

Allen-Bradley Unsolicited Ethernet

© 2018 PTC Inc. All Rights Reserved.

Table of Contents

Allen-Bradley Unsolicited Ethernet	1
Table of Contents	2
Allen-Bradley Unsolicited Ethernet	3
Overview	3
Setup	4
Channel Properties — General	4
Channel Properties — Ethernet Communications	5
Channel Properties — Write Optimizations	5
Channel Properties — Advanced	6
Device Properties — General	7
Device Properties — Scan Mode	9
Device Properties — Communication Parameters	9
Device Properties — Options	10
Master Device Configuration	11
Data Types Description	12
Address Descriptions	13
Event Log Messages	14
Failure to initiate winsock.dll. OS error = <error>.	15
Failure to start unsolicited communications.	15
Failed to stop winsock communications.	15
Ethernet Manager started.	15
Ethernet Manager stopped.	15
Index	16

Allen-Bradley Unsolicited Ethernet

Help version 1.033

CONTENTS

[Overview](#)

What is the Allen-Bradley Unsolicited Ethernet?

[Device Setup](#)

How do I configure a device for use with this driver?

[Data Types Description](#)

What data types does this driver support?

[Address Descriptions](#)

How do I address a data location on a simulated device?

[Event Log Messages](#)

What messages does this driver produce?

Overview

The Allen-Bradley Unsolicited Ethernet provides a reliable way to connect Allen-Bradley Ethernet PLCs to client applications; including HMI, SCADA, Historian, MES, ERP, and countless custom applications.

Setup

The Allen-Bradley Unsolicited Ethernet acts as a single simulated Allen-Bradley PLC-2. Up to 256 devices may connect to the simulated PLC-2 at any time. Only one Allen-Bradley Slave Ethernet device can be connected per channel in the server configuration. Only PLC-2 Unprotected Reads and Writes to register memory are currently supported.

• For more information, refer to [Master Device Configuration](#).

Supported Devices

Allen-Bradley PLCs are programmed to send PLC-2 type commands. In this case, ControlLogix processors are not supported. For more information on command compatibility, refer to the hardware programming manual.

• **Note:** Devices do not need to be PLC-2s to communicate with this driver, but do require specialized ladder programming.

Communication Protocol

Allen-Bradley Ethernet

PLC Configuration

Devices on the network must be programmed to send Read and Write messages to the driver as well as handle returned data.

Sockets

Sockets used by incoming connections are aged according to activity level. Sockets with little or no activity are closed and reused as new attempts to connect are made.

Channel Properties — General

This server supports the use of simultaneous multiple communications drivers. Each protocol or driver used in a server project is called a channel. A server project may consist of many channels with the same communications driver or with unique communications drivers. A channel acts as the basic building block of an OPC link. This group is used to specify general channel properties, such as the identification attributes and operating mode.

Property Groups	<input type="checkbox"/> Identification	
General	Name	
Write Optimizations	Description	
Advanced	Driver	
	<input type="checkbox"/> Diagnostics	
	Diagnostics Capture	Disable

Identification

Name: User-defined identity of this channel. In each server project, each channel name must be unique. Although names can be up to 256 characters, some client applications have a limited display window when browsing the OPC server's tag space. The channel name is part of the OPC browser information.

• For information on reserved characters, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in the server help.

Description: User-defined information about this channel.

Many of these properties, including Description, have an associated system tag.

Driver: Selected protocol / driver for this channel. This property specifies the device driver that was selected during channel creation. It is a disabled setting in the channel properties.

Note: With the server's online full-time operation, these properties can be changed at any time. This includes changing the channel name to prevent clients from registering data with the server. If a client has already acquired an item from the server before the channel name is changed, the items are unaffected. If, after the channel name has been changed, the client application releases the item and attempts to re-acquire using the old channel name, the item is not accepted. With this in mind, changes to the properties should not be made once a large client application has been developed. Utilize the User Manager to prevent operators from changing properties and restrict access rights to server features.

Diagnostics

Diagnostics Capture: When enabled, this option makes the channel's diagnostic information available to OPC applications. Because the server's diagnostic features require a minimal amount of overhead processing, it is recommended that they be utilized when needed and disabled when not. The default is disabled.

Note: This property is not available if the driver does not support diagnostics.

For more information, refer to "Communication Diagnostics" in the server help.

Channel Properties — Ethernet Communications

Ethernet Communication can be used to communicate with devices.

Property Groups	Ethernet Settings	
General	Network Adapter	Default
Ethernet Communications		
Write Optimizations		
Advanced		

Ethernet Settings

Network Adapter: Specify the network adapter to bind. When Default is selected, the operating system selects the default adapter.

Channel Properties — Write Optimizations

As with any server, writing data to the device may be the application's most important aspect. The server intends to ensure that the data written from the client application gets to the device on time. Given this goal, the server provides optimization properties that can be used to meet specific needs or improve application responsiveness.

Property Groups	Write Optimizations	
General	Optimization Method	Write Only Latest Value for All Tags
Write Optimizations	Duty Cycle	10

Write Optimizations

Optimization Method: controls how write data is passed to the underlying communications driver. The options are:

- **Write All Values for All Tags:** This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing data to the device as quickly as possible. This mode ensures that everything written from the client applications is sent to the target device. This mode should be selected if the write operation order or the write item's content must uniquely be seen at the target device.
- **Write Only Latest Value for Non-Boolean Tags:** Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the server's internal write queue and sent to the device at the next possible opportunity. This can greatly improve the application performance.
 - **Note:** This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.
- **Write Only Latest Value for All Tags:** This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write queue before they are sent. This is the default mode.

Duty Cycle: is used to control the ratio of write to read operations. The ratio is always based on one read for every one to ten writes. The duty cycle is set to ten by default, meaning that ten writes occur for each read operation. Although the application is performing a large number of continuous writes, it must be ensured that read data is still given time to process. A setting of one results in one read operation for every write operation. If there are no write operations to perform, reads are processed continuously. This allows optimization for applications with continuous writes versus a more balanced back and forth data flow.

● **Note:** It is recommended that the application be characterized for compatibility with the write optimization enhancements before being used in a production environment.

Channel Properties — Advanced

This group is used to specify advanced channel properties. Not all drivers support all properties; so the Advanced group does not appear for those devices.

Property Groups	<input type="checkbox"/> Non-Normalized Float Handling	
General	Floating-Point Values	Replace with Zero
Write Optimizations	<input type="checkbox"/> Inter-Device Delay	
Advanced	Inter-Device Delay (ms)	0

Non-Normalized Float Handling: A non-normalized value is defined as Infinity, Not-a-Number (NaN), or as a Denormalized Number. The default is Replace with Zero. Drivers that have native float handling may default to Unmodified. Non-normalized float handling allows users to specify how a driver handles non-normalized IEEE-754 floating point data. Descriptions of the options are as follows:

- **Replace with Zero:** This option allows a driver to replace non-normalized IEEE-754 floating point values with zero before being transferred to clients.
- **Unmodified:** This option allows a driver to transfer IEEE-754 denormalized, normalized, non-number, and infinity values to clients without any conversion or changes.

● **Note:** This property is not available if the driver does not support floating point values or if it only supports the option that is displayed. According to the channel's float normalization setting, only real-time driver tags (such as values and arrays) are subject to float normalization. For example, EFM data is not affected by this setting.

● For more information on the floating point values, refer to "How To ... Work with Non-Normalized Floating Point Values" in the server help.

Inter-Device Delay: Specify the amount of time the communications channel waits to send new requests to the next device after data is received from the current device on the same channel. Zero (0) disables the delay.

● **Note:** This property is not available for all drivers, models, and dependent settings.

Device Properties — General

A device represents a single target on a communications channel. If the driver supports multiple controllers, users must enter a device ID for each controller.

Property Groups	Identification	
General	Name	
Scan Mode	Description	
	Channel Assignment	
	Driver	
	Model	
	ID Format	Decimal
	ID	2
	Operating Mode	
	Data Collection	Enable
	Simulated	No

Identification

Name: This property specifies the name of the device. It is a logical user-defined name that can be up to 256 characters long, and may be used on multiple channels.

● **Note:** Although descriptive names are generally a good idea, some OPC client applications may have a limited display window when browsing the OPC server's tag space. The device name and channel name become part of the browse tree information as well. Within an OPC client, the combination of channel name and device name would appear as "ChannelName.DeviceName".

● For more information, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in server help.

Description: User-defined information about this device.

● Many of these properties, including Description, have an associated system tag.

Channel Assignment: User-defined name of the channel to which this device currently belongs.

Driver: Selected protocol driver for this device. This property specifies the driver selected during channel creation. It is disabled in the channel properties.

Model: This property specifies the specific type of device that is associated with this ID. The contents of the drop-down menu depends on the type of communications driver being used. Models that are not supported by a driver are disabled. If the communications driver supports multiple device models, the model selection can only be changed when there are no client applications connected to the device.

● **Note:** If the communication driver supports multiple models, users should try to match the model selection to the physical device. If the device is not represented in the drop-down menu, select a model that conforms closest to the target device. Some drivers support a model selection called "Open," which allows users to communicate without knowing the specific details of the target device. For more information, refer to the driver help documentation.

ID: This property specifies the device's station / node / identity / address. The type of ID entered depends on the communications driver being used. For many drivers, the ID is a numeric value. Drivers that support a Numeric ID provide users with the option to enter a numeric value whose format can be changed to suit the needs of the application or the characteristics of the selected communications driver. The ID format can be Decimal, Octal, and Hexadecimal. If the driver is Ethernet-based or supports an unconventional station or node name, the device's TCP/IP address may be used as the device ID. TCP/IP addresses consist of four values that are separated by periods, with each value in the range of 0 to 255. Some device IDs are string based. There may be additional properties to configure within the ID field, depending on the driver.

Operating Mode

Data Collection: This property controls the device's active state. Although device communications are enabled by default, this property can be used to disable a physical device. Communications are not attempted when a device is disabled. From a client standpoint, the data is marked as invalid and write operations are not accepted. This property can be changed at any time through this property or the device system tags.

Simulated: This option places the device into Simulation Mode. In this mode, the driver does not attempt to communicate with the physical device, but the server continues to return valid OPC data. Simulated stops physical communications with the device, but allows OPC data to be returned to the OPC client as valid data. While in Simulation Mode, the server treats all device data as reflective: whatever is written to the simulated device is read back and each OPC item is treated individually. The item's memory map is based on the group Update Rate. The data is not saved if the server removes the item (such as when the server is reinitialized). The default is No.

● Notes:

1. This System tag (`_Simulated`) is read only and cannot be written to for runtime protection. The System tag allows this property to be monitored from the client.
2. In Simulation mode, the item's memory map is based on client update rate(s) (Group Update Rate for OPC clients or Scan Rate for native and DDE interfaces). This means that two clients that reference the same item with different update rates return different data.

● Simulation Mode is for test and simulation purposes only. It should never be used in a production environment.

Device Properties — Scan Mode

The Scan Mode specifies the subscribed-client requested scan rate for tags that require device communications. Synchronous and asynchronous device reads and writes are processed as soon as possible; unaffected by the Scan Mode properties.

Property Groups	<input type="checkbox"/> Scan Mode	
General	Scan Mode	Respect Client-Specified Scan Rate ▾
Scan Mode	Initial Updates from Cache	Disable

Scan Mode: specifies how tags in the device are scanned for updates sent to subscribing clients.

Descriptions of the options are:

- **Respect Client-Specified Scan Rate:** This mode uses the scan rate requested by the client.
- **Request Data No Faster than Scan Rate:** This mode specifies the maximum scan rate to be used. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
 - **Note:** When the server has an active client and items for the device and the scan rate value is increased, the changes take effect immediately. When the scan rate value is decreased, the changes do not take effect until all client applications have been disconnected.
- **Request All Data at Scan Rate:** This mode forces tags to be scanned at the specified rate for subscribed clients. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
- **Do Not Scan, Demand Poll Only:** This mode does not periodically poll tags that belong to the device nor perform a read to get an item's initial value once it becomes active. It is the client's responsibility to poll for updates, either by writing to the `_DemandPoll` tag or by issuing explicit device reads for individual items. *For more information, refer to "Device Demand Poll" in server help.*
- **Respect Tag-Specified Scan Rate:** This mode forces static tags to be scanned at the rate specified in their static configuration tag properties. Dynamic tags are scanned at the client-specified scan rate.

Initial Updates from Cache: When enabled, this option allows the server to provide the first updates for newly activated tag references from stored (cached) data. Cache updates can only be provided when the new item reference shares the same address, scan rate, data type, client access, and scaling properties. A device read is used for the initial update for the first client reference only. The default is disabled; any time a client activates a tag reference the server attempts to read the initial value from the device.

Device Properties — Communication Parameters

Property Groups	<input type="checkbox"/> Communication Parameters	
General	Port	2222
Scan Mode		
Communication Parameters		
Options		

Port: Specify the port number on which the driver should listen. Devices must be configured to connect to this port. Messages sent to all other ports are ignored by the driver. Values may range from 0 to 65535. The standard Allen-Bradley default is 2222.

- Non-standard values may be necessary due to routing and firewall issues. Changing this value during runtime can cause the driver to drop all existing connections.

Device Properties — Options

This property is used to specify word order for 32-bit data types. To specify **First Word Low**, choose Enable. To specify first word high, choose Disable. The default is enabled.

Property Groups	Options	
General	First Word Low	Disable
Scan Mode		
Communication Parameters		
Options		

Master Device Configuration

Allen-Bradley PLCs must be programmed to issue Read and Write commands to this driver in addition to handling returned data. For more information, consult the Allen-Bradley PLC programming documentation. Messages must be sent to both the IP address of the selected Ethernet adapter of the host computer and the port number configured for the simulated device. Access the settings under channel properties.

Supported Commands

PLC-2 Unprotected Read

PLC-2 Unprotected Write

Error Codes

This driver responds to all messages it receives. If it cannot complete the request, it returns a response message with a non-zero error code in the STS status byte of the PCCC frame structure. Ladder programs should be written to handle these errors. The following error codes may be sent by the driver.

Error Code	Description
0x00	Request processed successfully.
0x10	Command not supported by driver.
0x20	Command supported by driver, but was found to have invalid parameters.
0x50	Address out of range.

Data Types Description

Data Type	Description
Boolean	Single bit
Word	Unsigned 16-bit value bit 0 is the low bit bit 15 is the high bit
Short	Signed 16-bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32-bit value bit 0 is the low bit bit 31 is the high bit
Long	Signed 32-bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
BCD	Two byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range.
LBCD	Four byte packed BCD Value range is 0-99999999. Behavior is undefined for values beyond this range.
Float	32-bit Floating point value. The driver interprets two consecutive registers as a Floating point value by making the second register the high word and the first register the low word.

Address Descriptions

The default data type for each address is shown in **bold**.

Address Type	Syntax and Range	Data Types	Access
Word Register	w where w is word number 0-77777.*	Word , Short, BCD	Read/Write
Word Register	w where w is start word number 0-77776.* Two adjacent word registers are used to store these values. The first register contains the high word.	DWord, Long, LBCD, Float	Read/Write
Word Register	w/b where w is word number 0-77777* and b is the bit number 0-17.*	Boolean	Read/Write

*Octal

Array Support

Arrays are supported for all data types except Booleans. Array dimensions may range from 1 to 256 (decimal). One and two dimensional arrays are supported. The syntax is as follows:

w [rows] [cols]

w [cols]

● **Note:** It is assumed that rows equal 1.

Event Log Messages

The following information concerns messages posted to the Event Log pane in the main user interface. Consult the server help on filtering and sorting the Event Log detail view. Server help contains many common messages, so should also be searched. Generally, the type of message (informational, warning) and troubleshooting information is provided whenever possible.

Failure to initiate winsock.dll. | OS error = <error>.

Error Type:

Error

Possible Cause:

Could not negotiate with the operating system winsock 1.1 functionality.

Possible Solution:

Verify that the winsock.dll is properly installed on the system or install the library.

Failure to start unsolicited communications.

Error Type:

Error

Possible Cause:

The driver is not able to create a listen socket for unsolicited communications. Another application may be using the specified port or there may be low system resources.

Possible Solution:

Verify no other application is using the port. Network monitor software can be used to diagnose this. Shut down any conflicting applications and restart the server. In many cases, the conflicting application is free to pick any available port. If this is the case, start the server first so it claims the required port. The PLC programming software and the driver may not be able to be used simultaneously if they both need to use the same port.

• See Also:

Communications

Failed to stop winsock communications.

Error Type:

Error

Possible Cause:

Could not negotiate with the operating system winsock 1.1 functionality.

Possible Solution:

Verify that the winsock.dll is properly installed on the system or install the library.

Ethernet Manager started.

Error Type:

Informational

Ethernet Manager stopped.

Error Type:

Informational

Index

A

Address Descriptions 13

B

BCD 12

Boolean 12

C

Channel Assignment 7

Communication Parameters 9

D

Data Collection 8

Data Types Description 12

Description 7

Device Properties — General 7

Do Not Scan, Demand Poll Only 9

Driver 8

DWord 12

E

Ethernet Manager started. 15

Ethernet Manager stopped. 15

Event Log Messages 14

F

Failed to stop winsock communications. 15

Failure to initiate winsock.dll. | OS error = <error>. 15

Failure to start unsolicited communications. 15

Float 12

I

ID 8

Initial Updates from Cache 9

L

LBCD 12

Long 12

M

Master Device Configuration 11

Model 8

N

Name 7

O

Options 10

Overview 3

R

Request All Data at Scan Rate 9

Request Data No Faster than Scan Rate 9

Respect Client-Specified Scan Rate 9

Respect Tag-Specified Scan Rate 9

S

Scan Mode 9

Setup 4

Short 12

Simulated 8

W

Word 12