OMEIP Communication Driver

Driver for Ethernet Communication with OMRON Safety
Relays through NE1A-EDR01 Router

Contents

Introduction

This document will help you to select, configure and execute the OMEIP 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 OMEIP driver in the Studio system.

- **Configuring the Driver**: Explains how to configure the OMEIP driver in the Studio system, including how to associate database tags with device registers.

- **Executing the Driver**: Explains how to execute the OMEIP 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 OMEIP driver configuration

- **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 7/XP/Vista 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 communication between the OMEIP driver in Studio and remote devices.

The information is organized into the following sections:

- Device Specifications
- Network Specifications
- Driver Characteristics
- Conformance Testing

Device Specifications

You can use this driver to communicate with any device using the NE1A-EDR01 router UDP protocol. (The devices used for conformance testing are listed on the next page.)

Network Specifications

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

- **Device Communication Port**: Ethernet
- **Physical Protocol**: UDP/IP
- **Logic Protocol**: NE1A-EDR01 UDP Protocol
- **Device Runtime Software**: None
- **Specific PC Board**: None
- **Adapters/Converters**: None
- **Cable Wiring Scheme**: None

Driver Characteristics

The OMEIP driver package consists of the following files, which are automatically installed in the `\DRV` subdirectory of Studio:

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

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

- Windows 7/XP/Vista
- Windows CE 4.x, 5.x, 6.x

For a description of the operating systems used to test driver conformance, see "Conformance Testing" below.

Conformance Testing

The following hardware/software was used for conformance testing:

- **UDP/IP Port**: 64001
- **Protocol**: NE1A-EDR01 UDP Access

- **Station**: Composed by Device Model, IP, Port and DeviceNet ID
- **Cable**: Ethernet Cable

Driver Version	Studio Version	Operating System	Equipment
1.0	7.0	Windows 7	NE1A-EDR01 routing to NE1A-SCPU01-V1 safety relay

The OMEIP driver supports the following device types:

Header Information		Operations		Field Type	
Device Name	Driver Name	Write	Read	Boolean	Byte
Input Monitor	IM	–	●	●	–
Input Status	IS	–	●	●	–
Input Error Cause	IE	–	●	–	●
Output Monitor	OM	–	●	●	–
Output Status	OS	–	●	●	–
Output Error Cause	OE	–	●	–	●
Overall Status	STATUS	–	●	●	–

The user must be aware that the number of registers for each of these Headers varies according to each of the devices supported, exposed on the table below:

| Safety Relay Model	Header		
	Inputs (IM, IS, IE)	Outputs (OM, OS, OE)	STATUS
NE1A-SCPU01-V1	16	8	8
NE1A-SCPU02	40	8	8
DST1-ID12SL-1	12	-	8
DST1-MD16SL-1	8	8	8
DST1-MRD08SL-1	4	4	8
DST1-XD0808SL-1	8	8	8

✎ **Note:**

Refer to the table above when necessary for filling the station field on driver sheets.

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 OMEIP driver for your Studio application:

1. From the main menu bar, select **Insert** → **Driver** to open the *Communication Drivers* dialog.

2. Select the **OMEIP** driver from the *Available Drivers* list, and then click the **Select** button.
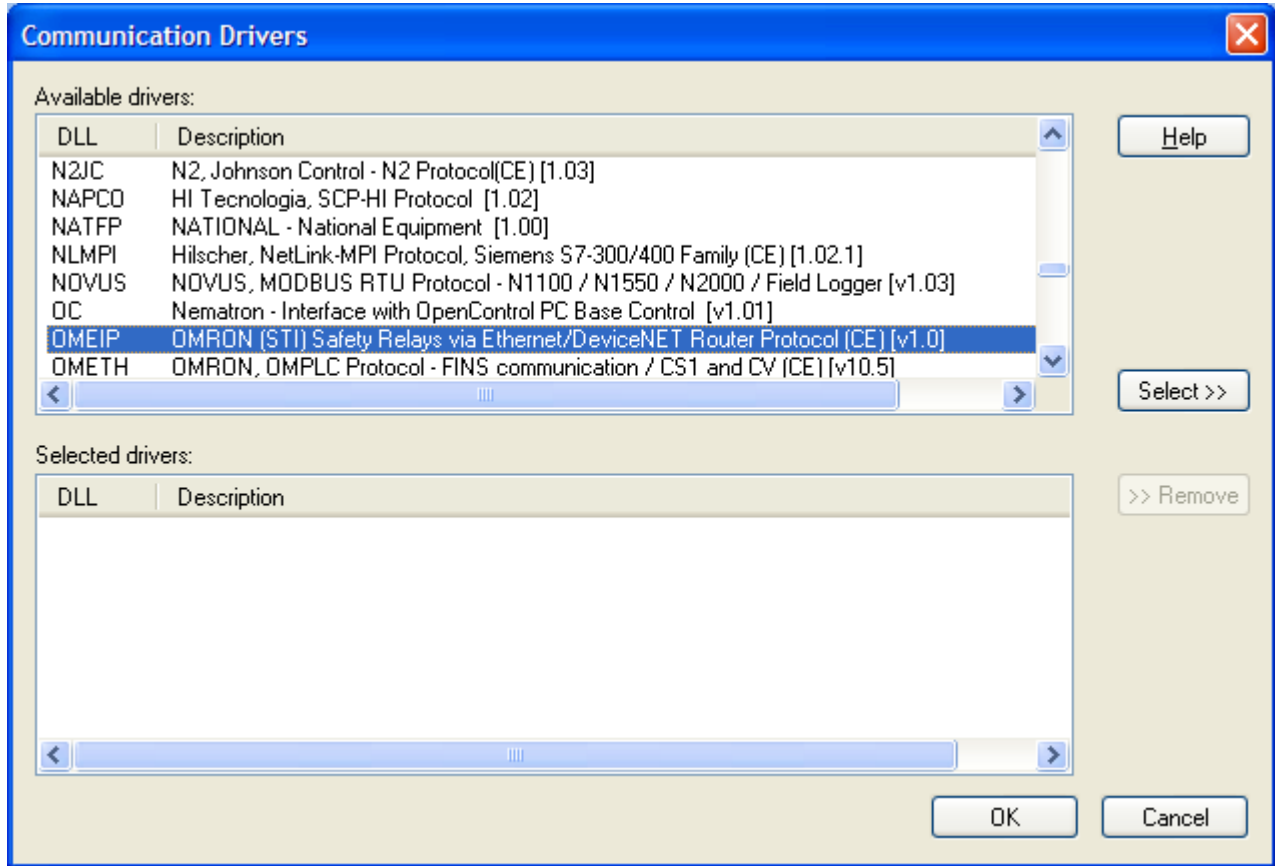
Communication Drivers Dialog

3. When the **OMEIP** 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.

➲ **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

Once you have selected the OMEIP driver in Studio, you must properly configure it to communicate with your target 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 OMEIP driver-specific settings and procedures will be discussed here. To configure the communication settings for the OMEIP driver:

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

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

Select Settings from the Pop-Up Menu

3. The OMEIP communications settings dialog is displayed
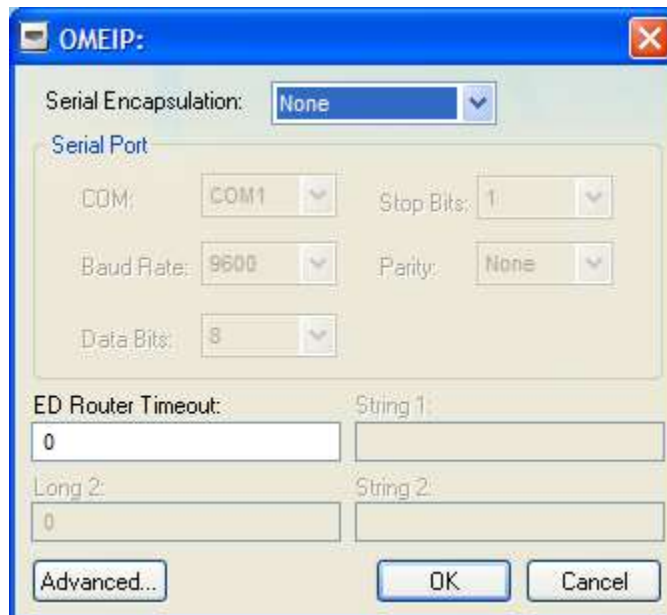
OMEIP Communication Settings Dialog

4. 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
ED Router Timeout	0	Any number in range 0 - 65535	Defines the timeout of the ED router in units of 10ms (hundredths of second). The zero value uses the default timeout of 10ms.

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 OMEIP driver-specific parameters and procedures will be discussed here.

MAIN DRIVER SHEET

When you select the OMEIP driver and add it to your application, Studio automatically inserts the *Main Driver Sheet* in the *OMEIP* 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 *OMEIP* subfolder:
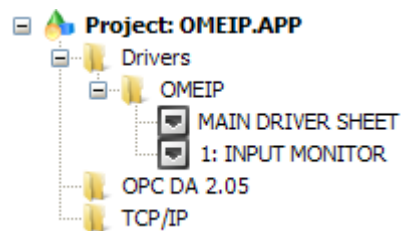
Main Driver Sheet in the OMEIP Subfolder

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

	Tag Name	Station	I/O Address	
1	INPUT_MONITOR[0]	NE1A-SCPU01-V1:5:10.168.23.64	IM:0	Re
2	INPUT_MONITOR[1]	NE1A-SCPU01-V1:5:10.168.23.64	IM:1	Re
3	INPUT_MONITOR[2]	NE1A-SCPU01-V1:5:10.168.23.64	IM:2	Re
4	OUTPUT_MONITOR[0]	NE1A-SCPU01-V1:5:10.168.23.64	OM:0	Re
5	OUTPUT_MONITOR[1]	NE1A-SCPU01-V1:5:10.168.23.64	OM:1	Re
6	OUTPUT_MONITOR[2]	NE1A-SCPU01-V1:5:10.168.23.64	OM:2	Re
7	INPUT_STATUS[0]	NE1A-SCPU01-V1:5:10.168.23.64	IS:0	Re
8	INPUT_STATUS[1]	NE1A-SCPU01-V1:5:10.168.23.64	IS:1	Re
9	INPUT_STATUS[2]	NE1A-SCPU01-V1:5:10.168.23.64	IS:2	Re
10	OUTPUT_STATUS[0]	NE1A-SCPU01-V1:5:10.168.23.64	OS:0	Re
11	OUTPUT_STATUS[1]	NE1A-SCPU01-V1:5:10.168.23.64	OS:1	Re
12	OUTPUT_STATUS[2]	NE1A-SCPU01-V1:5:10.168.23.64	OS:2	Re

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 OMEIP driver.

4. For each table row (i.e. each tag/register association), configure the **Station** and **I/O Address** fields as follows:

   ▪ **Station** field

      • Use the following syntax:

      <Device Model>:<DeviceNet ID>:<IP Address>:<Port Number>

      Or

      <Device Model>:<DeviceNet ID>:<IP Address>

      Where:

- – *<Device Model>* is the Safety Relay model of table of page 4.

- – *<DeviceNet ID>* is the ID of the Safety Relay on the DeviceNet network

- – *<IP Address>* is the IP address of the NE1A-EDR01 router.

- – *<Port Number>* is the UDP/IP port number of the router. If you do not configure this parameter, the default value of **64000** will be used

Examples:

```
NE1A-SCPU01-V1:5:10.168.23.64
```

```
DST1-ID12SL-1:20:10.168.23.64:35000
```

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 cannot leave the Station field **blank**

- ▪ **I/O Address** field — Specify the name or address of the associated device, using the following Syntax:

<Header>:<Address>

Where:

- – *<Header>*: Is one of the headers presented on the table of page 4: **IM**, **IS**, **IE**, **OM**, **OS**, **OE** or **STATUS**

- – *<Address>*: Is the number of the Input/Output or Bit (for the overall status) of the device register.

> ✎ **Note:**
>
> The Error Cause requests of headers **IE** and **OE** are always issued as single messages. Thus these addresses are not merged on virtual groups. Precautions must be taken not to overflow the network with these messages.

STANDARD DRIVER WORKSHEET

When you select the OMEIP 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.

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.

To insert a new driver worksheet:

1. In the *Comm* tab, open the *Drivers* folder and locate the *OMEIP* subfolder.

2. Right-click on the *OMEIP* subfolder, and then select **Insert** from the pop-up menu:

Inserting a New Worksheet

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

OMEIP Driver Worksheet

> ✎ **Note:**
>
> Worksheets are numbered in order of creation, so the first worksheet is `OMEIP001.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 **Header** and **Body** (as noted on the above picture) fields use syntax that is specific to the OMEIP driver.

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

▪   **Station** field

•   Use the following syntax:

<Device Model>:<DeviceNet ID>:<IP Address>:<Port Number>

Or

<Device Model>:<DeviceNet ID>:<IP Address>

Where:

– **`<Device Model>`** is the Safety Relay model of table of page 4.

– **`<DeviceNet ID>`** is the ID of the Safety Relay on the DeviceNet network

– **`<IP Address>`** is the IP address of the NE1A-EDR01 router.

– **`<Port Number>`** is the UDP/IP port number of the router. If you do not configure this parameter, the default value of **64000** will be used

Examples:

`NE1A-SCPU01-V1:5:10.168.23.64`

`DST1-ID12SL-1:20:10.168.23.64:35000`

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 cannot leave the Station field **blank**

▪ **Header** field: Specify the address of the first register of a block of registers on the target device. 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.

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

`<Header>:<Offset>`

Examples — **`IM:8`**

— **`OM:1`**

Where:

– **`<Header>`** is the device type (**`IM, IS, IE, OM, OS, OE, STATUS`**).

– **`<Offset>`** is the optional offset applied to all addresses on the driver sheet.

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 **`IM`**.

You can also specify a string tag (e.g. **`{header}`**), but the tag value 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 OMEIP driver:

Type	Syntax	Comments
Input Memory	**IM**	Physical Input Monitor. Reads the values of the inputs. 1 = On, 0 = Off
Input Status	**IS**	Physical Input Status. Reads the status of the inputs. 1 = Normal, 0 = Error
Input Error Cause	**IE**	Error code to indicate the cause of the error related to the input.
Output Monitor	**OM**	Physical Output Monitor. Reads the values of the outputs. 1 = On, 0 = Off
Output Status	**OS**	Physical Output Status. Reads the status of the outputs. 1 = Normal, 0 = Error
Output Error Cause	**OE**	Error code to indicate the cause of the error related to the output.
Overall Status	**STATUS**	Reads logical bit flags indicating the overall status of the device.

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

   For registers of type **AMINFO** and **DMINFO**, use the syntax:

   > ***\<Address\>***

   Where:

   – ***\<Address\>***: Is the number of the Input/Output or Bit (for the overall status) of the device register.

   Examples:

   – For a header **IM**:

   - o   0 – The value of the input number 1
   - o   37 – The value of the input number 37

   – For a header **OS:10**:

   - o   1 – The status of the output number 11 (notice the offset on the header)

⮑   **Attention:**

The user is allowed to type any number on the address field on the range 0 to 39. However, for some devices, as shown on page 4, not all these addresses are valid. This validation, based on the device model set on the Station field, is done before starting a read operation and may result on "Invalid Address" errors with a message on the log to allow the user to know which address is invalid.

Also, not all headers are valid for all devices. This is also validated before starting a read and implies on the error "Invalid Header" with a message on the log to allow the user to fix it.

Specifically to headers **IE** and **OE**, a single message must be sent for each line of the sheet. It is recommended for the user to use the Main Driver Sheet to read those fields or use a tag on the header to change offsets as necessary.

OVERALL STATUS

The Overall Status (read by the **STATUS** header) has different meanings depending on the device that it is read from. This section exposes this meanings for the devices supported.

NE1A Family

- Devices supported: NE1A-SCPU01-V1 and NE1A-SCPU02

Bit	Comments
0	Input Power Supply Voltage Status Flag OFF: Always ON, ON: Voltage error or power supply OFF
1	Output Power Supply Voltage Status Flag OFF: Always ON, ON: Voltage error or power supply OFF
2	Standard I/O Communications Error Flag OFF: No error, ON: Error
3	Standard I/O Communications Status Flag OFF: I/O communications stopped or error, ON: I/O communications being executed
4	Safety I/O Communications Error Flag OFF: No error, ON: Error
5	Safety I/O Communications Status Flag OFF: I/O communications stopped or error, ON: I/O communications being executed
6	Operation Mode Flag OFF: Other than RUN, ON: RUN
7	NE1A-series Status Flag OFF: Error, ON: Normal

DST1 Family

- Devices supported: DST1-ID12SL-1, DST1-MD16SL-1, DST1-MRD08SL-1 and DST1-XD0808SL-1

Bit	Comments
0	Input Power Supply Voltage Monitor Flag OFF: I/O power supply ON, ON: I/O power supply OFF
1	Output Power Supply Voltage Monitor Flag OFF: I/O power supply ON, ON: I/O power supply OFF
2	Network Power Supply Voltage Monitor Error Flag OFF: Normal (higher than monitor set value), ON: Error (equal to or higher than monitor)
3	Module Maintenance Flag OFF: Within range (lower than monitor set value), ON: Out of range (equal to or lower than monitor set value)

Bit	Comments
4	This bit varies according to model inside the family: | Model	Comments
DST1-XD0808SL-1	Operation Mode Flag OFF: Other than RUN, ON: RUN
Others	Reserved	
5	I/O Error Flag OFF: Normal (all I/O points normal), ON: Error (one or more I/O points abnormal)
6	This bit varies according to model inside the family: | Model	Comments
DST1-XD0808SL-1	Error Flag OFF: Error, ON: Normal
Others	Operation Time Monitor Error Flag OFF: Within range (all I/O sets lower than monitor set value), ON: Out of range (one or more I/O sets equal to or greater than monitor set value)	
7	Connected Device Maintenance Flag OFF: Within range (all I/O points lower than monitor set value), ON: Out of range (one or more I/O points equal to or greater than monitor set value)

ERROR CAUSES

The Error Causes (read by the **IE** and **OE** header) are read as codes with different meanings for Input and Output. The descriptions of those errors are listed below. For more information, consult your device's manual.

- Input Error Codes:

Code	Description
0	No error
1	Illegal configuration
2	Test signal error
3	Internal circuit error
4	Discrepancy error
5	Error at dual channel partner

- Output Error Codes:

Code	Description
0	No error
1	Illegal configuration
2	Overcurrent detected
3	Short-circuit detected
4	Output ON error
5	Error at dual channel partner
6	Internal bit circuit error
7	Bit error
8	Illegal data between dual channel outputs
9	Short-circuit detected between wires
10	EDM error

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 communication driver(s) will execute correctly:

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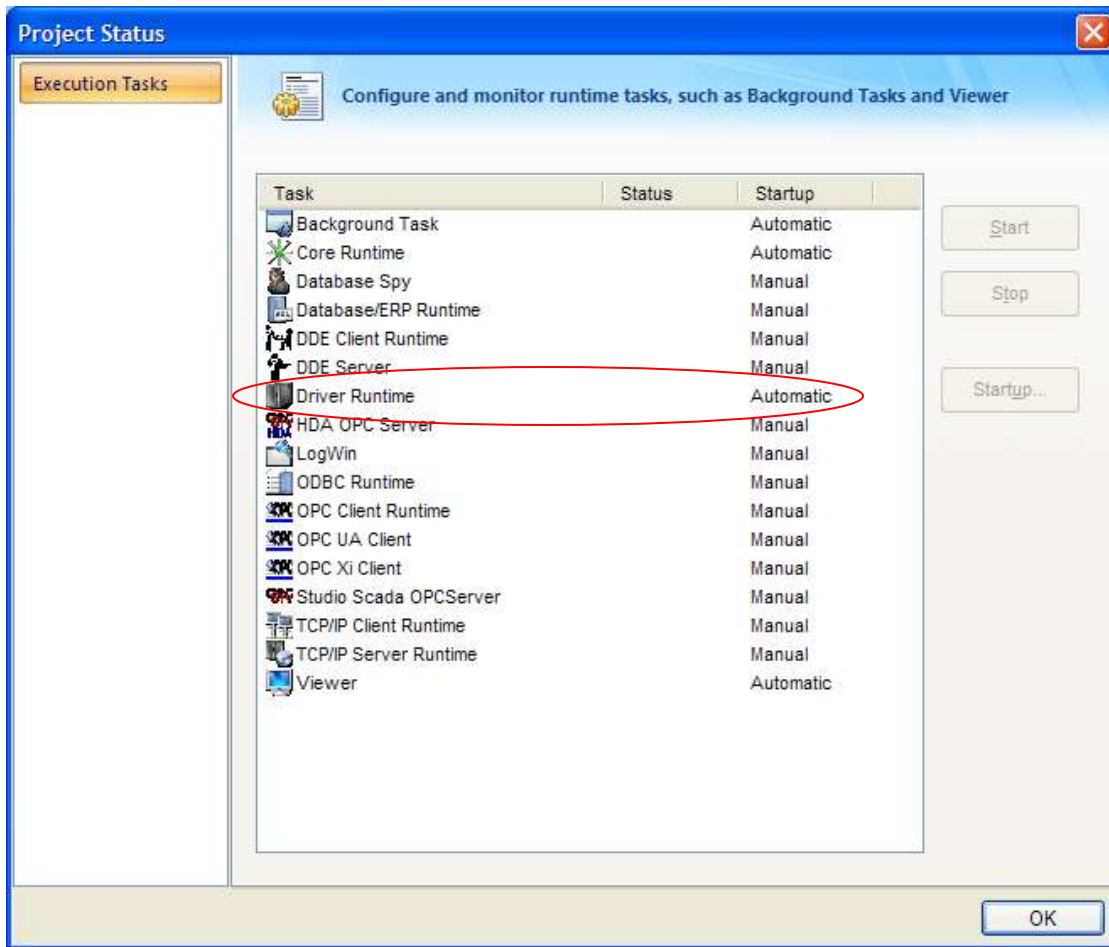
Project Status Dialog

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 OMEIP 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 Standard Driver Sheet will receive an error code. Use this error code and the following table to identify what kind of failure occurred. Error codes above 100 are associated with device errors.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	N/A	N/A
1	Invalid Command	The application attempted to write to the device.	- Check write triggers to not start write operations for headers that do not support this command.
2	Internal Driver Error	This is an internal error code.	- Please contact support to report and solve the issue
3	Invalid Message Length	The received message is invalid.	- Check cable connections - Check device configurations
511	Message Rejected	The device rejected the message. This error happens when the driver is running on a computer with IP in a different range than the device.	- Check device configurations - Check network configurations
-15	Timeout waiting start a message	PLC is not responsive.	- Check cable connections - Check Station field for wrong IP or port addresses
-34	Invalid Address	This error happens the address does not comply with the syntax for the header. It may also happen if the address is not available for the specific header and device	- Check if the address is correct according to the expected syntax. - Check if the address exists for the device model set on the Station and for the selected header.
-37	Invalid header	This error happens when you configure a {Tag} between curly brackets on the Header field and the value of this {Tag} does not comply with the Header syntax. This error may also happen if the header is not available for the device set on the Station.	- Change the Tag value in order to comply with the Header Syntax. - Check if the header is available for the device model (see table on page 4)
-38	Invalid station	Invalid configuration in the Station field, such as leaving it blank or configuring an invalid IP Address. For details of the error, see the messages on the log.	- Check the Station field on your driver sheet. Make sure that it is not blank and the syntax complies with the expected for the driver.
-60	Connection Error	Error to establish a TCP/IP connection with the Slave device. Possibly wrong IP Address or Port Number in the Station field	- Check the IP Address, port and ID number in the Station field - Try to *ping* the IP address that you configured in the Station field

⇨ **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 **Protocol Analyzer,** 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** → **LogWin**) to establish an event log on a remote unit that runs Windows CE

If you must contact us for technical support, please have the following information available:

- **Operating System and Project Information** (type and version): To find this information, select **Help → Support Information**.

- **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.

Sample Application

There is no official sample application available for this driver.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of Changes
A	1.0	André Körbes	Aug. 17, 2011	Initial version