AVEVA™ Communication Drivers Pack – Omron - OMRONFINS Driver

User Guide

Publication date: Tuesday, May 9, 2023

Publication ID: 868898

Contact Information

AVEVA Group Limited
High Cross
Madingley Road
Cambridge
CB3 0HB. UK

https://sw.aveva.com/

For information on how to contact sales and customer training, see https://sw.aveva.com/contact.

For information on how to contact technical support, see https://sw.aveva.com/support.

To access the AVEVA Knowledge and Support center, visit https://softwaresupport.aveva.com.

Contents

AVEVA

Chapter 1

Getting Started with OMRONFINS Communication Driver

This document describes the OMRONFINS Communication Driver, and the device and protocol environment in which it works. It includes application-level and bus-level communications protocols, item naming conventions, and Communication Driver features.

About the OMRONFINS Communication Driver

The OMRONFINS Communication Driver (referred to as the Communication Driver through the remainder of this user's guide) is a Microsoft® Windows® application program that acts as a communications protocol server. This Communication Driver is hosted by the OI Server Manager, a Microsoft Management Console (MMC) snap-in, which is part of the Operations Control Management Console (OCMC) suite of utilities.

This Communication Driver allows other Windows application programs access to data in PLCs (also referred to as devices) attached to an Ethernet network. The server can operate in either stand-alone mode or connect with any OPC, DDE, or SuiteLink compliant client application.

This Communication Driver documentation covers only the information you need to configure and run the Communication Driver component. See the documentation that comes with the related components for details on their operation. You can find installation instructions in a help file on the distribution CD. Many high-level functions and user-interface elements of the OI Server Manager are universal to all Communication Drivers, and only the documentation for the OI Server Manager contains descriptions of those universal functions/UI elements. Therefore, reading the documentation for both the MMC and the OI Server Manager is critical to understanding this user's guide. To read the documentation about the MMC and OI Server Manager, click the **Help Topics** on the OCMC **Help** menu. The Communication Drivers Pack documentation set opens.

Note: The shortcut menu items described in this document typically represent only a subset of any actual shortcut menu. Most items in each shortcut menu are standard Windows commands. See the MMC Help for more information about those commands.

Supported Hardware and Software

These are the technical specifications for the OMRONFINS Communication Driver.

Supported Hardware and Software

This Communication Driver connects to Omron PLCs on a FINS network, which includes the following devices:

- CS1, CJ1, CJ2, and CP1 series using CS1 mode
- CV, CVM1, and CVM1D series using CV mode
- Any other device that is FINS compatible

For Ethernet communication, CS1 PLCs can use either the FINS – UDP (Ethernet) protocol or the RS-232 protocol. (The specific protocol can be selected in the communication settings.).

For serial communication, CS1 PLCs use a FINS – Host link.

To program CS1 PLCs, you need the CX-Programmer programming software. For more information, go to: http://industrial.omron.eu/en/products/catalogue/automation_systems/software/configuration/cx-one/cx-programmer.html

---

Note: Connecting to a PLC via ETN is tested and supported. However, connecting via an EIP Ethernet card is not supported.

For a list of supported hardware and firmware versions,refer to the **Technology Matrix** section of the **Global Customer Support** (GCS) Site.

Conformance

The following hardware and software was used for conformance testing of this Communication Driver.

CS1 mode:

- CJ1 – CPU12
- CJ2 – CPU64 – EIP
- CS1G – CPU45H
- NJ501 -1300
- CP1L – CPU M

CV mode:

- CVM1-V2 – CPU11

C-series mode:

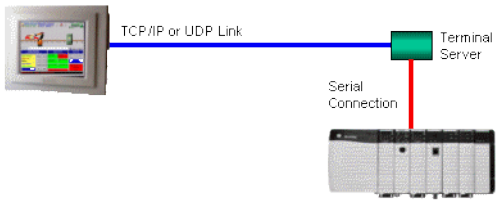
- C200HG - CPU43

Direct Serial Communication and Serial Encapsulation

This Communication Driver to match those settings.

Serial encapsulation enables serial communication with a target device that is connected to a terminal server on your TCP/IP or UDP/IP network. The terminal server is like a virtual serial port. It converts TCP or UDP messages to serial data and then relays them to the connected device. If you plan to use serial encapsulation, the target

device should already be connected to the terminal server, and you need to know the IP address of the terminal server and the port number assigned to the device.



Chapter 2

Configuring the OMRONFINS Communication Driver

Each server instance has its own hierarchy of objects, and each object has parameters that you need to configure in order to establish communication between the Communication Driver and individual devices on the network.

You can view the server configuration hierarchy of a OMRONFINS instance under its **Configuration** node.

This section only describes how to configure object parameters for a OMRONFINS server instance. For more general information about adding and configuring objects, see "Configuring Your OI Server" in the Communication Drivers Pack Help.

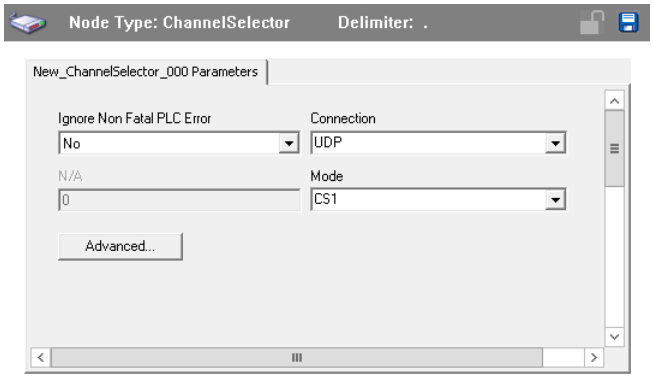
Adding and Configuring Channel Selector Object

The server-specific configuration of the OMRONFINS Communication Driver hierarchy tree under the OI Server Manager starts at the Channel Selector object. This object lets you set server parameters for communication with agents (devices) in the hierarchy tree.

To add a ChannelSelector connection to your OMRONFINS hierarchy

1. In the console tree, right-click **Configuration** and then click **Add ChannelSelector Connection**. The **New_ChannelSelector_000** object appears in the hierarchy.
2. Edit the object name to appropriately describe components of your specific hardware environment. If you do not rename the object at this time, a numeric sequencing system is applied. You can rename the hierarchy entry later.

   The **New_ChannelSelector_000 Parameters** view is displayed.



Configuring Channel Selector Objects

To configure the serial communication settings for a channel:

In the Operations Integration Server Manager, navigate to the ChannelSelector object that you want to configure. The following channel parameters are displayed in the details pane:

- Ignore Non Fatal PLC Error
- Connection
- Mode

Ignore Non Fatal PLC Error

The error mode, determines whether to ignore non-fatal errors on devices. Select one of the following options:

Option	Description
No	When a non-fatal error happens on a device, the channel shows status code 64 and all device items are set to BAD quality
Yes	When a non-fatal error happens on a device, the channel shows status code 0 (no errors) and sends a warning message to the log. All the device items stay at GOOD quality

Connection

The specific transport protocol to be used. Select one of the following options.

Option	Description
UDP	UDP/IP Ethernet network
Serial	Direct serial communication

If you select Connection as Serial, the Serial Encapsulation section appears. You have to configure the serial communication settings as described in the next section.

Configuring the Serial Encapsulation

From the **Serial Encapsulation** list, select one of the encapsulation modes:

- **None**: Direct serial communication with the device.
- **TCP/IP**: Serial encapsulation over a TCP/IP Ethernet connection.
- **UDP/IP**: Serial encapsulation over a UPD/IP Ethernet connection

1. If the Serial Encapsulation is set to None, the **Serial Port** section appears, and should be configured.
   a. **COM**: Select the COM port to which the target device is connected.
   b. **Stop Bits, Baud Rate**, **Parity** and **Data Bitsarity**: Configure the serial communication settings to match the settings that have already been configured on the target device.
2. If the Serial Encapsulation is set to TCP/IP or UDP/IP, the **Network** section appears, and should be configured.
   a. **IP Address**: Enter the **IP address** of the terminal server.
   b. **Port Number**: Enter the port number on the terminal server that has been assigned to the target device.
   c. **Server Mode**: Select Server Mode check-box if you want to make the target device responsible for establishing communication with this Communication Driver.

   Note: This option is not available for UDP/IP.
3. In the **Block Size (Words)** field, enter the maximum number of bytes that will be requested from the device in a single read operation.

   Note: You can type any value from 1 to 15. The default value is 7.
4. In the **Write word bits (0=No/1=Yes)** field**:**
   - Enter 1, to allow writing to the individual bits of a Word.
   - Enter 0, to not allow writing to individual bits of a Word.
5. In the **Signed or Unsigned Value** dropdown list, select the type of the register value.
6. Use **Advanced** option to modify additional communication settings. For more information see Advanced Settings.

Mode

The FINS compatibility mode, which determines the types of Omron devices that the Communication Driver can communicate with. Select one of the following options:

Option	Description
CS1	CS1, CJ1, CJ2, CP1 series
CV	CV, CVM1, CVM1D series
C-series	C200 series

For a single device, the Mode should be configured as CS1, as this acts as the default setting for all child objects. To add subsequent devices, a Channel Selector objects should be added for each additional device, and configured with mode as CV.

When you specify a mode for a single device, it overrides the global Mode setting that you previously configured in the channel's communication settings.

This parameter is optional; if no value is specified, the global setting is used by default.

Advanced

Click this button to open the **Advanced Settings** dialog box, which provides access to additional communication settings such as timeouts, retries, and buffer sizes. You might need to change these settings if the Communication Driver behaves unexpectedly during run time, but the default settings should work for most network configurations. For more information about these settings, see "Advanced Settings" in Communication Drivers Pack Help.

Adding and Configuring Device Selector Objects

The OMRONFINS Communication Driver can connect to different Windows agents, PLCs, and other data sources. These connections are modeled in the hierarchy by means of Device Selector objects, each of which models the end-point of the communications path.

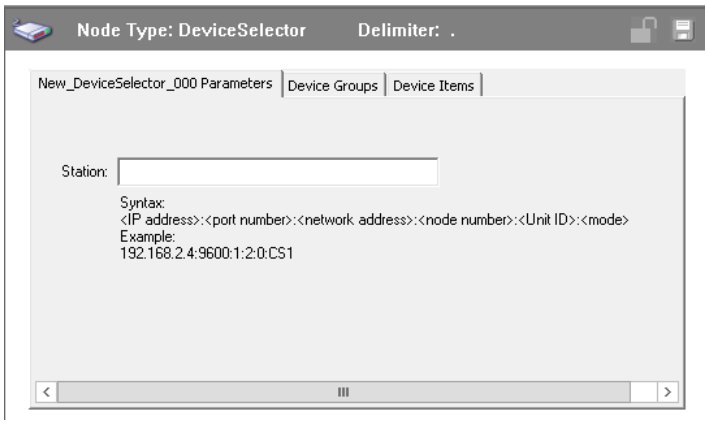
From the **ChannelSelector** branch of the Communication Driver hierarchy, create the new **DeviceSelector** object.

To add a Device Selector connection to your OMRONFINS hierarchy

1. Right-click the **New_ChannelSelector_000** object, and select **Add DeviceSelector Connection**.

   The **New_DeviceSelector_000** object is created.

2. Rename the object as appropriate.
3. **The New_DeviceSelector_000 Parameters** configuration view is displayed.



To configure the Device Selector connection

- Configure the **Station**.

  The Station field cannot be empty. For more information, see Setting Station ID of a Device.

Setting the Station ID of a Device

Set the station ID for a selected device so that the Communication Driver can identify and communicate with it on the network.

Syntax

For Ethernet communication, the device's station ID must use the following syntax:

```
<IP address>:<port number>:<network address>:<node number>:<Unit ID>:<Mode>
```

For serial communication, the device's station ID must use the following syntax:

```
<subnet>:<node>:[unit ID]
```

Note: Connecting to a PLC via UDP communication, routing through multiple remote PLCs over a secondary physical Ethernet subnet is not supported.

The following syntax diagram shows all of the possible options:

{ | IP address:port number: } subnet:node { | :unit ID { | : { CS1 | CV } } }

where,

IP address: The IP address of the device on your UDP/IP Ethernet network.

port number: The port number is the UDP port, used by the FINS protocol on the device. It is typically 9600, but it can be changed in the device settings.

network address: The network address is the device's subnet address, according to the FINS protocol.

node number: The device's node number (1-254) on the specified FINS subnet. If you do not specify a value, the last used node number is automatically inserted.

Unit ID: The ID number used for PC interface, as configured in the PLC program. This parameter is optional; if no value is specified, the default value is 0.

Note: This is not the Unit ID that is configured in the device's hardware settings.

Mode: The FINS communication mode determines the types of Omron devices that the Communication Driver can communicate with. This parameter is optional; if no value is specified, the default value is configured in the global setting.

This Communication Driver does not support simultaneous connections when communicating with devices that use either the old Ethernet communication modules (e.g., CS1W-ETN01, CS1W-ETN11, CJ1W-ETN11) or the new Ethernet communication modules that have "ETN11-compatible mode" enabled (e.g., CS1W-ETN21, CJ1W-ETN21). To address this problem, configure each device to use a different port number for the FINS protocol (e.g., 9600, 9601), and then when you specify the station ID for each device, include the complete IP address and port number (e.g., 192.168.1.10:9600, 192.168.1.20:9601).

This Communication Driver should support simultaneous connections when communicating with devices that use only the new Ethernet communication modules that are able to send responses to the source, even if the port number used by the FINS protocol on the source is different from port numbers used by the FINS protocol on the target devices. However, you must make sure those modules do not have "ETN11-compatible mode" enabled.

Examples

Examples of valid station IDs:

192.168.2.22:9600:0:22:0:CS1

0:1:0

In case of multiple devices being added to the OmronFINS Communication Driver configuration, the examples of station IDs are:

Station ID of all devices except the last device added to the configuration: 192.168.2.22:9600:0:22:0:CS1

Station ID of the last device added to the configuration: 192.168.2.22:9600:0:22

Configuring Station Parameters When Device Uses Controller Link

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

Driver Runtime Parameter	PLC 1 (Source) Parameters	PLC 2 (Destination) Parameters
Local IP	• Ethernet IP • Port Number • Network • Node Address • Unit ID (optional) • Mode (optional)	• Network • Node Address • Unit ID (optional) • Mode (optional)

Then to successfully communicate with PLC2:

1. Configure the Source station (PLC1) station under the Channel Selector settings -> **Advanced** as `<Network of PLC1>:<Node Address of PLC1>`

   The source node is configured in the **Station** field of **Advanced Settings** dialog box of the **Channel Selector** page with the syntax `<Network number>:<Source Node>`. For more information see Setting the Station ID of a Device.

2. Configure the Destination station (PLC2) 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)>`

Note: The driver does not validate the source node and accepts a non-zero value at runtime. If you are running the driver on a machine with multiple NICs, then the **Station** field has to be specified along with the source node incase the source node is a requirement to connect to the PLC. If you need to connect to multiple NICs, then another instance of the Communication Driver (another channel selector and device selector pair) should be recreated for specifying any additional source and Station IP.

Examples of Communication Driver Configurations

Example 1: Using an Ethernet PLC to route to various PLCc via Controller Link

Recommended Communication Driver Configuration:

1. Channel Selector Parameters -> Connection:

   Select **UDP**

2. Channel Selector Parameters -> Advanced -> Protocol -> Station:

   Enter the station of the local server X:YYY [where X is the Ethernet network ID; YYY is the last Octet of the IP of the local server]

   Example: 2:147

3. Device Selector Parameters -> Station:

Set the Station to <IP of the routing PLC>:9600:<Network ID#>:<Remote PLC Node #>:0:CS1

Example: xxx.xxx.xxx.xxx:9600:9:4:0:CS1

Device Groups and Device Items

Device Group Definitions

Use the **Device Groups** configuration view, to create, add, delete, and define device groups. You can also configure default update intervals for the objects and edit update intervals in this dialog box. To open the Device Groups dialog box, in the Device Selector configuration editor, click the **Device Groups** tab.

Note: When you select another part of the Communication Driver tree hierarchy, you are prompted to save the modifications to the configuration set.

To create or add device groups

1. Right-click anywhere in the table, and then click **Add**. A device group is added with a default name and update interval.
2. Enter a unique name up to 32 characters long for the device group.

To delete device groups

1. Right-click the device group to be deleted, and then click **Delete**.
2. Read the warning, and then click **Yes**.

To edit device groups

Use the **Edit** option from the **Device Groups** tab only for configuring the Communication Driver's unsolicited message handling.

To configure default update intervals

To configure a default update interval for the object, right-click in the **Device Groups** box and then click **Config Default Update Interval**.

To edit update intervals

To edit the update interval for an object, double-click its value in the **Update Interval** column and make the edits.

or

Right-click its value in the **Update Interval** column and then click **Modify Update** Interval.

The update interval is the frequency, in milliseconds, that the Communication Driver acquires data from the topics associated with that device group.

Different topics can be polled at different rates from a PLC by defining multiple device group names for the same PLC and setting a different update interval for each device group.

Device Item Definitions

The device item name is an is an "alias" or a label for the data in the device. It is an alternative name for the item reference, and ca be used instead of the item reference when you create the client application. Device item configuration is optional, but is strongly recommended.

To create or add device items

1. Right-click anywhere in the table, and then click **Add**.
2. In the **Name** column, type a unique item name. The maximum is 32 characters.
3. In the corresponding line, double-click the **Item Reference** column and enter the correlated item reference for the name you created.

To rename device items

Right-click the device item to be renamed and click **Rename**. Make the changes.

To delete device items

Right-click the device item to be deleted from the list and click **Delete**.

To clear all device items

Right-click in the **Device Items** box and click **Clear All**. All the device items listed are cleared after you confirm their deletion.

NOTE: You can import a .csv file containing your item definitions to help streamline configuration. See "Exporting and Importing CSV Files" in the Communication Drivers Pack Help.

AVEVA

Chapter 3

OMRONFINS Communication Driver Reference

Item Reference Syntax

Use item references to access data stored in memory registers in connected devices, as well as to access standard system items in the Communication Driver itself.

This section only describes the item reference syntax and options for the OMRONFINS server. For more general information about item references, see "Managing Device Items" and "Item Reference Descriptions" in the Communication Drivers Pack Help.

Item references in this Communication Driver use the following syntax.

For all multi-byte memory registers, use the following syntax:

```
<register type><address>@[data type]
<register type><address>[data suffix]
```

For bit-accessible memory registers — i.e., all types except PVC and PVT — use the following syntax:

```
<register type><address>:<bit>
<register type><address>.<bit>
```

For strings, use the following syntax:

```
<register type><starting address>-<register type><ending address>C
```

The following syntax diagram shows all of the possible options:

{ { CIO | A | H | W | D | EMarea }address (or starting address){ { @Short | S } | { @BCD | B } | { @Long | L } | @DWord | { @LBCD | M } | { @Float | F } | { : | . }bit | -{ CIO | A | H | W | D | EMarea }ending addressC } | { PVC | PVT }address (or starting address){ { @Short | S } | { | @Word | U } | { @BCD | B } | -{ PVC | PVT }ending addressC } }

where,

register type: The register type or memory area. For extended data memory (EM), the available areas include A through M.

address: The specific memory address on the device.

data type / data suffix: The formatted data type of the value read from the device memory. his parameter is optional; if the data type is not specified, the value will be formatted as Word (i.e., unsigned,16-bit decimal) by default.

Bit: The specific bit number in the memory address. Bit 0 is the low bit. Bit 15 is the high bit.

starting address, ending address: The starting and ending addresses of a string. Strings are stored as ASCII data. Each two-byte register contains two characters, and the string length can be from 1 to 40 registers (i.e., 2 to 80 characters).

Address Descriptions

The address descriptions consist of the register type, its item name and the allowable range of values, the default data type, allowable suffixes (if any), and allowable access methods.

Register Type	Item Reference		Length	Data Types	Access
	Item Name	Address Range			
CIO Auxiliary Relay	CIO A or AR	0000 to 6143	2 Bytes	Short, BCD	Read/Write
		0000 to 6142	4 Bytes	Long, DWord, LBCD, Float	Read/Write
		0000.00 to 6143.15 000 to 447	1Bit 2Bytes	Boolean Short, BCD	Read/Write Readonly
		448 to 959	2Bytes	Short, BCD	Read/Write
		000 to 446	4Bytes	Long, DWord, LBCD, Float	Readonly
		448 to 958 000.00 to 447.15	4Bytes 1Bit	Long,DWord, LBCD, Float Boolean	Read/Write Readonly
		448.00 to 959.15	1Bit	Boolean	Read/Write
Holding Relay Working Relay	H or HR W	000 to 511	2Bytes	Short, BCD	Read/Write
		000 to 510	4Bytes	Long, DWord, LBCD, Float	Read/Write
		000.00 to 511.15 000 to 511	1Bit 2Bytes	Boolean Short, BCD	Read/Write Read/Write
		000 to 510	4Bytes	Long, DWord, LBCD, Float	Read/Write
		000.00 to 511.15	1Bit	Boolean	Read/Write

Data Memory	D or DM	0000 to 32767	2Bytes	Short, BCD	Read/Write
		0000 to 32766	4Bytes	Long, DWord, LBCD, Float	Read/Write
		0000.00 to 32767.15	1Bit	Boolean	Read/Write
Expansion Data Memory	EM	A0000 to M32767 A0000 to M32766	2Bytes 4Bytes	Short, BCD Long, DWord, LBCD, Float	Read/ WriteRead/ Write
		A0000.00 to M32767.15	1Bit	Boolean	Read/Write
Present Value Counter	PVC	0000 to 4095	2Bytes	Short, Word, BCD	Read/Write
Present Value Timer	PVT	0000 to 4095	2Bytes	Short, Word, BCD	Read/Write

Supported Data Types

The data type is specified as a suffix in the item syntax.This Communication Driver supports the following datatypes.

DataType	**Suffix**	**Description**	**Range of Values**
Boolean	by bit	Single bit.	0 or 1
Short	@Short, S	Signed, 16-bit decimal. Bit0 is the low bit. Bit14 is the high bit. Bit 15 is the sign bit.	-32768 to 32767
Word	@Word, U	Unsigned, 16-bit decimal. Bit0 is the low bit. Bit15 is the high bit.	0 to 65535
Long	@Long, L	Signed, 32-bit decimal. Bit0 is the low bit. Bit30 is the high bit. Bit 31 is the sign bit.	-2147483648 to 2147483647
DWord	@DWord	Unsigned, 32-bit decimal. Bit0 is the low bit. Bit31is the high bit.	0 to 4294967295
Float	@Float, F	32-bit real number.	N/A
BCD	@BCD, B	Two-byte packed BCD. Value range is 0–9999. Behavior is undefined for values beyond this range.	0 to 9999

LBCD	@LBCD, M	Four-byte packed BCD. Value range is 0–99999999. Behavior is undefined for values beyond this range.	0 to 99999999
String	C	Null terminated ASCII string.	2 to 80 characters per string

Examples of Item References

These are examples of valid item references for this Communication Driver. For more information about the referenced addresses, see the documentation by the manufacturer for your device.

Register Type	**Address on the Device**	**Item Reference**
CIO	CIO 5000 as Short	CIO5000@Short
		CIO5000S
	CIO 5000 as BCD	CIO5000@BCD
		CIO5000B
	CIO 5000 as Long	CIO5000@Long
		CIO5000L
	CIO 5000 as DWord	CIO5000@DWord
	CIO 5000 as LBCD	CIO5000@LBCD
		CIO5000M
	CIO 5000 as Float	CIO5000@Float
		CIO5000F
	CIO 5000 as Bit	CIO5000.00
	CIO 5000 as String(length of 8 characters)	CIO5000-CIO5003C
Auxiliary Relay	A000 as Short	A000@Short
		A000S
	A000 as BCD	A000@BCD
	A000 as Long	A000B
		A000@Long
		A000L
	A000 as DWord	A000@DWord
	A000 as LBCD	A000@LBCD

Working Relay Data Memory		D102M
	D00102 as Float	D102@Float D102F
	D00102 as Bit D00102 as String(length of 8 characters)	D102.00 D102-D105C
Expansion Data Memory	EMA00200 as Short EMA00200 as BCD EMA00200 as Long	EMA00200@Short
		EMA00200S EMA00200@BCD
		EMA00200B EMA00200@Long
		EMA00200L
	EMA00200 as DWord EMA00200 as LBCD EMA00200 as Float	EMA00200@DWord EMA00200@LBCD
		EMA00200M EMA00200@Float
		EMA00200F
	EMA00200 as Bit	EMA00200.00
	EMA00200 as String(length of 8 characters)	EMA00200-EMA00203C
Present Value Counter	PVC100 as Short	PVC100@Short PVC100S
	PVC100 as Word	PVC100
		PVC100@Word
		PVC100U
	PVC100 as BCD	PVC100@BCD PVC100B
	PVC100 as String(length of 8 characters)	PVC100-PVC103C
Present Value Timer	PVT123 as Short PVT123 as Word PVT123 as BCD	PVT123@Short
		PVT123S PVT123

	PVT123 as String(length of 8 characters)	PVT123@Word
		PVT123U PVT123@BCD
		PVT123B PVT123-PVT126C

Chapter 4

Troubleshooting the OMRONFINS Communication Driver

OMRONFINS Communication Driver Error Codes

The following tables describe the additional error codes that you might receive when poll/poke requests and operations fail.

Code	Description	Possible Causes	Solution
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 communication 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 Communication Driver is connected to an unsupported device.
4	RX Buffer Over flow	The number of bytes sent by the PLC exceeds the configured buffer size.	Increase the buffer size in the communication 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.
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.
-34	Invalid Address	Invalid address specified in Address field	Specify a valid address.
-38	Invalid Station	Invalid station specified in Station field.	Specify a valid station number.
-15	Time out waiting to start a message.	• Disconnected cables. • PLC is turned off, or it is in Stop or Error mode. • Incorrect station number. • Incorrect RTS/CTS control settings. • Wrong parity(for serial communication).	• Check the cable wiring. • Check the PLC state. It must be in Run mode. • Check the station number. • Increase the time out in the driver's advanced settings. • Check the RTS/CTS control settings in the channel settings.
-17	Time out between rx char. Invalid Block Size	• PLC in Stop or Error mode. • Incorrect station number. • Incorrect parity. • Incorrect RTS/CTS control settings.	• Check the cable wiring. • Check the PLC state. It must be in Run mode. • Check the station number. • Check the RTS/CTS control settings in the channel settings.

Enabling Support for GB2312 encoding

The Long2 field, labelled as **N/A** in the Channel Selector settings, is by default set to 0. This field is disabled and cannot be edited in the Channel Selector settings.This default setting of the OMRONFINS Communication Driver does not support GB2312 encoding (Chinese characters).

To enable GB2312 encoding

1. Open the **OMETH.ini** file with a text editor:
   - In 32-bit machines, this file is located in:

     C:\Program Files\Wonderware\OI-Server\OI-OMRONFINS
   - In 64-bit machines, this file is located in:

     C:\Program Files (x86)\Wonderware\OI-Server\OI-OMRONFINS
2. Under the `[CommParamDefault]` entry, change `Long2=0` to `Long2=1`.

```
[CommParamDefault]
Station=0
String1=UDP
String2=CS1
Long2=1
SimultaneousRequestsMaximum=1
SimultaneousRequestsPerStation=1
```

3. Save the file in the same location.

   The value in the **N/A** field in the Channel Selector settings changes to 1. This implies that GB2312 encoding is enabled.

   Note: If the value does not change in the existing Channel Selector, create a new Channel Selector.