OMETH – *Driver Version 10.18*
Doc. Revision AG– July 23rd, 2018

OMETH Communication Driver

Driver for Serial and Ethernet Communication
with Omron Devices Using FINS Commands

Contents

Introduction

The OMETH driver enables communication between the Studio system and Omron devices using FINS commands with CS1, CV and C-series (routing over Ethernet) modes, according to the specifications discussed in this document.

This document will help you to select, configure and execute the OMETH 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 OMETH driver in the Studio system.
- **Configuring the Device**: Describes how the target device must be configured to receive communication from the OMETH driver.
- **Configuring the Driver**: Explains how to configure the OMETH driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver**: Explains how to execute the OMETH 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 OMETH 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 7/8 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.

Attention:

The OMETH driver is compatible with UNICODE versions of Studio, and with ANSI versions of Studio released after July 12, 2001 *only*.

General Information

This chapter identifies all of the hardware and software components required to implement serial communication between the OMETH driver in Studio and Omron 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**: Omron
- **Compatible Equipment**:
  - CS1, CJ1, CJ2, NJ1 and CP1 series using CS1 mode
  - CV, CVM1, and CVM1D series using CV mode
  - PLC model C200 using C-series routing over Ethernet mode
  - Omron PLC programmer software: CX-Programmer, Sysmac Gateway

This driver has been tested successfully with Omron devices using FINS commands with CS1, CV and C200 modes. (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**: Ethernet or RS-232 serial
- **Logic Protocol**: FINS – UDP (Ethernet), FINS – Hostlink (Serial)
- **Device Runtime Software**: None
- **Specific PC Board**: None
- **Cable**: Ethernet cable or RS-232 serial cable

Driver Characteristics

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

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

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

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

- Windows 7/8
- Windows CE

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

The OMETH driver supports the following registers:

Register Type	Length	Write	Read	Bit	Integer	Float	String
CIO (CIO)	2 Bytes	•	•	•	•	•	•
WR (Work)	2 Bytes	•	•	•	•	•	•
HR (Holding Bit)	2 Bytes	•	•	•	•	•	•
AR (Auxiliary Bit)	2 Bytes	•	•	•	•	•	—
DM (DM)	2 Bytes	•	•	•	•	•	•
TA (Timers)	2 Bytes	•	•	—	•	—	•
CA (Counters)	2 Bytes	•	•	—	•	—	•
EM# (Extended Memory, where # is a block number)	2 Bytes	•	•	•	•	•	•
G (CPU Bus Link Registers)	2 Bytes	•	•	•	•	•	—

Conformance Testing

The following hardware/software was used for conformance testing:

Driver Configuration:

- UDP: Ethernet Cable
- Serial:
  - **Cable**: 9-Pin Serial Cable
  - **Baud Rate**: 9600
  - **Data Bits**: 7
  - **Stop Bits**: 2
  - **Parity**: Even

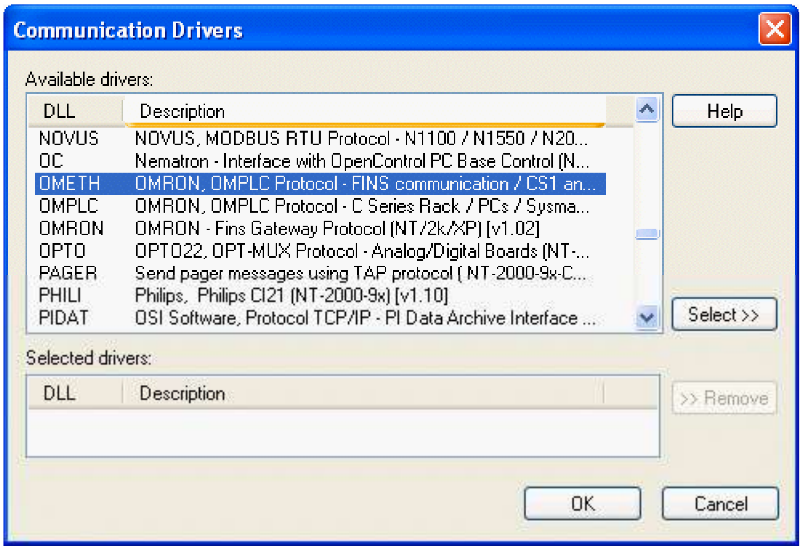
Driver Version	Studio Version	Operating System (development)	Operating System (runtime)	Equipment
10.18	8.1 + SP1	Windows 7/8/10	Windows 7/8/10	CS1 mode: • CJ1 – CPU12 • CJ2 – CPU64 – EIP • CS1G – CPU45H • NJ501 -1300 • CP1L – CPU M CV mode: • CVM1-V2 – CPU11 C-series mode: • C200HG – CPU43

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 OMETH driver for your Studio application:

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

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



Available drivers:

DLL	Description
NOVUS	NOVUS, MODBUS RTU Protocol - N1100 / N1550 / N20...
OC	Nematron - Interface with OpenControl PC Base Control (N...
OMETH	OMRON, OMPLC Protocol - FINS communication / CS1 an...
OMPLC	OMRON, OMPLC Protocol - C Series Rack / PCs / Sysma...
OMRON	OMRON - Fins Gateway Protocol (NT/2k/XP) [v1.02]
OPTO	OPTO22, OPT-MUX Protocol - Analog/Digital Boards (NT-...
PAGER	Send pager messages using TAP protocol (NT-2000-9x-C...
PHILI	Philips, Philips CI21 (NT-2000-9x) [v1.10]
PIDAT	OSI Software, Protocol TCP/IP - PI Data Archive Interface ...

Selected drivers:

DLL	Description

Communication Drivers Dialog

3. When the **OMETH** 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. However, if you are using this driver for UDP communication, you must enable the UDP protocol on your operating system.

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 Omron device, you must also install the Omron programming software (e.g., SYSMAC, CX Programmer etc.). 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 Omron documentation for information about configuring your device.

Configuring the Driver

Once you have selected the OMETH 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.

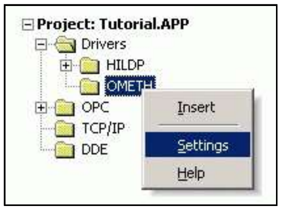
Configuring the Communication Settings

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 OMETH driver-specific settings and procedures will be discussed here. To configure the communication settings for the OMETH driver:

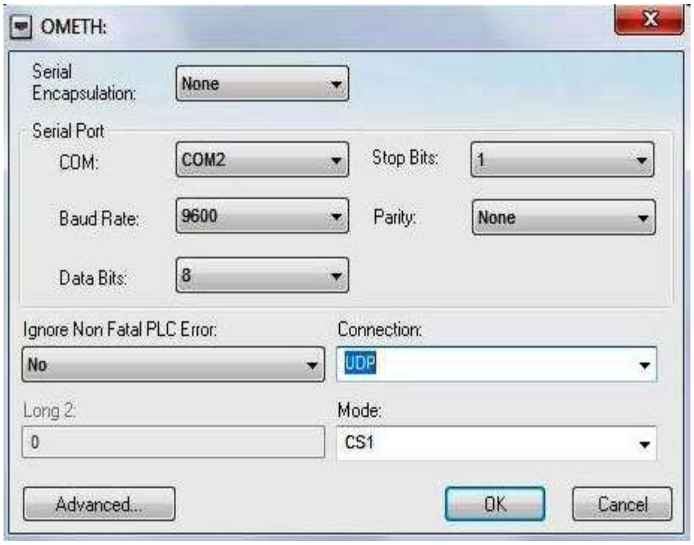
1. In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The OMETH driver is listed here as a subfolder.

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



Select Settings from the Pop-Up Menu

The OMETH: Communication Settings dialog is displayed*:*


OMETH:

Serial Encapsulation: None

Serial Port

COM: COM2

Stop Bits: 1

Baud Rate: 9600

Parity: None

Data Bits: 8

Ignore Non Fatal PLC Error: No

Connection: UDP

Long 2: 0

Mode: CS1

OMETH: 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.

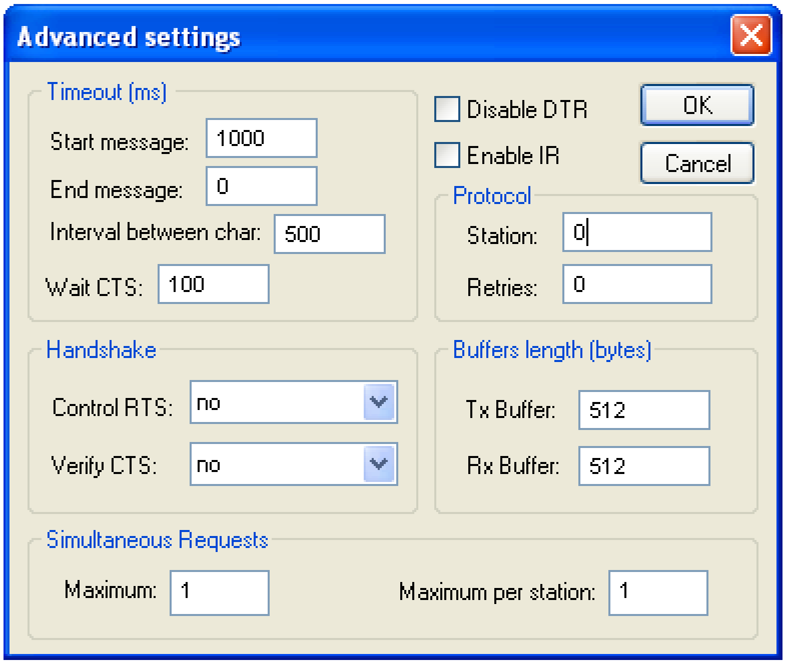
   Depending on your circumstances, you may need to configure the driver *before* you have configured your target device. If this is the case, then take note of the driver settings and have them ready when you later configure the device.

> ➲ **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
Connection	`UDP`	`UDP` or `Serial`	Type of communication. If UDP is specified, then Studio will use the PC's regular Network settings. If `Serial` is specified, then you must configure the serial communication settings above.
Mode	`CS1`	`CS1` or `CV or C`	FINS communication mode. **CS1 Mode**: CS1, CJ1, CJ2, NJ1, CP1 and compatibles. **CV Mode**: CV, CVM1-VM2 and CVM1D and compatibles. **C-series (Routing) Mode**: C200 PLC using routing over Ethernet
Ignore Non Fatal PLC Error	`No`	`No or Yes`	This option is to know if the driver ignores a non fatal PLC error. **No**: When a non fatal error happens in the PLC the driver shows Status 64 and all communication tags go to BAD quality. **Yes**: When a non fatal error happens in the PLC the driver shows Status 0 (No errors) and shows in the logwin/output window a warning message if the option Protocol Analyzer is enabled. All the communication tags stay with GOOD quality.

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



Advanced settings

Timeout (ms)

Start message: 1000

End message: 0

Interval between char: 500

Wait CTS: 100

☐ Disable DTR

☐ Enable IR

Protocol

Station: 0

Retries: 0

Handshake

Control RTS: no

Verify CTS: no

Buffers length (bytes)

Tx Buffer: 512

Rx Buffer: 512

Simultaneous Requests

Maximum: 1

Maximum per station: 1

Parameters	Syntax	Default Values	Valid Values	Description
Station	**<Network>:*[Node]***	**0**	**<0 ... 127>:<1 ... 254>**	**Network**: This value is the Fins Network. (For example, Fins Network 0 as in the **Default Value**). **Node:** (*Optional parameter*) This is the computer node address in the Fins Network, which must be a unique number in the network. If this parameter is not specified, then it uses the last number of your computer's IP address. (For example, if your computer IP address is 192.168.2.15, the node will be 15). If this filed is blank, the driver will assume Fins Network is 0 and node is the last number of the computer's IP address.

Note:

- You cannot have more than one station in the FINS network with the same NODE number. Therefore, if you create more than one instance of the OMETH driver, make sure to configure different local NODE numbers (Driver Settings -> Advanced -> Station) for each instance of the driver. Otherwise, the PLCs might be unable to answer some requests of the drivers, generating sporadic time-out errors.
- For the same reason, Simultaneous Connections are not supported by the OMETH driver when using the old Ethernet communication modules (e.g.: CS1W-ETN01, CS1W-ETN11, CJ1W-ETN11) or using new modules (e.g.: CS1W-ETN21, CJ1W-ETN21) in "ETN11-compatible mode". A work-around for this limitation is to configure a different FINS/UDP Port for each device (e.g.: 9600, 9601, etc) and to use the complete IP address with the port number when configuring the stations for the driver (e.g.: 192.168.1.10:9600, 192.168.1.20:9601, etc).
- Simultaneous connections are supported by the OMETH driver, when using the new Ethernet communication modules (e.g.: CS1W-ETN21, CJ1W-ETN21) which are able to send responses to

the source, even if the source’s FINS/UDP port number is different from the UDP Port number set as the FINS/UDP port number on the device. It is also important to make sure that the “ETN11-compatible mode” is NOT enabled on the new communication modules.

5. Click **OK** to close the *Advanced Settings* dialog, and then click **OK** to close the *Communication Settings* dialog.

✍ **Note:**

. See Appendix for configuring Station field when the destination device is using Controller Link

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.

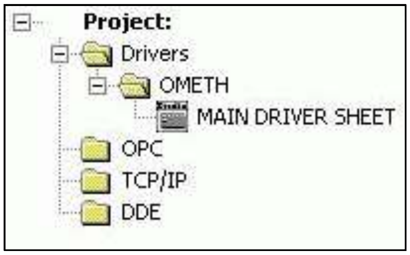
The configuration of these worksheets is described in detail in the “Communication” chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

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

Main Driver Sheet

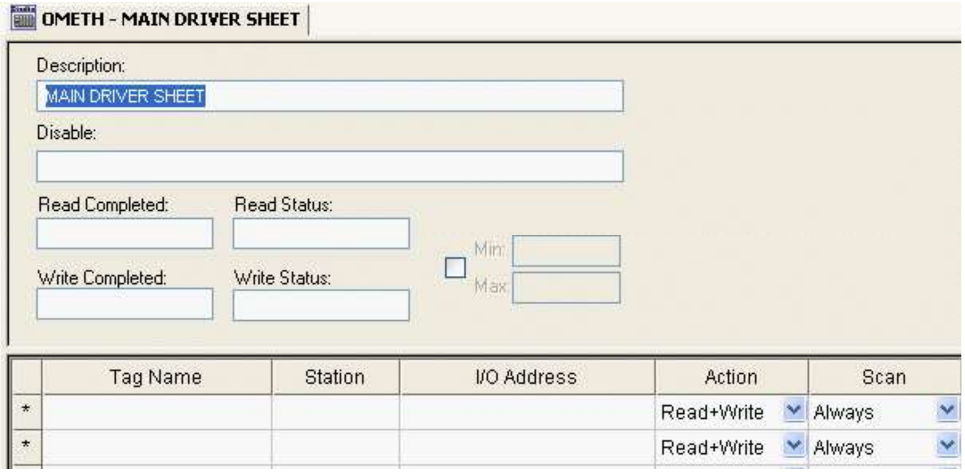
When you select the OMETH driver and add it to your application, Studio automatically inserts the *Main Driver Sheet* in the *OMETH* driver subfolder. To configure the Main Driver Sheet:

1. Select the *Comm* tab in the *Workspace* pane.
2. Open the *Drivers* folder, and then open the *OMETH* subfolder:



Main Driver Sheet in the OMETH Subfolder

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



Description: MAIN DRIVER SHEET
Disable:
Read Completed:
Read Status:
Write Completed:
Write Status:
□

	Tag Name	Station	I/O Address	Action	Scan
*				Read+Write	Always
*				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 OMETH driver.

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

   Use the **Station** 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:

   - For UDP Communication:

     <IP Address>:<Port Number>:<Network>:<Node Address>:<Unit ID>:<Mode>

     For example, **192.168.2.4:9600:1:2:1:CS1**

     Where:

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

     – ***<Port Number>*** is the communication port;

     – ***<Network Address>*** is the device's subnet address, according to the FINS protocol

     – ***<Node Number>*** is the device's node number, according to the FINS protocol. If you do not specify a value, then Studio will automatically insert the last-used node number.

     – ***<Unit ID>*** is the ID number used for PC Interface, as defined in the PLC program. If you do not specify a value, then Studio will automatically insert the default 0. **IMPORTANT:** This is *not* the Unit ID that is configured in the hardware settings.

     – ***<Mode>*** is the FINS communication mode. **CS1** mode is used *for CS1, CJ1, CJ2* and compatibles. **CV** mode is used for *CV, CVM1, CVM1D* and compatibles. **C** mode is used for *C200 PLC* using routing over Ethernet. This address component is used to override the global settings of the driver. The default value is **CS1**.

   - For Serial Communication:

     <Network Address>*:*<Node Number>*:*<Unit ID>

     For example, **0:1:0**

     Where:

     – ***<Network Address>*** is the device's network address, according to the FINS protocol;

     – ***<Node Number>*** is the device's node number, according to the FINS protocol; and

     – ***<Unit ID>*** is the ID number used for PC Interface, as defined in the PLC program. If you do not specify a value, then Studio will automatically insert the default 0. **IMPORTANT:** This is *not* the Unit ID that is configured in the hardware settings.

✍ **Note: Device Node**

- The CV PLCs only support nodes upto 126
- Some modules for CS1: ETN01 and ETN11 also only support nodes upto 126.
- Newer modules like the CS1 : ETN21, CJ1, CJ2, NS1 etc can support nodes upto 254.

✍ **Note:**

. See Appendix for configuring Station field when the destination device is using Controller Link

Use the **I/O Address** field to associate each tag to its respective device address. Type an address into the **Address** field that complies with the following syntax:

<Area>*:*<Format><Address>*:*[Len] (For example, `DM:ST2:10`)

<Area>*:*<Format><Address>*.*[Bit] (For example, `DM:BCD0.2`)

Where:

– ***<Area>*** is the Memory area (e.g., **CIO**, **WR**, **HR**, **AR**, and so forth).

– ***<Format>*** defines the format of the register. This is an *optional* parameter; if you leave this field blank, Studio provides the data in Unsigned Word format. The **[Format]** options are as follows:

- **BCD**: Binary Code Decimal format
- **SW**: Signed Word format
- **DW**: Double Word format
- **SDW**: Signed Double Word format
- **DWBCD**: Double Word BCD format
- **ST**: String format (String does not supports bits)
- **STS:** Byte Swapped String format (String does not support bits)
- **F**: Float format.

- ***<Address>*** is the specific register address.

- ***[Bit]*** is the bit number in the device address. This is an *optional* parameter.

- ***[Len]*** is the length to read or to write. It is in bytes.

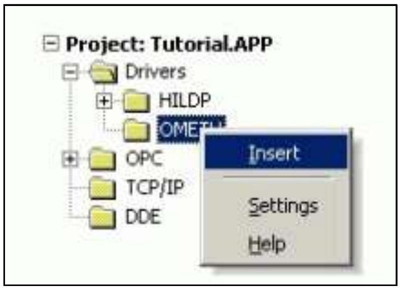
Standard Driver Worksheet

When you select the OMETH 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.

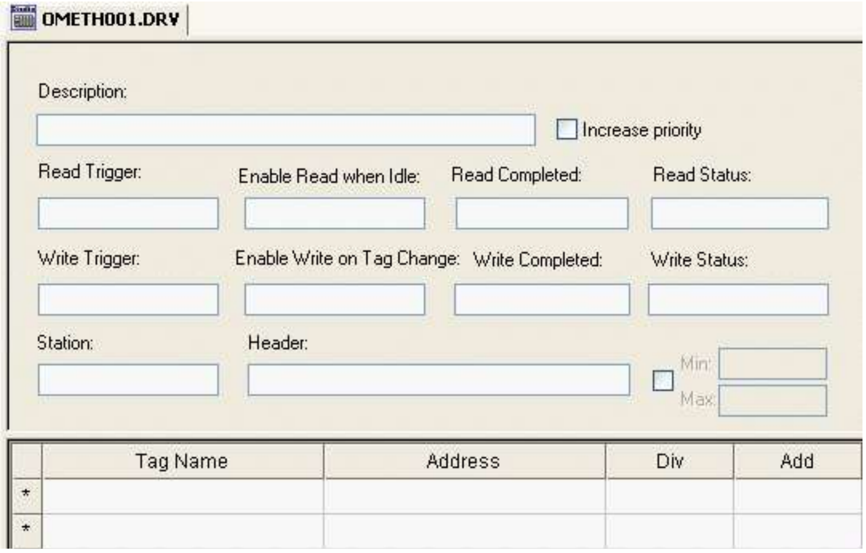
To insert a new Standard Driver Worksheet:

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



Inserting a New Worksheet

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



OMETH Driver Worksheet

> **Note:**
>
> Worksheets are numbered in order of creation, so the first worksheet is **`OMETH001.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 OMETH driver.

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

   - **Station** field: Specify the IP Address of the device and the slot number, using the following syntax:

     - For UDP Communication:

       `<IP Address>:<Port Number>:<Network>:<Node Address>:<Unit ID>:<Mode>`

       For example, **`192.168.2.4:9600:1:2:1:CS1`**

       Where:

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

       – ***`<Port Number>`*** is the communication port;

       – ***`<Network Address>`*** is the device's subnet address, according to the FINS protocol; and

       – ***`<Node Number>`*** is the device's node number, according to the FINS protocol. If you do not specify a value, then Studio will automatically insert the last-used node number.

       – ***<Unit ID>*** is the ID number used for PC Interface, as defined in the PLC program. If you do not specify a value, then Studio will automatically insert the default `0`. **IMPORTANT:** This is *not* the Unit ID that is configured in the hardware settings.

       – ***`<Mode>`*** is the FINS communication mode. **CS1** mode is used *for CS1, CJ1, CJ2* and compatibles. **CV** mode is used for *CV, CVM1, CVM1D* and compatibles. **C** mode is used for *C200 PLC* using routing over Ethernet. This address component is used to override the global settings of the driver. The default value is **CS1**.

     - For Serial Communication:

       `<Network Address>`**`:`*****`<Node Number>`*****`:`*****`<Unit ID>`***

       For example, **`0:1:0`**

       Where:

       – ***`<Network Address>`*** is the device's subnet address, according to the FINS protocol;

       – ***`<Node Number>`*** is the device's node number, according to the FINS protocol; and

       – ***`<Unit ID>`*** is the ID number used for PC Interface, as defined in the PLC program. If you do not specify a value, then Studio will automatically insert the default `0`. **IMPORTANT:** This is *not* the Unit ID that is configured in the hardware settings.

> **Note:**
>
> . See Appendix for configuring Station field when the destination device is using Controller Link

- **Header** field: Specify the address of the first register of a block of registers on the target device. The addresses declared in the *Body* of the worksheet are simply offsets of this **Header** address. 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:

  <Area>:<AddressReference>

  For example, **U:3**

  Where:

  – ***<Area>*** is the Memory Area (**CIO**, **WR**, **HR**, **AR** and so forth).

  – ***<AddressReference>*** is the initial (or reference) address of the configured group.

  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 **CIO:0**.

  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 OMETH driver:

Data Types	Sample Syntax	Valid Range of Initial Addresses per Worksheet	Comments
CIO Area	**CIO:10**	0 to 6143	CIO area
WR Area	**WR:0**	0 to 511	Work area
HR Area	**HR:100**	0 to 511	Holding bit area
AR Area	**AR:135**	0 to 959	Auxiliary area
DM Area	**DM:30000**	0 to 32767	DM area
Timer Area	**TA:50**	0 to 4095	Timer area
Counter Area	**CA:34**	0 to 4095	Counter area
EM#	**EM4:287**	0 to 32767	Extended memory area, where # is the block no.
G Area	**G:1**	0 to 255	CPU Bus Link Area

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

   [Format]<AddressOffset>.[Bit] (For example, **DW2.0**)

   [Format]<AddressOffset>:[Len] (For example, **ST2:10**)

   Where:

   – **Format** defines the format of the register. This is an *optional* parameter; if you leave this field blank, Studio provides the data in Unsigned Word format. The **[Format]** options are as follows:

   - **BCD**: Binary Code Decimal format
   - **SW**: Signed Word format
   - **DW**: Double Word format
   - **SDW**: Signed Double Word format
   - **DWBCD**: Double Word BCD format
   - **ST**: String format (String does not supports bits)
   - **STS:** Byte Swapped String format (String does not support bits)

  - **F**: Float format

- **AddressOffset** is a value that is added to the <***Address Reference***> configured in the **Header** field, to define the address of the group configured in the **Header** field.
- **[*Bit*]** is the bit number in the device address. This is an *optional* parameter.
- **[Len]** is the length to read or to write. It is in bytes.

➲ **Attention:**

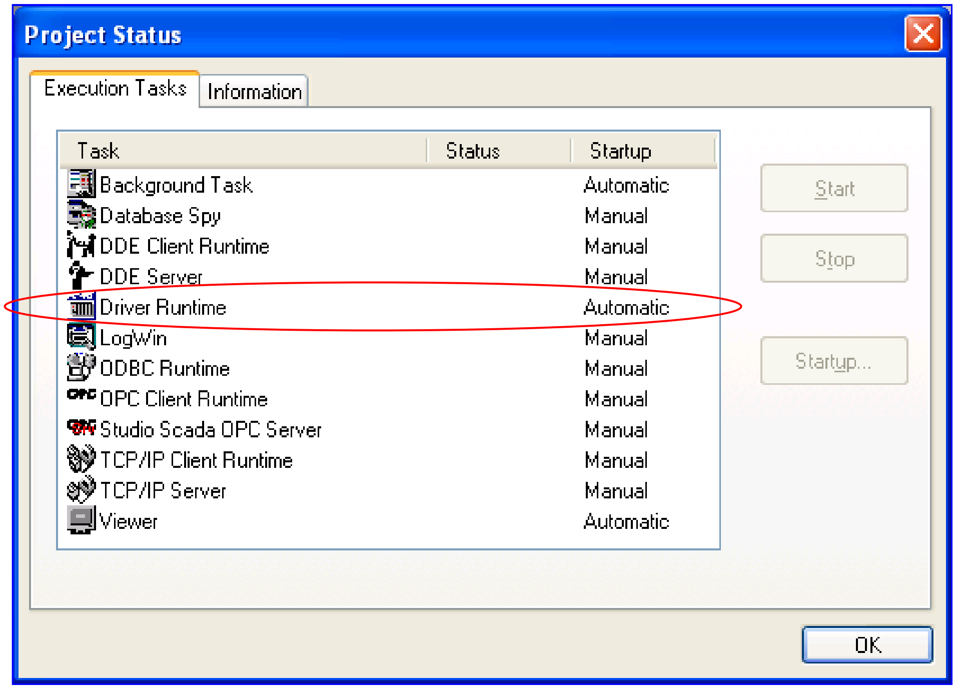
- You can use the Bit-writing function only when the **Write on tag change** parameter is enabled, which means you cannot use the **Write trigger** parameter with the Bit-writing function.
- You must not configure a range of addresses greater than the maximum block size (data buffer length) supported within the same worksheet. For UDP Communication, the maximum block size is 492 bytes (246 words). For Serial Communication, the maximum block size is 234 bytes (117 words). To calculate the worksheet block size (in bytes) use the **(<*HighestAddressOffset*> - <*LowestAddressOffset*> + 1) * 2** expression.
- The Timer and Counter operands share the same memory area, therefore, an address for Timer operand + 0x8000 has the same memory area of an address for Counter operand. For example: TA:20 + 0x8000 = CA:20.

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 OMETH 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	Invalid Unit	Unit number in the address field exceeds 31	Fix the unit number, for serial connection the unit cannot exceed 31
2	Fail to allocate memory	The driver could not allocate memory	▪ Check the buffer configuration in the driver settings and try to reduce the size. ▪ Increase the amount of RAM memory available on your PC or HMI.
3	Invalid Answer	The answer received from the PLC has less bytes than expected.	This error should never happen unless the driver is connected to an unsupported device.
4	RX Buffer Overflow	The number of bytes sent by the PLC exceeds the configured buffer size	Increase the buffer size in the driver settings
5	Invalid Message Size	The answer sent by the PLC does not have the expected size.	This error should never happen unless the driver is connected to an unsupported device.
6	Driver supports UDP or Serial connections	Incorrect configuration of the driver.	Check the configuration of the driver. Use the correct configuration in UDP or Serial connection mode.
33281	Invalid subnet	The subnet configured in the station field is invalid	Check if the subnet configured in the station is correct and that the subnet is properly configured in the PLC routing table.
-37	Invalid Header	Invalid tag value specified in Header field	Specify a valid tag value in Header field.
-34	Invalid Address	Invalid address specified in Address field	Specify a valid address in the Driver Worksheet.
-38	Invalid Station	Invalid station specified in Station field	Specify a valid station number in the Driver Worksheet.
-39	Block Size Error	Specified offset is greater than maximum allowed, and the message cannot be framed	Specify a valid offset or create a new Driver Worksheet.
-15	Timeout Start Message	▪ Disconnected cables ▪ PLC turned off or in Stop/Error mode ▪ Wrong station number ▪ Wrong RTS/CTS configuration settings	▪ Check the cable wiring. ▪ Check the PLC state—it must be RUN. ▪ Check the station number. ▪ Check the RTS/CTS configuration (see the Studio *Technical Reference Manual* for valid configurations).
-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 RTS/CTS configuration (see Studio *Technical Reference Manual* for 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., SYSMAC). 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.

Sample Application

A sample application that employs the OMETH driver is provided on the Studio installation CD. We strongly recommend that you use this sample application to test the driver *before* you develop your own applications, for the following reasons:

- To better understand the information and instructions provided in this document;
- To verify that your driver configuration is working satisfactorily with the target device; and
- To ensure that the all of hardware used in the test (i.e. the device, adapter, cable, and PC) is functioning safely and correctly.

Note:

The following instructions assume that you are familiar with developing project applications in Studio. If you are not, then please review the relevant chapters of the Studio *Technical Reference Manual* before proceeding.

To use the sample application:

1. Configure the device’s communication settings according to the manufacturer’s documentation.
2. Run Studio.
3. From the main menu bar, select **File → Open Project**.
4. Insert the Studio installation CD and browse it to find the sample application. It should be located in the directory **`\COMMUNICATION EXAMPLES\OMETH`**.
5. Select and open the sample application.
6. Configure and test the driver, as described in the rest of this document.

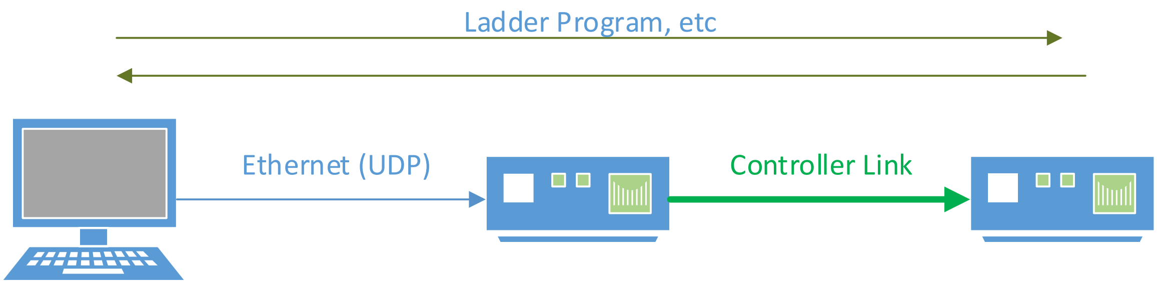
When you have thoroughly tested the driver with your target device, you may proceed with developing your own Studio application projects.

Tip:

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

Appendix : Configuring Station parameters when device uses Controller Link

If you device setup has a Source and Destination controllers that communicate with each other via Controller Link:



Driver Runtime
Parameter : Local IP

PLC 1 (Source)
Parameters:
1. Ethernet IP
2. Port Number
3. Network
4.Node Address
5. Unit ID (optional)
6. Mode (optional)

PLC 2 (Destination)
Parameters:
1. Network
2.Node Address
3. Unit ID (optional)
4. Mode (optional)

Then to successfully communicate with PLC2:

1. Configure the Source station (PLC1) station under Driver Settings-> Advanced as

<Network of PLC1>:<Node Address of PLC1>

2. Configure the Destination station (PLC2) in the Main or Standard Driver sheets under the Station field as

<Ethernet IP of PLC1>:<Port number of PLC1>:<Network of PLC2>:<Node Address of PLC2>:<Unit ID of PLC2 (optional)>: <Mode of PLC2 (optional)>

Revision History

Doc. Revision	Driver Version	Date	Description of Changes
A	1.00	10-Jan-2001	Initial version
B	1.02	11-Feb-2003	▪ Added ability to use serial communication ▪ Added BCD and DWBCD Address field formats
C	1.03	28-Feb-2003	▪ Added Main Driver Worksheet ▪ Added EM, TA and CA Headers
D	1.04	14-Mar-2003	▪ Added Signed/Unsigned Address field formats ▪ Corrected Memory Deallocation error in Serial mode
E	1.05	10-Jun-2003	Corrected problem writing Unsigned Double-Words
F	1.06	27-Sep-2004	Fixed problem when writing values in BCD format
G	1.07	18-Jan-2005	▪ Inserted String type ▪ Fixed problem when writing values in BCD format
H	1.08	26-Sep-2005	Fixed problems while reading counter area
I	1.09	19-Jan-2006	Fixed problem with strings for Windows CE (UNICODE).
J	1.10	14-Jun-2006	Added support for Unit ID in **Station** parameter.
L	1.10	31-Aug-2006	Edited for language and usability.
M	1.11	Mar- 01 -2007	▪ Float imprecision problem addressed ▪ CA operand problems corrected ▪ Explained CA operand's particularity. ▪ Solved problems about reading and writing Strings for different sizes. ▪ Solved problems about the unused spaces when writing Strings.
N	1.12	Jan-08-2008	▪ Implemented the option Ignore Non Fatal PLC Error in the communication parameters ▪ Fixed the writing problem with Float point. ▪ Fixed the writing offset for CA operand ▪ Fixed bug with the treatment of the equipment messages ▪ Modified the CA operand to work with CV Series
O	1.13	Feb-22-2008	▪ Modified the MDS default address ▪ Implement protection to avoid wrong message
O	10.1	Dec-16-2008	▪ Updated version, no changes in the contents.
P	10.3	Apr-21-2009	▪ Fixed issue with invalid headers and Virtual Groups at the Main Driver Sheet
Q	10.4	Jul-30-2009	▪ Improved the driver returned error codes.
R	10.5	Jan-26-2011	▪ Fixed issue with invalid headers and Virtual Groups at the Main Driver Sheet

S	10.6	Jul-23-2014	▪ Fixed string length in address validation
T	10.7	Aug-20-2014	▪ Fixed bit comparison
U	10.8	June-02-2015	▪ Updated version, no changes in the contents.
V	10.9	Nov-11-2015	▪ Included the FINS error code on drive logs.
W	10.10	Feb-22-2016	▪ Included mode in the station field to override the global settings of the driver. ▪ Added support for the PLC model C200.
X	10.11	May-05-2016	▪ Documentation updated with information about support for simultaneous connections. ▪ Modified driver settings to have the connection option using UDP instead of TCP/IP in settings ▪ Fixed issues with strings swapping bytes in NJ1 by creating header STS
Y	10.12	June-13-2016	▪ Fixed the issue of driver creating incorrect virtual groups. ▪ Fixed the issue of invalid addresses for the register type HR in the serial communication.
Z	10.12	Aug-16-2016	▪ Documentation improved for CS1 mode. No changes in the driver.
AA	10.13	Jan-9-2017	▪ Driver documentation updated with information about supported values for the parameter device node number in the station. ▪ Fixed issue of writing to bits of CIO register in CV Series PLCs ▪ Fixed issue of HR (holding registers) not accessible beyond address 511 ▪ Added support for writing to bits of all registers in the CV Series PLCs. ▪ Added support to G registers in CV series PLCs. ▪ Improved creation of virtual groups and for all different station formats. ▪ Added support for Array Access for OMRONFINS OI server. ▪ Improved documentation with the correct limit of the FINS node number
AB	10.14	Apr-14-2017	▪ Improved validation of responses sent. Also improved validation of responses when driver receives non-fatal PLC errors.
AC	10.15	May-9-2017	▪ Fixed issue when using multiple PLCs where communication would completely stop even if one PLC was offline. ▪ Fixed issue of not restoring communication after an offline PLC was reconnected.
AD	10.16	Aug-28-2017	▪ Fixed an issue related invalid station because of memory corruption
AE	10.17	Sept-12-2017	▪ Fixed issue with validation of addresses when using Serial communication

AF	10.17	Nov-08-2017	▪ Improved documentation for how to configure station field to successfully communicate with devices that are connected via controller link.
AG	10.18	Jul-23-2018	▪ Fixed a crash issue.