Allen-Bradley DF1 Driver Help

Table of Contents

Allen-Bradley DF1 Driver Help

Help version 1.021

CONTENTS

Overview

The Allen-Bradley DF1 Driver provides an easy and reliable way to connect Allen-Bradley DF1 devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. This driver supports Allen-Bradley Micrologix, SLC500 and PLC5 series PLCs.

Channel Setup

Supported Link Protocols

Allen-Bradley DF1 Full-Duplex (point-to-point communication)
Allen-Bradley DF1 Half-Duplex Master (multi-drop communication) also known as Allen-Bradley DF1 Polled-Mode.*
Allen-Bradley DF1 Radio Modem (point-to-point and multi-drop communication).**

*Slave-to-slave communication is not supported.
**Store and forward feature is not supported.

Note: For required firmware versions for Allen-Bradley DF1 Radio Modem support, refer to **Device Setup**.

Supported Communication Parameters*

Baud Rate: 300, 600, 1200, 2400, 9600, 19200
Parity: None, Even, or Odd
Data Bits: 5, 6, 7 or 8
Stop Bits: 1 or 2

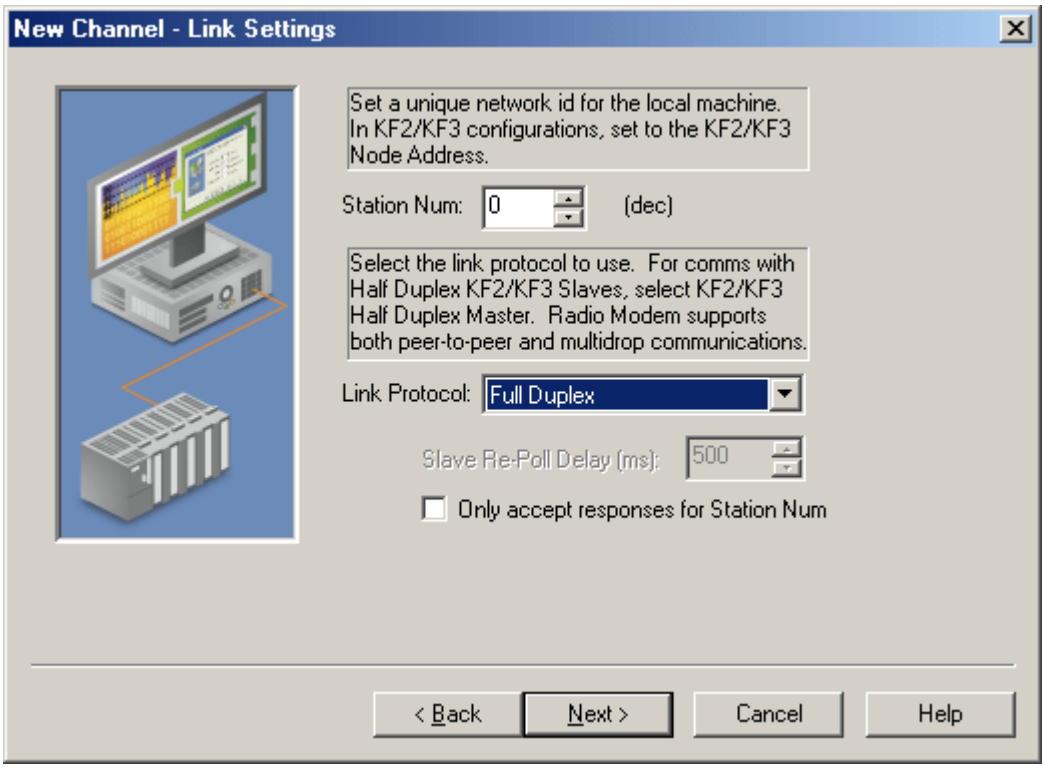
*Not all devices may support all listed configurations.

Note: Error Checking should be set to CRC or BCC within the device.

See Also: **Link Settings**

Link Settings

To access the Link Settings dialog after the Channel Wizard has closed, left-click on the channel and select **Properties**. Then, select the **Link Settings** tab.



Station Number

The Station Number should be set based on the device being communicated with (excluding radio modems). The following should be taken into account:

- If the destination device is on a DH+ or DH-485 network, communication must go through a Serial-to-DH+/DH-485 converter (i.e. KF2/KF3 module). In this case, the device being communicated with is the converter, not the destination device itself (which is a Micrologix, SLC500 or PLC-5). The station number for this configuration should be set to the converter's node address.
- If the destination device is not on a DH+ or DH-485 network, the device being communicated with is a Micrologix, SLC500 or PLC-5 PLC. The station number for this configuration can be set to an arbitrary unique address.

Standard Serial Configuration
Station Number = Arbitrary unique address on the network for the local PC. The range for DH-4850 is 1 to 63. Otherwise, the range is 0-255 otherwise.

DH+/DH-485 Converter Configuration
Station Number = Converter's node address (e.g. KF2/KF3 node address).

Link Protocol

The Allen-Bradley DF1 Driver implements the following link protocols:

Standard Serial Configuration
Full-Duplex (a.k.a Allen-Bradley DF1)
Half-Duplex Master (a.k.a Polled-Mode)
KF2/KF3 Half-Duplex Master
Radio Modem

DH+/DH-485 Converter Configuration
Full-Duplex (a.k.a Allen-Bradley DF1)
KF2/KF3 Half-Duplex Master

Slave Re-Poll Delay (ms)
For more information, refer to **Slave Re-Poll Delay**.

Only accept responses for Station Num
For more information, refer to **Full-Duplex**.

Full Duplex

Full-duplex protocol is used over a point-to-point link, allowing for high performance two-way communications between peers.

Only accept responses for Station Num:
When checked, this parameter limits the acceptance of responses to those that are destined for the station as indicated by the Station Num field.

Half Duplex Master

Half-Duplex Protocol is a multi-drop protocol with one master and one or more slaves. Generally, Half-Duplex provides lower data throughput than Full Duplex, but it adds the flexibility of being able to communicate with multiple devices from a single COM port. Half-Duplex is a master/slave protocol. In Half-Duplex Master mode, the driver is the master and all devices on the network are slaves. It is necessary that all the devices on the network be configured as Half-Duplex Slave since only one master is allowed on the network. For more information on configuring the Micrologix/ SLC500/PLC5 device using RSLogix, refer to the Rockwell documentation.

Note: If the destination device is on a DH-485 or DH+ network, communication must go through a KF2/KF3 module respectively. If the KF2/KF3 module is configured as a Half-Duplex Slave, the **KF2/KF3 Half-Duplex Master** Link Protocol must be chosen.

Master Responsibilities and Update Rates

The driver (master) is responsible for polling the slaves for data. In general, slaves would be polled in a round-robin manner. Due to the nature of OPC, how often a slave gets polled depends on the slave tags' update rate. In this manner, slaves are only polled when a Read/Write operation is requested from them. This reduces the traffic on the network and prevents unnecessary requests from taking place. In essence, the design of the client project (specifically update rates assigned) determines the traffic on the network. The faster the update rate, the more often a slave will be polled.

Messages, Sink and Source

There are three messages exchanged between master and slave in a Read/Write operation. The first is the master message requesting the slave to perform a Read/Write operation. The slave does not respond immediately with data as in full-duplex mode. The second message is a poll message from the master to the slave requesting the data gathered from the last master message operation. The third is the slave response with the data requested in the master message. Incoming requests to the slave are placed in what is termed a "sink". Once the slave performs the operation requested, it places the result in what is termed a "source".

Number of Attempts

The number of attempts for master messages and polls share the same number of attempts as configured in the device **Fail after xxx successive timeouts**. This attempt count is misleading in Half-Duplex mode since there are multiple messages sent from the master in a single data request. For all intents and purposes:

Let cnAttempts = xxx in Fail after xxx successive timeouts
attempts for master message timeout = cnAttempts
attempts for poll timeout = cnAttempts
attempts for request timeout = # attempts for master message timeout + # attempts for poll timeout == cnAttempts X 2

Source Empty and Slave Re-Poll Delay

The Allen-Bradley DF1 driver is optimized to send master messages and polls as quickly as possible to increase data throughput. The initial slave poll will not be delayed since a delay is unnecessary. If the slave needs time to process requests in its sink, it will be apparent in the initial poll response. It is at this time in which the driver introduces a delay and re-polls the slave. Such a delay allows the slave time to process the request before the next poll. This delay is set with the **Slave Re-Poll Delay** field. The slave will be re-polled cnAttempts times.

Sink and Source Full

Both sink and source are essentially buffers and buffers have limitations. More importantly, it is possible for the sink to fill up with requests. If this occurs, the slave will not acknowledge any master messages it receives. If after cnAttempts

the slave does not ACK the master, it is most likely the case that the slave sink is full. The driver will then poll the slave emptying any responses the slave may have, making room for responses of the requests that were in the full sink. This polling action will take place until the slave source is empty after cnAttempts. On the next slave request, it is likely that the sink will be empty. If it is not, it may mean the driver is polling the slave too quickly. If this is the case, increase the Slave Re-Poll Delay. Likewise, the slave source may also become full and the driver will again poll the slave until the source is empty after cnAttempts.

Accepted and Discarded Slave Reponses

In the above sections, it has been mentioned that the slave is polled until the emptied. This is possible if the slave source is full of queued up responses. On any given poll, only the response to the last master message is accepted, all others are discarded.

Note: Slave-to-slave communication is not supported.

KF2/KF3 Half Duplex Master

Half-Duplex Protocol is a multi-drop protocol with one master and one or more slaves. Generally, Half-Duplex provides lower data throughput than Full Duplex, but it adds the flexibility of being able to communicate with multiple KF2/KF3 modules from a single COM port. Half-Duplex is a master/slave protocol. In Half-Duplex Master mode, the driver is the master and all KF2/KF3 modules on the network are slaves. It is necessary that all the devices on the network be configured as Half-Duplex Slave since only one master is allowed on the network. For more information on configuring the KF2/KF3 module for Half-Duplex slave operation, refer to the Rockwell documentation.

Master Responsibilities and Update Rates

The driver (master) is responsible for polling the slaves for data. In general, slaves would be polled in a round-robin manner. Due to the nature of OPC, how often a slave gets polled depends on the update rate of the slave's tags. In this manner, slaves are only polled when a Read/Write operation is requested from them. This reduces the traffic on the network and prevents unnecessary requests from taking place. In essence, the design of the client project (specifically update rates assigned) determines the traffic on the network. The faster the update rate, the more often a slave will be polled.

Messages, Sink and Source

There are three messages exchanged between master and slave in a Read/Write operation. The first is the master message requesting the slave to perform a Read/Write operation. The slave does not respond immediately with data as in full-duplex mode. The second message is a poll message from the master to the slave requesting the data gathered from the last master message operation. The third is the slave response with the data requested in the master message. Incoming requests to the slave are placed in what is termed a "sink". Once the slave performs the operation requested, it places the result in what is termed a "source".

Number of Attempts

The number of attempts for master messages and polls share the same number of attempts as configured in the device **Fail after xxx successive timeouts**. This attempt count is misleading in Half-Duplex mode since there are multiple messages sent from the master in a single data request. For all intents and purposes:

Let cnAttempts = xxx in Fail after xxx successive timeouts
attempts for master message timeout = cnAttempts
attempts for poll timeout = cnAttempts
attempts for request timeout = # attempts for master message timeout + # attempts for poll timeout == cnAttempts X 2

Source Empty and Slave Re-Poll Delay

The Allen-Bradley DF1 driver is optimized to send master messages and polls as quickly as possible to increase data throughput. The initial slave poll will not be delayed since a delay is unnecessary. If the slave needs time to process requests in its sink, it will be apparent in the initial poll response. It is at this time in which the driver introduces a delay and re-polls the slave. Such a delay allows the slave time to process the request before the next poll. This delay is configured in Channel Properties (under the Protocol tab) and is called **Slave Re-Poll Delay**. The slave will be re-polled cnAttempts times.

Sink and Source Full

Both sink and source are essentially buffers and buffers have limitations. More importantly, it is possible for the sink to fill up with requests. If this occurs, the slave will not acknowledge any master messages it receives. If after cnAttempts the slave does not ACK the master, it is most likely the case that the slave sink is full. The driver will then poll the slave emptying any responses the slave may have, making room for responses of the requests that were in the full sink. This

polling action will take place until the slave source is empty after cnAttempts. On the next slave request, it is likely that the sink will be empty. If it is not, it may mean the driver is polling the slave too quickly. If this is the case, increase the Slave Re-Poll Delay. Likewise, the slave source may also become full and the driver will again poll the slave until the source is empty after cnAttempts.

Accepted and Discarded Slave Reponses

In the above sections, it has been mentioned that the slave is polled until the emptied. This is possible if the slave source is full of queued up responses. On any given poll, only the response to the last master message is accepted: all others are discarded.

Note: Slave-to-slave communication is not supported.

Radio Modem

The Radio Modem protocol is a command/reply protocol. There are no ACKs or NAKs during the request/response procedure. This reduces the number of bytes the radio modems have to transmit and receive in order to complete a transaction. This protocol supports full-duplex communications over a point-to-point link allowing for high performance two-way communications between peers. It also supports master/slave communications allowing for multi-drop configurations. Performance exceeds both Full-Duplex and Half-Duplex Protocols.

Device Setup

Supported Devices

Micrologix Series*
SLC500 Series*
PLC-5 Series (excluding the PLC-5/250 and PLC-5/VME series)

*Radio Modem link protocol requires the following firmware upgrades:

SLC 5/03, SLC 5/04 and SLC 5/05: Series C FRN6
MicroLogix 1200: Series C FRN7
MicroLogix 1500: Series C FRN8

Device ID

The Device ID is the Allen-Bradley DF1 network address of the PLC. For PLCs on a DH-485 or DH+ network, the range is 1-63. Otherwise, the range is 0-255.

For Full Duplex, the default address of 1 will work.
For Half-Duplex, the address must match the slave address.
For Radio Modem, the address must match the slave/peer address.

Ethernet Encapsulation

This driver supports Ethernet Encapsulation, which allows the driver to communicate with serial devices attached to an Ethernet network using a terminal server. Ethernet Encapsulation mode may be invoked through the COM ID dialog in Channel Properties. For more information, refer to the OPC Server's help documentation.

DH-485 and DH+ Support

An Allen Bradley KF3 or compatible device is needed to connect the driver to the DH-485 network. There are three options for communicating to a device on DH+ using the Allen-Bradley DF1 Device Driver.

- Allen Bradley KF2 or compatible device.
- DataLink DL Interface Cards (PCI/ISA/PC104). Card adds virtual serial ports for seamless configuration.
- DataLink DL4500 Ethernet-to-DH+ Converter. Configure the device for Ethernet Encapsulation. NIC is required.

See Also:

Cable Connections
Protocol Settings
Function File Options
SLC500 Slot Configuration

Modem Setup

This driver supports modem functionality. For more information, please refer to the topic "Modem Support" in the OPC Server Help documentation.

Protocol Settings

Error Checking Method

There are two methods of error checking available in the Allen-Bradley DF1 driver: Block Check Character (BCC) and 16 bit Cyclic Redundancy Check (CRC-16). Users must choose the checksum method expected by the device; otherwise, the device will not respond.

Request Size

This parameter is used to change the size of a data request. It can be important in refining the application's performance. If the application accesses large areas of PLC memory consecutively, then a large request size may be beneficial. If the data is spread throughout the PLC, then a small request size may be beneficial. The default setting is the large request size.

Swap PLC-5 Float Words?

PLC-5 Floats follow the IEEE 754 standard. They contain a sign bit S, an exponent E and a mantissa M. The 32 bit layout of this IEEE 754 Float is as shown below.

Upper Word Lower Word
SEEEEEEE EMMMMMMM MMMMMMMM MMMMMMMM
Byte 3 Byte 2 Byte 1 Byte 0

Allen-Bradley PLC-5 devices transfer binary Floating-point data on the serial link in the following order:
Upper Word Lower Word
Byte 2 Byte 3 Byte 0 Byte 1

This means the upper word is received first, followed by the lower word. Due to this ordering, a swap of the words is required, providing:
Lower Word Upper Word
Byte 0 Byte 1 Byte 2 Byte 3

The result passed on to the client is as follows:
Byte 3 Byte 2 Byte 1 Byte 0

Some PLC-5 emulated devices (such as the Avtron ADDvantage-32) already transfer binary Floating point data on the serial link with the lower word first.
Lower Word Upper Word
Byte 0 Byte 1 Byte 2 Byte 3

In this case, no swap is required. The result passed on to the client is as follows:
Byte 3 Byte 2 Byte 1 Byte 0

Thus, the rule of thumb is if the device transfers the lower word first, then the upper word in the packet on the serial link does not require word swapping. This only applies to PLC-5 emulated devices; that is, devices that use the Allen-Bradley DF1 protocol with PLC-5 commands. Allen-Bradley PLC-5 devices always transfer the upper word first followed by the lower word so the Float words must be swapped. This is the default setting.

Support Float Access to SLC/Micrologix N Files?

SLC and Micrologix users can select whether the driver will natively support Float access to Integer Files. The default setting is Yes.

Cable Connections

This driver is an Allen-Bradley DF1 RS232 serial driver. It does not handle the token ring passing of a DH-485 or DH+ network. When testing cable connections for this driver, first verify communications using the Allen Bradley programming software.

- For Micrologix and SLC 500 series PLCs, if the APS driver selected is KF3/KE, Full-Duplex, Half-Duplex Slave or Full Duplex (Micro) and communication is available with the PLC, this driver will also be able to communicate using the same cabling and communication settings.

- For PLC5 series PLCs, if the PLC-5 programming software driver selected is Serial to PLC or KE/KF and communication is available with the PLC, this driver will also be able to communicate using the same cabling and communication settings.

Micrologix Series

Use the same cable as when using the Allen Bradley APS software.

SLC 500 Series - Direct connection

If the PLC has an RS232 port it can be connected directly using a standard RS232 null modem cable (which is the same cable used in Allen Bradley APS software). The PLC port must be configured for Allen-Bradley DF1 communications, not DH-485 Master.

Note: This driver does not work in a direct connection to the DH-485 port of a SLC 500 series PLC using a 1747-PIC converter the way the APS software does.

PLC5 Series - Direct connection

A direct connection can be made to the CH0 port of enhanced PLC5 processors using a standard RS232 null modem cable. The port must be configured for Allen-Bradley DF1 communications.

DH-485 Networks

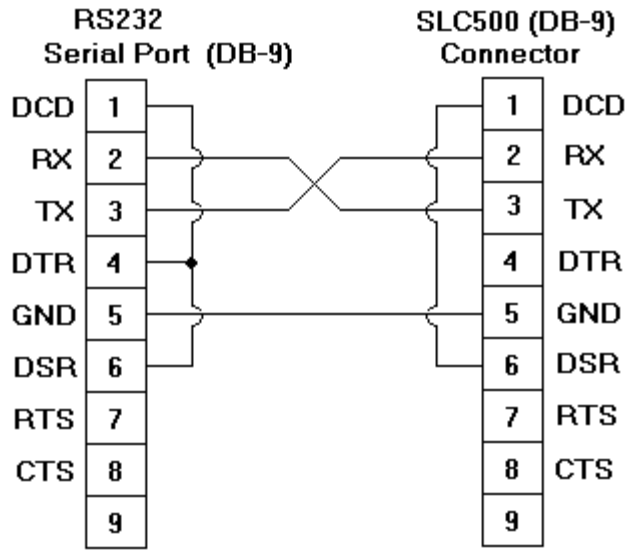
An Allen Bradley KF3 or compatible device is needed to connect the driver to the DH-485 network. A standard null modem cable is used to connect the PC to the KF3 device.

DH+ Networks

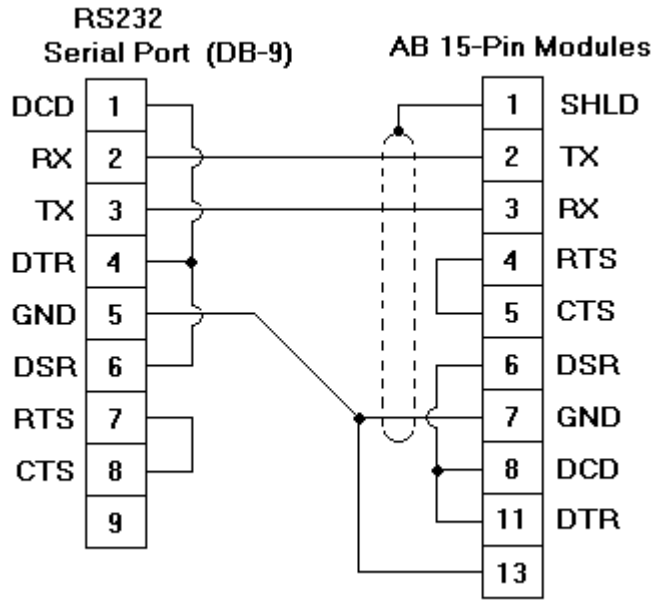
There are three options for communicating to a device on DH+ using the Allen-Bradley DF1 Device Driver:

- Allen Bradley KF2 or compatible device. A standard null modem cable is used to connect the PC to the KF2 device.
- DataLink DL Interface Cards (PCI/ISA/PC104). Consult AB documentation for DH+ wiring.
- DataLink DL4500 Ethernet-to-DH+ Converter. Consult DL4500 documentation for wiring.

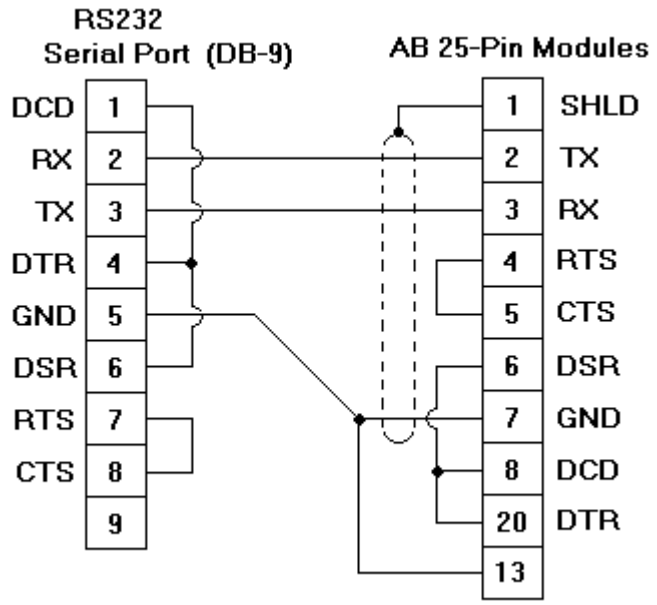
SLC500 Connection



15-Pin Module Connection



25-Pin Module Connection



Function File Options

Function files are structure-based files, similar to PD and MG data files, and are unique to the Micrologix 1200 and 1500. For more information on a specific function file are supported in the Allen-Bradley DF1 Device Driver, select a link from the list below.

High Speed Counter File (HSC)
Real-Time Clock File (RTC)
Channel 0 Communication Status File (CS0)

Channel 1 Communication Status File (CS1)
I/O Module Status File (IOS)

Function File Block Writes

For applicable function files, data can be written to the device in a single operation. By default, when data is written to a function file sub element (field within the function file structure), a write operation occurs immediately for that tag. For such files as the RTC file, whose sub elements include hour (HR), minute (MIN) and second (SEC), individual writes are not always acceptable. With such sub elements relying solely on time, values must be written in one operation to avoid time elapsing between sub elements writes. For this reason, there is the option to "block write" these sub elements.

Applicable Function Files/Sub Elements

RTC	
Year	YR
Month	MON
Day	DAY
Day of Week	DOW
Hour	HR
Minute	MIN
Second	SEC

How Block Writes Work

Block writing involves writing to the device the values of every Read/Write sub element in the function file in a single write operation. It is not necessary to write to every sub element prior to performing a block write. Sub elements not affected (written to) will have their current value written back to them. For example, if the current (last read) date and time is 1/1/2001, 12:00.00, DOW = 3, and the hour is changed to 1 o'clock, then the values written to the device would be 1/1/2001, 1:00.00, DOW = 3.

Instructions

1. Go to the Function File Options tab in the Device Properties dialog. Check the checkbox labeled **Block write supporting function files?**. This notifies the driver to utilize block writes on function files supporting block writes. The changes will be effective immediately after hitting the **OK** or **Apply** buttons.

2. Write the desired value to the sub element tag(s) in question. The sub element tag(s) will immediately take on the value(s) written to it.

   Note: After a sub element is written to at least once in block write mode, the tag's value does not originate from the controller, but instead from the driver's write cache. After the block write is done, all sub element tag values will originate from the controller.

3. Once the entire desired sub elements are written to, the block write that will send these values to the controller may be performed. To instantiate a block write, reference tag address **RTC:<element>._SET**. Setting this tag's value to "True" will cause a block write to occur based on the current (last read) sub elements and the sub elements affected (written to). The _SET tag is treated as a Write Only tag; meaning, a write to this tag is not reflected in subsequent reads on it. Setting this tag's value to "False" performs no action.

SLC500 Slot Configuration

SLC500 models with modular I/O racks need to be configured for use with the Allen-Bradley DF1 driver if the I/O is to be accessed by the driver. Up to 30 slots can be configured per device. 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 add the module.

4. To remove a module, select the slot in the slot/module list box and then click **Remove**.

Note: The module selections available are the same as those in the Allen Bradley APS software.

Knowledge of the number of input and output words in each slot is necessary for the driver to correctly address the I/O. Actually, only the numbers of input and output words in slots up to the slot of interest are needed to address I/O in that slot. For example, if accessing slot 3 only, slots 1 and 2 would have to be configured if they contain any I/O, slot 3, but not slot 4 or greater.

See Also: SLC 500 Modular I/O Selection Guide

SLC 500 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.

Module Type	Input Words	Output Words
1746-I*8 Any 8 pt Discrete Input Module	1	0
1746-I*16 Any 16 pt Discrete Input Module	1	0
1746-I*32 Any 32 pt Discrete Input Module	2	0
1746-O*8 Any 8 pt Discrete Output Module	0	1
1746-O*16 Any 16 pt Discrete Output Module	0	1
1746-O*32 Any 32 pt Discrete Output Module	0	2
1746-IA4 4 Input 100/120 VAC	1	0
1746-IA8 8 Input 100/120 VAC	1	0
1746-IA16 16 Input 100/120 VAC	1	0
1746-IB8 8 Input (Sink) 24 VDC	1	0
1746-IB16 16 Input (Sink) 24 VDC	1	0
1746-IB32 32 Input (Sink) 24 VDC	2	0
1746-IG16 16 Input [TTL] (Source) 5VDC	1	0
1746-IM4 4 Input 200/240 VAC	1	0
1746-IM8 8 Input 200/240 VAC	1	0
1746-IM16 16 Input 200/240 VAC	1	0
1746-IN16 16 Input 24 VAC/VDC	1	0
1746-ITB16 16 Input [Fast] (Sink) 24 VDC	1	0
1746-ITV16 16 Input [Fast] (Source) 24 VDC	1	0
1746-IV8 8 Input (Source) 24 VDC	1	0
1746-IV16 16 Input (Source) 24 VDC	1	0
1746-IV32 32 Input (Source) 24 VDC	2	0
1746-OA8 8 Output (Triac) 100/240 VAC	0	1
1746-OA16 16 Output (Triac) 100/240 VAC	0	1
1746-OB8 8 Output [Trans] (Source) 10/50 VDC	0	1
1746-OB16 16 Output [Trans] (Source) 10/50 VDC	0	1
1746-OB32 32 Output [Trans] (Source) 10/50 VDC	0	2
1746-OBP16 16 Output [Trans 1 amp] (SRC) 24 VDC	0	1
1746-OV8 8 Output [Trans] (Sink) 10/50 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-OW4 4 Output [Relay] VAC/VDC	0	1
1746-OW8 8 Output [Relay] VAC/VDC	0	1
1746-OW16 16 Output [Relay] VAC/VDC	0	1
1746-OX8 8 Output [Isolated Relay] VAC/VDC	0	1

1746-OVP 16 16 Output [Trans 1 amp] (Sink) 24VDC3	0	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-IO12 6 In 100/120 VAC 6 Out [Rly] VAC/VDC	1	1
1746-NI4 4 Ch Analog Input	4	0
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
1746-NO4V 4 Ch Analog Voltage Output	0	4
1746-NT4 4 Ch Thermocouple Input Module	8	8
1746-NR4 4 Ch Rtd/ Resistance Input Module	8	8
1746-HSCE High Speed Counter/Encoder	8	1
1746-HS Single Axis Motion Controller	4	4
1746-OG16 16 Output [TLL] (SINK) 5 VDC	0	1
1746-BAS Basic Module 500 5/01 Configuration	8	8
1746-BAS Basic Module 5/02 Configuration	8	8
1747-DCM Direct Communication Module (1/4 Rack)	2	2
1747-DCM Direct Communication Module (1/2 Rack)	4	4
1747-DCM Direct Communication Module (3/4Rack)	6	6
1747-DCM Direct Communication Module (Full Rack)	8	8
1747-SN Remote I/O Scanner	32	32
1747-DSN Distributed I/O Scanner 7 Blocks	8	8
1747-DSN Distributed I/O Scanner 30 Blocks	32	32
1747-KE Interface Module, Series A	1	0
1747-KE Interface Module, Series B	8	8
1746-NI8 8 Ch Analog Input, Class 1	8	8
1746-NI8 8 Ch Analog Input, Class 3	16	12
1746-IC16 16 Input (Sink) 48 VDC	1	0
1746-IH16 16 Input [Trans] (Sink) 125 VDC	1	0
1746-OAP12 12 Output [Triac] 120/240 VDC	0	1
1746-OB6EI 6 Output [Trans] (Source) 24 VDC	0	1
1746-OB16E 16 Output [Trans] (Source) Protected	0	1
1746-OB32E 32 Output [Trans] (Source) 10/50 VDC	0	2
1746-OBP8 8 Output [Trans 2 amp] (Source) 24 VDC	0	1
1746-IO12DC 6 Input 12 VDC, 6 Output [Rly	1	1
1746-INI4I Analog 4 Ch. Isol. Current Input	8	8
1746-INI4VI Analog 4 Ch. Isol. Volt./Current Input	8	8
1746-INT4 4 Ch. Isolated Thermocouple Input	8	8
1746-NT8 Analog 8 Ch Thermocouple Input	8	8
1746-HSRV Motion Control Module	12	8
1746-HSTP1 Stepper Controller Module	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

1394-SJT GMC Turbo System	32	32
1203-SM1 SCANport Comm Module - Basic	8	8
1203-SM1 SCANport Comm Module - Enhanced	32	32
AMCI-1561 AMCI Series 1561 Resolver Module	8	8

Data Types Description

Data Type	Description
Boolean	Single bit
Byte	Unsigned 8 bit value bit 0 is the low bit bit 7 is the high bit
Char	Signed 8 bit value bit 0 is the low bit bit 6 is the high bit bit 7 is the sign 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
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 will be the low word, and the next successive location will be the high word. For example, if N7:10 selected as a DWord data type, N7:10 would be 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.

Micrologix Addressing
SLC500 Addressing (Fixed I/O processor)
SLC5/01 Addressing
SLC5/02 Addressing
SLC5/03 Addressing
SLC5/04 Addressing
SLC5/05 Addressing
PLC5 Addressing

Micrologix Addressing

File Specific Addressing

Output Files
Input Files
Status Files
Binary Files
Timer Files
Counter Files
Control Files
Integer Files
Float Files
String Files
Long Files
PID Files
Message Files

Function Files

High Speed Counter File (HSC)
Real-Time Clock File (RTC)
Channel 0 Communication Status File (CS0)
Channel 1 Communication Status File (CS1)
I/O Module Status File (IOS)

SLC500 Addressing (Fixed I/O processor)

File Specific Addressing

Output Files
Input Files
Status Files
Binary Files
Timer Files
Counter Files
Control Files
Integer Files

SLC5/01 Addressing

File Specific Addressing

Output Files
Input Files
Status Files
Binary Files
Timer Files
Counter Files
Control Files
Integer Files

SLC5/02 Addressing

File Specific Addressing

Output Files
Input Files
Status Files

SLC5/03 Addressing

File Specific Addressing

SLC5/04 Addressing

File Specific Addressing

SLC5/05 Addressing

File Specific Addressing

PLC5 Addressing

File Specific Addressing

Output Files

The syntax for accessing data in an Output file differs depending on the PLC model. Arrays are not supported for output files. The default data types are shown in **bold**.

PLC-5 Syntax

Syntax	Data Type	Access
O:<word>	Short, **Word**, BCD	Read/Write
O:<word>/<bit>	**Boolean**	Read/Write
O/bit	**Boolean**	Read/Write

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

Micrologix Syntax

Syntax	Data Type	Access
O:<word>	Short, **Word**, BCD	Read/Write
O:<word>/<bit>	**Boolean**	Read/Write
O/bit	**Boolean**	Read/Write

Micrologix models have two types of I/O: embedded I/O and expansion I/O (not applicable for Micrologix 1000). Embedded I/O resides with the CPU base unit while Expansion I/O plugs into the CPU base unit. The table below lists the I/O capabilities of each Micrologix model.

Micrologix Model	Embedded I/O	Expansion I/O
1000	Slot 0	N/A
1100	Slot 0	Slots 1-4
1200	Slot 0	Slots 1-6
1400	Slot 0	Slots 1-7
1500	Slot 0	Slots 1-16

The address syntax for Micrologix I/O references a zero-based word offset, not a slot. Thus, calculations must be done to determine the word offset to a particular slot. This requires knowledge of the modules and their respective size in words. The table below specifies the size of some available modules; however, it is recommended that the Micrologix documentation and controller project be consulted in order to determine the true word size of a module. Instructions and examples in calculating word offset follow the table below.

Micrologix Embedded I/O Word Sizes

Micrologix Model	# Input Words	# Output Words
1000	2	1
1100	6	4
1200	4	4
1400	8	6
1500	4	4

Micrologix Expansion I/O Word Sizes

Modules	# Input Words	# Output Words
1769-HSC	35	34
1769-IA8I	1	0
1769-IA16	1	0
1769-IF4	6	0
1769-IF4XOF2	8	2
1769-IF8	12	1
1769-IM12	1	0
1769-IQ16	1	0
1769-IQ6XOW4	1	1
1769-IQ16F	1	0
1769-IQ32	2	0
1769-IR6	8	0
1769-IT6	8	0
1769-OA8	0	1
1769-OA16	0	1
1769-OB8	0	1
1769-OB16	0	1
1769-OB16P	0	1
1769-OB32	0	2
1769-OF2	2	2
1769-OF8C	11	9
1769-OF8V	11	9
1769-OV16	0	1
1769-OW8	0	1
1769-OW16	0	1
1769-OW8I	0	1
1769-SDN	66	2
1769-SM1	12	12
1769-SM2	7	7
1769-ASCII	108	108
1762-IA8	1	0
1762-IF2OF2	6	2
1762-IF4	7	0
1762-IQ8	1	0
1762-IQ8OW6	1	1
1762-IQ16	1	0
1762-OA8	0	1

1762-OB8	0	1
1762-OB16	0	1
1762-OW8	0	1
1762-OW16	0	1
1762-IT4	6	0
1762-IR4	6	0
1762-OF4	2	4
1762-OX6I	0	1

Calculation
Output Word Offset for Slot x = # Output Words in Slot 0 through Slot (x-1).

Note 1: The Embedded I/O needs to be taken into account when offsetting to Expansion I/O.

Note 2: The number of Input words does not factor into the calculation for Output Word Offset.

I/O Example
Let
Slot 0 = Micrologix 1500 LRP Series C = 4 Output Words
Slot 1 = 1769-OF2 = 2 Output Words
Slot 2 = 1769-OW8 = 1 Output Word
Slot 3 = 1769-IA16 = 0 Output Word
Slot 4 = 1769-OF8V = 9 Output Word
Bit 5 of Slot 4 = 4 + 2 + 1 = 7 words = O:7/5

SLC 500 Syntax
The default data type is shown in **bold**.

Syntax	Data Type	Access
O:<slot>	Short, **Word**, BCD	Read Only
O:<slot>.<word>	Short, **Word**, BCD	Read Only
O:<slot>/<bit>	**Boolean**	Read Only
O:<slot>.<word>/<bit>	**Boolean**	Read Only

Ranges
PLC Model	Min Slot	Max Slot	Max Word
Micrologix	NA	NA	2047
SLC 500 Open	NA	NA	1
SLC 5/01	1	30	*
SLC 5/02	1	30	*
SLC 5/03	1	30	*
SLC 5/04	1	30	*
SLC 5/05	1	30	*
PLC-5 Family	NA	NA	277**

*The number of Input or Output words available for each I/O module can be found in the **Modular I/O Selection Guide**. For slot configuration help, refer to **Device Setup**.
**Octal.

Examples
Micrologix	Address
O:0	Word 0
O/2	Bit 2

O:0/5	Bit 5

SLC 500 Fixed I/O	Address
O:0	Word 0
O:1	Word 1
O/16	Bit 16
O:1/0	Bit 0 word 1 (same as O/16)

PLC5	Address*
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)

*Addresses are shown in Octal.

SLC 500 Modular I/O	Address
O:1	Word 0 slot 1
O:1.0	Word 0 slot 1 (same as O:1)
O:12	Wword 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)

Input Files

The syntax for accessing data in an Input file differs depending on the PLC model. Arrays are not supported for Input files. The default data types are shown in **bold**.

PLC-5 Syntax

Syntax	Data Type	Access
I:<word>	Short, **Word**, BCD	Read/Write
I:<word>/<bit>	**Boolean**	Read/Write
I/bit	**Boolean**	Read/Write

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

Micrologix Syntax

Syntax	Data Type	Access
I:<word>	Short, **Word**, BCD	Read/Write
I:<word>/<bit>	**Boolean**	Read/Write
I/bit	**Boolean**	Read/Write

Micrologix models have two types of I/O: embedded I/O and expansion I/O (not applicable for Micrologix 1000). Embedded I/O resides with the CPU base unit while Expansion I/O plugs into the CPU base unit. The table below lists the I/O capabilities of each Micrologix model.

Micrologix Model	Embedded I/O	Expansion I/O
1000	Slot 0	N/A

1100	Slot 0	Slots 1-4
1200	Slot 0	Slots 1-6
1400	Slot 0	Slots 1-7
1500	Slot 0	Slots 1-16

The address syntax for Micrologix I/O references a zero-based word offset, not a slot. Thus, calculations must be done to determine the word offset to a particular slot. This requires knowledge of the modules and their respective size in words. The table below specifies the size of some available modules; however, it is recommended that the Micrologix documentation and controller project be consulted in order to determine the true word size of a module. Instructions and examples in calculating word offset follow the table below.

Micrologix Embedded I/O Word Sizes

Micrologix Model	# Input Words	# Output Words
1000	2	1
1100	6	4
1200	4	4
1400	8	6
1500	4	4

Micrologix Expansion I/O Word Sizes

Modules	# Input Words	# Output Words
1769-HSC	35	34
1769-IA8I	1	0
1769-IA16	1	0
1769-IF4	6	0
1769-IF4XOF2	8	2
1769-IF8	12	1
1769-IM12	1	0
1769-IQ16	1	0
1769-IQ6XOW4	1	1
1769-IQ16F	1	0
1769-IQ32	2	0
1769-IR6	8	0
1769-IT6	8	0
1769-OA8	0	1
1769-OA16	0	1
1769-OB8	0	1
1769-OB16	0	1
1769-OB16P	0	1
1769-OB32	0	2
1769-OF2	2	2
1769-OF8C	11	9
1769-OF8V	11	9
1769-OV16	0	1
1769-OW8	0	1
1769-OW16	0	1
1769-OW8I	0	1
1769-SDN	66	2
1769-SM1	12	12

1769-SM2	7	7
1769-ASCII	108	108
1762-IA8	1	0
1762-IF2OF2	6	2
1762-IF4	7	0
1762-IQ8	1	0
1762-IQ8OW6	1	1
1762-IQ16	1	0
1762-OA8	0	1
1762-OB8	0	1
1762-OB16	0	1
1762-OW8	0	1
1762-OW16	0	1
1762-IT4	6	0
1762-IR4	6	0
1762-OF4	2	4
1762-OX6I	0	1

Calculation

Input Word Offset for Slot x = # Input Words in Slot 0 through Slot (x-1).

Note 1: The Embedded I/O needs to be taken into account when offsetting to Expansion I/O.

Note 2: The number of Output words does not factor into the calculation for Input Word Offset.

I/O Example

Let
Slot 0 = Micrologix 1500 LRP Series C = 4 Input Words
Slot 1 = 1769-OF2 = 2 Input Words
Slot 2 = 1769-OW8 = 0 Input Word
Slot 3 = 1769-IA16 = 1 Input Word
Slot 4 = 1769-OF8V = 11 Input Word
Bit 5 of Slot 3 = 4 + 2 = 6 words = I:6/5

SLC 500 Syntax

The default data type is shown in **bold**.

Syntax	Data Type	Access
I:<slot>	Short, **Word**, BCD	Read Only
I:<slot>.<word>	Short, **Word**, BCD	Read Only
I:<slot>/<bit>	**Boolean**	Read Only
I:<slot>.<word>/<bit>	**Boolean**	Read Only

Ranges

PLC Model	Min Slot	Max Slot	Max Word
Micrologix	NA	NA	2047
SLC 500 Open	NA	NA	1
SLC 5/01	1	30	*
SLC 5/02	1	30	*
SLC 5/03	1	30	*
SLC 5/04	1	30	*
SLC 5/05	1	30	*

PLC-5 Family	NA	NA	277

*The number of Input or Output words available for each I/O module can be found in the **Modular I/O Selection Guide**. For more information, refer to **Device Setup**.
**Octal.

Examples

Micrologix	Address
I:0	Word 0
I/2	Bit 2
I:1/5	Bit 5 word 1

SLC 500 Fixed I/O	Address
I:0	Word 0
I:1	Word 1
I/16	Bit 16
I:1/0	Bit 0 word 1 (same as I/16)

PLC5	Address
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)

*Addresses are shown in Octal.

SLC 500 Modular I/O	Address
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)

Status Files

To access Status files, specify a word and an optional bit in the word. The default data types are shown in **bold**.

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

*Array types.

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.

Ranges

PLC Model	Max Word
Micrologix	96
All SLC	96
PLC-5	127

The maximum word location is one less when accessing as a 32 bit data type (Long, DWord or Long BCD).

Examples

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.

Binary Files

To access Binary files, specify a file number, a word and optional bit in the word. The default data types are shown in **bold**.

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

*Array types.

The number of array elements (in bytes) cannot exceed the block request size specified. This means that the array size cannot exceed 16 words given a block request size of 32 bytes.

Ranges

PLC Model	File Number	Max Word
Micrologix	3-255	255
All SLC	3-255	255
PLC-5	3-999	1999

The maximum word location is one less when accessing as a 32 bit data type (Long, DWord or Long BCD).

Examples

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.

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 types are shown in **bold**.

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

The following fields are allowed for each element. For the usage of each field, refer to the PLC's documentation.

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

Ranges

PLC Model	File Number	Max Element
Micrologix	3-255	255
All SLC	3–255	255
PLC-5	3-999	1999

Examples

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 types are shown in **bold**.

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

The following fields are allowed for each element. For the meaning of each field, refer to the PLC's documentation.

Element Field	Data Type	Access
ACC	**Word**, Short	Read/Write
PRE	**Word**, Short	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

Ranges

PLC Model	File Number	Max Element

Micrologix	3–255	255
All SLC	3-255	255
PLC-5	3-999	1999

Examples

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 types are shown in **bold**.

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

The following fields are allowed for each element. For the meaning of each field, refer to the PLC's documentation.

Element Field	Data Type	Access
LEN	**Word**, Short	Read/Write
POS	**Word**, Short	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

Ranges

PLC Model	File Number	Max Element
Micrologix	3–255	255
All SLC	3–255	255
PLC-5	3-999	1999

Examples

Example	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, a word and an optional bit in the word. The default data types are shown in **bold**.

Syntax	Data Type	Access
N<file>:<word>	Short, **Word**, BCD,	Read/Write

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

*Array types.

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.

Ranges

PLC Model	File Number	Max Word
Micrologix	3-255	255
All SLC	3-255	255
PLC-5	3-999	1999

The maximum word location is one less when accessing as a 32 bit data type (Long, DWord or Long BCD).

Examples

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.

Float Files

To access Float files, specify a file number and an element. The default data types are shown in **bold**.

Syntax	Data Type	Access
F<file>:<element>	**Float**	Read/Write
F<file>:<element> [rows][cols]	**Float***	Read/Write
F<file>:<element> [cols]	**Float***	Read/Write

*Array type.

The number of array elements (in bytes) cannot exceed the block request size specified. This means that the array size cannot exceed 8 Floats given a block request size of 32 bytes.

Ranges

PLC Model	File Number	Max Word
Micrologix	3-255	255
All SLC	3-255	255
PLC-5	3-999	1999

Examples

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 a character location. The default data types are shown in **bold**.

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

*The number of array elements cannot exceed the block request size specified. Internally, 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 Allen-Bradley DF1 driver allows accessing to the character level.

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 Allen-Bradley DF1 driver, the 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.

Ranges

PLC Model	File Number	Max Character
Micrologix	3-255	511
All SLC	3-255	511
PLC-5	3-999	1999

Note: Not all Micrologix and SLC 500 PLC devices support ASCII file types. For more information, refer to the PLC's documentation.

Examples

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. Strings are 82 character null terminated arrays. The driver places the null terminator based on the string length returned by the PLC. The default data types are shown in **bold**.

Note: Arrays are not supported for String files.

Syntax	Data Type	Access
ST<file>:<element>.<field>	**String**	Read/Write

Ranges

PLC Model	File Number	Max Word
Micrologix	3-255	255

All SLC	3-255	255
PLC-5	3-999	999

Examples

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

BCD Files

To access BCD files, specify a file number and a word. The default data types are shown in **bold**.

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

*Array types.

The number of array elements (in bytes) cannot exceed the block request size specified. This means that the array size cannot exceed 16 BCD, given a block request size of 32 bytes.

Ranges

PLC Model	File Number	Max Word
Micrologix	NA	NA
All SLC	NA	NA
PLC-5	3-999	999

Examples

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.

Long Files

To access Long files, specify a file number and a DWord. The default data types are shown in **bold**.

Syntax	Data Type	Access
L<file>:<DWord>	Long, **DWord**, LBCD	Read/Write
L<file>:<DWord> [rows][cols]	Long, **DWord**, LBCD*	Read/Write
L<file>:<DWord> [cols]	Long, **DWord**, LBCD*	Read/Write

*Array types.

The number of array elements cannot exceed 16.

Ranges

PLC Model	File Number	Max Word
Micrologix	3-255	255
All SLC	NA	NA
PLC5	NA	NA

Examples

Example	Description
L9:0	Word 0.
L9:10 [8]	8 Element array starting at word 10.
L15:0 [4] [5]	4 by 5 element array starting at word 0.

Micrologix 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 types are shown in **bold**.

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

The following fields are allowed for each element. For the meaning of each field, refer to the PLC's documentation.

Element Field	Data Type	Access
SPS	**Word**, Short	Read/Write
KC	**Word**, Short	Read/Write
TI	**Word**, Short	Read/Write
TD	**Word**, Short	Read/Write
MAXS	**Word**, Short	Read/Write
MINS	**Word**, Short	Read/Write
ZCD	**Word**, Short	Read/Write
CVH	**Word**, Short	Read/Write
CVL	**Word**, Short	Read/Write
LUT	**Word**, Short	Read/Write
SPV	**Word**, Short	Read/Write
CVP	**Word**, Short	Read/Write
TM	**Boolean**	Read/Write
AM	**Boolean**	Read/Write
CM	**Boolean**	Read/Write
OL	**Boolean**	Read/Write
RG	**Boolean**	Read/Write
SC	**Boolean**	Read/Write
TF	**Boolean**	Read/Write
DA	**Boolean**	Read/Write
DB	**Boolean**	Read/Write
UL	**Boolean**	Read/Write
LL	**Boolean**	Read/Write
SP	**Boolean**	Read/Write
PV	**Boolean**	Read/Write
DN	**Boolean**	Read/Write
EN	**Boolean**	Read/Write

Ranges

PLC Model	File Number	Max Element
Micrologix	3-255	255
All SLC	NA	NA

PLC-5	*	*

*For more information, refer to **PLC5 PID Files**.

Examples

Example	Description
PD14:0.KC	Proportional gain of PD 0 file 14.
PD18:6.EN	PID enable bit of PD 6 file 18.

PLC5 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 types are shown in **bold**.

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

The following fields are allowed for each element. For the meaning of each field, refer to the PLC's documentation.

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
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	**Word,** Short	Read/Write
ELEM	**Word,** Short	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

Ranges

PLC Model	File Number	Max Element
Micrologix	*	*
All SLC	NA	NA
PLC-5	3-999	999

*For more information, refer to **Micrologix PID Files**.

Examples

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

Micrologix 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 types are shown in **bold**.

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

The following fields are allowed for each element. For the meaning of each field, refer to the PLC's documentation.

Element Field	Data Type	Access
IA	**Word**, Short	Read/Write
RBL	**Word**, Short	Read/Write
LBN	**Word**, Short	Read/Write
RBN	**Word**, Short	Read/Write
CHN	**Word**, Short	Read/Write
NOD	**Word**, Short	Read/Write
MTO	**Word**, Short	Read/Write
NB	**Word**, Short	Read/Write
TFT	**Word**, Short	Read/Write
TFN	**Word**, Short	Read/Write
ELE	**Word**, Short	Read/Write

SEL	**Word**, Short	Read/Write
TO	**Boolean**	Read/Write
CO	**Boolean**	Read/Write
EN	**Boolean**	Read/Write
RN	**Boolean**	Read/Write
EW	**Boolean**	Read/Write
ER	**Boolean**	Read/Write
DN	**Boolean**	Read/Write
ST	**Boolean**	Read/Write
BK	**Boolean**	Read/Write

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

Ranges

PLC Model	File Number	Max Element
Micrologix	3-255	255
All SLC	NA	NA
PLC5	*	*

*For more information, refer to **PLC5 Message**.

Examples

Example	Description
MG14:0.TO	Ignore if timed out bit of MG 0 file 14.
MG18:6.CO	Continue bit of MG 6 file 18.

PLC5 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 types are shown in **bold**.

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

The following fields are allowed for each element. For the meaning of each field, refer to the PLC's documentation.

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

Ranges

PLC Model	File Number	Max Element

Micrologix	*	*
All SLC	NA	NA
PLC-5	3-999	999

*For more information, refer to **Micrologix Message Files**.

Examples

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 types are shown in **bold**.

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

The following fields are allowed for each element. For the meaning of each field, refer to the PLC's documentation.

Element Field	Data Type	Access
RLEN	**Word**, Short	Read/Write
DLEN	**Word**, Short	Read/Write
FILE	**Word**, Short	Read/Write
ELEM	**Word**, Short	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

Ranges

PLC Model	File Number	Max Element
Micrologix	NA	NA
All SLC	NA	NA
PLC-5	3-999	1999

Examples

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

High Speed Counter File (HSC)

The HSC files are a structured type whose data is accessed by specifying an element and a field. The default data types are shown in **bold**.

See Also: **Function File Options**

Syntax	Data Type	Access
HSC:<element>.<field>	Depends on field	Depends on field

The following fields are allowed for each element. For the meaning of each field, refer to the PLC's documentation.

Element Field	Default Type	Access
ACC	**DWord**, Long	Read Only
HIP	**DWord**, Long	Read/Write
LOP	**DWord**, Long	Read/Write
OVF	**DWord**, Long	Read/Write
UNF	**DWord**, Long	Read/Write
PFN	**Word**, Short	Read Only
ER	**Word**, Short	Read Only
MOD	**Word**, Short	Read Only
OMB	**Word**, Short	Read Only
HPO	**Word**, Short	Read/Write
LPO	**Word**, Short	Read/Write
UIX	**Boolean**	Read Only
UIP	**Boolean**	Read Only
AS	**Boolean**	Read Only
ED	**Boolean**	Read Only
SP	**Boolean**	Read Only
LPR	**Boolean**	Read Only
HPR	**Boolean**	Read Only
DIR	**Boolean**	Read Only
CD	**Boolean**	Read Only
CU	**Boolean**	Read Only
UIE	**Boolean**	Read/Write
UIL	**Boolean**	Read/Write
FE	**Boolean**	Read/Write
CE	**Boolean**	Read/Write
LPM	**Boolean**	Read/Write
HPM	**Boolean**	Read/Write
UFM	**Boolean**	Read/Write
OFM	**Boolean**	Read/Write
LPI	**Boolean**	Read/Write
HPI	**Boolean**	Read/Write
UFI	**Boolean**	Read/Write
OFI	**Boolean**	Read/Write
UF	**Boolean**	Read/Write
OF	**Boolean**	Read/Write
MD	**Boolean**	Read/Write

Ranges

PLC Model	File Number	Max Element
Micrologix	N/A	254
All SLC	N/A	N/A
PLC5	N/A	N/A

Examples

Example	Description
HSC:0.OMB	Output mask setting for high speed counter 0.
HSC:1.ED	Error detected indicator for high speed counter 1.

Real-Time Clock File (RTC)

The RTC files are a structured type whose data is accessed by specifying an element and a field. The default data types are shown in **bold**.

See Also: Function File Options

Syntax	Data Type	Access
RTC:<element>.<field>	Depends on field	Depends on field

The following fields are allowed for each element. For the meaning of each field, refer to the PLC's documentation.

Element Field	Data Type	Access
YR	**Word**, Short	Read/Write
MON	**Word**, Short	Read/Write
DAY	**Word**, Short	Read/Write
HR	**Word**, Short	Read/Write
MIN	**Word**, Short	Read/Write
SEC	**Word**, Short	Read/Write
DOW	**Word**, Short	Read/Write
DS	**Boolean**	Read Only
BL	**Boolean**	Read Only
_SET (for block writes)	**Boolean**	Read/Write

Ranges

PLC Model	File Number	Max Element
Micrologix	N/A	254
All SLC	N/A	N/A
PLC5	N/A	N/A

Examples

Example	Description
RTC:0.YR	Year setting for real-time clock 0.
RTC:0.BL	Battery low indicator for real-time clock 0.

Channel 0 Communication Status File (CS0)

To access the communication status file for channel 0, specify a word and optionally a bit in the word. The default data types are shown in **bold**.

See Also: Function File Options

Syntax	Data Type	Access
CS0:<word>	Short, **Word**, BCD, DWord, Long, LBCD	Depends on <word> and <bit>
CS0:<word>/<bit>	**Boolean**	Depends on <word> and <bit>

CS0/bit	**Boolean**	Depends on <word> and <bit>

Ranges

PLC Model	File Number	Max Element
Micrologix	N/A	254
All SLC	N/A	N/A
PLC5	N/A	N/A

Examples

Example	Description
CS0:0	Word 0.
CS0:4/2	Bit 2 word 4 = MCP.

Note: For more information on CS0 words/bit meanings, refer to the Rockwell documentation.

Channel 1 Communication Status File (CS1)

To access the communication status file for channel 1, specify a word and optionally a bit in the word. The default data types are shown in **bold**.

See Also: Function File Options

Syntax	Data Type	Access
CS1:<word>	Short, **Word**, BCD, DWord, Long, LBCD	Depends on <word> and <bit>
CS1:<word>/<bit>	**Boolean**	Depends on <word> and <bit>
CS1/bit	**Boolean**	Depends on <word> and <bit>

Ranges

PLC Model	File Number	Max Element
Micrologix	N/A	254
All SLC	N/A	N/A
PLC5	N/A	N/A

Examples

Example	Description
CS1:0	Word 0
CS1:4/2	Bit 2 word 4 = MCP

Note: For more information on CS1 words/bit meanings, refer to the Rockwell documentation.

I/O Module Status File (IOS)

To access the I/O module status file, specify a word and optionally a bit in the word. The default data type for each syntax is shown in **bold**.

See Also: Function File Options

Syntax	Data Type	Access
IOS:<word>	Short, **Word**, BCD, DWord, Long, LBCD	Depends on <word> and <bit>

IOS:<word>/<bit>	**Boolean**	Depends on <word> and <bit>
IOS/bit	**Boolean**	Depends on <word> and <bit>

Ranges

PLC Model	File Number	Max Element
Micrologix	N/A	254
All SLC	N/A	N/A
PLC5	N/A	N/A

Examples

Example	Description
IOS:0	Word 0.
IOS:4/2	Bit 2 word 4.

Note: For a listing of 1769 expansion I/O status codes, refer to the instruction manual.

Error Descriptions

The following error/warning 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>'

Serial Communications

COMn does not exist
Error opening COMn
COMn is in use by another application
Unable to set comm parameters on COMn
Communications error on COMn [<error mask>]

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 <start address> on device '<device name>' [Status=<STS>, Ext. Status=<EXT. STS>]
Unable to read data starting at <start address> on device '<device name>' [Status=<STS>, Ext. Status=<EXT. STS>]. Block deactivated
Unable to read function file <fun. file address> on device '<device name>' [Status=<STS>, Ext. Status=<EXT. STS>]
Unable to read function file <fun. file element> on device '<device name>' [Status=<STS>, Ext. Status=<EXT. STS>]. Block deactivated
Unable to read data starting at <start address> on device '<device name>'. Framing error
Unable to read function file <fun. file element> on device '<device name>'. Framing error

Address Validation

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

Address Validation

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.

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.

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

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>' 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.

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.

Serial Communications

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

Serial Communications
COMn does not exist
Error opening COMn
COMn is in use by another application
Unable to set comm parameters on COMn
Communications error on COMn [<error mask>]

COMn does not exist

Error Type:
Fatal

Possible Cause:
The specified COM port is not present on the target computer.

Solution:
Verify that the proper COM port has been selected.

Error opening COMn

Error Type:
Fatal

Possible Cause:
The specified COM port could not be opened due an internal hardware or software problem on the target computer.

Solution:
Verify that the COM port is functional and may be accessed by other Windows applications.

COMn is in use by another application

Error Type:
Fatal

Possible Cause:
The serial port assigned to a device is being used by another application.

Solution:

Verify that the correct port has been assigned to the channel.

Unable to set comm parameters on COMn

Error Type:
Fatal

Possible Cause:
The serial parameters for the specified COM port are not valid.

Solution:
Verify the serial parameters and make any necessary changes.

Communications error on COMn [<error mask>]

Error Type:
Serious

Error Mask Definitions:
B = Hardware break detected.
F = Framing error.
E = I/O error.
O = Character buffer overrun.
R = RX buffer overrun.
P = Received byte parity error.
T = TX buffer full.

Possible Cause:
1. The serial connection between the device and the Host PC is bad.
2. The communications parameters for the serial connection are incorrect.

Solution:
1. Verify the cabling between the PC and the device.
2. Verify that the specified communications parameters match those of the device.

Device Status Messages

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

Device Status Messages
Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>'

Device '<device name>' is not responding

Error Type:
Serious

Possible Cause:
1. The serial connection between the device and the Host PC is broken.
2. The communications parameters for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.

Solution:
1. Verify the cabling between the PC and the device.
2. Verify that the specified communication parameters match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.

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

Error Type:
Serious

Possible Cause:
1. The serial connection between the device and the Host PC is broken.
2. The communication parameters for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.

Solution:
1. Verify the cabling between the PC and the device.
2. Verify that the specified communication parameters match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.

Device Specific Messages

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

Device Specific Messages

Unable to read data starting at <start address> on device '<device name>' [Status=<STS>, Ext. Status=<EXT. STS>]

Error Type:
Serious

Possible Cause:
1. Node cannot be found.
2. Duplicate node detected.

Solution:
Check the status and extended status codes that are being returned by the PLC. The codes are displayed in hexadecimal.

Note:
This error message applies to local node errors. Status code errors in the low nibble of the status code indicate errors found by the local node. The driver will continue 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.

Unable to read data starting at <start address> on device '<device name>' [Status=<STS>, Ext. Status=<EXT. STS>]. Block deactivated

Error Type:
Serious

Possible Cause:
1. The address requested in the block does not exist in the PLC.
2. Processor is in program mode.

Solution:
Check the status and extended status codes that are being returned by the PLC. The codes are displayed in hexadecimal.

Note:
This error message applies to remote node errors. 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 will not ask for these blocks again after receiving this kind of error. This kind of error can be generated if the address does not exist in the PLC.

Unable to read function file <fun. file element> on device '<device name>' [Status=<STS>, Ext. Status=< EXT. STS>]

Error Type:
Serious

Possible Cause:
1. Node cannot be found.
2. Duplicate node detected.

Solution:
Check the status and extended status codes that are being returned by the PLC. The codes are displayed in hexadecimal.

Note:
This error message applies to local node errors. Status code errors in the low nibble of the status code indicate errors found by the local node. The driver will continue to retry reading this function file periodically. Errors found by the local node occur when the KF module cannot see the destination PLC on the network for some reason.

Unable to read function file <fun. file element> on device '<device name>' [Status=<STS>, Ext. Status=< EXT. STS>]. Block deactivated

Error Type:
Serious

Possible Cause:
1. The function file address requested in the block does not exist in the PLC.
2. Processor is in program mode.

Solution:
Check the status and extended status codes that are being returned by the PLC. The codes are displayed in hexadecimal.

Note:
This error message applies to remote node errors. Status code errors in the high nibble of the status code indicate errors found by the PLC. These errors are generated when the function file the driver is asking for is not available in the PLC. The driver will not ask for this function file again after receiving this kind of error. This kind of error can be generated if the function file address does not exist in the PLC.

Unable to read data starting at <start address> on device '<device name>'. Framing error

Error Type:
Serious

Possible Cause:
1. Unexpected frame received.
2. Frame size mismatch.

Solution:
The device is returning an invalid read response or one of unexpected size. If this error occurs frequently, contact Technical Support.

Unable to read function file <fun. file element> on device '<device name>'. Framing error

Error Type:
Serious

Possible Cause:
1. Unexpected frame received.
2. Frame size mismatch.

Solution:
The device is returning an invalid function file read response or one of unexpected size. If this error occurs frequently, contact Technical Support.

Unable to read data starting at <start address> on device '<device name>'. checksum error

Error Type:
Serious

Possible Cause:
There is bad cabling connecting the devices causing noise and checksum errors.

Solution:
Inspect cabling between the Host PC and the device.

Unable to read function file <fun. file element> on device '<device name>'. Checksum error

Error Type:
Serious

Possible Cause:
There is bad cabling connecting the devices causing noise and checksum errors.

Solution:
Inspect cabling between the Host PC and the device.

Unable to read data starting at <start address> on device '<device name>'. Slave sink/source full

Error Type:
Serious

Possible Cause:
The slave device cannot accept anymore requests from the master. The client may be requesting data too fast.

Solution:
The driver will automatically poll and re-poll the slave to empty its source and in turn make room for responses from requests previously in the full sink. If this error occurs too often, decrease the update rate on suspected tags.

Unable to read function file <fun. file element> on device '<device name>'. Slave sink/source full

Error Type:
Serious

Possible Cause:
The slave device cannot accept anymore requests from the master. The client may be requesting data too fast.

Solution:
The driver will automatically poll and re-poll the slave to empty its source and in turn make room for responses from requests previously in the full sink. If this error occurs too often, decrease the update rate on suspected function file tags.

Unable to read data starting at <start address> on device '<device name>'. Slave source empty

Error Type:
Serious

Possible Cause:
The slave device does not have a response prepared for the data request starting at <start address>. The slave re-poll delay may be set to short.

Solution:
The driver will automatically poll and re-poll the slave in seek of a poll response. If this error occurs too often, increase the slave re-poll delay on the given channel.

Unable to read function file <fun. file element> on device '<device name>'. Slave source empty

Error Type:
Serious

Possible Cause:
The slave device does not have a response prepared for the request of function file <fun. file element>. The slave re-poll delay may be set to short.

Solution:
The driver will automatically poll and re-poll the slave in seek of a poll response. If this error occurs too often, increase the slave re-poll delay on the given channel.

Error writing to address '<address>' on device '<device name>' [Status=<STS>, Ext. Status=<EXT. STS>]

Error Type:

Serious

Possible Cause:

1. Node cannot be found.
2. Duplicate node detected.
3. The address requested in the block does not exist in the PLC.
4. Processor is in program mode.

Solution:

Check the status and extended status codes that are being returned by the PLC. The codes are displayed in hexadecimal.

Note:

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 block of data the driver is asking for is not available in the PLC.

Error writing to address '<address>' on device '<device name>'. Framing error

Error Type:

Serious

Possible Cause:

1. Unexpected frame received.
2. Frame size mismatch.

Solution:

The device is returning an invalid write response or one of unexpected size. If this error occurs frequently, contact Technical Support.

Checksum error occurred writing to address '<address>' on device '<device name>'

Error Type:

Serious

Possible Cause:

There is bad cabling connecting the devices causing noise and checksum errors.

Solution:

Inspect cabling between the Host PC and the device.

Error writing to address '<address>' on device '<device name>'. Slave sink/source full

Error Type:

Serious

Possible Cause:

The slave device cannot accept any more requests from the master. The client may be requesting data too fast.

Solution:

The driver will automatically poll and re-poll the slave to empty its source and in turn make room for responses from requests previously in the full sink. If this error occurs too often, decrease the update rate on suspected tags, not necessarily the tag being written to.

Error writing to address '<address>' on device '<device name>'. Slave source empty

Error Type:
Serious

Possible Cause:
The slave device does not have a response prepared for the write request to address < address>. The slave re-poll delay may be set to short.

Solution:
The driver will automatically poll and re-poll the slave in seek of a poll response. If this error occurs too often, increase the slave re-poll delay on the given channel.

Device '<device name>' timed out writing to address '<address>'

Error Type:
Serious

Possible Cause:
1. The device is not responding.
2. The serial connection between the device and the Host PC is broken.

Solution:
Inspect cabling between the Host PC and the device. Verify the device is on and operating properly.

Unable to read data starting at <start address> on device '<device name>'. Device replied with a NAK

Error Type:
Serious

Possible Cause:
The server sent an invalid response or one of unexpected size.

Solution:
If this error occurs frequently, contact Technical Support.

Unable to read function file <fun. file element> on device '<device name>'. Device replied with a NAK

Error Type:
Serious

Possible Cause:
The server sent an invalid response or one of unexpected size.

Solution:
If this error occurs frequently, contact Technical Support.

Unable to read data starting at <start address> on device '<device name>'. Memory map error

Error Type:
Serious

Possible Cause:

There was an error writing to memory on the server.

Solution:
If this error occurs frequently, contact Technical Support.

Unable to read function file <fun. file element> on device '<device name>'. Memory map error

Error Type:
Serious

Possible Cause:
There was an error writing to memory on the server.

Solution:
If this error occurs frequently, contact Technical Support.

Index

- 1 -

- 2 -

- A -

- B -

- C -

- D -

- E -

- F -

- H -

- I -

- L -

- M -

- O -

- P -

- R -

- S -

- T -

- U -

- W -