Allen-Bradley Ethernet Driver Help

Table of Contents

Allen-Bradley Ethernet Driver Help

Help version 1.040

CONTENTS

Overview

The Allen-Bradley Ethernet Driver provides an easy and reliable way to connect Allen-Bradley Ethernet devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. This driver supports the Allen Bradley SLC 5/05 series, PLC-5 series, and SoftPLC PLCs. Address ranges are open to support future models for this series of PLCs.

Device Setup

Supported Devices

SLC 5/05 processor*
PLC-5 series (excluding the PLC-5/250 series)
SoftPLC

*Address ranges have been opened up in the driver to allow for future devices. Therefore, the driver may still support a device even if it is not listed above.

Communication Protocol

Allen-Bradley Ethernet

Timeout Parameters

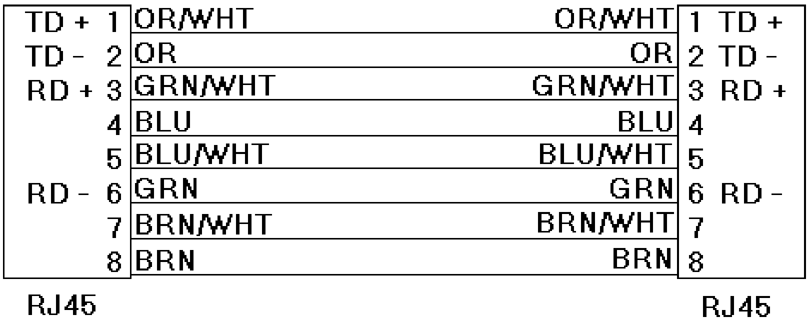
Connect Timeout
This parameter specifies the time that the driver waits for a connection to be made with a device. The valid range is 1 to 60 seconds.

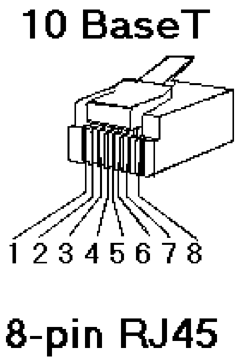
Request Timeout
This parameter specifies the time that the driver waits on a response from the device before giving up and going on to the next request. The valid range is 100 to 30000 milliseconds.

Retry Attempts
This parameter specifies the number of times that the driver attempt a message before giving up and going on to the next message. The valid range is 1 to 10.

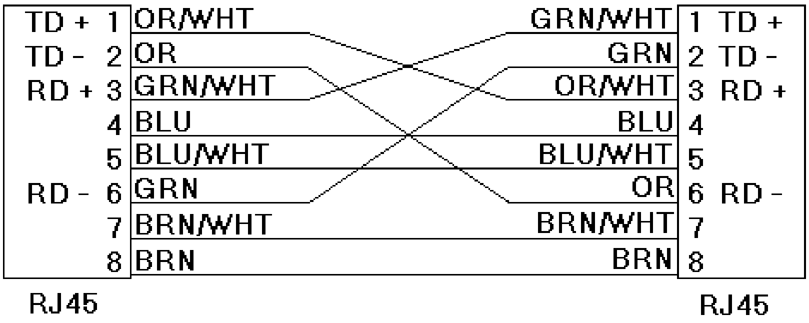
Cable Connections

Patch Cable (Straight Through)

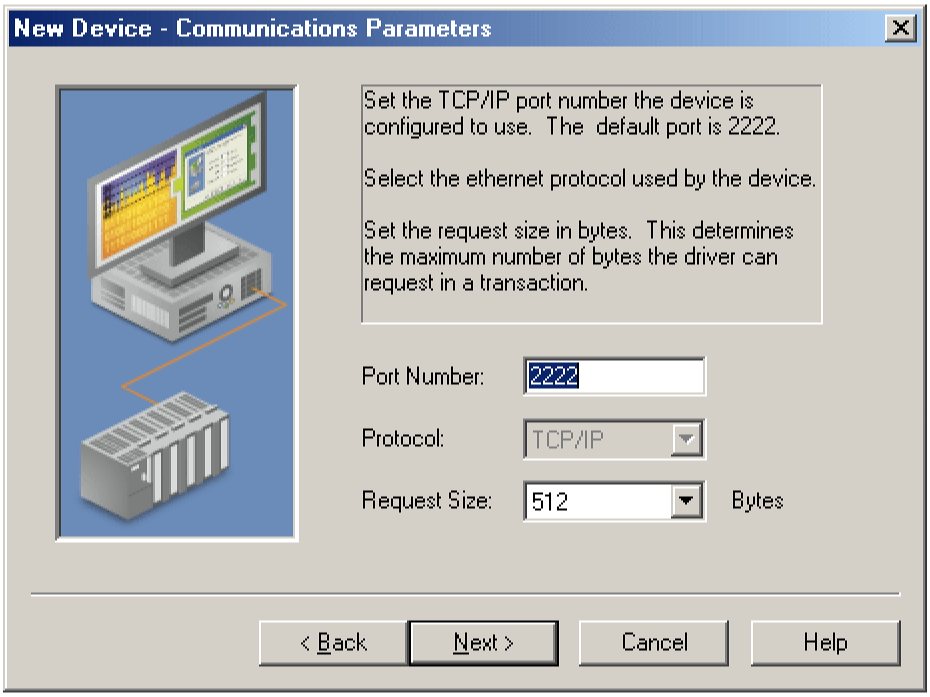




Crossover Cable



Communications Parameters



New Device - Communications Parameters

Set the TCP/IP port number the device is configured to use. The default port is 2222.

Select the ethernet protocol used by the device.

Set the request size in bytes. This determines the maximum number of bytes the driver can request in a transaction.

Port Number: 2222

Protocol: TCP/IP

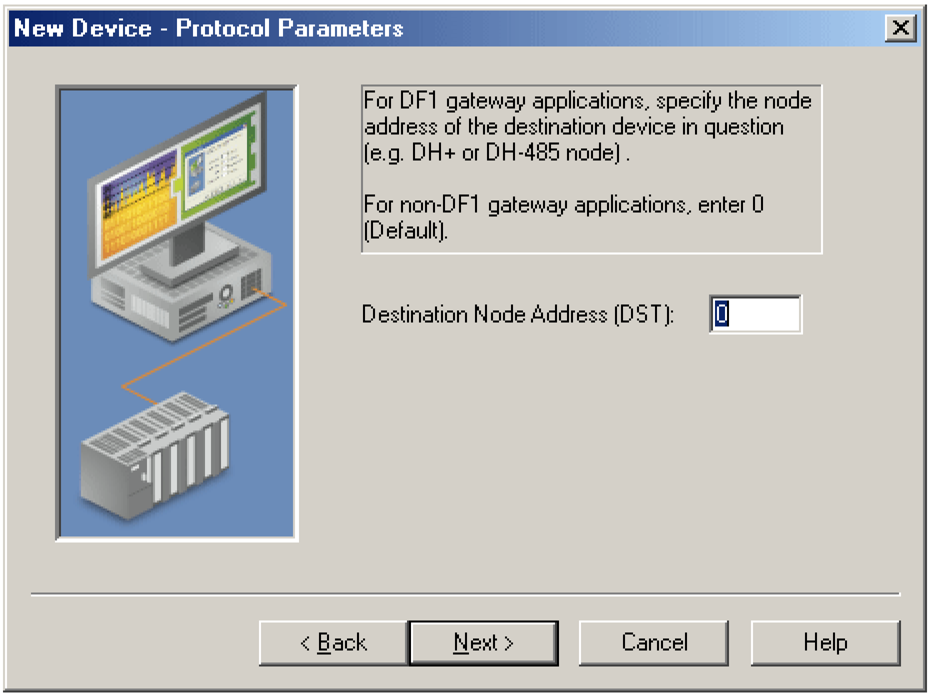
Request Size: 512 Bytes

Descriptions of the parameters are as follows:

- **Port Number:** This parameter specifies the port number that the remote device is configured to use. The default setting is 2222.
- **Protocol:** The Allen-Bradley Ethernet Driver connects to the supported devices using Transfer Control Protocol (TCP). Winsock V1.1 or higher is required.
- **Block Request Size:** This parameter specifies the number of bytes that may be requested from a device at one time. To refine the driver's performance, configure the request size to one of the following settings: 32, 64, 128, 256, 512, 1024 or 2000 bytes. The default setting is 512 bytes.

  Note: For Boolean arrays, the block size is the bit equivalent (or, block size multiplied by 8). For example, a block size of 512 bytes is equal to 512 * 8 = 4096 bits.

Protocol Parameters



For DF1 gateway applications, specify the node address of the destination device in question (e.g. DH+ or DH-485 node) .

For non-DF1 gateway applications, enter 0 (Default).

Destination Node Address (DST): 0

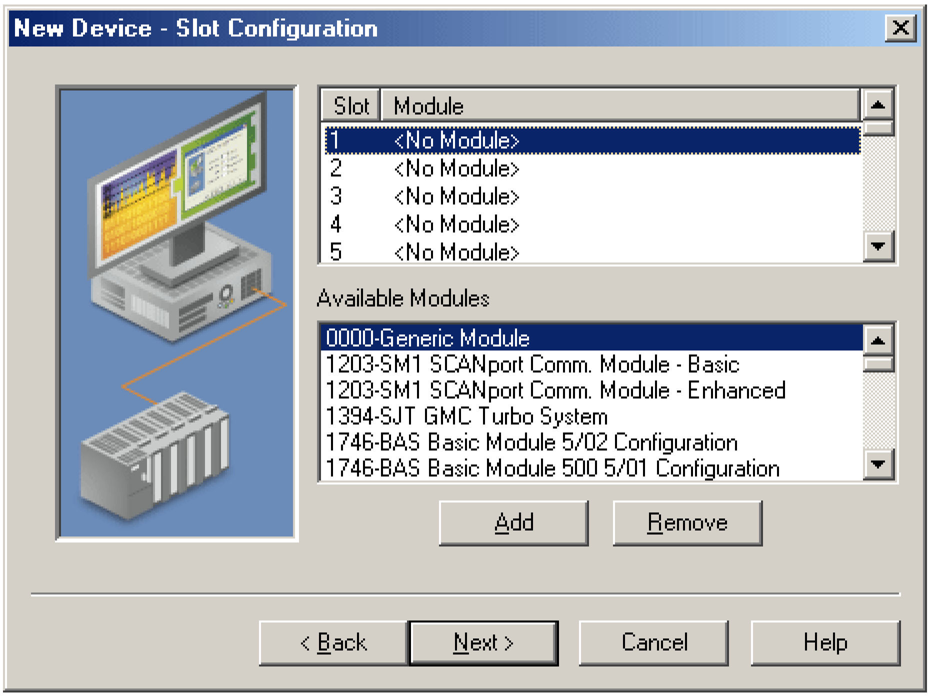
Descriptions of the parameters are as follows:

- **Destination Node Address (DST):** This parameter specifies the destination node address. For DF1 gateway applications, enter the node address of the destination device. For non-DF1 gateway applications, leave the node address at the default setting. The default setting is 0.

Note: The destination device is the DH+ or DH-485 device.

SLC 500 Slot Configuration

SLC 5/05 models (with modular I/O racks) must be configured for use with this driver if the I/O is to be accessed by the driver. Up to 30 slots can be configured per device.



New Device - Slot Configuration

Slot	Module
1	<No Module>
2	<No Module>
3	<No Module>
4	<No Module>
5	<No Module>

Available Modules

0000-Generic Module
1203-SM1 SCANport Comm. Module - Basic
1203-SM1 SCANport Comm. Module - Enhanced
1394-SJT GMC Turbo System
1746-BAS Basic Module 5/02 Configuration
1746-BAS Basic Module 500 5/01 Configuration

To use the slot configuration dialog, follow the instructions below:

1. Select the slot that will be configured by left-clicking on the row in the slot/module list box.
2. To select a module, left-click on it from the available modules list box.
3. Click **Add:** to include the selected module.
4. To delete a slot/module, select it and click **Remove**.
5. When finished, click **Next**.

Tip: Use the 0000-Generic Module to configure I/O that is not contained in the list of Available Modules.

Note: It is common to have open slots in the rack which do not contain a physical module. To correctly access data for the various slots that do contain a module, the preceding module(s) must have the correct number of words mapped. For example, if only interested in the I/O in slot 3, but slots 1 and 2 contain I/O modules, the correct modules must be selected for slots 1, 2, and 3 from this slot configuration tab.

Use of 0000-Generic Module

Use the Generic Module to map Input and Output words for modules that are not represented in the list of available modules. To correctly use the Generic Module users must know the number of Input and Output words required for each module. Consult Allen-Bradley I/O user manual documentation to confirm Input and Output requirements and be aware that requirements may be different based on Class 1 or Class 3 operation.

Note: For information on the number of input and output words available for each I/O module, refer to **Modular I/O Selection Guide**.

Modular I/O Selection Guide

The following table lists the number of input and output words available for each I/O module in the Slot Configuration list.

Tip: Use the Generic Module to map input and output words for modules that are not represented in the list of available modules. The range of accepted values is shown in the table below. Consult the Allen-Bradley user manual for the specific I/O module to configure to confirm input and output requirements. Requirements may be different based on Class 1 or Class 3 operation.

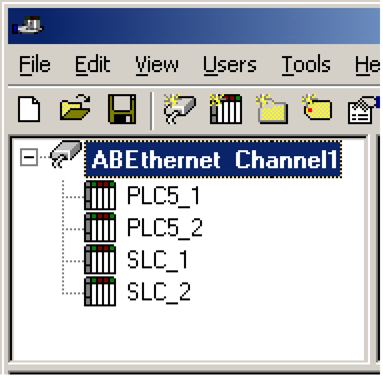
Module Type	Input Words	Output Words
0000-Generic Module	0-255	0-255
1203-SM1 SCANport Comm Module - Basic	8	8
1203-SM1 SCANport Comm Module - Enhanced	32	32
1394-SJT GMC Turbo System	32	32
1746-BAS Basic Module 500 5/01 Configuration	8	8
1746-BAS Basic Module 5/02 Configuration	8	8
1746-HS Single Axis Motion Controller	4	4
1746-HSCE High-Speed Counter/Encoder	8	1
1746-HSRV Motion Control Module	12	8
1746-HSTP1 Stepper Controller Module	8	8
1746-I*16 Any 16 pt Discrete Input Module	1	0
1746-I*32 Any 32 pt Discrete Input Module	2	0
1746-I*8 Any 8 pt Discrete Input Module	1	0
1746-IA16 16 Input 100/120 VAC	1	0
1746-IA4 4 Input 100/120 VAC	1	0
1746-IA8 8 Input 100/120 VAC	1	0
1746-IB16 16 Input (Sink) 24 VDC	1	0
1746-IB32 32 Input (Sink) 24 VDC	2	0
1746-IB8 8 Input (Sink) 24 VDC	1	0
1746-IC16 16 Input (Sink) 48 VDC	1	0
1746-IG16 16 Input [TTL] (Source) 5 VDC	1	0
1746-IH16 16 Input [Trans] (Sink) 125 VDC	1	0
1746-IM16 16 Input 200/240 VAC	1	0
1746-IM4 4 Input 200/240 VAC	1	0
1746-IM8 8 Input 200/240 VAC	1	0
1746-IN16 16 Input 24 VAC/VDC	1	0
1746-INI4I Analog 4 Ch. Isol. Current Input	8	8
1746-INI4VI Analog 4 Ch. Isol. Volt./Current Input	8	8
1746-INO4I Analog 4 Ch. Isol. Current Input	8	8
1746-INO4VI Analog 4 Ch. Isol. Volt./Current Input	8	8
1746-INT4 4 Ch. Isolated Thermocouple Input	8	8
1746-IO12 6 In 100/120 VAC 6 Out [Rly] VAC/VDC	1	1
1746-IO12DC 6 Input 12 VDC, 6 Output [Rly	1	1
1746-IO4 2 In 100/120 VAC 2 Out [Rly] VAC/VDC3	1	1
1746-IO8 4 In 100/120 VAC 4 Out [Rly] VAC/VDC4	1	1
1746-ITB16 16 Input [Fast] (Sink) 24 VDC	1	0
1746-ITV16 16 Input [Fast] (Source) 24 VDC	1	0
1746-IV16 16 Input (Source) 24 VDC	1	0
1746-IV32 32 Input (Source) 24 VDC	2	0
1746-IV8 8 Input (Source) 24 VDC	1	0
1746-NI4 4 Ch Analog Input	4	0
1746-NI8 8 Ch Analog Input, Class 1	8	8
1746-NI8 8 Ch Analog Input, Class 3	16	12
1746-NIO4I Analog Comb 2 in & 2 Current Out	2	2
1746-NIO4V Analog Comb 2 in & 2 Voltage Out	2	2
1746-NO4I 4 Ch Analog Current Output	0	4

Module Type	Input Words	Output Words
1746-NO4V 4 Ch Analog Voltage Output	0	4
1746-NR4 4 Ch Rtd/Resistance Input Module	8	8
1746-NT4 4 Ch Thermocouple Input Module	8	8
1746-NT8 Analog 8 Ch Thermocouple Input	8	8
1746-O*16 Any 16 pt Discrete Output Module	0	1
1746-O*32 Any 32 pt Discrete Output Module	0	2
1746-O*8 Any 8 pt Discrete Output Module	0	1
1746-OA16 16 Output (Triac) 100/240 VAC	0	1
1746-OA8 8 Output (Triac) 100/240 VAC	0	1
1746-OAP12 12 Output [Triac] 120/240 VDC	0	1
1746-OB16 16 Output [Trans] (Source) 10/50 VDC	0	1
1746-OB16E 16 Output [Trans] (Source) Protected	0	1
1746-OB32 32 Output [Trans] (Source) 10/50 VDC	0	2
1746-OB32E 32 Output [Trans] (Source) 10/50 VDC	0	2
1746-OB6EI 6 Output [Trans] (Source) 24 VDC	0	1
1746-OB8 8 Output [Trans] (Source) 10/50 VDC	0	1
1746-OBP16 16 Output [Trans 1 amp] (SRC) 24 VDC	0	1
1746-OBP8 8 Output [Trans 2 amp] (Source) 24 VDC	0	1
1746-OG16 16 Output [TLL] (SINK) 5 VDC	0	1
1746-OV16 16 Output [Trans] (Sink) 10/50 VDC	0	1
1746-OV32 32 Output [Trans] (Sink) 10/50 VDC	0	2
1746-OV8 8 Output [Trans] (Sink) 10/50 VDC	0	1
1746-OVP16 16 Output [Trans 1 amp] (Sink) 24VDC3	0	1
1746-OW16 16 Output [Relay] VAC/VDC	0	1
1746-OW4 4 Output [Relay] VAC/VDC	0	1
1746-OW8 8 Output [Relay] VAC/VDC	0	1
1746-OX8 8 Output [Isolated Relay] VAC/VDC	0	1
1747-DCM Direct Communication Module (1/2 Rack)	4	4
1747-DCM Direct Communication Module (1/4 Rack)	2	2
1747-DCM Direct Communication Module (3/4 Rack)	6	6
1747-DCM Direct Communication Module (Full Rack)	8	8
1747-DSN Distributed I/O Scanner 30 Blocks	32	32
1747-DSN Distributed I/O Scanner 7 Blocks	8	8
1747-KE Interface Module, Series A	1	0
1747-KE Interface Module, Series B	8	8
1747-MNET MNET Network Comm Module	0	0
1747-QS Synchronized Axes Module	32	32
1747-QV Open Loop Velocity Control	8	8
1747-RCIF Robot Control Interface Module	32	32
1747-SCNR ControlNet SLC Scanner	32	32
1747-SDN DeviceNet Scanner Module	32	32
1747-SN Remote I/O Scanner	32	32
AMCI-1561 AMCI Series 1561 Resolver Module	8	8

Optimizing Your Allen-Bradley Ethernet Communications

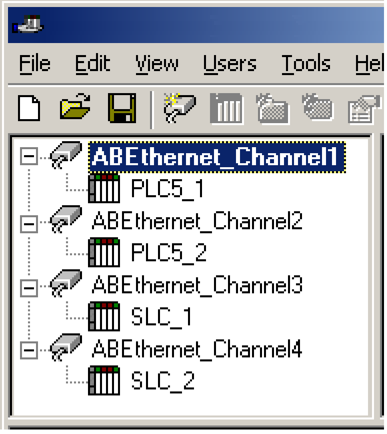
The Allen-Bradley Ethernet Driver has been designed to provide the best performance with the least amount of impact on the system's overall performance. While the Allen-Bradley Ethernet Driver is fast, there are a couple of guidelines that can be used in order to control and optimize the application and gain maximum performance.

Our server refers to communications protocols like Allen-Bradley Ethernet as a channel. Each channel defined in the application represents a separate path of execution in the server. Once a channel has been defined, a series of devices must then be defined under that channel. Each of these devices represents a single Allen-Bradley PLC from which data is collected. While this approach to defining the application provides a high level of performance, it doesn't take full advantage of the Allen-Bradley Ethernet Driver or the network. An example of how the application may appear when configured using a single channel is shown below.



Each device appears under a single Allen-Bradley Ethernet channel. In this configuration, the driver must move from one device to the next as quickly as possible in order to gather information at an effective rate. As more devices are added or more information is requested from a single device, the overall update rate begins to suffer.

If the Allen-Bradley Ethernet Driver could only define one single channel, then the example shown above would be the only option available; however, the Allen-Bradley Ethernet Driver can define up to 256 channels. Using multiple channels distributes the data collection workload by simultaneously issuing multiple requests to the network. An example of how the same application may appear when configured using multiple channels to improve performance is shown below.



Each device has now been defined under its own channel. In this new configuration, a single path of execution is dedicated to the task of gathering data from each device. If the application has 256 or fewer devices, it can be optimized exactly how it is shown here.

The performance can improve even if the application has more than 256 devices. While 256 or fewer devices may be ideal, the application still benefits from additional channels. Although spreading the device load across all channels causes the server to move from device to device again, it can do so with far fewer devices to process on a single channel.

Data Types Description

Data Type	Description
Boolean	Single bit
Byte	Unsigned 8-bit value
Char	Signed 8-bit value
Word	Unsigned 16-bit value
Short	Signed 16-bit value
DWord	Unsigned 32-bit value
Long	Signed 32-bit value
BCD	Two-byte packed BCD, four decimal digits
LBCD	Four-byte packed BCD, eight decimal digits
Float	32-bit IEEE floating-point
String	Null-terminated character array

Note: The DWord, Long, and LBCD data types are not native to any of the PLC models. When referencing a 16-bit location as a 32-bit value, the location referenced is the low word and the next successive location is the high word. For example, if N7:10 is selected as a DWord data type, N7:10 is the low word and N7:11 the high word.

Address Descriptions

Address specifications vary depending on the model in use. Select a link from the following list to obtain specific address information for the model of interest.

General Addressing
SLC 5/05 Open Addressing
PLC-5 Family and Soft PLC Addressing

General Addressing

The general addresses below pertain to SLC 5/05, PLC-5, and SoftPLC.

Output Files
Input Files
Status Files
Binary Files
Timer Files
Counter Files
Control Files
Integer Files
Float Files
ASCII Files
String Files

See Also:
SLC 5/05 Open Addressing
PLC-5 Family and SoftPLC Addressing

Output Files

The syntax for accessing data in the output file differs depending on the PLC model. Data locations are read / write for PLC-5 and SoftPLC models and read only for all other models. The default data type for all syntax is shown in **bold**.

PLC-5 and SoftPLC Model Syntax

Syntax	Data Type
O:<word>	Short, **Word**, BCD
O:<word>/<bit>	**Boolean**
O:<word>/<bit>[rows][cols]	**Boolean***
O:<word>/<bit>[cols]	**Boolean***
O/bit	**Boolean**
O/bit[rows][cols]	**Boolean***
O/bit[cols]	**Boolean***

*Array types

Note: Word and bit address information is in octal for PLC-5 and SoftPLC models. This follows the convention of the programming software.

SLC 5/05 Open Models (Modular I/O) Syntax

Syntax	Data Type
O:<slot>	Short, **Word**, BCD
O:<slot>.<word>	Short, **Word**, BCD
O:<slot>/<bit>	**Boolean**
O:<slot>/<bit>[rows][cols]	**Boolean***
O:<slot>/<bit>[cols]	**Boolean***
O:<slot>.<word>/<bit>	**Boolean**
O:<slot>.<word>/<bit>[rows][cols]	**Boolean***
O:<slot>.<word>/<bit>[cols]	**Boolean***

*Array types

Slot and Word Configurations

The following slot and word locations are allowed for each model. For information, refer to **Device Setup**.

PLC Model	Min Slot	Max. Slot	Max. Word
SLC 5/05 Open	1	30	*
PLC-5 Family	NA	NA	277 (octal)
SoftPLC	NA	NA	777 (octal)

*The number of input or output words available for each I/O module can be found in the **Modular I/O Selection Guide**.

Examples

All addresses are in octal.

PLC-5 / SoftPLC	Addresses
O:0	Word 0
O:37	Word 31 (37 octal=31 decimal)
O/42	Bit 34 (42 octal=34 decimal)
O:2/2	Bit 2 word 2 (same as O/42)
O/20[9]	9 element Boolean array starting at bit 16 (20 octal=16 decimal)
O/37[8][11]	8 by 11 element Boolean array starting at bit 31 (37 octal=31 decimal)
O:47/5[3]	3 element Boolean array starting at bit 5 word 39 (47 octal=39 decimal)
O:11/13[3][7]	3 by 7 element Boolean array starting at bit 11 (13 octal=11 decimal) word 9 (11 octal=9 decimal)

SLC 5/05	Addresses
O:1	Word 0 slot 1
O:1.0	Word 0 slot 1 (same as O:1)
O:12	Word 0 slot 12
O:12.2	Word 2 slot 12
O:4.0/0	Bit 0 word 0 slot 4
O:4/0	Bit 0 slot 4 (same as O:4.0/0)
O:4.2/0	Bit 0 word 2 slot 4
O:4/32	Bit 32 slot 4 (same as O:4.2/0)
O:2.12/3[17]	17 element Boolean array starting at bit 3 word 12 slot 2
O:2.2/0[12][12]	12 by 12 element Boolean array starting at bit 0 word 2 slot 2
O:2/43[5]	5 element Boolean array starting at bit 43 slot 2
O:2/11[6][12]	6 by 12 element Boolean array starting at bit 11 slot 2

Input Files

The syntax for accessing data in the input file differs depending on the PLC model. Data locations are read / write for PLC-5 models and read only for all other models. The default data type for all syntax is shown in **bold**.

PLC-5 and SoftPLC Model Syntax

Syntax	Data Type
I:<word>	Short, **Word**, BCD
I:<word>/<bit>	**Boolean**
I:<word>/<bit>[rows][cols]	**Boolean***
I:<word>/<bit>[cols]	**Boolean***
I/bit	**Boolean**
I/bit[rows][cols]	**Boolean***
I/bit[cols]	**Boolean***

*Array types

Note: Word and bit address information is in octal for PLC-5 and SoftPLC models. This follows the convention of the programming software.

SLC 5/05 Open Models (Modular I/O) Syntax

Syntax	Data Type
I:<slot>	Short, **Word**, BCD
I:<slot>.<word>	Short, **Word**, BCD
I:<slot>/<bit>	**Boolean**
I:<slot>/<bit>[rows][cols]	**Boolean***
I:<slot>/<bit>[cols]	**Boolean***
I:<slot>.<word>/<bit>	**Boolean**
I:<slot>.<word>/<bit>[rows][cols]	**Boolean***
I:<slot>.<word>/<bit>[cols]	**Boolean***

*Array types

Slot and Word Locations

The following slot and word locations are allowed for each model. For more information, refer to **Device Setup**.

PLC Model	Min Slot	Max. Slot	Max. Word
SLC 5/05 Open	1	30	*
PLC-5 Family	NA	NA	277 (octal)
SoftPLC Family	NA	NA	777 (octal)

*The number of input or output words available for each I/O module can be found in the **Modular I/O Selection Guide**.

Examples

All addresses are in octal.

PLC-5 / SoftPLC	Addresses
I:0	Word 0
I:10	Word 8 (10 octal = 8 decimal)
I/20	Bit 16 (20 octal = 16 decimal)
I:1/0	Bit 0 word 1 (same as I/20)
I/20[9]	9 element Boolean array starting at bit 16 (20 octal = 16 decimal)
I/37[8][11]	8 by 11 element Boolean array starting at bit 31 (37 octal = 31 decimal)
I:47/5[3]	3 element Boolean array starting at bit 5 word 39 (47 octal = 39 decimal)
I:11/13[3][7]	3 by 7 element Boolean array starting at bit 11 (13 octal = 11 decimal) word 9 (11 octal = 9 decimal)

SLC 5/05	Addresses
I:1	Word 0 slot 1
I:1.0	Word 0 slot 1 (same as I:1)
I:12	Word 0 slot 12
I:12.2	Word 2 slot 12
I:4.0/0	Bit 0 word 0 slot 4
I:4/0	Bit 0 slot 4 (same as I:4.0/0)
I:4.2/0	Bit 0 word 2 slot 4
I:4/32	Bit 32 slot 4 (same as I:4.2/0)
I:2.12/3[17]	17 element Boolean array starting at bit 3 word 12 slot 2
I:2.2/0[12][12]	12 by 12 element Boolean array starting at bit 0 word 2 slot 2
I:2/43[5]	5 element Boolean array starting at bit 43 slot 2
I:2/11[6][12]	6 by 12 element Boolean array starting at bit 11 slot 2

Status Files

To access Status files, specify a word (and optionally, a bit in the word). The default data type for all syntax is shown in **bold**.

Syntax	Data Type
S:<word>	Short, **Word**, BCD, DWord, Long, LBCD
S:<word> [rows][cols]	Short, **Word**, BCD, DWord, Long, LBCD*
S:<word> [cols]	Short, **Word**, BCD, DWord, Long, LBCD*
S:<word>/<bit>	**Boolean**
S:<word>/<bit> [rows][cols]	**Boolean***
S:<word>/<bit> [cols]	**Boolean***
S/bit	**Boolean**
S/bit [rows][cols]	**Boolean***
S/bit [cols]	**Boolean***

*Array types

Note: The number of array elements (in bytes) cannot exceed the block request size specified. This means that array size cannot exceed 16 words given a block request size of 32 bytes. For more information, refer to **Block Request Size**.

Word Locations

The following Word locations are allowed for each model. The maximum word location is one less when accessing as a 32-bit data type (Long, DWord, or Long BCD).

PLC Model	Max. Word
SLC 5/05 Open	999
PLC-5 Family	999
SoftPLC	31

Example	Description
S:0	Word 0.
S/26	Bit 26.
S:4/15	Bit 15 word 4.
S:10 [16]	16 element array starting at word 10.
S:0 [4][8]	4 by 8 element array starting at word 0.
S/9 [5]	5 element Boolean array starting at bit 9.
S/11 [3][7]	3 by 7 element Boolean array starting at bit 11.
S:6/1 [6]	6 element Boolean array starting at bit 1 word 6.
S:13/5 [2][3]	2 by 3 element Boolean array starting at bit 5 word 13.

Binary Files

To access Binary files, specify a file number and a word (and optionally, a bit in the word). The default data type for all syntax is shown in **bold**.

Syntax	Data Type
B<file>:<word>	Short, **Word**, BCD, DWord, Long, LBCD
B<file>:<word> [rows][cols]	Short, **Word**, BCD, DWord, Long, LBCD*
B<file>:<word> [cols]	Short, **Word**, BCD, DWord, Long, LBCD*
B<file>:<word>/<bit>	**Boolean**
B<file>:<word>/<bit> [rows][cols]	**Boolean***
B<file>:<word>/<bit> [cols]	**Boolean***
B<file>/bit	**Boolean**
B<file>/bit [rows][cols]	**Boolean***
B<file>/bit [cols]	**Boolean***

*Array types

Note: The number of array elements (in bytes) cannot exceed the block request size specified. This means that array size cannot exceed 16 words given a block request size of 32 bytes. For more information, refer to **Block Request Size**.

File Numbers and Word Locations

The following file numbers and word locations are allowed for each model. The maximum word location is one less when accessing as a 32-bit data type (Long, DWord, or Long BCD).

PLC Model	File Number	Max. Word
SLC 5/05 Open	3, 9-999	999
PLC-5 Family	3-999	1999
SoftPLC	3-9999	9999

Example	Description
B3:0	Word 0
B3/26	Bit 26
B12:4/15	Bit 15 word 4
B3:10 [20]	20 element array starting at word 10
B15:0 [6][6]	6 by 6 element array starting at word 0
B3/7 [8]	8 element Boolean array starting at bit 7
B3/32 [6][9]	6 by 9 element Boolean array starting at bit 32
B3:11/2 [12]	12 element Boolean array starting at bit 2 word 11
B3:23/4 [5][8]	5 by 8 element Boolean array starting at bit 4 word 23

Timer Files

Timer files are a structured type whose data is accessed by specifying a file number, an element and a field. The default data type depends on the field being accessed. Integer fields receive a default data type of Word.

Syntax	Data Type
T<file>:<element>.<field>	Depends on field

File Numbers and Elements

The following file numbers and maximum element are allowed for each model.

PLC Model	File Number	Max. Element
SLC 5/05 Open	4, 9-999	999
PLC-5 Family	3-999	1999
SoftPLC	3-9999	9999

The following fields are allowed for each element. Refer to the PLC documentation for the meaning of each field.

Element Field	Data Type	Access
ACC	Short, **Word**	Read/Write
PRE	Short, **Word**	Read/Write
DN	**Boolean**	Read Only
TT	**Boolean**	Read Only
EN	**Boolean**	Read Only

Example	Description
T4:0.ACC	Accumulator of timer 0 file 4.
T4:10.DN	Done bit of timer 10 file 4.
T15:0.PRE	Preset of timer 0 file 15.

Counter Files

Counter files are a structured type whose data is accessed by specifying a file number, an element and a field. The default data type depends on the field being accessed. Integer fields receive a default data type of Word.

Syntax	Data Type
C<file>:<element>.<field>	Depends on field

File Numbers and Elements

The following file numbers and maximum element are allowed for each model.

PLC Model	File Number	Max. Element
SLC 5/05 Open	5, 9-999	999
PLC-5 Family	3-999	1999
SoftPLC	3-9999	9999

The following fields are allowed for each element. Refer to the PLC documentation for the meaning of each field.

Element Field	Data Type	Access
ACC	Short, **Word**	Read/Write
PRE	Short, **Word**	Read/Write
UA	**Boolean**	Read Only
UN	**Boolean**	Read Only
OV	**Boolean**	Read Only
DN	**Boolean**	Read Only
CD	**Boolean**	Read Only
CU	**Boolean**	Read Only

Example	Description
C5:0.ACC	Accumulator of counter 0 file 5
C5:10.DN	Done bit of counter 10 file 5
C15:0.PRE	Preset of counter 0 file 15

Control Files

Control files are a structured type whose data is accessed by specifying a file number, an element and a field. The default data type depends on the field being accessed. Integer fields receive a default data type of Word.

Syntax	Data Type
R<file>:<element>.<field>	Depends on field

File Numbers and Elements

The following file numbers and maximum element are allowed for each model.

PLC Model	File Number	Max. Element
SLC 5/05 Open	6, 9-999	999
PLC-5 Family	3-999	1999
SoftPLC	3-9999	9999

The following fields are allowed for each element. Refer to the PLC documentation for the meaning of each field.

Element Field	Data Type	Access
LEN	Short, **Word**	Read/Write
POS	Short, **Word**	Read/Write
FD	**Boolean**	Read Only
IN	**Boolean**	Read Only
UL	**Boolean**	Read Only
ER	**Boolean**	Read Only
EM	**Boolean**	Read Only
DN	**Boolean**	Read Only
EU	**Boolean**	Read Only
EN	**Boolean**	Read Only

Examples	Description
R6:0.LEN	Length field of control 0 file 6
R6:10.DN	Done bit of control 10 file 6
R15:18.POS	Position field of control 18 file 15

Integer Files

To access Integer files, specify a file number and a word (and optionally, a bit in the word). The default data type for all syntax is shown in **bold**.

Syntax	Data Type
N<file>:<word>	Short, **Word**, BCD, DWord, Long, LBCD
N<file>:<word> [rows][cols]	Short, **Word**, BCD, DWord, Long, LBCD*
N<file>:<word> [cols]	Short, **Word**, BCD, DWord, Long, LBCD*
N<file>:<word>/<bit>	**Boolean**
N<file>:<word>/<bit> [rows][cols]	**Boolean***
N<file>:<word>/<bit> [cols]	**Boolean***
N<file>/bit	**Boolean**
N<file>/bit [rows][cols]	**Boolean***
N<file>/bit [cols]	**Boolean***

*Array types

Note: The number of array elements (in bytes) cannot exceed the block request size specified. This means that array size cannot exceed 16 words given a block request size of 32 bytes. For more information, refer to **Block Request Size**.

File Numbers and Word Locations

The following file numbers and maximum word locations are allowed for each model. The maximum word location is one less when accessing as a 32-bit data type (Long, DWord or Long BCD).

PLC Model	File Number	Max. Word
SLC 5/05 Open	7, 9-999	999
PLC-5 Family	3-999	1999
SoftPLC	3-9999	9999

Example	Description
N7:0	Word 0
N7/26	Bit 26
N12:4/15	Bit 15 word 4
N7:10 [8]	8 element array starting at word 10
N15:0 [4][5]	4 by 5 element array starting at word 0
N7/12 [9]	9 element Boolean array starting at bit 12
N7/19 [3][11]	3 by 11 element Boolean array starting at bit 19
N7:7/0 [10]	10 element Boolean array starting at bit 0 word 7
N7:29/13 [2][15]	2 by 15 element Boolean array starting at bit 13 word 29

Float Files

To access Float files, specify a file number and an element. The only data type allowed is Float.

Syntax	Data Type
F<file>:<element>	Float
F<file>:<element> [rows][cols]	Float array
F<file>:<element> [cols]	Float array

Note: The number of array elements (in bytes) cannot exceed the block request size specified. This means array size cannot exceed 8 floats given a block request size of 32 bytes. For more information, refer to **Block Request Size**.

File Numbers and Word Locations

The following file numbers and maximum word locations are allowed for each model.

PLC Model	File Number	Max. Word
SLC 5/05 Open	8-999	999
PLC-5 Family	3-999	1999
SoftPLC	3-9999	9999

Example	Description
F8:0	Float 0
F8:10 [16]	16-element array starting at word 10
F15:0 [4][4]	16-element array starting at word 0

ASCII Files

To access ASCII file data, specify a file number and character location. The default data type for all syntax is shown in **bold**.

Syntax	Data Type
A<file>:<char>	**Char**, Byte*
A<file>:<char> [rows][cols]	**Char**, Byte*
A<file>:<char> [cols]	**Char**, Byte*
A<file>:<word offset>/length	**String****

Note: The number of array elements cannot exceed the block request size specified. For more information, refer to **Block Request Size**.

*The PLC packs two characters per word in the file, with the high byte containing the first character and the low byte containing the second character. The PLC programming software allows access at the word level or two-character level. The AB Ethernet driver allows accessing to the character level. Examples are as follows:

- Using the programming software A10:0=AB would result in 'A' being stored in the high byte of A10:0 and 'B' being stored in the low byte.
- Using the AB Ethernet driver, two assignments, A10:0=A and A10:1=B, would result in the same data being stored in the PLC memory.

**Referencing this file as string data allows access to data at word boundaries like the programming software. The length can be up to 236 characters. If a string that is sent to the device is smaller in length than the length specified by the address, the driver null terminates the string before sending it down to the controller.

File Numbers and Character Locations

The following file numbers and maximum character locations are allowed for each model.

PLC Model	File Number	Max. Character
SLC 5/05 Open	9-999	1999
PLC-5 Family	3-999	1999
SoftPLC	N/A	N/A

Note: All SLC 500 PLCs do not support ASCII file types. For more information, refer to the PLC documentation.

Example	Description
A9:0	Character 0 (high byte of word 0)
A27:10 [80]	80-character array starting at character 10
A15:0 [4][16]	4 by 16 character array starting at character 0
A62:0/32	32-character string starting at word offset 0

String Files

To access data in a String file, specify a file number and an element. The only data type allowed is string, which are 82-character null-terminated arrays. The driver places the null terminator based on the string length returned by the PLC.

Syntax	Data Type
ST<file>:<element>	String

Note: Arrays of strings are not supported.

Tip: String length can be obtained with a COPY or MOVE function.

File Numbers and Word Locations

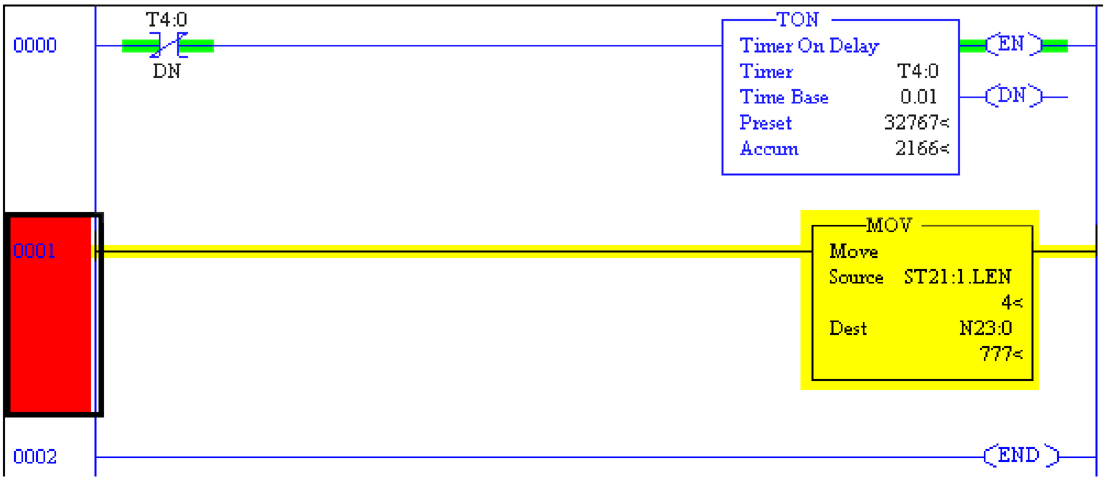
The following file numbers and maximum word locations are allowed for each model.

PLC Model	File Number	Max. Word
SLC 5/05 Open	9-999	999
PLC-5 Family	3-999	999
SoftPLC	3-9990	9999

Example	Description
ST9:0	String 0
ST18:10	String 10

String Length

While the .LEN field is not supported, string length can be obtained with a COPY or MOVE function, as shown below.



TON
Timer On Delay
Timer T4:0
Time Base 0.01
Preset 32767<
Accum 2166<

MOV
Move
Source ST21:1.LEN
4<
Dest N23:0
777<

SLC 5/05 Open Addressing

Open Addressing

The actual number of addresses available depends on the model of the PLC. The ranges have been opened up to allow for maximum flexibility with future models. If the driver finds at runtime that an address is not present in the device, it posts an error message and removes the tag from its scan list.

Note: This model has no specific addressing.

See Also: **General Addressing**

PLC-5 Family and SoftPLC Addressing

General Addressing

General Addressing

Model-Specific Addressing

BCD Files
PID Files
Message Files
Block Transfer Files

BCD Files

To access BCD files, specify a file number and a word. The only data types allowed are BCD and long BCD. The default data type is always BCD.

Syntax	Data Type
D<file>:<word>	**BCD**, LBCD
D<file>:<word> [rows][cols]	**BCD**, LBCD*
D<file>:<word> [cols]	**BCD**, LBCD*

*Array types

Note: The number of array elements (in bytes) cannot exceed the block request size specified. This means array size cannot exceed 16 BCDs given a block request size of 32 bytes. For more information, refer to **Block Request Size**.

File Numbers and Word Locations

The following file numbers and maximum word locations are allowed for each model.

PLC Model	File Number	Max. Word
SLC 5/05 Open	NA	NA
PLC-5 Family	3-999	1999
SoftPLC	3-9999	9999

Example	Description
D9:0	Word 0
D27:10 [16]	16 element array starting at word 10
D15:0 [4][8]	32 element array starting at word 0

PID Files

PID files are a structured type whose data is accessed by specifying a file number, an element and a field. The default data type depends on the field being accessed. Integer fields receive a default data type of Word.

Syntax	Data Type
PD<file>:<element>.<field>	Depends on field

File Numbers and Elements

The following file numbers and maximum element are allowed for each model.

PLC Model	File Number	Max. Element
SLC 5/05 Open	NA	NA
PLC-5 Family	3-999	999
SoftPLC	3-9999	9999

The following fields are allowed for each element. Refer to the PLC documentation for the meaning of each field.

Element Field	Data Type	Access
SP	**Real**	Read/Write
KP	**Real**	Read/Write
KI	**Real**	Read/Write
KD	**Real**	Read/Write
BIAS	**Real**	Read/Write
MAXS	**Real**	Read/Write
MINS	**Real**	Read/Write
DB	**Real**	Read/Write
SO	**Real**	Read/Write
MAXO	**Real**	Read/Write
MINO	**Real**	Read/Write

Element Field	Data Type	Access
UPD	**Real**	Read/Write
PV	**Real**	Read/Write
ERR	**Real**	Read/Write
OUT	**Real**	Read/Write
PVH	**Real**	Read/Write
PVL	**Real**	Read/Write
DVP	**Real**	Read/Write
DVN	**Real**	Read/Write
PVDB	**Real**	Read/Write
DVDB	**Real**	Read/Write
MAXI	**Real**	Read/Write
MINI	**Real**	Read/Write
TIE	**Real**	Read/Write
FILE	Short, **Word**	Read/Write
ELEM	Short, **Word**	Read/Write
EN	**Boolean**	Read/Write
CT	**Boolean**	Read/Write
CL	**Boolean**	Read/Write
PVT	**Boolean**	Read/Write
DO	**Boolean**	Read/Write
SWM	**Boolean**	Read/Write
CA	**Boolean**	Read/Write
MO	**Boolean**	Read/Write
PE	**Boolean**	Read/Write
INI	**Boolean**	Read/Write
SPOR	**Boolean**	Read/Write
OLL	**Boolean**	Read/Write
OLH	**Boolean**	Read/Write
EWD	**Boolean**	Read/Write
DVNA	**Boolean**	Read/Write
DVHA	**Boolean**	Read/Write
PVLA	**Boolean**	Read/Write
PVHA	**Boolean**	Read/Write

Example	Description
PD14:0.SP	Set point field of PD 0 file 14
PD18:6.EN	Status enable bit of PD 6 file 18

Message Files

Message files are a structured type whose data is accessed by specifying a file number, an element and a field. The default data type depends on the field being accessed. Integer fields receive a default data type of Word.

Syntax	Data Type
MG<file>:<element>.<field>	Depends on field

File Numbers and Elements

The following file numbers and maximum element are allowed for each model.

PLC Model	File Number	Max. Element
SLC 5/05 Open	NA	NA
PLC-5 Family	3-999	999
SoftPLC	3-9999	9999

The following fields are allowed for each element. Refer to the PLC documentation for the meaning of each field.

Element Field	Data Type	Access
ERR	Short, **Word**	Read/Write
RLEN	Short, **Word**	Read/Write
DLEN	Short, **Word**	Read/Write
EN	**Boolean**	Read/Write
ST	**Boolean**	Read/Write
DN	**Boolean**	Read/Write
ER	**Boolean**	Read/Write
CO	**Boolean**	Read/Write
EW	**Boolean**	Read/Write
NR	**Boolean**	Read/Write
TO	**Boolean**	Read/Write

Example	Description
MG14:0.RLEN	Requested length field of MG 0 file 14
MG18:6.CO	Continue bit of MG 6 file 18

Block Transfer Files

Block transfer files are a structured type whose data is accessed by specifying a file number, an element, and a field. The default data type depends on the field being accessed. Integer fields receive a default data type of Word.

Syntax	Data Type
BT<file>:<element>.<field>	Depends on field

File Numbers and Elements

The following file numbers and maximum element are allowed for each model.

PLC Model	File Number	Max. Element
SLC 5/05 Open	NA	NA
PLC-5 Family	3-999	1999
SoftPLC	3-9999	9999

The following fields are allowed for each element. Refer to the PLC documentation for the meaning of each field.

Element Field	Data Type	Access
RLEN	Short, **Word**	Read/Write
DLEN	Short, **Word**	Read/Write
FILE	Short, **Word**	Read/Write
ELEM	Short, **Word**	Read/Write
RW	**Boolean**	Read/Write
ST	**Boolean**	Read/Write
DN	**Boolean**	Read/Write
ER	**Boolean**	Read/Write
CO	**Boolean**	Read/Write
EW	**Boolean**	Read/Write
NR	**Boolean**	Read/Write
TO	**Boolean**	Read/Write

Example	Description
BT14:0.RLEN	Requested length field of BT 0 file 14
BT18:6.CO	Continue bit of BT 6 file 18

Error Descriptions

The following messages may be generated. Click on the link for a description of the message.

Address Validation

Missing address
Device address <address> contains a syntax error.
Address <address> is out of range for the specified device or register.
Device address <address> is not supported by model <model name>.
Data type <type> is not valid for device address <address>.
Device address <address> is read only.
Array size is out of range for address <address>.
Array support is not available for the specified address: <address>.

Communications Error Messages

Winsock initialization failed (OS Error = n).
Winsock V1.1 or higher must be installed to use the Allen-Bradley Ethernet device driver.

Device Status Messages

Device <device name> is not responding.
Unable to write to <address> on device <device name>.

Device-Specific Messages

Unable to read data starting at address <address> on device <device name>. Frame received contains errors.
Unable to read data starting at address <address> on device <device name>. [STS=<value>, EXT STS=<value>].
Unable to write to address <address> on device <device name>. Frame received contains errors.
Unable to write to address <address> on device <device name>. [STS=<value>, EXT STS=<value>].

Address <address> is out of range for the specified device or register.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically references a location that is beyond the range of supported locations for the device.

Solution:
Verify the address is correct; if it is not, re-enter it in the client application.

Array size is out of range for address <address>.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically is requesting an array size that is too large for the address type or block size of the driver.

Solution:
Re-enter the address in the client application to specify a smaller value for the array or a different starting point.

Array support is not available for the specified address: <address>.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically contains an array reference for an address type that doesn't support arrays.

Solution:
Re-enter the address in the client application to remove the array reference or correct the address type.

Data type <type> is not valid for device address <address>.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically has been assigned an invalid data type.

Solution:
Modify the requested data type in the client application.

Device address <address> contains a syntax error.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically contains one or more invalid characters.

Solution:
Re-enter the address in the client application.

Device address <address> is not supported by model <model name>.

Error Type:
Warning

Possible Cause:

A tag address that has been specified statically references a location that is valid for the communications protocol but not supported by the target device.

Solution:
Verify the address is correct; if it is not, re-enter it in the client application. Also verify that the selected model name for the device is correct.

Device address <address> is read only.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically has a requested access mode that is not compatible with what the device supports for that address.

Solution:
Change the access mode in the client application.

Device <device name> is not responding.

Error Type:
Warning

Possible Cause:

1. The Ethernet connection between the device and the host PC is broken.
2. The communication parameters for the Ethernet connection are incorrect.
3. The named device may have been assigned an incorrect IP address.
4. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.

Solution:

1. Verify the cabling between the PC and the device.
2. Verify that the correct port has been specified for the named device.
3. Verify that the IP address given to the named device matches that of the actual device.
4. Increase the Request Timeout setting so that the entire response can be handled.

Missing address.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically has no length.

Solution:
Re-enter the address in the client application.

Unable to read data starting at address <address> on device <device name>. Frame received contains errors.

Error Type:
Warning

The type of error could be:

1. An incorrect frame size received.
2. A TNS mismatch.
3. An invalid response command returned from the device.

Possible Cause:

1. Misalignment of packets due to connection/disconnection between the PC and device.
2. Bad cabling connecting the devices that is causing noise.

Solution:
The driver recovers from the error without intervention. If this error occurs frequently, there may be an issue with the cabling or the device itself.

Unable to read data starting at address <address> on device <device name>. [STS=<value>, EXT STS=<value>].

Error Type:
Warning

Possible Cause:

1. The address requested does not exist in the PLC.
2. The address requested cannot be accessed because the PLC is in an error state.
3. The communications parameters for the Ethernet connection are incorrect.

Solution:

1. Verify the address exists in the PLC.
2. Verify the PLC is not in an error state.
3. Verify the communications parameters for the Ethernet connection are correct.
4. Verify the correct port is specified for the named device.
5. Verify the IP address given to the named device matches that of the actual device.

Note:
Check the status and extended status codes that are being returned by the PLC. The extended status code may not always be returned; thus, error information is contained within the status code. The codes are displayed in hexadecimal.

Status code errors in the low nibble of the status code indicate errors found by the local node. The driver continues to retry reading these blocks of data periodically. Errors found by the local node occur when the KF module cannot see the destination PLC on the network for some reason.

Status code errors in the high nibble of the status code indicate errors found by the PLC. These errors are generated when the block of data the driver is asking for is not available in the PLC. The driver does not ask for these blocks again after receiving this error. This error can be generated if the address does not exist in the PLC.

Unable to write to <address> on device <device name>.

Error Type:
Warning

Possible Cause:

1. The Ethernet connection between the device and the host PC is broken.
2. The communication parameters for the Ethernet connection are incorrect.
3. The named device may have been assigned an incorrect IP address.

Solution:

1. Verify the cabling between the PC and the device.
2. Verify that the correct port has been specified for the named device.
3. Verify that the IP address given to the named device matches that of the actual device.

Unable to write to address <address> on device <device name>. Frame received contains errors.

Error Type:
Warning

The Error Could Be:

1. Incorrect frame size received.
2. TNS mismatch.
3. Invalid response command returned from device.

Possible Cause:

1. Misalignment of packets due to connection/disconnection between PC and device.
2. There is bad cabling connecting the devices causing noise.

Solution:
The driver recovers from this error without intervention. If this error occurs frequently, there may be an issue with the cabling or the device itself.

Unable to write to address <address> on device <device name>. [STS=<value>, EXT STS=<value>].

Error Type:
Warning

Possible Cause:
The address written to does not exist in the PLC.

Solution:
Check the status and extended status codes that are being returned by the PLC. Note that an extended status code may not always be returned and thus the error information is contained within the status code. The codes are displayed in hexadecimal.

Status code errors in the low nibble of the status code indicate errors found by the local node. Errors found by the local node occur when the KF module cannot see the destination PLC on the network for some reason.

Status code errors in the high nibble of the status code indicate errors found by the PLC. These errors are generated when the data location is not available in the PLC or is not writable.

Winsock initialization failed (OS error = n).

Error Type:
Fatal

OS Error:	Indication	Possible Solution
10091	Indicates that the underlying network subsystem is	Wait a few seconds and restart the

OS Error:	Indication	Possible Solution
	not ready for network communication.	driver.
10067	Limit on the number of tasks supported by the Windows Sockets implementation has been reached.	Close one or more applications that may be using Winsock and restart the driver.

Winsock V1.1 or higher must be installed to use the Allen-Bradley Ethernet device driver.

Error Type:

Fatal

Possible Cause:

The version number of the Winsock DLL found on the system is less than 1.1.

Solution:

Upgrade Winsock to version 1.1 or higher.

Index