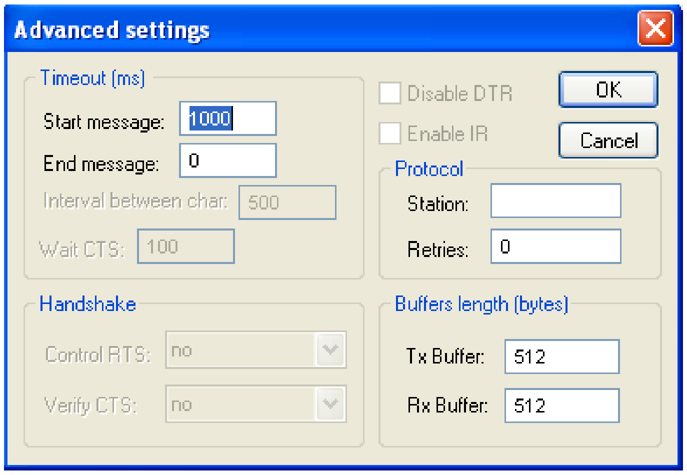
   Depending on your circumstances, you may need to configure the driver *before* you have configured your target device. If this is the case, then take note of the driver settings and have them ready when you later configure the device.

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

4. Configure the additional driver-specific settings, as described in the following table:

Setting	Default Value	Valid Values	Description
Maximum Block Size	32	`1 to 255`	This parameter is the maximum number of the lines in a worksheet that the driver will accept. **Note:** When the driver reads from the device and it is using the Standard Driver Sheet, the driver will wait the answer of all address configured in the worksheet before updating the values in the tags.

5. If you are using a Data Communication Equipment (DCE) converter (e.g., 232/485) between your PC and your target device, then you must also adjust the **Control RTS** (Request to Send) setting to account for the converter. In the *Communication Settings* dialog, click the **Advanced** button to open the *Advanced Settings* dialog:


Advanced settings
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

When the dialog is displayed, configure the **Control RTS** setting using the following information:

Setting	Default	Values	Description
Control RTS	no	no	Do not set the RTS (Request to Send) handshake signal. IMPORTANT: If you are using Windows 95/98 or Windows CE with the correct RS232/RS485 adapter (i.e. without RTS control), then you must select this option.
		yes	Set the RTS (Request to Send) handshake signal before communication. IMPORTANT: If you are using Windows NT and the Cutler-Hammer RS232/RS485 adapter, then you must select this option.
		yes+echo	Set the RTS (Request to Send) handshake signal before communication, and echo the signal received from the target device.

➲ **Attention**:

If you incorrectly configure the **Control RTS** setting, then runtime communication will fail and the driver will generate a –15 error. See “Troubleshooting” for more information.

You do not need to change any other advanced settings at this time. You can consult the Studio *Technical Reference Manual* later for more information about configuring these settings.

6. 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 Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

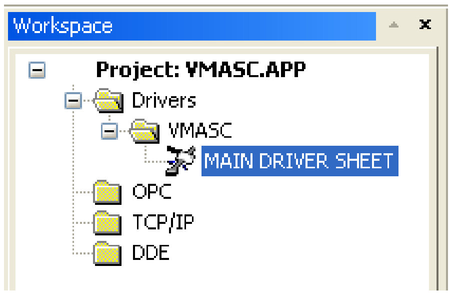
For the purposes of this document, only VMASC driver-specific parameters and procedures are discussed here.

MAIN DRIVER SHEET

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

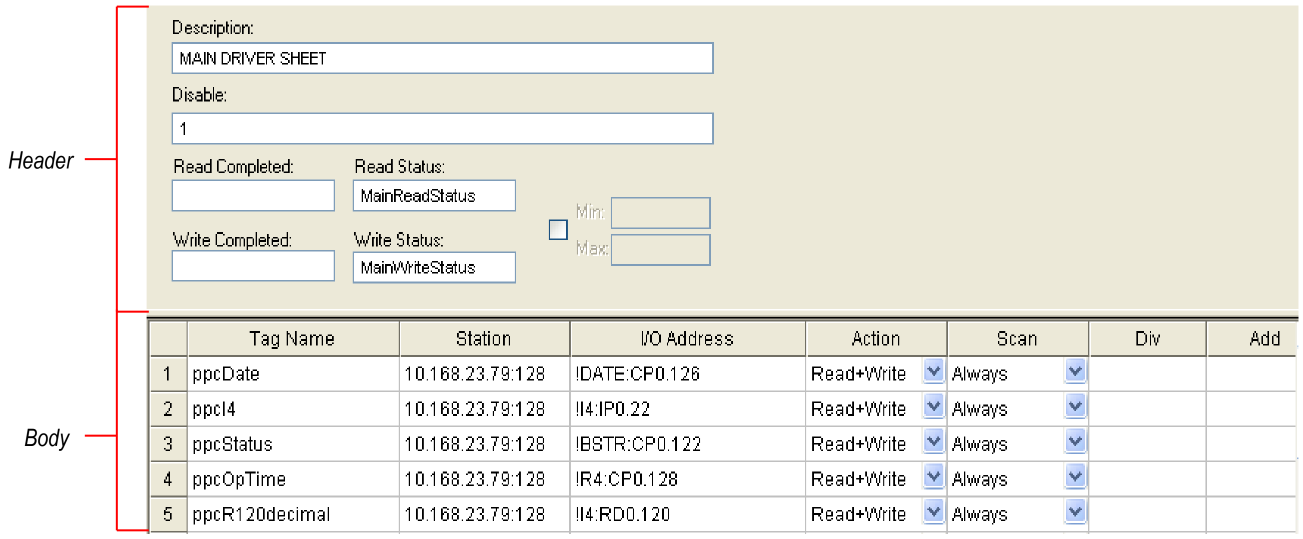
1. Select the *Comm* tab in the *Workspace* pane.

2. Open the *Drivers* folder, and then open the VMASC subfolder:



Main Driver Sheet in the VMASC Subfolder

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



Description: MAIN DRIVER SHEET
Disable: 1
Read Completed:
Read Status: MainReadStatus
Write Completed:
Write Status: MainWriteStatus

	Tag Name	Station	I/O Address	Action	Scan	Div	Add
1	ppcDate	10.168.23.79:128	!DATE:CP0.126	Read+Write	Always		
2	ppcI4	10.168.23.79:128	!I4:IP0.22	Read+Write	Always		
3	ppcStatus	10.168.23.79:128	!BSTR:CP0.122	Read+Write	Always		
4	ppcOpTime	10.168.23.79:128	!R4:CP0.128	Read+Write	Always		
5	ppcR120decimal	10.168.23.79:128	!I4:RD0.120	Read+Write	Always		

Opening the Main Driver Sheet

Most of the fields on this sheet are standard for all drivers; see the “Communication” chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station** and **I/O Address** fields use syntax that is specific to the VMASC driver.

4. For each table row (i.e., each tag/register association), configure the **Station** and **I/O Address** fields as follows:
5. **Station** field: Use this field to define the address of the device to be read from or written to. You must use the following syntax when defining this address:

   <IP Address>:<ID Number>

   Where:

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

   – ***<ID Number>*** is the ID of the device; Valid values are **0–254** (*no default*).

For example, **192.168.2.4:128**

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.

6. **I/O Address:** Specify the address of the associated device register, use the following syntax:

   <Type>:<Name><Row>.<Column> (for example **!I4:CP1.1**)

   Where:

   – **Type:** Item data type. Valid values are: **!I2**, **!I4**, **!R4**, **!R8**, **!BSTR**, **!BOOL** and **!DATE**.

   The following table lists all of the data types that are valid for the VMASC driver:

Data Types	Comments
!I2	Integer with 2 bytes
!I4	Integer with 4 bytes
!R4	Float Point with 4 bytes
!R8	Float Point with 8 bytes
!BSTR	String
!BOOL	Boolean
!DATE	Date

Name: The name identifies the element of the logical device.

Row: Row position of the data element. Usually this is the first information of that address, such as Axis Number, Task number or Drive Number.

Column: Column position of the element. Usually the parameter number for that address.

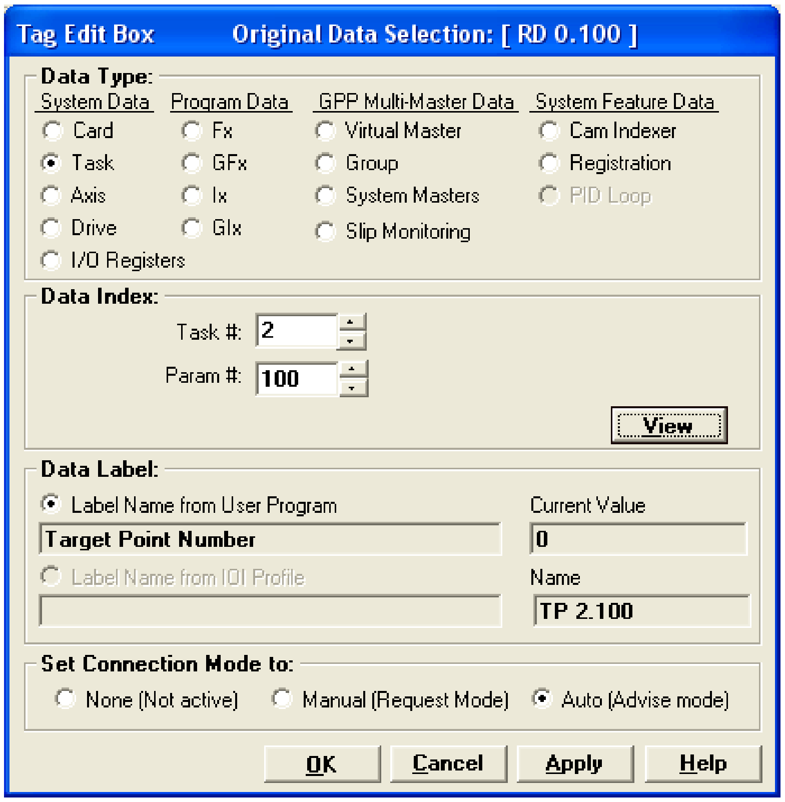
The following table lists all of the Names that are valid for the VMASC driver:

Command	Name	Description
Axis Parameters	AA	Attributes
	AB	Block List Parameter
	AD	Lists/Tables
	AH	Upper Limit
	AL	Lower Limit
	AP	Parameter Data
	AT	Name Text
	AU	Units Text
Control Parameters	CA	Attributes
	CB	Block List Parameter
	CD	Lists/Tables
	CH	Upper Limit
	CL	Lower Limit
	CP	Parameter Data
	CT	Name Text
	CU	Units Text
Drive Parameters	DA	Attributes
	DB	Block List Parameter
	DD	Lists/Tables
	DH	Upper Limit
	DL	Lower Limit
	DP	Parameter Data

	DT	Name Text
	DU	Units Text
Task Parameters	TA	Attributes
	TB	Block List Parameter
	TD	Lists/Tables
	TH	Upper Limit
	TL	Lower Limit
	TP	Parameter Data
	TT	Name Text
	TU	Units Text
Float Variables	FP	Data
	FT	Text Label
Global Integer Variables	GP	Data
	GT	Text Label
Global Float Variables	HP	Data
	HT	Text Label
Integer Variables	IP	Data
	IT	Text Label
I/O Registers	RB	Current State in Binary
	RC	Forcing State Change
	RD	Current State in Decimal
	RE	Erase all Forcing Masks
	RF	Forcing Mask
	RS	Binary Forcing State
	RT	NameText
	RX	Current State in Hexadecimal

➲ **Attention:**
The reading and writing of the *RB (*Current State in Binary*) or RX (*Current State in Hexadecimal) must use the Data type !BSTR and a String Tag.

For example, to read the **Data Display TP 2.100** in Decimal format, you would have in the **Visual Motion I/O Box,** something like this:



Data Type: System Data — [✓ Task]

Task #: 2

Param #: 100

[✓ Label Name from User Program]: Target Point Number

Current Value: 0

Name: TP 2.100

Set Connection Mode to: [✓ Auto (Advise mode)]

On the Studio **I/O Address**, you would configure:

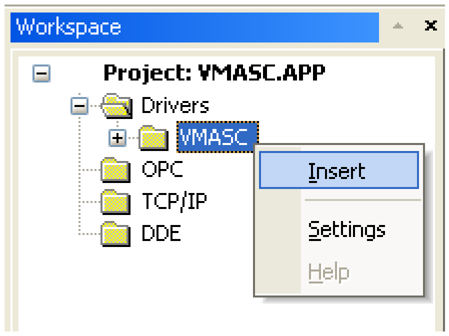
!I4:TP2.100

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

STANDARD DRIVER WORKSHEET

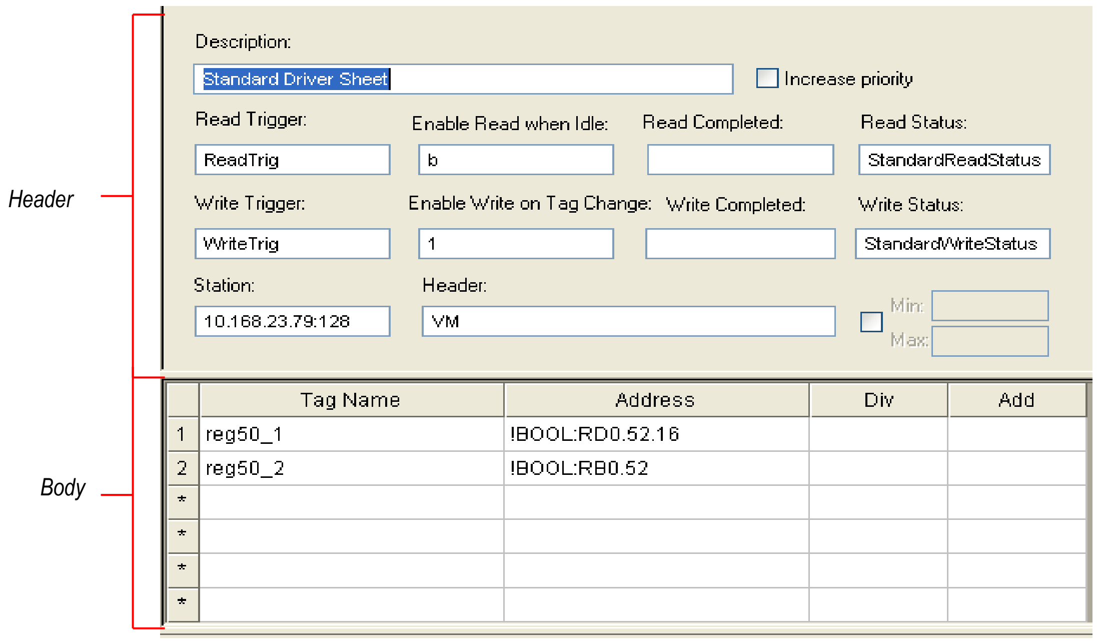
When you select the VMASC driver and add it to your application, it has only a Main Driver Sheet by default (see previous section). However, you may insert additional Standard Driver Worksheets to define tag/register associations that are triggered by specific application behaviors. Doing this will optimize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

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



Inserting a New Worksheet

A new VMASC driver worksheet is inserted into the VMASC subfolder, and the worksheet is opened for configuration:



Description: Standard Driver Sheet
☐ Increase priority

Read Trigger: ReadTrig	Enable Read when Idle: b	Read Completed:	Read Status: StandardReadStatus
Write Trigger: WriteTrig	Enable Write on Tag Change: 1	Write Completed:	Write Status: StandardWriteStatus
Station: 10.168.23.79:128	Header: VM	Min:	Max:

	Tag Name	Address	Div	Add
1	reg50_1	!BOOL:RD0.52.16		
2	reg50_2	!BOOL:RB0.52		
*				
*				
*				
*				

VMASC Driver Worksheet

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 VMASC driver.

3. Configure the **Station** and **Header** fields as follows:
   - **Station** field: **Station** field: Use this field to define the address of the device to be read from or written to. You must use the following syntax when defining this address:

     <IP Address>:<ID Number>

     Where:

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

     – ***<ID Number>*** is the ID of the device; Valid values are **0–254** (*no default*).

For example, **192.168.2.4:128**

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.

   - **Header** field: Use the information in the following table to define the type of variables that will be read from or written to the device.

The **Header** field uses the following syntax:

<Type>

Where:

Type: Item data type. The only valid value is: VM.

The following table lists all of the data types that are valid for the VMASC driver:

Data Types	Comments
!I2	Integer with 2 bytes
!I4	Integer with 4 bytes
!R4	Float Point with 4 bytes
!R8	Float Point with 8 bytes
!BSTR	String
!BOOL	Boolean
!DATE	Date

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

<Type>:<Name><Row>.<Column> (for example **!I4:CP1.1**)

Where:

Type: Item data type. Valid values are: **!I2**, **!I4**, **!R4**, **!R8**, **!BSTR**, **!BOOL** and **!DATE** (see table above).

Name: The name identifies the element of the logical device.

Row: Row position of the data element. Usually this is the first information of that address, such as Axis Number, Task number or Drive Number.

Column: Column position of the element. Usually the parameter number for that address.

The following table lists all of the Names that are valid for the VMASC driver:

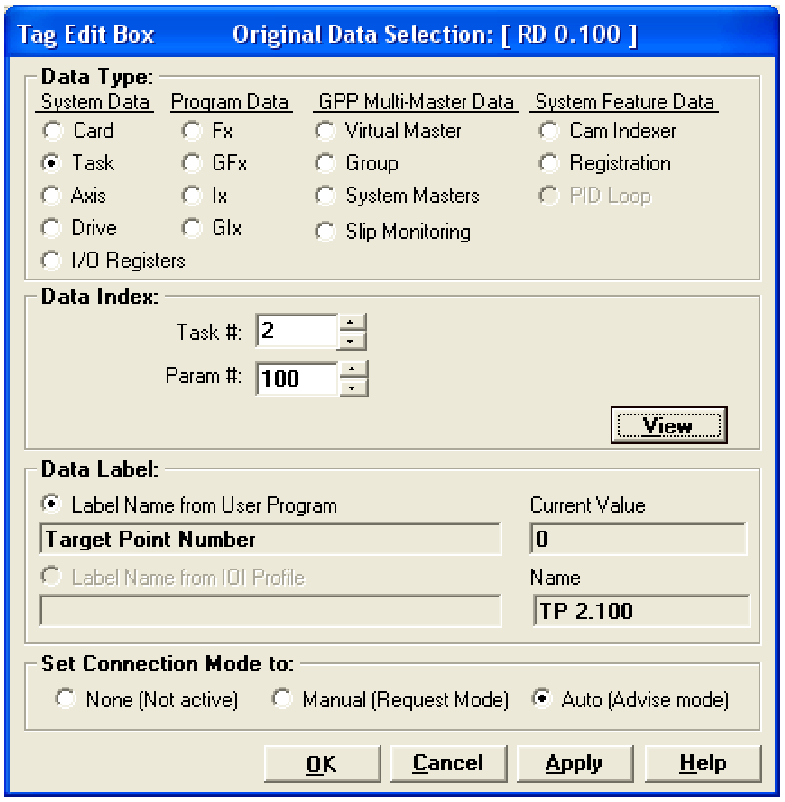
Command	Name	Description
Axis Parameters	AA	Attributes
	AB	Block List Parameter
	AD	Lists/Tables
	AH	Upper Limit
	AL	Lower Limit
	AP	Parameter Data
	AT	Name Text
	AU	Units Text
Control Parameters	CA	Attributes
	CB	Block List Parameter
	CD	Lists/Tables
	CH	Upper Limit
	CL	Lower Limit
	CP	Parameter Data
	CT	Name Text
	CU	Units Text
Drive Parameters	DA	Attributes
	DB	Block List Parameter
	DD	Lists/Tables
	DH	Upper Limit

	DL	Lower Limit
	DP	Parameter Data
	DT	Name Text
	DU	Units Text
Task Parameters	TA	Attributes
	TB	Block List Parameter
	TD	Lists/Tables
	TH	Upper Limit
	TL	Lower Limit
	TP	Parameter Data
	TT	Name Text
	TU	Units Text
Float Variables	FP	Data
	FT	Text Label
Global Integer Variables	GP	Data
	GT	Text Label
Global Float Variables	HP	Data
	HT	Text Label
Integer Variables	IP	Data
	IT	Text Label
I/O Registers	RB	Current State in Binary
	RC	Forcing State Change
	RD	Current State in Decimal
	RE	Erase all Forcing Masks
	RF	Forcing Mask
	RS	Binary Forcing State
	RT	NameText
	RX	Current State in Hexadecimal

➲ **Attention:**

The reading and writing of the *RB* (Current State in Binary*) or RX (*Current State in Hexadecimal) must use the Data type !BSTR and a String Tag.

For example, to read the **Data Display TP 2.100** in Decimal format, you would have in the **Visual Motion I/O Box,** something like this:


Tag Edit Box   Original Data Selection: [RD 0.100]

Data Type:

System Data	Program Data	GPP Multi-Master Data	System Feature Data
Card	Fx	Virtual Master	Cam Indexer
[✓ Task]	GFx	Group	Registration
Axis	Ix	System Masters	PID Loop
Drive	GIx	Slip Monitoring	
I/O Registers			

Data Index:
Task #: 2
Param #: 100

Data Label:
[✓ Label Name from User Program]: Target Point Number
Label Name from IOI Profile:
Current Value: 0
Name: TP 2.100

Set Connection Mode to: None (Not active) Manual (Request Mode) [✓ Auto (Advise mode)]

On the Studio **I/O Address**, you would configure:

Header: **!I4**

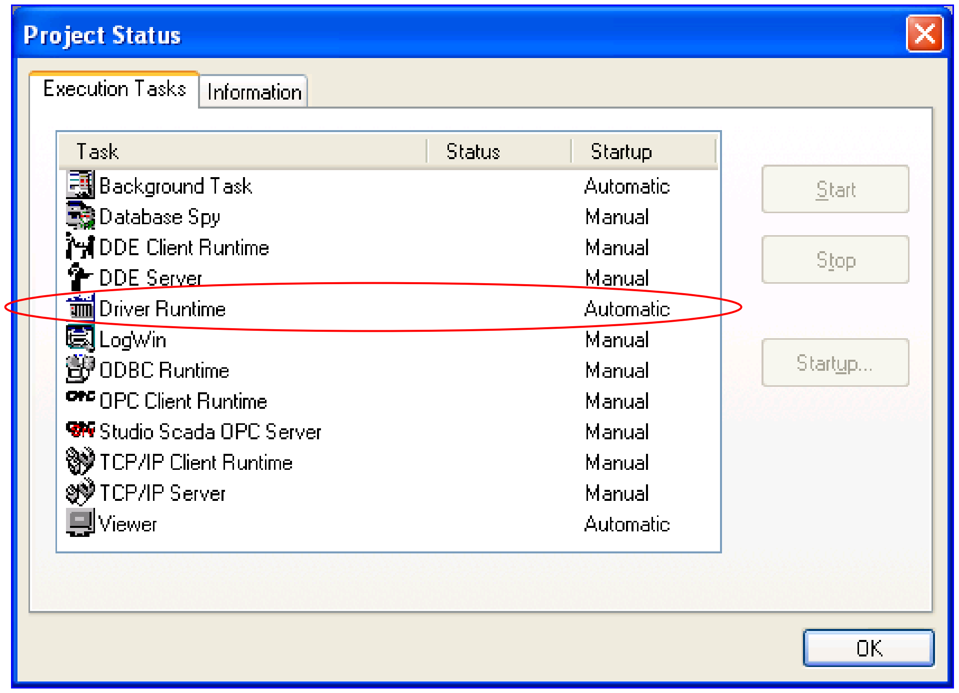
Address: **TP2.100**

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:



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 VMASC 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	ERROR INITIALIZE	Error in the initialization of the driver.	Check if the DrvStd.dll file is into the \BIN folder Check if the VMASC.ini file is into the \DRV folder
2	ERROR READ1	The driver can not start a reading.	Check if the DrvStd.dll file is into the \BIN folder Check if the VMASC.ini file is into the \DRV folder
3	ERROR READ2	The driver receives an invalid message from device.	Check in the worksheet is configured according to this documentation.
4	ERROR WRITE1	The driver can not start writing.	Check if the DrvStd.dll file is into the \BIN folder Check if the VMASC.ini file is into the \DRV folder
5	ERROR WRITE2	The driver receives an invalid message from device.	Check in the worksheet is configured according to this documentation.
6	ERROR INVALID MESSAGE	The driver receives an invalid message from device.	Check in the worksheet is configured according to this documentation.
7	ERROR INVALID BLOCK SIZE	The number of the lines configured in the worksheet is so big.	Check in the Communication Parameters the maximum block size.
-17	Timeout between RX char.	- PLC in stop or error mode - Wrong station number - Wrong parity - Wrong RTS/CTS configuration settings	Check the cable wiring Check the PLC state. It must be RUN Check the station number. Check the right configuration. See on the section 2.2 the different RTS/CTS valid configurations.
-15	Timeout Start Message	- Disconnected cables - PLC turned off, or in Stop or error mode - Wrong Station number - Wrong RTS/CTS control settings.	Check the cable wiring Check the PLC state. It must be RUN Check the station number. Check the right configuration. See on the section 2.2 the different RTS/CTS valid configurations.

⇨ **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 *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 Studio and the target device, then try instead to establish communication using the device's own programming software (e.g., WinProLadder). 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.

To test communication between Studio and the device, 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**.
- **Project Information**: To find this information, select **Project** → **Status.**
- **Driver Version** and **Communication Log**: Displays in the Studio *Output* window 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	Eric Vigiani	Dec/15/2006	Initial version
B	1.01	Plínio M. Santana	Feb/22/2007	The range of the bits was changed to 1-16.
C	1.01	Plínio M. Santana	Mar/05/2007	Document corrections (Header always filled with VM value).