

Alstom Redundant Ethernet Driver Help

© 2011 Kepware Technologies

Table of Contents

Table of Contents	2
Alstom Redundant Ethernet Driver Help	3
Overview	3
Channel Setup	4
Device Setup	5
Device ID	5
Block Sizes	6
Redundancy Settings	6
Data Types Description	8
Address Descriptions	9
Error Descriptions	10
Modbus Exception Codes	10
Address Validation	11
Address '<address>' is out of range for the specified device or register.....	11
Array size is out of range for address '<address>'.....	11
Array support is not available for the specified address: '<address>'.....	11
Data Type '<type>' is not valid for device address '<address>'.....	12
Device address '<address>' contains a syntax error.....	12
Device address '<address>' is not supported by model '<model name>'.....	12
Device address '<address>' is Read Only.....	12
Missing address.....	12
Device Status Messages	12
Device '<device name>' is not responding.....	13
Unable to bind to adapter: '<network adapter name>'. Connect failed.....	13
Unable to create a socket connection for device '<device name>'.....	13
Unable to write to address '<address>' on device '<device>': Device responded with exception code '<code>'.....	13
Device Specific Messages	13
Bad address in block [x to y] on device '<device name>'.....	14
Bad array spanning ['<address>' to '<address>'] on device '<device name>'.....	14
Bad received length [x to y] on device '<device name>'.....	14
Failure to initiate 'winsock.dll' [Error: <winsock error>].....	14

Alstom Redundant Ethernet Driver Help

Help version 1.002

CONTENTS

[Overview](#)

What is the Alstom Redundant Ethernet Driver?

[Channel Setup](#)

How do I configure channels for use with this driver?

[Device Setup](#)

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

[Data Types Description](#)

What data types does the Alstom Redundant Ethernet Driver support?

[Address Descriptions](#)

How do I reference a data location in a Alstom Redundant Ethernet device?

[Error Descriptions](#)

What error messages does the Alstom Redundant Ethernet Driver produce?

Overview

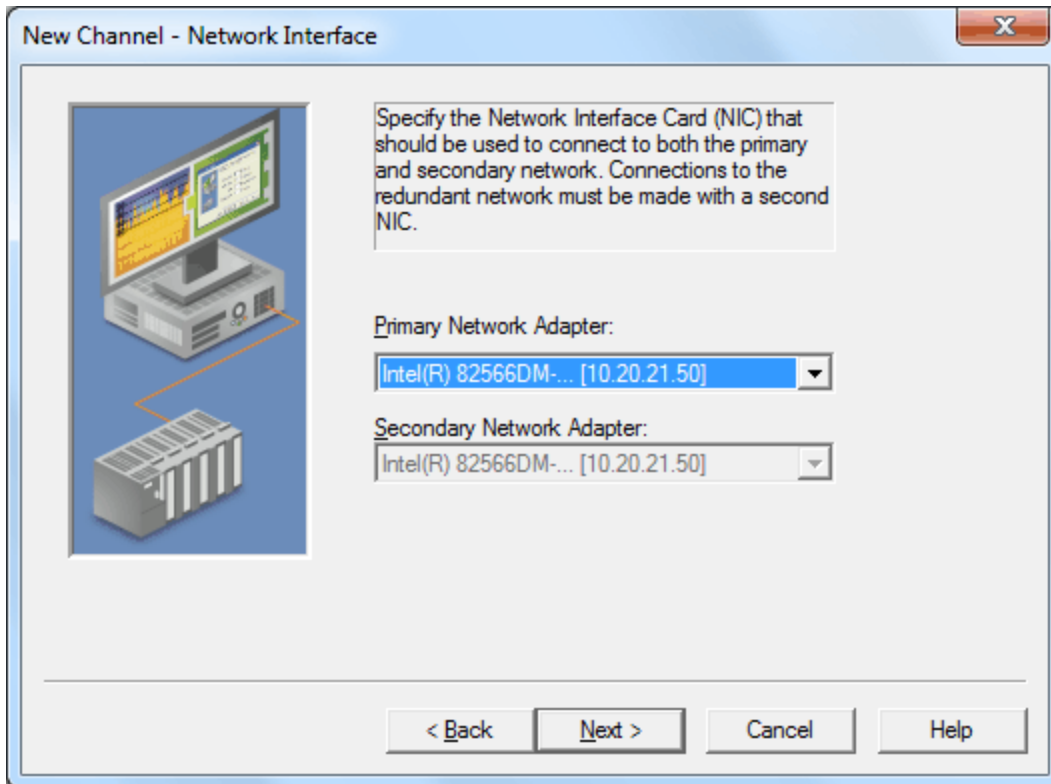
The Alstom Redundant Ethernet Driver provides an easy and reliable way to connect Non-Vital Custom Modbus TCP/IP Protocol with Alstom Redundancy Management Interface devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications.

Channel Setup

Network Interface

When configuring the Alstom Redundant Ethernet Driver, it is recommended that two physical network adapters be used in order to properly utilize the redundant system. The Network Interface dialog specifies the network adapters that will be used for the primary and the secondary networks. The selections available depend on the network configuration settings, the number of unique network adapters that are installed in the PC, and the number of unique IPs that are assigned to the adapters.

Note: The primary and secondary network adapters must be unique.



Note: Changes made to the network configuration will be respected by the server on startup.

Device Setup

Supported Models

Alstom iVPI

Maximum Number of Channels and Devices

The maximum number of channels supported by this driver is 256. The maximum number of devices supported per channel is 100.

Connect Timeout

This parameter specifies the amount of time that the driver will wait for a connection to be made with a device. The connect time may vary with each connection attempt (depending on network load). The default setting is 3 seconds. The valid range is 1 to 30 seconds.

Request Timeout

This parameter specifies the amount of time that the driver will wait for a response from the device before giving up and going on to the next request. Long timeouts will only affect performance if a device is not responding. The valid range is 100 to 30000 milliseconds. The default setting is 1000 milliseconds.

Retry Attempts

This parameter specifies the number of times that the driver will retry a message before giving up and going on to the next message. The valid range is 1 to 10. The default setting is 3 retries.

Auto-Demotion

For information on auto-demotion, refer to the server help file.

Device ID

The Device ID dialog specifies the primary and secondary networks' IP and Port. It consists of four unique IP addresses that are configured in the Non-Vital System Processor (NVSP) boards. The device contains two NVSP boards: Normal and Standby. Each NVSP board has an Ethernet connection to both the primary and secondary networks.

New Device - Device ID

Specify the IP and Port for the Primary and Secondary device. Valid ports for this device are 0 to 65535.

Primary Network

Normal IP:	Port:
255.255.255.255	502
Standby IP:	Port:
255.255.255.255	502

Secondary Network

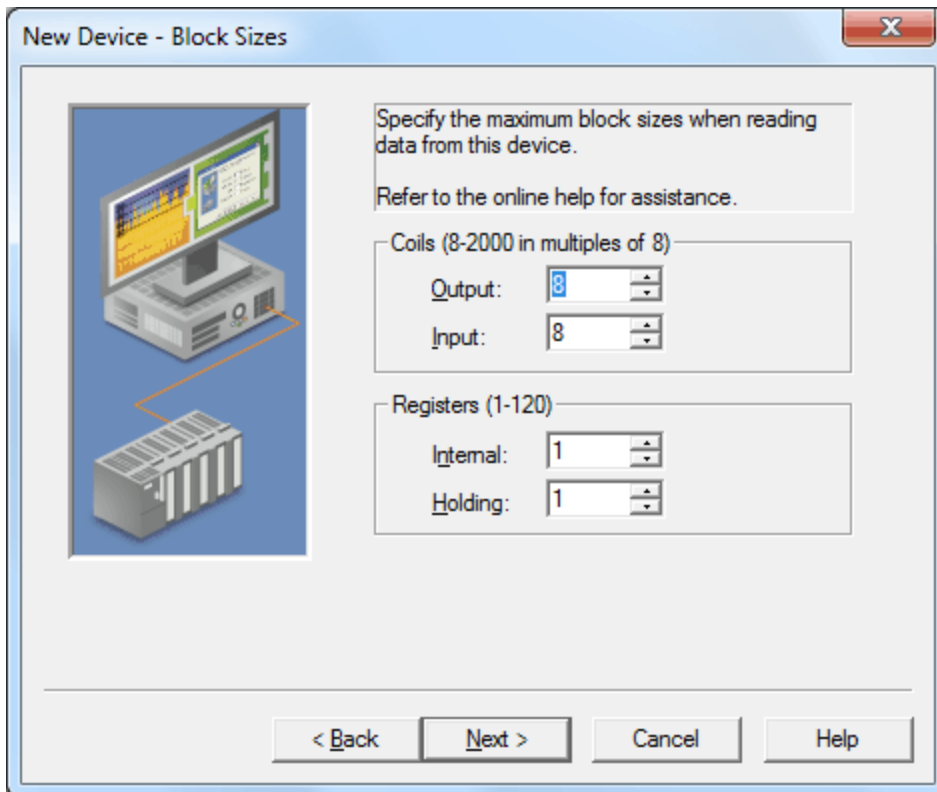
Normal IP:	Port:
255.255.255.255	1502
Standby IP:	Port:
255.255.255.255	1502

< Back Next > Cancel Help

Descriptions of the parameters are as follows:

- **IP:** The Normal and Standby IP parameters specify the IPs that the remote device is configured to use. IPs consist of four values that are separated by periods. The valid range of each value is 0 to 255.
- Note:** IPs must be unique.
- **Port:** This parameter specifies the port number that the remote device is configured to use. The valid range is 0 to 65535. The default setting for the primary network is 502. The default setting for the secondary network is 1502.

Block Sizes

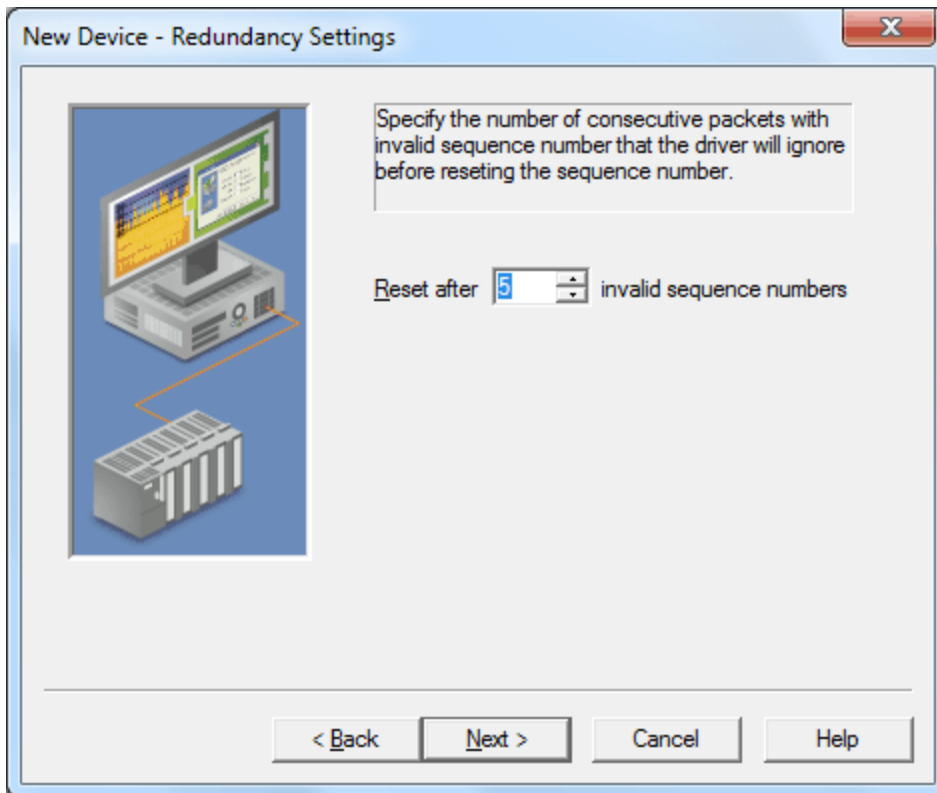


Descriptions of the parameters are as follows:

- **Coils:** This parameter specifies the Output and Input coils' values. Coils can be read from 8 to 2000 points (bits) at a time. The default setting is 8.
- **Registers:** This parameter specifies the Internal and Holding registers' values. Registers can be read from 1 to 120 standard Modbus registers (16 bit) at a time. The default setting is 1.

Redundancy Settings

The communication header utilizes sequence numbers to determine whether the data received is valid.



Description of the parameter is as follows:

- **Reset after x invalid sequence numbers:** This parameter specifies the number of invalid sequences that will be received before the device resets. When an unexpected sequence number is received, subsequent attempts will be made until the reset value is reached. At that point, a request will be issued by the driver to the device to reset the sequence number. Separate sequence numbers will be used for normal and standby boards. The valid range is 1 to 5 attempts. The default setting is 5.

Data Types Description

Data Type	Description
Boolean	Single bit
Word	Unsigned 16 bit value bit 0 is the low bit bit 15 is the high bit
Short	Signed 16 bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32 bit value bit 0 is the low bit bit 31 is the high bit
Long	Signed 32 bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
BCD	Two byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range.
LBCD	Four byte packed BCD Value range is 0-99999999. Behavior is undefined for values beyond this range.
String	Null terminated ASCII string Includes Hi-Lo Lo-Hi byte order selection.
Double*	64 bit floating point value The driver interprets four consecutive registers as a double precision value by making the last two registers the high DWord and the first two registers the low DWord.
Double Example	If register 40001 is specified as a double, bit 0 of register 40001 would be bit 0 of the 64 bit data type and bit 15 of register 40004 would be bit 63 of the 64 bit data type.
Float*	32 bit floating point value The driver interprets two consecutive registers as a single precision value by making the last register the high word and the first register the low word.
Float Example	If register 40001 is specified as a float, bit 0 of register 40001 would be bit 0 of the 32 bit data type and bit 15 of register 40002 would be bit 31 of the 32 bit data type.

*The descriptions assume the default; that is, first DWord low data handling of 64 bit data types and first word low data handling of 32 bit data types.

Address Descriptions

The default data types are shown in **bold**.

Decimal Addressing

Address	Range	Data Type	Access	Address Limit
0xxxx	0-9999	Boolean	Read/Write	800 Bits
1xxxx	0-9999	Boolean	Read Only	800 Bits
3xxxx	0-9999	Word , Short, BCD	Read Only	62 Registers
	0-9998	Float, DWord, Long, LBCD	Read Only	62 Registers
	0-9996	Double	Read Only	62 Registers
4xxxx	0-9999	Word , Short, BCD	Read/Write	62 Registers
	0-9998	Float, DWord, Long, LBCD	Read/Write	62 Registers
	0-9996	Double	Read/Write	62 Registers

Hexadecimal Addressing

Address	Range	Data Type	Access	Address Limit
H0xxxx	0-270F	Boolean	Read/Write	800 Bits
H1xxxx	0-270F	Boolean	Read Only	800 Bits
H3xxxx	0-270F	Word , Short, BCD	Read Only	62 Registers
	0-270E	Float, DWord, Long, LBCD	Read Only	62 Registers
	0-270C	Double	Read Only	62 Registers
H4xxxx	0-270F	Word , Short, BCD	Read/Write	62 Registers
	0-270E	Float, DWord, Long, LBCD	Read/Write	62 Registers
	0-270C	Double	Read/Write	62 Registers

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](#)

Device Status Messages

[Device '<device name>' is not responding](#)

[Unable to bind to adapter: '<network adapter name>'. Connect failed](#)

[Unable to create a socket connection for device '<device name>'](#)

[Unable to write to address '<address>' on device '<device>': Device responded with exception code '<code>'](#)

Device Specific Messages

[Bad address in block \[x to y\] on device '<device name>'](#)

[Bad array spanning \['<address>' to '<address>'\] on device '<device name>'](#)

[Bad received length \[x to y\] on device '<device name>'](#)

[Failure to initiate 'winsock.dll' \[Error: <winsock error>\]](#)

See Also: [Modbus Exception Codes](#)

Modbus Exception Codes

The following data is from Modbus Application Protocol Specifications documentation.

Code Dec/Hex	Name	Meaning
01/0x01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the server (or slave). This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server (or slave) is in the wrong state to process a request of this type; for example, because it is unconfigured and is being asked to return register values.
02/0x02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the server (or slave). More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 would succeed, a request with offset 96 and length 5 will generate exception 02.
03/0x03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for server (or slave). This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the MODBUS protocol is unaware of the significance of any particular value of any particular register.
04/0x04	SLAVE DEVICE FAILURE	An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.
05/0x05	ACKNOWLEDGE	The slave has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the master. The master can next issue a Poll Program Complete message to determine if processing is completed.
06/0x06	SLAVE DEVICE BUSY	The slave is engaged in processing a long duration program command. The master should retransmit the message later when the slave is free.
07/0x07	NEGATIVE ACKNOWLEDGE	The slave cannot perform the program function received in the query. This code is returned for an unsuccessful programming request using function code 13 or 14 decimal. The master should request diagnostic or error information from the slave.

08/0x08	MEMORY PARITY ERROR	The slave attempted to read extended memory but detected a parity error in the memory. The master can retry the request, but service may be required on the slave device.
10/0x0A	GATEWAY PATH UNAVAILABLE	Specialized use in conjunction with gateways indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. This usually means that the gateway is misconfigured or overloaded.
11/0x0B	GATEWAY TARGET DEVICE FAILED TO RESPOND	Specialized use in conjunction with gateways indicates that no response was obtained from the target device. This usually means that the device is not present on the network.

Note: For this driver, the terms Slave and Unsolicited are used interchangeably.

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 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.

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 either a smaller value for the array or a different starting point.

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

Error Type:

Warning

Possible Cause:

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

Solution:

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

Data Type '<type>' is not valid for device address '<address>'

Error Type:

Warning

Possible Cause:

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

Solution:

Modify the requested data type in the client application.

Device address '<address>' contains a syntax error

Error Type:

Warning

Possible Cause:

An invalid tag address has been specified in a dynamic request.

Solution:

Re-enter the address in the client application.

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

Error Type:

Warning

Possible Cause:

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

Solution:

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

Device address '<address>' is Read Only

Error Type:

Warning

Possible Cause:

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

Solution:

Change the access mode in the client application.

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 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 bind to adapter: '<network adapter name>'. Connect failed](#)

Unable to create a socket connection for device '<device name>'**Unable to write to address '<address>' on device '<device>': Device responded with exception code '<code>'****Device '<device name>' is not responding**

Error Type:

Serious

Possible Cause:

1. The connection between the device and the Host PC is broken.
2. The communication parameters for the 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 bind to adapter: '<network adapter name>'. Connect failed

Error Type:

Warning

Possible Cause:

Since the specified network adapter cannot be located in the system device list, it cannot be bound to for communications. This usually occurs when a project is moved from one PC to another (and when the project specifies a network adapter rather than using the default). The server falls back to the default adapter.

Solution:

Change the Network Adapter property to Default (or select a new adapter) and then save the project and retry.

Unable to create a socket connection for device '<device name>'

Error Type:

Warning

Possible Cause:

The server was unable to establish a TCP/IP socket connection to the specified device. It will continue to attempt connection.

Solution:

1. Verify that the device is online.
2. Verify that the device IP is within the subnet of the IP to which the server is bound. Alternatively, verify that a valid gateway is available that allows a connection the other network.

Unable to write to address '<address>' on device '<device>': Device responded with exception code '<code>'

Error Type:

Warning

Possible Cause:

See [Modbus Exception Codes](#) for a description of the exception code.

Solution:

See [Modbus Exception Codes](#).

Device Specific Messages

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

Device Specific Messages[Bad address in block \[x to y\] on device '<device name>'](#)[Bad array spanning \['<address>' to '<address>'\] on device '<device name>'](#)

[Bad received length \[x to y\] on device '<device name>'](#)
[Failure to initiate 'winsock.dll' \[Error: <winsock error>\]](#)

Bad address in block [x to y] on device '<device name>'

Error Type:

Fatal addresses falling in this block.

Possible Cause:

This error is reported when the driver attempts to read a location in a PLC that does not exist. For example, in a PLC that only has holding registers 40001 to 41400, requesting address 41405 would generate this error. Once this error is generated, the driver will not request the specified block of data from the PLC again. Any other addresses being requested that are in this same block will also go invalid.

Solution:

The client application should be modified to ask for addresses within the range of the device.

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

Error Type:

Fatal

Possible Cause:

An array of addresses was defined that extends past the end of the address space.

Solution:

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

Bad received length [x to y] on device '<device name>'

Error Type:

Fatal addresses falling in this block.

Possible Cause:

The driver attempted to read a block of memory in the PLC. The PLC responded with no error, but did not provide the driver with the requested block size of data.

Solution:

Ensure that the range of memory exists for the PLC.

Failure to initiate 'winsock.dll' [Error: <winsock error>]

Error Type:

Fatal

Possible Cause:

Could not negotiate with the operating system's Winsock 1.1 functionality.

Solution:

Verify that the winsock.dll is properly installed on the system.