Thermo Westronics
Serial Driver Help

© 2012 Kepware Technologies

Table of Contents

Thermo Westronics Serial Driver Help

Help version 1.014

CONTENTS

Overview
What is the Thermo Westronics Serial Driver?

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

Data Types Description
What data types does this driver support?

Address Descriptions
How do I address a data location on a Thermo Westronics Serial device?

Error Descriptions
What error messages does the Thermo Westronics Serial driver produce?

Overview

The Thermo Westronics Serial Driver provides an easy and reliable way to connect Thermo Westronics Serial devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It is intended for use with Thermo Westronics serial devices.

Device Setup

Supported Devices
Series 1200 Recorder
Series 1600 Recorder
Series 3000 Recorder
SM 100 Smart Multiplexer
SV 100
SV 180
SV 180 (2.0)-version 2.0A or later

Communication Protocol
Modbus RTU Protocol.

Supported Communication Parameters
Baud Rate: 300, 600, 1200, 2400, 4800, 9600, 19200
Parity: Odd, Even, None
Data Bits: 5, 6, 7, 8
Stop Bits: 1,2

Note: Not all devices support the listed configurations.

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. It may be invoked through the COM ID dialog in Channel Properties. For more information, refer to the OPC Server's help documentation.

Device ID (PLC Network Address)
Thermo Westronics Serial devices are assigned Device IDs in the range 1 to 255.

Flow Control
When using an RS232/RS485 converter, the type of flow control that is required will depend upon the needs of the converter. Some converters do not require any flow control whereas others require RTS flow. Consult the converter's documentation in order to determine its flow requirements. An RS485 converter that provides automatic flow control is recommended.

Note: When using the manufacturer's supplied communications cable, it is sometimes necessary to choose a flow control setting of **RTS** or **RTS Always** under the Channel Properties.

The Thermo Westronics Serial driver supports the RTS Manual flow control option. This selection allows the driver to be configured for operation with radio modems that require special RTS timing characteristics. For more information on RTS Manual flow, refer to the main OPC Server help file topic Channel Wizard.

Cable Connections
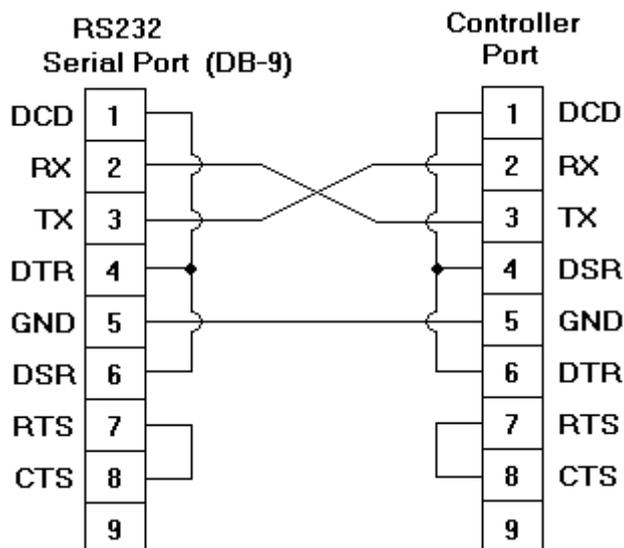
The diagram shown above is typical. Various Thermo Westronics devices support other types of connections such as RS422, RS485 and 20mA current loop. Modbus communications may not be possible through all of the ports on the device. For more information on connection options, refer to the specific device's documentation.

Modem Setup

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

Block Sizes

Coil Block Sizes

Coils can be read from 8 to 800 points (bits) at a time. A higher block size means more points will be read from the device in a single request. The block size can be reduced if data needs to be read from non-contiguous locations within the device.

Register Block Sizes

Registers can be read from 1 to 56 locations (16 or 32-bit registers) at a time. A higher block size means more register values will be read from the device in a single request. The block size can be reduced if data needs to be read from non-contiguous locations within the device.

Settings

1x32 Floating Point Format

Some Thermo Westronics devices, such as the SV100 and SV180, give the user the option of representing floating point data in either 1x32 or 2x16 format in Modbus communications. If the device has been configured to use the 1x32 format, make sure that this box is checked.

Data Types Description

Data Type	Description
Boolean	Single bit
Word	Unsigned 16 bit value bit 0 is the low bit bit 15 is the high bit
Short	Signed 16 bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32 bit value bit 0 is the low bit bit 31 is the high bit
Long	Signed 32 bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
Float	32 bit floating point value. The driver interprets two consecutive 16 bit registers as a floating point value by making the second register the high word and the first register the low word.
String	Zero terminated character array

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.

Series 1200
Series 1600
Series 3000
SM100
SV100
SV180*
SV180 (2.0)**

*Firmware prior to 2.0A.
**Firmware 2.0A and later.

Series 1200 Addressing

The Thermo Westronics Ethernet device driver automatically generates tags for many of the most important device parameters. Tags can be manually created to access all device parameters mapped to Modbus address locations within the device. The following table describes how to specify a Modbus address when creating a tag. The full range of addresses accepted by this driver may not be valid for a particular device.

Note: The default data types for dynamically defined tags are shown in **bold**.

Modbus Addressing Decimal Format

Address	Range	Data Type	Access
Output Coils	000001-065536	**Boolean**	Read/Write
Input Coils	100001-165536	**Boolean**	Read Only
Internal Registers	300001-305000	**Word**, Short	Read Only
	300001.0-300001.15...305000.0-305000.15	**Boolean**	
	305001-306000	**DWord**, Long	
	306001-307000	**Word**, Short	
	306001.0-306001.15... 307000.0-307000.15	**Boolean**	
	307001-308000	**Float**	
	308001-365536	**Word**, Short	
	308001.0-308001.15...365536.0-365536.15	**Boolean**	
Holding Registers	400001-406000	**Word**, Short	Read/Write
	400001.0-400001.15...406000.0-406000.15	**Boolean**	
	406001-407000	**DWord**, Long	
	407001-408000	**Word**, Short	
	407001.0-407001.15... 408000.0-408000.15	**Boolean**	
	408001-409000	**Float**	
	409001-465536	**Word**, Short	
	409001.0-409001.15...465536.0-465536.15	**Boolean**	

Modbus Addressing Hexadecimal Format

Address	Range	Data Type	Access
Output Coils	H000001-H010000	**Boolean**	Read/Write

Input Coils	H100001-H110000	**Boolean**	Read Only
Internal Registers	H300001-H301388	**Word**, Short	Read Only
	H300001.0-H300001.F...H301388.0-H301388.F	**Boolean**	
	H301389-H301770	**DWord**, Long	
	H301771-H301B58	**Word**, Short	
	H301771.0-H301771.F ...H301B58.0-H301B58.F	**Boolean**	
	H301B59-H301F40	**Float**	
	H301F41-H310000	**Word**, Short	
	H301F41.0-H301F41.F...H31000.0-H310000.F	**Boolean**	
Holding Registers	H400001-H401770	**Word**, Short	Read/Write
	H400001.0-H400001.F...H401770.0-H401770.F	**Boolean**	
	H401771-H401B58	**DWord**, Long	
	H401B59-H401F40	**Word**, Short	
	H401B59.0-H401B59.F...H401F40.0-H401F40.F	**Boolean**	
	H401F41-H402328	**Float**	
	H402329-H410000	**Word**, Short	
	H402329.0-H402329.F...H410000.0-H410000.F	**Boolean**	

Array Support

Arrays are supported for 16-bit internal and holding register locations for all data types except Boolean and strings. There are two methods of addressing an array. Examples are given using holding register locations.

4xxxx [rows] [cols]
4xxxx [cols] (this method assumes rows is equal to one)

Rows multiplied by cols cannot exceed the block size that has been assigned to the device for the register type. For arrays of 32 bit data types, rows multiplied by cols multiplied by 2 cannot exceed the block size.

Series 1600 Addressing

The Thermo Westronics Ethernet device driver automatically generates tags for many of the most important device parameters. Tags can be manually created to access all device parameters mapped to Modbus address locations within the device. The following table describes how to specify a Modbus address when creating a tag. The full range of addresses accepted by this driver may not be valid for a particular device.

Note: The default data types for dynamically defined tags are shown in **bold**.

Modbus Addressing Decimal Format

Address	Range	Data Type	Access
Output Coils	000001-065536	**Boolean**	Read/Write
Input Coils	100001-165536	**Boolean**	Read Only
Internal Registers	300001-305000	**Word**, Short	Read Only
	300001.0-300001.15...305000.0-305000.15	**Boolean**	
	305001-306000	**DWord**, Long	
	306001-307000	**Word**, Short	
	306001.0-306001.15... 307000.0-307000.15	**Boolean**	

	307001-308000	**Float**	
	308001-365536	**Word**, Short	
	308001.0-308001.15...365536.0-365536.15	**Boolean**	
Holding Registers	400001-406000	**Word**, Short	Read/Write
	400001.0-400001.15...406000.0-406000.15	**Boolean**	
	406001-407000	**DWord**, Long	
	407001-408000	**Word**, Short	
	407001.0-407001.15 ...408000.0-408000.15	**Boolean**	
	408001-409000	**Float**	
	409001-465536	**Word**, Short	
	409001.0-409001.15 ...465536.0-465536.15	**Boolean**	

Modbus Addressing Hexadecimal Format

Address	Range	Data Type	Access
Output Coils	H000001-H010000	**Boolean**	Read/Write
Input Coils	H100001-H110000	**Boolean**	Read Only
Internal Registers	H300001-H301388	**Word**, Short	Read Only
	H300001.0-H300001.F...H301388.0-H301388.F	**Boolean**	
	H301389-H301770	**DWord**, Long	
	H301771-H301B58	**Word**, Short	
	H301771.0-H301771.F...H301B58.0-H301B58.F	**Boolean**	
	H301B59-H301F40	**Float**	
	H301F41-H310000	**Word**, Short	
	H301F41.0-H301F41.F... H31000.0-H310000.F	**Boolean**	
Holding Registers	H400001-H401770	**Word**, Short	Read/Write
	H400001.0-H400001.F...H401770.0-H401770.F	**Boolean**	
	H401771-H401B58	**DWord**, Long	
	H401B59-H401F40	**Word**, Short	
	H401B59.0-H401B59.F... H401F40.0-H401F40.F	**Boolean**	
	H401F41-H402328	**Float**	
	H402329-H410000	**Word**, Short	
	H402329.0-H402329.F...H410000.0-H410000.F	**Boolean**	

Array Support

Arrays are supported for 16-bit internal and holding register locations for all data types except Boolean and strings. There are two methods of addressing an array. Examples are given using holding register locations.

4xxxx [rows] [cols]
4xxxx [cols] (this method assumes rows is equal to one)

Rows multiplied by cols cannot exceed the block size that has been assigned to the device for the register type. For arrays of 32 bit data types, rows multiplied by cols multiplied by 2 cannot exceed the block size.

Series 3000 Addressing

The Thermo Westronics Ethernet device driver automatically generates tags for many of the most important device parameters. Tags can be manually created to access all device parameters mapped to Modbus address locations within the device. The following table describes how to specify a Modbus address when creating a tag. The full range of addresses accepted by this driver may not be valid for a particular device.

Note: The default data types for dynamically defined tags are shown in **bold**.

Modbus Addressing Decimal Format

Address	Range	Data Type	Access
Output Coils	000001-065536	**Boolean**	Read/Write
Input Coils	100001-165536	**Boolean**	Read Only
Internal Registers	300001-365536 300001.0-300001.15...365536.0-365536.15	**Word**, Short **Boolean**	Read Only
Holding Registers	400001-465536 400001.0-400001.15...465536.0-465536.15	**Word**, Short **Boolean**	Read/Write

Modbus Addressing Hexadecimal Format

Address	Range	Data Type	Access
Output Coils	H000001-H010000	**Boolean**	Read/Write
Input Coils	H100001-H110000	**Boolean**	Read Only
Internal Registers	H300001-H310000 H300001.0-H300001.F...H310000.0-H310000.F	**Word**, Short **Boolean**	Read Only
Holding Registers	H400001-H410000 H400001.0-H400001.F...H410000.0-H410000.F	**Word**, Short **Boolean**	Read/Write

Array Support

Arrays are supported for 16-bit internal and holding register locations for all data types except Boolean and strings. There are two methods of addressing an array. Examples are given using holding register locations.

4xxxx [rows] [cols]
4xxxx [cols] this method assumes rows is equal to one

Rows multiplied by cols cannot exceed the block size that has been assigned to the device for the register type. For arrays of 32 bit data types, rows multiplied by cols multiplied by 2 cannot exceed the block size.

SM100 Addressing

The Thermo Westronics Ethernet device driver automatically generates tags for many of the most important device parameters. Tags can be manually created to access all device parameters mapped to Modbus address locations within the device. The following table describes how to specify a Modbus address when creating a tag. The full range of addresses accepted by this driver may not be valid for a particular device.

Note: The default data types for dynamically defined tags are shown in **bold**.

Modbus Addressing Decimal Format

Address	Range	Data Type	Access
Output Coils	000001-065536	**Boolean**	Read/Write
Input Coils	100001-165536	**Boolean**	Read Only
Internal Registers	300001-365536 300010-300109 300001.0-300001.15...365536.0-365536.15	**Word**, Short Float* **Boolean**	Read Only
Holding Registers	400001-465536	**Word**, Short	Read/Write

| | 400010-400109 | Float* | |
| | 400001.0-400001.15...465536.0-465536.15 | **Boolean** | |

Modbus Addressing Hexadecimal Format

Address	Range	Data Type	Access
Output Coils	H000001-H010000	**Boolean**	Read/Write
Input Coils	H100001-H110000	**Boolean**	Read Only
Internal Registers	H300001-H310000	**Word**, Short	Read Only
	H30000A-H30006D	Float*	
	H300001.0-H300001.F...H310000.0-H310000.F	**Boolean**	
Holding Registers	H400001-H410000	**Word**, Short	Read/Write
	H40000A-H40006D	Float*	
	H400001.0-H400001.F...H410000.0-H410000.F	**Boolean**	

*When point data values are read as floats, the raw data (registers 400010 to 400109, or 300010 to 300109) is automatically scaled using the decimal placement data (registers 405037, 405047, 405057, etc.). Decimal placement data is read from the device on server project startup. It is assumed that the raw data will be a 16-bit signed integer (−32768 to 32767).

Array Support

Arrays are supported for 16-bit internal and holding register locations for all data types except Boolean, float, and strings. There are two methods of addressing an array. Examples are given using holding register locations.

4xxxx [rows] [cols]
4xxxx [cols] (this method assumes rows is equal to one)

Rows multiplied by cols cannot exceed the block size that has been assigned to the device for the register type. For arrays of 32 bit data types, rows multiplied by cols multiplied by 2 cannot exceed the block size.

SV100 Addressing

The Thermo Westronics Ethernet device driver automatically generates tags for many of the most important device parameters. Tags can be manually created to access all device parameters mapped to Modbus address locations within the device. The following table describes how to specify a Modbus address when creating a tag. The full range of addresses accepted by this driver may not be valid for a particular device.

Note: The default data types for dynamically defined tags are shown in **bold**.

Modbus Addressing Decimal Format

Address	Range	Data Type	Access
Output Coils	000001-065536	**Boolean**	Read/Write
Input Coils	100001-165536	**Boolean**	Read Only
Internal Registers	300001-305000	**Word**, Short	Read Only
	300001.0-300001.15...305000.0-305000.15	**Boolean**	
	305001-306000	**DWord**, Long	
	306001-307000	**Word**, Short	
	306001.0-306001.15...307000.0-307000.15	**Boolean**	
	307001-308000	**Float**	
	308001-365536	**Word**, Short	
	308001.0-308001.15...365536.0-365536.15	**Boolean**	

Holding Registers	400001-406000	**Word**, Short	Read/Write
	400001.0-400001.15...406000.0-406000.15	**Boolean**	
	406001-407000	**DWord**, Long	
	407001-408000	**Word**, Short	
	407001.0-407001.15...408000.0-408000.15	**Boolean**	
	408001-409000	**Float**	
	409001-465536	**Word**, Short	
	409001.0-409001.15...465536.0-465536.15	**Boolean**	

Modbus Addressing Hexadecimal Format

Address	Range	Data Type	Access
Output Coils	H000001-H010000	**Boolean**	Read/Write
Input Coils	H100001-H110000	**Boolean**	Read Only
Internal Registers	H300001-H301388	**Word**, Short	Read Only
	H300001.0-H300001.F...H301388.0-H301388.F	**Boolean**	
	H301389-H301770	**DWord**, Long	
	H301771-H301B58	**Word**, Short	
	H301771.0-H301771.F...H301B58.0-H301B58.F	**Boolean**	
	H301B59-H301F40	**Float**	
	H301F41-H310000	**Word**, Short	
	H301F41.0-H301F41.F...H31000.0-H310000.F	**Boolean**	
Holding Registers	H400001-H401770	**Word**, Short	Read/Write
	H400001.0-H400001.F...H401770.0-H401770.F	**Boolean**	
	H401771-H401B58	**DWord**, Long	
	H401B59-H401F40	**Word**, Short	
	H401B59.0-H401B59.F...H401F40.0-H401F40.F	**Boolean**	
	H401F41-H402328	**Float**	
	H402329-H410000	**Word**, Short	
	H402329.0-H402329.F...H410000.0-H410000.F	**Boolean**	

Array Support

Arrays are supported for 16-bit internal and holding register locations for all data types except Boolean and strings. There are two methods of addressing an array. Examples are given using holding register locations.

4xxxx [rows] [cols]
4xxxx [cols] this method assumes rows is equal to one

Rows multiplied by cols cannot exceed the block size that has been assigned to the device for the register type.
For arrays of 32 bit data types, rows multiplied by cols multiplied by 2 cannot exceed the block size.

SV180 Addressing

The Thermo Westronics Ethernet device driver automatically generates tags for many of the most important device parameters. Tags can be manually created to access all device parameters mapped to Modbus address locations within the device. The following table describes how to specify a Modbus address when creating a tag. The full range of addresses accepted by this driver may not be valid for a particular device.

Important: The SV180 model is for SV180 devices using firmware version prior to 2.0A. If the device uses firmware version 2.0A or later, select the **SV180 (2.0)** model.

Note: The default data types for dynamically defined tags are shown in **bold**.

Modbus Addressing Decimal Format

Address	Range	Data Type	Access
Output Coils	000001-065536	**Boolean**	Read/Write
Input Coils	100001-165536	**Boolean**	Read Only
Internal Registers	300001-305000	**Word**, Short	Read Only
	300001.0-300001.15...305000.0-305000.15	**Boolean**	
	305001-306000	**DWord**, Long	
	306001-307000	**Word**, Short	
	306001.0-306001.15...307000.0-307000.15	**Boolean**	
	307001-308000	**Float**	
	308001-365536	**Word**, Short	
	308001.0-308001.15...365536.0-365536.15	**Boolean**	
Holding Registers	400001-405000	**Word**, Short	Read/Write
	400001.0-400001.15...405000.0-405000.15	**Boolean**	
	405001-406000	**DWord**, Long	
	406001-407000	**Word**, Short	
	406001.0-406001.15...407000.0-407000.15	**Boolean**	
	407001-408000	**Float**	
	408001-465536	**Word**, Short	
	408001.0-408001.1 ...465536.0-465536.15	**Boolean**	

Modbus Addressing Hexadecimal Format

Address	Range	Data Type	Access
Output Coils	H000001-H010000	**Boolean**	Read/Write
Input Coils	H100001-H110000	**Boolean**	Read Only
Internal Registers	H300001-H301388	**Word**, Short	Read Only
	H300001.0-H300001.F...H301388.0-H301388.F	**Boolean**	
	H301389-H301770	**DWord**, Long	
	H301771-H301B58	**Word**, Short	
	H301771.0-H301771.F...H301B58.0-H301B58.F	**Boolean**	
	H301B59-H301F40	**Float**	
	H301F41-H310000	**Word**, Short	

	H301F41.0-H301F41.F...H310000.0-H310000.F	**Boolean**	
Holding Registers	H400001-H401388	**Word**, Short	Read/Write
	H400001.0-H400001.F...H401388.0-H401388.F	**Boolean**	
	H401389-H401770	**DWord**, Long	
	H401771-H401B58	**Word**, Short	
	H401771.0-H401771.F...H401B58.0-H401B58.F	**Boolean**	
	H401B59-H401F40	**Float**	
	H401F41-H410000	**Word**, Short	
	H401F41.0-H401F41.F...H410000.0-H410000.F	**Boolean**	

Array Support

Arrays are supported for 16-bit internal and holding register locations for all data types except Boolean and strings. There are two methods of addressing an array. Examples are given using holding register locations.

4xxxx [rows] [cols]
4xxxx [cols] (this method assumes rows is equal to one)

Rows multiplied by cols cannot exceed the block size that has been assigned to the device for the register type. For arrays of 32 bit data types, rows multiplied by cols multiplied by 2 cannot exceed the block size.

SV180 (2.0) Addressing

The Thermo Westronics Ethernet device driver automatically generates tags for many of the most important device parameters. Tags can be manually created to access all device parameters mapped to Modbus address locations within the device. The following table describes how to specify a Modbus address when creating a tag. The full range of addresses accepted by this driver may not be valid for a particular device.

Important: The SV180 (2.0) model is for SV180 devices using firmware version 2.0A or later. If the device uses an earlier firmware version, select the **SV180** model.

Note: The default data types for dynamically defined tags are shown in **bold**.

Modbus Addressing Decimal Format

Address	Range	Data Type	Access
Output Coils	000001-065536	**Boolean**	Read/Write
Input Coils	100001-165536	**Boolean**	Read Only
Internal Registers	NA	NA	NA
Holding Registers	400001-465535*	**Float**	Read/Write

Modbus Addressing Hexadecimal Format

Address	Range	Data Type	Access
Output Coils	H000001-H010000	**Boolean**	Read/Write
Input Coils	H100001-H110000	**Boolean**	Read Only
Internal Registers	NA	NA	NA
Holding Registers	H400001-H410000*	**Float**	Read/Write

*Each value uses 2 registers when the device is configured to use the 2x16 floating point format. For example, Point 1 Data uses 400001 and 400002, Point 2 Data 2 uses 400003 and 400004 and etc. Tags should address the first register used for the value. Each value uses 1 register when the device is configured to use the 1x32 floating point format. For example, Point 1 Data 1 uses 400001, Point 2 Data uses 400002 and etc. For more information, refer to **Settings**.

Array Support

Arrays are supported for 16-bit internal and holding register locations for all data types except Boolean and strings. There are two methods of addressing an array. Examples are given using holding register locations.

4xxxx [rows] [cols]
4xxxx [cols] (this method assumes rows is equal to one)

Rows multiplied by cols cannot exceed the block size that has been assigned to the device for the register type.
For arrays of 32 bit data types, rows multiplied by cols multiplied by 2 cannot exceed the block size.

Error Descriptions

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

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

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

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

Thermo Westronics Serial Device Specific Messages
Bad address in block [<start address> to <end address>] on device '<device name>'
Bad array spanning ['<address>' to '<address>'] on device '<device name>'
Failed to read SM100 decimal placement data

Address Validation

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

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

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

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically 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.

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

Error Type:
Warning

Possible Cause:

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

Solution:
Verify that 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 dynamically 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.

Missing address

Error Type:
Warning

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

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

Serial Communications

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

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

Communications error on '<channel name>' [<error mask>]

Error Type:
Warning

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 communication parameters for the serial connection are incorrect.
3. There is a noise source disrupting communications somewhere in the cabling path between the PC and the device.

Solution:
1. Verify the cabling between the PC and the device.
2. Verify that the specified communication parameters match those of the device.
3. Reroute cabling to avoid sources of electrical interference such as motors, generators or high voltage lines.

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 in the Channel Properties.

COMn is in use by another application

Error Type:
Fatal

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

Solution:
1. Verify that the correct port has been assigned to the channel.
2. Close any other applications that are using the requested COM port.

Error opening COMn

Error Type:
Fatal

Possible Cause:
The specified COM port could not be opened due to 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.

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.

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 communication parameters for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.
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 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.
4. Increase the Request Timeout setting so that the entire response can be handled.

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.

Thermo Westronics Serial Device Specific Messages

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

Thermo Westronics Serial Device Specific Messages
Bad address in block [<start address> to <end address>] on device '<device name>'
Bad array spanning ['<address>' to '<address>'] on device '<device name>'
Failed to read SM100 decimal placement data

Bad address in block [<start address> to <end address>] on device '<device name>'

Error Type:
Serious

Possible Cause:
An attempt has been made to reference a nonexistent location in the specified device.

Solution:
Verify the tags assigned to addresses in the specified range on the device and eliminate ones that reference invalid locations.

Bad array spanning ['<address>' to '<address>'] on device '<device name>'

Error Type:
Fatal

Possible Cause:
An array of addresses was defined that spans past the end of the address space.

Solution:
Verify the size of the device's memory space and then redefine the array length accordingly.

Failed to read SM100 decimal placement data

Error Type:
Serious

Possible Cause:
The driver was not able to read the decimal placement data (registers 405037, 405047, etc.) needed to scale data point values to floats.

Solution:
Check the communication parameters and cabling. Then, restart the server.

Index

A

B

C

D